Knowledge Extraction

in the Quantum Random-Oracle Model

Jelle Don

Knowledge Extraction in the Quantum Random-Oracle Model

Proefschrift

ter verkrijging van
de graad van Doctor aan de Universiteit Leiden,
op gezag van Rector Magnificus prof.dr.ir. H. Bijl,
volgens besluit van het College voor Promoties
te verdedigen op *dinsdag 23 januari 2024*
klokke *15:00* uur

door **Jelle Wijnand Don**

geboren te *Amsterdam*

in 1990

Promotores:

Prof.dr. S.O. Fehr	(CWI Amsterdam & Universiteit Leiden)
Prof.dr. R.J.F. Cramer	(CWI Amsterdam & Universiteit Leiden)

Promotiecomissie:

dr. K. de Boer	(Universiteit Leiden)
dr. A. Broadbent	(University of Ottowa)
Prof.dr.ir. G.L.A. Derks	(Universiteit Leiden)
Prof.dr. L. Ducas	(Universiteit Leiden)
Prof.dr. V. Dunjko	(Universiteit Leiden)
dr. N. Spooner	(University of Warwick)
Prof.dr. D. Unruh	(University of Tartu & RWTH Aachen University)

The research in this thesis was carried out in the Cryptology Group at Centrum Wiskunde & Informatica, Amsterdam. The position was funded by the ERC Advanced Grant 740972 (ALGSTRONGCRYPTO).

Jelle Wijnand Don

Knowledge Extraction in the Quantum Random-Oracle Model

Nature isn’t classical, dammit, and if you want to make a simulation of nature, you’d better make it quantum mechanical, and by golly it’s a wonderful problem, because it doesn’t look so easy.

Richard P. Feynman, Nobel laureate in Physics

Illustration frontcover: ‘Priestess of Delphi´ by John Collier (1891)
Illustration backcover: Created by the author using AI (Nightcafe)

Table of Contents

Chapter 1

Introduction

Chapter contents

Section 1.1

Cryptography

Cryptography used to be the skill of hiding information in plain view. For thousands of years people have come up with ways to *encrypt* their communications, so that a sender would not have to rely on the message bearer (letter, telegraph line or radio wave packet) to remain hidden for its contents to still be inaccessible by an eavesdropper. Hence even if the messenger was compromised, as long as the employed method of encryption was *not*, no adversary would be able to infringe on privacy – preventing, for example, the discovery of a planned military maneuver.

A drawback was that sender and receiver had to agree on a secret method of encryption beforehand, and make sure it did not leak to the enemy. This task could be simplified if all secrecy in the scheme was focused in a short, easily substituted *secret key*, as advocated by the 19th century Dutch cryptographer Auguste Kerckhoffs [Ker83] (an idea now known as Kerckhoffs' principle). Nonetheless, the secret key had to be covertly distributed *before* encrypted communication could take place – a system that in some contexts would defeat its own purpose.

In particular, the rapidly expanding possibilities of 20th century telecommunication technologies begged for a solution to this problem. If every soldier on the battlefield could be equipped with its own communication device, key management would quickly become infeasible (unless they would all use the same set of keys, as for example the German naval ships did in WWII, but that would explode the likelihood of keys leaking). Similarly, there would simply be no way for private users on the emerging internet to pre-share a key with every merchant/bank/government agency with whom they wished to exchange private data.

The solution was independently found by two groups of researchers in the seventies.[1] *Public key encryption* (PKE) makes use of an asymmetry in so-called 'trapdoor one-way functions': Mathematical functions that are easy to compute

[1] While James Ellis, Clifford Cocks and Malcolm Williamson are now credited to be the first to develop public key encryption, they were forced by their employer (the British Government Communications Headquarters) to keep their discoveries secret until 1997 [Ell87]. In the meantime and independently, Martin Hellman, Whitfield Diffie, Ralph Merkle, Ron Rivest, Adi Shamir and Leonard Adleman had developed and published roughly the same concepts by 1977 [Mer78; DH76; RSA78].

but hard to invert, unless one knows the secret trapdoor. This structure is used to generate pairs of public and secret keys, such that anyone can use the public key to encrypt a message, but only the holder of the corresponding secret key can decrypt the resulting ciphertext. Communicating parties can then simply announce their public keys *in the open* the moment the conversation is initiated, since intercepting the public part of the key does not help an eavesdropper to decrypt.

There is a catch to the last remark; if an adversary can intercept a public key *and* prevent it from reaching the other party (the one who intends to use it for encryption, i.e. the 'sender'), he can attempt a 'man-in-the-middle' attack. In such an attack the adversary impersonates the sender by substituting his own public key for the original before passing it on to the receiver, so that all encrypted messages become transparent to him. If he pretends to be B to A and A to B, he can relay and inspect an entire conversation without the participants noticing.

Thus, for public key encryption to work in practice, some kind of authentication mechanism is required. While the notion of message authentication codes (MACs) had been around for some time, these again require sender and receiver to share a key beforehand. Instead, Diffie and Hellman proposed the notion of a Digital Signature Scheme (DSS) [DH76], which can protect the integrity of a message and tie it to a specific *public* key both at once. Indeed, signature schemes follow an asymmetric key pattern similar to PKE: A public key can be used to verify a signature, while no one except the holder of the corresponding secret key is able to sign messages. Of course this only defers the issue to proving authenticity of a party's DSS public key, but now a trusted institute (called a Certificate Authority) can use *their* signature to authenticate a list of users and their public keys. Different CA's can verify each other, and a few roots in this so-called 'web of trust' may for example have their public keys hardwired in an internet browser.

PKE and signatures solved the problem of secure communication over an insecure channel. In the current information age however, external eavesdroppers are not the only adversaries who threaten the security of complex interactions. Sometimes sensitive data will have to be protected even against parties who are authorized to participate in the interaction.

An example of a modern cryptographic task is the following: '*let several banks run a money laundering detection procedure on their collective transaction data, each of them learning nothing more than the outcome of the procedure*'. A naive protocol for money laundering detection would have the banks transfer their transaction data to each other so that each of them could compute the protocol locally, but this would violate the privacy of their clients. Instead, the rich field of *multi-party computation* provides cryptographic solutions for problems like the above, allowing for a distributed computation of which only the outcome is revealed, so that the different parties cannot learn each others' inputs.

Another example – which will feature prominently in this thesis – of cryptography that protects against dishonest insiders, is the concept of a *zero-knowledge proof*. Introduced by Shafi Goldwasser, Silvio Micali and Charles Rackoff [GMR85] in 1985, a zero-knowledge proof is an interactive protocol designed to let a *prover* that holds a witness w to some statement x in an NP-relation, convince a *verifier* of the truth of x. The challenge (and the reason for the name 'zero-knowledge') is to do so while revealing *nothing* (zero knowledge) except for the fact that x is true (in particular, the witness should not be revealed). In 1991 [GMW91] it was shown that every NP-language admits a zero-knowledge proof system.

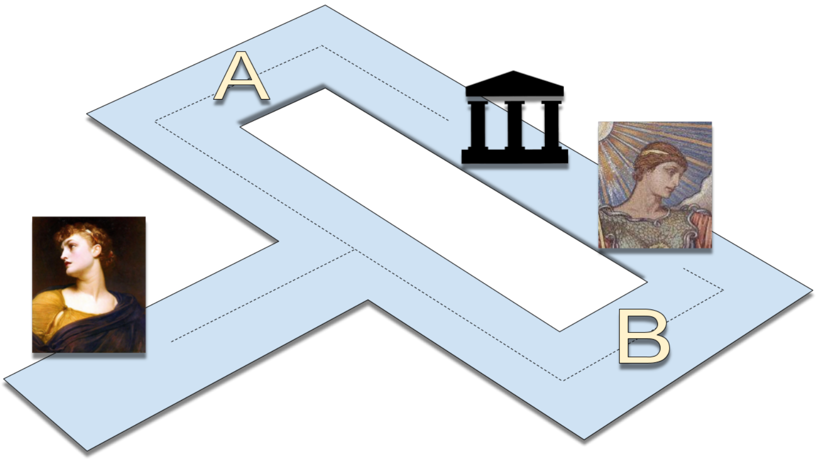

Fig. 1.1. Antigone and Broteas in a cave. Broteas proves to Antigone that he has the key to the temple – without showing her the key – by following the procedure of a *zero-knowledge proof*.

Figure 1.1 depicts a famous way of explaining the concept of a zero-knowledge proof. Broteas would like to prove to Antigone that he possesses the secret key to a temple inside a cave, without showing her the key itself. A good strategy is for Broteas to demonstrate his possession of the key by using it to solve some challenge that could otherwise not be solved. They come up with the following procedure: Antigone looks away while Broteas enters the cave through either one of the entrances (A or B). Antigone then shouts at Broteas which of the sides he should emerge from (chosen at random). If it happens to be the side that Broteas entered, then he does not need to use his key. But, if she wants him to emerge from the opposite entrance, then Broteas has to use his key to move through the temple in order to get to the other side. Should he not possess the key after all, then he will not be able to emerge from the correct side and hence get caught. By repeating the process n times, the probability that a cheating Broteas is not caught becomes 2^{-n}.

It may not be immediately clear why Antigone has to look away when Broteas enters the cave, or why she doesn't just ask him to walk from A to B, demonstrating his possession of the key beyond any doubt. The zero-knowledge property actually forces us to leave *some* room for cheating. To prove that no additional information is leaked, the procedure is set up in such a way that the participants can choose to simulate (fake) it without any outside observer noticing. In our example, Antigone and Broteas may collude by agreeing on a fixed order of A-B-challenges prior to execution, so that Broteas can always enter from the right direction. That way, just having a transcript of the procedure (say, the notes of an observer standing with Antigone) *cannot possibly* leak any information on the key, since it may just as well have been produced without the involvement of any key whatsoever! The nice thing is that Antigone, by the fact that she knows she did not collude with Broteas (and only she can know!), still has good reason to be convinced of Broteas' possession of the key.

In a more formal setting, when proving zero-knowledge for actual cryptographic protocols, we have to show the existence of an efficient *simulator* that can – without knowing the key – create accepting transcripts (a list of exchanged messages that lead to the verifier accepting a statement) that are indistinguishable from real executions (where the prover does use its key/witness to produce its messages).

As we have seen, modern cryptography gives us the tools to specify exact requirements for privacy, correctness, integrity and authenticity against any adversarial party in any interaction. In the next section we show how these requirements are formalized in rigorous security notions, and how we may attempt to mathematically prove them for any candidate scheme. Cryptography has thus evolved from the *skill* of hiding information in plain view to the *science* of secure communication in the broadest sense.

1.1.1 Provable security

A mathematical statement is an assertion about well-defined, unambiguous concepts. So whenever we want to apply the rigorous tools of mathematics to cryptography, we first have to decide how to model the different parties and their interactions (both honest and dishonest) that will take place in an application of our scheme. It is important to realize that a formal security proof can only support our confidence in a scheme's practical security if the model adequately represents the context and security requirements of the application.

One aspect to determine is the power of the adversary, the so-called attack model. Do we allow the adversary to choose the instance (e.g. ciphertext, identity of whose signature to forge) himself, or does it have to break the scheme on any given target? Does it have unlimited computational resources, or perhaps a bounded memory? Can it access some example broken instances different from its target, to learn from? If yes, does it get to choose which example instances?

The more power and freedom we grant the adversary in our attack-model, the stronger will be the claims of security that we derive. On the downside, such claims will also be harder to prove, and they might be overkill for applications where malicious parties are constrained by the context (e.g. it only pays off to inconspicuously alter the amount on your *own* savings account).

As an example, consider the case of an encryption scheme. At first glance, we simply desire that an adversary should not be able to decrypt a given ciphertext, i.e. figure out the corresponding encrypted plaintext. However, not being able to decrypt does not imply that the adversary is unable to extract any useful information from a ciphertext. In an encrypted voting system he might not be able to determine for which particular candidate a vote is cast, but still distinguish between votes for candidates from different political parties and thus influence the election outcome by discarding votes of a certain kind.

In other contexts, an attacker might be able to influence what messages are sent over an encrypted channel and hence obtain some knowledge on the relation between plaintexts and ciphertexts that can help him decrypt. A famous example comes from WWII, where the US knew from a partially decrypted ciphertext that Japan was planning an attack against 'AF', and suspected this to be an encryption of 'Midway Island'. By leaking a fake message about Midway Island, they observed the Japanese immediately reporting about AF again and thus confirmed their suspicion and were able to deflect the attack by sending reinforcements.

To model such and other circumstances that are potentially beneficial to the adversary, we define security in terms of abstract *games*. In a cryptographic game, an adversary goes through a series of computations and interactions with a *challenger* before producing an output. The adversary wins the game if its output satisfies a certain predicate. We usually say that a scheme satisfies a given security notion if any adversary (from the class considered in the attack-model) can win the corresponding game with at most a negligible success probability (respectively a negligible *advantage* – the difference between the success probability of the adversary and a random guess – in case of games where guessing the right outcome is feasible). Figure 1.1.1 depicts the game for the strongest security notion for public-key encryption: Indistinguiability under Chosen-Message Attack (IND-CCA). In some cases we intentionally move away from reality by letting the parties in our model interact with an *idealized* version of a certain *cryptographic primitive*. A primitive is a low-level algorithm like a signature scheme or a hash function, that can be used as a building block for high-level cryptographic protocols. In the idealized version we consider the primitive to be given as an *oracle*, i.e. as a black-box outside the control of any party. Parties are given only *query access*, meaning that they can ask for an output to a corresponding input of their choice, at unit cost per query. The functionality implemented by the oracle is an ideal version that would be impossible to implement in practice, with properties that the real-world primitive seeks to approximate.

```
GAME IND-CCA
1: $(pk, sk) \leftarrow \mathsf{Gen}$
2: $m_0, m_1 \leftarrow \mathcal{A}^{\textsc{Decrypt}}(pk)$
3: $b \xleftarrow{\$} \{0, 1\}$
4: $c^* \leftarrow \mathsf{Enc}(pk, m_b)$
5: $b' \leftarrow \mathcal{A}^{\textsc{Decrypt}}(c^*)$
6: return $b' == b$

DECRYPT($c \neq c^*$)
7: $m := Dec(sk, c)$
8: return $m$
```

Fig. 1.2. Definition of the cryptographic game IND-CCA for a public-key encryption scheme (Gen, Enc, *Dec*) using *pseudocode*. The adversary $\mathcal{A}$ has access to a *decryption oracle*, specified in lines 7-8. The challenge ciphertext c^* is not allowed as an input to the decryption oracle.

The reason for this abstraction is simply that it allows some constructions to be proven secure that would otherwise (in a model with the normal, non-idealized primitive, also known as the 'plain model') lack a formal justification. These constructions are often more simple (efficient) and elegant compared to the schemes that are provably secure in the plain model. On the downside, the gap between the real world and the idealized model leaves open the possibility that a scheme proven secure in the latter can actually be broken in practice. However, in spite of this apparent weakness of the method, we actually have enormous evidence of its soundness in specific cases.

Consider the case of an idealized hash function, where the corresponding model is known as the Random-Oracle Model (ROM) [BR93]. The 'Random Oracle' implements a perfectly random function, i.e. for every input an output is freshly sampled from a uniform distribution. A separation from the plain model was indeed proven by Canetti, Goldreich and Halevi, who constructed a protocol secure in the ROM which can nonetheless be shown *in*secure if the random oracle is instantiated with any concrete hash function [CGH04]. Their construction however is very artificial and goes against all sound practices of constructing cryptographic protocols (i.e. a certain message triggers the protocol to output its secret key). In a 'twenty year retrospective'[KM15], Neal Koblitz and Alfred Menezes concluded that "no real-world protocol failures have been found that result from the use of random oracles" and their "belief in the random-oracle [model] is unshaken".

Having fixed a model and a security notion, we proceed to mathematically prove that a certain scheme satisfies the latter in the chosen model. In many cases, these proofs are only conditional, depending on the assumed hardness of some computational problem – for example, the *computational assumption* that inverting a particular trapdoor one-way function without the trapdoor is unsolvable for an adversary with bounded resources (time and memory). The proof is given as a reduction; a hypothetical attacker that supposedly breaks the security notion is turned into an algorithm that solves the computational problem. If the assumption holds true, such an algorithm – hence attacker against the scheme – cannot exist.

Security reductions may be seen as a roadmap for cryptanalysts, pointing to a few clear-cut, well defined computational problems that they can try to attack, instead of the myriad of cryptographic schemes that are based on them. If the computational problem has resisted attack from focused effort by cryptanalysts for many years, then we can be reasonably confident in the truth of the assumption, and by extension (through a reduction) of the security of any related scheme.

Occasionally a computational assumption does get broken, usually one of the newer ones that had not yet been studied extensively. It also occurs that a new attack forces cryptographers to increase the parameters of their schemes, because an assumed hard problem turns out to be not as hard as expected, but still hard enough. In this case increasing the instance size of the problem brings the security back to level. It is quite a *unique* situation when an advancement in technology forces us to consider a whole new model of computation, opening up a new class of adversaries for which the existing computational assumptions may or may not hold. This is exactly the situation in which we have found ourselves since the advent of the quantum computer.

1.1.2 The threat of quantum computing

In 1981 physicist Richard Feynman suggested that computing devices based on the laws of quantum mechanics would be capable of outperforming any ‘classical’ computer, at least on some specific computational tasks. A fundamental difference between a classical state (describing for example the position and momentum of a coffee mug) and a quantum state (describing position and momentum of an electron), is that the latter may describe a *superposition* of many distinct positions/momenta. Only when the actual position or momentum or any other property of the quantum system is *measured* (observed) does it ‘collapse’ to a single determinate value.[2] Much like with waves in classical physics, superposing (adding together) two quantum states can lead to interference patterns between those states, and this interference is what a quantum computer exploits to find a solution faster than any classical machine (for ‘suitable’ problems). On a very high level, all quantum algorithms compute some function on a superposition of input bits – in a sense computing the function on all possible inputs simultaneously – then manipulate the resulting quantum state in such a way that interference (mostly) cancels out the outcomes that are not a solution, and then measure the state to (hopefully) find a bitstring that *is* a solution.

Exactly what computational problems are well-suited to quantum computing is an ongoing research question. By the early nineties David Deutsch, Richard Josza and Daniel Simon had at least proven Feynman’s general idea correct, by giving the first explicit quantum algorithms that (under certain constraints) perform provably better than their classical counterparts [DJ92; Sim97]. The big blow to cryptography came in 1994, when Peter Shor published a quantum algorithm for factoring large integers, which runs superpolynomially

[2] Incidentally, this collapse to a determinate value causes certain other properties to become *in*determinate, a phenomennon known as ‘Heisenbergs Uncertainty Principle’, which for example causes the position and momentum of an electron to never both be known for the same time t. This feature of quantum mechanics however only distracts from the point here.

faster than the best known classical algorithm, potentially breaking the RSA cryptosystem in seconds rather than millions of years [Sho94].

Interestingly, all of these theoretical results came about before any actual quantum computer was built, and therefore the threat was not immediate. Even today, the first quantum computers that are under development are still very far removed from the scale (in terms of memory and processing capabilities) required to implement Shor's algorithm on an actual RSA instance. However, the point remains that if large-scale quantum computers can be built at some point in the future, cryptosystems based on factoring will become vulnerable.

The break by Shor's algorithm affects not only RSA, but practically all of public key cryptography in use today. Symmetric schemes suffer from a quantum attack as well – via Grover's 1996 algorithm, which gives a quadratic speedup in unstructered search, hence a quadratic speedup in guessing the key (if it can be verified) – but here we have an example of a security breach that can be remedied by increasing the parameters, in this case by doubling the key size.

In 2016, the American National Institute of Standards and Technology announced a competition for new standards in cryptography for schemes that can withstand attacks from a quantum attacker. Indeed, while no cryptosystem used in practice has been broken by a quantum computer to date, there are good reasons to start thinking about more robust solutions sooner rather than later. The development of large-scale quantum computers seems to be 'merely' an engineering challenge, no fundamental law of physics that rules them out has been discovered yet. It may well take another 10, 20 or 30 years before the threat becomes actual, but the transition to new cryptosystems has a long time span as well. Apart from the logistical difficulty inherent in any big IT-migration, there are some specific factors that come into play here.

Firstly, there is the issue of *long term security.* An adversary could obtain some classified, encrypted data now, and break the encryption at a point in the future far enough that quantum computers have become available, yet near enough for the data to still be relevant. Secondly, the life cycle of products has to be taken into account. Expensive satellites that are launched today will ideally still be operational 30 years into the future, without the possibility of doing any cryptographic updates in the meantime. Lastly, the new cryptosystems, robust against quantum attacks, will have to rely on new computational problems that have not been studied as extensively as e.g. factoring. It takes many years of cryptanalysis to build trust in new computational assumptions and cryptoschemes.

1.1.3 Post-quantum cryptography

The study of cryptosystems that are robust against quantum attacks is known as *post-quantum cryptography.* It involves more than just finding computational problems for which no efficient quantum algorithm is known, though of course such new problems are an essential first step.

New computational assumptions. There are currently six main branches of computational problems that are assumed to be hard even for quantum algorithms and that lend themselves for public-key cryptography, giving rise to: lattice-based, isogeny-based, code-based, multi-variate, symmetric primitive and hash-based cryptography. Their respective merits are weighed along the vectors of efficiency – in terms of key and ciphertext/signature sizes as well as speed of basic operations – and how mature the cryptanalysis is to which they have been subject. See e.g. [RCB22] for an overview.

Provable security in a quantum world. Remember that in a security reduction, we start with a hypothetical adversary against the considered scheme and turn it into an algorithm that solves an assumed-to-be-hard computational problem. Just replacing the computational problem with one of the above (which are assumed to be hard even for quantum adversaries) is not enough to achieve post-quantum security. We need to make sure that the adaptation from an adversary into a solver still goes through when the adversary is a quantum algorithm. Indeed, some techniques we use in the classical case break down when considering quantum adversaries.

Two issues that are typical to obstruct a reduction in the quantum setting are the following: When observing an intermediate output of the considered quantum adversary – which may be a quantum state – a fundamental feature of quantum mechanics called *Born's rule* says that the observation potentially alters (disturbs) the state of the adversary, making it in general hard to predict its behavior after this point. Secondly, the *no-cloning principle* of quantum information prevents us from copying the initial state of the adversary, so that after a potential disturbance caused by an observation (called a 'measurement'), we cannot simply go back to the start and run it again, as is done in the common technique of *rewinding* in classical reductions.

A major goal of this thesis will be to remedy these issues, either by finding a way around them or by directly controlling the disturbance caused by a measurement in the reduction strategy.

The Quantum Random-Oracle Model. In 2011, [BDF+11] noted that in the case of an idealized hash function, the classical query model does not adequately reflect the capabilities of a quantum adversary. A quantum adversary can evaluate the real-world hash function on a superposition of inputs, potentially learning global information on the oracle function already in a single query. The Deutsch-Josza algorithm for example [DJ92] shows how a single superposition query can be exploited to extract information about the overall structure of an oracle function. Thus, to match this extra power of quantum algorithms in the real world, [BDF+11] proposed the Quantum Random-Oracle Model as the correct abstraction for idealized hash functions, where parties are given superposition query access to the random oracle.

The enhanced query access introduces three main difficulties for proving security reductions, compared to the classical ROM.

- *Efficient simulation.* A random oracle is infeasible to implement in practice, due to the exponential size of the function table that describes the oracle function (a truly random function cannot be compressed). In the classical ROM, the reduction can however *simulate* the random oracle to the adversary, by means of lazy sampling; simply picking a random output for each incoming query, and writing down each pair to ensure that future queries can be answered consistently.[3] It is straightforward to show that the adversary cannot distinguish this procedure from a true random oracle. With superposition queries, standard lazy sampling breaks down because the first query may already require the sampling of exponentially many values. The work of Zhandry [Zha12; Zha19a] offers two distinct solutions to this problem, see Section 2.4 for further details on the latter result.

- *Query Recording.* A key aspect of the classical ROM is query recording, the fact that the reduction can place itself between the adversary and the oracle (or simulate it) in order to find out which inputs the adversary has queried so far at any time during its execution. In the QROM, the potential disturbance caused by the measurement of a quantum state forces us to analyze the potentially altered behavior of the adversary after every sneak peak we take at its quantum queries. In general, the adversary may notice and simply choose to abort, rendering itself useless for our reduction purposes. In Chapter 3 of this thesis we show how to control the disturbance caused by

[3] The reason for wanting to simulate the oracle can be either because we need to reduce to some hardness assumption in the plain model and thus have to get rid of the oracle, or because we want to use the simulated oracle for query recording (next bullet).

measurements of the adversary's queries in certain circumstances, bounding it by a moderate factor in the total number of queries. Another approach – called the 'compressed oracle framework' – originates in [Zha19a], where a quantum version of the lazy sampling technique is given, that allows to keep a list of previous queries in superposition.[4] Measuring this list during execution may still cause disturbance to the adversary, which is why we extend the framework in Chapter 4 to again control the disturbance in a particular setting. Other works have extended the framework in different ways [CMS19; CFHL21] and we build upon the latter to prove new results in Chapter 5.

- *(Adaptive) reprogramming.* Another very useful technique in the (Q)ROM is known as 'reprogramming the oracle'. It allows a reduction to change some values of the oracle function, which can be easily done when using lazy sampling or by constructing an interface between the adversary and the oracle. The challenge however is to do it undetectably, and this is more difficult in the quantum setting. There are roughly three scenario's to discern; where the adversary has no, partial or full control of the reprogramming point (i.e. the input for which we want to change the output value). In the first case we can relate the probability of detection to the probability that we find the reprogramming point(s) in a measurement of one of the adversary's queries ('O2H lemma' [Unr14a]). Still in the first case, another thing we can do is choose a λ fraction of inputs and reprogram them to a single random looking value y, from the very start of the execution. Zhandry showed in [Zha15b] that a QROM adversary can distinguish the resulting distribution from a random oracle with probability $O(q^4\lambda^2)$, where q is the number of queries made by the adversary.[5] *Adaptive* reprogramming refers to the second case, where the point of reprogramming is controlled partially by the adversary (for example, it is determined by a query the adversary makes to a different oracle) and partially random – respectively computationally indistinguishable from random – in the view of the adversary. Here [GHHM21] and [ES15] give QROM bounds for distinguishing in the respective settings. Third is the case where the adversary has *full* control over the reprogramming point, such as when it may choose any input of the random oracle as a basis for its forgery of a signature, and the reduction needs the forgery to include a specific output value. The measure-and-reprogram technique,

[4] See Section 2.4 for an in-depth explanation of the compressed oracle technique.

[5] 'Big-O notation' such as $O(q^4\lambda^2)$ can be interpreted as 'at most roughly $q^4\lambda^2$, see Section 2.1.3 for a precise definition.

introduced in Chapter 3, is the first to solve this case for the QROM, at an $O(q^{2n})$ multiplicative loss for n reprogramming points.

Since we presented the above list of technical difficulties with corresponding solutions (albeit with some slightly worse bounds), the reader may be led to think that every ROM proof has a corresponding QROM proof, hence there is no fundamental difference in security between the two. Previous works [BDF+11; ARU14; YZ21] however established a separation; there exist schemes secure in the ROM that are demonstrably insecure in the QROM. This result strengthens our motivation to further develop tools for QROM analysis.

The NIST competition. As mentioned in the previous section, the American National Institute of Standards and Technology started a competition for post-quantum cryptography in 2016. In a call for proposals, teams of cryptographers were invited to give it their best shot at designing Key Encapsulation Mechanism's (KEM's) and Digital Signature Schemes (DSS). Among the candidates were several schemes that benefited from results in this thesis, in the sense that our tools allow for a more tight (Dilithium for NIST parameters) or even any (MQDSS) QROM analysis at all, or by removing the need for alterations to the schemes that facilitate a security reduction (Unruh transformation for Picnic, ‘key confirmation hash’ for practically all KEM's, i.e. the ones that use the Fujisaki-Okamoto transform).[6]

In 2022, NIST selected the lattice-based signature candidates Diltihium and Falcon as well as the hash-based signature scheme Sphincs+ to be standardized. On the KEM side, only the lattice-based scheme Kyber was selected. Some other candidates will continue to be studied and possibly standardized in the future, to allow for more diversity in the underlying computational problems.

Summary and outlook: *Knowledge Extraction in the QROM.* In modern cryptography we aim for provable security; we define rigorous notions of security that capture our intuition of what it means to have particular complex interactions that are private, authenticated, untampered and correct (i.e. achieve what they claim to achieve) – and then mathematically prove that certain candidate schemes (conditionally) satisfy these notions, for a given attack model. Often the proofs come in the form of security reductions, where we take

[6] There have been other QROM anlyses that do not require a key confirmation hash, but only for the ‘implicit rejection’ variant of the FO transform. While the NIST candidates may use implicit rejection, the advantage of an explicit rejection analysis is that it still holds in the presence of a potential side-channel attack that can detect rejection.

a hypothetical adversary that breaks a scheme, and turn it into an algorithm that solves some assumed-to-be-hard computational problem. Here we want to treat adversaries as a black-box as much as possible, because the less assumptions we make on *how* they attack our schemes, the more general will be the security claims we make based on reductions involving those adversaries. The random-oracle model however helps us break open these boxes just a little bit, by externalizing the evaluation of the (idealized) hash function, in giving the adversary only query access to a random oracle. This in turn will help us deduce what knowledge the adversary must necessarily posses whenever it succeeds in some particular objective.

For example, in the Fujisaki-Okamoto transform ([FO99], Section 4.6), an adversary cannot (up to negligible probability) query a valid ciphertext to the decryption oracle without first having queried the corresponding plaintext to the random oracle, allowing the reduction to extract its knowledge of the plaintext via query recording. In zero-knowledge proofs, the existence of a 'knowledge extractor' that can extract a witness from any successful prover is required for the 'proof-of-knowledge' property (which intuitively says that no prover can succeed without knowing a witness). Such an extractor does not contradict the zero-knowledge property because we give it more power than a normal verifier; in the random-oracle model this power consists of being allowed to observe the adversary's queries.

Interestingly, in the quantum world, taking a look inside the adversary's mind unavoidably alters its computation process. Whether this hurts our reduction goals – and if so by how much – is the common theme in the research questions of this thesis.

Section 1.2
Outline and contributions of this thesis

In this dissertation, we will introduce new techniques that facilitate security reductions in the quantum random-oracle model. We present them alongside a number of applications, obtaining rigorous relations between the security of post-quantum cryptosystems and the hardness of certain computational problems. The main chapters are based on the following papers:

[DFMS19] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. "Security of the Fiat-Shamir Transformation in the Quantum Random-Oracle Model". In: *Advances in Cryptology – CRYPTO 2019*. Ed. by Alexandra Boldyreva and Daniele Micciancio. Cham: Springer International Publishing, 2019, pp. 356–383.

[DFM20] Jelle Don, Serge Fehr, and Christian Majenz. "The Measure-and-Reprogram Technique 2.0: Multi-round Fiat-Shamir and More". In: *Advances in Cryptology – CRYPTO 2020*. Ed. by Daniele Micciancio and Thomas Ristenpart. Cham: Springer International Publishing, 2020, pp. 602–631.

[DFMS22a] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. "Online-Extractability in the Quantum Random-Oracle Model". In: *Advances in Cryptology – EUROCRYPT 2022*. Ed. by Orr Dunkelman and Stefan Dziembowski. Cham: Springer International Publishing, 2022, pp. 677–706.

[DFMS22b] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. "Efficient NIZKs and Signatures from Commit-and-Open Protocols in the QROM". In: *Advances in Cryptology – CRYPTO 2022*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Cham: Springer Nature Switzerland, 2022, pp. 729–757.

In the course of his PhD, the author has additionally co-authored the following papers, which are not included in this thesis:

[DFH22] Jelle Don, Serge Fehr, and Yu-Hsuan Huang. "Adaptive Versus Static Multi-oracle Algorithms, and Quantum Security of a Split-Key PRF". In: *Theory of Cryptography*. Ed. by Eike Kiltz and Vinod Vaikuntanathan. Cham: Springer Nature Switzerland, 2022, pp. 33–51. ISBN: 978-3-031-22318-1.

[BBD+23] Manuel Barbosa, Gilles Barthe, Christian Doczkal, Jelle Don, Serge Fehr, Benjamin Grégoire, Yu-Hsuan Huang, Andreas Hülsing, Yi Lee, and Xiaodi Wu. “Fixing and Mechanizing the Security Proof of Fiat-Shamir with Aborts and Dilithium”. In: *Advances in Cryptology – CRYPTO 2023*. Ed. by Helena Handschuh and Anna Lysyanskaya. Cham: Springer Nature Switzerland, 2023, pp. 358–389. ISBN: 978-3-031-38554-4.

[DFHS23] Jelle Don, Serge Fehr, Yu-Hsuan Huang, and Patrick Struck. *On the (In)Security of the BUFF Transform*. Cryptology ePrint Archive, Paper 2023/1634. 2023. URL: `https://eprint.iacr.org/2023/1634`.

Preliminaries Chapter 2 introduces some concepts that will return throughout the thesis. We outline our notation and definitions from the literature that will play a role in our security definitions. The chapter ends with an introduction to an important tool for QROM analysis known as the compressed oracle technique [Zha19a].

Security of the Fiat-Shamir Transformation in the QROM Chapter 3 is based on [DFMS19] and [DFM20]. In this chapter we establish the first post-quantum security proof of the Fiat-Shamir transformation – both the standard and multi-round versions – thereby enabling a QROM security reduction for a large class of post-quantum signature schemes [ABCP22; Beu20; BGKM23; BKV19; BMPS20; CDG+17; CHH+21; CHR+20; BSK+21; DKR+21; GPS22; TDJ+22].

The Fiat-Shamir transformation [FS87] is a generic method to turn any public-coin honest-verifier-zero-knowledge interactive proof system $\mathsf{\Pi}$ for an NP-relation R into a *non-interactive* proof system $\mathsf{FS}[\mathsf{\Pi}]$, or into a signature scheme $\mathsf{FSSIG}[\mathsf{\Pi}]$ (depending on the variant of the transformation). For three-round protocols, well-known as Σ-protocols (see Figure 1.2), $\mathsf{FS}[\mathsf{\Pi}]$ is obtained by specifying the challenge c as $c := H(x, a)$ instead of having it chosen by the verifier. Here H is a cryptographic hash function, which we will eventually model as a random oracle. The signature scheme $\mathsf{FSSIG}[\mathsf{\Pi}]$ is obtained by additionally including the to-be-signed message m in the hash, and letting $pk := x$ and $sk := w$ for some randomly chosen $(x, w) \in R$.

The goal of a security proof for the Fiat-Shamir transformation is to show that it preserves security against malicious provers. The two relevant security notions are *soundness* (no prover can convince the verifier on an instance x

that is not true, i.e. for which no witness w s.t. $(x, w) \in R$ exists) and *proof of knowledge* (no prover can convince the verifier on an instance x without knowing a witness w s.t. $(x, w) \in R$). It suffices to show that any prover $\mathcal{A}_{\mathsf{FS}}$ in the non-interactive scheme that convinces the verifier $\mathcal{V}_{\mathsf{FS}}$ with probability ϵ can be turned into a prover $\mathcal{A}_{\mathsf{\Pi}}$ that convinces the interactive verifier $\mathcal{V}_{\mathsf{\Pi}}$ with probability polynomially related to ϵ; if no such provers exist against the interactive scheme, the reduction ensures they cannot exist against the non-interactive scheme either.

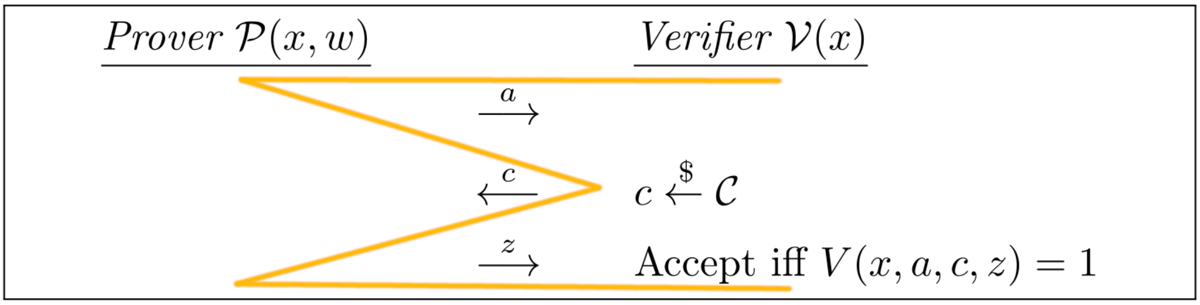
Prover $\mathcal{P}(x, w)$
Verifier $\mathcal{V}(x)$
$\xrightarrow{a}$
$\xleftarrow{c}$
$c \overset{\$}{\leftarrow} \mathcal{C}$
$\xrightarrow{z}$
Accept iff $V(x, a, c, z) = 1$

Fig. 1.3. Schematic representation of a Σ-protocol. The name derives from the shape of the interaction (we can either think of a Σ or of a 'zig-zag' movement combined with a Merlin-Arthur protocol).

Classically, different security proofs have been given in the random oracle-model [FS87; PS96; FKMV12]. The simplest reduction (for the three-round version) works by picking one of the adversary's queries $x_i||a_i$ to the random oracle at random, forwarding a_i to $\mathcal{V}_{\Pi}(x_i)$ and programming the returned challenge c into the oracle reply. Since the adversary will have to use one of its q queries for the forgery, with probabilty ϵ/q the adversary's final output z will be valid with respect to (x_i, a_i, c), and will thus be accepted by $\mathcal{V}_{\Pi}(x_i)$. The same argument works for multi-round protocols, except that now multiple queries have to be reprogrammed.

The main obstacle to a similar reduction in the QROM is that observing one of the adversary's queries potentially disturbs its state, making it in general hard to predict the adversary's behaviour during the rest of its execution. The post-measurement state may have only negligible overlap with the original, if it consisted of a superposition over exponentially many query inputs. Due to the importance of the FS transform in cryptographic applications, prior works had already studied it in the QROM, but they either claimed an impossibility result in a very constrained setting (e.g. where the reduction is not allowed to measure a query) [DFG13] or posited a crucial step in the analysis as an open question [Unr17].[7] Dominique Unruh [Unr15b] introduced an alternative transform that manages to side-step the difficulties in the QROM analysis of the FS transform, while achieving the same goal of removing interaction. This 'Unruh transform' is unfortunately much less efficient in terms of proof/signature size, although we somewhat improve this situation with a new analysis in Chapter 5.

In Chapter 3 we solve the problem by introducing the 'measure-and-reprogram' technique, which allows for an almost one-to-one emulation of the classical reduction strategy. The two differences are the following: First, once the reduction has picked a random query, it measures that query in the computational basis. The outcome of the measurement is forwarded to the interactive verifier, as in the classical case. Secondly, when the reduction received a challenge from the verifier and is ready to program it into the oracle, it flips a coin to decide on reprogramming now or at the next query. In the analysis we are then able to bound the disturbance caused by the measurement and this – slightly odd – way of reprogramming, achieving an $\epsilon/(2q+1)^2$ success probability for the reduction. The concurrent and independent work [LZ19a] presented a QROM reduction for the FS transform as well (based on the compressed-oracle technique), but their reduction incurs an $O(q^9)$ loss.

We extend our technique to multi-round Fiat-Shamir (for $2n+1$-round interactive protocols where the verifier sends n uniformly random challenges). Here we achieve a $O(q^{2n})$ loss, and show that this is tight in the general case.

Finally, we introduce the notion of *quantum computational unique responses* and show that it is sufficient for (Unruh-)rewinding, allowing special-sound Σ-protocols to be proven secure as a proof of knowledge against quantum adversaries in the plain model. Prior to our work, Unruh rewinding required the stricter notion of 'perfect unique responses', which is not satisfied by typical protocols in the post-quantum setting. We use the new notion to include a complete security reduction for signature schemes Picnic [CDG+17] and MQDSS [CHR+20], as well as Dilithium [BDK+21] under a plausible assumption.

Online Extractability Where the previous chapter enabled extraction of a query input at the expense of some (polynomial) disturbance, and extraction of the witness in the interactive protocol Π depended on rewinding, in Chapter 4 (based on [DFMS22a]) we show that such drawbacks are not always necessary. In certain contexts – for example when the first message of Π consists of a

[7] Or restricted to the special case of statistically sound Σ-protocols and preservation of only soundness, not proof-of-knowledge [Unr17].

hash based commitment that we model as a random oracle – at least in the classical case, *online extraction* [Fis05] is possible. 'Online' in this case means straight-line (no rewinding) and on-the-fly (during protocol execution and without disturbing it). Rewinding often causes a reduction loss (because we need the adversary to succeed twice) and evidently a disturbance in the adversary's state causes a loss as well. If possible, online extraction is thus the preferred option.

Building upon the compressed oracle framework [Zha19a], in this chapter we introduce a statistically indistinguishable simulator for a quantum random oracle, with both a query and an extraction interface. We show the following generic result: Consider an arbitrary quantum query algorithm $\mathcal{A}$ in the QROM, which announces during its execution some classical value t that is supposed to be equal to $f(x, H(x))$ for some x. Here, f is an arbitrary fixed function, subject to that it must tie t sufficiently to x and $H(x)$, e.g., there must not be too many y's with $f(x, y) = t$; a canonical example is the function $f(x, y) = y$ so that t is supposed to be $t = H(x)$. In general, it is helpful to think of $t = f(x, H(x))$ as a commitment to x. We then show that x can be efficiently online-extracted with almost certainty, by querying t to the extraction interface of our simulator, obtaining a 'guess' $\hat{x}$. Whenever $\mathcal{A}$ outputs x with $f(x, H(x)) = t$ at some later point, $\hat{x} = x$ holds except with negligible probability, while $\hat{x} = \emptyset$ (some special symbol) indicates that $\mathcal{A}$ will not be able to output such an x.

At the core of our result is a new commutator bound, that quantifies the potential disturbance caused by swapping an extraction measurement of the compressed oracle with a random-oracle query from the adversary. If the above relation is tight enough, the disturbance is negligible, and we can thus freely add extraction queries by inserting them at the end of the adverary's run and then swapping them up to any point after t was put on the table. Thus, under the right circumstances, online extraction is possible in the QROM as well.

As a not unimportant side-effect, the abstraction of our simulator, with its extraction interface and properties formulated in classical terms, cryptographers with no background in quantum information theory can argue about such examples as the above (in the QROM!) using only classical reasoning.

Our first main application is to so-called 'commit-and-open protocols'. C&O protocols form a subclass of Σ-protocols, were the first message a is a set of commitments, and the challenge determines what subset of these the prover has to open in the third message. We first introduce a generalized notion of special soundness that captures the intuition that a witness can be computed from valid responses to 'sufficiently many' challenges, and then use our technique to inconspiciously open *all* the adversary's commitments, which must contain sufficient valid responses if the adversary has a good chance of convincing the verifier. We thus obtain a tight bound for the proof of knowledge property of C&O schemes, i.e. our knowledge extractor succeeds with probability proportional to the advantage of a malicious prover over the trivial cheating probability (up to a negligible additive error that we show to be tight with an attack).

The second main application is to the Fujisaki-Okamoto transform [FO99], which underlies many KEM's in the NIST post-quantum competition. We give the first complete post-quantum security proof of the *textbook* (i.e. without any adjustments that facilitate the proof) FO tranform. Most of the prior post-quantum security proofs had to adjust the transformation to facilitate the proof (like [HHK17]); those security proofs either consider a FO variant that employs an *implicit-rejection* routine, i.e., where the decapsulation algorithm outputs a pseudo-random key upon an invalid ciphertext rather than a rejection message, or have to resort to an additional "key confirmation" hash [TU16] that is appended to the ciphertex, thus increasing the ciphertext size. The *unmodified* FO transformation was analyzed in [Zha19a] and [KKPP20]; however, as we explain in detail in Section 4.6.3, the given post-quantum security proofs are incomplete, both having the same gap.

Beyond its theoretical relevance of showing that no adjustment is necessary to admit a post-quantum security proof, the security of the original unmodified FO transformation with explicit rejection in particular ensures that the conservative variant with implicit rejection remains secure even when the decapsulation algorithm is not implemented carefully enough and admits a side-channel attack that reveals information on whether the submitted ciphertext is valid or not.

The core idea of our proof for the textbook FO transformation is to use the extractability of the RO-simulator to handle the decryption queries. Indeed, letting $f(x, y)$ be the encryption $Enc_{pk}(x; y)$ of the message x under the randomness y, a "commitment" $t = f(x, H(x))$ is then the encryption of x under the derandomized scheme, and so the extraction interface recovers x.

Efficient NIZK's and Signatures from Commit-and-Open Protocols Finally, in Chapter 5 we again show online extractability for commit-and-open protocols, but now of the Fiat-Shamir transformed non-interactive version of them. While the techniques from Chapter 3 and 4 could be combined to obtain a security reduction for such protocols, this strategy would not result in *online* extraction due to the disturbance caused by the measurement in the measure-

and-reprogram technique, inflicting a $(2q+1)^2$ multiplicative loss for the success probability of the reduction.

André Chailloux was the first to aim for online extractability of the Fiat-Shamir transformation in the QROM for this class of protocols. Indeed, the Fiat-Shamir transformation of C&O Σ-protocols are known to be online extractable in the classical ROM (see e.g. discussion in [Fis05]). In a first attempt [Cha19], Chailloux tried to lift the argument to the quantum setting by means of Zhandry's compressed-oracle technique [Zha19a], which offers a powerful approach for re-establishing ROM results in the QROM, that has been successful in many instances. Unfortunately, this first attempt contained a subtle flaw, which turned out to be unfixable, and despite changing the technical approach, the latest version [Cha21] of this work still contains a gap in the proof, which is put as an assumption.

The situation is complicated because the adversary queries the random oracle to determine both its first message $\mathbf{y} = H(\mathbf{m})$ (consisting of a set of hash-based commitments) and the corresponding challenge (computed as $H(x||\mathbf{y})$), and may use these queries to search for a suitable commitment-challenge pair that allows it to pass verification without actually knowing the witness.

To tackle the problem, we build upon and slightly extend the [CFHL21] framework for the compressed oracle. [CFHL21] introduced the notion of 'quantum transition capacity' of two 'database properties' P, P', a measure of how much more likely we are to find the recorded queries of the compressed oracle to satisfy a property P' after each query by the adversary, if the database started in P. We first extend the framework by proving a revised version of the main theorem that bounds a quantum transition capacity in terms of the considered properties. In our security reduction we then define a database property of exactly the case described above, where the queries to the oracle allow the adversary to find a commitment-challenge pair that help him forge a proof without knowing a witness. We are then able to show that quantum transition capacity from the empty to this special database is small, i.e. for a bounded query algorithm the described situation can be achieved only with negligible probability.

Our security reduction is tight: Whenever a prover outputs a valid proof, the online-extractor succeeds, except with a small probability accounting for collision and preimage attacks on the involved hash functions. Our result also applies to a variant of the Fiat-Shamir transformation where a digital signature scheme (DSS) is constructed. It thereby, for the first time, enables a multiplicatively tight security reduction for, e.g., DSS based on the MPC-in-the-head paradigm [IKOS07a], like Picnic [CDG+17], Banquet [BSK+21] and Rainier [DKR+21], in the QROM.

When a Σ-protocol does not have the mentioned C&O structure, a non-interactive proof of knowledge with online extractability in the QROM can be obtained using the Unruh transformation [Unr15b]. For technical reasons, the Unruh transformation requires the hash function to be *length preserving*, which may result in large commitments, and thus large NIZKs and digital signature schemes. We revisit this transformation and show, by a rather direct application of our main result above, that the online extractability of the Unruh transform still holds when using a *compressing* hash function. The crucial observation is that the Unruh transformation can be viewed as the composition of a pre-Unruh transformation, which makes use of hash-based commitments and results in a C&O protocol, and the Fiat-Shamir transformation. By applying our security reduction, we obtain the tight online extractability without requiring the hash function to be length preserving.

In real-world constructions based on C&O protocols, like e.g., the Picnic digital signature scheme, commitments and their openings are responsible for a significant fraction of the signature/proof size. For certain parameters, this cost can be reduced by using a collective commitment mechanism based on Merkle trees. This was observed in passing, e.g. in [Fis05], and is exploited in the most recent versions of Picnic. We formalize Merkle-tree-based C&O protocols and extend our main result to NIZKs constructed from them (see Theorem 5.23). Applications of this result include a security reduction of Picnic 3 [KZ20], the newest version of the Picnic digital signature scheme, that is significantly tighter than existing ones: An adversary against the Picnic 3 signature scheme in the QROM with success probability ε can now be used to break the underlying hard problem with probability ε, up to some additive error terms, while previous reductions yielded at most ε^5/q^{10}, where q is the number of random oracle queries.

Chapter 2

Preliminaries

Chapter contents

Section 2.1

Mathematical preliminaries

2.1.1 Basic notation and probabilities

Let $\mathbb{Z}, \mathbb{R}, \mathbb{R}^+, \mathbb{C}$ denote the set of integers, real numbers, non-negative real numbers and complex numbers respectively. For a set S, $|S|$ is the cardinality of that set. $[1, n]$ is a shorthand for $\{i \in \mathbb{Z} : 1 \leq i \leq n\}$. We write $:=$ to assign the right-hand side to the left-hand side. $a \overset{\$}{\leftarrow} A$ indicates that an element a is selected uniformly at random from a set A, whereas $a \leftarrow \mathcal{A}$ means that it is produced by the algorithm $\mathcal{A}$ (probabilistically or not). In the case of interactive algorithms, we let $\langle \mathcal{A}, \mathcal{V}(x) \rangle$ denote the outcome of an interaction between $\mathcal{A}$ and $\mathcal{V}$ when the latter is run on input x.

When considering the probability $\Pr[E]$ of some event E or the expectation $\mathbb{E}[X]$ of a random variable X, we often leave the underlying probability space implicit. Usually, it is well defined by means of a probabilistic algorithm or experiment, which is often clear from the context. Sometimes we make the probability space a bit more explicit and e.g. write $\Pr_H[E]$ (respectively $\mathbb{E}_H[X]$) to emphasize that the probability space includes the random choice of the function H, as opposed to contexts where H is fixed, or $\Pr_{a \leftarrow \mathcal{A}}$ if the probability is (also) over a produced by $\mathcal{A}$.

The *Chernoff bound* says that for any random variable X and any $\delta > 0$ we have

$$\Pr\left[X > (1+\delta) \cdot \mathbb{E}\left[X\right]\right] < \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mathbb{E}[X]},$$

allowing us to bound the probability that X deviates too much from its expected value. For convex functions $f : C \to \mathbb{R}$ with $C \subset \mathbb{R}$ a convex set, such as $f : C \to \mathbb{R};\ x \mapsto x^2$, we may invoke *Jensen's inequality*:

$$f\left(\frac{\sum a_i x_i}{\sum a_i}\right) \leq \frac{\sum a_i f(x_i)}{\sum a_i} \qquad \forall n \in \mathbb{N}, x_1, \ldots, x_n \in C, a_1, \ldots, a_n \geq 0.$$

2.1.2 Quantum computing

The state of a quantum system is described by a unit vector $|\psi\rangle$ (called a 'ket' vector) in a Hilbert space $\mathcal{H}$ over $\mathbb{C}$, for quantum computing purposes of finite dimension d. Elements from the dual space $\mathcal{H}^*$ of linear functionals $\mathcal{H} \to \mathbb{C}$ are denoted by 'bra' vectors $\langle\phi|$.[8] For every vector $|\chi\rangle \in \mathcal{H}$ there is a unique vector $\langle\chi| \in \mathcal{H}^*$ such that $\langle\chi|\psi\rangle := \langle\chi||\psi\rangle = (|\chi\rangle, |\psi\rangle)$, where $(\cdot,\cdot)$ is the inner product on the Hilbert space. If we fix an orthonormal basis $\{|e_i\rangle\}_{i\in I}$ of $\mathcal{H}$, we can express a ket vector $|\psi\rangle$ and the corresponding bra vector $\langle\psi|$ as

$$|\psi\rangle = \begin{bmatrix}\alpha_1\\ \alpha_2\\ \vdots\\ \alpha_d\end{bmatrix} \in \mathbb{C}^d \qquad \text{and } \langle\psi| = \begin{bmatrix}\bar{\alpha}_1\ \bar{\alpha}_2\ \ldots\ \bar{\alpha}_d\end{bmatrix} \in \left(\mathbb{C}^d\right)^*$$

with the α_i being the vector coefficients of $|\psi\rangle$ with respect to this basis, also known as its amplitudes. Given the properties above, the inner product is then pinned down to

$$(|\psi\rangle, |\phi\rangle) = \sum_i \bar{\alpha}_i\beta_i$$

(where $|\psi\rangle$ is as above and the β_i are the coefficients of $|\phi\rangle$). The inner product induces a norm:

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

The 2-norm is also called the *Euclidean* norm, and we sometimes drop the subscript 2 if it is clear from the context which norm we use. In the case of unit vectors, we have by definition $1 = \||\psi\rangle\|_2^2 = \left(\sqrt{\langle\psi|\psi\rangle}\right)^2 = \sum_i |\alpha_i|^2$, hence the squared moduli of a state's amplitudes must sum to 1.

If $d = 2$, the quantum system is called a 'qubit' (quantum bit), and the canonical basis is often written as $\{|0\rangle, |1\rangle\}$ and referred to as the *computational basis*. Each of the two basis states then represents the bit taking the classical value 0 or 1. An arbitrary qubit state $|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ however can be any linear combination of the basis states, and we say that the qubit is in a superposition of 0 and 1 if both coefficients α_0 and α_1 are non-zero.

Given two systems (e.g. two electrons) with respective state spaces $\mathcal{H}_1$ and $\mathcal{H}_2$, one can consider the joint system (the electrons considered as a pair) which then has state space $\mathcal{H}_{12} := \mathcal{H}_1 \otimes \mathcal{H}_2$. In quantum computing, we distinguish between different quantum (sub-)systems by labelling them as *registers*. For a register A consisting of n qubits, the dimension of the system is $d = 2^n$. The state $|\phi\rangle_A$ of this register is an element of $\bigotimes_i \mathcal{H}_i$ (with $i \in [1, n]$ and $\mathcal{H}_i$ the

[8] Bra's and ket's where introduced by Paul Dirac in 1939, and are therefore known as 'Dirac notation'.

respective state spaces of the individual qubits) and is (in general) a sum of product states $\sum_j |\phi_{j_1}\rangle \otimes |\phi_{j_2}\rangle \otimes \ldots \otimes |\phi_{j_n}\rangle$. We call it 'entangled' if it cannot be written as a *single* product state. In an n qubit system, the computational basis generalizes to $\{\bigotimes_{i=1}^{n} |0\rangle, \bigotimes_{i=1}^{n-1} |0\rangle \otimes |1\rangle, \ldots, \bigotimes_{i=0}^{n} |1\rangle\}$, more conveniently written as

$$\{|0\rangle, |1\rangle, \ldots, |N-1\rangle\} \qquad \text{(computational basis for system of } n \text{ qubits)}$$

where $N := 2^n$. The state $|\phi\rangle_A$ of the above register A can thus be written as

$$|\phi\rangle_A = \alpha_0|0\rangle + \alpha_1|1\rangle + \ldots + \alpha_{N-1}|N-1\rangle$$

and hence be in a superposition over 2^n distinct classical values, e.g. strings of classical bits or integers.

Next we consider the space $\mathcal{L}(\mathcal{H})$ of linear maps $\mathcal{H} \to \mathcal{H}$, also called operators on $\mathcal{H}$. Three types of operators are of particular interest for quantum computing: Unitaries, measurement operators and density operators.

A unitary is an operator $U \in \mathcal{U}(\mathcal{H}) \subseteq \mathcal{L}(\mathcal{H})$ that satisfies $U^\dagger U = UU^\dagger = \mathbb{1}$, where $U^\dagger$ is the *adjoint* of U, the unique operator that satisfies $(U|\psi\rangle, |\phi\rangle) = (|\psi\rangle, U^\dagger|\phi\rangle)$, and $\mathbb{1}$is the identity map. Quantum algorithms act on a quantum system by doing physical operations (usually one out of a fixed set of *gates*) whose effect is described by applying a unitary U to the state of the system. As can be seen from the equation above, $U^\dagger$ reverses that operation (gate).

As the system evolves through these deterministic and reversible operations in the form of gates described by unitaries, to produce a classical outcome the quantum algorithm will eventually have to perform a *measurement*, which has a probabilistic outcome and is non-reversible. A measurement on a state space $\mathcal{H}_\mathsf{A}$ is described by a finite set of measurement operators $\mathbf{M} := \{M_i\}_{i \in I}$, where each $i \in I$ is a possible measurement outcome, with the constraint that $\sum_i M^\dagger M = \mathbb{1}$ The set of all such measurements on the state space $\mathcal{H}_A$ for a given index set I is denoted by $\mathcal{M}eas_I(\mathcal{H}_\mathsf{A})$. If the state prior to the measurement is given by $|\psi\rangle$, the probability of finding outcome $i \in I$ is

$$\langle\psi|M_i^\dagger M_i|\psi\rangle = \||M_i|\psi\rangle\|_2^2 \qquad \text{(Born's rule).}$$

Importantly, the measurement alters the state, so that if i was indeed the outcome then the post-measurement state will be equal to

$$|\psi'\rangle = \frac{M_i|\psi\rangle}{\sqrt{\langle\psi|M_i^\dagger M_i|\psi\rangle}} \qquad \text{(post-measurement state).}$$

We say that the state $|\psi\rangle$ has *collapsed* to $|\psi'\rangle$.

If the measurement operators consist of orthogonal projectors[9] (i.e $M_i = M_i^\dagger = M_i^2$) then we say that the measurement is projective. Examples of projective measurements that we often use in this thesis are measurements in a certain basis. For example, the measurement of a single qubit in the computational basis consists of the projectors $\{|0\rangle\langle 0|, |1\rangle\langle 1|\}$.[10] Another important example is the *Hadamard* basis:

$$\{|+\rangle := \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle),\ \ |-\rangle := \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\} \qquad \text{(Hadamard basis)},$$

where the name derives from the Hadamard operator, a unitary operator that relates the Hadamard and the computational bases as follows:

$$H|0\rangle = |+\rangle \qquad H|1\rangle = |-\rangle \qquad H|+\rangle = |0\rangle \qquad H|-\rangle = |1\rangle.$$

For n qubit systems, the Hadamard basis can be generalized in two ways. One is via Walsh-Hadamard transformation $H^{\otimes n}$, which simply applies the Hadamard operator H qubit-wise. Applying $H^{\otimes n}$ to each of the (generalized) computational basis states gives us the Walsh-Hadamard basis. The other way is via the Quantum Fourier Transformation, leading to

$$\{|\hat{k}\rangle\}_{k\in\{0,\ldots,N-1\}} \qquad |\hat{k}\rangle := U_{QFT}|k\rangle = \frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}\omega_N^{kj}|j\rangle \qquad \text{(Fourier basis)}$$

where again $N := 2^n$ and $\omega_N \in C$ s.t. $\omega_N^N = 1$ is an N-th root of unity.

While a general quantum algorithm can perform a measurement at any time during its execution, with the help of some additional (ancilla) qubits it can always (without loss of generality) replace such measurements with unitaries and only do the actual measurement at the end of its execution, or as soon as it has to produce some classical output. This phenomenon is known as the Deferred Measurement Principle (and also as 'purifying' the algorithm), and is essentially a consequence of Naimark's Dilation Theorem. We state it here as formulated in [Feh22]:

Theorem 2.1 (Naimark's Dilation Theorem).
Let $\mathbf{M} = \{M_i\}_{i\in I} \in \mathcal{M}eas_I(\mathcal{H}_{\mathsf{A}})$*, and let* $\{|i\rangle\}_{i\in I}$ *be an orthonormal basis of*

[9] Since we only use orthogonal projectors in this thesis, we will often refer to them simply as 'projectors'.

[10] In Dirac notation, $|\psi\rangle\langle\phi|$ denotes the outer product between $|\psi\rangle$ and $|\phi\rangle$.

$\mathcal{H}_B = \mathbb{C}^{|I|}$. *Then there exists an isometry* $V \in \mathcal{L}(\mathcal{H}_A, \mathcal{H}_{AB})$ *such that for every* $|\phi\rangle \in \mathcal{S}(\mathcal{H}_A)$ *and* $i \in I$ *we have*

$$M_i|\phi\rangle \otimes |i\rangle = (\mathbb{1}_A \otimes |i\rangle\langle i|)V|\phi\rangle.$$

By basic properties of isometries, V can be chosen as $V = UV_\circ$ for $V_\circ = \mathcal{H}_A \to \mathcal{H}_{AB}, |\phi\rangle \mapsto |\phi\rangle|0\rangle$ and $U \in \mathcal{U}(\mathcal{H}_{AB})$. Thus, instead of performing the measurement $\mathbf{M}$, we may first append a number of ancilla qubits equal to $\lceil \log |\mathbf{M}| \rceil$, each of them initialized at $|0\rangle$, apply U to the joint system and then measure B. But now any action by the algorithm on the A register commutes with the measurement of B, because they act on separate systems. Hence we may perform the measurement at the very end of the algorithm's execution, and assume any quantum algorithm to be unitary up until its final measurement, without loss of generality.

So far we have been writing all quantum states as *pure* states; unit vectors in a finite dimensional Hilbert space. This formalism suffices as long as the states we consider are fully determined. In some cases however, we may want to consider a probabalistic mixture of states, for example when we consider a state that has been measured but the measurement outcome was not recorded. We call such states *mixed* states, and they are described by the density operator formalism. A density operator $\rho \in \mathcal{L}(\mathcal{H})$ satisfies

- $\mathrm{tr}(\rho) = 1$ (Trace one)
- $\rho \geq 0$ (Positive semi-definite).

The density operator corresponding to the pure state $|\psi\rangle \in \mathcal{H}$ is given by $|\psi\rangle\langle\psi| \in \mathcal{L}(H)$. In general, a mixed state (density operator) ρ can be any convex combination of pure states.

In cryptography we often consider how well an adversary is able to distinguish between two games, or how well he is able to detect a measurement on its state. The *Schatten-1* or *trace norm*, $\|A\|_1 = \mathrm{tr}\left[\sqrt{A^\dagger A}\right]$ is important in this context, because the related *trace distance* equals the probability that anyone using the optimal strategy (i.e. the optimal measurement) is able to distinguish two considered states. For density matrices ρ and σ, it is defined as

$$\delta(\rho, \sigma) := \frac{1}{2}\|\rho - \sigma\|_1 = \frac{1}{2}\,\mathrm{Tr}\left[\sqrt{(\rho - \sigma)^\dagger(\rho - \sigma)}\right] \qquad \text{(Trace distance).}$$

If both ρ and σ are actually pure states, the definition simplifies to

$$\delta(|\psi\rangle\langle\psi|, |\phi\rangle\langle\phi|) = \sqrt{1 - |\langle\psi|\phi\rangle|^2} \qquad \text{(Trace distance for pure states).}$$

where $|\langle\psi|\phi\rangle|^2$ is called the 'overlap' between $|\psi\rangle$ and $|\phi\rangle$.

By equation (9.110) in [NC11] and a short calculation, any norm-1 vectors $|\varphi\rangle$ and $|\psi\rangle$ satsify

$$\delta(|\varphi\rangle\langle\varphi|, |\psi\rangle\langle\psi|) \leq \| |\varphi\rangle - |\psi\rangle \|_2 \,. \tag{1}$$

For probability distributions p and q, we write $\delta(p, q)$ for the *total variational distance*; this is justified as $\|\rho_0 - \rho_1\|_1 = \delta(p_0, q_1)$ for $\rho_i = \sum_x p_i(x)|x\rangle\langle x|$, $i = 0, 1$.

For further background on quantum computing and its formalism, we refer the reader to [NC11].

2.1.3 Oracle algorithms and the QROM

A quantum algorithm is typically formalized by means of a quantum circuit, where the computational complexity is then given by the number of gates in the circuit. In this thesis, we mainly consider the *query* complexity of (quantum) *oracle* algorithms, which make queries to an external resource, referred to as an oracle. A quantum oracle algorithm $\mathcal{A}$ is formally specified by an initial state $|\phi_0\rangle$ and a sequence of unitaries $A_1, \ldots, A_q$ of unitaries, with the understanding that executing the algorithm means that $A_q\mathcal{O} \ldots \mathcal{O}A_1$ is applied to $|\phi_0\rangle$, possibly followed by a measurement if the output is classical. Here the operation $\mathcal{O}$ represents the oracle calls. In this thesis, we exclusively consider oracles that implement a classical (possibly randomized or stateful) function f. We then distinguish between classical and quantum queries to $\mathcal{O}$. In the latter, $\mathcal{O}$ is formally defined as the unitary $\mathcal{O}|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle$. In the former, the registers to which $\mathcal{O}$ is applied are first measured in the computational basis, before $\mathcal{O}$ is applied as above. We consider such an oracle algorithm *efficient* (also called 'bounded') if $|0\rangle := |00\ldots0\rangle$, and the A_i's are specified by quantum circuits with a polynomial number of gates.

The above naturally extends to oracle algorithms that make oracle calls to several oracles, where it is possibile to dinstinguish fixed and adaptively chosen query order. Our recent work [DFH22] however shows that this distinction is not very relevant, as the adaptive case can be simulated by a fixed-order algorithm with only a multiplicative blow-up of n to each of the individual number of queries per oracle, if there are n distinct oracles.

The (quantum) random-oracle model. In the random-oracle model, a cryptographic hash function, which is used as a building block in the construction of a cryptographic scheme, is abstracted away by a *random-oracle*, which implements a uniformly random function to which parties have no other access than via querying the random-oracle. Formally, this means that algorithms in the random-oracle model, and in particular the adversary, will be modelled as oracle algorithms with oracle access to a uniformly random function $H : \mathcal{X} \to \mathcal{Y}$. For concreteness, we restrict here to $\mathcal{Y} = \{0,1\}^n$; on the other hand, we do not further specify the domain $\mathcal{X}$ except that we assume it to have an efficiently computable order, so one may well think of $\mathcal{X}$ as $\mathcal{X} = \{1, \ldots, M\}$ for some positive $M \in \mathbb{Z}$ or as bit strings of bounded size. In terms of notation, we will write either H or $\mathcal{O}^H$ to denote the oracle.

In the classical random-oracle model (ROM), the attacker makes classical queries to $\mathcal{O}^H$. In a quantum setting it is natural to give the attacker quantum query access to $\mathcal{O}^H$, since the original hash function is an offline primitive that the attacker could easily evaluate on a superposition of inputs, as was argued in [BDF+11].

While there are artificial examples that show the existence of cryptographic schemes that are proven secure in the ROM but insecure in practice, the common belief is that such a failure of the random-oracle methodology does not occur in natural schemes. See Section 1.1.1 for a more in-depth discussion of this point.

Section 2.2

Zero-knowledge proof systems

A zero-knowledge proof system is a protocol between a prover and a verifier, where the prover may prove membership of statements x with respect to a certain language $\mathcal{L}$, in such a way that the verifier learns exactly $x \in \mathcal{L}$ and nothing more. For an NP-relation[11] we may consider zero-knowledge *proofs of knowledge*, where the verifier beyond learning $x \in \mathcal{L}$ is also convinced of the fact that the prover knows a witness w. Many results in this thesis consider a certain class of such protocols, called Σ-protocols.

2.2.1 Σ-protocols

Definition 2.2 (Σ-protocol). *A Σ-protocol $\Sigma = (\mathcal{P}, \mathcal{V})$ for a relation $R \subseteq \mathcal{X} \times \mathcal{W}$ is a three-round two-party interactive protocol of the form:*

$$\boxed{\begin{array}{lcl} \underline{\textit{Prover } \mathcal{P}(x, w)} & & \underline{\textit{Verifier } \mathcal{V}(x)} \\ & \xrightarrow{a} & \\ & \xleftarrow{c} & c \overset{\$}{\leftarrow} \mathcal{C} \\ & \xrightarrow{z} & \text{Accept iff } V(x, a, c, z) = 1 \end{array}}$$

Here $\mathcal{P}$ is an efficient two-stage algorithm, and we write

$$(a, z) \leftarrow \langle \mathcal{P}(x, w), c \rangle$$

for the generation of the *first message* a in the first stage and the *response* z in the second stage once given the *challenge* c, which the verifier draws uniformly at random from the *challenge set* $\mathcal{C}$. The *verification predicate* V is efficiently computable and determines whether the verifier accepts or not.

We allow the set of *instances* $\mathcal{X}$[12], the set of *witnesses* $\mathcal{W}$ and the relation R to depend on a security parameter η. Similarly, the interactive algorithms $\mathcal{P}$ and $\mathcal{V}$ may depend on η (or have η as part of their input). However in this thesis, for ease of notation, we suppress these dependencies on η unless they are crucial.

[11] An NP-relation is an efficiently computable relation $R \subseteq \mathcal{X} \times \mathcal{W}$ that defines the language $\mathcal{L} := \{x \in \mathcal{X} | \exists w \in \mathcal{W}. R(x, w)\}$.

[12] In chapters 4 and 5 and the definition of C&O protocols below we switch notation from $x \in \mathcal{X}$ to $\textsf{\textit{inst}} \in \mathcal{I}$.

Commit-and-Open Σ-Protocols. For the purpose of this thesis, a *commit-and-open Σ-protocol*, or *C&O Σ-protocol* or *C&O protocol* for short, is a Σ-protocol $\Pi = (\mathcal{P}, \mathcal{V})$ of a special form, involving a hash function $H : \mathcal{X} \to \mathcal{Y}$ that is modeled as a RO[13], where we fix $\mathcal{X} := \{0,1\}^{\leq B}$ and $\mathcal{Y} := \{0,1\}^n$. Concretely, in a C&O protocol, the transcript (a, c, z) is of the following form (see Figure 2.1). The first message a consists of *commitments* $y_1, \ldots, y_\ell$, computed as $y_i = H(m_i)$ for *messages* $m_1, \ldots, m_\ell \in \mathcal{M}$, and possibly an additional string $a_\circ$[14]. The challenge c is picked uniformly at random from the challenge space $\mathcal{C} \subseteq 2^{[\ell]}$, which is set to be a subset of $2^{[\ell]}$. Finally, the response z is given by $\mathbf{m}_c = (m_i)_{i \in c}$. Eventually, $\mathcal{V}$ accepts if and only if $H(m_i) = y_i$ for all $i \in c$ and some given predicate $V(\mathit{inst}, c, \mathbf{m}_c, a_\circ)$ is satisfied.

For the above to be meaningful, we obviously need that $\mathcal{M} \subseteq \mathcal{X}$, i.e., the bit size of the possible m_i's are upper bounded by B. Furthermore, the parameter n determines the hardness of finding a collision in H (in the random oracle model), and thus the level of binding the commitments provide.

Additionally, we may consider a generalization of C&O protocols, where the first message is parsed as a *single* commitment y of the ℓ messages $m_1, \ldots, m_\ell$ and where this commitment is computed by means of an arbitrary "multi-message" commitment scheme involving H, which has the property that any subset of $m_1, \ldots, m_\ell$ can be opened without revealing the remaining m_i's. The above component-wise hashing is then one particular instantiation, but alternatively one can for instance also compute y by means of a Merkle tree (see Section 5.5.1), and then open individual m_i's by revealing the corresponding authentication paths. Looking ahead, we stress that the concepts developed in Section 4.5.2: the notions of $\mathfrak{S}$-soundness and $\mathfrak{S}$-soundness* and the probability $p^{\mathfrak{S}}_{triv}$, do not depend on the choice of commitment scheme, and thus remain unaffected when considering such a *Merkle-tree-based C&O* protocol. To emphasize the default choice of the commitment scheme, which is element-wise hashing, we sometimes also speak of an *ordinary* C&O protocol.

Zero-knowledge. An important aspect of Σ-protocols is their zero-knowledge property. In this thesis we only consider *perfect* honest verifier zero-knowledge (HVZK):

[13] One could also refer to Σ-protocols that use non-hash-based commitments, and/or are analyzed in the standard model, as *C&O protocols*, but this is not the scope here.

[14] Note that $m_i \in \mathcal{M}$ may consist of the actual "message" (computed by the prover using the witness w), possibly concatenated with randomness.

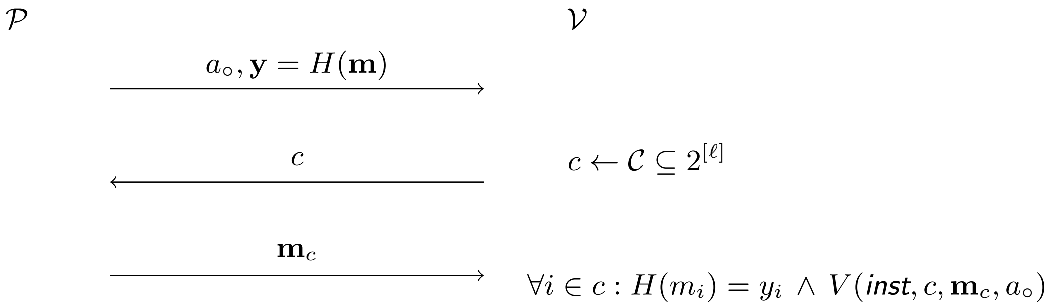
$\mathcal{P}$
$\mathcal{V}$
$a_\circ, \mathbf{y} = H(\mathbf{m})$
c
$c \leftarrow \mathcal{C} \subseteq 2^{[\ell]}$
$\mathbf{m}_c$
$\forall i \in c : H(m_i) = y_i \wedge V(\mathit{inst}, c, \mathbf{m}_c, a_\circ)$

Fig. 2.1. An (ordinary) C&O Σ-protocol.

Definition 2.3 (Honest Verifier Zero-knowledge). *A Σ-protocol is honest verfier zero-knowledge (HVZK) if there exists an efficient simulator $\mathcal{S}$ such that the distributions of*

$$(a, c, z) \leftarrow \langle \mathcal{P}(x, w), \mathcal{V}(x) \rangle \qquad \textit{and} \qquad (a, c, z) \leftarrow \mathcal{S}(x)$$

are the same for any $(x, w) \in R$.

We consider two variants: *Special* honest verifier zero-knowledge and *non-abort* honest verifier zero-knowledge (naHVZK). For the former we require the simulator to first sample z and c and then compute a as a function of those (on top of the above condition on the distribution). Non-abort HVZK (Definition 2.5 in [KLS18]) applies to protocols where in an honest run the prover may abort and try again. The simulator may thus produce 'abort' transcripts as well (with the same distribution as the honest prover), and the c produced by the simulator is required to be uniform in $\mathcal{C}$ *conditioned* on the simulated transcript being non-abort.

2.2.2 Soundness notions for Σ-protocols

When considering an *adversary* $\mathcal{A}$ that tries to *forge* a proof for some instance $x \in \mathcal{X}$, one can distinguish between an *arbitrary but fixed* x, and an x that is *chosen* by $\mathcal{A}$ and output along with a. If x is fixed then the adversary is called *static*, otherwise it is called *adaptive*. For the typical security definitions for Σ-protocols this distinction between a static and an adaptive $\mathcal{A}$ makes no difference (see Lemmas 2.6 and 2.8 below).

Definition 2.4. Σ *is **(computationally/statistically) sound** if for any (quantum polynomial-time/unbounded) adversary $\mathcal{A}$ there exists a negligible function*

$\mu(\eta)$ such that for any $\eta \in \mathbb{N}$:

$$\Pr\left[\langle \mathcal{A}, \mathcal{V}(x)\rangle = \mathit{accept}\right] \leq \mu(\eta)$$

for all $x \notin \mathcal{L}$; respectively, in case of an **adaptive** *$\mathcal{A}$:*

$$\Pr\left[x \notin \mathcal{L} \wedge v = \mathit{accept} : (x, v) \leftarrow \langle \mathcal{A}, \mathcal{V}\rangle\right] \leq \mu(\eta)\,.$$

Remark 2.5. In line with Section 3.2, the description of a quantum algorithm $\mathcal{A}$ is understood to include the initial state $|\phi_0\rangle$. As such, when quantifying over all $\mathcal{A}$ it is understood that this includes a quantification over all $|\phi_0\rangle$ as well. This stays true when considering $\mathcal{A}$ to be quantum polynomial-time, which means that the unitaries A_i can be computed by polynomial-time quantum circuits, and q is polynomial in size, but does not put any restriction on $|\phi_0\rangle$.[15] This is in line with [Unr12, Definition 1], which explicitly spells out this quantification.

Lemma 2.6. *If Σ is computationally/statistically sound for* static *adversaries then it is also computationally/statistically sound for* adaptive *adversaries.*

Proof. Let $\mathcal{A}$ be an adaptive Σ-protocol adversary, producing x and a in the first stage, and z in the second stage. We then consider the following algorithms. $\mathcal{A}_{init}$ runs the first stage of $\mathcal{A}$ (using the same initial state), outputting x and a. Let $|\psi_{x,a}\rangle$ be the corresponding internal state at this point. Furthermore, for any possible x and a, $\mathcal{A}_{x,a}$ is the following static Σ-protocol adversary. Its initial state is $|\psi_{x,a}\rangle|a\rangle$ and in the first stage it simply outputs a, and in the second stage, after having received the verifier's challenge, it runs the second stage of $\mathcal{A}$. We then see that

$$\begin{aligned}
&\Pr\bigl[x \notin \mathcal{L} \wedge v = \mathit{accept} : (x, v) \leftarrow \langle \mathcal{A}, \mathcal{V}\rangle\bigr] \\
&\qquad = \sum_{x_\circ \notin \mathcal{L}} \Pr\bigl[x = x_\circ \wedge v = \mathit{accept} : (x, v) \leftarrow \langle \mathcal{A}, \mathcal{V}\rangle\bigr] \\
&\qquad = \sum_{x_\circ \notin \mathcal{L}} \sum_{a} \Pr\bigl[\mathcal{A}_{init} = (x_\circ, a)\bigr] \Pr\bigl[\langle \mathcal{A}_{x_\circ,a}, \mathcal{V}(x_\circ)\rangle = \mathit{accept}\bigr]\,.
\end{aligned}$$

Since $\Pr\bigl[\langle \mathcal{A}_{x_\circ,a}, \mathcal{V}(x_\circ)\rangle = \mathit{accept}\bigr]$ is bounded by a negligible function, given that $\mathcal{A}_{x,a}$ is a (quantum polynomial-time/unbounded) static adversary, the claim follows. □

[15] In other words, $\mathcal{A}$ is then *non-uniform* quantum polynomial-time with *quantum* advice.

We now recall the definition of a proof of knowledge, sometimes also referred to as (witness) extractability, tailored to the case of a negligible "knowledge error". Informally, the requirement is that if $\mathcal{A}$ succeeds in proving an instance x, then by using $\mathcal{A}$ as a black-box only it is possible to extract a witness for x. In case of an arbitrary but fixed x, this property is formalized in a rather straightforward way; however, in case of an adaptive $\mathcal{A}$, the formalization is somewhat subtle, because one can then not refer to *the* x for which $\mathcal{A}$ manages to produce a proof. We adopt the approach (though not the precise formalization) from [Unr17], which requires x to satisfy an arbitrary but fixed predicate.

Definition 2.7. *Σ is a* ***(computational/statistical) proof of knowledge*** *if there exists a quantum polynomial-time black-box 'knowledge extractor' $\mathcal{K}$,*[16] *a polynomial $p(\eta)$ and a constant $d \geq 0$, such that for any (quantum polynomial-time/unbounded) adversary $\mathcal{A}$ there exist a negligible function $\kappa(\eta)$ such that for any $\eta \in \mathbb{N}$ and any $x \in \mathcal{X}$ we have:*

$$\Pr\left[(x,w) \in R : w \leftarrow \mathcal{K}^{\mathcal{A}}(x)\right] \geq \frac{1}{p(\eta)} \cdot \Pr\left[\langle \mathcal{A}, \mathcal{V}(x)\rangle = accept\right]^d - \kappa(\eta)\,;$$

respectively, in case of an **adaptive** *$\mathcal{A}$:*

$$\begin{aligned}\Pr\big[x \in X \,\wedge\, (x,w) \in R : (x,w) \leftarrow \mathcal{K}^{\mathcal{A}}\big]&\\ \geq \frac{1}{p(\eta)} \cdot \Pr\left[x \in X \wedge v = accept : (x,v) \leftarrow \langle \mathcal{A}, \mathcal{V}\rangle\right]^d - \kappa(\eta)&\end{aligned}$$

for any subset $X \subseteq \mathcal{X}$.

Also here, static security implies adaptive security.

Lemma 2.8. *If Σ is a computational/statistical proof of knowledge for* static *$\mathcal{A}$ then it is also a computational/statistical proof of knowledge for* adaptive *$\mathcal{A}$.*

Proof. Let $\mathcal{A}$ be an adaptive Σ-protocol adversary, producing x and a in the first stage, and z in the second stage. We construct a black-box knowledge extractor $\mathcal{K}_{ad}$ that works for any such $\mathcal{A}$. In a first step, $\mathcal{K}_{ad}^{\mathcal{A}}$ runs the first stage of $\mathcal{A}$ using the black-box access to $\mathcal{A}$ (and having access to the initial state of $\mathcal{A}$). Below, we call this first stage of $\mathcal{A}$ as $\mathcal{A}_{init}$. This produces x and a, and we write $|\psi_{x,a}\rangle$ for the corresponding internal state. Then, it runs $\mathcal{K}_{na}^{\mathcal{A}^{x,a}}$, where

[16] Black-box refers to the fact that we require $\mathcal{K}$ to have only oracle access to $\mathcal{A}$, i.e. to provide it with an input and answer its queries before receiving the output, without knowing its code.

$\mathcal{K}_{na}$ is the knowledge extractor guaranteed to exist for static adversaries, and $\mathcal{A}^{x,a}$ is the static adversary that works as follows. It's initial state is $|\psi_{x,a}\rangle|a\rangle$ and in the first stage it simply outputs a, and in the second stage it runs the second stage of $\mathcal{A}$ on the state $|\psi_{x,a}\rangle$. Note that having obtained x and a and the state $|\psi_{x,a}\rangle$ as first step of $\mathcal{K}_{ad}^{\mathcal{A}}$, $\mathcal{K}_{na}^{\mathcal{A}^{x,a}}$ can then be executed with black box access to (the second stage of) $\mathcal{A}$. For any subset $X \subseteq \mathcal{X}$, we now see that

$$\begin{aligned}
&\Pr\big[x \in X \wedge (x,w) \in R : (x,w) \leftarrow \mathcal{K}_{ad}^{\mathcal{A}}\big] \\
&\quad= \sum_{x\in X}\sum_{a} \Pr\big[\mathcal{A}_{init} = (x,a)\big]\Pr\Big[(x,w) \in R : w \leftarrow \mathcal{K}_{na}^{\mathcal{A}^{x,a}}\Big] \\
&\quad\geq \sum_{x\in X}\sum_{a} \Pr\big[\mathcal{A}_{init} = (x,a)\big]\cdot\frac{1}{p(\eta)}\cdot\Pr\big[\langle\mathcal{A}^{x,a},\mathcal{V}(x)\rangle = \mathit{accept}\big]^d - \kappa(\eta) \\
&\quad\geq \frac{1}{p(\eta)}\bigg(\sum_{x\in X}\sum_{a} \Pr\big[\mathcal{A}_{init} = (x,a)\big]\Pr\big[\langle\mathcal{A}^{x,a},\mathcal{V}(x)\rangle = \mathit{accept}\big]\bigg)^d - \kappa(\eta) \\
&\quad= \frac{1}{p(\eta)}\Pr\big[x \in X \wedge v{=}1 : (x,v) \leftarrow \langle\mathcal{A}^{x,a},\mathcal{V}(x)\rangle\big]^d - \kappa(\eta)\,,
\end{aligned}$$

where the first inequality is because of the static proof-of-knowledge property, and the second is Jensen's inequality, noting that we may assume without loss of generality that $d \geq 1$. □

Remark 2.9. We do not necessarily require a Σ-protocol to be perfectly or statistically correct. This allows us to include protocols that use *rejection sampling*, where with a constant probability, the value z would leak too much information on the witness w and so the prover sends $\perp$ instead. On the other hand, by default we consider the soundness/knowledge error to be negligible, i.e., a dishonest prover succeeds only with negligible probability to make the verifier accept if x is not a valid instance or the prover has no witness for it (depending on the considered soundness notion). Negligible soundness/knowledge error can always be achieved by parallel repetition (see e.g. [Dam10]).

Special soundness. Many Σ-protocols satisfy a notion called special-soundness, which is very useful for proving the proof of knowledge property. Most common are the variants 2-special-soundness and k-special-soundness (also called k-soundness, in Chapter 3 t-soundness), but we develop a generalized definition that we call $\mathfrak{S}$-soundness in Section 4.5.2). Here we include the standard definition of k-soundness. Note that we only consider perfect special soundness, while there also exist computational variants in the literature.

Definition 2.10. Σ *is k-special-sound if there exists an efficient algorithm* $\mathcal{E}(x, a, S, \{z_c\}_{c \in S})$ *that takes as input an instance* x*, a first message* a*, a subset* $S \subseteq \mathcal{C}$ *of challenges, where* S *has cardinality* k*, and responses* z_c *for* $c \in S$*, and it outputs a witness for* x *if* $\mathcal{V}(x, a, c, z_c)$ *for all* $c \in S$*.*

2.2.3 The Fiat-Shamir transformation

The *Fiat-Shamir transformation* turns a Σ-protocol Σ into a non-interactive proof system, denoted $\mathsf{FS}[\Sigma]$, by replacing the verifier's random choice of $c \in \mathcal{C}$ with $c := H(x, a)$, where $H : \mathcal{X}' \to \mathcal{C}$ is a hash function with a domain $\mathcal{X}'$ that contains all pairs $x' = (x, a)$ with $x \in \mathcal{X}$ and a produced by $\mathcal{P}$. In other words, upon input x and w, the honest FS-prover produces $\pi = (a, z)$ by running the two-stage Σ-protocol prover $\mathcal{P}$ but using $c = H(x, a)$ as challenge (i.e., as input to the second stage). In case Σ is not statistically correct, the above process of producing $\pi = (a, z)$ is repeated sufficiently many times until $V(x, a, H(x, a), z)$ is satisfied (or some bound is reached). In either case, we will write this as

$$\pi = (a, z) \leftarrow P_{FS}^{H}(x, w) \,.$$

We may write as $V_{FS}^{H}(x, \pi)$ the FS-verifier's check whether $V(x, a, H(x, a), z)$ is satisfied or not. In the security analysis, the hash function H is modeled by a random oracle, i.e. by oracle access to a uniformly random $H : \mathcal{X}' \to \mathcal{C}$.

For $\mathsf{FS}[\Sigma]$ we can define security notions similar to those of Σ-protocols. Here again, when considering an *adversary* $\mathcal{A}$ that tries to *forge* a proof for some instance $x \in \mathcal{X}$, one can distinguish between an *arbitrary but fixed* x, and an x that is *chosen* by $\mathcal{A}$ and output along with π. If x is fixed then the adversary is called *static*, otherwise it is called *adaptive.* In contrast to the case of Σ-protocols, for the Fiat-Shamir transformation there may actually be a difference between static and adaptive security, where the latter is the stronger notion.

We have soundness:

Definition 2.11. $\mathsf{FS}[\Sigma]$ *is* ***(computationally/statistically) sound*** *if for any (quantum polynomial-time/unbounded) adversary* $\mathcal{A}$ *making* q *queries to the random-oracle, there exists a negligible function* $\mu(\eta)$ *and a constant* e *such that for any* $\eta \in \mathbb{N}$*:*

$$\Pr_{H}\left[V_{FS}^{H}(x, \pi) : \pi \leftarrow \mathcal{A}^{H}\right] \leq q^{e} \mu(\eta)$$

for all $x \notin \mathcal{L}$*; respectively, in case of an* **adaptive** $\mathcal{A}$*:*

$$\Pr_{H}\left[V_{FS}^{H}(x, \pi) \wedge x \notin \mathcal{L} : (x, \pi) \leftarrow \mathcal{A}^{H}\right] \leq q^{e} \mu(\eta) \,.$$

And the proof of knowledge property:

Definition 2.12. $\mathsf{FS}[\Sigma]$ *is a* ***(computational/statistical) proof of knowledge*** *if there exists a polynomial-time black-box 'knowledge extractor' $\mathcal{E}$, such that for any (quantum polynomial-time/unbounded) algorithm $\mathcal{A}$ making q queries to the random-oracle, a polynomial $p(\eta)$, constants $d, e \geq 0$, and a negligible function $\mu(\eta)$ such that for any $\eta \in \mathbb{N}$ and any $x \in \mathcal{X}$:*

$$\Pr\big[(x,w) \in R : w \leftarrow \mathcal{E}^{\mathcal{A}}(x)\big] \geq \frac{1}{q^e p(\eta)} \cdot \Pr_H \left[V_{FS}^H(x,\pi) : \pi \leftarrow \mathcal{A}^H\right]^d - \mu(\eta) \ ;$$

respectively, in case of an **adaptive** $\mathcal{A}$*:*

$$\begin{aligned}\Pr\big[x \in X \ \wedge \ (x,w) \in R : (x,w) \leftarrow \mathcal{E}^{\mathcal{A}}\big] \\ \geq \frac{1}{q^e p(\eta)} \cdot \Pr_H \left[x \in X \wedge V_{FS}^H(x,\pi) : (x,\pi) \leftarrow \mathcal{A}^H\right]^d - \mu(\eta)\end{aligned}$$

for any subset $X \subseteq \mathcal{X}$, where q is the number of queries $\mathcal{A}$ makes.

Remark 2.13. Note that for the soundness and proof of knowledge property of $\mathsf{FS}[\Sigma]$, the adversary $\mathcal{A}$'s success probability may unavoidably grow with the number q of oracle queries, but we require that it grows only polynomially in q.

Fiat-Shamir signatures Any Fiat-Shamir non-interactive proof system can easily be transformed into a public-key signature scheme.[17] The signer simply proves knowledge of a witness (the secret key) for a composite statement $x^* := x \| m$, which includes the public key x as well as the message m. The signature σ then consists of a proof for x^*.

Definition 2.14. *A binary relation R with instance generator G is said to be* hard *if for any quantum polynomial-time algorithm $\mathcal{A}$ there exist a negligible function $\mu(\eta)$ for which*

$$\Pr\left[(x,w') \in R : (x,w) \leftarrow G, w' \leftarrow \mathcal{A}(x)\right] \leq \mu(\eta)$$

where G is such that it always outputs a pair $(x,w) \in R$.

Definition 2.15. *A* Fiat-Shamir signature scheme *based on a Σ-protocol $\Sigma = (\mathcal{P}, \mathcal{V})$ for a hard relation R with instance generator G, denoted by* $\mathsf{Sig}[\Sigma]$ *is defined by the triple (Gen, Sign, Verify), with*

[17] In fact, that is how the Fiat-Shamir transform was originally conceived in [FS87]. Only later [BG93] adapted the idea to construct a non-interactive zero-knowledge proof system.

- Gen*: Pick* $(x, w) \leftarrow G$*, set* $sk := (x, w)$ *and* $pk := x$*.*
- $\mathsf{Sign}^H(sk, m)$*: Return* (m, σ) *where* $\sigma \leftarrow P_{FS}^H(x, m, w)$*.*
- $\mathsf{Verify}^H(pk, m, \sigma)$*: Return* $V_{FS}^H(x, m, \sigma)$*.*

Here $(P_{FS}^H, V_{FS}^H) = \mathsf{FS}[\Sigma^*]$*, where* $\Sigma^* = (\mathcal{P}^*, \mathcal{V}^*)$ *is the* Σ*-protocol obtained from* Σ *by setting* $\mathcal{P}^*(x, m) = \mathcal{P}(x)$ *and* $\mathcal{V}^*(x, m) = \mathcal{V}(x)$ *for any* m*.*

Note that by definition of FS in Section 2.2.3, we use $V_{FS}^H(x\|m, \sigma)$ as shortcut for $V(x\|m, a, H(x\|m, a), z)$.

We investigate the following standard security notions for signature schemes.

Definition 2.16 ($\mathsf{sEUF{-}CMA/EUF{-}NMA}$)**.** *A signature scheme fulfills* strong existential unforgeability under chosen-message attack ($\mathsf{sEUF-CMA}$) *if for all quantum polynomial-time algorithms* $\mathcal{A}$ *and for uniformly random* $H : \mathcal{X}' \to \mathcal{C}$ *it holds that*

$$\Pr\Big[\mathsf{Verify}^H(pk, m, \sigma) \wedge (m, \sigma) \notin \mathbf{Sig}{-}\mathsf{q} : (pk, sk) \leftarrow \mathsf{Gen}, (m, \sigma) \leftarrow \mathcal{A}^{H, \mathbf{Sig}}(pk)\Big]$$

is negligible. Here **Sig** *is a classical oracle which upon classical input* m *returns* $\mathsf{Sign}^H(m, sk)$*, and* $\mathbf{Sig}{-}\mathsf{q}$ *is the list of all queries made to* **Sig***.*

Analogously, a signature scheme fulfills existential unforgeability under no-message attack ($\mathsf{EUF-NMA}$) *if for all quantum polynomial-time algorithms* $\mathcal{A}$ *and for uniformly random* $H : \mathcal{X}' \to \mathcal{C}$ *it holds that*

$$\Pr\Big[\mathsf{Verify}^H(pk, m, \sigma) : (pk, sk) \leftarrow \mathsf{Gen}, (m, \sigma) \leftarrow \mathcal{A}^H(pk)\Big]$$

is negligible.

Section 2.3

Public-key encryption and key encapsulation

2.3.1 Definitions for PKE's and KEM's

Following the presentation of [HHK17] in general lines, we recall the formal definition of a public-key encryption scheme.

Definition 2.17 (Public-Key Encryption). *A* public-key encryption scheme PKE *consists of algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$*, a message space* $\mathcal{M}$*, a ciphertext space* $\mathcal{C}$ *and a set of random coins* $\mathcal{R}$*, such that for any* $m \in \mathcal{M}$*,* $r \in \mathcal{R}$

$$(sk, pk) \leftarrow \mathsf{Gen}\ , \quad \mathcal{C} \ni c \leftarrow \mathsf{Enc}_{pk}(m) \quad \textit{and} \quad \mathsf{Dec}_{sk}(c) \in \mathcal{M} \cup \{\bot\}\,.$$

The security notion for a public-key encryption scheme that we will use in this thesis is OW-CPA (One-Way Chosen Plaintext Attack) security.

Definition 2.18 (OW-CPA security of a PKE). *The security game* $\mathsf{OW\text{-}CPA}$ *is given in Figure 2.2. We define the* $\mathsf{OW\text{-}CPA}$ *advantage function of an adversary* $\mathcal{A}$ *against* PKE *as*

$$\mathsf{ADV}^{\mathsf{OW\text{-}CPA}}_{\mathrm{PKE}}[\mathcal{A}] := \Pr[\mathsf{OW\text{-}CPA}(\mathcal{A}) \Rightarrow 1].$$

A public-key encryption scheme PKE *is* $\mathsf{IND\text{-}CCA}$ *if for any quantum polynomial time algorithm* $\mathcal{A}$ *the advantage function* $\mathsf{ADV}[\mathcal{A}]^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}}$ *is negligible.*

```
GAME OW-CPA
  $(pk, sk) \leftarrow \mathsf{Gen}$
  $m^* \xleftarrow{\$} \mathcal{M}$
  $c^* \leftarrow \mathsf{Enc}_{pk}(m^*)$
  $m' \leftarrow \mathcal{A}(pk, c^*)$
  return $m' == m^*$
```

```
GAME IND-CCA-KEM
  $(pk, sk) \leftarrow \mathsf{Gen}$
  $b \xleftarrow{\$} \{0, 1\}$
  $(K_0^*, c^*) \leftarrow \mathsf{Encaps}(pk)$
  $K_1^* \xleftarrow{\$} \mathcal{K}$
  $b' \leftarrow \mathcal{A}^{\textsc{Decaps}}(c^*, K_b^*)$
  return $b' == b$
```

```
Decaps($c \neq c^*$)
  $K := \mathsf{Decaps}_{sk}(c)$
  return $K$
```

Fig. 2.2. Games for OW-CPA security of a PKE and IND-CCA security of a KEM. In the latter, $\mathcal{A}$ is not allowed to query c^* to Decaps.

For a given public-key encryption scheme, it may be useful to consider the probability of encountering decryption failures.

Definition 2.19 (δ-correctness). *A public-key encryption scheme is* δ*-correct if*

$$\mathop{\mathbb{E}}_{(sk,pk)\leftarrow \mathsf{Gen}} \left[\max_{m \in \mathcal{M}} \Pr\left[\mathsf{Dec}_{sk}(c) \neq m : c \leftarrow \mathsf{Enc}_{pk}(m)\right] \right] \leq \delta$$

where the probability is over the randomness of the encryption.

Another important property of encryption schemes is the min-entropy of a ciphertext given the plaintext, measured by their γ*-spreadness.*

Definition 2.20 (γ-spreadness). *A public-key encryption scheme is γ-spread if*

$$\min_{\substack{m\in\mathcal{M};\\(sk,pk)}} \Big(-\log \max_{c\in\mathcal{C}} \Pr\big[c = \mathsf{Enc}_{pk}(m)\big]\Big) \geq \gamma\,,$$

where the probability is over the randomness of the encryption, and the minimum is over all key pairs that have positive probability of being produced by Gen*.*

The above definition can be relaxed to an *expectation* over the choice of pk, when the expectation is done inside the negative logarithm.

Definition 2.21 (weak γ-spreadness). *A public-key encryption scheme is weakly γ-spread if*

$$-\log \mathop{\mathbb{E}}_{(sk,pk)\leftarrow\mathsf{Gen}} \left[\max_{\substack{m\in\mathcal{M}\\c\in\mathcal{C}}} \Pr\big[c = \mathsf{Enc}_{pk}(m)\big]\right] \geq \gamma\,,$$

where again the probability is over the randomness of the encryption.

A key-encapsulation mechanism (KEM) is defined as follows:

Definition 2.22 (Key Encapsulation Mechanism). *A* key encapsulation mechanism KEM *consists of algorithms* (Gen, Encaps, Decaps) *and a key space $\mathcal{K}$, where*

$$(sk, pk) \leftarrow \mathsf{Gen}\ , \quad (K, c) \leftarrow \mathsf{Encaps}(pk) \quad \textit{and} \quad \mathsf{Decaps}_{sk}(c) \in \mathcal{K} \cup \{\bot\}\,.$$

For a key encapsulation mechanism, the security notion that we consider in this thesis is that of IND-CCA (Indistinguishability under Chosen Ciphertext Attack).

Definition 2.23 (IND-CCA security of a KEM). *The security game* IND-CCA-KEM *is given in Figure 2.2. We define the* IND-CCA *advantage function of an adversary $\mathcal{A}$ against* KEM *as*

$$\mathsf{ADV}[\mathcal{A}]^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}} := |\Pr[\mathsf{IND\text{-}CCA\text{-}KEM}(\mathcal{A}) \Rightarrow 1] - \frac{1}{2}|.$$

A public-key encryption scheme PKE *is* IND-CCA *if for any quantum polynomial time algorithm $\mathcal{A}$ the advantage function $\mathsf{ADV}[\mathcal{A}]^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}}$ is negligible.*

Section 2.4

The compressed oracle technique

The results in Chapters 4 and 5 build upon and extend the compressed oracle framework [Zha19a]. We recall here (some version of) the technique. Consider the multi-register $D = (D_x)_{x\in\mathcal{X}}$, where the state space of D_x is given by $\mathcal{H}_{D_x} = \mathbb{C}[\{0,1\}^n \cup \{\perp\}]$, meaning that it is spanned by an orthonormal set of vectors $|y\rangle$ labelled by $y \in \{0,1\}^n \cup \{\perp\}$. The initial state is set to be $|\perp\rangle_D := \bigotimes_x |\perp\rangle_{D_x}$. Consider the unitary F defined by

$$F|\perp\rangle = |\hat{0}\rangle\ , \quad F|\hat{0}\rangle = |\perp\rangle \quad \text{and} \quad F|\hat{y}\rangle = |\hat{y}\rangle\ \forall\, y \in \{0,1\}^n \setminus \{0^n\}\,,$$

where $|\hat{y}\rangle := H|y\rangle$ with H the Walsh-Hadamard transform on $\mathbb{C}[\{0,1\}^n] = (\mathbb{C}^2)^{\otimes n}$. Exploiting the relation $|y\rangle = 2^{-n/2}\sum_\eta(-1)^{\eta\cdot y}|\hat{\eta}\rangle$, we see that

$$F|y\rangle = |y\rangle + 2^{-n/2}\left(|\perp\rangle - |\hat{0}\rangle\right)\ . \tag{2}$$

When the oracle is queried, a unitary O_{XYD}, acting on the query registers X and Y and the oracle register D, is applied, given by

$$O_{XYD} = \sum_x |x\rangle\langle x|_X \otimes O^x_{YD_x} \qquad \text{with} \qquad O^x_{YD_x} = F_{D_x}\mathrm{CNOT}_{YD_x}F_{D_x}\ , \tag{3}$$

where $\mathrm{CNOT}_{YD_x}|y\rangle|y_x\rangle = |y \oplus y_x\rangle|y_x\rangle$ for $y, y_x \in \{0,1\}^n$ and acts as the identity on $|y\rangle|\perp\rangle$.

As long as no other operations are applied to the state of D, this compressed oracle is perfectly indistinguishable from the quantum random oracle. Also, the support of the state of D_x then remains orthogonal to $|\hat{0}\rangle$ for any x. However, these properties may change when, e.g., measurements are performed on D. The oracle may then behave differently than the quantum random oracle, and the state of D may then have a non-trivial overlap with $|\hat{0}\rangle$. We note that, by the convention on CNOT to act trivially when the control register is in state $|\perp\rangle$, it holds that $O^x_{YD_x}|y\rangle|\hat{0}\rangle = |y\rangle|\hat{0}\rangle$.

When considering a *classical* query, which is a query with the XY-register in state $|x\rangle|0\rangle$ for some x, it is understood that the Y-register is then measured after the application of O_{XYD}. If D_x is in state ρ then a classical query on x will give response h with probability $\mathrm{tr}(|h\rangle\langle h|F\rho F)$ — unless ρ has nontrivial overlap with $|\hat{0}\rangle$ and $h = 0$, in which a classical query on x will give response 0 with probability $\mathrm{tr}(|0\rangle\langle 0|F\rho F) + \mathrm{tr}(|\bot\rangle\langle\bot|F\rho F)$. We note that, for any $h \in \mathcal{Y}$ and $\rho = |h\rangle\langle h|$,

$$\begin{aligned}\mathrm{tr}(|h\rangle\langle h|F\rho F) = |\langle h|F|h\rangle|^2 &= \Big|\langle h|\Big(|h\rangle + 2^{-n/2}(|\bot\rangle - |\hat{0}\rangle)\Big)\Big|^2 \\ &= \Big|1 - 2^{-n/2}\langle h|\phi_0\rangle\Big|^2 = \Big|1-2^{-n}\Big|^2 \geq 1 - 2\cdot 2^{-n}\,. \qquad (4)\end{aligned}$$

Vice-versa, after a classical query on x with response h, the state of D_x is $F|h\rangle$ — unless the state of D_x prior to the query had a nontrivial overlap with $|\hat{0}\rangle$ and $h = 0$, in this case, the state after the query is supported by $F|0\rangle$ and $F|\bot\rangle = |\hat{0}\rangle$.

Efficient representation of the compressed oracle. By the techniques of [Zha19a], it is possible to make the (considered variant of the) compressed oracle efficient. Concretely, by means of a suitable encoding, it is possible to *efficiently* maintain the quantum state of the register D of the compressed oracle, compute the unitary O_{XYD}, and extract information from the state of D. We briefly describe this procedure below.

Writing $\bar{\mathcal{Y}} = \{0,1\}^n \cup \{\bot\}$, consider the following standard sparse encoding scheme

$$\mathsf{SparseEnc}^q : \bar{\mathcal{Y}}^{\mathcal{X}} \to \mathcal{D} = (\mathcal{X} \times \bar{\mathcal{Y}})^q\,,$$

which maps any "database" $\mathbf{y} = (y_x)_{x\in\mathcal{X}}$ with at most q non-$\bot$ entries to the "compressed database"

$$\mathsf{SparseEnc}^q(\mathbf{y}) = \big((x_1, y_{x_1}), \dots, (x_s, y_{x_s}), (0,\bot), \dots, (0,\bot)\big)$$

of pairs (x, y_x) with $y_x \neq \bot$, sorted as $x_1 < \cdots < x_s$, and padded with $(0,\bot)$s. Naturally, we then set

$$\big|\mathsf{SparseEnc}^q(\mathbf{y})\big\rangle = |x_1\rangle|y_{x_1}\rangle \cdots |x_s\rangle|y_{x_s}\rangle|0\rangle|\bot\rangle \cdots |0\rangle|\bot\rangle \in \big(\mathbb{C}[\mathcal{X}] \otimes \mathbb{C}[\bar{\mathcal{Y}}]\big)^{\otimes q}$$

for any such $\mathbf{y}$. The crucial observations now are:

1. Using the representation $H^{\otimes|\mathcal{X}|}|\mathbf{y}\rangle \mapsto |\mathsf{SparseEnc}^q(\mathbf{y})\rangle$ for the state of register D after q queries, the evolution of the compressed oracle, given by O_{XYD}, is an efficiently quantum computable isometry (this was shown by Zhandry, but is also easy to see from scratch). Here and below, H is the Walsh-Hadamard transform on $\mathbb{C}[\{0,1\}^n] = (\mathbb{C}^2)^{\otimes n}$, extended to act as identity on $|\bot\rangle$.

2. Using the representation $|\mathbf{y}\rangle \mapsto |\mathsf{SparseEnc}^q(\mathbf{y})\rangle$ instead, it follows from basic theory of quantum computation that for any classical function f with domain $\bar{\mathcal{Y}}^{\mathcal{X}}$ and that is classically efficiently computable using the representation $\mathbf{y} \mapsto \mathsf{SparseEnc}^q(\mathbf{y})$, the unitary $U : |\mathbf{y}\rangle|z\rangle \mapsto |\mathbf{y}\rangle|z + f(\mathbf{y})\rangle$ is efficiently quantum computable.

3. $|\mathbf{y}\rangle \mapsto |\mathsf{SparseEnc}^q(\mathbf{y})\rangle$ commutes with applying Walsh-Hadamards to the $\mathbb{C}[\bar{\mathcal{Y}}]$-components. Therefore, one can efficiently switch between the two representations above, simply by applying $H^{\otimes q}$ to the corresponding registers of $|\mathsf{SparseEnc}^q(\mathbf{y})\rangle$.

Thus, using either of the two representations for representing the internal state of the oracle, both the evolution of the oracle and the typical unitaries or measurements used to "read out" information are efficiently quantum computable. For example, checking if $y_x = \bot$ for a given $x \in \mathcal{X}$, or if there exists $x \in \mathcal{X}$ for which x and y_x satisfy some given (efficiently computable) relation, etc. Formally:

Lemma 2.24. *Let $f : (\{0,1\}^n \cup \{\bot\})^{|\mathcal{X}|} \to \mathcal{T}$ be a function such that $\tilde{f} = f \circ \mathsf{SparseDec}^q$ can be computed in polynomial time in q. Then the measurement $\{\tilde{\Pi}^t\}_{t\in\mathcal{T}}$ given by the projections*

$$\tilde{\Pi}^t = \sum_{\mathbf{y}:\tilde{f}(\mathbf{y})=t} |\mathbf{y}\rangle\langle\mathbf{y}|$$

can be implemented in time linear in $\mathrm{Time}[\tilde{f}]$ *and thus in quantum polynomial time in q.*

Chapter 3

Security of the Fiat-Shamir Transformation

Chapter contents

Section 3.1
Introduction

The Fiat-Shamir transformation [FS87] (Section 2.2.3) turns any public-coin three-round interactive proof, i.e., any Σ-protocol, into a non-interactive proof in the (Q)ROM. In the classical case it is well known that the security properties of the Σ-protocol are inherited by the Fiat-Shamir transformation [BR93; FKMV12]. In the quantum setting, when considering the security of the Fiat-Shamir transformation against quantum dishonest provers in the QROM, mainly negative results are known — see below for a more detailed exposition of previous results and how they compare to the results in this chapter.

Measure-and-reprogram. The main technical result (Theorem 3.3) of the current chapter can be understood as a particular way to overcome — to some extent — the limitation in the QROM of not being able to "read out" any query to the random oracle and to then reprogram the corresponding hash value, as described in Section 1.1.3. Concretely, we achieve the following.

We consider an arbitrary quantum algorithm $\mathcal{A}$ that makes queries to the random oracle and in the end outputs a pair (x, z), where z is supposed to satisfy some relation with respect to $H(x)$, e.g., $z = H(x)$. We then show how to *extract* early on, by measuring one of the queries that $\mathcal{A}$ makes, the very x that $\mathcal{A}$ will output, and to *reprogram* the random oracle at the point x with a fresh random value $\varTheta$, with the effect that the pair (x, z) that $\mathcal{A}$ then outputs now satisfies the given relation with respect to $\varTheta$, with a not too large loss in probability.

The way this works is surprisingly simple. We choose the query that we measure uniformly at random among all the queries that $\mathcal{A}$ makes (also counting $\mathcal{A}$'s output), in order to (hopefully) obtain x. Subsequently we reprogram the RO, so as to answer x with $\varTheta$, *either* from this point on *or* from the following query on, where this binary choice is made at random. This last random decision seems counter-intuitive, but it makes our proof work. Indeed, we prove that the probability that (x, z) satisfies the required relation drops by no more than a factor $O(q^2)$, where q is the number of oracle queries $\mathcal{A}$ makes.

Application to the Fiat-Shamir transformation. It is quite easy to see that the above result on the reprogrammability of the random oracle is exactly what is needed to turn a quantum prover that attacks the Fiat-Shamir transformation into a quantum prover that attacks the underlying Σ protocol. Indeed, from any Fiat-Shamir dishonest prover $\mathcal{A}$ that tries to produce a proof $\pi = (a, z)$ for a statement x, we obtain an interactive dishonest prover for the Σ protocol that extracts a from $\mathcal{A}$ and sends it to the verifier, and then uses the received challenge c to reprogram the RO, so that the z output by $\mathcal{A}$ will be a correct reply with respect to c with a probability not much smaller than the probability that $\mathcal{A}$ succeeds in forging π in the QROM.

This gives us a very generic transformation (stated in Theorem 3.7 below) from a Fiat-Shamir dishonest prover to a Σ-protocol dishonest prover that is similarly successful, up to a loss in probability of order $O(q^2)$. Applied to the standard notions of soundness and proof-of-knowledge, we prove that both these security properties, in both the computational and the statistical variant, are preserved under the Fiat-Shamir transformation in the QROM (Corollaries 3.8 and 3.9).

Comparison with prior results. Before our contribution, the Fiat-Shamir tranform in the QROM was studied in a number of works [Unr17; DFG13; KLS18], where weaker security properties were shown. In addition, Unruh developed an alternative transform [Unr15b] that provided QROM security at the expense of an increased proof size. The Unruh transform was later generalized to apply to 5-round public coin interactive proof systems [CHR+18].

Mainly negative results were known about the security of the Fiat-Shamir transformation against quantum attacks. Figure 3.1 shows a table copied from [ARU14], which outlines the different negative results on the security of Σ-protocols against quantum attackers that carry over to the Fiat-Shamir transformation. All the potential positive claims on the security of the Fiat-Shamir transformation were left unanswered (see Figure 3.1).

The only known positive result on the security of the Fiat-Shamir transformation against quantum attacks was the result by Unruh [Unr17], which showed that *statistical* soundness carries over from a Σ-protocol to the Fiat-Shamir transformation.

Our generic transformation from a Fiat-Shamir dishonest prover to a Σ-protocol dishonest prover implies that *all* the (considered) security properties of the Σ-protocol carry over´ to the Fiat-Shamir transformation. Hence, we

Properties of $\boldsymbol{\Sigma}$-protocol		**$\boldsymbol{\Sigma}$-protocol directly**		**Fiat-Shamir transf.**	
special soundness	strict soundness	PoK	proof	PoK	proof
perf	comp	attack	stat	attack	**stat**
comp	comp	attack	attack	attack	attack
perf	perf	stat	stat	**stat**	**stat**

Fig. 3.1. Table adapted from [ARU14], showing which versions of *special soundness* and *strict soundness*, which we call *unique responses*, imply that the Σ-protocol is a *proof of knowledge (PoK)* or a *proof* (in the sense of ordinary soundness). The values comp, stat and perf mean that the considered property holds respectively computationally, statistically and perfectly, and attack means that there exist example schemes that allow an attack. Gray values are copied from [ARU14]. The last column shows that the negative results carry over to the Fiat-Shamir transformation, while our results (in **bold face**) complete the table by showing that also the positive results carry over. (Previously, only the lower right corner entry was known [Unr17].) If the computational version of the unique responses, or strict soundness, property is replaced by our quantum strengthening (Definition 3.24), all instances of attack can be replaced by **comp**.

show that all the three open settings from [ARU14] are statistically secure, as shown in Figure 3.1.[18]

We point out that [DFG13] claims an impossibility result about the soundness of the Fiat-Shamir transformation as a quantum proof of knowledge, which contradicts one of our implications above. However, their result only applies to a restricted notion of proof of knowledge where the extractor is not allowed to measure any of the adversary's queries to the random oracle. The rationale for this restriction was that such a measurement would disturb the adversary's quantum state beyond control; however, our technical result shows that it actually is possible to measure one of the adversary's queries and still have sufficient control over the adversary's behavior.

Placing prior negative results into perspective. At first glance, the negative results from [ARU14] together with our new positive results, as shown in the Fiat-Shamir column in Figure 3.1, seem to give a complete answer to the question of the security of the Fiat-Shamir transformation against quantum attacks. However, there is actually more to it.

We consider a *stronger* but still meaningful notion of *computational unique responses*, which is in the spirit of the *collapsing property* as introduced by Unruh [Unr16]. We call the new notion *quantum computationally unique responses*

[18] In the (quantum) random-oracle model, *statistical* security considers a computationally unbounded attacker with a polynomially bounded number of oracle queries.

and define it in Definition 3.24. Adapting a proof from [Unr12], it is not hard to see that a Σ-protocol with (perfect or computational) special soundness and quantum computational unique responses is a computational proof of knowledge. Therefore, our result then implies that its Fiat-Shamir transformation is a computational proof of knowledge as well.

Finally, our result also implies that if the Σ-protocol is *computationally sound* (as a 'proof'), then its Fiat-Shamir transformation is computationally sound as well. Interestingly, Unruh seems to suggest in [Unr17] (right after Theorem 21) that this is not true in general, due to a counterexample from [ARU14]. The counter example is, however, a Σ-protocol that is computationally *special* sound but not computationally sound (the issue being that in the quantum setting, special soundness does not imply ordinary soundness).

Thus, with the right adjustments of the considered *computational* soundness properties, the three negative answers in the Fiat-Shamir column in Figure 3.1 may actually be turned into positive answers. One caveat here is that we expect proving quantum computationally unique responses to be much harder than computational unique responses.

Application to signatures. Our positive results on the Fiat-Shamir transformation have direct applications to the security of Fiat-Shamir signatures. From the proof-of-knowledge property of the Fiat-Shamir transformation we immediately obtain the security of the Fiat-Shamir signature scheme under a *no-message attack*, assuming that the public key is a hard instance (Theorem 3.33). Furthermore, [Unr17] and [KLS18; BBD+23] have shown that for Fiat-Shamir signatures, up to some loss in the security parameter and under some additional mild assumptions on the underlying Σ-protocol, one can also derive security under *chosen-message attack*.

In conclusion, Fiat-Shamir signatures offer security against quantum attacks (in the QROM) if the underlying Σ-protocol is a proof of knowledge against quantum attacks and satisfies a few additional natural assumptions (Theorem 3.34).

As a concrete application, using Unruh's result[19] on the collapsing property of the RO [Unr16] to argue the collapsing version of computational unique responses (which we call *quantum* computational unique responses) for the underlying Σ-protocol, we can conclude that the non-optimized version of Fish, which is the Fiat-Shamir variant of Picnic, is secure in the QROM.

[19] The proof in [Unr16] has been shown incomplete, but the updated eprint version [Unr15a] contains a new proof.

Comparison with concurrent results. In concurrent and independent work [LZ19a][20], Liu and Zhandry show results that are very similar to ours: they also show the security of the Fiat-Shamir transformation in the QROM, and they introduce a similar stronger version of the computational unique responses property in order to argue that a Σ-protocol is a (computational) proof of knowledge against a quantum adversary. In short, [LZ19a] differs from the work here in the following aspects. In [LZ19a], the result on the Fiat-Shamir transformation is obtained using a very different approach, resulting in a greater loss in the reduction: $O(q^9)$ compared to the $O(q^2)$ loss that we obtain. On the other hand, on the quantum proof of knowledge front, Liu and Zhandry introduce some additional techniques that, for instance, allow them to prove that the Σ-protocol underlying Dilithium satisfies (their variant) of the newly introduced strong version of the computational unique responses property, while we phrase this as a conjecture in order to conclude the security of (some variant of) the Dilithium signature scheme.

Measure-and-reprogram 2.0. Given important examples of *multi-round* public-coin interactive proofs, used in, e.g., MQDSS [CHR+16] and for Bulletproofs [BBB+18], a natural question that arises is whether the measure-and-reprogram technique extends to the reprogrammability of the QROM at *multiple* inputs and the security of the Fiat-Shamir transformation (in the QROM) of *multi-round* public-coin interactive proofs. Another question is whether the $O(q^2)$ loss (for the original Σ-protocols) is optimal, or whether one might hope for a linear loss as in the classical case.

A technical hurdle for generalizing to multi-round Fiat-Shamir. To answer the first question, we observe that the naive approach of applying our original result from [DFMS19] (Lemma 3.1 and Equation 5 in this thesis) inductively so as to reprogram multiple inputs one by one does not work. This is due to a subtle technical issue that has to do with the precise statement of the original result. In more detail, the statement involves an additive error term $\varepsilon_x \geq 0$ that depends on the particular choice of the point x, which is (adaptively) chosen to be the input on which the random oracle (RO) is reprogrammed. The guarantee provided in [DFMS19] is that this error term stays negligible even *when summed over* all x's, i.e., $\sum_x \varepsilon_x = negl$. The formulation of the result for individual x's with control over $\sum_x \varepsilon_x$ is important for the later applications

[20] The paper [LZ19a] was put on eprint (`ia.cr/2019/262`) a few days after our eprint version (`ia.cr/2019/190`).

to the Fiat-Shamir transformation. However, when applying the result twice in a row, with the goal being to reprogram the random oracle at two inputs x_1, x_2, then we end up with two error terms ε_{x_1} and $\varepsilon_{x_2}^{x_1}$ (with the second one depending on x_1), where the first one stays negligible when summed over x_1 and the second one stays negligible when summed over x_2 (for any x_1); but it is unclear that the sum $\varepsilon_{x_1,x_2} := \varepsilon_{x_1} + \varepsilon_{x_2}^{x_1}$ stays negligible when summed over x_1 *and* x_2, which is what we would need to get the corresponding generalized statement.

An improved version of the technique. Our work [DFM20] revised the *original* result from [DFMS19] of reprogramming the QROM at one input by showing an *improved* version that has *no* additive error term, but only the original multiplicative $O(q^2)$ loss. In Section 3.2.2 we present both side by side for comparison, but we omit the proof of the original result.

For typical direct cryptographic applications, the improvement makes no big quantitative difference due to the error term being negligible, but: (1) it makes the statement cleaner and easier to formulate, (2) somewhat surprisingly, the proof is simpler than that of the original result in [DFMS19], and (3) most importantly, it removes the technical hurdle to extend to multiple inputs. Indeed, we then get the desired multi-input reprogrammability result by means of a not too difficult, though somewhat tedious, induction argument.

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

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

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

As a second application of our multi-input reprogrammability result, we show security (in the QROM) of the non-interactive OR-proof introduced by Liu, Wei and Wong [LWW04], further analyzed by Fischlin, Harasser and Janson [FJ20]. While the well-known (interactive) OR-proof by Cramer, Damgård and Schoenmakers [CDS94] is a Σ-protocol and thus the results from [DFMS19] apply, the inherently non-interactive OR-proof by Liu et al. does *not* follow this blueprint of being obtained as the Fiat-Shamir transformation of a Σ-protocol (though in some sense it is "close" to being of this form). We show here how the 2-input version of our multi-input reprogrammability result implies security of this OR-proof in the QROM.

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

Section 3.2

Reprogramming the quantum random oracle

We show and analyze a particular way to reprogram a random oracle in the quantum setting, where the oracle can be queried in superposition.

3.2.1 Notation

In line with Section 2.1.3, we consider a quantum oracle algorithm $\mathcal{A}$ that makes q queries to an *oracle*, i.e., an unspecified function $H : \mathcal{X} \to \mathcal{Y}$ with finite non-empty sets $\mathcal{X}, \mathcal{Y}$. We may assume without loss of generality that $\mathcal{A}$ makes no intermediary measurements (see Section 2.1.2 for a proof of this fact). Formally, $\mathcal{A}$ is then described by a sequence of unitaries $A_1, \ldots, A_q$ and an initial state $|\phi_0\rangle$.[21] The unitaries A_i act on registers $\mathsf{X}, \mathsf{Y}, \mathsf{Z}, \mathsf{E}$, where X and Y have respective $|\mathcal{X}|$- and $|\mathcal{Y}|$-dimensional state spaces, while Z and E is arbitrary. As will become clear, X and Y are the quantum registers for the queries to H as well as for the final output x, Z is for the output z, and E is internal memory. For any concrete choice of $H : \mathcal{X} \to \mathcal{Y}$, we can write

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

for the execution of $\mathcal{A}$ with the oracle instantiated by H, where $\mathcal{O}^H$ is the unitary $\mathcal{O}^H : |x\rangle|y\rangle \mapsto |x\rangle|y \oplus H(x)\rangle$ that acts on registers X and Y.

It will be convenient to introduce the following notation. For $0 \leq i, j \leq q$ we set

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

with the convention that $\mathcal{A}^H_{i\to j} := \mathbb{1}$ for $j \leq i$. Furthermore, we set

$$|\phi^H_i\rangle := \big(\mathcal{A}^H_{0\to i}\big)|\phi_0\rangle$$

to be the state of $\mathcal{A}$ after the i-th step but right before the $(i+1)$-st query, and so that $|\phi^H_q\rangle$ equals $\big(\mathcal{A}^H_{0\to q}\big)|\phi_0\rangle = \mathcal{A}^H|\phi_0\rangle$, the output state produced by $\mathcal{A}$.

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

$$\big(\mathcal{A}^{H*\Theta x}_{i\to q}\big)\,\big(\mathcal{A}^H_{0\to i}\big)\,|\phi_0\rangle = \big(\mathcal{A}^{H*\Theta x}_{i\to q}\big)|\phi^H_i\rangle$$

for an execution of $\mathcal{A}$ where the oracle is reprogrammed at a given point x after the i-th query.

We are interested in the probability that after the execution of $\mathcal{A}^H$ and upon measuring register X in the computational basis to obtain $x \in \mathcal{X}$, the state of register Z is of a certain form dependent on x and $H(x)$. This relation is captured by a projection G^H_x, where, more generally, for $x, x' \in \mathcal{X}$ and $\Theta \in \mathcal{Y}$ we set

$$G^\Theta_{x,x'} = |x'\rangle\langle x'| \otimes \mathbb{1} \otimes \Pi_{\,x,\Theta} \otimes \mathbb{1},$$

[21] Alternatively, we may understand $|\phi_0\rangle$ as an auxiliary input given to $\mathcal{A}$.

where $\{\Pi_{x,\Theta}\}_{x\in\mathcal{X},\Theta\in\mathcal{Y}}$ is a family of projections acting on Z, which we refer to as a *quantum predicate*. We use the short hands G_x^{Θ} for $G_{x,x}^{\Theta}$ and G_x^H for $G_x^{H(x)}$, i.e.,

$$G_x^H = |x\rangle\langle x| \otimes \mathbb{1} \otimes \Pi_{\,x,H(x)} \otimes \mathbb{1}.$$

For an arbitrary but fixed $x_\circ \in \mathcal{X}$, we then consider the probability

$$\| G_{x_\circ}^H |\phi_q^H\rangle \|_2^2 \,.$$

Understanding $\mathcal{A}^H$ as an algorithm that outputs the measured x together with the state z in register Z, we will denote this probability also by

$$\Pr\big[x = x_\circ \wedge V(x, H(x), z) : (x, z) \leftarrow \mathcal{A}^H\big] \,,$$

understanding V to be a quantum predicate specified by the projections $\Pi_{x,H(x)}$.

3.2.2 Main technical result

As explained in the introduction, [DFM20] contains an improvement of the original measure-and-reprogram theorem from [DFMS19]. In this section we present both for comparison, but omit the proof of the technical lemma (Lemma 3.1) that underlies the original theorem.

We consider a quantum oracle algorithm $\mathcal{A}$ as formalized above, and we define a two-stage algorithm $\mathcal{S}$ with black-box access to $\mathcal{A}$ as follows. In the first stage, $\mathcal{S}$ tries to predict $\mathcal{A}$'s future output x, and then, upon input a (random) Θ, in the second stage tries to output what $\mathcal{A}$ is supposed to output, but now with respect to Θ instead of $H(x)$.

$\mathcal{S}$ works by running $\mathcal{A}$, but with the following modifications. First, one of the $q+1$ queries of $\mathcal{A}$ (also counting the final output in register X) is selected uniformly at random and this query is measured, and the measurement outcome x is output by (the first stage of) $\mathcal{S}$. Then, this very query of $\mathcal{A}$ is answered either using the original H *or* using the reprogrammed oracle $H * \Theta x$, with the choice being made at random, while all the remaining queries of $\mathcal{A}$ are answered using oracle $H * \Theta x$.[22] Finally, (the second stage of) $\mathcal{S}$ outputs whatever $\mathcal{A}$ outputs.

Here, the figure of merit is the probability that for a fixed x, both the intermediate measurement and a measurement of the register X return x *and* that the register Z contains a state that satisfies the considered quantum predicate with respect to x and its (now reprogrammed) hash value Θ. Formally, this probability is captured by

[22] If it is the final output that is measured then there is nothing left to reprogram.

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

where here and from now on, we use X as a short hand for the projection $|x\rangle\langle x|$ acting on X. The expectation is taken over $\Theta \in \mathcal{Y}$, $i \in \{0,...,q\}$ and $b \in \{0,1\}$ uniformly random. Note that the random bit $b \in \{0,1\}$ determines whether the measured query is answered with H or with $H*\Theta x$.

We write $\mathcal{S}^{\mathcal{A}}[H]$ to emphasize that $\mathcal{S}$ only makes black-box access to $\mathcal{A}$ and that it depends on H. Our main technical lemma below then ensures that for any H and for a random $\Theta \in \mathcal{Y}$, the success probability of $\mathcal{S}^{\mathcal{A}}[H]$ is up to an order-q^2 loss not much smaller than that of $\mathcal{A}^{H*\Theta x}$, and therefore not much smaller than that of $\mathcal{A}^H$ in case of a random H.

Lemma 3.1 (Original version from [DFMS19]). *For any $H : \mathcal{X} \to \mathcal{Y}$ and $x \in \mathcal{X}$, it holds that*

$$\mathop{\mathbb{E}}_{\Theta,i,b}\left[\left\|G_x^{\Theta}\left(\mathcal{A}_{i+b\to q}^{H*\Theta x}\right)\left(\mathcal{A}_{i\to i+b}^{H}\right)X|\phi_i^H\rangle\right\|_2^2\right] \geq \frac{\mathbb{E}_{\Theta}\left[\left\|G_x^{\Theta}|\phi_q^{H*\Theta x}\rangle\right\|_2^2\right]}{2(q+1)(2q+3)} - \frac{\left\|X|\phi_q^H\rangle\right\|_2^2}{2(q+1)|\mathcal{Y}|}.$$

where the expectation is over random $\Theta \in \mathcal{Y}$, $i \in \{0,\dots,q\}$ and $b \in \{0,1\}$.[23]

The proof of this version of the lemma is omitted in this thesis, it is available as the proof of Lemma 1 in [DFMS19].

Introducing more algorithmic-probabilistic notation, we write

$$(x, x', z) \leftarrow \langle \mathcal{S}^{\mathcal{A}}[H], \Theta \rangle$$

to specify the probability space determined as follows, relying on the above construction of the two-stage algorithm $\mathcal{S}$ when given $\mathcal{A}$. In the first stage $\mathcal{S}^{\mathcal{A}}[H]$ produces x, and then in the second stage, upon receiving Θ, it produces x' and z, where z may be quantum. Our figure of merit above, i.e., the left hand side of the bound in Lemma 3.1 (with x replaced by $x_\circ$), is then denoted by

$$\Pr_{\Theta}\left[x = x_\circ \wedge x' = x_\circ \wedge V(x, \Theta, z) : (x, x', z) \leftarrow \langle \mathcal{S}^{\mathcal{A}}[H], \Theta \rangle\right],$$

where the subscript Θ in $\Pr_\Theta$ denotes that the probability is averaged over a random choice of Θ.

[23] We consider $|\mathcal{Y}|$ to be superpolynomial in the security parameter, so that $\frac{1}{2(q+1)|\mathcal{Y}|}$ is negligible and can be neglected. In cases where $|\mathcal{Y}|$ is polynomial, the presented bound is not optimal, but an improved bound can be derived with the same kind of techniques. However, the improved variant Lemma 3.3 does not suffer from this impairment.

Using this notation, but also weakening the bound slightly by not requiring $x' = x_\circ$ (thus dropping x' from the output of $\mathcal{S}$), for any H and $x_\circ$ the bound from Lemma 3.1 then becomes

$$\begin{aligned}&\Pr_{\Theta}\big[x{=}x_\circ \wedge V(x,\Theta,z) : (x,z) \leftarrow \langle \mathcal{S}^{\mathcal{A}}[H],\Theta\rangle\big]\\ &\qquad \gtrsim \frac{1}{O(q^2)}\Pr_{\Theta}\big[x{=}x_\circ \wedge V(x,\Theta,z) : (x,z) \leftarrow \mathcal{A}^{H*\Theta x}\big]\end{aligned} \tag{5}$$

where the approximate inequality $\gtrsim$ hides the term

$$\epsilon_{x_\circ} := \frac{1}{2(q+1)|\mathcal{Y}|}\Pr_{H}\big[x{=}x_\circ : (x,z) \leftarrow \mathcal{A}^{H}\big].$$

Recall that the output z may be a quantum state, in which case the predicate V is given by a measurement that depends on x, and $H(x)$ or Θ, respectively.

We now fix a family $\mathcal{H}$ of $2(q+1)$-wise independent hash functions and average the above inequality over a random choice of $H \in \mathcal{H}$ from this family. We simply write $\mathcal{S}$ for $\mathcal{S}[H]$ with H chosen like that. Furthermore, we observe that, for any fixed x, the family $\{H * \Theta x \,|\, H \in \mathcal{H}, \Theta \in \{0,1\}^n\}$ is a family of $2(q+1)$-wise independent hash functions as well. Finally, we use that $\mathcal{A}$ (together with the check $V(x, H(x), z)$) cannot distinguish a random function $H{*}\Theta x$ in that family from a fully random function H [Zha12]. We then obtain the following result from Lemma 3.1.

Theorem 3.2 (Measure-and-reprogram, deprecated version from [DFMS19]). *Let $\mathcal{X}, \mathcal{Y}$ be finite non-empty sets. There exists a black-box two-stage quantum algorithm $\mathcal{S}$ with the following property. Let $\mathcal{A}$ be an arbitrary oracle quantum algorithm that makes q queries to a uniformly random $H : \mathcal{X} \to \mathcal{Y}$ and that outputs some $x \in \mathcal{X}$ and a (possibly quantum) output z. Then, the two-stage algorithm $\mathcal{S}^{\mathcal{A}}$ outputs some $x \in \mathcal{X}$ in the first stage and, upon a random $\Theta \in \mathcal{Y}$ as input to the second stage, a (possibly quantum) output z, so that for any $x_\circ \in \mathcal{X}$ and any (possibly quantum) predicateV:*

$$\begin{aligned}&\Pr_{\Theta}\big[x{=}x_\circ \wedge V(x,\Theta,z) : (x,z) \leftarrow \langle \mathcal{S}^{\mathcal{A}},\Theta\rangle\big]\\ &\qquad \gtrsim \frac{1}{O(q^2)}\Pr_{H}\big[x{=}x_\circ \wedge V(x,H(x),z) : (x,z) \leftarrow \mathcal{A}^{H}\big],\end{aligned}$$

where the $\gtrsim$ hides a term that is bounded by $\frac{1}{2q|\mathcal{Y}|}$ when summed over all $x_\circ$.

We go on to show the improved variant (from [DFM20]) of Theorem 3.2, which avoids the additive error term $\epsilon_{x_\circ}$. While having negligible quantitative effect in typical situations, it makes the statement simpler. In addition, as explained in the introduction, it circumvents a technical issue one encounters when trying to extend to the multi-input case. Furthermore, the improved version comes with a simpler proof.[24]

The approach is to avoid the additive error term in Lemma 3.1. We achieve this by slightly tweaking the simulator $\mathcal{S}$. From the technical perspective, while on the left hand side of Lemma (3.1) the expectation is over a random $i \in \{0,\dots,q\}$, selecting one of the $q+1$ queries of $\mathcal{A}$ at random (where the X register of the output state is considered to be a final query), and a random $b \in \{0,1\}$, our new version has syntactically the same left hand side, but with the expectation over a random pair $(i,b) \in (\{0,\dots,q\text{-}1\} \times \{0,1\}) \cup \{(q,0)\}$ instead. This allows us to absorb the additive error term into the success probability of the simulator. Furthermore, it holds for any *fixed* choice of Θ (and not only on average for a random choice).

Lemma 3.3 (Improved variant from [DFM20]). *Let $\mathcal{A}$ be a q-query oracle quantum algorithm. Then, for any function $H : \mathcal{X} \to \mathcal{Y}$, any $x \in \mathcal{X}$ and $\Theta \in \mathcal{Y}$, and any projection $\Pi_{x,\Theta}$, it holds that*

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

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

This new version of Lemma 3.1 translates to a simulator $\mathcal{S}$ that works by running $\mathcal{A}$, but with the following modifications. First, one of the $q+1$ queries of $\mathcal{A}$ (also counting the final output in register X) is measured, and the measurement outcome x is output by (the first stage of) $\mathcal{S}$. We emphasize that the crucial difference to the old version is that each of the q actual queries is picked with probability $\frac{2}{2q+1}$, while the final output is picked with probability $\frac{1}{2q+1}$. Then, very much as before, this very query of $\mathcal{A}$ is answered either using the original H *or* using the reprogrammed oracle $H*\Theta x$, with the choice being made at random[25], while all the remaining queries of $\mathcal{A}$ are answered using oracle $H*\Theta x$. Finally, (the second stage of) $\mathcal{S}$ outputs whatever $\mathcal{A}$ outputs. We thus get the following result.

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

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

Theorem 3.4 (Measure-and-reprogram, single input). *Let $\mathcal{X}$ and $\mathcal{Y}$ be finite non-empty sets. There exists a black-box two-stage quantum algorithm $\mathcal{S}$ with the following property. Let $\mathcal{A}$ be an arbitrary oracle quantum algorithm that makes q queries to a uniformly random $H : \mathcal{X} \to \mathcal{Y}$ and that outputs some $x \in \mathcal{X}$ and a (possibly quantum) output z. Then, the two-stage algorithm $\mathcal{S}^{\mathcal{A}}$ outputs some $x \in \mathcal{X}$ in the first stage and, upon a random $\Theta \in \mathcal{Y}$ as input to the second stage, a (possibly quantum) output z, so that for any $x_\circ \in \mathcal{X}$ and any (possibly quantum) predicate V:*

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

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

The proof of Lemma 3.3 follows closely the proof of Lemma 3.1 (available as Lemma 1 in [DFMS19]), but the streamlined statement and simulator allow to cut some corners.

Proof (of Lemma 3.3). For any $0 \leq i \leq q$, inserting a resolution of the identity and exploiting that

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

we can write

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

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

$$\begin{aligned}\big\|G^{\Theta}_x\big(\mathcal{A}^{H*\Theta x}_{i\to q}\big)|\phi^H_i\rangle\big\|_2 \leq \big\|G^{\Theta}_x\big(\mathcal{A}^{H*\Theta x}_{i+1\to q}\big)|\phi^H_{i+1}\rangle\big\|_2&\\ + \big\|G^{\Theta}_x\big(\mathcal{A}^{H*\Theta x}_{i\to q}\big)X|\phi^H_i\rangle\big\|_2&\\ + \big\|G^{\Theta}_x\big(\mathcal{A}^{H*\Theta x}_{i+1\to q}\big)\big(\mathcal{A}^{H}_{i\to i+1}\big)X|\phi^H_i\rangle\big\|_2&\,.\end{aligned}$$

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

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

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

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

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

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

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

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

□

Remark 3.5. We do not spell out in detail what it means for a quantum algorithm like $\mathcal{S}$ to be *black-box*; see e.g. [Unr17] for a rigorous definition. What we obviously need here is that $\mathcal{S}^{\mathcal{A}}$ has access to $\mathcal{A}$'s initial state $|\phi_0\rangle$ and to q, and is given black-box access to the unitaries A_i. Furthermore, for later purposes, we need the following composition property: if $\mathcal{S}$ is a black-box algorithm with access to $\mathcal{A}$, and $\mathcal{K}$ is a black-box algorithm with access to $\mathcal{S}^{\mathcal{A}}$, then there exists a black-box algorithm $\mathcal{K}^{\mathcal{S}}$ with access to $\mathcal{A}$ so that $(\mathcal{K}^{\mathcal{S}})^{\mathcal{A}} = \mathcal{K}^{(\mathcal{S}^{\mathcal{A}})}$.

Section 3.3
Security of the Fiat-Shamir transformation

In this section, we show how to reduce security of the Fiat-Shamir transformation (Section 2.2.3) to the security of the underlying Σ-protocol (Definition 2.2): any dishonest prover attacking the Fiat-Shamir transformation can be turned into a dishonest prover that succeeds to break the underlying Σ-protocol with the same probability up to a polynomial loss. This reduction is obtained by a straightforward application of Theorem 3.4. Our security reduction holds very generically and is not strongly tied to the considered notion of security, as long as the respective security definitions for the Σ-protocol and the Fiat-Shamir transformation "match up".

3.3.1 The generic security reduction

Since an adaptive adversary is clearly not less powerful than a static adversary, we restrict our attention for the moment to the adaptive case. Recall that such an adaptive FS-adversary $\mathcal{A}$ outputs the instance $x \in \mathcal{X}$ along with the proof $\pi = (a, z)$, and the figure of merit is the probability that x, a, z satisfies $V(x, a, H(x, a), z)$. Thus, we can simply apply Theorem 3.4, with (x, a) playing the role of what is referred to as x in the theorem statement, to obtain the existence of an adaptive Σ-adversary $\mathcal{S}^{\mathcal{A}}$ that produces (x, a) in a first stage, and upon receiving challenge c produces z, such that for any $x_\circ \in \mathcal{X}$

$$\begin{aligned}&\Pr\big[x = x_\circ \wedge V(x, a, c, z) : (x, a, z) \leftarrow \langle \mathcal{S}^{\mathcal{A}}, c\rangle\big] \\ &\qquad\qquad \geq \frac{1}{(2q+1)^2} \Pr_H\big[x = x_\circ \wedge V(x, a, H(x, a), z) : (x, a, z) \leftarrow \mathcal{A}^H\big] .\end{aligned}$$

Understanding that x is given to $\mathcal{V}$ along with the first message a but also treating it as an output of $\mathcal{S}^{\mathcal{A}}$, while $\mathcal{V}$'s output v is its decision to accept or not, we write this as

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

Summed over all $x_\circ \in \mathcal{X}$, this in particular implies that

$$\Pr\big[\langle \mathcal{S}^{\mathcal{A}}, \mathcal{V}\rangle = \mathit{accept}\big] \geq \frac{1}{(2q+1)^2} \Pr_H\big[V^H_{FS}(x, \pi) : (x, \pi) \leftarrow \mathcal{A}^H\big] .$$

Remark 3.6. We point out that the above arguments extend to a FS-adversary $\mathcal{A}$ that, besides the instance x and the proof $\pi = (a, z)$, also produces some local (possibly quantum) output satisfying some (quantum) predicate that may depend on x, a, z. The resulting Σ-adversary $\mathcal{S}^{\mathcal{A}}$ is then ensured to produce a local output that satisfies the considered predicate as well, up to the given loss in the probability. Indeed, we can simply include this local output in z and extend the predicate V accordingly.

In a very broad sense, the above means that for *any* FS-adversary $\mathcal{A}$ there exists a Σ-adversary $\mathcal{S}^{\mathcal{A}}$ that "*achieves the same thing*" up to a $(2q+1)^2$ loss in success probability. Hence, for matching corresponding security definitions, security of a Σ-protocol (against a dishonest prover) implies security of its Fiat-Shamir transform.

We summarize here the above basic transformation from an adaptive FS-adversary $\mathcal{A}$ to an adaptive Σ-adversary $\mathcal{S}^{\mathcal{A}}$.

Theorem 3.7. *There exists a black-box quantum polynomial-time two-stage quantum algorithm $\mathcal{S}$ such that for any adaptive Fiat-Shamir adversary $\mathcal{A}$, making q queries to a uniformly random function H with appropriate domain and range, and for any $x_\circ \in \mathcal{X}$:*

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

Below, we apply the above general reduction to the respective standard definitions for *soundness* and *proof of knowledge.* Each property comes in the variants *computational* and *statistical*, for guarantees against computationally bounded or unbounded adversaries respectively, and one may consider the static or the adaptive case.

3.3.2 Preservation of soundness

Let $\Sigma = (\mathcal{P}, \mathcal{V})$ be a Σ-protocol for a relation R, and let $\mathsf{FS}[\Sigma]$ be its Fiat-Shamir transformation. We set $\mathcal{L} := \{x \in \mathcal{X} \mid \exists\, w \in \mathcal{W} : R(x,w)\}$. It is understood that $\mathcal{P}$ and $\mathcal{V}$, as well as R and thus $\mathcal{L}$, may depend on a security parameter η. We note that in the following definition, we overload notation a bit by writing $\mathcal{A}$ for both for the ordinary static and for the adaptive adversary (even though a given $\mathcal{A}$ is usually either static or adaptive).

The following is now an immediate application of Theorem 3.7 and Lemma 2.6.

Corollary 3.8. *Let Σ be a Σ-protocol. If Σ is computationally/statistically sound against a static adversary then* $\mathsf{FS}[\Sigma]$ *is computationally/statistically sound against an adaptive adversary.*

Proof. Applying Theorem 3.7, we find that for any adaptive FS-adversary $\mathcal{A}$, polynomially bounded in the computational setting, there exists an adaptive Σ-protocol adversary $\mathcal{S}^{\mathcal{A}}$, polynomially bounded if $\mathcal{A}$ is, so that

$$\begin{aligned}
&\Pr\big[x \notin \mathcal{L} \,\wedge\, V_{FS}^{H}(x,\pi) : (x,\pi) \leftarrow \mathcal{A}^{H}\big] \\
&\quad= \sum_{x_\circ \notin \mathcal{L}} \Pr\big[x{=}x_\circ \,\wedge\, V_{FS}^{H}(x,\pi) : (x,\pi) \leftarrow \mathcal{A}^{H}\big] \\
&\quad\le (2q+1)^2 \cdot \left(\sum_{x_\circ \notin \mathcal{L}} \Pr\big[x{=}x_\circ \wedge v = \mathit{accept} : (x,v) \leftarrow \langle \mathcal{S}^{\mathcal{A}}, \mathcal{V}\rangle\big]\right) \\
&\quad= (2q+1)^2 \cdot \left(\Pr\big[x \notin \mathcal{L} \wedge v = \mathit{accept} : (x,v) \leftarrow \langle \mathcal{S}^{\mathcal{A}}, \mathcal{V}\rangle\big]\right) \\
&\quad\le (2q+1)^2 \cdot \mu(\eta),
\end{aligned}$$

where the last inequality holds for some negligible function $\mu(\eta)$ if Σ is sound against an adaptive adversary. The latter is ensured by the assumed soundness against a static adversary and Lemma 2.6. This bound can obviously be written as $q^2\mu'(\eta)$ for another negligible function $\mu'(\eta)$, showing the claimed soundness of $\mathsf{FS}[\Sigma]$. □

3.3.3 Preservation as a proof of knowledge

Again, the following is now an immediate application of Theorem 3.7 and Lemma 2.8.

Corollary 3.9. *Let Σ be a Σ-protocol with superpolynomially sized $\mathcal{C}$. If Σ is a computational/statistical proof of knowledge for static adversaries then* $\mathsf{FS}[\Sigma]$ *is a computational/statistical proof of knowledge for adaptive adversaries.*

Proof. First, we observe that by Lemma 2.8, we may assume Σ to be a computational/statistical proof of knowledge for *adaptive* adversaries. Let $\mathcal{K}$ be the black-box knowledge extractor. Let $\mathcal{A}$ be an (quantum polynomial-time/unbounded) adaptive FS-adversary $\mathcal{A}$. We define a black-box knowledge extractor $\mathcal{E}$ for $\mathsf{FS}[\Sigma]$ as follows. $\mathcal{E}^{\mathcal{A}}$ simply works by running $\mathcal{K}^{\mathcal{S}^{\mathcal{A}}}$, where $\mathcal{S}^{\mathcal{A}}$ is the adaptive Σ-protocol adversary obtained by invoking Theorem 3.7. For any subset $X \subseteq \mathcal{X}$, invoking the proof-of-knowledge property of Σ and using Theorem 3.7, we see that

$$\begin{aligned}
&\Pr\big[x \in X \wedge (x,w) \in R : (x,w) \leftarrow \mathcal{E}^{\mathcal{A}}\big] \\
&\quad = \Pr\big[x \in X \wedge (x,w) \in R : (x,w) \leftarrow \mathcal{K}^{\mathcal{S}^{\mathcal{A}}}\big] \\
&\quad = \frac{1}{p(\eta)} \cdot \Pr\big[x \in X \wedge v = \mathit{accept}(x,v) \leftarrow \langle \mathcal{S}^{\mathcal{A}}, \mathcal{V}\rangle\big]^d - \kappa(\eta) \\
&\quad = \frac{1}{p(\eta)} \cdot \Bigg(\sum_{x_\circ \in X} \Pr\big[x = x_\circ \wedge v = \mathit{accept}(x,v) \leftarrow \langle \mathcal{S}^{\mathcal{A}}, \mathcal{V}\rangle\big] \Bigg)^d - \kappa(\eta) \\
&\quad \geq \frac{1}{p(\eta)} \Bigg(\frac{1}{(2q+1)^2} \sum_{x_\circ \in X} \Pr_H\big[x = x_\circ \wedge V_{FS}^H(x,\pi) : (x,\pi) \leftarrow \mathcal{A}^H\big] \Bigg)^d - \kappa(\eta) \\
&\quad \geq \frac{1}{p(\eta) \cdot (2q+1)^2} \cdot \Pr_H\left[x \in X \wedge V_{FS}^H(x,\pi) : (x,\pi) \leftarrow \mathcal{A}^H\right]^d - \kappa(\eta)
\end{aligned}$$

for some negligible function $\kappa(\eta)$. □

Remark 3.10. We point out that in [Unr17] Unruh considers a stronger notion of extractability than our Definition 2.7, where it is required that, in some sense, the extractor also recovers any local (possibly quantum) output of the adversary $\mathcal{A}$. In the light of Remark 3.6, we expect that our result also applies to this stronger notion of extractability.

Section 3.4

Multi-input reprogrammability

In this section, we extend our (improved) results on adaptively reprogramming the quantum random oracle at *one* point $x \in \mathcal{X}$ to *multiple* points $x_1, \ldots, x_n \in \mathcal{X}$. This in turn will allow us to extend the results on the security of the Fiat-Shamir transformation to *multi-round* protocols. We point out again that the improvement of Lemma 3.3 over Lemma 3.1 in Section 3.2.2 plays a crucial role here, in that it circumvents the trouble with the negligible error term that occurs when trying to extend the single-input result to the setting considered here.

The starting point is the following generalized version of the problem considered in Section 3.2.2. We assume an oracle quantum algorithm $\mathcal{A}^H$ that makes q queries to a random oracle $H : \mathcal{X} \to \mathcal{Y}$ and then produces an output of the form $(x_1, \ldots, x_n, z)$, where z may be quantum, such that a certain (quantum) predicate $V(x_1, H(x_1), \ldots, x_n, H(x_n), z)$ is satisfied with some probability. The goal then is to turn such an $\mathcal{A}^H$ into a multi-stage quantum algorithm $\mathcal{S}$ (the *simulator*) that, stage by stage, outputs the x_i's and takes corresponding Θ_i's as input, and eventually outputs a (possibly quantum) z with the property that $V(x_1, \Theta_1, \ldots, x_n, \Theta_n, z)$ is satisfied with similar probability.

3.4.1 Notation

Up to some modifications, we follow closely the notation introduced in Section 3.2.1. We consider a (purified) oracle quantum algorithm $\mathcal{A}$ that makes q queries to an *oracle*, i.e., an unspecified function $H : \mathcal{X} \to \mathcal{Y}$ with finite non-empty sets $\mathcal{X}, \mathcal{Y}$. Formally, $\mathcal{A}$ is described by a sequence of unitaries $A_1, \ldots, A_q$ and an initial state $|\phi_0\rangle$.[26] For technical reasons that will become clear later, we actually allow (some of) the A_i's to be a *projection* followed by a unitary (or vice versa). One can think of such a projection as a measurement performed by the algorithm, with the algorithm aborting except in case of a particular measurement outcome.

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

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

where $\mathcal{O}^H$ is the unitary defined by $\mathcal{O}^H : |c\rangle|x\rangle|y\rangle \mapsto |c\rangle|x\rangle|y \oplus c{\cdot}H(x)\rangle$ for any triple $c \in \{0,1\}$, $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, with $\mathcal{O}^H$ acting on appropriate registers. We emphasize that we allow *controlled* queries to H. Per se, this gives the algorithm more power, and thus will make our result only stronger, but it is easy to see that such controlled queries to the standard quantum oracle for a function can always be simulated by means of ordinary queries.[27] The final state $\mathcal{A}^H|\phi_0\rangle$ is considered to be a state over registers $\mathsf{X} = \mathsf{X}_1 \ldots \mathsf{X}_n$, Z and E.

Recall from Section 3.2.1 the following notation. For $0 \le i, j \le q$ we set

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

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

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

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

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

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

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

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

for an execution of $\mathcal{A}$ where the oracle is reprogrammed at a given point x after the i-th query. We stress that $(\mathcal{A}^{H*\Theta x}_{i\to q})(\mathcal{A}^H_{0\to i})$ can again be considered to be an oracle quantum algorithm $\mathcal{B}$, which depends on $\Theta \in \mathcal{Y}$, that makes q queries to (the unprogrammed) function H. Indeed, the (controlled) queries to the reprogrammed oracle $H * \Theta x$ can be simulated by means of controlled queries to H (using one additional "work qubit").[28] Exploiting that, in addition to unitaries, we allow projections as elementary operations, we can also understand $(\mathcal{A}^{H*\Theta x}_{i\to q})X(\mathcal{A}^H_{0\to i})$ to be an oracle quantum algorithm again that makes oracle queries to H, where X is the projection $X = |x\rangle\langle x|$, acting on the corresponding query register to the oracle.

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

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

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

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

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

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

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

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

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

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

3.4.2 The general case

Naively, one might hope for an $\mathcal{S}$ that outputs x_1 in the first stage (obtained by measuring one of the queries of $\mathcal{A}^H$), and then on input Θ_1 proceeds by outputting x_2 in the second stage (obtained by measuring one of the subsequent queries of $\mathcal{A}^H$), etc. However, since $\mathcal{A}^H$ may query the hashes of $x_1, \ldots, x_n$ in an arbitrary order, we cannot hope for this to work. Therefore, we have to allow $\mathcal{S}$ to produce $x_1, \ldots, x_n$ in an arbitrary order as well.[29] Formally, we consider $\mathcal{S}$ with the following syntactic behavior: in the first stage it outputs a permutation π together with $x_{\pi(1)}$ and takes as input $\Theta_{\pi(1)}$, and then for every subsequent stage $1 < i \leq n$ it outputs $x_{\pi(i)}$ and takes as input $\Theta_{\pi(i)}$; eventually, in the final stage (labeled by $n + 1$) it outputs z. In line with earlier notation, but taking this additional complication into account, we denote such an execution of $\mathcal{S}$ as $(\pi, \pi(\mathbf{x}), z) \leftarrow \langle\mathcal{S}^{\mathcal{A}}, \pi(\mathbf{\Theta})\rangle$.

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

For us to be able to mathematically reason about the simulator described above, we introduce some additional notation. For the basic simulator from Lemma 3.3 we write, using $r_1 = (b_1, i_1)$, as

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

[29] Looking ahead, in Section 3.4.3 we will force $\mathcal{A}^H$ to query, and thus $\mathcal{S}$ to extract, $x_1, \ldots, x_n$ in the *right* order by requiring x_2 to contain $H(x_1)$ as a substring, x_3 to contain $H(x_2)$ as a substring, etc. This will be important for the the multi-round Fiat-Shamir application.

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

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

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

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

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

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

At the core of our multi-round result will be the following technical lemma, which generalizes Lemma 3.3.

Lemma 3.11. *Let $\mathcal{A}$ be a q-query oracle quantum algorithm. Then, for any function $H : \mathcal{X} \to \mathcal{Y}$, any $\mathbf{x} \in \mathcal{X}^n$ and $\mathbf{\Theta}^n \in \mathcal{Y}^n$, and any projection $\varPi_{\mathbf{x},\mathbf{\Theta}}$, it holds that*

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

Proof. The proof is by induction on n, where the base case is given by Lemma 3.3.

For the induction step we first apply the base case, substituting x_n for x_1, Θ_n for Θ_1, r_n for r_1, $H*\overline{\mathbf{\Theta}\mathbf{x}}$ for H, and $\hat{\varPi}_{x_n,\Theta_n}$ for $\varPi_{x_1,\Theta_1}$, where

$$\hat{\varPi}_{x_n,\Theta_n} = |x_1\rangle\langle x_1| \otimes \ldots \otimes |x_{n\text{-}1}\rangle\langle x_{n\text{-}1}| \otimes \varPi_{\mathbf{x},\mathbf{\Theta}}$$

to obtain

$$\frac{\big\|\big(|x_n\rangle\langle x_n| \otimes \hat{\varPi}_{x_n,\Theta_n}\big)\mathcal{A}^{(H*\overline{\mathbf{\Theta}}\overline{\mathbf{x}})*\Theta_n x_n}|\phi_0\rangle\big\|_2^2}{(2q+1)^2}$$
$$\le \mathop{\mathbb{E}}_{r_n}\Big[\big\|\big(|x_n\rangle\langle x_n|_A \otimes \hat{\varPi}_{x_n,\Theta_n}\big)\mathcal{S}^{H*\overline{\mathbf{\Theta}}\overline{\mathbf{x}}}_{r_n}(\mathcal{A})|\phi_0\rangle\big\|_2^2\Big]$$

which we can write as

$$\frac{\big\|\big(|\mathbf{x}\rangle\langle\mathbf{x}| \otimes \varPi_{\mathbf{x},\mathbf{\Theta}}\big)\mathcal{A}^{H*\mathbf{\Theta}\mathbf{x}}|\phi_0\rangle\big\|_2^2}{(2q+1)^{2n}} \le \frac{\mathbb{E}_{r_n}\Big[\big\|\big(|\mathbf{x}\rangle\langle\mathbf{x}| \otimes \varPi_{\mathbf{x},\mathbf{\Theta}}\big)\mathcal{S}^{H*\overline{\mathbf{\Theta}}\overline{\mathbf{x}}}_{r_n}(\mathcal{A})|\phi_0\rangle\big\|_2^2\Big]}{(2q+1)^{2(n\text{-}1)}} \tag{6}$$

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

Now fix r_n. We define

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

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

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

Since this inequality holds for any fixed r_n, it also holds in expectation over r_n. Substituting it in Equation 6, we retrieve the statement of the lemma. □

Remark 3.12. In case of $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{X}^n$ *without duplicate entries*, it follows from the resulting mutual orthogonality of the projections X_j and the definition of $\mathcal{S}_{\mathbf{r}}^{H}(\mathcal{A})$ that the following holds. The term in the expectation $\mathbb{E}_{\mathbf{r}}$ in the inequality of Lemma 3.11 vanishes for any $\mathbf{r} = (\mathbf{i}, \mathbf{b})$ for which there exist two distinct coordinates $j \neq k$ with $i_j = i_k$. As such, we may well understand this expectation to be over $\mathbf{r} = (\mathbf{i}, \mathbf{b})$ for which $i_j \neq i_k$ whenever $j \neq k$; this only increases the expectation.[30] In other words, we may assume that random *distinct* queries are measured in order to extract $x_1, \ldots, x_n$.

Theorem 3.13 (Measure-and-reprogram, multiple inputs). *Let n be a positive integer, and let $\mathcal{X}, \mathcal{Y}$ be finite non-empty sets. There exists a black-box polynomial-time $(n{+}1)$-stage quantum algorithm $\mathcal{S}$ with the syntax as outlined at the start of this section, satisfying the following property. Let $\mathcal{A}$ be an arbitrary oracle quantum algorithm that makes q queries to a uniformly random $H : \mathcal{X} \to \mathcal{Y}$ and that outputs a tuple $\mathbf{x} \in \mathcal{X}^n$ and a (possibly quantum) output z. Then, for any $\mathbf{x}^\circ \in X^n$* without duplicate entries *and for any predicate V:*

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

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

Proof. We consider the inequality of Lemma 3.11 with the expectation over $\mathbf{r}$ understood as in Remark 3.12. Additionally taking the expectation over H and $\mathbf{\Theta}$ on both sides, we obtain

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

and note that this is equivalent to

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

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

Next, we note that the value of the right hand side does not change [Zha12] when instead of giving $\mathcal{S}$ oracle access to H, we let it choose a random instance from a family of $2q$-wise independent hash functions to simulate $\mathcal{A}$ on. The choice of $\mathbf{r}$ uniquely determines the permutation π with the property $i_{\pi(1)} < \cdots < i_{\pi(n)}$; by definition of $\mathcal{S}^{H,\mathcal{A}}_{\mathbf{\Theta},\mathbf{x},\mathbf{r}}$, the values $\mathbf{x} = (x_1, \ldots, x_n)$ are then extracted from the adversary's queries in the order $\pi(\mathbf{x}) = (x_{\pi(1)}, \ldots, x_{\pi(n)})$. Since $\mathcal{S}$ chooses this $\mathbf{r}$ itself, we can assume that it includes π in its output. Likewise, the simulator takes as input to every stage — from the second to the $(n+1)$-st — a fresh random value, in the order given by $\pi(\mathbf{\Theta})$. However, by definition of $\mathit{\Pi}_{\mathbf{x},\mathbf{\Theta}}$ the final output of the simulator satisfies the predicate V with respect to the given order (without π), i.e. such that $V(\mathbf{x}, \mathbf{\Theta}, z) = 1$, as is the claim of the theorem. □

3.4.3 The time-ordered case

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

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

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

for $2 \le i \le n$.

Theorem 3.14 (Measure-and-reprogram, enforced extraction order). *Let n be a positive integer, and let $\mathcal{X}_0, \mathcal{X}$ and $\mathcal{Y}$ be finite non-empty sets. There exists a black-box polynomial-time $(n{+}1)$-stage quantum algorithm $\mathcal{S}$, satisfying the following property. Let $\mathcal{A}$ be an arbitrary oracle quantum algorithm that makes q queries to a uniformly random $H : (\mathcal{X}_0 \cup \mathcal{Y}) \times \mathcal{X} \to \mathcal{Y}$ and that outputs a tuple $\mathbf{x} = (x_0, x_1, \dots, x_n) \in (\mathcal{X}_0 \times \mathcal{X}^n)$ and a (possibly quantum) output z. Then, for any $\mathbf{x}^\circ \in (\mathcal{X}_0 \times \mathcal{X}^n)$ without duplicate entries and for any predicate V:*

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

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

Remark 3.15. The additive error term $n!/|\mathcal{Y}|$ stems from the fact that the extraction in the right order fails if $\mathcal{A}$ succeeds in guessing one (or more) of the hashes in the hash chain. The claimed term can be improved to $(n-1)^2/|\mathcal{Y}| + n!/|\mathcal{Y}|^2$ by doing a more fine-grained analysis, distinguishing between permutations $\pi \neq \mathrm{id}$ that bring 2 elements "out of order" or more. In any case, it can be made arbitrary small by extending the range $\mathcal{Y}$ of H for computing the hash chain.

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

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

Let $\mathbf{v}^\circ = (v_1^\circ, \ldots, v_n^\circ)$ with $v_i^\circ := (h_i^\circ, x_i^\circ)$, where $h_1^\circ = x_0^\circ$ and $h_i^\circ \in \mathcal{Y}$ is arbitrary but fixed for $i \geq 2$. Let $\boldsymbol{\Theta}$ be uniformly random in $\mathcal{Y}^n$. An application of Theorem 3.13 yields a simulator $\hat{\mathcal{S}}$ with

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

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

$$\begin{aligned}&\Pr_{\boldsymbol{\Theta}}\bigl[\mathbf{x}{=}\mathbf{x}^\circ \wedge V'(\mathbf{v}, \boldsymbol{\Theta}, z) : (\pi, \pi(\mathbf{v}), z) \leftarrow \langle \hat{\mathcal{S}}^{\mathcal{A}_+}, \pi(\boldsymbol{\Theta})\rangle\bigr] \\ &\qquad\qquad \geq \frac{1}{(q+n+1)^{2n}} \Pr_H\bigl[\mathbf{x}{=}\mathbf{x}^\circ \wedge V'(\mathbf{v}, H(\mathbf{v}), z) : (\mathbf{v}, z) \leftarrow \mathcal{A}_+^H\bigr] \qquad (7) \\ &\qquad\qquad = \frac{1}{(q+n+1)^{2n}} \Pr_H\bigl[\mathbf{x}{=}\mathbf{x}^\circ \wedge V(\mathbf{x}, \mathbf{h}^{H,\mathbf{x}}, z) : (\mathbf{x}, z) \leftarrow \mathcal{A}^H\bigr] \,.\end{aligned}$$

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

$$\begin{aligned}&\Pr_{\boldsymbol{\Theta}}\bigl[\mathbf{x}{=}\mathbf{x}^\circ \wedge V'(\mathbf{v}, \boldsymbol{\Theta}, z) : (\pi, \pi(\mathbf{v}), z) \leftarrow \langle \hat{\mathcal{S}}^{\mathcal{A}_+}, \pi(\boldsymbol{\Theta})\rangle\bigr] \\ &\quad = \frac{1}{n!}\sum_{\sigma \in S_n} \Pr_{\boldsymbol{\Theta}}\bigl[\mathbf{x}{=}\mathbf{x}^\circ \wedge V'(\mathbf{v}, \boldsymbol{\Theta}, z) : (\pi, \pi(\mathbf{v}), z) \leftarrow \langle \hat{\mathcal{S}}^{\mathcal{A}_+}, \pi(\boldsymbol{\Theta})\rangle \big| \pi = \sigma\bigr] \,.\end{aligned} \qquad (8)$$

By definition, the predicate $V'(\mathbf{v}, \boldsymbol{\Theta}, z)$ (with $\mathbf{v}$ of the form as explained above) is false whenever there exists an $i \geq 2$ such that $h_i \neq \Theta_{i-1}$. Now suppose that $\pi \neq \mathrm{id}$, then there must be some j such that $\pi(j) < \pi(j-1)$. This implies that the first $\pi(j)$ stages of $\hat{\mathcal{S}}^{\mathcal{A}_+}$ which together (in the $\pi(j)$-th stage) produce $v_j = (h_j, x_j)$ are independent of Θ_{j-1}, since Θ_{j-1} is given as input only at the *later* stage $\pi(j-1)$. We thus have the following, taking it as understood, here and in the sequel, that the random variables $\pi, \mathbf{v}, \boldsymbol{\Theta}$ and z are as in (8).

$$\begin{aligned}\Pr\bigl[\mathbf{x}{=}\mathbf{x}^\circ \wedge V'(\mathbf{v}, \boldsymbol{\Theta}, z) \big| \pi \neq \mathrm{id}\bigr] &\leq \Pr\bigl[\mathbf{x}{=}\mathbf{x}^\circ \wedge h_j = \Theta_{j-1} | \pi \neq \mathrm{id}\bigr] \\ &= \frac{\Pr\bigl[\mathbf{x}{=}\mathbf{x}^\circ | \pi \neq \mathrm{id}\bigr]}{|\mathcal{Y}|} \,.\end{aligned}$$

Using Equation (8), we can bound

$$\frac{1}{n!}\sum_{\sigma \in S_n} \Pr\bigl[\mathbf{x}{=}\mathbf{x}^\circ \wedge V'(\mathbf{v}, \boldsymbol{\Theta}, z) \big| \pi{=}\sigma\bigr]$$

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

We note that by definition of V',

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

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

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

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

Section 3.5

The multi-round Fiat-Shamir transformation

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

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

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

Definition 3.16 (Public coin interactive proof system (PCIP)). *A $(2n{+}1)$-round public coin interactive proof system (PCIP) $\Pi = (\mathcal{P}, \mathcal{V})$ for a language $\mathcal{L}$ is a $(2n{+}1)$-round two-party interactive protocol of the form depicted in Figure 3.2.*

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

Fig. 3.2. The structure of a PCIP. Here $\mathcal{C}$ is a finite non-empty set and V is a predicate.

Remark 3.17. If the language $\mathcal{L}$ is definied by means of an (efficiently verifiable) witness relation $R \subseteq \mathcal{X} \times \mathcal{W}$, then the prover typcially gets a witness w for x as an additional input. We then also say that Π is a PCIP *for the relation R*. In case of a $(2n{+}1)$-round PCIP Π for a witness relation R that is *hard on average*, meaning that there exists an instance generator Gen with the property that for $(w, x) \leftarrow \mathsf{Gen}$ it holds that $(w, x) \in R$, but given x alone it is computationally hard to find w with $(w, x) \in R$, Π is also called an *identification scheme.*

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

Definition 3.18 (Fiat-Shamir transformation for general PCIP (mFS)).
Given an $(2n{+}1)$-round PCIP $\Pi = (\mathcal{P}, \mathcal{V})$ for a language $\mathcal{L}$ and a hash function H with appropriate domain, and range equal to $\mathcal{C}$, we define the non-interactive proof system $\mathsf{FS}[\Pi] = (\mathcal{P}^H_{FS}, \mathcal{V}^H_{FS})$ as follows. The prover $\mathcal{P}$ outputs

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

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

$$c_1 = H(0, x, a_1) \textit{ and}$$

$$c_i = H(i-1, c_{i-1}, a_i) \text{ for } i = 2, ..., n\,,$$

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

Remark 3.19. The challenge number i (minus 1) is included in the hash input to ensure that the challenges are generated using distinct inputs to H with probability 1. This is to enable us to apply Theorem 3.14, which only holds for duplicate-free lists of hash inputs. In fact, any additional strings can be included in the argument when computing c_i using H, without influencing the security properties of the non-interactive proof system in a detrimental way. In the literature one sometimes sees that the entire previous transcript is hashed (in which case the counter number i may then be omitted).

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

When constructing a reduction for mFS, this reduction is participating as a prover in the underlying PCIP, and is hence only provided with random challenges one at a time. We thus need the special simulator from Theorem 3.14, which always outputs the corresponding messages in the right order. The success of this simulator is based on the very essence of the Fiat-Shamir transformation, namely the fact that the intractability of the hash function takes the role of the interaction in enforcing a time order in the transcript of the PCIP.

The security of the multi-round Fiat-Shamir transformation follows as a simple Corollary of Theorem 3.14.

Corollary 3.20. *There exists a black-box quantum polynomial-time* $(n+1)$*-stage quantum algorithm* $\mathcal{S}$ *such that for any adaptive adversary* $\mathcal{A}$ *against the multi-round Fiat-Shamir transformed version* $\mathsf{FS}[\Pi]$ *of a* $(2n{+}1)$*-round PCIP* Π*, making* q *queries to a uniformly random function* H *with appropriate domain and range equal* $\mathcal{C}$*, and for any* $x^\circ \in \mathcal{X}$*:*

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

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

Proof. We may simply set $\mathbf{x}^\circ = (x^\circ, (0, a_1), \ldots, (n-1, a_n))$ for arbitrary $a_1, \ldots, a_n$, apply Theorem 3.14 and then sum over all choices of $a_1, \ldots, a_n$ to obtain the claimed inequality. Note that the round indices ensure that every such $\mathbf{x}^\circ$ is duplicate free, satisfying the corresponding requirement of Theorem 3.14.

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

Remark 3.21. There exist PCIPs with soundness error much smaller than $1/|\mathcal{C}|$. As an example, consider the sequential repetition of a Σ-protocol with special soundness. Here, the soundness error is $1/|\mathcal{C}|^n$. In this case, the term proportional to $1/|\mathcal{C}|$ renders the bound from the above theorem trivial. Note however, that (i) this situation is extremely artificial, as there is absolutely no reason to repeat sequentially instead of in parallel, and (ii) the additive error term can be made arbitrarily small by considering a variant Π' of Π where the random challenges are enlarged with a certain number of bits that are ignored otherwise, see Remark 3.15.

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

As for the case of the Fiat-Shamir transformation for Σ-protocols, the general reduction implies that security properties that protect against dishonest provers carry over from the interactive to the non-interactive proof system. For a definition of the properties considered in the following theorem see Section 3.3.3. The quantum proof-of-knowledge-property was introduced in [Unr12].

Corollary 3.22 (Preservation of Soundness/PoK). *Let Π be a constant-round PCIP that has (statistical/computational) soundness, and/or the (statistical/computational) quantum proof-of-knowledge-property, respectively. Then, in the QROM, $\mathsf{FS}[\Pi]$ has (statistical/computational) soundness, and/or the (statistical/computational) quantum proof-of-knowledge-property, too.*

Proof. Corollary 3.20 turns any dishonest prover $\mathcal{A}_{\mathsf{FS}[\Pi]}$ for $\mathsf{FS}[\Pi]$ with success probability ϵ into a dishonest prover $\mathcal{A}_\Pi$ for Π, with success probability $\epsilon \cdot (2q+1)^{-2n}$, where $2n+1$ is the number of rounds in Π. Since n is constant and q is polynomial in the security parameter, the success probabilities of the respective provers are polynomially related. The claimed implications follow now using the same arguments as in Corollaries 3.8 and 3.9. □

Section 3.6

Extractable PCIP's from quantum computationally unique responses

In Section 3.7.1, we will see that the proof-of-knowledge property of the underlying Σ-protocol is crucial for a Fiat-Shamir signature scheme to be unforgeable. In [Unr12], Unruh proved that special soundness (a witness can be constructed efficiently from two different accepting transcripts) and perfect unique responses are sufficient conditions for a Σ-protocol to achieve this property in the context of quantum adversaries. The perfect-unique-responses property is used to show that the final measurement of the Σ-protocol adversary that produces the response is nondestructive conditioned on acceptance. This property ensures that the extractor can measure the response, and then rewind "as if nothing had happened".

A natural question is therefore which other property except the arguably quite strict condition of perfect unique responses is sufficient to imply extractability together with special soundness. In [ARU14], the authors show that computationally unique responses is insufficient to replace perfect unique responses. A Σ-protocol has computationally unique responses if the verification relation V is collision-resistant from responses to commitment-challenge pairs in the sense that it is computationally hard to find two valid responses for the same commitment-challenge pair.

3.6.1 Generalizing Unruh-rewinding

In [Unr16], Unruh introduced the notion of collapsingness, a quantum generalization of the collision-resistance property for hash functions. The same work also showed how to adapt a transformation from [Unr12] to allow for rewinding of Σ-protocols with the help of collapsingness instead of perfect-unique responses, but this transformation suffers from the same inefficiency as the

Unruh-transform. Here, we take a more direct approach of generalizing collapsingness to the setting of Σ-protocols.

Definition 3.23 (generalized from [Unr16]). *Let $R : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ be a relation with $|X|$ and $|Y|$ superpolynomial in the security parameter η, and define the following two games for polynomial-time two-stage adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$,*

$$\begin{aligned}&\text{Game 1}: \\ &(S, X, Y) \leftarrow \mathcal{A}_1,\ r \leftarrow R(X, Y),\ X \leftarrow \mathcal{M}(X),\ Y \leftarrow \mathcal{M}(Y),\ b \leftarrow \mathcal{A}_2(S, X, Y) \\ &\text{Game 2}: \\ &(S, X, Y) \leftarrow \mathcal{A}_1,\ r \leftarrow R(X, Y),\ \qquad\qquad\qquad Y \leftarrow \mathcal{M}(Y),\ b \leftarrow \mathcal{A}_2(S, X, Y).\end{aligned}$$

Here, X and Y are registers of dimension $|X|$ and $|Y|$, respectively, $\mathcal{M}$ denotes a measurement in the computational basis, and applying R to quantum registers is done by computing the relation coherently and measuring it. R is called collapsing from $\mathcal{X}$ to $\mathcal{Y}$*, if an adversary cannot distinguish the two experiments if the relation holds, i.e. if for all adversaries $\mathcal{A}$ it holds that*

$$\left| \Pr_{\mathcal{A},\ \text{Game 1}}[r = b = 1] - \Pr_{\mathcal{A},\ \text{Game 2}}[r = b = 1] \right| \leq \text{negl}(\eta). \tag{9}$$

Note that this definition is equivalent to Definition 23 in [Unr16] for functions, i.e. if $R(x, y) = 1$ if and only if $f(x) = y$ for some function f.

Via the relation that is computed by the second stage of the verifier, the collapsingness property can be naturally defined for Σ-protocols.

Definition 3.24 (Quantum computationally unique responses). *A Σ-protocol has* quantum computationally unique responses*, if the verification predicate $V(x, \cdot, \cdot, \cdot) : \mathcal{Y} \times \mathcal{C} \times \mathcal{Z} \to \{0,1\}$ seen as a relation between $\mathcal{Y} \times \mathcal{C}$ and $\mathcal{Z}$ is collapsing from $\mathcal{Z}$ to $\mathcal{Y} \times \mathcal{C}$, where $\mathcal{Y}$, $\mathcal{C}$ and $\mathcal{Z}$ are the commitment, challenge and response spaces of the protocol, respectively.*

Intuitively, for fixed commitment-challenge pairs, no adversary should be able to determine whether a superposition over successful responses z has been measured or not. As in the case of hash functions (where collapsingness is a natural stronger quantum requirement than collision-resistance), quantum computationally unique responses is a natural stronger quantum requirement than computationally unique responses. In Section 3.6.3 we define (Definition 3.29) a generalized notion of quantum computationally unique responses for general PCIP's.

The following is a generalization of Theorem 9 in [Unr12] where the assumption of perfect unique responses is replaced by the above quantum computational version. Additionally, we relax the special soundness requirement to *t-soundness*, which requires that for any first message a, for uniformly random chosen challenges $c_1, \dots, c_t$, and for any responses $z_1, \dots, z_t$ with $V(x, a_i, c_i, z_i)$ for all $i \in \{1, \dots, t\}$, a witness w for x can be efficiently computed except with negligible probability (over the choices of the c_i). See Definition 4.22 and the text below it for a formal definition of t-soundness.

Theorem 3.25 (Generalization of Theorem 9 from [Unr12]). *Let Π be a Σ-protocol with t-soundness for some constant t and with quantum computationally unique responses. Then Π is a computational proof of knowledge as in Definition 2.7.*

The proof follows very much the proof of Theorem 9 in [Unr12], up to some small extensions; thus, we only give a proof sketch here.

Proof (sketch). We consider the following extractor $\mathcal{K}$. It runs $\mathcal{A}$ to the point where it outputs a. Then, it chooses a random challenge c_1 and sends it to $\mathcal{A}$, and obtains a response z_1 by measuring $\mathcal{A}$'s corresponding register. $\mathcal{K}$ then rewinds $\mathcal{A}$ (on the measured state!) and chooses and sends to $\mathcal{A}$ a fresh random challenge c_2, resulting in a response z_2, etc., up to obtaining response z_t. If $V(x, a_i, c_i, z_i)$ for all $i \in \{1, \dots, t\}$ then $\mathcal{K}$ can compute w except with negligible probability by the t-soundness property; otherwise, it aborts.

It remains to analyze the probability, denoted by F below, that $V(x, a_i, c_i, z_i)$ for all i. If the Σ-protocol has *perfect* unique responses then measuring the response z is *equivalent* to measuring whether the response satisfies the verification predicate V (with respect to x, a, c). Lemma 3.26 in Section 3.6.2, which generalizes Lemma 7 in [Unr12], allows us then to control the probability F by means of the probability V that $\mathcal{A}$ succeeds in convincing the verifier in an ordinary run (this holds for an arbitrary but fixed a, and on average over a by means of Jensen's inequality).[31] If the Σ-protocol has *quantum computationally* unique responses instead, then measuring the response z is *computationally indistinguishable* from measuring whether the response satisfies the verification predicate, and so there can only be a negligible loss in the success probability of $\mathcal{K}$ compared to above. □

We expect the above theorem to be very useful in practice, for the following reason. Usually, Σ-protocols deployed in Fiat-Shamir signature schemes have

[31] For the case of 2-special soundness, the originial Lemma 7 in [Unr12] suffices.

computationally unique responses to ensure strong unforgeability via Theorem 3.34 or similar reductions. On the other hand, only very artificial separations between the notions of collision resistance and collapsingness for hash functions are known (e.g. the one presented in [Zha19b]). It is therefore plausible that many Σ-protocols deployed in strongly unforgeable Fiat-Shamir signature schemes have quantum computationally unique responses as well. In Section 3.7 we take a look at a couple of examples that form the basis of signature schemes that featured in the NIST competition for the standardization of post-quantum cryptographic schemes.

3.6.2 Quantum extractability of t-special sound Σ-protocols

In order to do quantum rewinding in the case of t-special sound Σ-protocols (as opposed to 2-special sound protocols), we need to generalize Lemma 7 from [Unr12]. The generalized lemma relates the success probability of applying a random projection to a state vector with the success probability of sequentially applying t random projections, where "success probability" here is in terms of the (average) square-norm of the projected state vector. This statement gives us the means to relate the probability of a interactive prover making the verifier accept with the probability of an extractor making the verifier accept t times, when rewinding $t-1$ times and using a freshly random (and independent) challenge each time.

Lemma 3.26. *Let $P_1, \dots, P_n$ be projections and $|\psi\rangle$ a state vector, and set*

$$V := \frac{1}{n}\sum_i \langle\psi|P_i|\psi\rangle = \frac{1}{n}\sum_i \|P_i|\psi\rangle\|^2 \quad \textit{and} \quad F := \frac{1}{n^t}\sum_{i_1\cdots i_t} \|P_{i_t}\cdots P_{i_1}|\psi\rangle\|^2 .$$

Then $F \geq V^{2t-1}$.

The case $t = 2$ was proven in [Unr12, Lemma 7]. We show here how to extend the proof to $t = 3$; the general case works along the same lines.

Proof (of the case $t = 3$). For convenience, set $A := \frac{1}{n}\sum_i P_i$ and $|\psi_{ijk}\rangle := P_k P_j P_i|\psi\rangle$. Then, using convexity of the function $x \mapsto x^5$ to argue the first inequality, we get

$$V^5 = (\langle\psi|A|\psi\rangle)^5 = \langle\psi|A^5|\psi\rangle = \frac{1}{n^5}\sum_{ijk\ell m} \langle\psi|P_i P_j P_k P_\ell P_m|\psi\rangle$$

$$= \frac{1}{n^5} \sum_{ijk\ell m} \langle \psi_{ijk} | \psi_{m\ell k} \rangle = \frac{1}{n} \sum_{k} \Bigg(\frac{1}{n^2} \sum_{ij} \langle \psi_{ijk} | \Bigg) \Bigg(\frac{1}{n^2} \sum_{\ell m} | \psi_{m\ell k} \rangle \Bigg)$$

$$= \frac{1}{n} \sum_{k} \Big\| \frac{1}{n^2} \sum_{ij} | \psi_{ijk} \rangle \Big\|^2 \leq \frac{1}{n^3} \sum_{ijk} \| | \psi_{ijk} \rangle \|^2 = F \,,$$

where the last inequality is Claim 2 in the proof of Lemma 7 in [Unr12]. □

The following is a generalization of Lemma 7 from [Unr12] in a different direction. It gives us control over the success probability of the extractor when the challenge consists of two parts, and the extractor works by rewinding once with a freshly chosen challenge pair, and once more where now one part of the challenge is re-used and only the other part is freshly chosen.

Lemma 3.27. *Let P_{ij} ($1 \leq i \leq n$, $1 \leq j \leq m$) be projections $|\psi\rangle$ a state vector, and set*

$$V := \frac{1}{nm} \sum_{i,j} \| P_{i,j} |\psi\rangle \|^2 \qquad \textit{and} \qquad F := \frac{1}{n^2 m^3} \sum_{\substack{i_1,i_2 \\ j_1,j_2,j_3}} \| P_{i_2 j_3} P_{i_2 j_2} P_{i_1 j_1} |\psi\rangle \|^2 \,.$$

Then $F \geq V^6$.

Proof. We set $|\varphi_{i_1 j_1}\rangle := P_{i_1 j_1} |\psi\rangle / \langle \psi | P_{i_1 j_1} | \psi \rangle$. Then

$$\begin{aligned}
F &= \frac{1}{n^2 m^3} \sum_{\substack{i_1,i_2 \\ j_1,j_2,j_3}} \| P_{i_2 j_3} P_{i_2 j_2} P_{i_1 j_1} |\psi\rangle \|^2 \\
&= \frac{1}{n^2 m} \sum_{i_1,i_2,j_1} \frac{1}{m^2} \sum_{j_2,j_3} \| P_{i_2 j_3} P_{i_2 j_2} |\varphi_{i_1 j_1}\rangle \|^2 \, \langle \psi | P_{i_1 j_1} | \psi \rangle \\
&\geq \frac{1}{n^2 m} \sum_{i_1,i_2,j_1} \Bigg(\frac{1}{m} \sum_{j_2} \| P_{i_2 j_2} |\varphi_{i_1 j_1}\rangle \|^2 \Bigg)^3 \langle \psi | P_{i_1 j_1} | \psi \rangle && \text{(Lemma 3.26)} \\
&= \frac{1}{n^2 m} \sum_{i_1,i_2,j_1} \Bigg(\frac{1}{m} \sum_{j_2} \| P_{i_2 j_2} P_{i_1 j_1} |\psi\rangle \|^2 / \langle \psi | P_{i_1 j_1} | \psi \rangle^{2/3} \Bigg)^3 \\
&\geq \Bigg(\frac{1}{n^2 m^2} \sum_{i_1,i_2,j_1,j_2} \| P_{i_2 j_2} P_{i_1 j_1} |\psi\rangle \|^2 / \langle \psi | P_{i_1 j_1} | \psi \rangle^{2/3} \Bigg)^3 && \text{(Jensen's inequality)} \\
&\geq \Bigg(\frac{1}{n^2 m^2} \sum_{i_1,i_2,j_1,j_2} \| P_{i_2 j_2} P_{i_1 j_1} |\psi\rangle \|^2 \Bigg)^3
\end{aligned}$$

$$\geq \left(\frac{1}{nm}\sum_{i_1,j_1}\|P_{i_1j_1}|\psi\rangle\|^2\right)^6 \qquad \text{(Lemma 3.26)}$$

This proves the claim. □

3.6.3 Quantum extractability of PCIP's: the q2 identification scheme case

A class of identification schemes that is of particular interest in the multi-round setting are so-called q2-identification schemes. The signature scheme MQDSS, for example, is obtained from such an identification scheme via the multi-round Fiat-Shamir transformation from Definition 3.38 (with some additional strings included in the hash arguments). In this section, we will prove that a PCIP with a so-called "q2 extractor" [CHR+16, Definition 4.6] is a quantum proof of knowledge if it has an additional collapsingness property. This is necessary for its Fiat-Shamir transformation to fulfill (s)UF-CMA in the QROM (for (s)UF-CMA in the ROM, the q2-extractor alone is sufficient [CHR+16]).

We begin by defining q2 identification schemes and their extractors.

Definition 3.28. *A 5-round identification scheme is a q2 identification scheme, if the second challenge is a single bit. A q2 identification scheme is called q2-extractable if there exists a polynomial-time algorithm that, on input accepting transcripts* $t^{(i)} = (a_1, c_1^{(i)}, a_2^{(i)}, c_2^{(i)}, z^{(i)})$, $i = 1, 2, 3, 4$, *such that*

$$\begin{aligned} &c_1^{(1)} = c_1^{(2)} \neq c_1^{(3)} = c_1^{(4)} \text{ and} \\ &c_2^{(1)} = c_3^{(3)} \neq c_2^{(2)} = c_2^{(4)}, \end{aligned} \tag{10}$$

outputs the secret key with non-negligible probability.

For ease of exposition we have assumed that the different challenges of a single PCIP come all from the same challenge space. A q2 identification scheme can be brought into this form by having the prover compute the second challenge by selecting the first bit of an augmented second challenge that is as large as the first one. For classical provers, four transcripts as required by the above definition can be obtained by straightforward rewinding. In the following, we show that, if the q2 identification scheme has an additional property similar to the quantum-computationally unique responses property introduced at the start of this section (Definition 3.24) and [LZ19a], then the existence of a q2 extractor implies that there exists a quantum extractor. This makes the scheme a quantum proof of knowledge. The argument follows the same lines as the the proof (sketch) of Theorem 3.25 – showing that t-soundness and quantum-computationally unique responses imply the quantum proof-of-knowledge-property – which in turn is an extension of the result by Unruh for Σ-protocols with perfect unique responses [Unr12].

Recall the definition of a collapsing relation, Definition 3.24, a generalization of the notion of a collapsing hash function [Unr16]. We define the notion of collapsingness for interactive proof systems as follows:

Definition 3.29. *A $(2n+1)$-round interactive proof system Π is called collapsing, if the relation $R_\Pi : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ with $\mathcal{X} = \mathcal{C}^n \times \mathcal{A}_1$ and $\mathcal{Y} = \mathcal{A}_2 \times ... \times \mathcal{A}_n \times \mathcal{Z}$ given by the verification predicate V_Π of Π is collapsing from $\mathcal{X}$ to $\mathcal{Y}$.*

Note that for $n = 1$, this notion of collapsingness coincides with the notion of quantum-computationally unique responses from Definition 3.24.

Given a q2-identification scheme Π, consider the following straightforward (first stage of a) quantum extractor $\mathcal{E}_\Pi^{\mathcal{A}}$. The extractor runs the prover $\mathcal{A}$ using honestly sampled challenges to obtain a first transcript $t^{(1)}$. Now it rewinds three times and reruns $\mathcal{A}$, each time with a fresh pair of challenges, chosen such as to obtain $t^{(i)}$, $i = 2, 3, 4$ such that the four transcripts fulfill the conditions (10). For this extractor, we obtain the following

Theorem 3.30. *Let Π a q2-extractable q2-identification scheme that is also collapsing. Then the success probability of the extractor $\mathcal{E}_\Pi^{\mathcal{A}}$ is lower-bounded in terms of the success probability of the prover $\mathcal{A}$ as*

$$\Pr[\mathcal{E}_\Pi^{\mathcal{A}} \text{ extracts}] \geq \left(\Pr\left[v = \mathit{accept} : (x, v) \leftarrow \langle \mathcal{A}, \mathcal{V}_\Pi \rangle\right]\right)^7 \tag{11}$$

The proof of this theorem is essentially the same as for Theorem 3.25, which is a slight modification of an argument from [Unr12].

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

Corollary 3.31. *Let Π a q2-extractable q2-identification scheme that is also collapsing. Then it is a quantum proof of knowledge.*

In particular, the 5-round identification scheme Π_{SSH} from [SSH11] which is used to construct the post-quantum digital signature scheme MQDSS has these properties under plausible assumptions, namely that it is instantiated with the standard hash-based commitment scheme using a collapsing hash function [Unr16] (see discussion towards the end of Section 3.7.2). For MQDSS, this is no additional assumption, as the Fiat-Shamir transformation uses the QROM anyway, and a quantum accessible random oracle is collapsing by [Unr16].

Corollary 3.32. *If the 5-round identification scheme from [SSH11] is instantiated with the standard hash-based commitment scheme using a collapsing hash function, it is a quantum proof of knowledge.*

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

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

Section 3.7

Applications

3.7.1 Unforgeability of Fiat-Shamir signatures

Any Fiat-Shamir non-interactive proof system can easily be transformed into a public-key signature scheme.[32] The signer simply proves knowledge of a witness (the secret key) for a composite statement $x^* := x\|m$, which includes the public key x as well as the message m. The signature σ then consists of a proof for x^*.

The unforgeability (against no-message attacks) of a Fiat-Shamir signature scheme is shown below to follow from the proof-of-knowledge property of the underlying proof system (hence, as we now know, of the underlying Σ-protocol),

[32] In fact, that is how the Fiat-Shamir transform was originally conceived in [FS87]. Only later [BG93] adapted the idea to construct a non-interactive zero-knowledge proof system.

under the assumption that the relation is hard, i.e. it is infeasible to compute sk from pk.

Theorem 3.33. *Let* Σ *be Σ-protocol for some hard relation* R*,* $\mathcal{C}$ *and the proof-of-knowledge property according to Definition 2.7. Then, the Fiat-Shamir signature scheme* $\mathsf{Sig}[\Sigma]$ *fulfills* $\mathsf{EUF-NMA}$ *security.*

Proof. Let $\mathcal{A}$ be an adversary against $\mathsf{EUF-NMA}$, issuing at most q quantum queries to H. We show that

$$\mathrm{Adv}^{\mathsf{EUF-NMA}}_{\mathsf{Sig}[\Sigma]}(\mathcal{A}) := \Pr\left[\mathsf{Verify}^{H}(pk, m, \sigma) : (pk, sk) \leftarrow \mathsf{Gen}, (m, \sigma) \leftarrow \mathcal{A}^{H}(pk)\right]$$

is negligible.

Recall from Definition 2.15 of Fiat-Shamir signatures that the Σ-protocol Σ^* is the Σ-protocol Σ where the prover and verifier ignore the message part m of the instance $x\|m$. A successful forgery (m, σ) is such that $V^{H}_{FS}(x\|m, \sigma)$ accepts the proof σ. Therefore,

$$\mathrm{Adv}^{\mathsf{EUF-NMA}}_{\mathsf{Sig}[\Sigma]}(\mathcal{A}) = \mathop{\mathbb{E}}_{(x,w)\leftarrow G}\left[\Pr_{H}\left[V^{H}_{FS}(x\|m, \sigma) : (m, \sigma) \leftarrow \mathcal{A}^{H}(x)\right]\right]. \tag{12}$$

Note that if Σ is a proof of knowledge, so is Σ^*. Our Corollary 3.9 assures that if Σ^* is a proof of knowledge, then also $\mathsf{FS}[\Sigma^*]$ is a proof of knowledge.

For fixed instance x, let X be the set of instance/message strings $x'\|m$ where $x' = x$. We apply the knowledge extractor from Definition 2.7 to the adaptive FS-attacker $\mathcal{A}^{H}(x)$ that has x hard-wired and outputs it along with a message m and the proof/signature σ: There exists a knowledge extractor $\mathcal{E}$, constants d, e and a polynomial p (all independent of x) such that

$$\begin{aligned}&\Pr_{H}\left[x'\|m \in X \wedge V^{H}_{FS}(x'\|m, \sigma) : (x'\|m, \sigma) \leftarrow \mathcal{A}^{H}(x)\right]\\ &\quad\leq \left(\Pr\left[x'\|m \in X \ \wedge\ (x', w) \in R : (x'\|m, w) \leftarrow \mathcal{E}^{\mathcal{A}}\right] q^{e} p(\eta) + \mu(\eta)\right)^{1/d}\end{aligned} \tag{13}$$

Finally, taking the expected value of (13) over the choice of the instance x according to the hard-instance generator G, we obtain that the left hand side equals $\mathrm{Adv}^{\mathsf{EUF-NMA}}_{\mathsf{Sig}[\Sigma]}(\mathcal{A})$. For the right-hand side, we can use the concavity of $(\cdot)^{1/d}$ (note that we can assume without loss of generality that $d > 1$) and apply Jensen's inequality to obtain

$$\mathop{\mathbb{E}}_{x\leftarrow G}\left[\left(\Pr\left[x'\|m \in X \ \wedge\ (x', w) \in R : (x'\|m, w) \leftarrow \mathcal{E}^{\mathcal{A}}\right] q^{e} p(\eta) + \mu(\eta)\right)^{1/d}\right]$$

$$\leq \left(\mathop{\mathbb{E}}_{x \leftarrow G} \Pr\left[x' \| m \in X \wedge (x', w) \in R : (x' \| m, w) \leftarrow \mathcal{E}^{\mathcal{A}}\right] q^e p(\eta) + \mu(\eta) \right)^{1/d} .$$

Note that the expected probability is the success probability of the extractor to produce a witness w matching the instance x. As long as the relation R is hard according to Definition 2.14, this success probability is negligible, proving our claim.

□

If we wish for unforgeability *under chosen-message attack*, zero-knowledge is required as well. [Unr17] and [KLS18] contain partial results that formalize this intuition, but they were unable to derive the extractability of the non-interactive proof system. Instead, they modify the Σ-protocol to have a *lossy mode* [AFLT12], i.e. a special key-generation procedure that produces key pairs whose public keys are computationally indistinguishable from the real ones, but under which it is impossible for any (even unbounded) quantum adversary to answer correctly.

Our new result above completes these previous analyses, so that we can now state precise conditions under which a Σ-protocol gives rise to a (strongly) unforgeable Fiat-Shamir signature scheme, without the need for lossy keys.

Theorem 3.34. *Let Σ be Σ-protocol for some hard relation R, with superpolynomially sized challenge space $\mathcal{C}$ and the proof-of-knowledge property according to Definition 2.7. Assume further that Σ is ε-perfect sigma-(non-abort) honest-verifier zero-knowledge (naHVZK), has α bits of min entropy and computationally unique responses as defined in [KLS18]. Then,* $\mathsf{Sig}[\Sigma]$ *fulfills* $\mathsf{sEUF-CMA}$ *security.*

Proof. By Theorem 3.3 of [KLS18] and Theorem 2.2 of [BBD+23], we can use the naHVZK, min-entropy and computationally-unique-response properties of Σ to reduce an $\mathsf{sEUF-CMA}$ adversary to an $\mathsf{EUF-NMA}$ adversary[33]. The conclusion then follows immediately from our Theorem 3.33 above. □

Several schemes of the Fiat-Shamir kind have featured (and have been selected) in the NIST post-quantum standardization process. Below we outline how our result might be applied to some of these schemes, and under which additional assumptions. We leave the problems of applying our techniques to the actual (highly optimized) signature schemes and of working out the concrete security bounds for future work.

[33] See also Theorem 25 in [Unr17] for a different proof technique.

Picnic In order to obtain QROM-security, Picnic uses the *Unruh transform* [Unr15b] instead of the Fiat-Shamir transformation, incurring a 1.6x loss in efficiency (according to [CDG+17]) compared to Fish, which is the same scheme under plain Fiat-Shamir.

The underlying sigma-protocol for these schemes is ZKB++ [CDG+17], an optimized version of ZKBoo [GMO16], which uses an arbitrary one-way function ϕ, a commitment scheme COM and a multi-party computation protocol to prove knowledge of a secret key. Roughly, a prover runs the multi-party protocol 'in its head' (i.e. simulates the three agents from the protocol, see [IKOS07b]) to compute $pk := \phi(sk)$. Only a prover who knows the secret key can produce the correct view of all three agents, but the public key suffices to verify the correctness of two of the views. In the first round, the prover uses COM to commit to all three views separately, and sends these commitments to the verifier. The verifier replies with a random challenge $i \in \{1, 2, 3\}$, to which the prover in turn responds by opening the i-th and $i+1$-th commitment.

ZKBoo does not specify a concrete commitment scheme for COM. A natural option is to commit by hashing the input together with some random bits.

Corollary 3.35. Sig[ZKBoo] *is strongly existentially unforgeable in the QROM when* COM *is instantiated with a hash function H.*

Proof. If we treat H as a quantum-accessible random oracle, then H is collapsing by [Unr16]. Since the response of the prover in the third round consists only of openings to the commitments c_i, c_{i+1}, i.e. preimages of c_i and c_{i+1} under H, and since collapsingness is closed under concurrent composition [Feh18], the collapsingness of H implies that ZKBoo has quantum computational unique responses. ZKBoo further has 3-soundness, and thus the claim follows using Theorems 3.25 and 3.34. □

ZKB++ improves on ZKBoo by introducing optimizations specific to the signature context, which complicate the analysis of the overall scheme. We therefore leave the adaption of Corollary 3.35 to ZKB++ and Fish for future work.

We also point out that Picnic2 is not t-sound because a witness can be computed from 3 responses only under certain restrictions on the challenges. However, this can be taken care of by a variation of the t-soundness property, as proven in Lemma 3.27 in Section 3.6.2.

Lattice-based Fiat-Shamir signature schemes – CRYSTALS-Dilithium and qTesla In [Lyu09] and [Lyu12], Lyubashevsky developed a Fiat-Shamir signature scheme based on (ring) lattice assumptions. In the following, we explain the lattice case and mention ring-based lattice terms in parentheses. The underlying sigma protocol, which forms the basis of the NIST submissions CRYSTALS-Dilithium and qTesla, can be roughly described as follows. The instance is given by a key pair $((A,T),S)$, with $T = AS$. Here, A and S are matrices of appropriate dimensions over a finite field (polynomials of appropriate degree), and S is *small*. For the first message to the verifier, the prover selects a random short vector (small polynomial) y, and sends over Ay. The second message, from the verifier to the prover, is a random vector (polynomial) c with entries (coefficients) in $\{-1,0,1\}$ and a small Hamming weight. The third message, i.e. the response of the prover, is $z = Sc + y$, which is short (small) as well. The prover actually sends z only with a particular probability, which is chosen so as to make the distribution of (sent) z independent of S. Otherwise, it aborts and tries again. Verification is done by checking whether z is indeed short (small), and whether $Az - Ay = Tc$. Let us denote this protocol by LatticeΣ. In the following we restrict our attention to the lattice case, but we expect that one can do a similar analysis for the ring-based schemes.

The security of the scheme is, in the lattice case, based on the Short Integer Solution (SIS) problem, which essentially guarantees that it is hard to find an integral solution to a linear system that has a small norm. The computationally unique responses property for the simple Σ-protocol described above, in fact, follows directly from SIS: If one can find a vector c and two short vectors x_i, $i = 1, 2$ such that $Ax_0 = c = Ax_1$, then the difference $x = x_1 - x_0$ is a short solution to the linear system $Ax = 0$.

Another way to formulate the computationally unique responses property for the above Σ-protocol is as follows. Let $S \subset \mathbb{F}_q^n$ be the set of short vectors. Let $f_A : S \to \mathbb{F}_q^m$ be the restriction to S of the linear map given by the matrix $A \in \mathbb{F}_q^{m\times n}$. The Σ-protocol above has computationally unique responses if and only if f_A is collision resistant. As pointed out at the end of Section 3.6, the known examples that separate the collision resistance and collapsingness properties are fairly artificial. Hence it is a natural to assume that f_A is collapsing as well.

Assumption 3.36. *For m, n and q polynomial in the security parameter η, the function family f_A keyed by a uniformly random matrix $A \in \mathbb{F}_q^{m\times n}$ is collapsing.*

Under Assumption 3.36, LatticeΣ has quantum computational unique responses, and hence gives rise to an unforgeable Fiat-Shamir signature scheme.

Corollary 3.37. *Under Assumption 3.36,* $\mathsf{Sig}[\mathsf{Lattice}\Sigma]$ *is strongly existentially unforgeable in the QROM.*

As mentioned at the end of the introduction, in their concurrent and independent work [LZ19a], Lie and Zhandry show that f_A satisfies their notion of *weak*-collapsingness (assuming hardness of LWE), which roughly says that there is some non-negligible probability that the adversary *does not* notice a measurement. Weak-collapsingness implies a similarly weakened variant of our property 'quantum computational responses', which is still sufficient to let the proof of Theorem 3.25 go through, albeit with a worse but still non-negligible success probability for the knowledge-extractor.

3.7.2 Signature schemes from multi-round Fiat-Shamir

In the context of Fiat-Shamir signature schemes, multi-round variants have also been used. One example is MQDSS [CHR+16], a digital signature scheme that made it to the second round of the NIST standardization process for post-quantum cryptographic schemes. This digital signature scheme is constructed by applying the multi-round Fiat-Shamir transformation to the 5-round identification scheme by Sakumoto, Shirai, and Hiwatari [SSH11] based on the hardness of solving systems of multivariate quadratic equations.

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

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

Definition 3.38 (Fiat-Shamir signatures from a general PCIP). *Given an* $(2n+1)$*-round public coin identification scheme* $\mathsf{\Pi} = (\mathsf{Gen}, \mathcal{P}, \mathcal{V})$ *for a witness relation* R *and a hash function* H *with appropriate domain and range equal to* $\mathcal{C}$*, we define the digital signature scheme* $\mathsf{Sig}[\mathsf{\Pi}] = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ *as follows. The key generation algorithm* Gen *is just the one from* Π*. The signing algorithm* Sign*, on input a secret key* sk *and a message* m*, outputs*

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

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

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

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

We note that the above definition is equivalent to the following, alternative formulation: Let $\mathsf{Sign}_{sk}(m)$ *produce σ by running $P^H_{FS}(x||m)$, and let* $\mathsf{Verify}(m, \sigma)$ *be equal to the outcome of $V^H_{FS}(x||m)$, where $(P^H_{FS}, V^H_{FS}) = \mathsf{FS}[\Pi^*]$ and $\Pi^* = (\mathcal{P}^*, \mathcal{V}^*)$ is the identification scheme obtained from Π by setting $\mathcal{P}^*(x||m) = \mathcal{P}(x)$ and $\mathcal{V}^*(x||m) = \mathcal{V}(x)$ for any m. This alternative formulation will be convenient in the proof of Theorem 3.41.*

Remark 3.39. As in the case of the plain multi-round Fiat-Shamir transformation, one can include arbitrary additional strings in the argument when computing the challenges c_i. Examples where this is done include the MQDSS signature scheme [CHR+16], where the message m and the first commitment a_1 are also included in the argument for computing the second challenge, and Bulletproofs, where the challenges are computed by hashing the entire transcript up to that point [BBB+18].

As an identification scheme is an interactive honest-verifier zero knowledge proof of knowledge of a secret key, the above signature scheme is a a non-interactive zero knowledge proof of knowledge of a secret key according to Corollary 3.20. For a digital signature scheme, however, the stronger security notion of (strong) unforgeability against chosen message ((s)UF-CMA) attacks is required.

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

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

Definition 3.40 (Computationally unique responses - PCIP). *A* $(2n+1)$*-round PCIP* $\Pi = (\mathcal{P}, \mathcal{V})$ *is said to have* computationally unique responses *if given a partial transcript* $(x, a_1, c_1, \dots a_i, c_i)$ *it is computationally hard to find two accepting conversations that both extend the partial transcript but differ in (at least)* a_{i+1} *(here we consider* z *to be equal to* a_{n+1}*), i.e. for* $con_i = x, a_1, c_1, \dots a_i, c_i, a_{i+1}^{(j)}, c_{i+1}^{(j)} \dots, a_n^{(j)}, c_n^{(j)}, z^{(j)}$, $j = 1, 2$ *we have that*

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

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

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

Theorem 3.41 ((s)UF-CMA of multi-round FS signatures). *Let* Π *be a PCIP for some hard relation R, which is a quantum proof of knowledge and satisfies completeness, HVZK, and has unpredictable commitments*[34] *as well as a superpolynomially large challenge space. Then* $\mathsf{Sig}[\Pi]$ *is existentially unforgeable under chosen message attack (UF-CMA). If* Π *in addition has computationally unique responses,* $\mathsf{Sig}[\Pi]$ *is* strongly *existentially unforgeable under chosen message attack (sUF-CMA).*

In [Unr17] (Theorem 24, and 25, respectively), it is proven that an extractable FS proof system (of an HVZK Σ-protocol, and of an HVZK Σ-protocol with computationally unique responses, respectively) satisfies the stronger notion of *(strong) simulation-sound extractability.* In addition, it is shown that such a FS proof system gives rise to a (s)UF-CMA signature scheme if the underlying relation is hard. Corollary 3.22 implies that $\mathsf{FS}[\Pi^*]$ is indeed extractable if Π is extractable. Below we rely on the proof in [Unr17] to argue simulation-sound extractability, only pointing out a particular difference for the multi-round case.

Proof (sketch). Since Π is a quantum proof of knowledge, so is Π^*. By Corollary 3.22, $\mathsf{FS}[\Pi^*]$ is a quantum proof of knowledge (extractable), and by Theorem 20 in [Unr17] (which easily generalizes to the multi-round setting), completeness, unpredictable commitments[35] and HVZK of Π^* together imply ZK for

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

except with negligible probability.

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

Together with the fact that commit-and-open PCIPs can easily be made quantum extractable in the right sense by using standard hash-based commitments based on a collapsing hash function, we obtain the security of the MQDSS signature scheme. Recall that the standard hash-based commitment scheme works as follows. On input s, the commitment algorithm samples a random opening string u and outputs it together with the commitment $c = H(s, u)$. Opening just works by recomputing the hash and comparing it with c . Note that, while this commitment scheme is collapse-binding [Unr16], we need the stronger property of collapsingness of the function defined by the commitment algorithm that, on input a string and some randomness, outputs a commitment (collapse-binding only requires the collapsingness with respect to the committed string, not the opening information).[36]

Corollary 3.42 (sUF-CMA of MQDSS). *Let Π_{SSH} be the 5-round identification scheme from [SSH11] repeated in parallel a suitable number of times and instantiated with the standard hash-based commitment scheme using a collapsing hash function. Then the Fiat-Shamir signature scheme constructed from Π_{SSH} is sUF-CMA.*

Proof (sketch). In Π_{SSH}, the honest prover's first message consists of two commitments, and the second and final messages contain functions of the strings committed to in the first message. This structure, together with the computational binding property (implied by the collapse binding property) of the commitments, immediately implies that Π_{SSH} has computationally unique responses. According to Corollary 3.32 in the appendix, Π_{SSH} is a quantum proof of knowledge. It also has HVZK according to [SSH11]. Finally, the first message of Π_{SSH} is clearly unpredictable. An application of Theorem 3.41 finishes the proof. □

[36] Alternatively, the extractor should be adjusted to only measure the message part of the response.

3.7.3 Sequential Or-Proofs

A second application of our multi-input version of the measure-and-reprogram result is to the OR-proof as introduced by Liu, Wei and Wong [LWW04] and further analyzed by Fischlin, Harasser and Janson [FJ20]. This is an alternative (non-interactive) proof for proving existence/knowledge of (at least) one of two witnesses without revealing which one, compared to the well known technique by Cramer, Damgård and Schoenmakers [CDS94].

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

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

is satisfied. Fischlin et al. call this construction *sequential OR proof.* We emphasize that the two challenges c_0 and c_1 are computed "over cross", i.e., the challence c_0 for the execution of $\mathbf{\Sigma}_0$ is computed by hashing a_1, and vice versa. It is straightforward to verify that if $\mathbf{\Sigma}_0$ and $\mathbf{\Sigma}_1$ are special honest-verifier zero-knowledge, meaning that for any challenge c and response z one can efficiently compute a first message a such that (a, c, z) is accepted, then it is sufficient to be able to succeed in *one* of the two *interactive* protocols $\mathbf{\Sigma}_0$ and $\mathbf{\Sigma}_1$ in order to honestly produce such an OR-proof $\pi_\vee$. Thus, depending on the context, it is sufficient that one instance is in the corresponding language, or that the prover knows one of the two witnesses, to produce $\pi_\vee$. Indeed, if, say, $x_0 \in \mathcal{L}_0$ (and a witness w_0 is available), then $\pi_\vee$ can be produced as follows. Prepare a_0 according to $\mathbf{\Sigma}_0$, compute $c_1 := H(0, x_0, x_1, a_0)$ and simulate z_1 and a_1 using the special honest-verifier zero-knowledge property of $\mathbf{\Sigma}_1$ so that $V_1(x_1, a_1, c_1, z_1)$ is satisfied, and then compute the response z_0 for the challenge $c_0 := H(1, x_0, x_1, a_1)$ according to $\mathbf{\Sigma}_0$.

On the other hand, intuitively one expects that one of the two instances must be true in order to be able to successfully produce a proof. Indeed, [LWW04] shows security of the sequential OR in the (classical) ROM. [FJ20] go a step further and show security in the (classical) *non-programmable* ROM. Here we show that our multi-input version of the measure-and-reprogram result (as a matter of fact the 2-input version) implies security in the QROM.

Theorem 3.43. *There exists a black-box quantum polynomial-time interactive algorithm $\hat{\mathcal{P}}$, which first outputs a bit b and two instances x_0, x_1, and in a second stage acts as an interactive prover that runs $\mathbf{\Sigma}_b$ on instance x_b, such that for* *any adversary $\mathcal{A}$ making q queries to a uniformly random function H and for any x_0°, x_1°:*

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

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

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

Proof. The algorithm $\mathcal{A}$ fits well into the statement of Theorem 3.13 with the two extractable inputs $\tilde{x}_0 = (0, x_0, x_1, a_0)$ and $\tilde{x}_1 = (1, x_0, x_1, a_1)$. Thus, we can consider the 3-stage algorithm $\mathcal{S}$ ensured by Theorem 3.13, which behaves as follows with at least the probability given by the right hand side of the claimed inequality. In the first stage, it outputs a permutation on the set $\{0, 1\}$, which we represent by a bit $b \in \{0, 1\}$ with $b = 0$ corresponding to the identity permutation, as well as $\tilde{x}_b = (b, x_0, x_1, a_b)$. On input a random $\Theta_b = c_{1-b}$ ("locally" chosen by $\hat{\mathcal{P}}$), $\mathcal{S}$ then outputs $\tilde{x}_{1-b} = (1 - b, x_0, x_1, a_{1-b})$. Finally, on input a random $\Theta_{1-b} = c_b$ (provided by $\mathcal{V}_b$ as the challenge upon the first message a_b), $\mathcal{S}$ outputs z_0, z_1 so that $V_\vee$ is satisfied with the challenges c_b and c_{1-b}, and thus in particular $V_b\big(x_b, a_b, c_b, z_b\big)$ is satisfied. This directly shows the existence of $\hat{\mathcal{P}}$ as claimed.

□

Section 3.8

Tightness of the reductions

Here, we show tightness of our generic Fiat-Shamir reduction, for both the Σ-protocol and the multi-round versions. We start with proving tightness of Theorem 3.7 (up to essentially a factor 4). This implies that a $O(q^2)$-loss is unavoidable in general. Indeed, the following result shows that for a large and natural class of Σ-protocols Σ, there exists an attack against $\mathsf{FS}[\Sigma]$ that succeeds with a probability q^2 times larger than the best attack against Σ. The attack is based on an application of Grover's quantum algorithm for unstructured search.

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

Theorem 3.44. *Let $\mathcal{L}$ be a language, and let Σ be a Σ-protocol for $\mathcal{L}$ with challenge set $\mathcal{C}$, special soundness and perfect honest-verifier zero-knowledge. Furthermore, we assume that the triples (a, c, z) produced by the simulator $\mathcal{S}_{\text{ZK}}(x)$ are always accepted by the verifier even for instances $x \notin \mathcal{L}$, and that a has min-entropy γ.*[37] *Then for any q such that $(q^2+1)\cdot e^2 \cdot (5q)^6 < |\mathcal{C}|$ and $2^\gamma/(5q)^3 > 2$, there exists a q-query dishonest prover that succeeds with probability at least $q^2/|\mathcal{C}|$ in producing a valid $\mathsf{FS}[\Sigma]$-proof for an instance $x \notin \mathcal{L}$.*

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

Remark 3.45. It is not hard to see that Theorem 3.44 still holds in the following two variations of the statement. (1) $H(x, a)$ is random and independent for

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

different choices of a, but is *not* necessarily independent for different choices of x. (2) The Σ-protocol Σ is replaced by Σ', which has its challenge enlarged with a certain number of bits that are ignored otherwise, in line with Remark 3.21, and $\mathsf{FS}[\Sigma']$ then uses an H with a correspondingly enlarged range.[38]

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

- Define the function $f^H : R \to \{0,1\}$ (where R is the set of random coins for $\mathcal{S}_{\mathrm{ZK}}$) as
$$f^H(\rho) = \begin{cases} 1 & \text{for } \mathcal{S}_{\mathrm{ZK}}(x;\rho) \to (a,c,z) \wedge H(x||a) = c \\ 0 & \text{otherwise.} \end{cases}$$
- Use Grover's algorithm for q steps, to try and find ρ s.t. $f(\rho) = 1$
- Run $\mathcal{S}_{\mathrm{ZK}}(x;\rho) \to (a,c,z)$ and output $(x, a||z)$.

Let p_1^H be the fraction of random coins from R that map to 1 under f^H. Note that by the special soundness of Σ, in any accepting triple a determines c and we thus have $\mathbb{E}_H[p_1^H] = \frac{1}{|\mathcal{C}|}$. By the way Grover works, after q iterations (requiring q queries to H) the probability p_2^H of finding such an input is $\sin^2((2q+1)\Theta^H)$, where $0 \le \Theta^H \le \pi/2$ is such that $\sin^2(\Theta^H) = p_1^H$. Now as long as Θ is not too large to begin with (i.e. as long as the Grover search will not 'overshoot'), p_2^H is approximately a factor q^2 larger than p_1^H. Our goal will be to show that also on average over H, the improvement is at least q^2. To this end we define $H_{\mathrm{bad}} := \{H : p_1^H > \sin^2(\frac{\pi}{6q+3})\}$ and H_{good} its complement. Then,

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

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

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

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

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

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

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

Therefore,

$$\begin{aligned} \mathop{\mathbb{E}}_H[p_2^H] &\geq \mathop{\mathbb{E}}_H\left[p_2^H | H \in H_{\text{good}}\right] \cdot \Pr_H[H \in H_{\text{good}}] \\ &\geq (q+1)^2 \cdot \mathop{\mathbb{E}}_H\left[p_1^H | H \in H_{\text{good}}\right] \cdot \Pr_H[H \in H_{\text{good}}] \\ &\geq (q+1)^2 \cdot \left(\mathop{\mathbb{E}}_H[p_1^H] - \Pr_H[H \in H_{\text{bad}}]\right) . \end{aligned} \tag{14}$$

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

We first define the following equivalence relation:

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

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

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

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

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

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

$$\alpha = \Pr_H[p_1^H > \sin^2\left(\frac{\pi}{6q+3}\right)]$$

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

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

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

We use the following Chernoff bound:

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

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

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

where we used $\frac{2^\gamma}{(5q)^3} > 2$ in the second to last, and $|\mathcal{C}| > (q^2+1)\cdot e^2\cdot(5q)^6$ in the last inequality. Plugging this bound into Equation 14, we get

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

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

The tightness of Corollary 3.20 follows from the above tightness result for the case of Σ-protocols in a fairly straightforward manner.

Theorem 3.46. *For all positive integers n and q, there exists a $(2n+1)$-round PCIP Π with soundness error ϵ and challenge space $\mathcal{C}$ such that $|\mathcal{C}| \geq 1/\epsilon$ and such that there exists a q-query dishonest prover $\mathcal{A}$ on $\mathsf{FS}(\Pi)$ with success probability at least $n^{-2n}q^{2n}\epsilon$.*

Before proving the theorem, we show how it implies the tightness of Corollary 3.20.

Corollary 3.47. *The security loss in the bound in Corollary 3.20 is optimal (i.e. matched by an attack), up to a multiplicative factor that depends on n only.*

Proof. Let $\mathsf{\Pi}$ be a PCIP as shown to exist in Theorem 3.46. Let ϵ_{Π}, and $\epsilon_{\mathsf{FS}(\mathsf{\Pi})}(q)$, be the soundness error of $\mathsf{\Pi}$, and the one of its Fiat Shamir transformation against q-query adversaries, respectively. By Theorem 3.46,

$$\epsilon_{\mathsf{FS}(\mathsf{\Pi})}(q) \geq n^{-2n} q^{2n} \epsilon_{\mathsf{\Pi}}. \tag{15}$$

Theorem 3.20, on the other hand, yields

$$\begin{aligned} \epsilon_{\mathsf{\Pi}} &\geq \frac{n!}{(2q+n+1)^{2n}} \epsilon_{\mathsf{FS}(\mathsf{\Pi})}(q) - \frac{n!}{|\mathcal{C}|} && (16) \\ &\geq \frac{n!}{(2q+n+1)^{2n}} \epsilon_{\mathsf{FS}(\mathsf{\Pi})}(q) - n! \epsilon_{\mathsf{\Pi}}, && (17) \end{aligned}$$

where we used the condition on the challenge space size from Theorem 3.46 in the last line. Rearranging terms we obtain

$$\begin{aligned} \epsilon_{\mathsf{FS}(\mathsf{\Pi})}(q) &\leq (2q+n+1)^{2n} \left(1 + \frac{1}{n!}\right) \epsilon_{\mathsf{\Pi}}(q) && (18) \\ &\leq 2(n+3)^2 q^{2n} \epsilon_{\mathsf{\Pi}}(q), && (19) \end{aligned}$$

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

$$c_1 q^{2n} \epsilon_{\mathsf{\Pi}} \leq \epsilon_{\mathsf{FS}(\mathsf{\Pi})}(q) \leq c_2 q^{2n} \epsilon_{\mathsf{\Pi}}. \tag{20}$$

□

Proof (of Theorem 3.46). Let $\hat{\Sigma}$ be a Σ-protocol for a language $\mathcal{L}$ fulfilling the requirements of Theorem 3.44. Let the challenge space be denoted by $\hat{\mathcal{C}}$. Given an arbitrary positive integer, we define an $(2n+1)$-round PCIP $\mathsf{\Pi}$ for the same language $\mathcal{L}$ by means of n sequential independent executions of $\hat{\Sigma}$. Concretely, the $2n+1$ messages of $\mathsf{\Pi}$ are given in terms of the messages $\hat{a}_i, \hat{c}_i$ and $\hat{z}_i$ of the i-th repetition of $\hat{\Sigma}$ as

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

where r_i is an independent random string of arbitrary (but fixed) length, which is ignored otherwise (in line with Remark 3.21). The purpose of r_i is to make the challenge space $\mathcal{C}$ of $\mathsf{\Pi}$ arbitrary large, as required. The verification procedure of $\mathsf{\Pi}$ simply checks if all the triples $(\hat{a}_i, \hat{c}_i, \hat{z}_i)$ are accepted by $\hat{\Sigma}$. By the special soundness property of $\hat{\Sigma}$, the soundness error of this PCIP is $\epsilon = |\hat{\mathcal{C}}|^{-n}$.

Using Theorem 3.44, we can attack the Fiat-Shamir transformation of $\hat{\Sigma}$ repeatedly to devise an attack agains $\mathsf{FS}(\mathsf{\Pi})$: first use Theorem 3.44 to find $\hat{a}_1$ and $\hat{z}_1$, then use it again to find $\hat{a}_2$ and $\hat{z}_2$, etc., having the property that with the correctly computed challenges these form valid triples for an instance $x \notin \mathcal{L}$. In each invocation of Theorem 3.44 we use a q'-query attack, which then succeeds with probability $q'^2/|\hat{\mathcal{C}}|$. Thus, using in total $q = nq'$ queries, we succeed in breaking $\mathsf{FS}[\mathsf{\Pi}]$ with probability $q'^{2n}/|\hat{\mathcal{C}}|^n = n^{-2n}q^{2n}\epsilon$, as claimed.

There are two issues we neglected in the above argument. First, we actually employ Theorem 3.44 for attacking a *variant* of $\hat{\Sigma}$ that has its challenge enlarged (and thus is not special sound); and, second, the challenge c_i is computed as

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

which is *not* a uniformly random function of x and $\hat{a}_i$ (but only of $\hat{a}_i$). However, by Remark 3.45, the attack from Theorem 3.44 still applies. □

Chapter 4

Online Extractability

Chapter contents

Section 4.1

Introduction

Background. *Extractability* plays an important role in cryptography. In an extractable protocol, on a high level, an algorithm $\mathcal{A}$ sends messages that depend on some secret s, and while the secret remains private in an honest run of the protocol, an *extractor* can learn s via some form of enhanced access to $\mathcal{A}$. The probably most prominent example is that of (zero-knowledge) *proofs* (or *arguments*) *of knowledge*, for which, by definition, there must exist an extractor that manages to extract a witness from any successful yet possibly dishonest prover. Another example are *extractable commitments*, which have a wide range of applications. Hash-based extractable commitments are extremely simple to construct and prove secure in the random-oracle model (ROM) [Pas03]. Indeed, when the considered hash function H is modelled as a random oracle, the hash input x for the commitment $c = H(x)$, where $x = s\|r$ consists of the actual secret s and randomness r, can be extracted simply by finding a query x to the random oracle that yielded c as an output.

The general notion of extractability comes in different flavors. The most well-known example is extraction by *rewinding*. Here, the extractor is allowed to run $\mathcal{A}$ several times, on the same private input and using different randomness. This is the notion usually considered in the context of proofs/arguments of knowledge. In some contexts, extraction via rewinding access is not possible. For example, the UC security model prohibits the simulator to rewind the adversary. In other occasions, rewinding may be possible but not desirable due to a loss of efficiency, which stems from having to run $\mathcal{A}$ multiple times. In comparison, so-called *straightline* extraction works with a single ordinary run of $\mathcal{A}$, without rewinding. Instead, the extractor is then assumed to know some trapdoor information, or it is given enhanced control over some part of the setting. For instance, in the above construction of an extractable commitment, the extractor is given “read access” to $\mathcal{A}$’s random-oracle queries.

Another binary criterion is whether the extraction takes place *on-the-fly*, i.e., during the run of the protocol, or *after-the-fact*, i.e., at the end of the execution. For instance, in the context of proving CCA security for an encryption scheme, to simulate decryption queries without knowing the secret key, it is necessary to extract the plaintext for a queried ciphertext on-the-fly; otherwise, the attacker may abort and not produce the output for which the reduction is waiting.

The extractability of our running example of an extractable commitment in the ROM is *both*, straightline and on-the-fly; we refer to this combination as *online* extraction. This is what we are aiming for in this work: online extractability of (general) hash-based commitments, but now with *post-quantum security.*

For post-quantum security, the ROM needs to be replaced by the *quantum* random-oracle model (QROM) [BDF+11], to reflect the fact that attackers can implement hash functions on a quantum computer. Here, adversaries have quantum superposition access to the random oracle. Many ROM techniques fail in the QROM due to fundamental features of quantum information, such as the so-called *no-cloning principle.* In particular, it is impossible to maintain a query transcript (a fact sometimes referred to as the *recording barrier*), and so one cannot simply "search for a query x to the random oracle", as was exploited for the (classical) RO-security of the extractable-commitment example.

A promising step in the right direction is the compressed-oracle technique, developed by Zhandry [Zha19a]. This technique enables to maintain *some sort* of a query transcript, but now in the form of a quantum state. This state can be inspected via quantum measurements, offering the possibility to learn some information about the interaction history of an algorithm $\mathcal{A}$ and the random oracle. However, since quantum measurements disturb the state to which they are applied, and this disturbance is often hard to control, this inspection of the query transcript can *per-se*, i.e., without additional argumentation, only be done at the end of the execution (see the Related Work paragraph for more on this).

Our Results. Our main contribution is the following generic extractability result in the QROM. We consider an arbitrary quantum query algorithm $\mathcal{A}$ in the QROM, which announces during its execution some classical value t that is supposed to be equal to $f(x, H(x))$ for some x. Here, f is an arbitrary fixed function, subject to that it must tie t sufficiently to x and $H(x)$, e.g., there must not be too many y's with $f(x, y) = t$; a canonical example is the function $f(x, y) = y$ so that t is supposed to be $t = H(x)$. In general, it is helpful to think of $t = f(x, H(x))$ as a commitment to x. We then show that x can be *efficiently extracted* with almost certainty. The extraction works *online* and is by means of a simulator $\mathcal{S}$ that simulates the quantum random oracle, but which additionally offers an *extraction interface* that produces a guess $\hat{x}$ for x when queried with t. The simulation is statistically indistiguishable from the real quantum random oracle, and $\hat{x}$ is such that whenever $\mathcal{A}$ outputs x with

$f(x, H(x)) = t$ at some later point, $\hat{x} = x$ except with negligible probability, while $\hat{x} = \emptyset$ (some special symbol) indicates that $\mathcal{A}$ will not be able to output such an x.

The simulator $\mathcal{S}$ simulates the random oracle using Zhandry's compressed-oracle technique, and extraction is done via a suitable measurement of the compressed oracle's internal register. The technical core of our result is a new bound for the operator norm $\|[O, M]\|$ of the commutator of O, the unitary operator that describes the evolution of the compressed oracle, and of M, the measurement that is used to extract x. This commutator bound allows us to show that the extraction measurement disturbs the behavior of the compressed oracle only by a negligible amount, and so can indeed be performed *on-the-fly*. At first glance, our technical result has some resemblance with Lemma 39 in [Zha19a], which also features an almost-commutativity property, and, indeed, with Lemma 4.7 we use (a reformulated version of) Lemma 39 in [Zha19a] as a first step in our proof. However, the challenging part of the main proof consists of lifting the almost-commutativity property of the "local" projectors Π^x from Lemma 4.7 to the "global" measurement M (Lemma 4.7).

We emphasize that even though the existence of the simulator with its extraction interface is proven using the compressed-oracle technique, our presentation is in terms of a black-box simulator $\mathcal{S}$ with certain interfaces and with certain promises on its behavior, abstracting away all the (mainly internal) quantum workings. This makes our generic result applicable (e.g. for the applications discussed below) without the need to understand the underlying quantum aspects.

A first concrete application of our generic result is in the context of so-called commit-and-open Σ-protocols. These are (typically honest-verifier zero-knowledge) interactive proofs of a special form, where the prover first announces a list of commitments and is then asked to open a subset of them, chosen at random by the verifier. We show that, when implementing the commitments with a typical hash-based commitment scheme (like committing to s by $H(s\|r)$ with a random r), such Σ-protocols allow for *online* extraction of a witness in the QROM, with a *smaller security loss* than witness extraction via rewinding.

Equipped with our extractable RO-simulator $\mathcal{S}$, the idea for the above online extraction is very simple: we simulate the random oracle using $\mathcal{S}$ and use its extraction interface to extract the prover's commitments from the first message of the Σ-protocol. As we work out in detail, this procedure gives rise to an online witness extractor that has a polynomial additive overhead in running time compared to the considered prover, and that outputs a valid witness with a probability that is *linear* in the difference of the prover's success probability and the trivial cheating probability, up to an additive error. Using rewinding techniques, on the other hand, incurs a *square-root* loss in success probability classically and a *cube-root* loss quantumly for special-sound Σ-protocols, and typically an even worse loss in case of weaker soundness guarantees, like a k-th-root loss classically and a $(2k+1)$-th-root loss quantumly for k-sound protocols. Furthermore, we show that the dominating additive loss of our reduction is necessary in general, due to attacks on the computational binding property of the random-oracle-based commitments. Along the way, we set up a definitional framework for generalized special soundness notions that might be of independent interest.

A second application of our extractable RO-simulator is a security reduction for the Fujisaki-Okamoto (FO) transformation. We offer the first complete post-quantum security proof of the *textbook* FO transformation [FO99], with concrete security bounds. Most of the prior post-quantum security proofs had to adjust the transformation to facilitate the proof (like [HHK17]); those security proofs either consider a FO variant that employs an *implicit-rejection* routine, i.e., where the decapsulation algorithm outputs a pseudo-random key upon an invalid ciphertext rather than a rejection message, or have to resort to an additional "key confirmation" hash [TU16] that is appended to the ciphertex, thus increasing the ciphertext size. The *unmodified* FO transformation was analyzed in [Zha19a] and [KKPP20]; however, as we explain in detail in Section 4.6.3, the given post-quantum security proofs are incomplete, both having the same gap.

Beyond its theoretical relevance of showing that no adjustment is necessary to admit a post-quantum security proof, the security of the original unmodified FO transformation with explicit rejection in particular ensures that the conservative variant with implicit rejection remains secure even when the decapsulation algorithm is not implemented carefully enough and admits a side-channel attack that reveals information on whether the submitted ciphertext is valid or not.

The core idea of our proof for the textbook FO transformation is to use the extractability of the RO-simulator to handle the decryption queries. Indeed, letting $f(x,y)$ be the encryption $Enc_{pk}(x;y)$ of the message x under the randomness y, a "commitment" $t = f(x,H(x))$ is then the encryption of x under the derandomized scheme, and so the extraction interface recovers x.

Related Work. The compressed-oracle technique has proven to be a powerful tool for lifting classical ROM proofs to the QROM setting. Examples are [LZ19a; CFHL21] for quantum query complexity lower bounds and [HM21] for space-time trade-off bounds, [CMS19] for the security of succinct arguments, [AMRS20] for quantum-access security, and [BHH+19] for a new "double-sided" O2H lemma in the context of the FO transformation. In these cases, the argument exploits the possibility to extract information on the interaction history of the algorithm $\mathcal{A}$ and the (compressed) oracle *after-the-fact*, i.e., at the very end of the run.

In addition, some tools have been developed that allow measuring (the internal state of) the compressed oracle *on-the-fly*, which then causes the state, and thus the behavior of the oracle, to change. In some cases, the disturbance is significant yet asymptotically good enough for the considered application, causing "only" a polynomial blow-up of a negligible error term, as, e.g., in [LZ19b] for proving the security of the Fiat-Shamir transformation. In other cases [Zha19a; CMSZ19], it is shown for some limited settings that certain measurements do not render the simulation of the random oracle distinguishable (except for negligible advantage). The indifferentiability result in [CMSZ19], for example, only uses measurements that have an almost certain outcome.

In particular, [Zha19a] contains a security reduction for the Fujisaki-Okamoto (FO) transformation that implicitly uses a measurement similar to the one we analyze in Section 4.3, but without analyzing the disturbance it causes. We discuss this in more detail in Section 4.6.3. The same gap exists in recent follow-up work by Katsumata, Kwiatkowski, Pintore and Prest [KKPP20], who follow the FO proof outline from [Zha19a].

Section 4.2

Preliminaries

For Section 4.3 and 4.4 (only), we assume some familiarity with the mathematics of quantum information as well as with the compressed-oracle technique of [Zha19a]. We refer the reader to Chapter 2 for a general introduction to both, and summarize below the concepts that will be of particular importance.

For simplicity, we will express things in the remainder of this chapter in terms of the inefficient variant of the compressed oracle, but we stress that all relevant unitaries and measurements can be efficiently computed, as is explained in Section 2.4.

4.2.1 Mathematical Preliminaries

For a classical or quantum algorithm $\mathcal{A}$, we denote by Time$[\mathcal{A}]$ the time complexity of $\mathcal{A}$ (say, given by the number of gates from some universal gate set in a circuit for $\mathcal{A}$). For a function or algorithm f, we slightly abuse notation and write Time$[f]$ to denote the time complexity of (an algorithm computing) f. (The functions considered here come with an algorithm to compute them).

Let $\mathcal{H}$ be a finite-dimensional complex Hilbert space. We use the standard bra-ket notation for the vectors in $\mathcal{H}$ and its dual space. For an operator $A \in \mathcal{L}(\mathcal{H})$, we denote by $\|A\|$ its *operator norm*, i.e., $\|A\| = \max_{|\psi\rangle} \|A|\psi\rangle\|$, where the max is over all $|\psi\rangle \in \mathcal{H}$ with norm 1. We assume the reader to be familiar with basic properties of these norms, like triangle inequality, $\||\varphi\rangle\langle\psi|\| = \||\varphi\rangle\|\||\psi\rangle\|$, $\|A|\varphi\rangle\| \le \|A\|\||\varphi\rangle\|$, $\|AB\| \le \|A\|\|B\|$, etc. Less well known may be the inequality[39]

$$\||\varphi\rangle\langle\psi| - |\psi\rangle\langle\varphi|\| \le \||\varphi\rangle\|\||\psi\rangle\| \, . \tag{21}$$

Another basic yet important property that we will exploit is the following.

Lemma 4.1. *Let A and B be operators in $\mathcal{L}(\mathcal{H})$ with $A^\dagger B = 0$ (i.e., they have orthogonal images) and $AB^\dagger = 0$ (i.e., they have orthogonal supports). Then, $\|A + B\| \le \max\{\|A\|, \|B\|\}$.*

Exploiting that $\|A \otimes B\| = \|A\|\|B\|$, the following is a direct consequence of Lemma 4.1.

Corollary 4.2. *If $A = \sum_x |x\rangle\langle x| \otimes A^x$, i.e., A is a controlled operator, then $\|A\| \le \max_x \|A^x\|$.*

Definition 4.3. *For operators $A, B \in \mathcal{L}(\mathcal{H})$, the* commutator *is defined as $[A, B] := AB - BA$.*

Some obvious properties of the commutator are:

[39] It is immediate for normalized $|\phi\rangle$ and $|\psi\rangle$ when expanding both vectors in an orthonormal basis containing $|\varphi\rangle$ and $\frac{|\psi\rangle - \langle\varphi|\psi\rangle|\varphi\rangle}{\sqrt{1-|\langle\varphi|\psi\rangle|^2}}$, and the general case then follows by homogeneity of the norms.

$$[B, A] = -[A, B] = [A, \mathbb{1} - B] \qquad \text{and} \quad [A \otimes \mathbb{1}, B \otimes C] = [A, B] \otimes C\,, \tag{22}$$

as well as

$$[AB, C] = A[B, C] + [A, C]B \tag{23}$$

Combining the right equality in (22) with basic properties of the operator norm, if $\|C\| \leq 1$, e.g., if C is a unitary or a projection, we have

$$\|[A \otimes \mathbb{1}, B \otimes C]\| = \|[A, B]\|\|C\| \leq \|[A, B]\|\,. \tag{24}$$

It is common in quantum information science to write A_X to emphasize that the operator A acts on *register* X, i.e., on a Hilbert space $\mathcal{H}_X$ that is labeled by the letter/symbol X. It is then understood that when applied to registers X and Y, say, A_X acts as A on register X and as identity $\mathbb{1}$ on register Y, i.e., A_X is identified with $A_X \otimes \mathbb{1}_Y$. Property (24) would then e.g. be written as $\|[A_X, B_X \otimes C_Y]\| \leq \|[A_X, B_X]\|$. In this chapter, we will write or not write these subscripts emphasizing the register(s) at our convenience; typically we write them when the argument crucially depends on the registers, and we may omit them otherwise.

In case of a hybrid classical-quantum state, consisting of a randomized classical value x that follows a distribution p and of a quantum register W with a state ρ_W^x that depends on x, we write $[x, W] = \sum_x p(x)|x\rangle\langle x| \otimes \rho_W^x$.[40] When the distribution p and the density operators ρ_W^x are implicitly given by a game (or experiment) $\mathcal{G}$ then we may write $[x, W]_{\mathcal{G}}$, in particular when considering and comparing different such games. For instance, we write $\delta\big([x, W]_{\mathcal{G}}, [x, W]_{\mathcal{G}'}\big)$ for the trace distance of the respective density matrices in game $\mathcal{G}$ and in game $\mathcal{G}'$.

[40] In this equality and at other occasions, we use the same letter, here x, for the considered *random variable* as well as for a *particular value*.

Section 4.3

Main Technical Result: A Commutator Bound

Our main technical result is a bound on the operator norm of the commutator $[O_{XYD}, M_{DP}]$ of the unitary O_{XYD}, which describes the evolution of the compressed oracle, and the (purified) measurement M_{DP}. Informally, this measurement checks if there is a pair (x, y) in the database satisfying a given relation. If yes, it outputs (the smallest such) x, otherwise it outputs $\emptyset$. A small bound on this commutator means that performing this measurement during the runtime of an oracle algorithm $\mathcal{A}$ interacting with a (compressed) random oracle, has little effect.

4.3.1 Setup and the Technical Statement

Throughout this section, we consider an arbitrary but fixed relation $R \subset \mathcal{X} \times \{0,1\}^n$. A crucial parameter of the relation R is the number of y's that fulfill the relation together with x, maximized over all possible $x \in \mathcal{X}$:

$$\varGamma_R := \max_{x \in \mathcal{X}} \left| \left\{ y \in \{0,1\}^n \middle| (x,y) \in R \right\} \right| . \tag{25}$$

Given the relation R, we consider the following projectors:

$$\varPi^x_{D_x} := \sum_{\substack{y \text{ s.t.} \\ (x,y) \in R}} |y\rangle\langle y|_{D_x} \quad \text{and} \quad \varPi^\emptyset_D := \mathbb{1}_D - \sum_{x \in \mathcal{X}} \varPi^x_{D_x} = \bigotimes_{x \in \mathcal{X}} \bar{\varPi}^x_{D_x} \tag{26}$$

with $\bar{\varPi}^x_{D_x} := \mathbb{1}_{D_x} - \varPi^x_{D_x}$. Informally, $\varPi^x_{D_x}$ checks whether register D_x contains a value $y \neq \bot$ such that $(x, y) \in R$. We then define the measurement $\mathcal{M} = \mathcal{M}^R$ to be given by the projectors

$$\varSigma^x := \bigotimes_{x' < x} \bar{\varPi}^{x'}_{D_{x'}} \otimes \varPi^x_{D_x} \quad \text{and} \quad \varSigma^\emptyset := \mathbb{1} - \sum_{x'} \varSigma^{x'} = \bigotimes_{x'} \bar{\varPi}^{x'}_{D_{x'}} = \varPi^\emptyset \tag{27}$$

i where x ranges over all $x \in \mathcal{X}$. Informally, a measurement outcome x means that register D_x is the first that contains a value y such that $(x, y) \in R$; outcome $\emptyset$ means that no register contains such a value. For technical reasons, we consider the *purified* measurement $M_{DP} = M^R_{DP} \in \mathcal{L}(\mathcal{H}_D \otimes \mathcal{H}_R)$ given by the unitary[41]

$$M_{DP} := \sum_{x \in \mathcal{X} \cup \{\emptyset\}} \Sigma^x \otimes \mathsf{X}^x : |\varphi\rangle_D |w\rangle_P \mapsto \sum_{x \in \mathcal{X} \cup \{\emptyset\}} \Sigma^x |\varphi\rangle_D |w + x\rangle_P \,. \tag{28}$$

The following main technical result is a bound on the norm of the commutator$[O_{XYD}, M_{DP}]$.

Theorem 4.4. *For any relation $R \subset \mathcal{X} \times \{0,1\}^n$ and Γ_R as defined in Equation* (25)*, the purified measurement M_{DP} defined in Equation* (28) *almost commutes with the oracle unitary O_{XYD}:*

$$\big\| \, [O_{XYD}, M_{DP}] \, \big\| \leq 8 \cdot 2^{-n/2} \sqrt{2\Gamma_R} \,.$$

We note that Lemma 8 in [CMS19] (with the subsequent discussion there) also provides a bound on the norm of a commutator involving O_{XYD}; however, there are various differences that make the two bounds incomparable. E.g., we consider a specific *measurement* whereas Lemma 8 in [CMS19] is for a rather general *projector*. See further down for a comparison with Lemma 39 in [Zha19a].

Corollary 4.5. *For any state vector $|\psi\rangle \in \mathcal{H}_{WXYDP}$, with W an arbitrary additional register, the state vectors$|\psi'\rangle := O_{XYD} M_{DP} |\psi\rangle$ and $|\psi''\rangle := M_{DP} O_{XYD} |\psi\rangle$ satisfy*

$$\delta\big(|\psi'\rangle\langle\psi'|, |\psi''\rangle\langle\psi''|\big) \leq 8 \cdot 2^{-n/2} \sqrt{2\Gamma_R} \,.$$

The same holds for mixed states $\rho' := O_{XYD} M_{DP} \rho M_{DP}^\dagger O_{XYD}^\dagger$ and $\rho'' := M_{DP} O_{XYD} \rho O_{XYD}^\dagger M_{DP}^\dagger$.

Proof. By elementary properties and applying Theorem 4.4, we have that

$$\begin{aligned} \big\| |\psi'\rangle - |\psi''\rangle \big\| = \big\| (O_{XYD} M_{DP} - M_{DP} O_{XYD}) |\psi\rangle \big\| &\leq \big\| [O_{XYD}, M_{DP}] \big\| \\ &\leq 8 \cdot 2^{-n/2} \sqrt{2\Gamma_R} \,, \end{aligned}$$

and the claim on the trace distance then follows from (1). The claim for mixed states follows from purification. □

[41] Both in X^x and in $w + x$ we understand $x \in \mathcal{X} \cup \{\emptyset\}$ to be encoded as an element in $\mathbb{Z}/(|\mathcal{X}|+1)\mathbb{Z}$, $\dim(\mathcal{H}_P) = d := |\mathcal{X}| + 1$, and $\mathsf{X} \in \mathcal{L}(\mathcal{H}_P)$ is the generalized Pauli of order d that maps $|w\rangle$ to $|w + 1\rangle$.

4.3.2 The Proof

We prove the Theorem 4.4 by means of the following two lemmas.

Lemma 4.6. *Let F and $O^x_{YD_x}$ be the unitaries introduced in Section* **??**, *and let $\mathit{\Pi}^x_{D_x}$ and $\mathit{\Pi}^\emptyset_D$ be as in (26). Set $\mathit{\Gamma}_x := \big|\{y \in \{0,1\}^n \big| (x,y) \in R\}\big|$. Then*

$$\big\|\,[F_{D_x}, \mathit{\Pi}^x_{D_x}]\,\big\| \leq 2^{-n/2}\sqrt{2\mathit{\Gamma}_x}\,, \qquad \textit{as well as}$$

$$\big\|\,[O^x_{YD_x}, \mathit{\Pi}^x_{D_x}]\,\big\| \leq 2 \cdot 2^{-n/2}\sqrt{2\mathit{\Gamma}_x} \quad \textit{and} \quad \big\|\,[O^x_{YD_x}, \mathit{\Pi}^\emptyset_D]\,\big\| \leq 2 \cdot 2^{-n/2}\sqrt{2\mathit{\Gamma}_x}\,.$$

The bound on $\|[F, \mathit{\Pi}^x]\|$ can be considered a compact reformulation of (a variant of) Lemma 39 in [Zha19a]. We state it in this form and (re-)prove it for convenience and completeness.

Proof (Lemma 4.6). Recalling from (2) that $F|y\rangle = |y\rangle + 2^{-n/2}|\delta\rangle$ with $|\delta\rangle := |\bot\rangle - |\hat{0}\rangle$, we have

$$[F, |y\rangle\langle y|] = F|y\rangle\langle y| - |y\rangle\langle y|F = 2^{-n/2}|\delta\rangle\langle y| - 2^{-n/2}|y\rangle\langle\delta|\,.$$

From this, it follows that

$$[F, \mathit{\Pi}^x] = \sum_{\substack{y\in\{0,1\}^n \\ (x,y)\in R}} [F, |y\rangle\langle y|] \leq 2^{-n/2}\,|\delta\rangle \sum_{\substack{y\in\{0,1\}^n \\ (x,y)\in R}} \langle y| - 2^{-n/2} \sum_{\substack{y\in\{0,1\}^n \\ (x,y)\in R}} |y\rangle\langle\delta|$$

and thus, using (21), that

$$\|[F, \mathit{\Pi}^x]\| \leq 2^{-n/2}\,\||\delta\rangle\| \Bigg\| \sum_{\substack{y\in\{0,1\}^n \\ (x,y)\in R}} \langle y| \Bigg\| \leq 2^{-n/2}\sqrt{2}\sqrt{\mathit{\Gamma}_x}\,.$$

For the second bound, let $C_{YD_x} = \text{CNOT}$ with CNOT as in (3), with the understanding that D_x is the control register and Y the target. Recall from (3) that $O^x_{YD_x} = F_{D_x}C_{YD_x}F_{D_x}$. Thus, using (23) twice and omitting the registers, we obtain

$$[O^x, \mathit{\Pi}^x] = F[CF, \mathit{\Pi}^x] + [F, \mathit{\Pi}^x]CF = FC[F, \mathit{\Pi}^x] + F[C, \mathit{\Pi}^x]F + [F, \mathit{\Pi}^x]CF\,.$$

Finally, we notice that $[C_{YD_x}, \mathit{\Pi}^x_{D_x}] = 0$, since projections on the control register of a CNOT commute with the CNOT. The claimed bound now follows from the derived bound on $[F, \mathit{\Pi}^x]$ together with Equation (24).

The third bound follows by recalling that $\mathit{\Pi}^{\emptyset}_{D} = \bigotimes_{x'} \bar{\mathit{\Pi}}^{x'}_{D_{x'}}$ is a tensor-product for which $O^x_{YD_x}$ acts trivially on all the components except for the component $\bar{\mathit{\Pi}}^{x}_{D_x}$, so with Equation (24) we obtain,

$$\|[O^x_{YD_x}, \mathit{\Pi}^{\emptyset}_{D}]\| \leq \|[O^x_{YD_x}, \bar{\mathit{\Pi}}^{x}_{D_x}]\| = \|[O^x_{YD_x}, \mathit{\Pi}^{x}_{D_x}]\|\,,$$

which completes the proof. □

The conceptually new and technically challenging ingredient to the proof of Theorem 4.4 is Lemma 4.7 below.[42]

Lemma 4.7. *The purified measurement M_{DP} defined in Equation* (28) *satisfies*

$$\big\|[F_{D_x}, M_{DP}]\big\| \leq 3\big\|[F_{D_x}, \mathit{\Pi}^{x}_{D}]\big\| + \big\|[F_{D_x}, \mathit{\Pi}^{\emptyset}_{D}]\big\| \qquad \textit{and}$$

$$\big\|[O^x_{YD_x}, M_{DP}]\big\| \leq 3\big\|[O^x_{YD_x}, \mathit{\Pi}^{x}_{D}]\big\| + \big\|[O^x_{YD_x}, \mathit{\Pi}^{\emptyset}_{D}]\big\|\,.$$

Proof. We do the proof for the second claim. The first is proven exactly the same way: the sole property we exploit from $O^x_{YD_x}$ is that it acts only on the D_x register within D, which holds for F_{D_x} as well. Let

$$\bar{\mathit{\Delta}}^{\xi} := \bigotimes_{\xi' < \xi} \bar{\mathit{\Pi}}^{\xi'}_{D_{\xi'}}$$

be the projection that accepts if no register $D_{\xi'}$ with $\xi' < \xi$ contains a value y' with $(\xi', y') \in R$, and let $\mathit{\Delta}^{\xi}$ be the complement. We then have, using that $\mathit{\Pi}^{\xi}$ and $\bar{\mathit{\Delta}}^{\xi}$ act on disjoint registers,

$$\mathit{\Sigma}^{\xi} = \bar{\mathit{\Delta}}^{\xi} \otimes \mathit{\Pi}^{\xi} = \mathit{\Pi}^{\xi}\bar{\mathit{\Delta}}^{\xi} = \bar{\mathit{\Delta}}^{\xi}\mathit{\Pi}^{\xi}\,. \tag{29}$$

We also observe that, with respect to the Loewner order, $\bar{\mathit{\Delta}}^{\xi'} \geq \bar{\mathit{\Delta}}^{\xi}$ for $\xi' < \xi$. Taking it as understood that $O^x_{YD_x}$ acts on registers Y and D_x, we can write

$$[O^x, M_{DP}] = \sum_{\xi} [O^x, \mathit{\Sigma}^{\xi}] \otimes \mathsf{X}^{\xi} + [O^x, \mathit{\Sigma}^{\emptyset}] \otimes \mathsf{X}^{\emptyset}\,. \tag{30}$$

[42] The challenging aspect of Lemma 4.7 is that M_{DP} is made up of an exponential number of projectors $\mathit{\Pi}^x$, and thus the obvious approach of using triangle inequality leads to an exponential blow-up of the error term. Naively, one might hope to avoid the exponential blow-up (at the cost of introducing a blow-up linear in the number of prior queries) by using the efficient representation of the compressed oracle (which is discussed in Section 2.4); however, the two representations are isometrically equivalent, and so switching the representation has no effect in that respect.

Exploiting basic properties of the operator norm and recalling that $\Sigma^{\emptyset} = \Pi^{\emptyset}_{D}$, we see that the norm of the last term is bounded by $\|[O^x, \Sigma^{\emptyset}]\| = \|[O^x, \Pi^{\emptyset}]\|$.

To deal with the sum in (30), we use $\mathbb{1} = \Delta^{\xi} + \bar{\Delta}^{\xi}$ to further decompose

$$[O^x, \Sigma^{\xi}] = \bar{\Delta}^{\xi}[O^x, \Sigma^{\xi}]\bar{\Delta}^{\xi} + \bar{\Delta}^{\xi}O^x, \Sigma^{\xi}]\Delta^{\xi} + \Delta^{\xi}[O^x, \Sigma^{\xi}]\bar{\Delta}^{\xi} + \Delta^{\xi}[O^x, \Sigma^{\xi}]\Delta^{\xi} \,. \tag{31}$$

We now analyze the four different terms. For the first one, using (29) we see that

$$\begin{aligned}\bar{\Delta}^{\xi}[O^x\!,\Sigma^{\xi}]\bar{\Delta}^{\xi} &= \bar{\Delta}^{\xi}\big(O^x\Sigma^{\xi} - \Sigma^{\xi}O^x\big)\bar{\Delta}^{\xi} \\ &= \bar{\Delta}^{\xi}O^x\Pi^{\xi}\bar{\Delta}^{\xi} - \bar{\Delta}^{\xi}\Pi^{\xi}O^x\bar{\Delta}^{\xi} = \bar{\Delta}^{\xi}[O^x\!,\Pi^{\xi}]\bar{\Delta}^{\xi}\,,\end{aligned}$$

which vanishes for $\xi \neq x$, since then O^x and Π^{ξ} act on different registers and thus commute. For $\xi = x$, its norm is upper bounded by $\|[O^x, \Pi^x]\|$.

We now consider the second term; the third one can be treated the same way by symmetry, and the fourth one vanishes, as will become clear immediately from below. Using (29) and $\bar{\Delta}^{\xi}\Delta^{\xi} = 0$, so that $\bar{\Delta}^{\xi}\Sigma^{\xi} = 0$, we have

$$\bar{\Delta}^{\xi}[O^x, \Sigma^{\xi}]\Delta^{\xi} = \bar{\Delta}^{\xi}\big(O^x\Sigma^{\xi} - \Sigma^{\xi}O^x\big)\Delta^{\xi} = \Sigma^{\xi}O^x\Delta^{\xi} =: N_{\xi}\,. \tag{32}$$

Looking at (30), we want to control the norm of the sum $N := \sum_{\xi} N_{\xi} \otimes X^{\xi}$. To this end, we show that N_{ξ} and $N_{\xi'}$ have orthogonal images and orthogonal support, i.e., $N^{\dagger}_{\xi'}N_{\xi} = 0 = N_{\xi'}N^{\dagger}_{\xi}$, for all $\xi \neq \xi'$. We first observe that if $x \geq \xi$ then O^x commutes with Δ^{ξ}, since they act on different registers then, and thus

$$N_{\xi} = \Sigma^{\xi}O^x\Delta^{\xi} = \Sigma^{\xi}\Delta^{\xi}O^x = \Pi^{\xi}\bar{\Delta}^{\xi}\Delta^{\xi}O^x = 0\,,$$

exploiting once more that $\bar{\Delta}^{\xi}\Delta^{\xi} = 0$. Therefore, we only need to consider $N_{\xi}, N_{\xi'}$ for $\xi, \xi' > x$ (see Figure 4.1 top left), where we may assume $\xi > \xi'$. For the orthogonality of the images, we observe that

$$\Pi^{\xi'}\bar{\Delta}^{\xi} = 0 \tag{33}$$

by definition of $\bar{\Delta}^{\xi}$ as a tensor product with $\bar{\Pi}^{\xi'}$ being one of the components. Therefore,

$$(\Sigma^{\xi'})^{\dagger}\Sigma^{\xi} = \Sigma^{\xi'}\Sigma^{\xi} = \bar{\Delta}^{\xi'}\Pi^{\xi'}\bar{\Delta}^{\xi}\Pi^{\xi} = 0\,,$$

and $N^{\dagger}_{\xi'}N_{\xi} = 0$ follows directly (see also Figure 4.1 top right). For the orthogonality of the supports, we recall that $\bar{\Delta}^{\xi'} \geq \bar{\Delta}^{\xi}$, and thus $\Delta^{\xi'} \leq \Delta^{\xi}$, from which it follows that $\Delta^{\xi}\Delta^{\xi'} = \Delta^{\xi'}$. $N_{\xi'}N^{\dagger}_{\xi} = 0$ then follows by exploiting (33) again (see Figure 4.1 bottom).

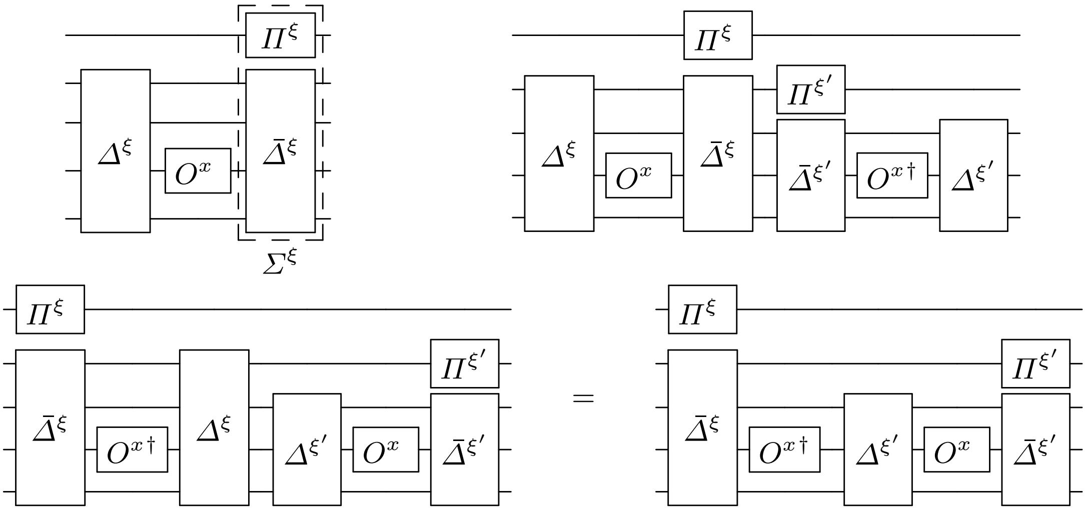

Fig. 4.1. The operators N_ξ (top left), $N^\dagger_{\xi'}N_\xi$ (top right), and $N_{\xi'}N^\dagger_\xi$ (bottom), for $x < \xi' < \xi$.

These orthogonality properties for the images and supports of the N_ξ immediately extend to $N_\xi \otimes X^\xi$, so we have

$$\|N\| \le \max_{\xi > x} \|N_\xi \otimes \mathsf{X}^\xi\| \le \max_{\xi > x} \|N_\xi\|$$

by Lemma 4.1. Recall from (32) that $N_\xi = \bar\Delta^\xi[\Sigma^\xi, O^x]\Delta^\xi$. Furthermore, we exploit that, by definition, Σ^ξ is in tensor-product form and O^x acts trivially on all components in this tensor product except for the component $\bar\Pi^x$, so that $[\Sigma^\xi, O^x] = [\bar\Pi^x, O^x]$ by property (24). Thus,

$$\|N_\xi\| \le \|[\Sigma^\xi, O^x]\| = \|[\bar\Pi^x, O^x]\| = \|[\Pi^x, O^x]\| \, .$$

Using the triangle inequality with respect to the sum versus the last term in (30), and another triangle inequality with respect to the decoimposition (31), we obtain the claimed inequality. □

The proof of Theorem 4.4 is now an easy consequence.

Proof (of Theorem 4.4). Since O_{XYD} is a control unitary $O_{XYD} = \sum_x |x\rangle\langle x| \otimes O^x_{YD_x}$, controlled by $|x\rangle$, while M_{DP} does not act on register X, it follows that

$$\big\|[O_{XYD}, M_{DP}]\big\| \le \max_x \big\|[O^x_{YD_x}, M_{DP}]\big\| \, .$$

The claim of the theorem now follows by combining Lemma 4.7 with Lemma 4.6. □

4.3.3 A First Immediate Application

As an immediate application of the commutator bound of Theorem 4.4, we can easily derive the following generic query-complexity bound for finding x with $(x, H(x)) \in R$ and $\mathit{\Gamma}_R$ as defined in Equation (25).

Proposition 4.8. *For any algorithm $\mathcal{A}$ that makes q queries to the random oracle RO,*

$$\Pr_{x \leftarrow \mathcal{A}^{RO}} \big[\big(x, RO(x)\big) \in R\big] \leq 152(q+1)^2 \mathit{\Gamma}_R / 2^n \,. \tag{34}$$

Proof. Consider the modified algorithm $\mathcal{A}'$ that runs $\mathcal{A}$ to obtain output x, makes a query to obtain $RO(x)$ and outputs $(x, RO(x))$. By Lemma 5 in [Zha19a], we have that[43]

$$\sqrt{\Pr_{x \leftarrow \mathcal{A}'^H} [(x, RO(x)) \in R]} \leq \sqrt{\Pr_{x' \leftarrow G^R} [x' \neq \emptyset]} + 2^{-n/2}, \tag{35}$$

where G^R is the following procedure/game: (1) run $\mathcal{A}'$ using the compressed oracle, and (2) apply the measurement $\mathcal{M}^R$ to obtain $x' \in \mathcal{X} \cup \{\emptyset\}$, which is the same as preparing a register P, applying $M_{DP} = M^R_{DP}$, and measuring P.

In other words, writing $|\psi\rangle_{WXY}$ for the initial state of $\mathcal{A}'$ and V_{WXY} for the unitary applied between any two queries of $\mathcal{A}'$(which we may assume to be fixed without loss of generality), and setting $U_{WXYD} := V_{WXY} O_{XYD}$, $\mathit{\Pi}_P := \mathbb{1}_P - |\emptyset\rangle\langle\emptyset|_P$ and $|\mathit{\Psi}\rangle := |\psi\rangle_{WXY} \otimes |\bot\rangle_D^{\otimes|\mathcal{X}|} \otimes |0\rangle_P$, we have, omitting register subscripts,

$$\begin{aligned}
&\sqrt{\Pr[x' \neq \emptyset]} \\
&\quad = \big\| \mathit{\Pi} M U^{q+1} |\mathit{\Psi}\rangle \big\| \\
&\quad \leq \sum_{i=1}^{q+1} \big\| \mathit{\Pi} U^{i-1} [M, U] U^{q+1-i} |\mathit{\Psi}\rangle \big\| + \big\| \mathit{\Pi} U^{q+1} M |\mathit{\Psi}\rangle \big\| \\
&\quad \leq (q+1) \big\| [M_{DP}, O_{XYD}] \big\| + \big\| \mathit{\Pi}_P M_{DP} |\mathit{\Psi}\rangle \big\| \\
&\quad = (q+1) \big\| [M_{DP}, O_{XYD}] \big\| \leq 8 \cdot 2^{-n/2} (q+1) \sqrt{2\mathit{\Gamma}_R} \,,
\end{aligned}$$

where the last equation exploits that $\mathit{\Pi}_P M_{DP}$ applied to $|\bot\rangle_D^{\otimes|\mathcal{X}|} \otimes |0\rangle_P$ vanishes, and the final inequality is by Theorem 4.4. Observing $(8\sqrt{2}+1)^2 = 129 + 16\sqrt{2} \approx 151.6$ finishes the proof. □

[43] Lemma 5 in [Zha19a] applies to an algorithm $\mathcal{A}$ that outputs both x and what is supposed to be its hash value; this is why we need to do this additional query.

Applied to $R = \mathcal{X} \times \{0^n\}$, where $\Gamma_R = 1$, we recover the famous lower bound for search in a random function. In essence, our commutator bound replaces the "progress-measure" argument in the search-lower-bound proof from [Zha19a].

Corollary 4.9. *For any algorithm $\mathcal{A}$ that makes q queries to the random oracle RO,*

$$\Pr_{x \leftarrow \mathcal{A}^{RO}} [RO(x) = 0^n] \leq 152(q+1)^2/2^n. \tag{36}$$

Section 4.4

Extraction of Random-Oracle Based Commitments

Throughout this Section 4.4, let $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{T}$ be an arbitrary fixed function with $\mathcal{Y} = \{0,1\}^n$. For a hash function $H : \mathcal{X} \to \mathcal{Y}$, which will then be modelled as a random oracle RO, we will think and sometimes speak of $f(x, H(x))$ as a *commitment* of x (though we do not require it to be a commitment scheme in the strict sense). Typical examples are $f(x,y) = y$ and $f(x,y) = \mathsf{Enc}_{pk}(x;y)$, where the latter is the encryption of x under public key pk with randomness y.

4.4.1 Informal Problem Description

Consider a query algorithm $\mathcal{A}^{RO}$ in the random oracle model, which, during the course of its run, announces some $t \in \mathcal{T}$. This t is supposed to be $t = f(x, RO(x))$ for some x, and, indeed, $\mathcal{A}^{RO}$ may possibly reveal x later on, i.e., open the commitment. Intuitively, in order for the required relation between x and t to hold, we expect that $\mathcal{A}^{RO}$ *first* has to query RO on x and only *then* can output t; thus, one may hope to be able to extract x from RO *early on*, i.e., at the time $\mathcal{A}^{RO}$ announces t.

This is clearly true when $\mathcal{A}$ is restricted to classical queries, simply by checking all the queries made so far. This observation was first made and utilized by Pass [Pas03] and only requires looking at the query transcript (it can be done in the *non-programmable* ROM). As the extractor does not change the course of the experiment, it is in particular also suitable in situations where it is necessary to extract an opening on the fly, i.e., while guaranteeing that $\mathcal{A}$ still proceeds to produce its output (e.g. for multiple-committer parallel extraction [ABG+20]).

In the setting considered here, $\mathcal{A}^{RO}$ may query the random oracle in *superposition* over various choices of x, making it impossible to maintain a classical query transcript. On the positive side, since the output t is required to be classical, $\mathcal{A}^{RO}$ has to perform a measurement before announcing t, enforcing such a superposition to collapse.[44] We show here that early extraction of x is indeed possible in this quantum setting as well.

Note that if the goal is to extract *the same* x as $\mathcal{A}$ will (potentially) output, which is what we aim for, then we must naturally assume that it is hard for $\mathcal{A}$ to find $x \neq x'$ that are both consistent with the same t, i.e., we must assume the commitment to be binding. Formally, for the upcoming discussion in this section to be meaningful, we will think of $\Gamma(f)$ and $\Gamma'(f)$, defined as follows, to be small compared to $|\mathcal{Y}| = 2^n$. When f is fixed, we simply write Γ and Γ'.

Definition 4.10. *For $f : \mathcal{X} \times \{0,1\}^n \to \mathcal{T}$, we define*

$$\Gamma(f) := \max_{x,t} |\{y \mid f(x,y) = t\}| \quad \textit{and} \quad \Gamma'(f) := \max_{x \neq x', y'} |\{y \mid f(x,y) = f(x',y')\}| \,.$$

For the example $f(x,y) = y$, we have $\Gamma(f) = 1 = \Gamma'(f)$. For the example $f(x,y) = \mathsf{Enc}_{pk}(x;y)$, they both depend on the choice of the encryption scheme but typically are small, e.g. $\Gamma(f) = 1$ if Enc is injective as a function of the randomness y and $\Gamma'(f) = 0$ if there are no decryption errors.

Remark 4.11. We note that the ratio $\Gamma(f)/2^n$ remains unaffected when n is increased, i.e., if $\tilde{n} \geq n$ and $\tilde{f} : \mathcal{X} \times \{0,1\}^{\tilde{n}} \to \mathcal{T}$ is given by $\tilde{f}(x, y\|y') := f(x,y)$ for all $x \in \mathcal{X}$, $y \in \{0,1\}^n$ and $y' \in \{0,1\}^{\tilde{n}-n}$, then $\Gamma(\tilde{f})/2^{\tilde{n}} = \Gamma(f)/2^n$, because the additional $\tilde{n}-n$ bits of y' do not affect the conditions on $\tilde{f}$ in Definition 4.10, so both numerator and denominator of the fraction get multiplied by $2^{\tilde{n}-n}$. The same holds for $\Gamma'(f)/2^n$.

4.4.2 The Extractable RO-Simulator $\mathcal{S}$

Towards formalizing the above goal, we introduce a simulator $\mathcal{S}$ that replaces RO and tries to extract x early on, right after $\mathcal{A}$ announces t. In more detail, $\mathcal{S}$ acts as a black-box oracle with two interfaces, the *RO-interface* $\mathcal{S}.RO$ providing access to the simulated random oracle, and the *extraction interface* $\mathcal{S}.E$ providing the functionality to extract x early on (see Figure 4.3, left). In principle, both interfaces can be accessed quantumly, i.e., in superposition over different classical inputs, but in our applications we only use classical access to

[44] We can also think of this measurement being done by the interface that receives t.

$\mathcal{S}.E$. We stress that $\mathcal{S}$ is per-se *stateful* and thus may change its behavior from query to query.

Formally, the considered simulator $\mathcal{S}$ is defined to work as follows. It simulates the random oracle and answers queries to $\mathcal{S}.RO$ by means of the compressed oracle. For the $\mathcal{S}.E$ interface, upon a classical input $t \in \mathcal{T}$, $\mathcal{S}$ applies the measurement $\mathcal{M}^t := \mathcal{M}^{R_t}$ from (27) for the relation $R_t := \{(x, y) \,|\, f(x, y) = t\}$ to obtain $\hat{x} \in \mathcal{X} \cup \{\emptyset\}$, which it then outputs (see Figure 4.2). In case of a *quantum* query to $\mathcal{S}.E$, the above is performed coherently: given the query registers TP, the unitary $\sum_t |t\rangle\langle t|_T \otimes M^{R_t}_{DP}$ is applied to TPD, and registers TP are then returned.

The extractable RO-oracle $\mathcal{S}$:

Initialization: $\mathcal{S}$ prepares its internal register D to be in state $|\bot\rangle_D := \bigotimes_x |\bot\rangle_{D_x}$.

$\mathcal{S}$.RO-query: Upon a (quantum) RO-query, with query registers XY, $\mathcal{S}$ applies O_{XYD} to registers XYD.

$\mathcal{S}$.E-query: Upon a classical extraction-query with input t, $\mathcal{S}$ applies $\mathcal{M}^t$ to D and returns the outcome $\hat{x}$.

Fig. 4.2. The (inefficient version of the) simulator $\mathcal{S}$, restricted to classical extraction queries.

We note that, as described here, the simulator $\mathcal{S}$ is inefficient, having to maintain an exponential number of qubits; however, using the sparse representation of the internal state D, as discussed in Section 2.4, $\mathcal{S}$ can well be made efficient without affecting its query-behavior (see Theorem 4.12 for details).

The following statement captures the core properties of $\mathcal{S}$. We refer to two subsequent queries as being *independent* if they can in principle be performed in either order, i.e., if the input to one query does not depend on the output of the other. More formally, e.g., two $\mathcal{S}.RO$ queries are independent if they can be captured by first preparing the two in-/output registers XY and $X'Y'$, and then doing the two respective queries with XY and $X'Y'$. The commutativity claim then means that the order does not matter. Furthermore, whenever we speak of a *classical* query (to $\mathcal{S}.RO$ or to $\mathcal{S}.E$), we consider the obvious classical variant of the considered query, with a classical input and a classical response. Finally, the almost commutativity claims are in terms of the trace distance of the (possibly quantum) output of any algorithm interacting with $\mathcal{S}$ arbitrarily and doing the two considered independent queries in one or the other order.

Theorem 4.12. *The extractable RO-simulator $\mathcal{S}$ constructed above, with interfaces $\mathcal{S}.RO$ and $\mathcal{S}.E$, satisfies the following properties.*

1. *If $\mathcal{S}.E$ is unused, $\mathcal{S}$ is perfectly indistinguishable from the random oracle RO.*

2.a *Any two subsequent independent queries to $\mathcal{S}.RO$ commute. In particular, two subsequent* classical *$\mathcal{S}.RO$-queries with the same input x give identical responses.*

2.b *Any two subsequent independent queries to $\mathcal{S}.E$ commute. In particular, two subsequent* classical *$\mathcal{S}.E$-queries with the same input t give identical responses.*

2.c *Any two subsequent independent queries to $\mathcal{S}.E$ and $\mathcal{S}.RO$ $8\sqrt{2\Gamma(f)/2^n}$-almost-commute.*

3.a *Any classical query $\mathcal{S}.RO(x)$ is idempotent.*[45]

3.b *Any classical query $\mathcal{S}.E(t)$ is idempotent.*

4.a *If $\hat{x} = \mathcal{S}.E(t)$ and $\hat{h} = \mathcal{S}.RO(\hat{x})$ are two subsequent classical queries then*

$$\Pr[f(\hat{x},\hat{h}) \neq t \wedge \hat{x} \neq \emptyset] \leq \Pr[f(\hat{x},\hat{h}) \neq t \mid \hat{x} \neq \emptyset] \leq 2 \cdot 2^{-n}\Gamma(f) \qquad (37)$$

4.b *If $h = \mathcal{S}.RO(x)$ and $\hat{x} = \mathcal{S}.E(f(x,h))$ are two subsequent classical queries such that no prior query to $\mathcal{S}.E$ has been made, then*

$$\Pr[\hat{x} = \emptyset] \leq 2 \cdot 2^{-n}. \qquad (38)$$

Furthermore, the total runtime of $\mathcal{S}$, when implemented using the sparse representation of the compressed oracle described in Section 2.4, is bounded as

$$T_{\mathcal{S}} = O\big(q_{RO} \cdot q_E \cdot \mathrm{Time}[f] + q_{RO}^2\big)\,,$$

where q_E and q_{RO} are the number of queries to $\mathcal{S}.E$ and $\mathcal{S}.RO$, respectively.

Proof. All the properties follow rather directly by construction of $\mathcal{S}$. Indeed, without $\mathcal{S}.E$-queries, $\mathcal{S}$ is simply the compressed oracle, known to be perfectly indistinguishable from the random oracle, confirming 1. Property 2.a follows from the fact that the unitaries O_{XYD} and $O_{X'Y'D}$, acting on the same register D but on distinct query registers, are both controlled unitaries with control register D, conjugated by a fixed unitary ($F^{\otimes|\mathcal{X}|}$). They thus commute. For 2.b, the claim follows from the fact that the unitaries M^t_{DP} and $M^{t'}_{DP'}$ commute, as they are both controlled unitaries with control register D. 2.c is a direct consequence of our main technical result Theorem 4.4 (in the form of

[45] I.e., applying it twice in a row has the same effect on the state of $\mathcal{S}$ as applying it once.

Corollary 4.5). 3.a follows from the fact that a classical $\mathcal{S}.RO$ query with input x acts as a projective measurement on register D_x, which is, as any projective measurement, idempotent. Thus, so is the measurement $\mathcal{M}^t$, confirming 3.b.

To prove 4.a, consider the state $\rho_{D_{\hat{x}}}$ of register $D_{\hat{x}}$ after the measurement $\mathcal{M}^t$ that is performed by the extraction query $\hat{x} = \mathcal{S}.E(t)$, assuming $\hat{x} \neq \emptyset$. Let $|\psi\rangle$ be a purification of $\rho_{D_{\hat{x}}}$. By definition of $\mathcal{M}^t$, it holds that $\Pi^{\hat{x}}_{D_{\hat{x}}}|\psi\rangle = |\psi\rangle$. Then, understanding that all operators act on register $D_{\hat{x}}$, by definition of $\bar{\Pi}^{\hat{x}}$ the probability of interest is bounded as[46]

$$\begin{aligned}\Pr[f(\hat{x},\hat{h}) \neq t \,|\, \hat{x} \neq \emptyset] \leq \big\|\bar{\Pi}^{\hat{x}} F|\psi\rangle\big\|^2 = \big\|\bar{\Pi}^{\hat{x}} F \Pi^{\hat{x}}|\psi\rangle\big\|^2 &\leq \big\|\bar{\Pi}^{\hat{x}} F \Pi^{\hat{x}}\big\|^2 \\ &\leq \big\|[F, \Pi^{\hat{x}}]\big\|^2 ,\end{aligned}$$

where the last inequality exploits that $\bar{\Pi}^{\hat{x}}\Pi^{\hat{x}} = 0$. The claim now follows from Lemma 4.6.

For 4.b, we first observe that, given that there were no prior extraction queries, the state of D_x before the $h = \mathcal{S}.RO(x)$ query has no overlap with $|\hat{0}\rangle$, and thus the state after the query is $F|h\rangle$ (see the discussion above Equation (4)). For the purpose of the argument, instead of applying the measurement $\mathcal{M}^{f(x,h)}$ to answer the $\mathcal{S}.E(f(x,h))$ query, we may equivalently consider a measurement in the basis $\{|\mathbf{y}\rangle\}$, and then set $\hat{x}$ to be the smallest element $\mathcal{X}$ so that $f(\hat{x}, y_{\hat{x}}) = t := f(x,h)$, with $\hat{x} = \emptyset$ if no such element exists. Then,

$$\begin{aligned}\Pr[\hat{x} \neq \emptyset] = \Pr[\exists\, \xi : f(\xi, y_\xi) = t] \geq \Pr[f(x, y_x) = t] &\geq \Pr[y_x = h] \\ &= |\langle h|F|h\rangle|^2 \geq 1 - 2 \cdot 2^{-n}\end{aligned}$$

where the last two (in)equalities are by Equation (4).

□

4.4.3 Two More Properties of $\mathcal{S}$

On top of the above basic features of our extractable RO-simulator $\mathcal{S}$, we show the following two additional, more technical, properties, which in essence capture that the extraction interface cannot be used to bypass query hardness results.

The first property is easiest to understand in the context of the example $f(x,y) = y$, where $\mathcal{S}.E(t)$ tries to extract a hash-preimage of t, and where

[46] The first inequality is an artefact of the $|\bot\rangle\langle\bot|$-term in $\bar{\Pi}^{\hat{x}}$ contributing to the probability of $\hat{h} = 0$, as discussed in Section 2.4.

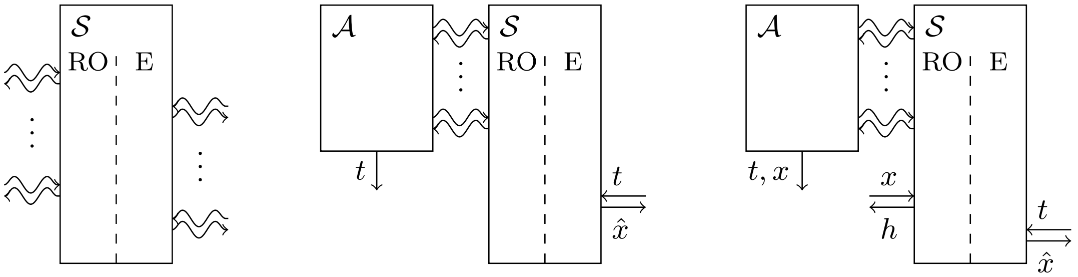

Fig. 4.3. The extractable RO-simulator $\mathcal{S}$, with its $\mathcal{S}.RO$ and $\mathcal{S}.E$ interfaces, distinguished here by queries from the left and right (left), and the games considered in Proposition 4.13 (middle) and 4.14 (right) for $\ell = 1$. Waved arrows denote quantum queries, straight arrows denote classical queries.

the relations R and R' in Proposition 4.13 below then coincide. In this case, recall from Proposition 4.8 that, informally, if Γ_R is small then it is hard to find $x \in \mathcal{X}$ so that $t := RO(x)$ satisfies $(x,t) \in R$. The statement below ensures that this hardness cannot be bypassed by first selecting a "good" hash value t and then trying to extract a preimage by means of $\mathcal{S}.E$ (Figure 4.3, middle). For instance, setting $t := t_\circ$ for a given target $t_\circ$ and extracting $\hat{x} := \mathcal{S}.E(t)$, we cannot hope for $\hat{x}$ to satisfy $\mathcal{S}.RO(\hat{x}) = t$; unless there was a prior query to $\mathcal{S}.RO$ with response $t_\circ$, the extraction will provide $\hat{x} = \emptyset$ most likely.

Proposition 4.13. *Let $R' \subseteq \mathcal{X} \times \mathcal{T}$ be a relation. Consider a query algorithm $\mathcal{A}$ that makes q queries to the $\mathcal{S}.RO$ interface of $\mathcal{S}$ but no query to $\mathcal{S}.E$, outputting some $\mathbf{t} \in \mathcal{T}^\ell$. For each i, let $\hat{x}_i$ then be obtained by making an additional query to $\mathcal{S}.E$ on input t_i (see Figure 4.3, middle). Then*

$$\Pr_{\substack{\mathbf{t} \leftarrow \mathcal{A}^{\mathcal{S}.RO} \\ \hat{x}_i \leftarrow \mathcal{S}.E(t_i)}} [\exists\, i : (\hat{x}_i, t_i) \in R'] \leq 128 \cdot q^2 \Gamma_R / 2^n \,,$$

where $R \subseteq \mathcal{X} \times \mathcal{Y}$ is the relation $(x,y) \in R \Leftrightarrow (x, f(x,y)) \in R'$ and Γ_R as in (25).

Proof. The considered experiment is like the experiment G^R in the proof of Proposition 4.8, the only difference being that in G^R the measurement $\mathcal{M}^R$ is applied to register D to obtain x' (see Figure 4.4*b*), while here we have ℓ measurements $\mathcal{M}^{t_i}$ that are applied to obtain $\hat{x}_i$ (see Figure 4.4*a*). Since all measurements are defined by means of projections that are diagonal in the same basis $\{|\mathbf{y}\rangle\}$ with $|\mathbf{y}\rangle$ ranging over $\mathbf{y} \in (\mathcal{Y} \cup \{\perp\})^{\mathcal{X}}$, we may equivalently measure D in that basis to obtain $\mathbf{y}$ (see Figure 4.4*c*), and let $\hat{x}_i$ be minimal so that $f(\hat{x}_i, y_{\hat{x}_i}) = t_i$ (and $\hat{x}_i = \emptyset$ if no such value exists), and let x' be minimal so that $(x', y_{x'}) \in R$ (and $x' = \emptyset$ if no such value exists). By the respective definitions of $\mathcal{M}_i^t$ and $\mathcal{M}^R$, both pairs of random variables $(\hat{\mathbf{x}}, \mathbf{t})$ and $(x', \mathbf{t})$ then have the same distributions as in the respective original two games. But now, we can consider their joint distribution and argue that

$$\begin{aligned}\Pr[\exists\, i : (\hat{x}_i, t_i) \in R'] &= \Pr[\exists\, i : (\hat{x}_i, f(\hat{x}_i, y_{\hat{x}_i})) \in R'] \\ &= \Pr[\exists\, i : (\hat{x}_i, y_{\hat{x}_i}) \in R] \leq \Pr[\exists\, x : (x, y_x) \in R] = \Pr[x' \neq \emptyset]\,.\end{aligned}$$

The bound on $\Pr[x' \neq \emptyset]$ from the proof of Proposition 4.8 concludes the proof. □

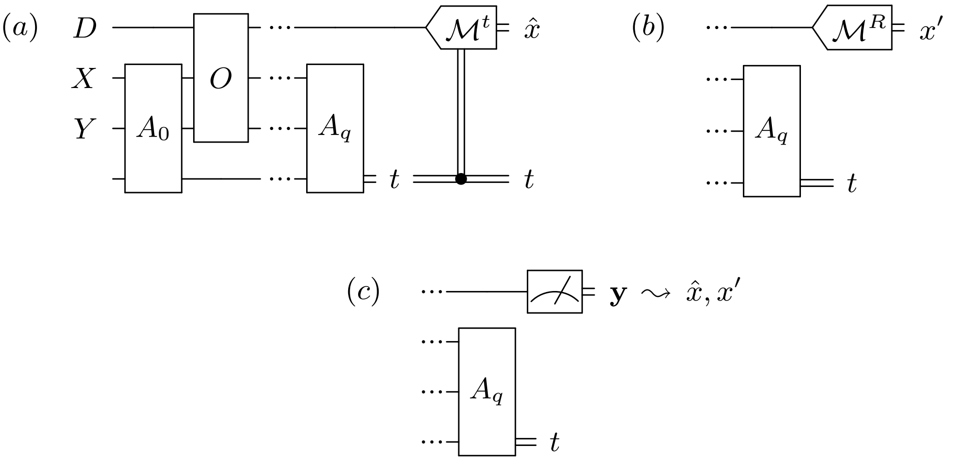

Fig. 4.4. Quantum circuit diagrams for the experiments in the proof of Proposition 4.13 for the case $\ell = 1$.

In a somewhat similar spirit, the following ensures that if it is hard in the QROM to find x and x' with $f(x, RO(x)) = f(x', RO(x'))$ then this hardness cannot be bypassed by, say, first choosing x, querying $h = \mathcal{S}.RO(x)$, computing $t := f(x, h)$, and then extracting $\hat{x} := \mathcal{S}.E(t)$. The latter will most likely give $\hat{x} = x$, except, intuitively, if $\mathcal{S}.RO$ has additionally been queried on a colliding x'.

Proposition 4.14. *Consider a query algorithm $\mathcal{A}$ that makes q queries to $\mathcal{S}.RO$ but no query to $\mathcal{S}.E$, outputting some $t \in \mathcal{T}$ and $x \in \mathcal{X}$. Let h then be obtained by making an additional query to $\mathcal{S}.RO$ on input x, and $\hat{x}$ by mak-*

ing an additional query to $\mathcal{S}.E$ on input t (see Figure 4.3, right). Then

$$\Pr_{\substack{t,x \leftarrow \mathcal{A}^{\mathcal{S}.RO} \\ h \leftarrow \mathcal{S}.RO(x) \\ \hat{x} \leftarrow \mathcal{S}.E(t)}} [\hat{x} \neq x \wedge f(x,h) = t] \leq \frac{40e^2(q+2)^3\Gamma'(f)+2}{2^n} .$$

More generally, if $\mathcal{A}$ outputs ℓ-tuples $\mathbf{t} \in \mathcal{T}^\ell$ and $\mathbf{x} \in \mathcal{X}^\ell$, and $\mathbf{h} \in \mathcal{Y}^\ell$ is obtained by querying $\mathcal{S}.RO$ component-wise on $\mathbf{x}$, and $\hat{\mathbf{x}} \in (\mathcal{X} \cup \{\emptyset\})^\ell$ by querying $\mathcal{S}.E$ component-wise on $\mathbf{t}$, then

$$\Pr_{\substack{\mathbf{t},\mathbf{x} \leftarrow \mathcal{A}^{\mathcal{S}.RO} \\ \mathbf{h} \leftarrow \mathcal{S}.RO(\mathbf{x}) \\ \hat{\mathbf{x}} \leftarrow \mathcal{S}.E(\mathbf{t})}} [\exists\, i : \hat{x}_i \neq x_i \wedge f(x_i,h_i) = t] \leq \frac{40e^2(q+\ell+1)^3\Gamma'(f)+2}{2^n} .$$

The proof is similar in spirit to the proof of Proposition 4.13, but relying on the hardness of collision finding (Lemma 4.15) rather than on (the proof of) Proposition 4.8. The following can be easily extracted from the derivation of the general collision-finding bound Theorem 5.29 from [CFHL21]. It expresses that, for any algorithm with bounded query complexity, it is unlikely that one encounters a collision within the superposition oracle.

Lemma 4.15. *Let $f : \mathcal{X} \times \{0,1\}^n \to \mathcal{T}$, and let Π^{col} be the projection into the space spanned by $|\mathbf{y}\rangle \in \mathcal{H}_D$ for $\mathbf{y} = (y_x)_{x\in\mathcal{X}} \in (\mathcal{Y} \cup \{\perp\})^{\mathcal{X}}$ such that there exist $x \neq x'$ with $y_x, y_{x'} \neq \perp$ and $f(x,y_x) = f(x',y_{x'})$. Then, for any oracle algorithm $\mathcal{A}$ with query complexity q, at the end of the execution the state ρ of the compressed oracle is such that*

$$\mathrm{tr}(\Pi^{col}\rho) \leq 40e^2q^2(q+1)\Gamma'(f)/2^n ,$$

where $\Gamma'(f) = \max\limits_{x\neq x',y'} |\{y \mid f(x,y) = f(x',y')\}|$ and $e \approx 2.718$ is Euler's number.

Proof (of Proposition 4.14). Circuit (a) in Figure 4.5 defines (the distribution of) the considered variables $x, \hat{x}, h, t$. We also consider the circuit that applies the measurement $\{\Pi^{col}, \Pi^{\neg col}\}$ instead of $\mathcal{M}^t$, where Π^{col} is as in Lemma 4.15 and $\Pi^{\neg col} = \mathbb{1} - \Pi^{col}$ (Figure 4.5b). Since the projections defining either measurement are all diagonal in the basis $\{|\mathbf{y}\rangle\}$, we may equivalently measure register D in that basis (Figure 4.5c), and then set $\hat{x}$ to be the smallest element $\mathcal{X}$ so that $f(\hat{x}, y_{\hat{x}}) = t$ (with $\hat{x} = \emptyset$ if no such element exists) and consider the event *col* given by $\exists\, x' \neq x'' : f(x', y_{x'}) = f(x'', y_{x''})$. By the respective definitions of $\mathcal{M}^t$ and Π^{col}, both, the variables $\hat{x}, x, h, t$ and the event and variable *col* and x, h, t then have the same distributions as in the respective original two games. But now, we can consider their joint distribution and argue that

$$\Pr[\hat{x} \neq x \wedge f(x,h) = t] \leq \Pr[\hat{x} \neq x \mid f(x,h) = t \wedge \neg col] + \Pr[col]\,.$$

We now observe that right before the considered measurement, by definition of O, the state of D is supported by vectors $F|\mathbf{y}\rangle$ with $y_x = h$ (here we use the assumption that no previous extraction queries have been made, see Preliminaries for further detail), and so the measurement outcome $\mathbf{y}$ satisfies $y_x = h$ with probability $1 - 2 \cdot 2^{-n}$ by Equation (4). Therefore, the first term is bounded by $2 \cdot 2^{-n}$ by definition of col and $\hat{x}$, while $\Pr[col]$ is bounded by $\frac{40e^2(q+2)^3\Gamma'(f)+2}{2^n}$, using Lemma 4.15. □

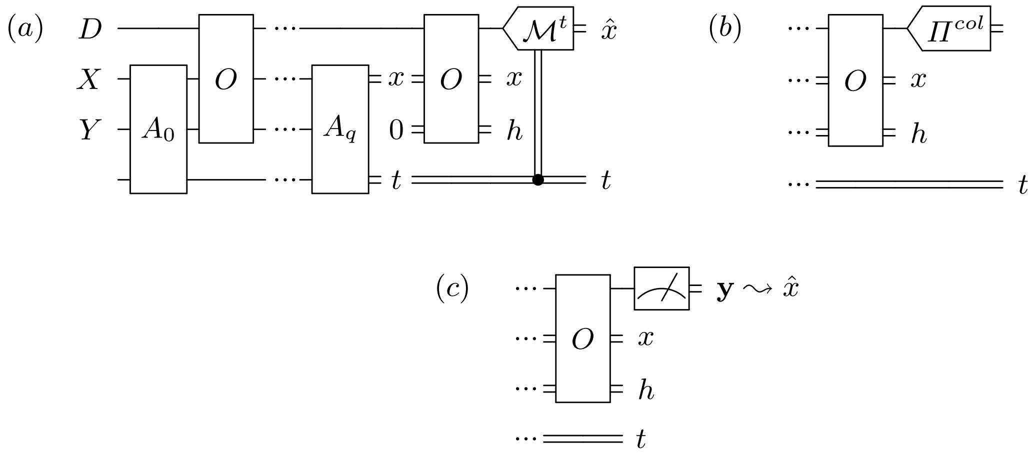

Fig. 4.5. Quantum circuit diagrams for the experiments in the proof of Proposition 4.14.

Remark 4.16. The claim of Proposition 4.14 stays true when the queries $\mathcal{S}.RO(x_i)$ are not performed as *additional* queries *after* the run of $\mathcal{A}$ but are explicitly *among* the q queries that are performed by $\mathcal{A}$ *during* its run. One way to see this is to use 2.a and 3.a of Theorem 4.12 to re-do these queries once more after the run of $\mathcal{A}$, which does not affect the subsequent $\mathcal{S}.E$-queries. Alternatively, we observe that the proof does not exploit that these queries are performed at the end, which additionally shows that in this case the ℓ-term on the right hand side of the bound vanishes, i.e., scales as $(q+1)^3$ rather than as $(q+\ell+1)^3$.

4.4.4 Early Extraction

We consider here the following concrete setting. Let $\mathcal{A}$ be a two-round query algorithm, interacting with the random oracle RO and behaving as follows. At the end of the first round, $\mathcal{A}^{RO}$ outputs some $t \in \mathcal{T}$, and at the end of the second round, it outputs some $x \in \mathcal{X}$ that is supposed to satisfy $f(x, RO(x)) = t$; on top, $\mathcal{A}^{RO}$ may have some additional (possibly quantum) output W (see Figure 4.6, left).

We now show how the extractable RO-simulator $\mathcal{S}$ provides the means to extract x early on, i.e., right after $\mathcal{A}$ has announced t. To formalize this claim, we consider the following experiment, which we denote by $G_{\mathcal{S}}^{\mathcal{A}}$. The RO-interface $\mathcal{S}.RO$ of $\mathcal{S}$ is used to answer all the oracle queries made by $\mathcal{A}$. In addition, as soon as $\mathcal{A}$ outputs t, the interface $\mathcal{S}.E$ is queried on t to obtain $\hat{x} \in \mathcal{X} \cup \{\emptyset\}$, and after $\mathcal{A}$ has finished, $\mathcal{S}.RO$ is queried on $\mathcal{A}$'s final output x to generate h; see Figure 4.6 (right).

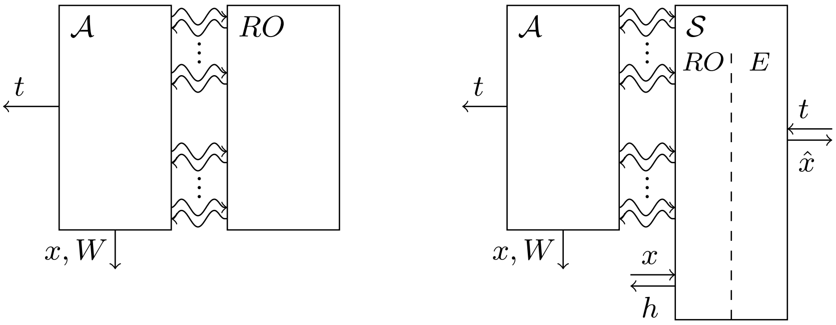

Fig. 4.6. The original execution of $\mathcal{A}^{RO}$ (left), and the experiment $G_{\mathcal{S}}^{\mathcal{A}}$ with RO simulated by $\mathcal{S}$ (right).

Informally, we want that $\mathcal{A}$ does not notice any difference when RO is replaced by $\mathcal{S}.RO$, and that $\hat{x} = x$ whenever $f\big(x, h\big) = t$, while $\hat{x} = \emptyset$ implies that $\mathcal{A}$ will fail to output x with $f\big(x, h\big) = t$. This situation is captured by the following statement.

Corollary 4.17. *The extractable RO-simulator $\mathcal{S}$ is such that the following holds. For any $\mathcal{A}$ that outputs t after q_1 queries and $x \in \mathcal{X}$ and W after an additional q_2 queries, it holds that*

$$\delta\big([t, x, RO(x), W]_{\mathcal{A}^{RO}}, [t, x, h, W]_{G_{\mathcal{S}}^{\mathcal{A}}}\big) \le 8(q_2 + 1)\sqrt{2\Gamma/2^n} \qquad \textit{and}$$

$$\Pr_{G_{\mathcal{S}}^{\mathcal{A}}}\big[x \neq \hat{x} \wedge f(x,h) = t\big] \leq 8(q_2+1)\sqrt{2\Gamma/2^n} + \frac{40e^2(q+2)^3\Gamma'(f)+2}{2^n}\,,$$

where $q = q_1 + q_2$.

Proof. The first claim follows from the fact that the trace distance vanishes when $\mathcal{S}.E(t)$ is performed at the very end, after the $\mathcal{S}.RO(x)$-query, in combination with the (almost-)commutativity of the two interfaces (Theorem 4.12, 2.a to 2.c). Similarly, the second claim follows from Proposition 4.14 when considering the $\mathcal{S}.E(t)$ query to be performed at the very end, in combination with the (almost-)commutativity of the interfaces again. □

The statements above extend easily to *multi*-round algorithms $\mathcal{A}^{RO}$ that output $t_1,\ldots,t_\ell$ in (possibly) different rounds, and $x_1,\ldots,x_\ell \in \mathcal{X}$ and some (possibly quantum) output W at the end of the run. We then extend the definition of $G_{\mathcal{S}}^{\mathcal{A}}$ in the obvious way: $\mathcal{S}.E$ is queried on each output t_i to produce $\hat{x}_i$, and at the end of the run $\mathcal{S}.RO$ is queried on each of the final outputs $x_1,\ldots,x_\ell$ of $\mathcal{A}$ to obtain $\mathbf{h} = (h_1,\ldots,h_\ell) \in \mathcal{Y}^\ell$. As a minor extension, we allow some of the x_i to be $\perp$, i.e., $\mathcal{A}^{RO}$ may decide to not output certain x_i's; the $\mathcal{S}.RO$ query on x_i is then not done and h_i is set to $\perp$ instead, and we declare that $RO(\perp) = \perp$ and $f(\perp, h_i) \neq t_i$. To allow for a compact notation, we write $RO(\mathbf{x}) = (RO(x_1),\ldots,RO(x_\ell))$ for $\mathbf{x} = (x_1,\ldots,x_\ell)$.

Corollary 4.18. *The extractable RO-simulator $\mathcal{S}$ is such that the following holds. For any $\mathcal{A}$ that makes q queries in total, it holds that*

$$\delta\big([\mathbf{t},\mathbf{x},RO(\mathbf{x}),W]_{\mathcal{A}^{RO}}, [\mathbf{t},\mathbf{x},\mathbf{h},W]_{G_{\mathcal{S}}^{\mathcal{A}}}\big) \leq 8\ell(q+\ell)\sqrt{2\Gamma/2^n} \qquad \textit{and}$$

$$\begin{aligned}\Pr_{G_{\mathcal{S}}^{\mathcal{A}}}&\big[\exists\, i : x_i \neq \hat{x}_i \wedge f(x_i,h_i) = t_i\big] \\ &\leq 8\ell(q+1)\sqrt{2\Gamma/2^n} + \frac{40e^2(q+\ell+1)^3\Gamma'(f)+2}{2^n}.\end{aligned}$$

Proof. The first claim follows from the fact that the trace distance vanishes when the $\mathcal{S}.E(t_i)$-queries are performed at the very end, after all $\mathcal{S}.RO(x_i)$-queries, in combination with the (almost-) commutativity of the interfaces. Similarly, the second claim follows from (the more general second part of) Proposition 4.14 when considering the $\mathcal{S}.E(t_i)$-queries to be performed at the very end, in combination with the (almost-)commutativity of the interfaces again. □

Section 4.5

Extractability of Commit-And-Open Σ-protocols

4.5.1 Commit-and-Open Σ-protocols

We refer the reader to Section 2.2 for the concept of an interactive proof system for a language $\mathcal{L}$ or a relation R, and specifically the notion of a commit-and-open Σ-protocol.

Commit-and-open Σ-protocols are (classically) extractable in a straightforward manner as soon as a witness can be computed from sufficiently many of the x_i's: rewind the prover a few times until it has opened every commitment a_i at least once.[47] There is, however, an alternative (classical) *online* extractor if the hash function H is modelled as a random oracle: simply look at the query transcript of the prover to find preimages of the commitments $a_1, ..., a_\ell$. As the challenge is chosen independently, the extractability and collision resistance of the commitments implies that for a prover with a high success probability, the ℓ extractions succeed simultaneously with good probability. This is roughly how the proof of online extractability of the ZK proof system for graph 3-coloring by Goldreich, Micali and Wigderson [GMW91], instantiated with random-oracle based commitments, works that was announced in [Pas03] and shown in [Pas04] (Proposition 5).

Equipped with our extractable RO-simulator $\mathcal{S}$, we can mimmic the above in the quantum setting. Indeed, the only change is that the look-ups in the transcript are replaced with the additional interface of the simulator $\mathcal{S}$. Corollary 4.18 can then be used to prove the success of extraction using essentially the same extractor as in the classical case.

4.5.2 Notions of Special Soundness

The property that allows such an extraction is most conveniently expressed in terms of special soundness and its variants. Because there are, next to special and k-soundness, a number of additional variants in the literature (e.g. in the context of Picnic2/Picnic3 [CDG+20; KZ20] or MQDSS [CHR+16]), we begin by formulating a generalized notion of special soundness that captures in a broad sense that a witness can be computed from correct responses to

[47] Naturally, we can assume $[\ell] = \bigcup_{c \in C} c$

"*sufficiently many*" challenges.[48] While the notions introduced below can be formulated for arbitrary public-coin interactive proof systems, we first present them tailored to our use-case of commit-and-open Σ-protocols. For completeness we then also include a variant for arbitrary Σ-protocols.

In the remainder, $\mathit{\Pi}$ is thus assumed to be an arbitrary commit-and-open Σ-protocol for a relation R with associated language $\mathcal{L}$, and C is the challenge space of $\mathit{\Pi}$. Furthermore, we consider a non-empty, monotone increasing set $\mathfrak{S}$ of subsets $S \subseteq C$, i.e., such that $S \in \mathfrak{S} \wedge S \subseteq S' \Rightarrow S' \in \mathfrak{S}$, and we let $\mathfrak{S}_{\min} := \{S \in \mathfrak{S} \,|\, S_\circ \subsetneq S \Rightarrow S_\circ \notin \mathfrak{S}\}$ consist of the minimal sets in $\mathfrak{S}$.

Definition 4.19. *$\mathit{\Pi}$ is called* $\mathfrak{S}$-sound *if there exists an efficient algorithm $\mathcal{E}_{\mathfrak{S}}(I, x_1, \ldots, x_\ell, S)$ that takes as input an instance $I \in \mathcal{L}$, strings $x_1, \ldots, x_\ell \in \mathcal{X}$ and a set $S \in \mathfrak{S}_{\min}$, and outputs a witness for I whenever $V(c, (x_i)_{i \in c}) = 1$ for all $c \in S$, and outputs $\perp$ otherwise.*[49]

Note that there is no correctness requirement on the x_i's with $i \notin \bigcup_{c \in S} c$; thus, those x_i's may just as well be set to be empty strings.

This property generalizes k-soundness, which is recovered for $\mathfrak{S} = \mathfrak{T}_k := \{S \subseteq C \,|\, |S| \geq k\}$, but it also captures more general notions. For instance, the r-fold parallel repetition of a k-sound protocol is not k-sound anymore, but it is $\mathfrak{T}_k^{\vee r}$-sound with $\mathfrak{T}_k^{\vee r}$ consisting of those subsets of challenge-sequences $(c_1, \ldots, c_r) \in C^r$ for which the restriction to at least one of the positions is a set in $\mathfrak{T}_k$. This obviously generalizes to the parallel repetition of an arbitrary $\mathfrak{S}$-sound protocol, with the parallel repetition then being $\mathfrak{S}^{\vee r}$-sound with

$$\mathfrak{S}^{\vee r} := \{S \subseteq C^r \,|\, \exists\, i : S_i \in \mathfrak{S}\},$$

where $S_i := \{c \in C \,|\, \exists\, (c_1, ..., c_r) \in S : c_i = c\}$ is the i-th *marginal* of S.

For our result to apply, we need a strengthening of the above soundness condition where $\mathcal{E}_{\mathfrak{S}}$ has to find the set S himself. This is clearly the case for $\mathfrak{S}$-sound protocols that have a *constant sized* challenge space C, but also for the parallel repetition of $\mathfrak{S}$-sound protocols with a constant sized challenge space. Formally, we require the following strengthened notion of $\mathfrak{S}$-sound protocols.

Definition 4.20. *$\mathit{\Pi}$ is called* $\mathfrak{S}$-sound* *if there exists an efficient algorithm $\mathcal{E}^*_{\mathfrak{S}}(I, x_1, \ldots, x_\ell)$ that takes as input an instance $I \in \mathcal{L}$ and strings $x_1, \ldots, x_\ell \in$*

[48] Using the language from secret sharing, we consider an arbitrary access structure $\mathfrak{S}$, while the k-soundness case corresponds to a threshold access structure.

[49] The restriction for S to be in $\mathfrak{S}_{\min}$, rather than in $\mathfrak{S}$, is only to avoid an exponentially sized input while asking $\mathcal{E}_{\mathfrak{S}}$ to be efficient. When C is constant in size, we may admit any $S \in \mathfrak{S}$.

$\mathcal{X}$, and outputs a witness for I whenever there exists $S \in \mathfrak{S}$ with $V(c,(x_i)_{i\in c}) = 1$ for all $c \in S$, and outputs $\bot$ otherwise.

$\mathfrak{S}$-sound Σ-protocols may — and often do — have the property that a dishonest prover can pick any set $\hat{S} = \{\hat{c}_1, \ldots, \hat{c}_m\} \notin \mathfrak{S}$ of challenges $\hat{c}_i \in C$ and then prepare $\hat{x}_1, \ldots, \hat{x}_\ell$ in such a way that $V(c,(\hat{x}_i)_{i\in c}) = 1$ if $c \in \hat{S}$, i.e., after having committed to $\hat{x}_1, \ldots, \hat{x}_\ell$ the prover can successfully answer challenge c if $c \in \hat{S}$. We call this a *trivial* attack. The following captures the largest success probability of such a trivial attack, maximized over the choice of $\hat{S}$:

$$p_{triv}^{\mathfrak{S}} := \frac{1}{|C|} \max_{\hat{S} \notin \mathfrak{S}} |\hat{S}| \,. \tag{39}$$

When there is no danger of confusion, we omit the superscript $\mathfrak{S}$. Looking ahead, our result will show that for any prover that does better than the trivial attack by a non-negligible amount, online extraction is possible. For special sound Σ-protocols, $p_{triv} = 1/|C|$, and for k-sound Σ-protocols, $p_{triv} = (k-1)/|C|$. Furthermore, our definition of $\mathfrak{S}$-soundness allows a straightforward parallel repetition lemma on the combinatorial level providing an expression for p_{triv} of parallel-repeated Σ-protocols.

Lemma 4.21. *Let Π be an $\mathfrak{S}$-sound Σ-protocol. Then $p_{triv}^{\mathfrak{S}^{\vee r}} = \big(p_{triv}^{\mathfrak{S}}\big)^r$.*

Proof. To prove the lemma, we simplify

$$p_{triv}^{\mathfrak{S}^{\vee r}} = \frac{1}{|C|^r} \max_{\hat{S} \notin \mathfrak{S}^{\vee r}} |\hat{S}| = \frac{1}{|C|^r} \max_{\substack{\hat{S} \subset C^r: \\ \forall i: \hat{S}_i \notin \mathfrak{S}}} |\hat{S}| = \frac{1}{|C|^r} \left(\max_{\hat{S} \notin \mathfrak{S}} |\hat{S}| \right)^r = \Big(p_{triv}^{\mathfrak{S}}\Big)^r .$$

□

$\mathfrak{S}$-soundness for general Σ-protocols Let $\mathfrak{S}$ be a non-empty, monotone increasing set of subsets $S \subseteq \mathcal{C}$, and $\mathfrak{S}_{\min} := \{S \in \mathfrak{S} \,|\, S_\circ \subsetneq S \Rightarrow S_\circ \notin \mathfrak{S}\}$.

Definition 4.22. *A Σ-protocol Σ is called* $\mathfrak{S}$-sound *if there exists an efficient deterministic algorithm $\mathcal{E}_{\mathfrak{S}}(x, a, S, \{z_c\}_{c\in S})$ that takes as input an instance x, a first message a, a subset $S \subseteq \mathcal{C}$ of challenges, and responses z_c for $c \in S$, and it outputs a witness for x if $S \in \mathfrak{S}_{\min}$ and $\mathcal{V}(x, a, c, z_c)$ for all $c \in S$.*

The common notion of a *special-sound* Σ-protocol is then the special case of a $\mathfrak{S}$-sound Σ-protocol with $\mathfrak{S} := \{S \subseteq \mathcal{C} \,|\, |S| \geq 2\}$, and similarly a *$k$-sound* ($t$-sound in Chapter 3) Σ-protocol is a $\mathfrak{S}$-sound Σ-protocol with $\mathfrak{S} := \{S \subseteq \mathcal{C} \,|\, |S| \geq k\}$. The quantity

$$p_{triv}^{\mathfrak{S}} := \frac{1}{|\mathcal{C}|} \max_{\hat{S} \notin \mathfrak{S}} |\hat{S}|$$

then again captures the "trivial" attack that may potentially (and typically does) apply to a $\mathfrak{S}$-sound Σ-protocol, where the dishonest prover prepares a first message a so that he has valid responses z ready for all the challenges c in some arbitrarily chosen set $\hat{S} \notin \mathfrak{S}$.

Remark 4.23. We note that Wikström [Wik18] also considers a general notion of special soundness (but then for multi-round protocols); however, the notion in [Wik18] is more restrictive in that it requires some matroid structure on top. For instance, the r-fold parallel repetition of a k-sound protocol does not fit into the formalism by Wikström.

4.5.3 Online Extractability in the QROM

We are now ready to define our extractor and prove that it succeeds. Equipped with the results from the previous section, the intuition is very simple. Given a (possibly dishonest) prover $\mathcal{P}$, running the considered Σ-protocol in the QROM, we use the simulator $\mathcal{S}$ to answer $\mathcal{P}$'s queries to the random oracle but also to extract the commitments $a_1, \ldots, a_\ell$, and if the extracted $\hat{x}_1, \ldots, \hat{x}_\ell$ satisfy the verification predicate V for sufficiently many challenges, we can compute a witness by applying $\mathcal{E}^*_{\mathfrak{S}}$.

The following relates the success probability of this extraction procedure to the success probability of the (possibly dishonest) prover.

Theorem 4.24. *Let Π be an $\mathfrak{S}$-sound* commit-and-open Σ-protocol where the first message consists of ℓ commitments. Then it admits an online extractor $\mathcal{E}$ in the QROM that succeeds with probability*

$$\Pr[\mathcal{E} \text{ succeeds}] \geq \frac{1}{1-p_{triv}}\big(\Pr[\mathcal{P}^{RO} \text{ succeeds}] - p_{triv} - \varepsilon\big) \qquad \textit{where}$$

$$\varepsilon = 8\sqrt{2}\,\ell(2q+\ell+1)/\sqrt{2^n} + \frac{40e^2(q+\ell+1)^3\Gamma'(f)+2}{2^n}$$

and p_{triv} is defined in Equation (47). For $q \geq \ell+1$, the bound simplifies to

$$\varepsilon \leq 34\ell q/\sqrt{2^n} + 2365q^3/2^n\,.$$

Furthermore, the running time of $\mathcal{E}$ is bounded as $T_{\mathcal{E}} = T_{\mathcal{P}_1} + T_{\mathcal{E}^*_{\mathfrak{S}}} + O(q_1^2)$, where $T_{\mathcal{P}_1}$ and $T_{\mathcal{E}^*_{\mathfrak{S}}}$ are the respective runtimes of $\mathcal{P}_1$ and $\mathcal{E}^*_{\mathfrak{S}}$.

Recall that $p_{triv} = (k-1)/|C|$ for k-soundness, giving a corresponding bound.

Proof. We begin by describing the extractor $\mathcal{E}$. In a first step, using $\mathcal{S}.RO$ to answer $\mathcal{P}$'s queries, $\mathcal{E}$ runs the prover $\mathcal{P}$ until it announces $a_1, \ldots, a_\ell$, and then it uses $\mathcal{S}.E$ to extract $\hat{x}_1, ..., \hat{x}_\ell$. I.e., $\mathcal{E}$ acts as $\mathcal{S}$ in Corollary 4.18 for the function $f(x, h) = h$ and runs the game $G^{\mathcal{P}}_{\mathcal{S}}$ to the point where $\mathcal{S}.E$ outputs $\hat{x}_1, ..., \hat{x}_\ell$ on input $a_1, \ldots, a_\ell$. As a matter of fact, for the purpose of the analysis, we assume that $G^{\mathcal{P}}_{\mathcal{S}}$ is run until the end, with the challenge c chosen uniformly at random, and where $\mathcal{P}$ then outputs x_i for all $i \in c$ (and $\perp$ for $i \notin c$) at the end of $G^{\mathcal{P}}_{\mathcal{S}}$; we also declare that $\mathcal{P}$ additionally outputs c and $a_1, \ldots, a_\ell$ at the end. Then, upon having obtained $\hat{x}_1, ..., \hat{x}_\ell$, the extractor $\mathcal{E}$ runs $\mathcal{E}^*_{\mathfrak{S}}$ on $\hat{x}_1, ..., \hat{x}_\ell$ to try to compute a witness. By definition, this succeeds if $\hat{S} := \{\hat{c} \in C \,|\, V(\hat{c}, (\hat{x}_i)_{i\in\hat{c}}) = 1\}$ is in $\mathfrak{S}$.

It remains to relate the success probability of $\mathcal{E}$ to that of the prover $\mathcal{P}^{RO}$. By the first statement of Corollary 4.18, writing $\mathbf{x}_c = (x_i)_{i\in c}$, $RO(\mathbf{x}_c) = (RO(x_i))_{i\in c}$, $\mathbf{a}_c = (a_i)_{i\in c}$, etc., we have

$$\begin{aligned}\Pr[\mathcal{P}^{RO} \text{ succeeds}] &= \Pr_{\mathcal{P}^{RO}}[V(c, \mathbf{x}_c) = 1 \wedge RO(\mathbf{x}_c) = \mathbf{a}_c] \\ &\le \Pr_{G^{\mathcal{P}}_{\mathcal{S}}}[V(c, \mathbf{x}_c) = 1 \wedge \mathbf{h}_c = \mathbf{a}_c] + \delta_1\end{aligned} \tag{40}$$

with $\delta_1 = 8\sqrt{2}\,\ell(q + \ell)/\sqrt{2^n}$. Omitting the subscript $G^{\mathcal{P}}_{\mathcal{S}}$ now,

$$\begin{aligned}&\Pr[V(c, \mathbf{x}_c) = 1 \wedge \mathbf{h}_c = \mathbf{a}_c] \\ &\quad \le \Pr[V(c, \mathbf{x}_c) = 1 \wedge \mathbf{h}_c = \mathbf{a}_c \wedge \mathbf{x}_c = \hat{\mathbf{x}}_c] + \Pr[\mathbf{h}_c = \mathbf{a}_c \wedge \mathbf{x}_c \neq \hat{\mathbf{x}}_c] \\ &\quad \le \Pr[V(c, \hat{\mathbf{x}}_c) = 1] + \Pr[\exists\, j \in c : x_j \neq \hat{x}_j \wedge h_j = a_j] \\ &\quad \le \Pr[V(c, \hat{\mathbf{x}}_c) = 1] + \delta_2\end{aligned} \tag{41}$$

with $\delta_2 = 8\sqrt{2}\,\ell(q+1)/\sqrt{2^n} + \frac{40e^2(q+\ell+1)^3\Gamma'(f)+2}{2^n}$, where the last inequality is by the second statement of Corollary 4.18, noting that, by choice of f, the event $h_j = a_j$ is equal to $f(x_j, h_j) = a_j$. Recalling the definition of $\hat{S}$,

$$\begin{aligned}\Pr[V(c, \hat{\mathbf{x}}_c) = 1] = \Pr[c \in \hat{S}] &\le \Pr[\hat{S} \in \mathfrak{S}] + \Pr[c \in \hat{S} \,|\, \hat{S} \notin \mathfrak{S}] \Pr[\hat{S} \notin \mathfrak{S}] \\ &\le \Pr[\mathcal{E} \text{ succeeds}] + p_{triv}(1 - \Pr[\mathcal{E} \text{ succeeds}])\end{aligned} \tag{42}$$

where the final inequality exploits that c is chosen at random and independent of $\hat{x}_1, \ldots, \hat{x}_\ell$, and thus is independent of the event $\hat{S} \notin \mathfrak{S}$. Combining (40), (41) and (42), we obtain

$$\Pr[\mathcal{P}^{RO} \text{ succeeds}] \le \Pr[\mathcal{E} \text{ succeeds}] + p_{triv}(1 - \Pr[\mathcal{E} \text{ succeeds}]) + \delta_1 + \delta_2$$

and solving for $\Pr[\mathcal{E}$ succeeds$]$ gives the claimed bound. □

4.5.4 Tightness

The bound given by Theorem 4.24 is tight in the sense that the extraction success probability is proportional to the advantage of a malicious prover over the trivial success probability, up to a negligible additive error term. On top, the additive error term is asymptotically tight: ε remains negligible in n for $q = 2^{\alpha n}$ with any $\alpha < \frac{1}{3}$, while with $q = 2^{n/3}$ queries a collision in the hash function can be found with constant success probability [BHT98; Zha15a] , breaking the binding property of the commitment scheme upon which typical soundness proofs for commit-and-open Σ-protocols rely.

It is even not too hard to find relevant examples of commit-and-open Σ-protocols where a collision-finding attack not only invalidates the soundness proof but leads to an actual attack against extractability. Consider e.g. the Σ-protocol ZKBoo that underlies the signature scheme Picnic. Here, the prover commits to three messages m_1, m_2, m_3 as $a_i = H(m_i, r_i)$ for random strings r_1, r_2, r_3, and where the m_i's are the respective views of the three parties in an "in-the-head" execution of a 3-party-computation protocol. The challenge space is $C = \{\{1,2\},\{1,3\},\{2,3\}\}$, which means that the prover is then asked to open two out of the three commitments. Now consider the following attack. The attacker can easily find pairs (m_1, m_2), (m_1', m_3) and (m_2', m_3'), so that each pair consists of two mutually consistent views of the considered 3-party-computation protocol. Now the only thing the attacker has to do is to find three collisions in the hash function of the form $a_i = H(m_i, r_i) = H(m_i', r_i')$, $i = 1, 2, 3$. This can be done using e.g. the BHT algorithm [BHT98] if r_i are sufficiently long. The attacker now sends (a_1, a_2, a_3), receives a challenge and responds with the appropriate preimages of the two commitments indicated by the challenge.

4.5.5 Application to Fiat Shamir Signatures

Σ-protocols are commonly used to obtain non-interactive zero-knowledge proofs and digital signatures via the Fiat Shamir (FS) transform. Here, the random challenges are (possibly after a suitable number of parallel repetitions) replaced by the hash of the first message in the 3-round protocol, thus making the protocol non-interactive. To construct a digital signature scheme (DSS), the message to be signed is included in the hash argument.[50]

[50] For FS DSS, the relation R needs to admit an efficient generator of hard instances.

The post-quantum security of FS signatures has recently drawn additional attention. This is mainly because FS signatures are some of the most promising candidates for replacing RSA and elliptic curve signatures, which can be broken by quantum adversaries. Indeed, two out of the 6 round-3 candidate DSSs in the NIST standardization process for post-quantum cryptographic schemes, CRYSTALS Dilithium [DKL+18a] and Picnic [CDG+17], are FS signature schemes. In the QROM,[51] the chain of arguments for reducing the UF-CMA security of a FS signature scheme $\mathsf{Sig}[\Sigma]$ to the i) honest-verifier zero-knowledge, and ii) (some variant of the) special soundness, properties of the underlying Σ-protocol Σ as follows (also depicted in Figure 4.7).

- First, the UF-CMA security of $\mathsf{Sig}[\Sigma]$ is reduced to plain unforgeability (UF-NMA), using the HVZK property of Σ [KLS18; GHHM21; BBD+23].
- The UF-NMA property of $\mathsf{Sig}[\Sigma]$ follows from the extractability of the Fiat Shamir transformation $\mathsf{FS}[\Sigma]$ of Σ.
- The extractability of $\mathsf{FS}[\Sigma]$ is then reduced to the extractability of Σ (Chapter 3).
- Finally, the extractability of Σ is reduced to the (variant of) special soundness of Σ [Unr12].

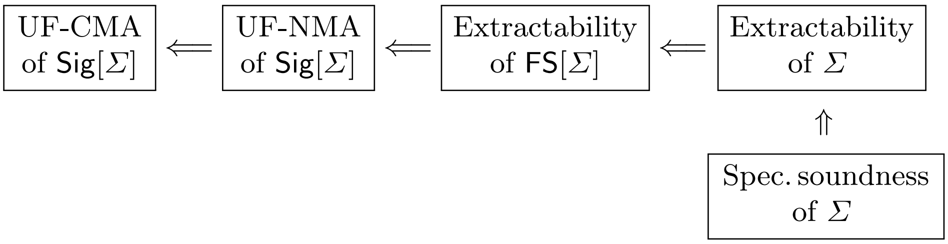
UF-CMA of $\mathsf{Sig}[\Sigma]$
$\Longleftarrow$
UF-NMA of $\mathsf{Sig}[\Sigma]$
$\Longleftarrow$
Extractability of $\mathsf{FS}[\Sigma]$
$\Longleftarrow$
Extractability of Σ
$\Uparrow$
Spec. soundness of Σ

Fig. 4.7. Chain of arguments for proving security of FS signatures.

Prior to this work, the last step (arguing extractability from special soundness) has relied on Unruh's rewinding lemma [Unr12], which after suitable generalization leads, e.g., to a $2k+1$-th root loss for a k-sound Σ. For commit-and-open Σ-protocols, Theorem 4.24 can replace Unruhs rewinding lemma when working in the QROM, making the last step above tight up to unavoidable additive errors.

[51] The typical ROM reductions proceed similarly

As an example, Theorem 4.24 implies a sizeable improvement over the current best QROM security proof of Picnic2 [CDG+17; CDG+20]. Indeed, Unruh's rewinding lemma implies a 6-th root loss for the variant of special soundness the underlying Σ-protocol possesses (Lemma 3.26), while Theorem 4.24 is tight.

We note that for commit-and-open Σ-protocols, there is room for further improvement by means of combining the last two steps and doing a *direct* analysis of $\mathsf{FS}[\Sigma]$. Indeed, [Cha21] suggested such an approach, but the analysis provided there there still relies on some unproven assumption. In Chapter 5 we solve this problem by giving a direct analysis wihtout any gaps.

Section 4.6

QROM-Security of Textbook Fujisaki-Okamoto

4.6.1 The Fujisaki-Okamoto Transformation

The Fujisaki-Okamoto (FO) transform [FO99] is a general method to turn any public-key encryption scheme secure against *chosen-plaintext attacks* (CPA) into a key-encapsulation mechanism (KEM) that is secure against *chosen-ciphertext attacks* (CCA). We can start either from a scheme with one-way security against CPA attacks (OW-CPA) or from one with indistinguishability against CPA attacks (IND-CPA), and in both cases obtain an IND-CCA secure KEM. We recall that a KEM establishes a shared key, which can then be used for symmetric encryption.

Recall the formal definitions of a public-key encryption scheme, of a KEM and of the notions of δ-correctness and γ-spreadness in Section 2.3. In addition, we define a relaxed version of the latter property, *weak* γ-spreadness (see Definition 2.21), where the ciphertexts are only required to have high min-entropy when averaged over key generation.[52]. The security games for OW-CPA security of a public-key encryption scheme and for IND-CCA security of a KEM are given in Figure 4.8.

The formal specification of the FO transformation, mapping a public-key encryption scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ and two suitable hash functions H

[52] This seems relevant e.g. for lattice-based schemes, where the ciphertext has little (or even no) entropy for certain very unlikely choices of the key (like being all 0)

```
GAME OW-CPA
  $(pk, sk) \leftarrow \mathsf{Gen}$
  $m^* \xleftarrow{\$} \mathcal{M}$
  $c^* \leftarrow \mathsf{Enc}_{pk}(m^*)$
  $m' \leftarrow \mathcal{A}(pk, c^*)$
  return $m' == m^*$

GAME IND-CCA-KEM
  $(pk, sk) \leftarrow \mathsf{Gen}$
  $b \xleftarrow{\$} \{0, 1\}$
  $(K_0^*, c^*) \leftarrow \mathsf{Encaps}(pk)$
  $K_1^* \xleftarrow{\$} \mathcal{K}$
  $b' \leftarrow \mathcal{A}^{\textsc{Decaps}}(c^*, K_b^*)$
  return $b' == b$

Decaps($c \neq c^*$)
  $K := \mathsf{Decaps}_{sk}(c)$
  return $K$
```

Fig. 4.8. Games for OW-CPA security of a PKE and IND-CCA security of a KEM. In the latter, $\mathcal{A}$ is not allowed to query c^* to Decaps.

and G (which will then be modeled as random oracles) into a key encapsulation mechanism $\mathsf{FO}[\mathsf{PKE}, H, G] = (\mathsf{Gen}, \mathsf{Encaps}, \mathsf{Decaps})$, is given in Figure 4.9.

```
Gen
  $(sk, pk) \leftarrow \mathsf{Gen}$
  return $(sk, pk)$

Encaps($pk$)
  $m \xleftarrow{\$} \mathcal{M}$
  $c \leftarrow \mathsf{Enc}_{pk}(m; H(m))$
  $K := G(m)$
  return $(K, c)$

Decaps$_{sk}(c)$
  $m := \mathsf{Dec}_{sk}(c)$
  if $m = \perp$ or $\mathsf{Enc}_{pk}(m; H(m)) \neq c$
      return $\perp$
  else return $K := G(m)$
```

Fig. 4.9. The KEM $\mathsf{FO}[\mathsf{PKE}, H, G]$, obtained by applying the FO transformation [FO99] to PKE.

4.6.2 Post-Quantum Security of FO in the QROM

Our main contribution here is the following security result for the FO transformation in the QROM. In contrast to most of the previous works on the topic, our result applies to the *standard* FO transformation, without any adjustments. Next to being CPA secure, we require the underlying public-key encryption scheme to be so that ciphertexts have a lower-bounded amount of min-entropy (resulting from the encryption randomness), captured by the aforementioned spreadness property. This seems unavoidable for the FO transformation with explicit rejection and without any adjustment, like an additional key confirmation hash (as e.g. in [TU16]).

Theorem 4.25. *Let* PKE *be a δ-correct public-key encryption scheme satisfying weak γ-spreadness. Let $\mathcal{A}$ be any* IND-CCA *adversary against* $\mathsf{FO}[\mathsf{PKE}, H, G]$, *making $q_D \geq 1$ queries to the decapsulation oracle* Decaps *and q_H and q_G queries to $H : \mathcal{M} \to \mathcal{R}$ and $G : \mathcal{M} \to \mathcal{K}$, respectively, where H and G are mod-*

eled as random oracles. Let $q := q_H + q_G + 2q_D$*. Then, there exists a* $\mathsf{OW\text{-}CPA}$ *adversary* $\mathcal{B}$ *against* PKE *with*

$$\mathsf{ADV}[\mathcal{A}]^{\mathsf{IND\text{-}CCA}}_{\mathrm{KEM}} \leq 2q\sqrt{\mathsf{ADV}^{\mathsf{OW\text{-}CPA}}_{\mathrm{PKE}}[\mathcal{B}]} + 24q^2\sqrt{\delta} + 24q\sqrt{qq_D} \cdot 2^{-\gamma/4}\,.$$

Furthermore, $\mathcal{B}$ *has a running time* $T_{\mathcal{B}} \leq T_{\mathcal{A}} + O\big(q_H \cdot q_D \cdot \mathrm{Time}[\mathsf{Enc}] + q^2\big)$.

We start with a proof outline, which is somewhat simplified in that it treats $\mathsf{FO}[\mathsf{PKE}, H, G]$ as an encryption scheme rather than as a KEM. We will transform the adversary $\mathcal{A}$ of the $\mathsf{IND\text{-}CCA}$ game into a $\mathsf{OW\text{-}CPA}$ adversary against the PKE in a number of steps. There are two main challenges to overcome. (1) We need to switch from the *deterministic* challenge ciphertext $c^* = \mathsf{Enc}_{pk}(m^*; H(m^*))$ that $\mathcal{A}$ attacks to a *randomized* challenge ciphertext $c^* = \mathsf{Enc}_{pk}(m^*; r^*)$ that $\mathcal{B}$ is then supposed to attack. We do this switch by re-programming $H(m^*)$ to a random value right after the computation of c^*, which is equivalent to keeping H but choosing a random r^* for computing c^*. For reasons that we explain later, we do this switch from H to its re-programmed variant, denoted $H^\diamond$, in two steps, where the first step (from **Game 0** to **1**) will be "for free", and the second step (from **Game 1** to **2**) is argued using the O2H lemma ([Unr14b], we use the version given in [AHU19], Theorem 3). (2) We need to answer decryption queries without knowing the secret key. At this point our extractable RO-simulator steps in. We replace $H^\diamond$, modelled as a random oracle, by $\mathcal{S}$, and we use its extraction interface to extract m from any correctly formed encryption $c = \mathsf{Enc}_{pk}(m; H^\diamond(m))$ and to identify incorrect ciphertexts.

One subtle issue in the argument above is the following. The O2H lemma ensures that we can find m^* by measuring one of the queries to the random oracle. However, given that also the decryption oracle makes queries to the random oracle (for performing the re-encryption check), it could be the case that one of those decryption queries is the one selected by the O2H extractor. This situation is problematic since, once we switch to $\mathcal{S}$ to deal with the decryption queries, some of these queries will be dropped (namely when $\mathcal{S}.E(c) = \emptyset$). This is problematic because, per-se, we cannot exclude that this is the one query that will give us m^*. We avoid this problem by our two-step approach for switching from H to $H^\diamond$, which ensures that the only ciphertext c that would bring us in the above unfortunate situation is the actual (randomized) *challenge ciphertext* $c^* = \mathsf{Enc}_{pk}(m^*; r^*)$, which is not submitted by the specification of the security game.

```
GAME SETUP $G_0$-$G_8$
$(pk, sk) \leftarrow \mathsf{Gen}$                                   $/\!\!/ G_0$-$G_7$
$(b, m^*) \xleftarrow{\$} \{0,1\} \times \mathcal{M}$                 $/\!\!/ G_0$-$G_7$
$c^* := \mathsf{Enc}_{pk}(m^*; H(m^*))$                            $/\!\!/ G_0$-$G_7$
$\mathbf{input}(pk, c^* = \mathsf{Enc}_{pk}(m^*))$                  $/\!\!/ G_8$
$c^\diamond := \mathsf{Enc}_{pk}(m^*; H^\diamond(m^*))$             $/\!\!/ G_0$-$G_6$
$K_0^* := G(m^*)$                                                  $/\!\!/ G_0$-$G_2$
$K_1^* \xleftarrow{\$} \mathcal{K}$
$j \xleftarrow{\$} J_{\mathcal{A}} \cup J_{D(c^\diamond)}$          $/\!\!/ G_3$-$G_6$
$j \xleftarrow{\$} J$                                              $/\!\!/ G_7$-$G_8$

MAIN PHASE $G_0$-$G_2$
$b' \leftarrow \mathcal{A}^{\text{DECAPS},H,G}(c^*, K_b^*)$         $/\!\!/ G_0$-$G_1$
$b' \leftarrow \mathcal{A}^{\text{DECAPS},H^\diamond,G^\diamond}(c^*, K_b^*)$   $/\!\!/ G_2$
$\mathbf{return}\ b' == b$

MAIN PHASE $G_3$-$G_8$
$m' \leftarrow \mathcal{MA}_j^{\text{DECAPS},H^\diamond,G^\diamond}(c^*, K_1^*)$          $/\!\!/ G_3$
$m' \leftarrow \mathcal{MA}_j^{\text{DECAPS},\mathcal{S}.RO,G^\diamond}(c^*, K_1^*)$       $/\!\!/ G_4$-$G_5$
$m' \leftarrow \mathcal{EA}_j^{\text{DECAPS},\mathcal{S}.RO,G^\diamond}(c^*, K_1^*)$       $/\!\!/ G_6$-$G_8$
$\mathbf{while}\ i \in I\ \mathbf{do}$                              $/\!\!/ G_4$
    $\hat{m}_i \leftarrow \mathcal{S}.E(c_i)$                       $/\!\!/ G_4$
$\mathbf{return}\ m'$

DECAPS$(c \neq c^*)$ $G_0$-$G_5$
$m := \mathsf{Dec}_{sk}(c)$                                        $/\!\!/ G_0$-$G_5$
$\mathbf{if}\ m = \bot\ \mathbf{return}\ \bot$                      $/\!\!/ G_0$-$G_5$
$h := H(m), g := G(m)$                                             $/\!\!/ G_0$
$\mathbf{if}\ c = c^\diamond$                                       $/\!\!/ G_1$
    $h := H(m), g := G(m)$                                         $/\!\!/ G_1$
$\mathbf{else}$                                                    $/\!\!/ G_1$
    $h := H^\diamond(m), g := G^\diamond(m)$                       $/\!\!/ G_1$
$h := H^\diamond(m), g := G^\diamond(m)$                           $/\!\!/ G_2$-$G_3$
$h := \mathcal{S}.RO(m), g := G^\diamond(m)$                       $/\!\!/ G_3$-$G_5$
$\mathbf{if}\ \mathsf{Enc}_{pk}(m; h) \neq c$                       $/\!\!/ G_0$-$G_5$
    $\mathbf{return}\ \bot$                                        $/\!\!/ G_0$-$G_5$
$\mathbf{else\ return}\ K := g$                                     $/\!\!/ G_0$-$G_5$
$\hat{m} \leftarrow \mathcal{S}.E(c)$                              $/\!\!/ G_5$

DECAPS$(c \neq c^*)$ $G_6$-$G_8$
$m := \mathsf{Dec}_{sk}(c)$                                        $/\!\!/ G_6$-$G_7$
$\mathbf{query}\ \mathcal{S}.RO(m)$                                $/\!\!/ G_6$-$G_7$
$\hat{m} \leftarrow \mathcal{S}.E(c)$                              $/\!\!/ G_6$-$G_8$
$\mathbf{if}\ \hat{m} = \bot\ \mathbf{return}\ \bot$                $/\!\!/ G_6$-$G_8$
$\mathbf{else\ return}\ K := G^\diamond(\hat{m})$                   $/\!\!/ G_6$-$G_8$
```

Fig. 4.10. Games 0 to 8. H and G are independent random oracles; $H^\diamond$ and $G^\diamond$ coincide with H and G, respectively, except that $H^\diamond(m^*)$ and $G^\diamond(m^*)$ are freshly chosen. We consider the oracle queries to $H^\diamond$ (respectively to $\mathcal{S}.RO$ later on) and to $G^\diamond$ to be labeled by indices $j \in J$, where $J = J_{\mathcal{A}} \cup J_D$ decomposes this set into those queries made by $\mathcal{A}$ and those made by DECAPS, respectively, and $J_{D(c^\diamond)} \subseteq J_D$ consists of DECAPS' queries upon input $c^\diamond$. Similarly, we consider the queries to DECAPS to be indexed by $i \in I$, with c_i then being the corresponding ciphertext. Since $\mathcal{A}$ is not allowed to query c^* to DECAPS, we have $c_i \neq c^*$ $\forall\, i \in I$. For $j \in J$, $\mathcal{MA}_j^{\text{DECAPS}}$ denotes the execution of $\mathcal{A}^{\text{DECAPS}}$ up to the query indexed by j, and followed by measuring this query and outputting the result. $\mathcal{EA}_j^{\text{DECAPS}}$ coincides with $\mathcal{MA}_j^{\text{DECAPS}}$, except that if $j \in J_D$ then it outputs the corresponding $\hat{m}_i$ instead. The colors are meant to help the reader track (the use of) some variables and concepts that occur in different places across the code.

Proof (of Theorem 4.25). **Games 0** to **8** below show how to turn $\mathcal{A}$ into $\mathcal{B}$ (see also Figure 4.10). We first analyze the sequence of hybrids for a fixed key pair (sk, pk). Let therefore $\mathsf{ADV}_{sk}[\mathsf{A}]^{\mathsf{IND\text{-}CCA}}_{\mathrm{KEM}}$ be A's advantage for key pair (sk, pk). In addition, for a fixed pair (sk, pk), let δ_{sk} be the maximum probability of a decryption error and g_{sk} be the maximum probability of any ciphertext, so that $\mathbb{E}\big[\delta_{sk}\big] \le \delta$ and $\mathbb{E}\big[g_{sk}\big] \le 2^{-\gamma}$, with the expectation over $(sk, pk) \leftarrow \mathsf{Gen}$ (we can assume without loss of generality that pk is included in sk).

Game 0 is the $\mathsf{IND\text{-}CCA}$ game for KEMs, except that we replace the random oracles G and H with a single random oracle F, by setting $H(x) := F(0||x)$ and $G(x) := F(1||x)$.[53] When convenient, we still refer to $F(0\|\cdot)$ as H and $F(1\|\cdot)$ as G. This change does not affect the view of the adversary nor the outcome of the game; therefore,

$$\Pr[b = b' \text{ in } \mathbf{Game\ 0}] = \frac{1}{2} + \mathsf{ADV}_{sk}[\mathsf{A}]^{\mathsf{IND\text{-}CCA}}_{\mathrm{KEM}}.$$

In **Game 1**, we introduce a new oracle $F^\diamond$ by setting $F^\diamond(0\|m^*) := r^\diamond$ and $F^\diamond(1\|m^*) := k^\diamond$ for uniformly random $r^\diamond \in \mathcal{R}$ and $k^\diamond \in \mathcal{K}$, while letting $F^\diamond(b\|m) := F(b\|m)$ for $m \neq m^*$ and $b \in \{0, 1\}$. We note that while the *joint* behavior of $F^\diamond$ and F depends on the choice of the challenge message m^*, each one individually is a purely random function, i.e., a random oracle. In line with F, we write $H^\diamond$ for $F^\diamond(0\|\cdot)$ and $G^\diamond$ for $F^\diamond(1\|\cdot)$ when convenient.

Using these definitions, **Game 1** is obtained from **Game 0** via the following modifications. After m^* and c^* have been produced and before $\mathcal{A}$ is executed, we compute $c^\diamond := \mathsf{Enc}_{pk}(m^*; r^\diamond) = \mathsf{Enc}_{pk}(m^*; H^\diamond(m^*))$, making a query to $H^\diamond$ to obtain $r^\diamond$. Furthermore, for every decapsulation query by $\mathcal{A}$, we let DECAPS use $H^\diamond$ and $G^\diamond$ instead of H and G for checking correctness of the queried ciphertexts c_i and for computing the key K_i, *except* when $c_i = c^\diamond$ (which we may assume to happen at most once), in which case DECAPS still uses H and G. We claim that

$$\Pr[b = b' \text{ in } \mathbf{Game\ 1}] = \Pr[b = b' \text{ in } \mathbf{Game\ 0}] = \frac{1}{2} + \mathsf{ADV}_{sk}[\mathsf{A}]^{\mathsf{IND\text{-}CCA}}_{\mathrm{KEM}}.$$

Indeed, for any decryption query c_i, we either have $\mathsf{Dec}_{sk}(c_i) =: m_i \neq m^*$ and thus $F^\diamond(b\|m_i) = F(b\|m_i)$, or else $m_i = m^*$; in the latter case we then either have $c_i = c^\diamond$, where nothing changes by definition of the game, or else

[53] These assignments seem to suggest that $\mathcal{R} = \mathcal{K}$, which may not be the case. Indeed, we understand here that $F : \mathcal{M} \to \{0, 1\}^n$ with n large enough, and $F(0||x)$ and $F(1||x)$ are then cut down to the right size.

$\mathsf{Enc}_{pk}(m^*; H(m^*)) = c^* \neq c_i \neq c^\diamond = \mathsf{Enc}_{pk}(m^*; H^\diamond(m^*))$, and hence the re-encryption check fails and $K_i := \perp$ in either case, without querying G or $G^\diamond$. Therefore, the input-output behavior of Decaps is not affected.

In **Game 2**, all oracle calls by Decaps (also for $c_i = c^\diamond$) and all calls by $\mathcal{A}$ are now to $F^\diamond$. Only the challenge ciphertext $c^* = \mathsf{Enc}_{pk}(m^*; H(m^*))$ is still computed using H, and thus with randomness $r^* = H(m^*)$ that is random and independent of m^* and $F^\diamond$. Hence, looking ahead, we can think of c^* as the input to the $\mathsf{OW\text{-}CPA}$ game that the to-be-constructed attacker $\mathcal{B}$ will attack. Similarly, $K_0^* = G(m^*)$ is random and independent of m^* and $F^\diamond$, exactly as K_1^* is, which means that $\mathcal{A}$ can only win with probability $\frac{1}{2}$.

By the O2H lemma ([AHU19], Theorem 3), the difference between the respective probabilities of $\mathcal{A}$ in guessing b in **Game 1** and **2** gives a lower bound on the success probability of a particular procedure to find an input on which F and $F^\diamond$ differ, and thus to find m^*. Formally,

$$\begin{aligned} 2(q_H + q_G + 2)\sqrt{\Pr[m' = m^* \text{ in } \textbf{Game 3}]} \\ &\geq |\Pr[b' = b \text{ in } \textbf{Game 1}] - \Pr[b' = b \text{ in } \textbf{Game 2}]| \\ &= \frac{1}{2} + \mathsf{ADV}_{sk}[\mathsf{A}]^{\mathsf{IND\text{-}CCA}}_{\mathrm{KEM}} - \frac{1}{2} \\ &= \mathsf{ADV}_{sk}[\mathsf{A}]^{\mathsf{IND\text{-}CCA}}_{\mathrm{KEM}} \end{aligned}$$

where **Game 3** is identical to **Game 2** above, except that we introduce and consider a new variable m' (with the goal that $m' = m^*$), obtained as follows. Either one of the $q_H + q_G$ queries from $\mathcal{A}$ to $H^\diamond$ and $G^\diamond$ is measured, or one of the two respective queries from DECAPS to $H^\diamond$ and $G^\diamond$ upon a possible decryption query $c^\diamond$ is measured, and, in either case, m' is set to be the corresponding measurement outcome. The choice of which of these $q_H + q_G + 2$ queries to measure is done uniformly at random.[54]

We note that, since we are concerned with the measurement outcome m' only, it is irrelevant whether the game stops right after the measurement, or it continues until $\mathcal{A}$ outputs b'. Also, rather than actually measuring DECAPS' classical query to $H^\diamond$ or $G^\diamond$ upon decryption query $c_i = c^\diamond$ (if instructed to do so), we can equivalently set $m' := m_i = \mathsf{Dec}_{sk}(c^\diamond)$.

For **Game 4**, we consider the function $f : \mathcal{M} \times \mathcal{R} \to \mathcal{C}$, $(m, r) \mapsto \mathsf{Enc}_{pk}(m; r)$, and we replace the random oracle $H^\diamond$ with the extractable RO-simulator $\mathcal{S}$ from Theorem 4.12. Furthermore, *at the very end* of the game, we

[54] If this choice instructs to measure DECAPS's query to $H^\diamond$ or to $G^\diamond$ for the decryption query $c^\diamond$, but there is no decryption query $c_i = c^\diamond$, $m' := \perp$ is output instead.

invoke the extractor interface $\mathcal{S}.E$ to compute $\hat{m}_i := \mathcal{S}.E(c_i)$ for each c_i that A queried to Decaps in the course of its run. By the first statement of Theorem 4.12, given that the $\mathcal{S}.E$ queries take place only *after* the run of $\mathcal{A}$,

$$\Pr[m' = m^* \text{ in } \textbf{Game 4}] = \Pr[m' = m^* \text{ in } \textbf{Game 3}] .$$

Furthermore, applying Proposition 4.13 for $R' := \{(m, c) : \mathsf{Dec}_{sk}(c) \neq m\}$, we get that the event

$$P^\dagger := \big[\forall i : \hat{m}_i = m_i \vee \hat{m}_i = \emptyset\big]$$

holds except with probability $\varepsilon_1 := 128(q_H + q_D)^2 \varGamma_R/|\mathcal{R}|$ for $\varGamma_R$ as in Proposition 4.13, which here means that $\varGamma_R/|\mathcal{R}| = \delta_{sk}$. Thus

$$\Pr[m' = m^* \wedge P^\dagger \text{ in } \textbf{Game 4}] \geq \Pr[m' = m^* \text{ in } \textbf{Game 4}] - \varepsilon_1 .$$

In **Game 5**, we query $\mathcal{S}.E(c_i)$ *at runtime*, that is, as part of the Decaps procedure upon input c_i, right after $\mathcal{S}.RO(m)$ has been invoked as part of the re-encryption check (line 27 of Figure 4.10). Since $\mathcal{S}.RO(m)$ and $\mathcal{S}.E(c_i)$ now constitute two subsequent classical queries, it follows from the contraposition of 4.b of Theorem 4.12 that except with probability $2 \cdot 2^{-n}$, $\hat{m}_i = \emptyset$ implies $\mathsf{Enc}_{pk}(m_i; \mathcal{S}.RO(m_i)) \neq c_i$. Applying the union bound, we find that $P^\dagger$ implies

$$P := \big[\forall i : \hat{m}_i = m_i \ \vee \ (\hat{m}_i = \emptyset \ \wedge \ \mathsf{Enc}_{pk}(m_i; \mathcal{S}.RO(m_i)) \neq c_i)\big]$$

except with probability $q_D \cdot 2 \cdot 2^{-n}$. Furthermore, By 2.c of that same Theorem 4.12, each swap of a $\mathcal{S}.RO$ with a $\mathcal{S}.E$ query affects the final probability by at most $8\sqrt{2\varGamma(f)/|\mathcal{R}|} = 8\sqrt{2g_{sk}}$. Thus

$$\Pr[m' = m^* \ \wedge \ P \text{ in } \textbf{Game 5}] \geq \Pr[m' = m^* \ \wedge \ P^\dagger \text{ in } \textbf{Game 4}] - \varepsilon_2$$

with $\varepsilon_2 := 2q_D \cdot \big((q_H + q_D) \cdot 4\sqrt{2g_{sk}} + 2^{-n}\big)$.

In **Game 6**, Decaps uses $\hat{m}_i$ instead of m_i to compute K_i. That is, it sets $K_i := \bot$ if $\hat{m}_i = \emptyset$ and $K_i := G^\diamond(\hat{m}_i)$ otherwise. Also, if instructed to output $m' := m_i$ where $c_i = c^\diamond$, then the output is set to $m' := \hat{m}_i$ instead. In all cases, Decaps still queries $\mathcal{S}.RO(m_i)$, so that the interaction pattern between Decaps and $\mathcal{S}.RO$ remains as in **Game 5**.

Here, we note that if the event

$$P_i := \big[\hat{m}_i = m_i \ \vee \ (\hat{m}_i = \emptyset \ \wedge \ \mathsf{Enc}_{pk}(m_i; \mathcal{S}.RO(m_i)) \neq c_i)\big]$$

holds for a given i then the above change will not affect Decaps' response K_i, and thus also not the probability for P_{i+1} to hold as well. Therefore, by induction, $\Pr[P$ in **Game 6**$] = \Pr[P$ in **Game 5**$]$, and since conditioned on the event P the two games are identical, we have

$$\Pr[m' = m^* \wedge P \text{ in } \textbf{Game 6}] = \Pr[m' = m^* \wedge P \text{ in } \textbf{Game 5}].$$

In **Game 7**, instead of obtaining m' by measuring a random query of $\mathcal{A}$ to either $\mathcal{S}.RO$ or G, or outputting $\hat{m}_i$ with $c_i = c^\diamond$, here m' is obtained by measuring a random query of $\mathcal{A}$ to either $\mathcal{S}.RO$ or G, or outputting $\hat{m}_i$ for a *random* $i \in \{1, \dots, q_D\}$, where the former case is chosen with probability $(q_H + q_G)/(q_H + q_G + 2q_D)$ and the latter with probability $2q_D/(q_H + q_G + 2q_D)$. Since conditioned on the first case being chosen or the latter with $i = i_\diamond$, **Game 7** coincides with **Game 6**, we have

$$\Pr[m' = m^* \text{ in } \textbf{Game 7}] \geq \frac{q_H + q_G + 2}{q_H + q_G + 2q_D} \cdot \Pr[m' = m^* \text{ in } \textbf{Game 6}]\,.$$

In **Game 8**, we observe that the response to the query $\mathcal{S}.RO(m^*)$, introduced in **Game 1** in order to compute $c^\diamond$, and the responses to the queries that Decaps makes to $\mathcal{S}.RO$ on input m_i do not affect the game anymore, and thus we can drop all these queries, or, equivalently, move them to the very end of the execution of the game. Invoking once again 2.c of Theorem 4.12, we then get

$$\Pr[m' = m^* \text{ in } \textbf{Game 8}] \geq \Pr[m' = m^* \text{ in } \textbf{Game 7}] - \varepsilon_3\,,$$

for $\varepsilon_3 = (q_D + 1) \cdot q_H \cdot 8\sqrt{2g_{sk}}$.

With these queries now dropped, we observe that **Game 8** works without knowledge of the secret key sk, and thus constitutes a $\mathsf{OW\text{-}CPA}$ attacker $\mathcal{B}$ against PKE, which takes as input a public key pk and an encryption c^* of a random message $m^* \in \mathcal{M}$, and outputs m^* with the given probability, i.e, $\mathsf{ADV}_{sk}[\mathsf{B}]^{\mathsf{OW\text{-}CPA}}_{\mathrm{PKE}} \geq \Pr[m' = m^*$ in **Game 8**$]$. We note that the oracle $G^\diamond$ can be simulated using standard techniques.

Backtracking all the above (in)equalities and setting $\varepsilon_{23} := \varepsilon_2 + \varepsilon_3$, $q_{HG} := q_H + q_G$ etc. and $q := q_H + q_G + 2q_D$, we get the following bound:

$$\begin{aligned}
\mathsf{ADV}_{sk}[\mathcal{A}]^{\mathsf{IND\text{-}CCA}}_{\mathrm{KEM}} &\leq 2(q_{HG} + 2)\sqrt{\frac{q_{HG} + 2q_D}{q_{HG} + 2}\big(\mathsf{ADV}_{sk}[\mathsf{B}]^{\mathsf{OW\text{-}CPA}}_{\mathrm{PKE}} + \varepsilon_3\big) + \varepsilon_1 + \varepsilon_2} \\
&\leq 2(q_{HG} + 2q_D)\sqrt{\mathsf{ADV}_{sk}[\mathsf{B}]^{\mathsf{OW\text{-}CPA}}_{\mathrm{PKE}} + \varepsilon_{23}} + 2(q_{HG} + 2)\sqrt{\varepsilon_1} \\
&\leq 2q\Big(\sqrt{\mathsf{ADV}_{sk}[\mathsf{B}]^{\mathsf{OW\text{-}CPA}}_{\mathrm{PKE}}} + \sqrt{\varepsilon_{23}} + \sqrt{\varepsilon_1}\Big)\,.
\end{aligned}$$

Additionally,

$$\begin{aligned}\sqrt{\varepsilon_{23}} =& \sqrt{2q_D \cdot \Big(4\big((q_H+q_D)+(q_D+1)q_H\big)\sqrt{2g_{sk}}+2^{-n}\Big)}\\ \leq& 6\sqrt{q_H q_D}\cdot\Big(g_{sk}^{1/4}+2^{-n/2}\Big)\\ \leq& 12\sqrt{q q_D}\cdot g_{sk}^{1/4},\end{aligned}$$

where we have used the fact that $2^{-n} \leq g_{sk} \leq 1$ in the last line. Taking the expectation over $(sk, pk) \leftarrow \mathsf{Gen}$, applying Jensen's inequality and using $q_H + q_D \leq q$ once more, we get the claimed bound. Finally, we note that the runtime of $\mathcal{B}$ is given by $T_{\mathcal{B}} = T_{\mathcal{A}} + T_{\text{DECAPS}} + T_G + T_{\mathcal{S}}$, where apart from its oracle queries DECAPS runs in time linear in q_D, and $\mathcal{S}$ can be simulated in time

$$T_{\mathcal{S}} = O\big(q_{RO}\cdot q_E \cdot \mathrm{Time}[f] + q_{RO}^2\big) = O\big(q_H \cdot q_D \cdot \mathrm{Time}[\mathsf{Enc}] + q^2\big)$$

by Theorem 4.12, and similarly for G. □

4.6.3 A gap in the security proof from [Zha19a] for the FO transformation

In his seminal paper [Zha19a], Zhandry introduced the so-called compressed-oracle technique, a ground-breaking method that led to many new results in post-quantum cryptography, quantum query complexity and beyond. One of the most important features of the compressed-oracle methodology is that it allows the approximate recovery of several features of the classical ROM, that were previously believed lost when moving to the QROM.

The new, "virtually classical" ways of reasoning about quantum access to a random oracle are very intuitive. This fact bears a certain risk that the reach of classical intuition in the compressed-oracle framework is overestimated. In the following, we describe a gap in the security proof for the Fujisaki-Okamoto (FO) transformation given in [Zha19a], which was likely caused by following the classical intuition too closely.

One step in security reductions for the FO transformation is the simulation of the decryption or decapsulation oracle without making use of the secret key. This simulation is done by accessing (either actively by programming, or passively by preimage awareness) the adversary's random-oracle interface. For proofs in the QROM, the adversary's queries cannot be compiled into a list in a straight-forward manner (due to the no-cloning principle, if you will). If a reduction collects information about an adversary's QROM queries *during runtime*, be it by directly accessing the adversary's query input or output, or by acting on the compressed-oracle register, it needs to be analyzed to which degree the information-collection operation can be noticed by the adversary.

In the security proof for the FO transformation in [Zha19a], the replacement of the decryption oracle by a simulated version happens gradually in Hybrids 2 to 4 (Lemma 43 and 44 in the full version of [Zha19a]). In more detail, in Hybrid 2 a (purified) "test" is performed on the state of the compressed oracle before the reply to the decryption query is prepared and sent, and then uncomputed again right afterwards; since (due to Lemma 39 of [Zha19a]) the uncomputation almost commutes with the re-encryption check performed as part of the preparation of the reply, this "test" and its uncomputation have negligibe effect. In Hybrid 3, the result of the "test" is then used in the derivation of the reply to the decryption query by setting the reply to $\perp$ in case the "test" fails. Finally, in Hybrid 4, it is declared that the (simulated) decryption oracle *"scans over the inputs of the [compressed oracle] database for G, looking for inputs [of a certain form]. For each one, we will check if [it encrypts to the queried ciphertext]"*; the first database entry where the check succeeds is then used to answer the query.

Using a more formal language, in each of these hybrids the reply to the decryption query is obtained by means of applying a measurement to the state of the compressed oracle (where the measurement depends on the queried ciphertext c, and on the secret key in Hybrids 2 and 3). In Hybrid 2, the measurement consists of the "test", the (ordinary) derivation of the oracle response, and the uncomputation of the "test". At the other end, in Hybrid 4, it consists of all the "scanning" and "checking" etc. By the nature of quantum measurements, in both steps, from Hybrid 2 to 3 and from Hybrid 3 to 4, *both* the reply of the (simulated) decryption oracle *and* the post-measurement state of the compressed oracle (and thus the future behavior of the compressed oracle) may change. While in the proof in [Zha19a] it is argued for both steps, from Hybrid 2 to 3 and from Hybrid 3 to 4, that the reply of the (simulated) decryption oracle does (almost) not change, for neither of the two steps is it argued that the post-measurement state is not (much) affected. As a matter of fact, Hybrids 3 and 4 are described in such a "virtually classical" way that there is ambiguity to translate them into proper descriptions of quantum measurements, necessary to analyze the effect on the post-measurement state.

It seems to us that completing the proof in [Zha19a], which requires to rigorously specify the respective quantum measurements in Hybrids 3 and 4 and to analyze the resulting disturbance of the state of the compressed oracle, is non-trivial. Given the informal description of the hybrids, we find it hard to judge whether it is “only” a question of filling in the gaps, or whether the claimed indistinguishability of the hybrids is actually false (our proof uses a different sequence of hybrids).

Exactly the same problem exists in follow-up work by Katsumata, Kwiatkowski, Pintore and Prest [KKPP20], who follow the FO proof outline from [Zha19a].

Chapter 5

Efficient NIZKs and Signatures from Commit-and-Open Protocols

Chapter contents

Section 5.1
Introduction

Some interactive proofs come with amazing properties like *zero-knowledge*, which intuitively allows a prover to convince a verifier that she knows the witness to an NP-statement without giving away any information about this witness. Such zero-knowledge proofs of knowledge are some of the most fascinating objects in cryptography, and possibly in all of theoretical computer science. One might suspect that their "magic" is rooted in the fact that the prover and verifier run an *interactive* protocol with each other, and that this interaction causes the verifier to be convinced. Surprisingly, if the interactive proof is of suitable form, e.g. a Σ-protocol (a 3-round public-coin protocol, Section 2.2.1), the Fiat-Shamir transformation [FS87] (Section 2.2.3) provides a natural way to remove the interaction from such protocols while preserving (most of) the security properties, resulting in *non-interactive zero-knowledge* proofs (NIZKs). The idea is to compute the challenge c as a hash $c = H(a)$ of the first message, rather than letting the verifier choose c. If the original Σ-protocol has additional soundness properties, the resulting NIZK after the Fiat-Shamir transformation is ideally suited to be turned into a *digital-signature scheme*, simply by hashing the message m to be signed together with the first message a in order to obtain the challenge c. The (former) candidates Picnic [CDG+17] and Dilithium [DKL+18b] in the NIST post-quantum cryptography competition follow this design paradigm.

This intuitive preservation of security properties under the Fiat-Shamir transformation can be formalized in the random-oracle model (ROM), where the hash function H is treated as a uniformly random function, and the security reduction gets *enhanced access* to anybody who queries the random oracle, by seeing which values are queried, and by possibly returning (random-looking) outputs. While this situation is conveniently easy to handle in a non-quantum world, complications arise in the context of post-quantum security. When studying the security of these non-quantum protocols against attackers equipped with large-enough quantum computers, it is natural to assume that such attackers have access to the public description of the employed hash function, and can therefore compute it in superposition on their quantum computers. Therefore, the proper notion of post-quantum security for random oracles is the *quantum-accessible random-oracle model (QROM)* as introduced in [BDF+11]. Due to the difficulty of recording adversarial random-oracle queries in superposition (explained in more detail in Section 1.1.3), establishing post-quantum security in the QROM has turned out to be quite a bit more difficult compared to the regular ROM.

Previous results in [DFMS19] (Chapter 3) (and concurrently in [LZ19b]) establish that for any interactive Σ-protocol Π that is a proof of knowledge, the non-interactive $\mathsf{FS}[\Pi]$ is a proof of knowledge in the QROM. [DFM20] simplified the technical proof and extended these results to multi-round interactive proofs (Section 3.5). However, the most desirable property from such a proof of knowledge is *online extractability*. Indeed, online extractability avoids *rewinding*, which typically causes a significant loss in the security reduction (see later for a comparison) and has other disadvantages. Thus, online extractability allows for the tightest security reductions.

Chailloux was the first to aim for showing online extractability of the Fiat-Shamir transformation in the QROM when considering the relevant class of *commit-and-open* (C&O) Σ-protocols and modelling the hash function used for the commitments (and for computing the challenge) as a random oracle. Indeed, the Fiat-Shamir transformation of such C&O Σ-protocols are known to be online extractable in the classical ROM (see e.g. discussion in [Fis05]). In a first attempt [Cha19], Chailloux tried to lift the argument to the quantum setting by means of Zhandry's compressed-oracle technique [Zha19a] (Section 2.4), which offers a powerful approach for re-establishing ROM results in the QROM, that has been successful in many instances. Unfortunately, this first attempt contained a subtle flaw, which turned out to be unfixable, and despite changing the technical approach, the latest version [Cha21] of this work still contains a gap in the proof, which is put as an assumption.[1]

In Chapter 4 we established online extractability of *interactive* C&O Σ-protocols Π in the QROM; the result applies as soon as Π satisfies some liberal notion of *special soundness*, which is typically satisfied. As pointed out in Section 4.5.5, one can use previous results from [DFMS19; LZ19a; DFM20] to reduce the extractability of the resulting non-interactive protocol $\mathsf{FS}[\Pi]$ to

[1] Informally, quoting from [Cha21], the considered Assumption 2 is that the random oracle can be replaced with a random function of a particular form *"without harming too much the studied scheme"*. More formally, the security loss caused by the considered replacement is assumed to remain bounded by a given function of the number of oracle queries. This assumption is rather ad-hoc and non-standard in that it is very much tailored to the scheme and its proof. Furthermore, even though Assumption 2 is an assumption that could potentially be proven in future work, it is hard to judge whether proving the assumption is actually any easier than proving the security of the considered scheme *directly*, avoiding Assumption 2 — as a matter of fact, in this work we show that the latter is feasible, while Assumption 2 remains open.

the extractability of the interactive protocol Π. However, the resulting extraction error still scales as $O(\varepsilon/q^2)$, which results in a prohibitive loss for digital-signature schemes (see Table 1), leaving open the main question originally posed by Chailloux:

> *How to establish tight security reductions of the Fiat-Shamir transformation for commit-and-open Σ-protocols in the QROM?*

As the technical quantum details of Zhandry's compressed-oracle technique are rather complicated and only accessible for experts, a recent article by Chung, Fehr, Huang and Liao [CFHL21] attempts to give a comprehensive exposition of Zhandry's technique. In addition, they establish a framework that allows researchers without extensive quantum knowledge to still deploy the compressed-oracle technique (in certain cases), basically by reasoning about classical quantities only. In short, the punchline of [CFHL21] is that, if applicable, one can prove *quantum* query complexity lower bounds (think of collision finding, for instance) by means of the following recipe, which is an abstraction of the technique developed in a line of works started by Zhandry [Zha19a; LZ19a; CGLQ20; HM21]. First, one considers the corresponding *classical* query complexity problem, analyzing it by simulating the random oracle using lazy sampling and showing that the database, which keeps track of the oracle queries and the responses, is unlikely to satisfy a certain property (e.g. to contain a collision) after a bounded number of queries. Then, one lifts the analysis to the quantum setting by plugging certain key observations from the classical analysis into generic theorems provided by the [CFHL21] framework.

Contributions. In this chapter, we extend the framework from [CFHL21], and use it in a conceptually new way to establish strong and tight security statements for a large, popular class of non-interactive zero-knowledge proofs and digital signature schemes. In broad strokes, our contributions are threefold.

Online extractability for a class of NIZKs in the QROM. We prove online extractability of the Fiat-Shamir transformation in the QROM for (a large class of) C&O Σ-protocols. This solves the problem considered and attacked by Chailloux. In more detail, we prove that if the considered C&O Σ-protocol satisfies some very liberal notion of special soundness, then the resulting NIZK is a proof of knowledge with online extractability in the QROM, i.e., when the hash function used for the commitments and the Fiat-Shamir transformation is modeled as a quantum-accessible random oracle.

Our security reduction is tight: Whenever a prover outputs a valid proof, the online-extractor succeeds, except with a small probability accounting for collision and preimage attacks on the involved hash functions. For previous reductions, the guaranteed extraction success probability was at least by a factor of q^2 smaller than the succes probability of the prover subjected to extraction (see Table 1). This is our main technical contribution, see Theorem 5.17. Our result also applies to a variant of the Fiat-Shamir transformation where a digital signature scheme (DSS) is constructed. It thereby, for the first time, enables a multiplicatively tight security reduction for, e.g., DSS based on the MPC-in-the-head paradigm [IKOS07a], like Picnic [CDG+17], Banquet [BSK+21] and Rainier [DKR+21], in the QROM.

A more efficient Unruh transformation. When a Σ-protocol does not have the mentioned C&O structure, a non-interactive proof of knowledge with online extractability in the QROM can be obtained using the Unruh transformation [Unr15b]. For technical reasons, the Unruh transformation requires the hash function to be *length preserving*, which may result in large commitments, and thus large NIZKs and digital signature schemes. We revisit this transformation and show, by a rather direct application of our main result above, that the online extractability of the Unruh transform still holds when using a *compressing* hash function. The crucial observation is that the Unruh transformation can be viewed as the composition of a pre-Unruh transformation, which makes use of hash-based commitments and results in a C&O protocol, and the Fiat-Shamir transformation. By applying our security reduction, we obtain the tight online extractability without requiring the hash function to be length preserving.

More efficient NIZKs via Merkle tree based commitments. In real-world constructions based on C&O protocols, like e.g., the Picnic digital signature scheme, commitments and their openings are responsible for a significant fraction of the signature/proof size. For certain parameters, this cost can be reduced by using a collective commitment mechanism based on Merkle trees. This was observed in passing, e.g. in [Fis05], and is exploited in the most recent versions of Picnic. We formalize Merkle-tree-based C&O protocols and extend our main result to NIZKs constructed from them (see Theorem 5.23). Applications of this result include a security reduction of Picnic 3, the newest version of the Picnic digital signature scheme, that is significantly tighter than existing ones: An adversary against the Picnic 3 signature scheme in the QROM with success probability ε can now be used to break the underlying hard problem with probability ε, up to some additive error terms, while previous reductions yielded at most ε^5/q^{10}, where q is the number of random oracle queries. We outline this reduction in Section 5.5.3.

We compare our reductions in detail to existing techniques in Table 1.

	2-s$\Rightarrow$PoK	PoK$\overset{\text{FS}}{\Rightarrow}$NIZK-PoK, PoK$\overset{\text{FS}}{\Rightarrow}$UF-NMA DSS	2-s$\overset{\text{FS}}{\Rightarrow}$NIZK-PoK, 2-s$\overset{\text{FS}}{\Rightarrow}$UF-NMA DSS
Unruh rewinding [Unr12] + generic FS [DFMS19]	$O(\varepsilon^3)$	$O(\varepsilon/q^2)$	$O(\varepsilon^3/q^6)$
Σ-protocol OE [DFMS22a] + generic FS [DFMS19]	$\varepsilon - g(q,r,n)$	$O(\varepsilon/q^2)$	$O(\varepsilon/q^2) - g(q,r,n)$
this work: **NIZK OE**	-	-	$\boldsymbol{\varepsilon - h(q,r,n)}$

Table 1. Comparison of the losses of different reductions for the construction of a NIZK proof of knowledge (NIZK-PoK) from a special-sound (Merkle tree based) C&O protocol with constant challenge space size C using r-fold parallel repetition and the Fiat-Shamir transformation. "OE" stands for online extraction, 2-s for special soundness, UF-NMA for plain unforgeability and DSS for digital signature scheme. If the content of a cell in row "security property A ⇒ security property B" is $f(\varepsilon)$, this means that an adversary breaking property B with probability ε yields an adversary breaking property A with probabilty $f(\varepsilon)$. Grey text indicates results that do not apply to Merkle-tree-based C&O protocols like the one used to construct the digital signature schemes Picnic 2 [CDG+20] and Picnic 3 [KZ20]. The additive error terms are $g(q,r,n) = C^{-r} + O(rq2^{-n/2}) + O(q^3 2^{-n})$ and $h(q,r,n) = O(q^3 2^{-n}) + O(q^2 C^{-r})$, where n is the output length of the random oracles, and q is the number of adversarial (quantum) queries to the random oracle. Finally, we note that the constants hidden by the big-O in $h(q,r,n)$ are reasonable, see Theorems 5.17 and 5.23.

Technical Overview. Our starting point is the fact that the compressed-oracle technique (Section 2.4) can be appreciated as a variant of the classical lazy-sampling technique that is applicable in the QROM. Namely, to some extent and informally described here, the compressed-oracle technique gives access to a database that contains the hash values that the adversary $\mathcal{A}$, who has interacted with the random oracle (RO), may know. In particular, up to a small error, for any claimed-to-be hash value y output by $\mathcal{A}$, one can find its preimage x by inspecting the database (and one can safely conclude that $\mathcal{A}$ does not know a preimage of y if there is none in the database). Recalling that a C&O Σ-protocol Π (formally defined in Section 2.2.1) is an interactive proof where the first message consists of hash-based commitments, and exploiting that typically some sort of special soundness property ensures that knowing sufficiently many preimages of these commitments/hashes allows one to efficiently compute a witness, constructing an online extractor for the Fiat-Shamir transformation $\mathsf{FS}[\Pi]$ then appears straightforward: The extractor $\mathcal{E}$ simply runs the (possibly dishonest) prover P^*, answering random oracle queries using the compressed oracle. Once P^* has finished and outputs a proof, $\mathcal{E}$ measures the compressed-oracle database and classically reads off any preimages of the commitments in the proof. Finally, $\mathcal{E}$ runs the special soundness extractor that computes a witness from the obtained preimages. It is, however, not obvious that the database contains the preimages of the commitments that are *not* opened in the proof, or that these preimages are correctly formed. Intuitively this should be the case: the random oracle used for the Fiat-Shamir transformation replaces interaction in that it forces the prover to choose a full set of commitments *before* knowing which ones need to be opened. The crux lies in replacing this intuition by a rigorous proof.

The main insight leading to our proof is that the event that needs to be controlled, namely that *the prover succeeds yet the extractor fails*, can be translated into a property SUC (as in "adversarial SUCcess") of the compressed-oracle database, which needs to be satisfied for the event to hold. It is somewhat of a peculiar property though. The database properties that have led to query complexity lower bounds in prior work, e.g. for (multi-)collision finding [LZ19a; HM21; CFHL21] and similar problems [Zha19a; CGLQ20; BLZ21], require the database to contain some particular input-output pairs (e.g. pairs that collide), while the database property SUC additionally *forbids* certain input-output pairs to be contained.

Indeed, the framework from [CFHL21] is almost expressive enough to treat our problem. So, after a mild extension, we can apply it to prove that it is hard for any query algorithm to cause the compressed-oracle database to have property SUC. Analyzing the relevant classical statistical properties of SUC is somewhat tedious but can be done (see the proof of Lemma 5.22). The resulting bound on the probability for the database to satisfy SUC then gives us a bound on the probability of the event that the prover succeeds in producing a valid proof while at the same time fooling the extractor.

Whenever it is advantageous for communication complexity, a Merkle tree can be used to collectively commit to all required messages in a C&O protocol. This collective commitment is one of the optimizations that improve the performance of, e.g. Picnic 2 [CDG+20] over Picnic [CDG+17]. As the above-described argument for the extractability of C&O protocols already analyses iterated hashing (the hash-based commitments are hashed to compute the challenge), it generalizes to Merkle-tree-based C&O protocols without too much effort. We present this generalization in Section 5.5, and obtain similar bounds (see Theorem 5.23).

Additional Related Work. Besides the already mentioned work above, we note that Chiesa, Manohar and Spooner [CMS19] consider and prove security of various SNARG constructions, while we consider the Fiat-Shamir transformation of C&O protocols with a form of special soundness. Similar to [CFHL21], they also provide some tools for deducing security of certain oracle games against quantum attacks by bounding a natural classical variant of the game.

Section 5.2

Preliminaries

Our main technical proofs rely on the recently introduced framework by Chung, Fehr, Huang, and Liao [CFHL21] for proving query complexity bounds in the QROM. This framework exploits Zhandry's compressed-oracle technique but abstracts away all the quantum aspects, so that the reasoning becomes purely classical. We give here an introduction to a simplified, and slightly adjusted version that does not consider parallel queries. We start with recalling (a particular view on) the compressed oracle. Along the way, we also give an improved version of Zhandry's central lemma for the compressed oracle.

Before getting into this, we fix the following standard notation. For any positive integer $\ell > 0$, we set $[\ell] := \{1, 2, \dots, \ell\}$, and we let $2^{[\ell]}$ denote the power set of $[\ell]$, i.e., the set of all subsets of $[\ell]$. We write $\{0,1\}^{\leq \ell}$ for the set of bit strings of size at most ℓ, including the empty string denoted $\emptyset$; similarly for $\{0,1\}^{<\ell}$. Concatenation of two bit strings $v \in \{0,1\}^m$ and $w \in \{0,1\}^n$ is denoted by $v\|w \in \{0,1\}^{m+n}$. For any finite non-empty set $\mathcal{Z}$, $\mathbb{C}[\mathcal{Z}]$ denotes the Hilbert space $\mathbb{C}^{|\mathcal{Z}|}$ together with a basis $\{|z\rangle\}$ labeled by the elements $z \in \mathcal{Z}$.

Finally, we consider a hash function $H : \mathcal{X} \to \mathcal{Y}$, to be modeled as a random oracle. For concreteness and simplicity, we assume that all relevant variables are encoded as bit strings, and that we can therefore choose $H : \{0,1\}^{\leq B} \to \{0,1\}^n$ for sufficiently large B and n.[2]

[2] B and n may depend on the security parameter $\lambda \in \mathbb{N}$. We will then assume that B and n can be computed from λ in polynomial time (in λ).

5.2.1 The Compressed Oracle — Seen as Quantum Lazy Sampling

The compressed oracle technique was formally introduced in Section 2.4. In the current chapter we use slightly different notation in order to better connect with [CFHL21], whose framework we build upon. For purposes of exposition it will also be useful to approach the technique explicitly from the perspective of (quantum) lazy sampling. In this section we give a self-contained introduction of the technique in the form that we will need it later on.

With the goal to analyze oracle algorithms that interact with a random oracle $H : \mathcal{X} \to \mathcal{Y}$, consider the set $\mathfrak{D}$ of all functions $D : \mathcal{X} \to \mathcal{Y} \cup \{\perp\}$, where $\perp$ is a special symbol. Such a function is referred to as a *database*. Later, we will fix $\mathcal{X} = \{0,1\}^{\leq B}$ and $\mathcal{Y} = \{0,1\}^n$. For $D \in \mathfrak{D}$, $x \in \mathcal{X}$ and $y \in \mathcal{Y} \cup \{\perp\}$, $D[x \mapsto y]$ denotes the database that maps x to y and otherwise coincides with D, i.e., $D[x \mapsto y](x) = y$ and $D[x \mapsto y](\bar{x}) = D(\bar{x})$ for all $\bar{x} \in \mathcal{X} \setminus \{x\}$.

Following the exposition of [CFHL21], the compressed-oracle technique is a quantum analogue of the classical lazy-sampling technique, commonly used to analyze algorithms in the classical ROM. In the classical lazy-sampling technique, the (simulated) random oracle starts off with the empty database, i.e., with $D_0 = \perp$, which maps any $x \in \mathcal{X}$ to $\perp$. Then, recursively, upon a query x, the current database D_i is updated to $D_{i+1} := D_i$ if $D_i(x) \neq \perp$, and to $D_{i+1} := D_i[x \mapsto y]$ for a randomly chosen $y \in \mathcal{Y}$ otherwise. This construction ensures that $|\{x \,|\, D_i(x) \neq \perp\}| \leq i$; after i queries thus, using standard sparse-encoding techniques, the database D_i can be efficiently represented and updated.

In the compressed-oracle quantum analogue of this lazy-sampling technique, the (simulated) random oracle also starts off with the empty database, but now considered as a quantum state $|\perp\rangle$ in the $|\mathfrak{D}|$-dimensional state space $\mathbb{C}[\mathfrak{D}]$, and after i queries the state of the compressed oracle is then supported by databases $|D_i\rangle$ for which $|\{x \,|\, D_i(x) = \perp\}| \leq i$.[3] Here, the update is given by a unitary operator cO acting on $\mathbb{C}[\mathcal{X}] \otimes \mathbb{C}[\mathcal{Y}] \otimes \mathbb{C}[\mathfrak{D}]$, i.e., on the query register, the response register, and the state of the compressed oracle. With respect to the computational basis $\{|x\rangle\}$ of $\mathbb{C}[\mathcal{X}]$ and the Fourier basis $\{|\hat{y}\rangle\}$ of $\mathbb{C}[\mathcal{Y}]$, cO is a *control* unitary, i.e., of the form $\mathsf{cO} = \sum_{x,\hat{y}} |x\rangle\langle x| \otimes |\hat{y}\rangle\langle\hat{y}| \otimes \mathsf{cO}_{x,\hat{y}}$, where $\mathsf{cO}_{x,\hat{y}}$ is a unitary on $\mathbb{C}[\mathcal{Y} \cup \{\perp\}]$, which in the above expression is understood to act on the register that carries the value of the database at the point x. More

[3] This means that the density operator that describes the state of the compressed oracle has its support contained in the span of these $|D_i\rangle$.

formally, $\mathsf{cO}_{x,\hat{y}}$ acts on register R_x when identifying $\mathbb{C}[\mathfrak{D}]$ with $\bigotimes_{x\in\mathcal{X}} \mathbb{C}[\mathcal{Y}\cup\{\bot\}]$ by means of the isomorphism $|D\rangle \mapsto \bigotimes_{x\in\mathcal{X}} |D(x)\rangle_{R_x}$. We refer to Lemma 4.3 in the full version of [CFHL21] for the full specification of $\mathsf{cO}_{x,\hat{y}}$; it is not really relevant here.

The compressed oracle is tightly related to the *purified* oracle, which initiates its internal state with a uniform superposition $\sum_h |H\rangle \in \mathbb{C}[\mathfrak{D}]$ of all functions $H : \mathcal{X} \to \mathcal{Y}$, and then answers queries "in superposition". Indeed, at any point in time during the interaction with an oracle quantum algorithm $\mathcal{A}$, the joint state of $\mathcal{A}$ and the compressed oracle coincides with the joint state of $\mathcal{A}$ and the purified oracle after "compressing" the latter.[4] Formally, identifying $\mathbb{C}[\mathfrak{D}]$ with $\bigotimes_{x\in\mathcal{X}} \mathbb{C}[\mathcal{Y}\cup\{\bot\}]$ again, the compression of the state of the purified oracle works by applying the unitary Comp to each register R_x, where

$$\mathsf{Comp} : |y\rangle \mapsto (|y\rangle + \frac{1}{\sqrt{|\mathcal{Y}|}}(|\bot\rangle - |\hat{0}\rangle)$$

for any $y \in \mathcal{Y}$, and $\mathsf{Comp} : |\bot\rangle \mapsto |\hat{0}\rangle$. Here, $|\hat{0}\rangle$ is the $\hat{0}$-vector from the Fourier basis $\{|\hat{y}\rangle\}$ of $\mathbb{C}[\mathcal{Y}]$.

Similarly to the classical case, by exploiting a quantum version of the sparse-encoding technique, both the internal state of the compressed oracle and the evolution cO can be efficiently computed. Furthermore, for any classical function $f : \mathfrak{D} \to \mathcal{T}$ that can be efficiently computed when given the sparse representation of $D \in \mathfrak{D}$, the corresponding quantum measurement given by the projections $P_t = \sum_{D:f(D)=t} |D\rangle\langle D|$ can be efficiently performed when given the sparse representation of the internal state of the compressed oracle. In particular, in Lemma 5.1 below, the condition $\mathbf{y} = D(\mathbf{x})$ for given $\mathbf{x}$ and $\mathbf{y}$ can be efficiently checked by a measurement. See Section 2.4 for more details on this technique.

In the classical lazy-sampling technique, if at the end of the execution of an oracle algorithm $\mathcal{A}$, having made q queries to the (lazy-sampled) RO, the database D_q is such that, say, $D_q(x) \neq 0$ for any $x \in \mathcal{X}$, then $\mathcal{A}$'s output is unlikely to be a 0-preimage, i.e., an x that is hashed to 0 upon one more query. $\mathcal{A}$'s best chance is to output an x that he has not queried yet, and thus $D_q(x) = \bot$, and then he has a $1/|\mathcal{Y}|$-chance that $D_{q+1}(x) := D_q[x \mapsto y](x) = 0$, given that y is randomly chosen. Something similar holds in the quantum setting, with some adjustments. The general statement is given by the following result by Zhandry.

[4] The terminology is somewhat misleading here; the actual compression takes place when invoking the sparse encoding (see below).

Lemma 5.1 (Lemma 5 in [Zha19a]). *Let $R \subseteq \mathcal{X}^\ell \times \mathcal{Y}^\ell \times \mathcal{Z}$ be a relation, and let $\mathcal{A}$ be an oracle quantum algorithm that outputs $\mathbf{x} \in \mathcal{X}^\ell$, $\mathbf{y} \in \mathcal{Y}^\ell$ and $z \in \mathcal{Z}$. Furthermore, let*

$$p = p(\mathcal{A}) := \Pr[\mathbf{y} = H(\mathbf{x}) \wedge (\mathbf{x}, \mathbf{y}, z) \in R]$$

be the considered probability when $\mathcal{A}$ has interacted with the standard RO, initialized with a uniformly random function H, and let

$$p' = p'(\mathcal{A}) := \Pr[\mathbf{y} = D(\mathbf{x}) \wedge (\mathbf{x}, \mathbf{y}, z) \in R]$$

be the considered probability when $\mathcal{A}$ has interacted with the compressed oracle instead and D is obtained by measuring its internal state (in the basis $\{|D\rangle\}_{D \in \mathfrak{D}}$). Then

$$\sqrt{p} \leq \sqrt{p'} + \sqrt{\frac{\ell}{|\mathcal{Y}|}}\,.$$

Remark 5.2. This bound is particular useful in case $\mathcal{Z} = \emptyset$ (or R does not depend on its third input z), since then p' is bounded by $\Pr[\exists\, \tilde{\mathbf{x}} : (\tilde{\mathbf{x}}, D(\tilde{\mathbf{x}})) \in R]$ and the latter is determined solely by the evolution of the compressed oracle (when interacting with $\mathcal{A}$) and does not depend on the actual output of $\mathcal{A}$.

In Section 5.2.3, Corollary 5.8, we will give an alternative such relation between the success probability of an algorithm interacting with the actual RO, and probabilities obtained by inspecting the compressed oracle instead. Strictly speaking, the results of Lemma 5.1 and Corollary 5.8 are incomparable, but in typical applications the latter gives a significantly better bound.

5.2.2 The Quantum Transition Capacity and Its Relevance

The above discussion shows that, in order to bound the success probability p of an oracle algorithm $\mathcal{A}$, it is sufficient to bound the probability of the database D, obtained by measuring the internal state of the compressed oracle after the interaction with $\mathcal{A}$, satisfying a certain property (e.g., the property of there existing an x such that $D(x) = 0$).

To facilitate that latter, Chung et al. [CFHL21] introduced a framework that, in certain cases, allows to bound this alternative figure of merit by means of purely classical reasoning. We briefly recall here some of the core elements of this framework, which are relevant to us. Note that [CFHL21] considers the parallel-query model, where in each of the q (sequential) interactions with the

RO, an oracle algorithm $\mathcal{A}$ can make k queries simultaneously in parallel with each interaction. Here, we consider the (more) standard model of one query per interaction, i.e., setting $k = 1$. On the other hand, we state and prove a slight generalization of Theorem 5.16 in [CFHL21] (when restricted to $k = 1$).

A subset $\mathsf{P} \subseteq \mathfrak{D}$ is called a *database property.* We say that $D \in \mathfrak{D}$ *satisfies* P if $D \in \mathsf{P}$, and the complement of P is denoted $\neg\mathsf{P} = \mathfrak{D} \setminus \mathsf{P}$. For such a database property P, [CFHL21] defines $[\![\bot \overset{q}{\Longrightarrow} \mathsf{P}]\!]$ as the square-root of the maximal probability of D satisfying P when D is obtained by measuring the internal state of the compressed oracle after the interaction with $\mathcal{A}$, maximized over all oracle quantum algorithms $\mathcal{A}$ with query complexity q, i.e., in short

$$[\![\bot \overset{q}{\Longrightarrow} \mathsf{P}]\!] := \max_{\mathcal{A}} \sqrt{\Pr[D \in \mathsf{P}]}\,. \tag{43}$$

In the context of Lemma 5.1 for the case $\mathcal{Z} = \emptyset$ (see Remark 5.2), we can define the database property $\mathsf{P}^R := \{D \in \mathfrak{D} \mid \exists\, \mathbf{x} \in \mathcal{X}^\ell : (\mathbf{x}, D(\mathbf{x})) \in R\}$ induced by R, and thus bound

$$p'(\mathcal{A}) \leq \Pr[(\mathbf{x}, D(\mathbf{x})) \in R] \leq \Pr[D \in \mathsf{P}^R] \leq [\![\bot \overset{q}{\Longrightarrow} \mathsf{P}^R]\!]^2 \tag{44}$$

for any oracle quantum algorithm $\mathcal{A}$ with query complexity q.

Furthermore, Lemma 5.6 in [CFHL21] shows that for any target database property P and for any sequence $\mathsf{P}_0, \mathsf{P}_1, \ldots, \mathsf{P}_q$ with $\neg\mathsf{P}_0 = \{\bot\}$ and $\mathsf{P}_q = \mathsf{P}$,

$$[\![\bot \overset{q}{\Longrightarrow} \mathsf{P}]\!] \leq \sum_{s=1}^{q} [\![\neg\mathsf{P}_{s-1} \to \mathsf{P}_s]\!]\,, \tag{45}$$

where, for any database properties P and P', the definition of the *quantum transition capacity* $[\![\mathsf{P} \to \mathsf{P}']\!]$ is recalled in Definition 5.3.

The nice aspect of the framework of [CFHL21] is that it provides means to manipulate and bound quantum transition capacities using purely classical reasoning, i.e., without the need to understand and work with the definition. Indeed, for instance Theorem 5.4 below, which is a variant of Theorem 5.17 in (the full version of) [CFHL21], shows how to bound $[\![\mathsf{P} \to \mathsf{P}']\!]$ by means of a certain classical probability; furthermore, to facilitate the application of such theorems, [CFHL21] showed that the quantum transition capacity satisfies several natural manipulation rules, like $[\![\mathsf{P} \to \mathsf{P}']\!] = [\![\mathsf{P}' \to \mathsf{P}]\!]$ (i.e., it is symmetric), and

$$\begin{gathered}[\![\mathsf{P} \cap \mathsf{Q} \to \mathsf{P}']\!] \leq \min\big\{ [\![\mathsf{P} \to \mathsf{P}']\!], [\![\mathsf{Q} \to \mathsf{P}']\!] \big\} \qquad \text{and} \\ \min\big\{ [\![\mathsf{P} \to \mathsf{P}']\!], [\![\mathsf{P} \to \mathsf{Q}']\!] \big\} \leq [\![\mathsf{P} \to \mathsf{P}' \cup \mathsf{Q}']\!] \leq [\![\mathsf{P} \to \mathsf{P}']\!] + [\![\mathsf{P} \to \mathsf{Q}']\!]\,,\end{gathered} \tag{46}$$

which allow to decompose complicated capacities into simpler ones. Therefore, by means of the above series of inequalities with p from Lemma 5.1 on the left hand side, it is possible (in certain cases) to bound the success probability of any oracle quantum algorithm $\mathcal{A}$ in the QROM by means of the following recipe: (1) Choose suitable transitions $\mathsf{P}_{s-1} \to \mathsf{P}_s$, (2) decompose the capacities $[\![\neg\mathsf{P}_{s-1} \to \mathsf{P}_s]\!]$ into simpler ones using manipulation rules as above, and (3) bound the simplified capacities by certain classical probabilities, exploiting results like Theorem 5.4. We will closely follow this recipe.

In order to state and later use Theorem 5.4, we need to introduce the following additional concepts. As explained above, there is no need to actually spell out the definition of the quantum transition capacity in order to use Theorem 5.4; for completeness, and since it is needed for the proof of Theorem 5.4, we do provide it below.

For any database $D \in \mathfrak{D}$ and any $x \in \mathcal{X}$,

$$D|^x := \{D[x \mapsto y] \mid y \in \mathcal{Y} \cup \{\bot\}\}$$

denotes the set of all databases that coincide with D outside of x. Furthermore, for a database property P,

$$\mathsf{P}|_{D|^x} := \{y \in \mathcal{Y} \cup \{\bot\} \mid D[x \mapsto y] \in \mathsf{P}\} \subseteq \mathcal{Y} \cup \{\bot\}$$

denotes the set of values y for which $D[x \mapsto y]$ satisfies P. Following the convention used in [CFHL21], we identify the subset $\mathsf{P}|_{D|^x} \subseteq \mathcal{Y} \cup \{\bot\}$ with the projector $\mathsf{P}|_{D|^x} = \sum_y |y\rangle\langle y|$ acting on $\mathbb{C}[\mathcal{Y} \cup \{\bot\}]$, where the sum is over all $y \in \mathsf{P}|_{D|^x}$.

Definition 5.3 (Definition 5.5 of [CFHL21], case $k = 1$**).** *Let* P, P' *be two database properties. Then, the* quantum transition capacity *(of order* 1*) is defined as*

$$[\![\mathsf{P} \to \mathsf{P}']\!] := \max_{\mathbf{x}, \hat{\mathbf{y}}, D} \|\mathsf{P}'|_{D|^{\mathbf{x}}}\, \mathsf{cO}_{\mathbf{x},\hat{\mathbf{y}}}\, \mathsf{P}|_{D|^{\mathbf{x}}}\|$$

where the max is over all $\mathbf{x} \in \mathcal{X}^k$, $\hat{\mathbf{y}} \in \hat{\mathcal{Y}}^k$, *and* $D \in \mathfrak{D}$.

The following is a variation of Theorem 5.17 in (the full version of) [CFHL21], obtained by restricting k to 1. On the other hand, we exploit and include some symmetry that is not explicit in the original statement. The proof is a small adjustment to the original proof.

Theorem 5.4. *Let* P *and* P' *be database properties with trivial intersection, i.e.,* $\mathsf{P} \cap \mathsf{P}' = \emptyset$*, and for every* $D \in \mathfrak{D}$ *and* $x \in \mathcal{X}$ *let*

$$\mathsf{L}^{x,D} := \begin{cases} \mathsf{P}|_{D|^x} & \textit{if } \bot \in \mathsf{P}'|_{D|^x} \\ \mathsf{P}'|_{D|^x} & \textit{if } \bot \in \mathsf{P}|_{D|^x} \end{cases},$$

with $\mathsf{L}^{x,D}$ *being either of the two if* $\bot \notin \mathsf{P}|_{D|^x} \cup \mathsf{P}'|_{D|^x}$.[5] *Then*

$$[\![\mathsf{P} \to \mathsf{P}']\!] \leq \max_{x,D} \sqrt{10P\big[U \in \mathsf{L}^{x,D}\big]}\,,$$

where U *is uniform over* $\mathcal{Y}$*, and the maximization can be restricted to* $D \in \mathfrak{D}$ *and* $x \in \mathcal{X}$ *for which both* $\mathsf{P}|_{D|^x}$ *and* $\mathsf{P}'|_{D|^x}$ *are non-empty.*

Remark 5.5. Both, $\mathsf{P}|_{D|^x}$ and $\mathsf{P}'|_{D|^x}$, and thus also $\mathsf{L}^{x,D}$, do not depend on the value of $D(x)$, only on the values of D outside of x.

Proof. For any $D \in \mathfrak{D}$ and $x \in \mathcal{X}$, we observe that $\mathsf{cO}_{x,-\hat{y}} = (\mathsf{cO}_{x,\hat{y}})^\dagger$ and hence

$$\|\mathsf{P}'|_{D|^x}\, \mathsf{cO}_{x,\hat{y}}\, \mathsf{P}|_{D|^x}\| = \|\left(\mathsf{P}|_{D|^x}\right)^\dagger \mathsf{cO}_{x,-\hat{y}} \left(\mathsf{P}'|_{D|^x}\right)^\dagger\| = \|\mathsf{P}|_{D|^x}\, \mathsf{cO}_{x,-\hat{y}}\, \mathsf{P}'|_{D|^x}\|\,,$$

and so it is sufficient to argue for the case when $\mathsf{L}^{x,D}$ is set to $\mathsf{P}'|_{D|^x}$. By the disjointness requirement, as subsets of $\mathcal{Y} \cup \{\bot\}$, the complement of $\mathsf{L}^{x,D} = \mathsf{P}'|_{D|^x}$ is a superset of $\mathsf{P}|_{D|^x}$. Thus, as projections acting on $\mathbb{C}[\mathcal{Y} \cup \{\bot\}]$, $\mathsf{P}|_{D|^x} \leq \mathbb{1} - \mathsf{L}^{x,D}$. Therefore, the above norm is upper bounded by $\|\mathsf{L}^{x,D}\, \mathsf{cO}_{x,y}\, (\mathbb{1} - \mathsf{L}^{x,D})\|$. Given that $\bot \notin \mathsf{L}^{x,D}$, the square norm $\|\mathsf{L}^{x,D}\, \mathsf{cO}_{x,\hat{y}}\, (\mathbb{1} - \mathsf{L}^{x,D})\|^2$ can be upper bounded exactly as in the proof of Theorem 5.17 in [CFHL21] by $10P\big[U \in \mathsf{L}^{x,D}\big]$, giving the claimed bound. □

5.2.3 An Improved Variant of Zhandry's Lemma

We show here an alternative to Zhandry's lemma (Lemma 5.1), which offers a better bound in typical applications. To start with, note that Lemma 5.1 considers an algorithm $\mathcal{A}$ that not only outputs $\mathbf{x} = (x_1, \ldots, x_\ell)$ but also $\mathbf{y} = (y_1, \ldots, y_\ell)$, where the latter is supposed to be the point-wise hash of $\mathbf{x}$; indeed, this is what is being checked in the definition of the probability p, along with $(\mathbf{x}, \mathbf{y}, z) \in R$. This requirement is somewhat unnatural, in that an algorithm $\mathcal{A}$ for, say, finding a collision, i.e., $x_1 \neq x_2$ with $H(x_1) = H(x_2)$, does *not*

[5] By the disjointness requirement, $\bot$ cannot be contained in both.

necessarily output the (supposed to be equal) hashes $y_1 = H(x_1)$ and $y_2 = H(x_2)$. Of course, this is no problem since one can easily transform such an algorithm $\mathcal{A}$ that does not output the hashes into one that does, simply by making a few more (classical) queries to the random oracle at the end of the execution, and then one can apply Lemma 5.1 to this tweaked algorithm $\tilde{\mathcal{A}}$.

We show below that if we anyway consider this tweaked algorithm $\tilde{\mathcal{A}}$, which is *promised* to query the random oracle to obtain and then output the hashes of $\mathbf{x} = (x_1, \ldots, x_\ell)$, then we can actually improve the bound and avoid the square-roots in Lemma 5.1. On top, the proof is much simpler than Zhandry's proof for his lemma.

At the core is the following lemma; Corollary 5.8 below then puts it in a form that is comparable to Lemma 5.1 and shows the improvement.

Lemma 5.6. *Let $\mathcal{A}$ be an oracle quantum algorithm that outputs $\mathbf{x} = (x_1, ..., x_\ell) \in \mathcal{X}^\ell$ and $z \in \mathcal{Z}$. Let $\tilde{\mathcal{A}}$ be the oracle quantum algorithm that runs $\mathcal{A}$, makes ℓ classical queries on the outputs x_i to obtain $\mathbf{y} = H(\mathbf{x})$, and then outputs $(\mathbf{x}, \mathbf{y}, z)$. When $\tilde{\mathcal{A}}$ interacts with the compressed oracle instead, and at the end D is obtained by measuring the internal state of the compressed oracle, then, conditioned on $\tilde{\mathcal{A}}$'s output $(\mathbf{x}, \mathbf{y}, z)$,*

$$\Pr[\mathbf{y} = D(\mathbf{x}) | (\mathbf{x}, \mathbf{y}, z)] \geq 1 - \frac{2\ell}{|\mathcal{Y}|}\,.$$

Proof. Consider first $\tilde{\mathcal{A}}$ interacting with the *purified* (yet uncompressed) oracle. Conditioned on $\tilde{\mathcal{A}}$'s output $(\mathbf{x}, \mathbf{y}, z)$, the state of the oracle is then supported by $|H\rangle$ with $H(x_i) = y_i$ for all $i \in \{1, \ldots, \ell\}$, i.e., the registers labeled by $x_1, ..., x_\ell$ are in state $|y_1\rangle \cdots |y_\ell\rangle$. Given that the compressed oracle is obtained by applying Comp to all the registers, we thus have that

$$\begin{aligned}\Pr[y_i = y_i' | (\mathbf{x}, \mathbf{y}, z)] &= \big|\langle y_i | \mathsf{Comp} | y_i \rangle\big|^2 = \Big|\langle y_i | \Big(|y_i\rangle + \tfrac{1}{\sqrt{|\mathcal{Y}|}}(|\bot\rangle - |\hat{0}\rangle)\Big)\Big|^2 \\ &= \Big|1 - \tfrac{1}{\sqrt{|\mathcal{Y}|}}\langle y_i | \hat{0}\rangle\Big|^2 = \Big|1 - \tfrac{1}{|\mathcal{Y}|}\Big|^2 \geq 1 - \frac{2}{|\mathcal{Y}|}\,.\end{aligned}$$

Applying union bound concludes the claim. □

The following generalization of Lemma 5.6 follows immediately by enhancing $\mathcal{A}$ so that it computes and outputs all the values x that need to be queried in order to compute $\mathcal{F}^H(z)$, and then apply Lemma 5.6 above.

Corollary 5.7. *Let $\mathcal{A}$ be an oracle quantum algorithm that produces an arbitrary output $z \in \mathcal{Z}$, and let $\mathcal{F}$ be an arbitrary* classical *ℓ-query oracle algorithm. Let $\tilde{\mathcal{A}} := \mathcal{F} \circ \mathcal{A}$ be the oracle quantum algorithm that first runs $\mathcal{A}$ to obtain z, then $\mathcal{F}$ to obtain $y := \mathcal{F}^H(z)$, and finally outputs (y, z). When $\tilde{\mathcal{A}}$ interacts with the compressed oracle instead, and at the end D is obtained by measuring the internal state of the compressed oracle, then, conditioned on $\tilde{\mathcal{A}}$'s output (y, z),*

$$\Pr[y = \mathcal{F}^D(z) | (y, z)] \geq 1 - \frac{2\ell}{|\mathcal{Y}|}.$$

The following corollary of Lemma 5.6 is put in a form that can be nicely compared with Lemma 5.1, understanding that typically Lemma 5.1 is applied to $\tilde{\mathcal{A}}$.

Corollary 5.8. *Let $R \subseteq \mathcal{X}^\ell \times \mathcal{Y}^\ell \times \mathcal{Z}$ be a relation. Let $\mathcal{A}$ be an oracle quantum algorithm that outputs $\mathbf{x} \in \mathcal{X}^\ell$ and $z \in \mathcal{Z}$, and let $\tilde{\mathcal{A}}$ be as in Lemma 5.6. Let*

$$p_\circ(\mathcal{A}) := \Pr[(\mathbf{x}, H(\mathbf{x}), z) \in R]$$

be the considered probability when $\mathcal{A}$ has interacted with the RO. Furthermore, let $p(\tilde{\mathcal{A}})$ and $p'(\tilde{\mathcal{A}})$ be defined as in Lemma 5.1 (but now for $\tilde{\mathcal{A}}$). Then

$$p_\circ(\mathcal{A}) = p(\tilde{\mathcal{A}}) \leq p'(\tilde{\mathcal{A}}) + \frac{2\ell}{|\mathcal{Y}|}.$$

For convenience, we recall that

$$p'(\tilde{\mathcal{A}}) = \Pr[\mathbf{y} = D(\mathbf{x}) \wedge (\mathbf{x}, \mathbf{y}, z) \in R] \leq \Pr[(\mathbf{x}, D(\mathbf{x}), z) \in R].$$

Proof. The equality holds by construction of $\tilde{\mathcal{A}}$. For the first inequality, we observe that

$$\begin{aligned} p'(\tilde{\mathcal{A}}) &= \Pr[\mathbf{y} = D(\mathbf{x}) | (\mathbf{x}, \mathbf{y}, z) \in R] \Pr[(\mathbf{x}, \mathbf{y}, z) \in R] \\ &\geq \big(1 - \tfrac{2\ell}{|\mathcal{Y}|}\big) \Pr[(\mathbf{x}, \mathbf{y}, z) \in R] \geq \big(1 - \tfrac{2\ell}{|\mathcal{Y}|}\big) p(\tilde{\mathcal{A}}) \geq p(\tilde{\mathcal{A}}) - \tfrac{2\ell}{|\mathcal{Y}|}, \end{aligned}$$

where the first inequality is by Lemma 5.6. The second and last inequality in the statement holds trivially by definition of p'. □

Section 5.3

Some Background on (Non-)Interactive Proofs

Let $\{\mathcal{I}_\lambda\}_{\lambda\in\mathbb{N}}$ and $\{\mathcal{W}_\lambda\}_{\lambda\in\mathbb{N}}$ be two families of sets, with the members being labeled by the security parameter $\lambda \in \mathbb{N}$. Let $R_\lambda \subseteq \mathcal{I}_\lambda \times \mathcal{W}_\lambda$ be a relation that is polynomial-time computable in λ. $w \in \mathcal{W}_\lambda$ is called a *witness* for $\mathit{inst} \in \mathcal{I}_\lambda$ if $R_\lambda(\mathit{inst}, w)$, and $L_\lambda := \{\mathit{inst} \in \mathcal{I}_\lambda \mid \exists\, w \in \mathcal{W}_\lambda : R_\lambda(\mathit{inst}, w)\}$.

Below, we recall some concepts in the context of interactive and non-interactive proofs for such families $\{R_\lambda\}_{\lambda\in\mathbb{N}}$ of relations. We start by discussing the aspired security definition for non-interactive proofs.

5.3.1 Non-interactive Proofs and Online Extractability

An *non-interactive proof in the random-oracle model* for a family $\{R_\lambda\}_{\lambda\in\mathbb{N}}$ of relations consists of a pair $(\mathcal{P}, \mathcal{V})$ of oracle algorithms, referred to as *prover* and *verifier*, both making queries to the random oracle $H : \mathcal{X} \to \mathcal{Y}$. The prover $\mathcal{P}$ takes as input $\lambda \in \mathbb{N}$ and an instance $\mathit{inst} \in L_\lambda$ and outputs a *proof* $\pi \in \Pi_\lambda$, and $\mathcal{V}$ takes as input $\lambda \in \mathbb{N}$ and a pair $(\mathit{inst}, \pi) \in \mathcal{I}_\lambda \times \Pi_\lambda$ and outputs a Boolean value, 0 or 1, or `accept` or `reject`. The verifier $\mathcal{V}$ is required to run in time polynomial in λ, while, *per-se*, $\mathcal{P}$ may have unbounded running time.[6]

By default, we require correctness and soundness, i.e., that for any $\lambda \in \mathbb{N}$ and any $\mathit{inst} \in L_\lambda$

$$\Pr\big[\mathcal{V}^H(\lambda, \mathit{inst}, \pi) : \pi \leftarrow \mathcal{P}^H(\lambda, \mathit{inst})\big] \geq 1 - \varepsilon_{\mathrm{cor}}(\lambda),$$

while for any $\lambda \in \mathbb{N}$ and any oracle quantum algorithm $\mathcal{P}^*$ (a *dishonest prover*) with query complexity q

$$\Pr\big[\mathit{inst} \notin L_\lambda \wedge \mathcal{V}^H(\lambda, \mathit{inst}, \pi) : (\mathit{inst}, \pi) \leftarrow \mathcal{P}^{*H}(\lambda)\big] \leq \varepsilon_{\mathrm{snd}}(\lambda, q, n)$$

for certain $\varepsilon_{\mathrm{cor}}$ and $\varepsilon_{\mathrm{snd}}$, respectively referred to as *correctness error* and *soundness error*. The fact that the instance inst, for which $\mathcal{P}^*$ tries to forge a proof, is not given as input to $\mathcal{P}^*$ but is instead chosen by $\mathcal{P}^*$ is referred to as $\mathcal{P}^*$ being *adaptive*.

[6] Alternatively, one may consider a witness w for inst to be given as additional input to $\mathcal{P}$, and then ask $\mathcal{P}$ to be polynomial-time as well.

We now move towards defining *online extractability* (for adaptive $\mathcal{P}^*$). For that purpose, let $\mathcal{P}^*$ be a dishonest prover as above, except that it potentially outputs some additional auxiliary (possibly quantum) output Z next to (inst, π). We then consider an interactive algorithm $\mathcal{E}$, called *online extractor*, which takes $\lambda \in \mathbb{N}$ as input and simulates the answers to the oracle queries in the execution of $\mathcal{V}^H \circ \mathcal{P}^{*H}(\lambda)$, which we define to run $(\mathsf{inst}, \pi, Z) \leftarrow \mathcal{P}^{*H}(\lambda)$ followed by $v \leftarrow \mathcal{V}^H(\lambda, \mathsf{inst}, \pi)$; furthermore, at the end, $\mathcal{E}$ outputs $w \in \mathcal{W}_\lambda$. We denote the execution of $\mathcal{V}^H \circ \mathcal{P}^{*H}(\lambda)$ with the calls to H simulated by $\mathcal{E}$, and considering $\mathcal{E}$'s final output w as well, as $(\mathsf{inst}, \pi, Z; v; w) \leftarrow \mathcal{V}^{\mathcal{E}} \circ \mathcal{P}^{*\mathcal{E}}(\lambda)$.

Definition 5.9. *A non-interactive proof in the (quantum-accessible) random-oracle model (QROM) for $\{R_\lambda\}_{\lambda\in\mathbb{N}}$ is a* proof of knowledge with online extractability *(PoK-OE) against adaptive adversaries if there exists an online extractor $\mathcal{E}$, and functions $\varepsilon_{\mathrm{sim}}$ (the* simulation error*) and $\varepsilon_{\mathrm{ex}}$ (the* extraction error*), with the following properties. For any $\lambda \in \mathbb{N}$ and for any dishonest prover $\mathcal{P}^*$ with query complexity q,*

$$\delta\big([(\mathsf{inst}, \pi, Z, v)]_{\mathcal{V}^H \circ \mathcal{P}^{*H}(\lambda)}, [(\mathsf{inst}, \pi, Z, v)]_{\mathcal{V}^{\mathcal{E}} \circ \mathcal{P}^{*\mathcal{E}}(\lambda)}\big) \leq \varepsilon_{\mathrm{sim}}(\lambda, q, n)$$

and

$$\Pr\big[v = \mathtt{accept} \wedge (\mathsf{inst}, w) \notin R : (\mathsf{inst}, \pi, Z; v; w) \leftarrow \mathcal{V}^{\mathcal{E}} \circ \mathcal{P}^{*\mathcal{E}}(\lambda)\big] \leq \varepsilon_{\mathrm{ex}}(\lambda, q, n)\,.$$

Furthermore, the runtime of $\mathcal{E}$ is polynomial in $\lambda + q + n$, and $\varepsilon_{\mathrm{sim}}(\lambda, q, n)$ and $\varepsilon_{\mathrm{ex}}(\lambda, q, n)$ are negligible in λ whenever q and n are polynomial in λ.

Remark 5.10. In the classical definition of a proof of knowledge, the extractor $\mathcal{E}$ interacts with $\mathcal{P}^*$ only, and the verifier $\mathcal{V}$ is not explicitly involved, but would typic´ally be run by $\mathcal{E}$. Here, in the context of online extractability, it is necessary to explicitly go through the verification procedure, which also makes oracle queries, to determine whether a proof is valid, i.e., for the event $v = \mathtt{accept}$ to be well defined.

5.3.2 $\mathfrak{S}$-soundness of C&O Σ-Protocols

C&O protocols are a subclass of Σ-protocols for which the first message a consists of (hash based) commitments $y_1, \ldots, y_\ell$ for messages $m_1, \ldots, m_\ell \in \mathcal{M}$, and possibly and additional string $a_\circ$. The challenge c is chosen uniformly at random from a subset $\mathcal{C} \subseteq 2^{[\ell]}$ of indices, which point to the messages the prover opens in its response $z = \mathbf{m}_c = (m_i)_{i\in c}$. In this chapter we consider the hash based version where $y_i = H(m_i)$, and we model the hash function H as a random oracle. See Section 2.2.1 for a complete formal specification of C&O protocols.

We briefly recall the notions of $\mathfrak{S}$-soundness and $\mathfrak{S}$-soundness* as developed in Section 4.5.2, which offer a convenient general notion of special soundness, or more generally k-soundness for C&O protocols.

Here and below, given a C&O protocol Π with challenge space $\mathcal{C} \subseteq 2^{[\ell]}$, we let $\mathfrak{S} \subseteq 2^{\mathcal{C}}$ be an arbitrary non-empty, monotone increasing set of subsets $S \subseteq \mathcal{C}$, where the monotonicity means that $S \in \mathfrak{S} \wedge S \subseteq S' \Rightarrow S' \in \mathfrak{S}$. We then also set $\mathfrak{S}_{\min} := \{S \in \mathfrak{S} \mid S_\circ \subsetneq S \Rightarrow S_\circ \notin \mathfrak{S}\}$ to be the minimal sets in $\mathfrak{S}$.

For simplicity, the reader can consider $\mathfrak{S} = \mathfrak{T}_k := \{S \subseteq \mathcal{C} \mid |S| \geq k\}$ for some threshold k, and thus $\mathfrak{S}_{\min} = \{S \subseteq \mathcal{C} \mid |S| = k\}$. This then corresponds to the notion of k-soundness for C&O protocols, which in turn means that the witness can be computed from valid responses to k (or more) distinct challenges for a given first message $y_1, \dots, y_\ell$, assuming the messages $m_1, \dots, m_\ell$ to be uniquely determined by their commitments.

Definition 5.11. *A C&O protocol Π is* $\mathfrak{S}$-sound *if there exists an efficient deterministic algorithm $\mathcal{E}_{\mathfrak{S}}(\mathsf{inst}, m_1, \dots, m_\ell, a_\circ, S)$ that takes as input an instance $\mathsf{inst} \in \mathcal{I}$, messages $m_1, \dots, m_\ell \in \mathcal{M} \cup \{\perp\}$, a string $a_\circ$, and a set $S \in \mathfrak{S}_{\min}$, and outputs a witness for inst if $V(\mathsf{inst}, c, \mathbf{m}_c, a_\circ)$ for all $c \in S$.*[7]

We note the clash in terminology with Definition 4.22. However, the current definition applies exclusively to C&O Σ-protocols in the (Q)ROM, whereas Definition 4.22 applies exclusively to Σ-protocols in the standard model; so there should be no confusion. The two definitions are of course related: a $\mathfrak{S}$-sound C&O Σ-protocol becomes a $\mathfrak{S}$-sound plain Σ-protocol when the commitments are instantiated with a perfectly binding commitment scheme (rather than with a hash function).

A slightly stronger condition than $\mathfrak{S}$-*soundness* is the following variant, which differs in that the extractor needs to work as soon as there *exists* a set S as specified, without the extractor being given S as input. We refer to Section 4.5.2 for a more detailed discussion of this aspect. As explained there, whether S is given or not often makes no (big) difference.

For instance, when $\mathfrak{S}_{\min}$ consists of a polynomial number of sets S then the extractor can do a brute-force search to find S, and so $\mathfrak{S}$-soundness* is

[7] The restriction for S to be in $\mathfrak{S}_{\min}$, rather than in $\mathfrak{S}$, is to avoid an exponentially sized input while asking $\mathcal{E}_{\mathfrak{S}}$ to be efficient.

then implied by $\mathfrak{S}$-soundness. Also, the r-fold parallel repetition of a $\mathfrak{S}$-sound protocol, which by default is a $\mathfrak{S}^{\vee r}$-sound protocol (see Section 4.5.2), is automatically $\mathfrak{S}^{\vee}$-sound* if $\mathfrak{S}_{\min}$ is polynomial in size: the extractor can then do a brute-force search in every repeated instance.

Definition 5.12. *A C&O protocol Π is* $\mathfrak{S}$-sound* *if there exists an efficient deterministic algorithm $\mathcal{E}^*_{\mathfrak{S}}(\mathsf{inst}, m_1, \ldots, m_\ell, a_\circ)$ that takes as input an instance $\mathsf{inst} \in \mathcal{I}$ and strings $m_1, \ldots, m_\ell \in \mathcal{M} \cup \{\bot\}$ and $a_\circ$, and it outputs a witness for inst if there exists $S \in \mathfrak{S}$ such that $V(\mathsf{inst}, c, \mathbf{m}_c, a_\circ)$ for all $c \in S$.*

As for plain Σ-protocols, we define

$$p^{\mathfrak{S}}_{triv} := \frac{1}{|\mathcal{C}|} \max_{\hat{S} \notin \mathfrak{S}} |\hat{S}| \,, \tag{47}$$

capturing the "trivial" attack of picking a set $\hat{S} = \{\hat{c}_1, \ldots, \hat{c}_m\} \notin \mathfrak{S}$ of challenges $\hat{c}_i \in \mathcal{C}$ and then prepare $\hat{\mathbf{m}} = (\hat{m}_1, \ldots, \hat{m}_\ell)$ and $a_\circ$ in such a way that $V(\mathsf{inst}, c, \hat{\mathbf{m}}_c, a_\circ)$ holds if $c \in \hat{S}$. After committing to $\hat{m}_1, \ldots, \hat{m}_\ell$, the prover can successfully answer to challenges $c \in \hat{S}$.

5.3.3 The Fiat-Shamir Transformation of (C&O) Σ-Protocols

The Fiat-Shamir (FS) transformation [FS87] turns arbitrary Σ-protocols into non-interactive proofs in the random oracle model by setting the challenge $c \in \mathcal{C}$ to be the hash of the instance and the first message a. For this transformation to work smoothly, it is typically assumed that $|\mathcal{C}|$ is a power of 2 and its elements are represented as bit strings of size $\log|\mathcal{C}|$, so that one can indeed set c to be (the first $\log|\mathcal{C}|$ bits of) the hash $H(\mathsf{inst}, a)$. The assumption on $|\mathcal{C}|$ is essentially without loss of generality (WLOG), since one can always reduce the size of $|\mathcal{C}|$ to the next lower power of 2, at the cost of losing at most 1 bit of security. However, for a C&O Σ-protocol, where a challenge space $\mathcal{C}$ is a (typically strict) subset of $2^{[\ell]}$, there is not necessarily a natural way to represent $c \in \mathcal{C}$ as a bitstring of size $\log|\mathcal{C}|$. Therefore, we will make it explicit that the challenge-set $c \in \mathcal{C} \subset 2^{[\ell]}$ is computed from the "raw randomness" $H(\mathsf{inst}, y_1, \ldots, y_\ell, a_\circ)$ in a deterministic way as $c = \gamma \circ H(\mathsf{inst}, y_1, \ldots, y_\ell, a_\circ)$ for an appropriate function $\gamma : \mathcal{Y} \to \mathcal{C}$, mapping a uniformly random hash in $\mathcal{Y}$ to a random challenge-set in $\mathcal{C}$. Obviously, for $H(\mathsf{inst}, y_1, \ldots, y_\ell, a_\circ)$ to be defined, in addition to $\mathcal{M} \subseteq \mathcal{X}$ we also need that $\mathcal{I} \times \mathcal{Y}^\ell \subseteq \mathcal{X}$, which again just means that B needs to be large enough. We write $\mathsf{FS}[\Pi]$ for the Fiat-Shamir transformation of a (C&O) Σ-protocol Π.

Remark 5.13. Additionally, we need that n is sufficiently large, so that there is a sufficient amount of randomness in the hash value $H(\mathit{inst}, y_1, \dots, y_\ell)$ in order to be mapped to a random $c \in \mathcal{C}$. The canonical choice for γ is then the function that the *interactive* verifier applies to his local randomness to compute the random challenge $c \in \mathcal{C}$. To simplify the exposition, we assume that n is indeed sufficiently large. Otherwise, one can simply set $\mathcal{Y} := \{0,1\}^{n'}$ instead, for sufficiently large n', and then let y_i be $H(m_i)$ *truncated* to the original number n of bits again. This truncation has no effect on our results.

Remark 5.14. We assume WLOG that the two kinds of inputs to H, i.e., m_i and $(\mathit{inst}, y_1, \dots, y_\ell, a_\circ)$, are differently formatted, e.g., bit strings of different respective sizes or prefixes (this is referred to as *domain separation*). In other words, we assume that $\mathcal{M}$ and $\mathcal{I} \times \mathcal{Y}^\ell$ are disjoint.

Remark 5.15. When considering the adaptive security of a Fiat-Shamir transformation $\mathsf{FS}[\Pi]$ of a C&O protocol Π for a relation R, the additional string $a_\circ$, which may be part of the first message a of the original protocol Π, may WLOG be considered to be part of the instance inst instead.

Indeed, any dishonest prover $\mathcal{P}^*$ against $\mathsf{FS}[\Pi]$, which (by Definition 5.9) outputs an instance inst and a proof $\pi = (a_\circ, y_1, \dots y_\ell)$, can alternatively be parsed as a dishonest prover that outputs an instance $\mathit{inst}' = (\mathit{inst}, a_\circ)$ and a proof $\pi' = (y_1, \dots y_\ell)$. Thus, $\mathcal{P}^*$ can be parsed as a dishonest prover against $\mathsf{FS}[\Pi']$, where the C&O protocol Π' works as Π, except that $a_\circ$ is considered as part of the instance, rather than as part of the first message, and thus Π' is a C&O protocol for the relation $((\mathit{inst}, a_\circ), w) \in R' :\Leftrightarrow (\mathit{inst}, w) \in R$.[8] Therefore, security (in the sense of Definition 5.9) for $\mathsf{FS}[\Pi']$ implies that of $\mathsf{FS}[\Pi]$.

Section 5.4

Online Extractability of the FS-Transformation: The Case of Ordinary C&O Protocols

We now consider the Fiat-Shamir transformation $\mathsf{FS}[\Pi]$ of an ordinary C&O protocol Π. Our goal is to show that $\mathsf{FS}[\Pi]$ admits online extraction. We note

[8] We do not specify the local computation of the honest prover $\mathcal{P}'$ in $\Pi' = (\mathcal{P}', \mathcal{V}')$, i.e., how to act when $a_\circ$ is part of the input, and in general it might not be efficient, but this is fine since we are interested in the security against dishonest provers.

that by exploiting Remark 5.15, we may assume WLOG that the first message of $\mathit{\Pi}$ consists of the commitments $y_1, \dots, y_\ell$ only, and no additional string $a_\circ$. In Section 5.5, we then consider the case of Merkle-tree-based C&O protocols.

Our analysis of the online extractability of $\mathsf{FS}[\mathit{\Pi}]$ uses the framework of Chung et al. [CFHL21], discussed and outlined in Section 5.2. Thus, at the core of our analysis is a bound on a certain quantum transition capacity. This is treated in the upcoming subsection.

5.4.1 Technical Preface

We first introduce a couple of elementary database properties (related to CoLlisions and the SiZe of the database) that will be useful for us:

$$\mathsf{CL} := \{D \,|\, \exists\, x \neq x' : D(x) = D(x') \neq \bot\} \quad \text{and} \quad \mathsf{SZ}_{\leq s} := \{D \,|\, \#\{z | D(z) \neq \bot\} \leq s\}.$$

Next, for an instance $\mathit{inst} \in \mathcal{I}$, we want to specify the database property that captures a cheating prover that succeeds in producing an accepting proof while fooling the extractor. For the purpose of specifying this database property, we introduce the following notation. For a given database $D \in \mathfrak{D}$ and for a commitment $y \in \mathcal{Y}$, we define $D^{-1}(y)$ to be the smallest $x \in \mathcal{X}$ with $D(x) = y$, with the convention that $D^{-1}(y) := \bot$ if there is no such x, as well as $D^{-1}(\bot) := \bot$. We note that by removing collisions, we ensure that there is at most one such x; thus, taking the smallest one in case of multiple choices is not important but only for well-definedness.

The database property of interest can now be defined as

$$\mathsf{SUC} := \left\{ D \;\middle|\; \begin{array}{c} \exists\, \mathbf{y} \in \mathcal{Y}^\ell \text{ and } \mathit{inst} \in \mathcal{I} \text{ so that } \mathbf{m} := D^{-1}(\mathbf{y}) \text{ satisfies} \\ V(\mathit{inst}, c, \mathbf{m}_c) \text{ for } c := \gamma \circ D(\mathit{inst}, \mathbf{y}) \text{ and } \big(\mathit{inst}, \mathcal{E}^*(\mathit{inst}, \mathbf{m})\big) \notin R \end{array} \right\}. \tag{48}$$

Informally, assuming no collisions (i.e., restricting to $D \notin \mathsf{CL}$), the database property SUC captures whether a database D admits a *valid* proof $\pi = (\mathbf{y}, \mathbf{m}_c)$ for an instance inst for which the (canonical) extractor, which first computes $\mathbf{m}$ by inverting D and then runs $\mathcal{E}^*$, *fails* to produce a witness.

Our (first) goal is to show that $[\![\bot \overset{q}{\Longrightarrow} \mathsf{SUC} \cup \mathsf{CL}]\!]$ is small, capturing that it is unlikely that after q queries the compressed database contains collisions or admits a valid proof upon which the extractor fails. Indeed, we show the following, where $p_{triv}^{\mathfrak{S}}$ is the trivial cheating probability of $\mathit{\Pi}$ as defined in (47).

Lemma 5.16. $[\![\bot \overset{q}{\Longrightarrow} \mathsf{SUC} \cup \mathsf{CL}]\!] \leq 2eq^{3/2}2^{-n/2} + q\sqrt{10 \max\left(q\ell \cdot 2^{-n}, p_{triv}^{\mathfrak{S}}\right)}$.

The formal proof is given below; we first give some informal outline here. In a first step, by using (45) and union-bound-like properties of the transition capacity, and additionally exploiting a bound from [CFHL21] to control the transition capacity of CL, we reduce the problem to bounding the quantum transition capacity $[\![\mathsf{SZ}_{\leq s} \backslash \mathsf{SUC} \to \mathsf{SUC}]\!]$ for $s < q$. Informally, this capacity is a measure of the "likelihood" — but then in a *quantum*-sense — that a database $D \in \mathfrak{D}$ that is bounded in size and not in SUC turns into a database D' that *is* in SUC, when D is updated to $D' = D[x \mapsto U]$ with U uniformly random in $\mathcal{Y}$.

We emphasize that in the considered quantum setting, the state of the compressed oracle at any point is a *superposition* of databases, and a query is made up of a *superposition* of inputs; nevertheless, due to Theorem 5.4, the above classical intuition is actually very close to what needs to be shown to rigorously bound the considered quantum transition capacity. Formally, as will become clear in the proof below, we need to show that for any database $D \in \mathsf{SZ}_{\leq s} \backslash \mathsf{SUC}$ and for any $x \in \mathcal{X}$ with $D(x) = \bot$, the probability that $D[x \mapsto U] \in \mathsf{SUC}$ is small. Below, this probability is bounded in the *Case 2* and *Case 3* parts of the proof, where the two cases distinguish between x being a "commit query" or a "challenge query".

Informally, for D with $D(x) = \bot$, if x is a "commit query" then assigning a value to $D(x)$ can only make a difference, i.e., turn $D \notin \mathsf{SUC}$ into $D[x \mapsto u] \in \mathsf{SUC}$, if u is a coordinate of some $\mathbf{y} \in \mathcal{Y}^\ell$ for which $D(\mathit{inst}, \mathbf{y}) \neq \bot$ for some inst. Indeed, otherwise, $D[x \mapsto u]$ does not contribute to a valid proof π that did not exist before. Thus, given the bound $s < q$ on the size of D, this happens with probability at most $q\ell/2^n$ for a random u. Similarly, if x is a "challenge query", i.e. of the form $x = (\mathit{inst}, \mathbf{y})$, then assigning a value u to $D(x)$ can only make a difference if $V(\mathit{inst}, c, \mathbf{m}_c)$ is satisfied for $c = \gamma(u)$ and $\mathbf{m} = D^{-1}(\mathbf{y})$, while $\mathcal{E}^*(\mathit{inst}, \mathbf{m})$ is not a witness for inst. However, for a random u, this is bounded by $p^{\mathfrak{S}}_{triv}$.

But then, on top of the above, due to the quantum nature of the quantum transition capacity,[9] Theorem 5.4 requires to also show the "reverse", i.e., that for any $D \in \mathsf{SUC}$ and for any $x \in \mathcal{X}$ with $D(x) \neq \bot$, the probability that $D[x \mapsto U] \in \mathsf{SZ}_{\leq s} \backslash \mathsf{SUC}$ is small; this is analyzed in *Case 1* below.

Thus, by exploiting the framework of [CFHL21], the core of the reasoning is purely classical, very closely mimicking how one would have to reason the classical setting with a classical RO. Due to the rather complex definition of SUC, the formal argument in each case is still somewhat cumbersome.

[9] At the core, this is related to the reversibility of quantum computing and the resulting ability to "uncompute" a query.

Proof. We first observe that, by (45) (which is Lemma 5.6 in [CFHL21]) and basic properties of the quantum transition capacity as in (46),

$$\begin{aligned}\llbracket \bot \overset{q}{\Longrightarrow} \mathsf{SUC} \cup \mathsf{CL} \rrbracket &\le \sum_{s=0}^{q-1} \llbracket \mathsf{SZ}_{\le s} \backslash \mathsf{SUC} \backslash \mathsf{CL} \to \mathsf{SUC} \cup \mathsf{CL} \cup \neg \mathsf{SZ}_{\le s+1} \rrbracket \\ &\le \sum_{s=0}^{q-1} \big(\llbracket \mathsf{SZ}_{\le s} \backslash \mathsf{SUC} \backslash \mathsf{CL} \to \neg \mathsf{SZ}_{\le s+1} \rrbracket + \llbracket \mathsf{SZ}_{\le s} \backslash \mathsf{SUC} \backslash \mathsf{CL} \to \mathsf{CL} \rrbracket \\ &\qquad\qquad + \llbracket \mathsf{SZ}_{\le s} \backslash \mathsf{SUC} \backslash \mathsf{CL} \to \mathsf{SUC} \rrbracket \big) \\ &\le \sum_{s=0}^{q-1} \big(\llbracket \mathsf{SZ}_{\le s} \to \neg \mathsf{SZ}_{\le s+1} \rrbracket + \llbracket \mathsf{SZ}_{\le s} \backslash \mathsf{CL} \to \mathsf{CL} \rrbracket + \llbracket \mathsf{SZ}_{\le s} \backslash \mathsf{SUC} \to \mathsf{SUC} \rrbracket \big) .\end{aligned} \tag{49}$$

The first term, $\llbracket \mathsf{SZ}_{\le s} \to \neg \mathsf{SZ}_{\le s+1} \rrbracket$, vanishes, while the second term was shown to be bounded as

$$\llbracket \mathsf{SZ}_{\le s} \backslash \mathsf{CL} \to \mathsf{CL} \rrbracket \le 2e\sqrt{(s+1)/|\mathcal{Y}|} \le 2e\sqrt{q/2^n} \tag{50}$$

in Example 5.28 in [CFHL21]. Thus, it remains to control the third term, which we will do by means of Theorem 5.4 with $\mathsf{P} := \mathsf{SZ}_{\le s} \setminus \mathsf{SUC}$ and $\mathsf{P}' := \mathsf{SUC}$.

To this end, we consider arbitrary but fixed $D \in \mathfrak{D}$ and input $x \in \mathcal{X}$. By Remark 5.5, we may assume that $D(x) = \bot$. Furthermore, for $\mathsf{P}|_{D|^x}$ to be non-empty, it must be that $D \in \mathsf{SZ}_{\le s}$, i.e., D is bounded in size. We now distinguish between the following cases for the considered D and x.

Case 1: $D \in \mathsf{SUC}$. In particular, $\bot \in \mathsf{SUC}|_{D|^x} = \mathsf{P}'_{D|^x}$. So, Theorem 5.4 instructs us to set $:= \mathsf{P}_{D|^x}$, where we leave the dependency of on D and x implicit to simplify notation. Given that $D \in \mathsf{SUC}$, we can consider inst and $\mathbf{y}$ as promised by the definition of SUC in (48), i.e., such that $V(\mathit{inst}, c, \mathbf{m}_c)$ and $\big(\mathit{inst}, \mathcal{E}^*(\mathit{inst}, \mathbf{m})\big) \notin R$ for

$$c := \gamma \circ D(\mathit{inst}, \mathbf{y}) \quad \text{and} \quad m_i := D^{-1}(y_i) ,$$

where it is understood that $\mathbf{m} = (m_1, \ldots, m_\ell)$. Recall that $D(x) = \bot$; thus, by definition of the m_i's, it must be that $x \ne m_i$ for all i, and the fact that $V(\mathit{inst}, c, \mathbf{m}_c)$ is satisfied for c as defined implies that $x \ne (\mathit{inst}, \mathbf{y})$. Furthermore,

$$u \in \mathsf{L} \iff D[x \mapsto u] \in \mathsf{P} \implies D[x \mapsto u] \notin \mathsf{SUC} \implies u \in \{y_1, \ldots, y_\ell\} ,$$

where the last implication is easiest seen by contraposition: Assume that $u \notin \{y_1, \ldots, y_\ell\}$. Then, also recalling that $x \ne m_i$, we have that $m_i = D^{-1}(y_i) =$

$D[x \mapsto u]^{-1}(y_i)$. But also $c = \gamma \circ D(\mathit{inst}, \mathbf{y}) = \gamma \circ D[x \mapsto u](\mathit{inst}, \mathbf{y})$. Together, this implies that the defining property of SUC is also satisfied for $D[x \mapsto u]$, i.e., $D[x \mapsto u] \in \mathsf{SUC}$, as was to be shown. Thus, we can bound

$$P[U \in] \le P[U \in \{y_1, \dots, y_\ell\}] \le \frac{\ell}{|\mathcal{Y}|} \,. \tag{51}$$

Case 2: $D \notin \mathsf{SUC}$, and x is a "commit query", i.e., $x = m \in \mathcal{M}$. In particular, $\bot \notin \mathsf{P}'|_{D|^x}$ (by the assumption that $D(x) = \bot$) and so in light of Theorem 5.4 we may choose $\mathsf{L} := \mathsf{P}'|_{D|^x}$. We then have

$$u \in \mathsf{L} \iff D[x \mapsto u] \in \mathsf{P}' = \mathsf{SUC} \implies \exists\, \mathit{inst}, \mathbf{y}, i : D(\mathit{inst}, \mathbf{y}) \ne \bot \,\wedge\, u = y_i \,. \tag{52}$$

This final implication can be seen as follows. By definition of SUC, the assumption $D[x \mapsto u] \in \mathsf{SUC}$ implies the existence of inst and $\mathbf{y} = (y_1, \dots, y_\ell)$ with $V(\mathit{inst}, c, \mathbf{m}_c)$ and $\big(\mathit{inst}, \mathcal{E}^*(\mathit{inst}, \mathbf{m})\big) \notin R$ for

$$c := \gamma \circ D[x \mapsto u](\mathit{inst}, \mathbf{y}) = \gamma \circ D(\mathit{inst}, \mathbf{y}) \quad \text{and} \quad m_i := D[x \mapsto u]^{-1}(y_i)\,,$$

where the equality in the definition of c exploits that x is not a "challenge" query. With the goal to reach a contradiction, assume that $u \ne y_i$ for all i. This assumption implies that $D[x \mapsto u](x) = u \ne y_i$. But also $D(x) = \bot \ne y_i$, and hence for all $\xi \in \mathcal{X}$ and $i \in \{1, \dots, \ell\}$: $D(\xi) = y_i \Leftrightarrow D[x \mapsto u](\xi) = y_i$. Therefore, $m_i = D[x \mapsto u]^{-1}(y_i) = D^{-1}(y_i)$ for all i, and the above then implies that $D \in \mathsf{SUC}$, a contradiction. Thus, there exists i for which $u = y_i$; furthermore, $D(\mathit{inst}, \mathbf{y}) \ne \bot$ given that $V(\mathit{inst}, u, \mathbf{m}_c)$ is satisfied for $c = \gamma \circ D(\mathit{inst}, \mathbf{y})$. This shows the claimed implication.

Thus, we can bound

$$P[U \in] \le P[\exists\, \mathit{inst}, \mathbf{y}, i : D(\mathit{inst}, \mathbf{y}) \ne \bot \,\wedge\, u = y_i] \le \frac{s\ell}{|\mathcal{Y}|} \le \frac{q\ell}{|\mathcal{Y}|} \,. \tag{53}$$

Case 3: $D \notin \mathsf{SUC}$, and x is a "challenge query", i.e., $x = (\mathit{inst}, \mathbf{y}) \in \mathcal{I} \times \mathcal{Y}^\ell$. Set $\mathbf{m} = (m_1, \dots, m_\ell)$ for $m_i := D^{-1}(y_i)$. Again, we have that $\bot \notin \mathsf{SUC}|_{D|^x} = \mathsf{P}'_{D|^x}$, and so by Theorem 5.4 we may set $:= \mathsf{P}'_{D|^x}$. Here, we can argue that

$$\begin{aligned} u \in \mathsf{L} &\iff D[x \mapsto u] \in \mathsf{P}' = \mathsf{SUC} \\ &\implies V(\mathit{inst}, u, \mathbf{m}_{\gamma(u)}) \text{ and } \big(\mathit{inst}, \mathcal{E}^*(\mathit{inst}, \mathbf{m})\big) \notin R\,, \end{aligned}$$

where the final implication can be seen as follows. By definition of SUC, the assumption $D[x \mapsto u] \in \mathsf{SUC}$ implies the existence of inst' and $\mathbf{y}' = (y'_1, \dots, y'_\ell)$

with $V(\mathsf{inst}', u, \mathbf{m}'_c)$ and $\mathcal{E}^*(\mathsf{inst}', \mathbf{m}') \neq w$ for

$$c := \gamma \circ D[x \mapsto u](\mathsf{inst}', \mathbf{y}') \quad \text{and} \quad m'_i := D[x \mapsto u]^{-1}(y'_i) = D^{-1}(y'_i)\,,$$

where the very last equality exploits that x is not a “commit” query. With the goal to come to a contradiction, assume that $(\mathsf{inst}', \mathbf{y}') \neq (\mathsf{inst}, \mathbf{y}) = x$. Then, $c = \gamma \circ D[x \mapsto u](\mathsf{inst}', \mathbf{y}') = \gamma \circ D(\mathsf{inst}', \mathbf{y}')$, and the above then implies that $D \in \mathsf{SUC}$, a contradiction. Thus, $(\mathsf{inst}', \mathbf{y}') = (\mathsf{inst}, \mathbf{y}) = x$. In particular, $\mathbf{m}' = \mathbf{m}$ and $c = \gamma \circ D[x \mapsto u](\mathsf{inst}', \mathbf{y}') = \gamma \circ D[x \mapsto u](x) = \gamma(u)$. Hence, the claimed implication holds.

Thus, we can bound

$$\begin{aligned}
P[U \in] &\leq P[V(\mathsf{inst}, \gamma(U), \mathbf{m}_{\gamma(U)}) \wedge \mathcal{E}^*(\mathsf{inst}, \mathbf{m}) \neq w] \\
&\leq P[V(\mathsf{inst}, \gamma(U), \mathbf{m}_{\gamma(U)}) \wedge S := \{c \,|\, V(\mathsf{inst}, c, \mathbf{m}_c)\} \notin \mathfrak{S}] \\
&\leq P[\gamma(U) \in S := \{c \,|\, V(\mathsf{inst}, c, \mathbf{m}_c)\} \notin \mathfrak{S}] \\
&\leq \max_{S \notin \mathfrak{S}} P[\gamma(U) \in S] \\
&\leq p^{\mathfrak{S}}_{triv}\,. \qquad (54)
\end{aligned}$$

By Theorem 5.4, we now get

$$\begin{aligned}
[\![\mathsf{SZ}_{\leq s} \backslash \mathsf{SUC} \backslash \mathsf{CL} \to \mathsf{SUC}]\!] &\leq \max_{x,D} \sqrt{10 P\big[U \in \mathsf{L}^{x,D}\big]} \\
&\leq \sqrt{10} \sqrt{\max\left(\frac{\ell}{|\mathcal{Y}|}, \frac{q\ell}{|\mathcal{Y}|}, p^{\mathfrak{S}}_{triv}\right)} \\
&\leq \sqrt{10} \sqrt{\max\left(q\ell \cdot 2^{-n}, p^{\mathfrak{S}}_{triv}\right)},
\end{aligned}$$

where we have used Equations (51), (53) and (54) in the second inequality. Combining with Equations (50) and (49) yields the desired bound. □

5.4.2 Online Extractability of the Fiat-Shamir Transformation

We are now ready to state and proof the claimed online-extractability result for the Fiat-Shamir transformation of (ordinary) C&O protocols.

Theorem 5.17. *Let Π be a $\mathfrak{S}$-sound* ordinary C&O protocol with challenge space $\mathcal{C}_\lambda$ and $\ell = \ell(\lambda)$ commitments, and set $\kappa = \kappa(\lambda) := \max_{c \in \mathcal{C}_\lambda} |c|$. Then,*

$\mathsf{FS}[\Pi]$ is a proof of knowledge with online extractability in the QROM (as in Definition 5.9), with $\varepsilon_{\text{sim}}(\lambda, q, n) = 0$ and

$$\begin{aligned}\varepsilon_{\text{ex}}(\lambda, q, n) &\leq 2(\kappa+1)\cdot 2^{-n} + \left(2eq^{3/2}2^{-n/2} + q\sqrt{10\max\left(q\ell\cdot 2^{-n}, p^{\mathfrak{S}}_{triv}\right)}\right)^2 \\ &\leq (22\ell+60)q^3 2^{-n} + 20q^2 p^{\mathfrak{S}}_{triv}\,.\end{aligned}$$

The runtime of the extractor is dominated by running the compressed oracle, which has complexity $O(q^2)\cdot poly(n, B)$, and running $\mathcal{E}^*$.

We note that the above bound on ε_{ex} is asymptotically tight, except for the factor ℓ. Indeed, the binding property of the hash-based commitment can be invalidated by means of a collision finding attack, which succeeds with probability $\Omega(q^3/2^n)$. Furthermore the trivial soundness attack, which potentially applies to a $\mathfrak{S}$-sound* C&O protocol Π, can be complemented with a Grover search, yielding an attack against $\mathsf{FS}[\Pi]$ that succeeds with probability $\Omega(q^2 p^{\mathfrak{S}}_{triv})$. The non-tightness by a factor of ℓ is very mild in most cases. In particular, the number of commitments ℓ is polynomial in λ and thus in n. For the most common case of a parallel repetition of a protocol with a constant number of commitments, using a hash function with output length linear in λ (e.g. $n = 3\lambda$) results in $\ell = O(n) = O(\lambda)$.

Proof. We consider an arbitrary but fixed $\lambda \in \mathbb{N}$. For simplicity, we assume that $|c|$ is the same for all $c \in \mathcal{C}_\lambda$, and thus equal to $\kappa = \kappa(\lambda)$. If it is not, we could always make the prover output a couple of dummy outputs m_i to match the upper bound on $|c|$. Let $\mathcal{P}^*$ be a dishonest prover that, after making q queries to a random oracle H, outputs $(\mathit{inst}, \pi) = (\mathit{inst}, \mathbf{y}, \mathbf{m}_\circ)$ plus some (possibly quantum) auxiliary output Z. In the experiment $\mathcal{V}^{\mathcal{E}} \circ \mathcal{P}^{*\mathcal{E}}(\lambda)$, our extractor $\mathcal{E}$ works as follows while simulating all queries to H (by $\mathcal{P}^*$ and $\mathcal{V}$) with the compressed oracle:

1. Run $\mathcal{P}^*(\lambda)$ to obtain (inst, π, Z) where $\pi = (\mathbf{y}, \mathbf{m}_\circ)$ with $\mathbf{m}_\circ = (m_1, \ldots, m_\kappa)$.
2. Run $\mathcal{V}(\lambda, \mathit{inst}, \pi)$ to obtain v. In detail: obtain $h_0 := H(\mathit{inst}, \mathbf{y})$ and $h_j := H(m_j)$ for $j \in \{1, \ldots, \kappa\}$, and set $v :=$ `accept` if and only if the pair consisting of $\mathbf{x} = \big((\mathit{inst}, \mathbf{y}), m_1, \ldots, m_\kappa\big)$ and $\mathbf{h} = (h_0, h_1, \ldots, h_\kappa)$ satisfies the relation $\tilde{R}$, defined to hold if and only if

$$(h_1, \ldots, h_\kappa) = \mathbf{y}_c \quad \wedge \quad V(\mathit{inst}, c, \mathbf{m}_\circ) \quad \text{where} \quad c := \gamma(h_0)\,.$$

3. Measure the internal state of the compressed oracle to obtain D.

4. Run $\mathcal{E}^*(\mathit{inst}, \mathbf{m})$ on input inst and $\mathbf{m} := D^{-1}(\mathbf{y})$ to obtain w.

Note that in the views of both $\mathcal{P}^*$ and $\mathcal{V}$, the interaction with H and the interaction with $\mathcal{E}$ differ only in that their oracle queries are answered by a compressed oracle instead of a real random-oracle in the latter case. This simulation is perfect and therefore $\varepsilon_{\text{sim}}(\lambda, q, n) = 0$.

Considering $\mathcal{P}^*$ as the algorithm $\mathcal{A}$ in Lemma 5.6, the additional classical oracle queries that $\mathcal{V}$ performs in $\mathcal{V} \circ \mathcal{P}^*$ then match up with the algorithm $\tilde{\mathcal{A}}$, with $h_0, \dots, h_\kappa$ here playing the role of $y_1, \dots, y_\ell$ in Lemma 5.6. Thus,

$$\Pr\big[\mathbf{h} \neq D(\mathbf{x})\big] \leq 2(\kappa(\lambda) + 1) \cdot 2^{-n} \,.$$

Therefore, we can bound the figure of merit ε_{ex} as

$$\begin{aligned}
&\varepsilon_{\text{ex}}(\lambda, q, n) = \Pr\big[v = \mathtt{accept} \,\wedge\, (\mathit{inst}, w) \notin R\big] = \Pr\big[(\mathbf{x}, \mathbf{h}) \in \tilde{R} \,\wedge\, (\mathit{inst}, w) \notin R\big] \\
&\leq \Pr\big[\big(\mathbf{x}, D(\mathbf{x})\big) \in \tilde{R} \,\wedge\, (\mathit{inst}, w) \notin R\big] + 2(\kappa(\lambda) + 1) \cdot 2^{-n} \\
&\leq \Pr\big[\big(\mathbf{x}, D(\mathbf{x})\big) \in \tilde{R} \wedge (\mathit{inst}, w) \notin R | D \notin \mathsf{SUC} \cup \mathsf{CL}\big] + \Pr[D \in \mathsf{SUC} \cup \mathsf{CL}] + 2(\kappa(\lambda) + 1) \cdot 2^{-n}
\end{aligned}$$

Using the definition of $\tilde{R}$, understanding that $c := \gamma \circ D(\mathit{inst}, \mathbf{y})$, we can write the first term as

$$\begin{aligned}
&\Pr\big[D(\mathbf{m}_\circ) = \mathbf{y}_c \,\wedge\, V(\lambda, \mathit{inst}, c, \mathbf{m}_\circ) \,\wedge\, (\mathit{inst}, w) \notin R \,|\, D \notin \mathsf{SUC} \cup \mathsf{CL}\big] \\
&\qquad \leq \Pr\big[V(\lambda, \mathit{inst}, c, \mathbf{m}_c) \text{ for } \mathbf{m} := D^{-1}(\mathbf{y}) \,\wedge\, (\mathit{inst}, w) \notin R \,|\, D \notin \mathsf{SUC} \cup \mathsf{CL}\big] \\
&\qquad \leq \Pr\big[D \in \mathsf{SUC} \,|\, D \notin \mathsf{SUC} \cup \mathsf{CL}\big] \\
&\qquad = 0 \,,
\end{aligned}$$

where the first equality exploits that $D(m) = y$ iff $m = D^{-1}(y)$ for $D \notin \mathsf{CL}$.

We may thus conclude that

$$\begin{aligned}
\varepsilon_{\text{ex}}(\lambda, q, n) &\leq (2\kappa(\lambda) + 1) \cdot 2^{-n} + \Pr\big[D \in \mathsf{SUC} \cup \mathsf{CL}\big] \\
&\leq (2\kappa(\lambda) + 1) \cdot 2^{-n} + \big[\!\!\big[\bot \overset{q}{\Longrightarrow} \mathsf{SUC} \cup \mathsf{CL}\big]\!\!\big]^2 \,,
\end{aligned}$$

where the last inequality is by definition (43) of $\big[\!\!\big[\bot \overset{q}{\Longrightarrow} \cdot \,\big]\!\!\big]$. The claimed bound now follows from Lemma 5.16. □

5.4.3 The Unruh-Transformation with a Compressing Hash Function

We conclude this section by showing an improvement to the *Unruh transformation* [Unr15b], which follows directly from our result above. At the core of the Unruh transformation is a generic technique to transform any Σ-protocol into a C&O protocol. In [Unr15b], this transformation is presented in combination with parallel repetition and the Fiat-Shamir transformation as a means to construct (online-extractable) NIZK proofs of knowledge in the QROM. The entire transformation was later dubbed the Unruh transformation.

In fact, the Unruh transformation was the first NIZK proof of knowledge in the QROM; the QROM security of the Fiat-Shamir transformation was only established several years later [DFMS19; LZ19a]. Despite being significantly less efficient than the Fiat-Shamir transformation, the Unruh transformation is still useful in certain cases because it puts weaker requirements on the underlying Σ-protocol.

Here, to allow for a modular analysis, we consider the first step of the Unruh transformation, i.e., the transformation from a Σ-protocol into a C&O protocol, as an individual transformation, which we refer to as the *pre-Unruh transformation*, formally defined below. We stress that we allow the random oracle H to be *compressing*, i.e. $|\mathcal{Y}| < |\mathcal{X}|$, while the extraction technique of [Unr15b] required H to be a *length-preserving* RO. This obviously has a significant positive impact on the efficiency of the Unruh transformation.

Let $\varSigma = (\mathcal{P}_\circ, \mathcal{V}_\circ)$ be a Σ-protocol. We write $a_\circ \leftarrow \mathcal{P}_\circ$ to denote the first message in $\varPi_\circ$ as produced by $\mathcal{P}_\circ$ (for a given instance inst). Furthermore, we write $z(a_\circ, c)$ for $\mathcal{P}_\circ$'s response then upon receiving challenge $c \in \mathcal{C}$.[10]

Definition 5.18 (Pre-Unruh transformation). *Let $\varSigma = (\mathcal{P}_\circ, \mathcal{V}_\circ)$ be a Σ-protocol as above. Then, the pre-Unruh-transformation $\mathsf{pU}[\varSigma] = (\mathcal{P}, \mathcal{V})$ of $\varPi_\circ$ is the C&O protocol with first message*

$$a := (a_\circ, (y_i)_{i \in \mathcal{C}})$$

where $a_\circ \leftarrow \mathcal{P}_\circ$ and for each $i \in \mathcal{C}$, $y_i := H(z_i)$ for $z_i := z(a_\circ, i)$), and with response $z := z_c$ upon challenge $c \in \mathcal{C}$. To verify, $\mathcal{V}$ runs $\mathcal{V}_\circ$ on $(a_\circ, c, z)$ and checks if $H(z) = y_c$; if both are true, it accepts, otherwise it rejects.

Clearly, $\mathsf{pU}[\varSigma]$ is only efficient if $\varSigma$ has at most polynomially many possible challenges (which can always be obtained by restricting the challenge space). As mentioned, the resulting C&O protocol can then be repeated in parallel and made non-interactive using the Fiat-Shamir transformation. We will now provide a fairly straightforward corollary to conclude the security of the more

[10] We note that $z(a_\circ, c)$ may be a *randomized* function of $a_\circ$ and c. Furthermore, $z(a_\circ, c)$ is typically computed by $\mathcal{P}_\circ$ by means of the *randomness used to produce* $a_\circ$.

efficient variant of the (full) Unruh transformation that allows for a compressing RO, given by the composition of the pre-Unruh transformation introduced above, parallel repetition and the Fiat-Shamir transformation. In the following, denote the r-fold parallel repetition of a (C&O) Σ-protocol Π by Π^r and use the notation $\mathsf{Unr}_r[\Sigma] := \mathsf{FS}\,[\mathsf{pU}[\Sigma]^r]$ for the Unruh transformation with r-fold parallel repetition.

Remark 5.19. A proof in $\Pi = \mathsf{Unr}_r[\Sigma]$ can be generated in time $T^{\Pi}_{\mathcal{P}} = rT^{\Sigma}_{\mathcal{P}} + (\ell_0 r + 1)T_H$, and verified in time $T^{\Pi}_{\mathcal{V}} = rT^{\Sigma}_{\mathcal{V}} + (1+r)T_H$, where $T^{\Sigma}_{\mathcal{P}}, T^{\Sigma}_{\mathcal{V}}$ and T_H are the prover and verifier runtime of Σ, and the time required for computing one hash, respectively.

It is straightforward to verify that the pre-Unruh transformation does not harm most security properties of the Σ-protocol. In particular, it tightly preserves soundness and honest-verifier zero-knowledge (in the QROM). It also preserves $\mathfrak{S}$-soundness in a certain sense.

Proposition 5.20. *Let Σ be an $\mathfrak{S}$-sound Σ-protocol with challenge space size $\ell = \ell(\lambda)$ with extractor runtime T. Then $\Pi := \mathsf{pU}[\Sigma]$ is $\mathfrak{S}$-sound as a C&O protocol with extractor runtime $T' \leq T + O(\ell)$. Furthermore, suppose that membership in $\mathfrak{S}$ is checkable in time $T_{\mathfrak{S}}$. Then Π is $\mathfrak{S}$-sound* with extractor runtime $T'' \leq T' + \ell^2 T_{\mathfrak{S}} + \ell T_{\mathcal{V}}$, where $T_{\mathcal{V}}$ is the runtime of Π's verification predicate $\mathcal{V}$.*

Proof. Let $\mathcal{E}_{\Sigma}$ be the extractor for Σ guaranteed to exist by Definition 4.22. Note that for $\Pi = \mathsf{pU}[\Sigma]$ regarded as a C&O protocol, for each challenge exactly one of the commitments has to be opened. For such protocols, we use c and $\{c\}$ interchangeably (where c is a challenge in Π). We define an extractor $\mathcal{E}_{\Pi}$ as follows. On input $(\mathit{inst}, m_1, ..., m_\ell, a_\circ, S)$, run $w = \mathcal{E}_{\Sigma}(\mathit{inst}, a_\circ, S, \{m_c\}_{c \in S})$, then output w. The only runtime overhead of $\mathcal{E}_{\Pi}$ results from having to parse its input and preparing the input for $\mathcal{E}_{\Sigma}$.

We continue to define an $\mathfrak{S}$-soundness* extractor $\mathcal{E}^*_{\Pi}$ for Π as follows. On input $(\mathit{inst}, m_1, ..., m_\ell, a_\circ)$, compute $b_c = \mathcal{V}(\mathit{inst}, a_\circ, c, m_c)$ for all $c \in \mathcal{C}$, and set $\hat{S} = \{c \in \mathcal{C} \,|\, b_c = 1\}$. Using at most $\ell(\ell+1)/2$ membership tests for $\mathfrak{S}$, find $S \subseteq \hat{S}$ such that $S \in \mathfrak{S}_{\min}$. Finally, run $w = \mathcal{E}_{\Pi}(\mathit{inst}, m_1, ..., m_\ell, a_\circ, S)$ and output w. The runtime statement is straightforward. □

Using Proposition 5.20 above and Lemma 5.3 from [DFMS22a] to argue $\mathfrak{S}^{\vee r}$-soundness* of the parallel repetition of $\mathsf{pU}[\Pi]$, and using Theorem 5.17 to argue online extractability of its Fiat-Shamir transformation, we obtain the online-extractability of the Unruh transformation with computationally binding commitments, i.e., when using a *compressing* hash function for the commitments.

Corollary 5.21. *Let Σ be an $\mathfrak{S}$-sound Σ-protocol with challenge space size ℓ_0. Then $\Pi := \mathsf{Unr}_r[\Sigma] = \mathsf{FS}[\mathsf{pU}[\Sigma]^r]$ is a proof of knowledge with online extractability in the QROM (as in Definition 5.9) with $\varepsilon_{\mathrm{sim}} = 0$ and*

$$\varepsilon_{\mathrm{ex}}(\lambda, q, n) \leq (22r\ell_0 + 60)q^3 2^{-n} + 20q^2 \left(p^{\mathfrak{S}}_{triv}\right)^r . \tag{55}$$

The online extractor for Π runs in time $T^{\Pi}_{\mathcal{E}} \leq rT^{\mathsf{pU}[\Sigma]}_{\mathcal{E}} + O(q^2) \cdot poly(n, B)$, where $T^{\mathsf{pU}[\Sigma]}_{\mathcal{E}}$ is the runtime of $\mathsf{pU}[\Sigma]$'s $\mathfrak{S}$-soundness extractor as given in Proposition 5.20.*

Section 5.5

Online Extractability of the FS-Transformation: The Case of Merkle-tree-based C&O Protocols

For an ordinary C&O protocol with reasonable concrete security (e.g., 128 bits), the number of commitments ℓ might be considerable. In this case, the communication complexity of the protocol (and thus the size of the non-interactive proof system, or digital-signature scheme, obtained via the Fiat-Shamir transformation) can be reduced by using a *Merkle tree* to collectively commit to the ℓ strings m_i. Such a construction is mentioned in [Fis05], and it is used in the construction of the digital-signature schemes Picnic2 and Picnic3 [KKW18; CDG+20; KZ20; CDG+19]. The Merkle-tree-based C&O mechanism shrinks the commitment information from $\ell \cdot n$ to n, at the expense of increasing the cost of opening $|c|$ values m_i by an additive term of about $\lessapprox |c| \cdot n \cdot \log \ell$.

The cost of opening can, in fact, be slightly reduced again, by streamlining the opening information. When opening several leaves of a Merkle tree, the authentication paths overlap, so opening requires a number of hash values less than h per leaf, where h is the height of the tree. This overlap was observed and exploited in the octopus authentication algorithm which constitutes one of the optimizations of the stateless hash-based signature scheme gravity-SPHINCS [AE18], as well as in Picnic2 and Picnic3 [CDG+20; KZ20]. In the following section, we formalize tree-based collective commitment schemes with "octopus" opening.

5.5.1 Merkle-tree-based C&O Protocols

As was noted in Section 2.2.1, we can consider C&O protocols with a different choice of commitment scheme, compared to the default choice of committing by element-wise hashing. Here, we discuss a particular choice of an alternative commitment scheme, which gives rise to more efficient C&O protocols in certain cases when ℓ is large. Informally, we consider C&O protocols where $m_1, \dots, m_\ell$ is committed to by using a *Merkle tree*, and individual m_i's are opened by announcing the corresponding authentication paths.

To make this more formal, we introduce the following notation. For simplicity, we assume that ℓ is a power of 2, and thus $\ell = 2^h$ for $h \in \mathbb{N}$. We then consider the *full binary tree* $\mathsf{Tree} = \{0,1\}^{\leq h}$ of depth h, where the vertices are identified by bit strings. The root is denoted by $\emptyset$; the i-th leave is denoted by $\mathrm{lf}(i) \in \{0,1\}^h$ and is given by the binary representation of $i \in [\ell]$. The *authentication path* for the i-th leaf is the subtree that consists of all the ancestors of $\mathrm{lf}(i)$ and their siblings:

$$\mathsf{Auth}(i) := \mathsf{Anc}(\mathrm{lf}(i)) \cup \{\mathsf{sib}(v) \,|\, \emptyset \neq v \in \mathsf{Anc}(\mathrm{lf}(i))\}\,,$$

where $\mathsf{Anc}(v) := \{u \in \mathsf{Tree} \,|\, \exists\, w : u \| w = v\}$ and $\mathsf{sib}(u\|b) := u\|(1-b)$ for any $b \in \{0,1\}$. Finally, for any subset $c \subseteq \{1, \dots, \ell\}$, we let $\mathsf{Auth}(c) := \bigcup_{i \in c} \mathsf{Auth}(i)$ be the union of the authentication paths of the considered leaves, and we define the *octopus* $\mathsf{Octo}(c)$ to be the restriction of $\mathsf{Auth}(c)$ to its leaves, but excluding the leaves $\mathrm{lf}(i)$ for $i \in c$, i.e.,

$$\mathsf{Octo}(c) := \mathsf{leaves}(\mathsf{Auth}(c)) \setminus \{\mathrm{lf}(i) \,|\, i \in c\}$$

where, for any subtree T of Tree, $\mathsf{leaves}(T) := \{v \in T \,|\, (v\|0), (v\|1) \notin T\}$.

Extending on the above notation, for a given hash function $H : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X} = \{0,1\}^{\leq B}$ and $\mathcal{Y} = \{0,1\}^n$ for sufficiently large B, we define the *Merkle tree* of $\mathbf{m} = (m_1, ..., m_\ell) \in \mathcal{X}^\ell$ to be the *labeled* binary tree that has its leaves $\mathrm{lf}(1), \dots, \mathrm{lf}(\ell)$ labeled by $H(m_1), ..., H(m_\ell)$, respectively, and each internal vertex is labeled by the hash of the labels of its two children. Formally,

$$\mathsf{MTree}_H(\mathbf{m}) := \big\{ \big(v, l_v(\mathbf{m})\big) \,\big|\, v \in \mathsf{Tree} \big\}$$

with the labeling $l_v(\mathbf{m})$ recursively defined as

$$l_v(\mathbf{m}) := H\big(l_{v\|0}(\mathbf{m}) \| l_{v\|1}(\mathbf{m})\big) \ \text{ for } \ v \in \{0,1\}^{<h}$$

and

$$l_{\mathrm{lf}(i)}(\mathbf{m}) := H(m_i) \text{ for } i \in \{1, \ldots, \ell\},$$

where we leave the dependency of the labeling on H, i.e., $l_v = l_v^H$, implicit. We also write $\mathsf{MRoot}_H(\mathbf{m})$ then for the root label $l_\emptyset(\mathbf{m})$. In the same spirit, we write $\mathsf{MAuth}_H(c, \mathbf{m}) := \big\{\big(v, l_v(\mathbf{m})\big) \,\big|\, v \in \mathsf{Auth}(c)\big\}$ for the labeled authentication path and $\mathsf{MOcto}_H(c, \mathbf{m}) := \big\{\big(v, l_v(\mathbf{m})\big) \,\big|\, v \in \mathsf{Octo}(c)\big\}$ for the labeled octopus, using the same labeling function as for the Merkle tree.

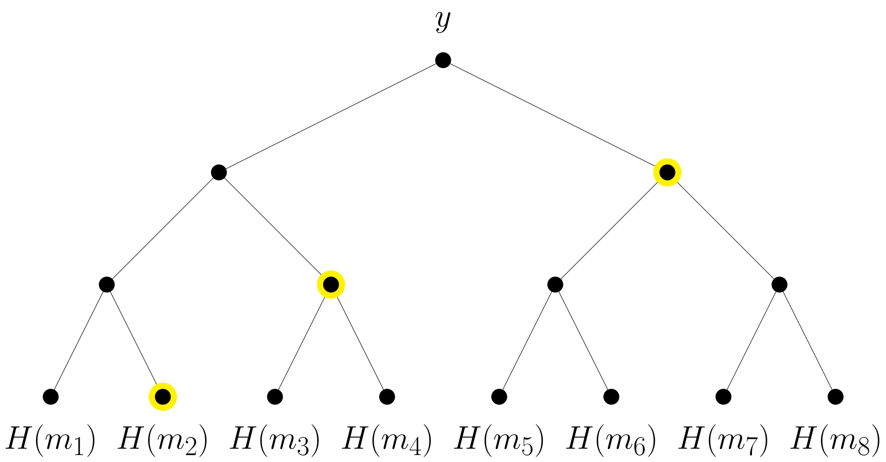

Fig. 5.1. The Merkle tree $\mathsf{MTree}_H(\mathbf{m})$ for $\mathbf{m} = (m_1, \ldots, m_8)$ with $\mathsf{MRoot}_H(\mathbf{m}) = y$. The yellow vertices mark the octopus $\mathsf{MOcto}_H(\{1\}, \mathbf{m})$, which is revealed (along with m_1) when opening the commitment y to m_1.

A *Merkle-tree-based C&O* protocol is now defined to be a variation of a C&O protocol, where the first message of the protocol, i.e., the commitment of $\mathbf{m} = (m_1, \ldots, m_\ell)$, is computed as $y = \mathsf{MRoot}_H(\mathbf{m})$, and the response z for challenge-set c then consists of the messages $\mathbf{m}_c = (m_i)_{i \in c}$ together with $O = \mathsf{MOcto}_H(c, \mathbf{m})$. The verifier $\mathcal{V}$ then accepts if and only if $\mathbf{m}_c$ and O "hash down to" y and the predicate $V(\lambda, \mathit{inst}, c, \mathbf{m}_c, a)$ is satisfied. More formally, the former means that $\mathcal{V}$ computes $\mathsf{MAuth}_H(c, \mathbf{m})$ from $O \cup \{(\mathrm{lf}(i), H(m_i)) \,|\, i \in c\}$ in the obvious way, and then checks whether $l_\emptyset(\mathbf{m}) = y$. This verification is denoted by $\mathit{OctoVerify}^H(c, y, \mathbf{m}_c, O)$, see Figure 5.2.

Looking ahead, we may also consider a variation where the verifier resamples the challenge c if the resulting octopus is bigger than a given bound. Formally, this means that the challenge space of the Merkle-tree-based C&O protocol is restricted to those challenges $c \in [\ell]$ for which $\mathsf{Octo}(c)$ is not too large.

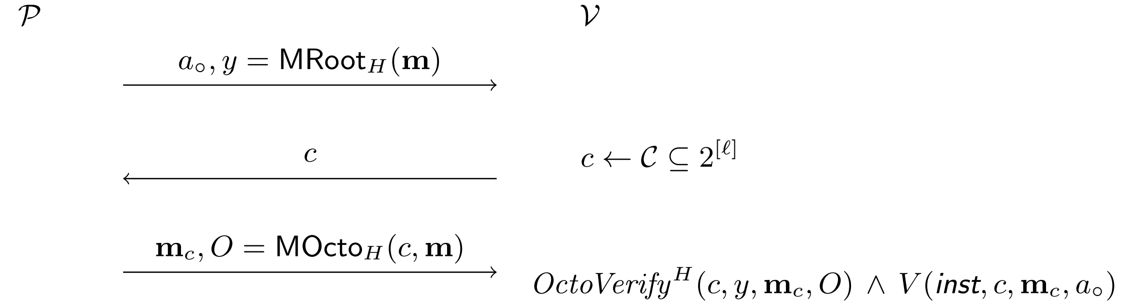
$\mathcal{P}$
$\mathcal{V}$
$a_\circ, y = \mathsf{MRoot}_H(\mathbf{m})$
c
$c \leftarrow \mathcal{C} \subseteq 2^{[\ell]}$
$\mathbf{m}_c, O = \mathsf{MOcto}_H(c, \mathbf{m})$
$\mathit{OctoVerify}^H(c, y, \mathbf{m}_c, O) \wedge V(\mathit{inst}, c, \mathbf{m}_c, a_\circ)$

Fig. 5.2. A Merkle-tree based C&O Σ-protocol, formally introduced in Section 5.5.1.

5.5.2 Online Extractability of the Fiat-Shamir Transformation

The analysis in Section 5.4 can be generalized to the case of FS-transformed Merkle-tree-based C&O protocols. To that end, we generalize the notation from that section as follows. Let $\mathit{\Pi}$ be a Merkle-tree-based C&O protocol with number of messages to be committed equal to $\ell = 2^h$ where h is the height of the commitment Merkle tree.[11]

For a given database $D \in \mathfrak{D}$, recall from Section 5.4 the definition of D^{-1}; applied to a tuple $\mathbf{y} = (y_1, \dots, y_\ell) \in \mathcal{Y}^\ell$ of commitments, D^{-1} attempts to recover the corresponding committed messages $m_1, \dots, m_\ell$. Here, in a similar spirit but now considering the Merkle-tree commitment, MRoot_D^{-1} attempts to recover the committed messages from the root label of the Merkle tree.

In more detail, for a commitment $y \in \mathcal{Y} = \{0,1\}^n$ we reverse engineer the Merkle tree in the obvious way (see Figure 5.3 for an example); namely, accepting a small clash in notation with the labeling function $l_v(\mathbf{m})$ defined for a tuple $\mathbf{m} \in \mathcal{M}^\ell$, we set the root label $l_\emptyset(y) := y$, and recursively define

$$\big(l_{v\|0}(y), l_{v\|1}(y)\big) := \mathsf{split} \circ D^{-1}\big(l_v(y)\big) \in \mathcal{Y} \times \mathcal{Y}$$

for $\emptyset \neq v \in \{0,1\}^{\leq h}$, where split maps any $2n$-bit string, parsed as $y_1\|y_2$ with $y_1, y_2 \in \{0,1\}^n$, to the pair (y_1, y_2) of n-bit strings, while it maps anything else to $(\bot, \bot)$. Then, accepting a small clash in notation again, we set

$$\mathsf{MTree}_D(y) := \{l_v(y) \,|\, v \in \{0,1\}^{\leq h}\}\,,$$

and finally

$$\mathsf{MRoot}_D^{-1}(y) := \big(D^{-1}\big(l_{\mathrm{lf}(1)}(y)\big), \dots, D^{-1}\big(l_{\mathrm{lf}(\ell)}(y)\big)\big)\,.$$

[11] As in the previous section we assume that ℓ is a power of 2 for ease of exposition.

Following the strategy we used in Section 5.4, we define the database property

$$\mathsf{SUC} := \left\{ D \;\middle|\; \begin{array}{c} \exists\, y \in \mathcal{Y} \text{ and } \textit{inst} \in \mathcal{I} \text{ so that } \mathbf{m} := \mathsf{MRoot}_D^{-1}(y) \text{ satisfies} \\ V(\textit{inst}, c, \mathbf{m}_c) \text{ for } c := \gamma \circ D(\textit{inst}, y) \text{ and } \big(inst, \mathcal{E}^*(\textit{inst}, \mathbf{m})\big) \notin R \end{array} \right\},$$

and our first goal is to show that $[\![\bot \overset{q}{\Longrightarrow} \mathsf{SUC} \cup \mathsf{CL}]\!]$ is small.

Lemma 5.22. *Let Π be an $\mathfrak{S}$-sound C&O protocol with $p^{\mathfrak{S}}_{triv}$ as defined in* (47)*. Then*

$$[\![\bot \overset{q}{\Longrightarrow} \mathsf{SUC} \cup \mathsf{CL}]\!] \leq 2eq^{3/2}2^{-n/2} + q\sqrt{10 \max\left(q\ell \cdot 2^{-n+1}, p^{\mathfrak{S}}_{triv}\right)}\,.$$

The proof works exactly as the proof of Lemma 5.16, accounting for some syntactic differences due to the Merkle tree commitment. In particular, where in Case 1 and 2 of the proof of Lemma 5.16 we have to exclude U from falling on one of the hash values $y_1, \ldots, y_\ell$ in order to keep the $\mathbf{m}$ that was constructed from the database intact, we now have a similar restriction for U, but with respect to the whole tree $\mathsf{MTree}_D(y)$.

Proof. As in the proof of of Lemma 5.16, we can bound

$$[\![\bot \overset{q}{\Longrightarrow} \mathsf{SUC} \cup \mathsf{CL}]\!] \leq \sum_{s=0}^{q-1} \left([\![\mathsf{SZ}_{\leq s}\backslash\mathsf{CL} \to \mathsf{CL}]\!] + [\![\mathsf{SZ}_{\leq s}\backslash\mathsf{SUC} \to \mathsf{SUC}]\!]\right) \qquad (56)$$

and use that

$$[\![\mathsf{SZ}_{\leq s}\backslash\mathsf{CL} \to \mathsf{CL}]\!] \leq 2e\sqrt{(s+1)/2^n} \leq 2e\sqrt{q/2^n}\,. \qquad (57)$$

Thus, it remains to control the second term, which we will do again by means of Theorem 5.4 with $\mathsf{P} := \mathsf{SZ}_{\leq s} \setminus \mathsf{SUC}$ and $\mathsf{P}' := \mathsf{SUC}$.

To this end, we consider arbitrary but fixed $D \in \mathfrak{D}$ and input $x \in \mathcal{X}$. By Remark 5.5, we may assume that $D(x) = \bot$. Furthermore, for $\mathsf{P}|_{D|^x}$ to be non-empty, it must be that $D \in \mathsf{SZ}_{\leq s}$, i.e., D is bounded in size. We now distinguish between the following cases for the considered D and x.

Case 1: $D \in \mathsf{SUC}$. In particular, $\bot \in \mathsf{SUC}|_{D|^x} = \mathsf{P}'_{D|^x}$. So, Theorem 5.4 instructs us to set $\;:= \mathsf{P}_{D|^x}$, where we leave the dependency of on D and x implicit. Given that $D \in \mathsf{SUC}$, we can consider *inst* and y as promised by the definition of SUC above, i.e., such that $V(\textit{inst}, c, \mathbf{m}_c)$ and $(\textit{inst}, \mathcal{E}^*(\textit{inst}, \mathbf{m})) \notin R$ for

$$c := \gamma \circ D(\textit{inst}, y) \quad \text{and} \quad \mathbf{m} := \mathsf{MRoot}_D^{-1}(y)\,. \qquad (58)$$

Note that, since $D(x) = \bot$ and $V(\mathit{inst}, c, \mathbf{m}_c)$ holds, which in particular means that c must be defined, it must be that $x \neq (\mathit{inst}, y)$. Therefore

$$\gamma \circ D(\mathit{inst}, y) = \gamma \circ D[x \mapsto u](\mathit{inst}, y)\,. \tag{59}$$

Our goal now is to show the final implication in

$$u \in \mathsf{L} \iff D[x \mapsto u] \in \mathsf{P} \implies D[x \mapsto u] \notin \mathsf{SUC} \implies u \in \mathsf{MTree}_D(y)\,.$$

We will do this by showing that $u \notin \mathsf{MTree}_D(y)$ implies

$$\mathsf{MRoot}_D^{-1}(y) = \mathsf{MRoot}_{D[x\mapsto u]}^{-1}(y)\,. \tag{60}$$

Indeed, the contraposition $u \notin \mathsf{MTree}_D(y) \Rightarrow D[x \mapsto u] \in \mathsf{SUC}$ of the claimed implication then follows from the fact that (59) and (60) together imply that c and $\mathbf{m}$ remain unchanged when replacing D by $D[x \mapsto u]$ in (58), and so $D[x \mapsto u] \in \mathsf{SUC}$ as well.

Towards showing (60), exploiting again that $D(x) = \bot$, it follows by definition of the reverse engineered labeling function $l_v(y)$ that $x \neq (l_{v\|0}(y), l_{v\|1}(y))$ for any v with $l_{v\|0}(y) \neq \bot \neq l_{v\|1}(y)$, i.e., x is not equal to any pair of siblings in $\mathsf{MTree}_D(y)$ with non-$\bot$ labeling (see Figure 5.3). Due to a similar reasoning, $x \neq m_i$ for any i. It now follows by definition of the reverse engineered Merkle tree and of MRoot^{-1} that if $u \notin \mathsf{MTree}_D(y)$ then $\mathsf{MTree}_D(y) = \mathsf{MTree}_{D[x\mapsto u]}(y)$ and $\mathsf{MRoot}_D^{-1}(y) = \mathsf{MRoot}_{D[x\mapsto u]}^{-1}(y)$, as claimed.

Thus, we can bound

$$P[U \in] \leq P[U \in \mathsf{MTree}_D(y)] \leq \frac{2 \cdot 2^h - 1}{|\mathcal{Y}|} = \frac{2\ell - 1}{|\mathcal{Y}|}\,. \tag{61}$$

Case 2: $D \notin \mathsf{SUC}$, and x is a “commit query”, i.e., $x = m \in \mathcal{M}$ or $x = (l_{v\|0}, l_{v\|1})$ for two labels $l_{v\|0}, l_{v\|1} \in \mathcal{Y}$. In particular, $\bot \notin \mathsf{P}'|_{D|^x}$ (given that $D(x) = \bot$) and so in the light of Theorem 5.4 we may choose $\mathsf{L} := \mathsf{P}'|_{D|^x}$. We then have

$$u \in \mathsf{L} \iff D[x \mapsto u] \in \mathsf{P}' = \mathsf{SUC} \implies \exists\, \mathit{inst}, y : D(\mathit{inst}, y) \neq \bot \wedge u \in \mathsf{MTree}_D(y)$$

where final implication can be seen as follows. By definition of SUC, the assumption $D[x \mapsto u] \in \mathsf{SUC}$ implies the existence of inst and y with $V(\mathit{inst}, c, \mathbf{m}_c)$ and $\big(\mathit{inst}, \mathcal{E}^*(\mathit{inst}, \mathbf{m})\big) \notin R$ for

$$c := \gamma \circ D[x \mapsto u](\mathit{inst}, y) = \gamma \circ D(\mathit{inst}, y) \quad \text{and} \quad \mathbf{m} := \mathsf{MRoot}_{D[x\mapsto u]}^{-1}(y)\,,$$

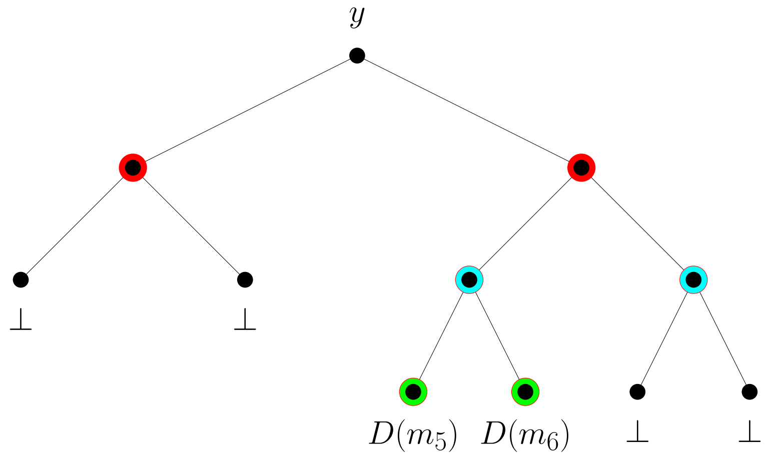

Fig. 5.3. Example of a reverse engineered Merkle tree $\mathsf{MTree}_D(y)$, with the $\bot$-children of the $\bot$-labels omitted. Since $D(x) = \bot$, $x \neq (l_u(y), l_w(y))$ for any two siblings (u, w) in $\mathsf{MTree}_D(y)$, i.e., nodes with the same color. Assuming that $u \notin \mathsf{MTree}_D(y)$ then implies that reprogramming D to $D[x \mapsto u]$ does not affect the reverse engineered Merkle tree.

where the equality in the definition of c exploits that x is not a “challenge” query. The fact that $V(\mathit{inst}, c, \mathbf{m_c})$ is satisfied for this c thus implies that $D(\mathit{inst}, y) \neq \bot$. Next, with the goal to reach a contradiction, assume that $u \notin \mathsf{MTree}_D(y)$. Then for all $\bot \neq h \in \mathsf{MTree}_D(y)$ we have that $D^{-1}(h) = D[x \mapsto u]^{-1}(h)$ except if $D(x) = h$, but this cannot be since $D(x) = \bot$. It follows that $\mathsf{MTree}_D(y) = \mathsf{MTree}_D[x \mapsto u](y)$ and $\mathsf{MRoot}_D^{-1}(y) = \mathsf{MRoot}_{D[x\mapsto u]}^{-1}(y)$. The above then implies that $D \in \mathsf{SUC}$, a contradiction.

Thus, we can bound

$$P[U \in] \le P[\exists\, \mathit{inst}, y : D(\mathit{inst}, y) \neq \bot \wedge U \in \mathsf{MTree}_D(y)] \le \frac{s(2\ell-1)}{|\mathcal{Y}|} \le \frac{q(2\ell-1)}{|\mathcal{Y}|}\,. \tag{62}$$

Case 3: $D \notin \mathsf{SUC}$, and x is a “challenge query”, i.e., $x = (\mathit{inst}, y) \in \mathcal{I} \times \mathcal{Y}$. Set $\mathbf{m} := \mathsf{MRoot}_D^{-1}(y)$. Again, we have that $\bot \notin \mathsf{SUC}|_{D|^x} = \mathsf{P}'_{D|^x}$, and so by Theorem 5.4 we may set $:= \mathsf{P}'_{D|^x}$. Here, we can argue that

$$\begin{aligned} u \in \mathsf{L} \iff & D[x \mapsto u] \in \mathsf{P}' = \mathsf{SUC} \\ \implies & V(\mathit{inst}, \gamma(u), \mathbf{m}_{\gamma(u)}) \text{ and } \big(\mathit{inst}, \mathcal{E}^*(\mathit{inst}, \mathbf{m})\big) \notin R\,, \end{aligned}$$

where the final implication can be seen as follows. By definition of SUC, the assumption $D[x \mapsto u] \in \mathsf{SUC}$ implies the existence of inst' and y' with $V(\mathit{inst}', c, \mathbf{m}'_c)$ and $\big(\mathit{inst}', \mathcal{E}^*(\mathit{inst}', \mathbf{m}')\big) \notin R$ for

$$c := \gamma \circ D[x \mapsto u](\mathit{inst}', y') \quad \text{and} \quad \mathbf{m}' := \mathsf{MRoot}_{D[x\mapsto u]}^{-1}(y') = \mathsf{MRoot}_D^{-1}(y')\,,$$

where the very last equality exploits that x is not a "commit" query. With the goal to come to a contradiction, assume that $(\mathit{inst}', y') \neq (\mathit{inst}, y) = x$. Then, $c = \gamma \circ D[x \mapsto u](\mathit{inst}', y') = \gamma \circ D(\mathit{inst}', y')$, and the above then implies that $D \in \mathsf{SUC}$, a contradiction. Thus, $(\mathit{inst}', y') = (\mathit{inst}, y) = x$. In particular, $\mathbf{m}' = \mathbf{m}$ and $c = \gamma \circ D[x \mapsto u](\mathit{inst}', y') = \gamma \circ D[x \mapsto u](x) = \gamma(u)$. Hence, the claimed implication holds.

Thus, we can bound

$$\begin{aligned} P[U \in] &\leq P[V(\mathit{inst}, \gamma(U), \mathbf{m}_{\gamma(U)}) \wedge \big(\mathit{inst}, \mathcal{E}^*(\mathit{inst}, \mathbf{m})\big) \notin R] \\ &\leq P[V(\mathit{inst}, \gamma(U), \mathbf{m}_{\gamma(U)}) \wedge S := \{c \,|\, V(\mathit{inst}, c, \mathbf{m}_c)\} \notin \mathfrak{S}] \\ &\leq P[\gamma(U) \in S := \{c \,|\, V(\mathit{inst}, c, \mathbf{m}_c)\} \notin \mathfrak{S}] \\ &\leq \max_{S \notin \mathfrak{S}} P[\gamma(U) \in S] \\ &\leq p^{\mathfrak{S}}_{triv} \,. \end{aligned} \tag{63}$$

By Theorem 5.4, we now get

$$\begin{aligned} [\![\mathsf{SZ}_{\leq s} \backslash \mathsf{SUC} \backslash \mathsf{CL} \to \mathsf{SUC}]\!] &\leq \max_{x,D} \sqrt{10 P\big[U \in \mathsf{L}^{x,D}\big]} \\ &\leq \sqrt{10}\sqrt{\max\left(\frac{2\ell - 1}{|\mathcal{Y}|}, \frac{q(2\ell-1)}{|\mathcal{Y}|}, p^{\mathfrak{S}}_{triv}\right)} \\ &\leq \sqrt{10}\sqrt{\max\big(q\ell \cdot 2^{-n+1}, p^{\mathfrak{S}}_{triv}\big)}, \end{aligned}$$

where we have used Equations (61), (62) and (63) in the second inequality. Combining with Equations (57) and (56) yields the desired bound. □

Similarly to Theorem 5.17, we now obtain the following.

Theorem 5.23. *Let $\mathit{\Pi}$ be an $\mathfrak{S}$-sound* Merkle-tree-based C&O protocol with challenge space $\mathcal{C}_\lambda$. Then $\mathsf{FS}[\Pi]$ is a proof of knowledge with online extractability in the QROM (as in Definition 5.9), with $\varepsilon_{\mathrm{sim}}(\lambda, q, n) = 0$ and*

$$\begin{aligned} \varepsilon_{\mathrm{ex}}(\lambda, q, n) &\leq 2(\kappa \log \ell + 1) \cdot 2^{-n} + \left(2eq^{3/2} 2^{-n/2} + q\sqrt{10 \max\big(q\ell \cdot 2^{-n+1}, p^{\mathfrak{S}}_{triv}\big)}\right)^2 \\ &\leq (22\ell \log \ell + 60)\, q^3 2^{-n} + 20 q^2 p^{\mathfrak{S}}_{triv} \end{aligned}$$

where $\kappa = \kappa(\lambda) := \max_{c \in \mathcal{C}_\lambda} |c|$ and ℓ is the number of leaves of the Merkle-tree-based commitment. The running time of the extractor is dominated by running the compressed oracle, which has complexity $O(q^2) \cdot poly(n, B)$, and by computing $\mathsf{MRoot}_D^{-1}(y)$ and running $\mathcal{E}^*$.

Here again the proof follows exactly the outline of its counterpart from Section 5.4.2, with some minor alterations to cope with the formalism of a Merkle-tree based C&O Σ-protocol. The difference in the bound is simply due to the difference between Lemmas 5.16 and 5.22.

Proof. We consider an arbitrary but fixed $\lambda \in \mathbb{N}$. Let $\mathcal{P}^*$ be a dishonest prover that, after making q queries to a random oracle H, outputs and instance inst and a proof $\pi = (y, \mathbf{m}_\circ, O)$ plus some (possibly quantum) auxiliary output Z, where O is an authentication octopus as defined in Section 5.5.1. For simplicity, we assume that $|c|$ is the same for all $c \in \mathcal{C}_\lambda$, and thus equal to κ. If it is not, we could always make the prover output a couple of dummy outputs m_i to match the upper bound on $|c|$. In the experiment $\mathcal{V}^{\mathcal{E}} \circ \mathcal{P}^{*\mathcal{E}}(\lambda)$, our extractor $\mathcal{E}$ works as follows while simulating all queries to H (by $\mathcal{P}^*$ and $\mathcal{V}$) with the compressed oracle:

1. Run $\mathcal{P}^*(\lambda)$ to obtain (inst, π, Z) with $\pi = (y, \mathbf{m}_\circ, O)$.
2. Compute $v \leftarrow \mathcal{V}^H(\mathsf{inst}, \pi)$, given by the truth value of

$$\mathit{OctoVerify}^H(c, y, \mathbf{m}_\circ, O) \quad \wedge \quad V(\mathsf{inst}, c, \mathbf{m}_\circ) \quad \text{with} \quad c := \gamma(H(\mathsf{inst}, y))\,.$$

3. Measure the internal state of the compressed oracle to obtain D.
4. Run $\mathcal{E}^*$ on input $\mathsf{MRoot}_D^{-1}(y)$ to obtain w.

Note that in the views of both $\mathcal{P}^*$ and $\mathcal{V}$, the interaction with H and the interaction with $\mathcal{E}$ differ only in that their oracle queries are answered by a compressed oracle instead of a real random oracle in the latter case. This simulation is perfect and therefore $\varepsilon_{\text{sim}}(\lambda, q, n) = 0$.

Considering $\mathcal{P}^*$ as the algorithm $\mathcal{A}$ in Corollary 5.7, the composition $\mathcal{V} \circ \mathcal{P}^*$ then matches up with the algorithm $\tilde{\mathcal{A}}$ for $\mathcal{F} = \mathcal{V}$. Thus, noting that $\kappa(\log \ell + 1)$ is an upper bound on the amount of queries that *OctoVerify* makes,

$$\Pr\big[v \neq \mathcal{V}^D(\mathsf{inst}, \pi)\big] \leq 2(\kappa \log \ell + 1) \cdot 2^{-n}\,.$$

Therefore, we can bound bound the figure of merit ε_{ex} as

$$\begin{aligned}
\varepsilon_{\text{ex}}(\lambda, q, n) &= \Pr\big[v = 1 \,\wedge\, (\mathsf{inst}, w) \notin R\big] \\
&\leq \Pr\big[\mathcal{V}^D(\mathsf{inst}, \pi) \,\wedge\, (\mathsf{inst}, w) \notin R\big] + 2(\kappa \log \ell + 1) \cdot 2^{-n} \\
&\leq \Pr[\mathcal{V}^D(\mathsf{inst}, \pi) \,\wedge\, (\mathsf{inst}, w) \notin R \,|\, D \notin \mathsf{SUC} \cup \mathsf{CL}] \\
&\qquad + \Pr[D \in \mathsf{SUC} \cup \mathsf{CL}] + 2(\kappa \log \ell + 1) \cdot 2^{-n}\,.
\end{aligned}$$

Using the definition of $\mathcal{V}^D(\textsf{\textit{inst}}, \pi)$, understanding that $c := \gamma \circ D(\textsf{\textit{inst}}, y)$, we can write the first term as

$$\begin{aligned}
&\Pr\big[\mathit{OctoVerify}^D(c, y, \mathbf{m}_\circ, O) \wedge V(\textsf{\textit{inst}}, c, \mathbf{m}_\circ) \wedge (\textsf{\textit{inst}}, w) \notin R \,|\, D \notin \mathsf{SUC} \cup \mathsf{CL}\big] \\
&\quad \leq \Pr\big[V(\textsf{\textit{inst}}, c, \mathbf{m}_c) \text{ for } \mathbf{m} := \mathsf{MRoot}_D^{-1}(y) \wedge (\textsf{\textit{inst}}, w) \notin R \,|\, D \notin \mathsf{SUC} \cup \mathsf{CL}\big] \\
&\quad \leq \Pr\big[D \in \mathsf{SUC} \,|\, D \notin \mathsf{SUC} \cup \mathsf{CL}\big] \\
&\quad = 0\,,
\end{aligned}$$

where the first equality exploits that $D(m) = h$ iff $m = D^{-1}(h)$ for $D \notin \mathsf{CL}$.

We may thus conclude that

$$\begin{aligned}
\varepsilon_{\mathrm{ex}}(\lambda, q, n) &\leq 2(\kappa \log \ell + 1) \cdot 2^{-n} \cdot 2^{-n} + \Pr\big[D \in \mathsf{SUC} \cup \mathsf{CL}\big] \\
&\leq 2(\kappa \log \ell + 1) \cdot 2^{-n} + \big[\!\big[\bot \overset{q}{\Longrightarrow} \mathsf{SUC} \cup \mathsf{CL}\big]\!\big]^2\,,
\end{aligned}$$

where the last inequality is by definition of $\big[\!\big[\bot \overset{q}{\Longrightarrow} \cdot\,\big]\!\big]$. The claimed bound now follows from Lemma 5.22. □

5.5.3 Discussion: Application to Picnic, and Limiting the Proof Size

Application to Picnic. A prominent use case of C&O protocols is the construction of digital signature schemes via the Fiat-Shamir transformation. An important example is Picnic [CDG+17] currently under consideration as an alternate candidate in the NIST standardization process for post-quantum cryptographic schemes. On a high level, the design of Picnic can be described as follows. A C&O Σ-protocol is constructed using the MPC-in-the-head paradigm [IKOS07a]. Then, the Fiat-Shamir transformation is applied in the usual way to obtain a digital signature scheme. There are three evolutions of Picnic: Picnic-FS, Picnic 2 and Picnic 3.[12] Picnic-FS uses plain hash-based commitments, while Picnic 2 and Picnic 3 use a Merkle-tree-based collective commitment.

All three evolutions enjoy provable post-quantum security when the hash function used for the Fiat-Shamir transformation is modeled as a (quantum-accessible) RO. The best reduction applying to all of them proceeds as follows. First, Unruh's rewinding lemma [Unr12] is used to construct a knowledge extractor for the underlying Σ-protocol based on an appropriate $\mathfrak{S}$-soundness notion. Then, the *generic* QROM reduction for the Fiat-Shamir transformation

[12] The original evolution also came with a variant using the Unruh transformation, Picnic-Ur. We restrict our attention to the variants using the Fiat-Shamir transformation.

from Theorem 3.7 is used to construct a knowledge extractor for the signature scheme in the QROM from the extractor for the Σ-protocol. Finally, the technique from [GHHM21] is used for simulating the chosen-message oracle to reduce breaking NMA (no-message attack) security to breaking CMA (chosen-message attack) security. This final step connects to the previous one because for the signature scheme the witness extracted from an NMA attacker is the secret key.

The first two steps in this chain of reductions, i.e. Unruh's rewinding and Theorem 3.7, are, however, not tight: The former loses at least a fifth power in the Picnic case, and the latter a factor of q^2, where q is the number of random oracle queries. This means that an NMA attacker with success probability ϵ can be used to break the underlying hard problem with probability $\Omega(\epsilon^5/q^{10})$ (or worse, depending on the Picnic variant).

For Picnic-FS (only), when in addition modeling the hash function used for the commitments as a RO, Unruh's rewinding can be replaced with the tight online extraction technique from Chapter 4. The remaining loss due to the Fiat-Shamir reduction is of order ϵ/q^2, up to some additive terms accounting for search and collision finding in the RO, a sizable improvement over the above but still not tight.

By analyzing the Fiat-Shamir transformation of a C&O protocol (with or without Merkle tree commitments) directly, our results provide a tight alternative to the above lossy reductions. Using Theorems 5.17 (for Picnic-FS) and 5.23 (for Picnic 2 and Picnic 3) we can avoid all multiplicative/power losses in the reduction for NMA security. An NMA attacker with success probability ϵ can, in other words, be used to break the underlying hard problem with probability ϵ, up to some unavoidable additive terms accounting for search and collision finding in the RO.

An observation about octopus opening sizes. Depending on the parameters of the C&O protocol, the octopus opening information, $\mathsf{MOcto}(c, \mathbf{m})$ can be significantly smaller than the concatenation of the individual authentication paths. On the other hand, it is also *variable in size* (namely dependent on the choice of the challenge c), and the variance can be significant (see e.g. the computations for gravity SPHINCS in [AE18]). In the context of a digital signature scheme constructed via the Fiat-Shamir transformation of a Merkle-tree-based C&O protocol, like, e.g., Picnic 2 and Picnic 3, this leads to the undesirable property of a variable signature size, where signatures can be quite a bit larger in the worst case than on average. This might, e.g., lead to problems when looking for a drop-in replacement for quantum-broken digital signature schemes for use in a larger protocol, where signatures need to be stored in a data field of fixed size.

One option to mitigate this situation is to cut off the tail of the octopus size distribution, i.e. to restrict the challenge space of the Merkle-tree-based C&O protocol to the set of challenges whose octopus is not larger than some bound. This can be done before applying the Fiat-Shamir transformation, e.g. using rejection sampling. In that way, one obtains a digital signature scheme with significantly reduced worst case signature size, at the expense of a tiny security loss.

5.5.4 The Merkle-Tree-Based Unruh Transformation

The Merkle tree based commitment mechanism can replace plain random-oracle based commitments in *any* ordinary C&O protocol, in particular in $\mathit{\Pi} := \mathsf{pU}[\mathit{\Sigma}]$ for any Σ-protocol $\mathit{\Sigma}$. The result is a Merkle-tree-based C&O protocol and we obtain a corollary analogous to Corollary 5.21.

Corollary 5.24. *Let $\mathit{\Sigma}$ be an $\mathfrak{S}$-sound $\mathit{\Sigma}$-protocol with challenge space size ℓ_0. Then $\mathsf{FS}[\mathsf{MPpU}_r[\mathit{\Sigma}]]$ is online-extractable with*

$$\varepsilon_{\mathrm{ex}} \leq \left(22r\ell_0 \log\left(r\ell_0\right) + 60\right) q^3 2^{-n} + 20q^2 \left(p^{\mathfrak{S}}_{triv}\right)^r \tag{64}$$

*where $\mathsf{MPpU}_r[\mathit{\Sigma}]$ is the **M**erkle-tree-based, **P**arallel-repeated, **p**re-**U**nruh transformation of $\mathit{\Sigma}$, i.e., the Merkle-tree-based C&O protocol obtained by replacing the commitments of $\mathsf{pU}[\mathit{\Pi}]^r$ with a Merkle-tree-based collective commitment.*

Bibliography

[ABCP22] Shahla Atapoor, Karim Baghery, Daniele Cozzo, and Robi Pedersen. *CSI-SharK: CSI-FiSh with Sharing-friendly Keys*. Cryptology ePrint Archive, Paper 2022/1189. https://eprint.iacr.org/2022/1189. 2022. URL: https://eprint.iacr.org/2022/1189.

[ABG+20] Amit Agarwal, James Bartusek, Vipul Goyal, Dakshita Khurana, and Giulio Malavolta. *Post-Quantum Multi-Party Computation*. 2020. arXiv: 2005.12904 [quant-ph].

[AE18] Jean-Philippe Aumasson and Guillaume Endignoux. "Improving Stateless Hash-Based Signatures". In: *Topics in Cryptology – CT-RSA 2018*. Ed. by Nigel P. Smart. Cham: Springer International Publishing, 2018, pp. 219–242. ISBN: 978-3-319-76953-0.

[AFLT12] Michel Abdalla, Pierre-Alain Fouque, Vadim Lyubashevsky, and Mehdi Tibouchi. "Tightly-Secure Signatures from Lossy Identification Schemes". In: *Advances in Cryptology – EUROCRYPT 2012*. Ed. by David Pointcheval and Thomas Johansson. Berlin, Heidelberg: Springer, 2012, pp. 572–590. ISBN: 978-3-642-29011-4.

[AHU19] Andris Ambainis, Mike Hamburg, and Dominique Unruh. "Quantum Security Proofs Using Semi-classical Oracles". In: *Advances in Cryptology – CRYPTO 2019*. Ed. by Alexandra Boldyreva and Daniele Micciancio. Cham: Springer International Publishing, 2019, pp. 269–295. ISBN: 978-3-030-26951-7.

[AMRS20] Gorjan Alagic, Christian Majenz, Alexander Russell, and Fang Song. "Quantum-Access-Secure Message Authentication via Blind-Unforgeability". In: *Advances in Cryptology – EUROCRYPT 2020*. Ed. by Anne Canteaut and Yuval Ishai. Cham: Springer International Publishing, 2020, pp. 788–817. ISBN: 978-3-030-45727-3.

[ARU14] A. Ambainis, A. Rosmanis, and D. Unruh. "Quantum Attacks on Classical Proof Systems: The Hardness of Quantum Rewinding". In: *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*. Oct. 2014, pp. 474–483. DOI: 10.1109/FOCS.2014.57.

[BBB+18] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell. "Bulletproofs: Short Proofs for Confidential Transactions and More". In: *2018 IEEE Symposium on Security and Privacy (SP).* May 2018, pp. 315–334. DOI: `10.1109/SP.2018.00020`.

[BBHT98] Michel Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp. "Tight Bounds on Quantum Searching". In: *Fortschritte der Physik* 46.4-5 (1998), pp. 493–505.

[BDF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. "Random Oracles in a Quantum World". In: *Advances in Cryptology – ASIACRYPT 2011.* Ed. by Dong Hoon Lee and Xiaoyun Wang. Berlin, Heidelberg: Springer, 2011, pp. 41–69. ISBN: 978-3-642-25385-0.

[BDK+21] Shi Bai, Leo Ducas, Eike Kiltz, Lepoint Tancrede, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehle. *CRYSTALS-Dilithium Algorithm Specifications and Supporting Documentation (Version 3.1).* `https://pq-crystals.org/dilithium/data/dilithium-specification-round3-20210208.pdf`, retrieved on 19.03.2023. 2021.

[Beu20] Ward Beullens. "Sigma Protocols for MQ, PKP and SIS, and Fishy Signature Schemes". In: *Advances in Cryptology – EUROCRYPT 2020.* Ed. by Anne Canteaut and Yuval Ishai. Cham: Springer International Publishing, 2020, pp. 183–211. ISBN: 978-3-030-45727-3.

[BG93] Mihir Bellare and Oded Goldreich. "On Defining Proofs of Knowledge". In: *Advances in Cryptology — CRYPTO' 92.* Ed. by Ernest F. Brickell. Berlin, Heidelberg: Springer, 1993, pp. 390–420. ISBN: 978-3-540-48071-6.

[BGKM23] Loïc Bidoux, Philippe Gaborit, Mukul Kulkarni, and Victor Mateu. "Code-based signatures from new proofs of knowledge for the syndrome decoding problem". In: *Designs, Codes and Cryptography* 91.2 (2023), pp. 497–544.

[BHH+19] Nina Bindel, Mike Hamburg, Kathrin Hövelmanns, Andreas Hülsing, and Edoardo Persichetti. "Tighter Proofs of CCA Security in the Quantum Random Oracle Model". In: *Theory of Cryptography.* Ed. by Dennis Hofheinz and Alon Rosen. Cham: Springer International Publishing, 2019, pp. 61–90. ISBN: 978-3-030-36033-7.

[BHT98] Gilles Brassard, Peter Høyer, and Alain Tapp. "Quantum cryptanalysis of hash and claw-free functions". In: *LATIN'98: The-*

oretical Informatics. Ed. by Cláudio L. Lucchesi and Arnaldo V. Moura. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 163–169. ISBN: 978-3-540-69715-2.

[BKV19] Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. "CSI-FiSh: Efficient Isogeny Based Signatures Through Class Group Computations". In: *Advances in Cryptology – ASIACRYPT 2019.* Ed. by Steven D. Galbraith and Shiho Moriai. Cham: Springer International Publishing, 2019, pp. 227–247. ISBN: 978-3-030-34578-5.

[BLZ21] Jeremiah Blocki, Seunghoon Lee, and Samson Zhou. *On the Security of Proofs of Sequential Work in a Post-Quantum World.* 2021. arXiv: 2006.10972 [cs.CR].

[BMPS20] Jean-François Biasse, Giacomo Micheli, Edoardo Persichetti, and Paolo Santini. "LESS is More: Code-Based Signatures Without Syndromes". In: *Progress in Cryptology - AFRICACRYPT 2020.* Ed. by Abderrahmane Nitaj and Amr Youssef. Cham: Springer International Publishing, 2020, pp. 45–65. ISBN: 978-3-030-51938-4.

[BR93] Mihir Bellare and Phillip Rogaway. "Random oracles are practical: A paradigm for designing efficient protocols". In: *Proceedings of the 1st ACM conference on Computer and communications security.* ACM. 1993, pp. 62–73.

[BSK+21] Carsten Baum, Cyprien Delpech de Saint Guilhem, Daniel Kales, Emmanuela Orsini, Peter Scholl, and Greg Zaverucha. "Banquet: Short and Fast Signatures from AES". In: *Public-Key Cryptography – PKC 2021.* Ed. by Juan A. Garay. Cham: Springer International Publishing, 2021, pp. 266–297. ISBN: 978-3-030-75245-3.

[CDG+17] Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, and Greg Zaverucha. "Post-Quantum Zero-Knowledge and Signatures from Symmetric-Key Primitives". In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security.* CCS '17. Dallas, Texas, USA: ACM, 2017, pp. 1825–1842. ISBN: 978-1-4503-4946-8. DOI: 10.1145/3133956.3133997.

[CDG+19] Melissa Chase, David Derler, Steven Goldfeder, Jonathan Katz, Vladimir Kolesnikov, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, Xiao Wang, et al. "The picnic signature scheme". In: *Submission to NIST Post-Quantum Cryptography project* (2019).

[CDG+20] Melissa Chase, David Derler, Steven Goldfeder, Jonathan Katz, Vladimir Kolesnikov, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, Xiao Wang, and Greg Zaverucha. *The Picnic Signature Scheme, Design Document v2.1.* 2020. URL: `https://github.com/microsoft/Picnic/blob/master/spec/design-v2.2.pdf`.

[CDS94] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. "Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols". In: *Advances in Cryptology — CRYPTO '94*. Ed. by Yvo G. Desmedt. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 174–187. ISBN: 978-3-540-48658-9.

[CFHL21] Kai-Min Chung, Serge Fehr, Yu-Hsuan Huang, and Tai-Ning Liao. "On the Compressed-Oracle Technique, and Post-Quantum Security of Proofs of Sequential Work". In: *Advances in Cryptology – EUROCRYPT 2021*. Ed. by Anne Canteaut and François-Xavier Standaert. Cham: Springer International Publishing, 2021, pp. 598–629. ISBN: 978-3-030-77886-6.

[CGH04] Ran Canetti, Oded Goldreich, and Shai Halevi. "The random oracle methodology, revisited". In: *Journal of the ACM* 51.4 (July 2004), pp. 557–594. ISSN: 00045411. DOI: `10.1145/1008731.1008734`. arXiv: `0010019 [cs]`. URL: `http://arxiv.org/abs/cs/0010019`.

[CGLQ20] Kai-Min Chung, Siyao Guo, Qipeng Liu, and Luowen Qian. "Tight Quantum Time-Space Tradeoffs for Function Inversion". In: *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*. 2020, pp. 673–684. DOI: `10.1109/FOCS46700.2020.00068`.

[Cha19] André Chailloux. *Tight quantum security of the Fiat-Shamir transform for commit-and-open identification schemes with applications to post-quantum signature schemes.* Cryptology ePrint Archive, Report 2019/699, version 1 Jul 2019. `https://eprint.iacr.org/2019/699/20190701:091436`. 2019.

[Cha21] André Chailloux. *Tight quantum security of the Fiat-Shamir transform for commit-and-open identification schemes with applications to post-quantum signature schemes.* Cryptology ePrint

Archive, Report 2019/699, version 16 Mar 2021. https://eprint.iacr.org/2019/699/20210316:124850. 2021.

[CHH+21] Kai-Min Chung, Yao-Ching Hsieh, Mi-Ying Huang, Yu-Hsuan Huang, Tanja Lange, and Bo-Yin Yang. *Isogeny-based Group Signatures and Accountable Ring Signatures in QROM*. Cryptology ePrint Archive, Paper 2021/1368. 2021. URL: https://eprint.iacr.org/2021/1368.

[CHR+16] Ming-Shing Chen, Andreas Hülsing, Joost Rijneveld, Simona Samardjiska, and Peter Schwabe. "From 5-Pass MQ-Based Identification to MQ-Based Signatures". In: *Advances in Cryptology – ASIACRYPT 2016*. Ed. by Jung Hee Cheon and Tsuyoshi Takagi. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 135–165. ISBN: 978-3-662-53890-6.

[CHR+18] Ming-Shing Chen, Andreas Hülsing, Joost Rijneveld, Simona Samardjiska, and Peter Schwabe. "SOFIA: MQ-Based Signatures in the QROM". In: *Public-Key Cryptography – PKC 2018*. Ed. by Michel Abdalla and Ricardo Dahab. Cham: Springer International Publishing, 2018, pp. 3–33. ISBN: 978-3-319-76581-5.

[CHR+20] Ming-Shing Chen, Andreas Hülsing, Joost Rijneveld, Simona Samardjiska, and Peter Schwabe. *MQDSS Specfications version 2.1*. https://repository.ubn.ru.nl/bitstream/handle/2066/236576/236576.pdf, retrieved on 16.03.2023. 2020.

[CMS19] Alessandro Chiesa, Peter Manohar, and Nicholas Spooner. "Succinct Arguments in the Quantum Random Oracle Model". In: *Theory of Cryptography*. Ed. by Dennis Hofheinz and Alon Rosen. Cham: Springer International Publishing, 2019, pp. 1–29. ISBN: 978-3-030-36033-7.

[CMSZ19] Jan Czajkowski, Christian Majenz, Christian Schaffner, and Sebastian Zur. *Quantum Lazy Sampling and Game-Playing Proofs for Quantum Indifferentiability*. Cryptology ePrint Archive, Report 2019/428. https://eprint.iacr.org/2019/428. 2019.

[Dam10] Ivan Damgard. *On Sigma-Protocols, Lecture notes, Faculty of Science Aarhus University, Department of Computer Science*. 2010. URL: http://www.cs.au.dk/~ivan/Sigma.pdf.

[DFG13] Özgür Dagdelen, Marc Fischlin, and Tommaso Gagliardoni. "The Fiat-Shamir Transformation in a Quantum World". In: *Advances in Cryptology - ASIACRYPT 2013*. Ed. by Kazue Sako and

Palash Sarkar. Berlin, Heidelberg: Springer, 2013, pp. 62–81. ISBN: 978-3-642-42045-0.

[DFM20] Jelle Don, Serge Fehr, and Christian Majenz. "The Measure-and-Reprogram Technique 2.0: Multi-round Fiat-Shamir and More". In: *Advances in Cryptology – CRYPTO 2020*. Ed. by Daniele Micciancio and Thomas Ristenpart. Cham: Springer International Publishing, 2020, pp. 602–631.

[DFMS19] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. "Security of the Fiat-Shamir Transformation in the Quantum Random-Oracle Model". In: *Advances in Cryptology – CRYPTO 2019*. Ed. by Alexandra Boldyreva and Daniele Micciancio. Cham: Springer International Publishing, 2019, pp. 356–383.

[DFMS22a] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. "Online-Extractability in the Quantum Random-Oracle Model". In: *Advances in Cryptology – EUROCRYPT 2022*. Ed. by Orr Dunkelman and Stefan Dziembowski. Cham: Springer International Publishing, 2022, pp. 677–706.

[DFMS22b] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. "Efficient NIZKs and Signatures from Commit-and-Open Protocols in the QROM". In: *Advances in Cryptology – CRYPTO 2022*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Cham: Springer Nature Switzerland, 2022, pp. 729–757.

[DH76] W. Diffie and M. Hellman. "New directions in cryptography". In: *IEEE Transactions on Information Theory* 22.6 (1976), pp. 644–654. DOI: 10.1109/TIT.1976.1055638.

[DJ92] David Deutsch and Richard Jozsa. "Rapid Solution of Problems by Quantum Computation". In: *Proceedings of the Royal Society of London Series A* 439.1907 (Dec. 1992), pp. 553–558. DOI: 10.1098/rspa.1992.0167.

[DKL+18a] Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. "CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2018.1 (Feb. 2018), pp. 238–268. DOI: 10.13154/tches.v2018.i1.238-268. URL: https://tches.iacr.org/index.php/TCHES/article/view/839.

[DKL+18b] Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. "CRYSTALS-

Dilithium: A Lattice-Based Digital Signature Scheme". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2018.1 (Feb. 2018), pp. 238–268. DOI: 10.13154/tches.v2018.i1.238-268. URL: https://tches.iacr.org/index.php/TCHES/article/view/839.

[DKR+21] Christoph Dobraunig, Daniel Kales, Christian Rechberger, Markus Schofnegger, and Greg Zaverucha. *Shorter Signatures Based on Tailor-Made Minimalist Symmetric-Key Crypto.* Cryptology ePrint Archive, Report 2021/692. https://ia.cr/2021/692. 2021.

[Ell87] James Ellis. *The history of Non-Secret Encryption.* 1987. URL: https://cryptocellar.org/cesg/ellis.pdf.

[ES15] Edward Eaton and Fang Song. "Making Existential-unforgeable Signatures Strongly Unforgeable in the Quantum Random-oracle Model". In: *10th Conference on the Theory of Quantum Computation, Communication and Cryptography.* 2015, p. 147.

[Feh18] Serge Fehr. "Classical Proofs for the Quantum Collapsing Property of Classical Hash Functions". In: *Theory of Cryptography Conference - TCC2018, volume 11240 of Lecture Notes in Computer Science* (2018), pp. 315–338. eprint: 2018/887.

[Feh22] Serge Fehr. *Multipartite Quantum Systems.* University Lecture Notes. 2022. URL: https://homepages.cwi.nl/~fehr/QC2022/Ch2.pdf.

[Fis05] Marc Fischlin. "Communication-Efficient Non-interactive Proofs of Knowledge with Online Extractors". In: *Advances in Cryptology – CRYPTO 2005.* Ed. by Victor Shoup. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 152–168. ISBN: 978-3-540-31870-5.

[FJ20] Patrick Fischlin Marc and Harasser and Christian Janson. "Signatures from Sequential-OR Proofs". In: *Advances in Cryptology - EUROCRYPT 2020.* Ed. by Anne Canteaut and Yuval Ishai. Vol. 12107. Lecture Notes in Computer Science. Springer, 2020, pp. 212–244. DOI: 10.1007/978-3-030-45727-3_8.

[FKMV12] Sebastian Faust, Markulf Kohlweiss, Giorgia Azzurra Marson, and Daniele Venturi. "On the Non-malleability of the Fiat-Shamir Transform". In: *Indocrypt 2012.* Vol. 7668 LNCS. 2012, pp. 60–79. ISBN: 9783642349300. DOI: 10.1007/978-3-642-34931-7_5.

[FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. “How to Enhance the Security of Public-Key Encryption at Minimum Cost”. In: *Public Key Cryptography*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 53–68. ISBN: 978-3-540-49162-0.

[FS87] Amos Fiat and Adi Shamir. “How To Prove Yourself: Practical Solutions to Identification and Signature Problems”. In: *Advances in Cryptology — CRYPTO’ 86*. Ed. by Andrew M. Odlyzko. Berlin, Heidelberg: Springer, 1987, pp. 186–194. ISBN: 978-3-540-47721-1.

[GHHM21] Alex B. Grilo, Kathrin Hövelmanns, Andreas Hülsing, and Christian Majenz. “Tight Adaptive Reprogramming in the QROM”. In: *Advances in Cryptology – ASIACRYPT 2021*. Ed. by Mehdi Tibouchi and Huaxiong Wang. Cham: Springer International Publishing, 2021, pp. 637–667. ISBN: 978-3-030-92062-3.

[GMO16] Irene Giacomelli, Jesper Madsen, and Claudio Orlandi. “ZKBoo: Faster Zero-Knowledge for Boolean Circuits”. In: *25th USENIX Security Symposium (USENIX Security 16)*. Austin, TX: USENIX Association, 2016, pp. 1069–1083. ISBN: 978-1-931971-32-4. URL: `https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/giacomelli`.

[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. Providence, Rhode Island, USA: Association for Computing Machinery, 1985, pp. 291–304. ISBN: 0897911512. DOI: `10.1145/22145.22178`. URL: `https://doi.org/10.1145/22145.22178`.

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. “Proofs That Yield Nothing but Their Validity or All Languages in NP Have Zero-Knowledge Proof Systems”. In: *J. ACM* 38.3 (July 1991), pp. 690–728. ISSN: 0004-5411. DOI: `10.1145/116825.116852`. URL: `https://doi.org/10.1145/116825.116852`.

[GPS22] Shay Gueron, Edoardo Persichetti, and Paolo Santini. “Designing a Practical Code-Based Signature Scheme from Zero-Knowledge Proofs with Trusted Setup”. In: *Cryptography* 6.1 (2022). ISSN: 2410-387X. URL: `https://www.mdpi.com/2410-387X/6/1/5`.

[HHK17] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. “A Modular Analysis of the Fujisaki-Okamoto Transformation”. In: *Theory* *of Cryptography*. Ed. by Yael Kalai and Leonid Reyzin. Cham: Springer International Publishing, 2017, pp. 341–371. ISBN: 978-3-319-70500-2.

[HM21] Yassine Hamoudi and Frédéric Magniez. "Quantum Time-Space Tradeoff for Finding Multiple Collision Pairs". In: *16th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2021)*. Ed. by Min-Hsiu Hsieh. Vol. 197. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, 1:1–1:21. ISBN: 978-3-95977-198-6. DOI: `10.4230/LIPIcs.TQC.2021.1`. URL: `https://drops.dagstuhl.de/opus/volltexte/2021/13996`.

[IKOS07a] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. "Zero-Knowledge from Secure Multiparty Computation". In: *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*. STOC '07. San Diego, California, USA: Association for Computing Machinery, 2007, pp. 21–30. ISBN: 9781595936318. DOI: `10.1145/1250790.1250794`. URL: `https://doi.org/10.1145/1250790.1250794`.

[IKOS07b] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. "Zero-knowledge from secure multiparty computation". In: *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing - STOC '07* (2007), p. 21. ISSN: 07378017. URL: `http://portal.acm.org/citation.cfm?doid=1250790.1250794`.

[Ker83] Auguste Kerckhoffs. "La cryptographie militaire". In: *Journal des sciences militaires* IX.Jan. (1883), pp. 5–83. URL: `http://www.petitcolas.net/fabien/kerckhoffs/`.

[KKPP20] Shuichi Katsumata, Kris Kwiatkowski, Federico Pintore, and Thomas Prest. "Scalable Ciphertext Compression Techniques for Post-quantum KEMs and Their Applications". In: *Advances in Cryptology – ASIACRYPT 2020*. Ed. by Shiho Moriai and Huaxiong Wang. Cham: Springer International Publishing, 2020, pp. 289–320. ISBN: 978-3-030-64837-4.

[KKW18] Jonathan Katz, Vladimir Kolesnikov, and Xiao Wang. "Improved Non-Interactive Zero Knowledge with Applications to Post-Quantum Signatures". In: *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. CCS '18. Toronto, Canada: Association for Computing Machinery, 2018, pp. 525–537. ISBN: 9781450356930. URL: https://doi.org/10.1145/3243734.3243805.

[KLS18] Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner. “A Concrete Treatment of Fiat-Shamir Signatures in the Quantum Random-Oracle Model”. In: *Advances in Cryptology – EUROCRYPT 2018*. Ed. by Jesper Buus Nielsen and Vincent Rijmen. Cham: Springer, 2018, pp. 552–586. ISBN: 978-3-319-78372-7.

[KM15] Neal Koblitz and Alfred J. Menezes. “The random oracle model: a twenty-year retrospective”. In: *Designs, Codes and Cryptography* 77 (2015), pp. 587–610. DOI: 10.1007/s10623-015-0094-2.

[KZ20] Daniel Kales and Greg Zaverucha. “Improving the Performance of the Picnic Signature Scheme”. English. In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2020.4 (Sept. 2020). CHES 2020 : 2020 Annual Conference on Cryptographic Hardware and Embedded Systems ; Conference date: 14-09-2020 Through 17-09-2020, pp. 154–188. ISSN: 2569-2925. DOI: https://doi.org/10.13154/tches.v2020.i4.154-188.

[LWW04] Joseph K. Liu, Victor K. Wei, and Duncan S. Wong. “Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups”. In: *Information Security and Privacy*. Ed. by Huaxiong Wang, Josef Pieprzyk, and Vijay Varadharajan. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 325–335. ISBN: 978-3-540-27800-9.

[Lyu09] Vadim Lyubashevsky. “Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures”. In: *Advances in Cryptology – ASIACRYPT 2009*. Ed. by Mitsuru Matsui. Berlin, Heidelberg: Springer, 2009, pp. 598–616. ISBN: 978-3-642-10366-7.

[Lyu12] Vadim Lyubashevsky. “Lattice Signatures without Trapdoors”. In: *Advances in Cryptology – EUROCRYPT 2012*. Ed. by David Pointcheval and Thomas Johansson. Berlin, Heidelberg: Springer, 2012, pp. 738–755. ISBN: 978-3-642-29011-4.

[LZ19a] Qipeng Liu and Mark Zhandry. “On Finding Quantum Multi-collisions”. In: *Advances in Cryptology – EUROCRYPT 2019*. Ed. by Yuval Ishai and Vincent Rijmen. Cham: Springer International Publishing, 2019, pp. 189–218. ISBN: 978-3-030-17659-4.

[LZ19b] Qipeng Liu and Mark Zhandry. “Revisiting Post-quantum Fiat-Shamir”. In: *Advances in Cryptology – CRYPTO 2019*. Ed. by Alexandra Boldyreva and Daniele Micciancio. Cham: Springer International Publishing, 2019, pp. 326–355. ISBN: 978-3-030-26951-7.

[Mer78] Ralph C. Merkle. “Secure Communications over Insecure Channels”. In: *Commun. ACM* 21.4 (Apr. 1978), pp. 294–299. ISSN: 0001-0782. URL: `https://doi.org/10.1145/359460.359473`.

[NC11] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition.* 10th. New York, NY, USA: Cambridge University Press, 2011. ISBN: 1107002176, 9781107002173.

[Pas03] Rafael Pass. “On Deniability in the Common Reference String and Random Oracle Model”. In: *Advances in Cryptology - CRYPTO 2003.* Ed. by Dan Boneh. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 316–337. ISBN: 978-3-540-45146-4.

[Pas04] Rafael Pass. “Alternative variants of zero-knowledge proofs”. PhD thesis. KTH Stockholm, 2004.

[PS96] David Pointcheval and Jacques Stern. “Security Proofs for Signature Schemes”. In: *LNCS* 1070 (1996), pp. 387–398. URL: `https://www.di.ens.fr/%7B~%7Dpointche/Documents/Papers/1996%7B%5C_%7Deurocrypt.pdf`.

[RCB22] Prasanna Ravi, Anupam Chattopadhyay, and Shivam Bhasin. “Security and Quantum Computing: An Overview”. In: *2022 IEEE 23rd Latin American Test Symposium (LATS).* 2022, pp. 1–6. DOI: `10.1109/LATS57337.2022.9936966`.

[RSA78] R. L. Adleman Rivest, A. Shamir, and L. Adleman. “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”. In: *Commun. ACM* 21.2 (Feb. 1978), pp. 120–126. ISSN: 0001-0782. URL: `https://doi.org/10.1145/359340.359342`.

[Sho94] P.W. Shor. “Algorithms for quantum computation: discrete logarithms and factoring”. In: *Proceedings 35th Annual Symposium on Foundations of Computer Science.* 1994, pp. 124–134. DOI: `10.1109/SFCS.1994.365700`.

[Sim97] Daniel R. Simon. “On the Power of Quantum Computation”. In: *SIAM Journal on Computing* 26.5 (1997), pp. 1474–1483. DOI: `10.1137/S0097539796298637`.

[SSH11] Koichi Sakumoto, Taizo Shirai, and Harunaga Hiwatari. “Public-Key Identification Schemes Based on Multivariate Quadratic Polynomials”. In: *Advances in Cryptology – CRYPTO 2011.* Ed.

by Phillip Rogaway. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 706–723. ISBN: 978-3-642-22792-9.

[TDJ+22] Gang Tang, Dung Hoang Duong, Antoine Joux, Thomas Plantard, Youming Qiao, and Willy Susilo. "Practical Post-Quantum Signature Schemes from Isomorphism Problems of Trilinear Forms". In: *Advances in Cryptology – EUROCRYPT 2022*. Ed. by Orr Dunkelman and Stefan Dziembowski. Cham: Springer International Publishing, 2022, pp. 582–612. ISBN: 978-3-031-07082-2.

[TU16] Ehsan Ebrahimi Targhi and Dominique Unruh. "Post-Quantum Security of the Fujisaki-Okamoto and OAEP Transforms". In: *Theory of Cryptography*. Ed. by Martin Hirt and Adam Smith. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 192–216. ISBN: 978-3-662-53644-5.

[Unr12] Dominique Unruh. "Quantum Proofs of Knowledge". In: *Advances in Cryptology – EUROCRYPT 2012*. Ed. by David Pointcheval and Thomas Johansson. Berlin, Heidelberg: Springer, 2012, pp. 135–152. ISBN: 978-3-642-29011-4.

[Unr14a] Dominique Unruh. "Quantum Position Verification in the Random Oracle Model". In: *Advances in Cryptology – CRYPTO 2014*. Ed. by Juan A. Garay and Rosario Gennaro. Berlin, Heidelberg: Springer, 2014, pp. 1–18. ISBN: 978-3-662-44381-1.

[Unr14b] Dominique Unruh. "Revocable Quantum Timed-Release Encryption". In: *Advances in Cryptology – EUROCRYPT 2014*. Ed. by Phong Q. Nguyen and Elisabeth Oswald. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 129–146. ISBN: 978-3-642-55220-5.

[Unr15a] Dominique Unruh. *Computationally binding quantum commitments*. Cryptology ePrint Archive, Paper 2015/361. `https://eprint.iacr.org/2015/361`. 2015. URL: `https://eprint.iacr.org/2015/361`.

[Unr15b] Dominique Unruh. "Non-Interactive Zero-Knowledge Proofs in the Quantum Random Oracle Model". In: *Advances in Cryptology - EUROCRYPT 2015*. Ed. by Elisabeth Oswald and Marc Fischlin. Berlin, Heidelberg: Springer, 2015, pp. 755–784. ISBN: 978-3-662-46803-6.

[Unr16] Dominique Unruh. "Computationally Binding Quantum Commitments". In: *Advances in Cryptology – EUROCRYPT 2016*. Ed.

by Marc Fischlin and Jean-Sébastien Coron. Berlin, Heidelberg: Springer, 2016, pp. 497–527. ISBN: 978-3-662-49896-5.

[Unr17] Dominique Unruh. "Post-quantum Security of Fiat-Shamir". In: *Advances in Cryptology – ASIACRYPT 2017.* Ed. by Tsuyoshi Takagi and Thomas Peyrin. Cham: Springer, 2017, pp. 65–95. ISBN: 978-3-319-70694-8.

[Wik18] Douglas Wikström. *Special Soundness Revisited.* Cryptology ePrint Archive, Report 2018/1157. `https://ia.cr/2018/1157`. 2018.

[YZ21] Takashi Yamakawa and Mark Zhandry. "Classical vs Quantum Random Oracles". In: *Advances in Cryptology – EUROCRYPT 2021.* Ed. by Anne Canteaut and François-Xavier Standaert. Cham: Springer International Publishing, 2021, pp. 568–597. ISBN: 978-3-030-77886-6.

[Zha12] Mark Zhandry. "How to Construct Quantum Random Functions". In: *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science.* IEEE, Oct. 2012, pp. 679–687. ISBN: 978-0-7695-4874-6. DOI: `10.1109/FOCS.2012.37`. URL: `https://eprint.iacr.org/2012/182.pdf`.

[Zha15a] Mark Zhandry. "A note on the quantum collision and set equality problems". In: *Quantum Information and Computation* 15.7-8 (2015), pp. 557–567.

[Zha15b] Mark Zhandry. "Secure identity-based encryption in the quantum random oracle model". In: *International Journal of Quantum Information* 13.04 (2015), p. 1550014.

[Zha19a] Mark Zhandry. "How to Record Quantum Queries, and Applications to Quantum Indifferentiability". In: *Advances in Cryptology – CRYPTO 2019.* Ed. by Alexandra Boldyreva and Daniele Micciancio. Full Version (1 March 2019): `https://eprint.iacr.org/2018/276/20190301:184107`. Cham: Springer International Publishing, 2019, pp. 239–268. ISBN: 978-3-030-26951-7.

[Zha19b] Mark Zhandry. "Quantum Lightning Never Strikes the Same State Twice". In: *Advances in Cryptology – EUROCRYPT 2019.* Ed. by Yuval Ishai and Vincent Rijmen. Cham: Springer International Publishing, 2019, pp. 408–438. ISBN: 978-3-030-17659-4.

Samenvatting

Het doel van dit proefschrift is om nieuwe technieken te presenteren voor het geven van veiligheidsbewijzen voor cryptografische protocollen die opgewassen zijn tegen kwantumtegenstanders. De meeste resultaten vallen binnen de context van een geïdealiseerd model dat het 'quantum random-oracle model' (QROM) wordt genoemd. Een bijzondere uitdaging is om – enkel uit de interactie met het orakel – een stukje kennis te extraheren dat een tegenstander bezit, zonder dat de effecten van de ineenstorting van de golffunctie, veroorzaakt door de observatie van een kwantumtoestand, roet in het eten gooien.

In hoofdstuk 3 gaan we die uitdaging aan door de nieuwe 'measure-and-reprogram' techniek te introduceren. We gebruiken deze techniek om het eerste post-kwantum veiligheidsbewijs van de Fiat-Shamir transformatie te bewerkstelligen – zowel de standaard als de multi-ronde versies – waardoor een QROM veiligheidsreductie mogelijk wordt voor een grote klasse van post-kwantum handtekeningsystemen.

Het belangrijkste obstakel dat overwonnen moet worden is dat het observeren van een van de queries van de tegenstander mogelijk zijn toestand verstoort, waardoor het in het algemeen moeilijk is om het gedrag van de tegenstander tijdens het verdere verloop te voorspellen. De toestand na de meting kan potentieel slechts een verwaarloosbare overeenkomst hebben met het origineel, als deze bestaat uit een superpositie over exponentieel veel query-inputs. In onze analyse zijn we echter in staat om de verstoring veroorzaakt door de meting te beperken. We breiden onze techniek uit naar multi-ronde Fiat-Shamir en laten zien dat het opgelopen reductieverlies optimaal is in het algemene geval.

Waar in het vorige hoofdstuk extractie van een query-input mogelijk was ten koste van enige (polynomiale) verstoring, en extractie in een interactief protocol Π afhankelijk was van 'terugspoelen', laten we in hoofdstuk 4 zien dat zulke nadelen niet altijd nodig zijn. We tonen aan dat in bepaalde gevallen – bijvoorbeeld wanneer het eerste bericht van Π bestaat uit een hash-gebaseerde commitment die we modelleren als een random-oracle – *online extractie* mogelijk is. 'Online' betekent in dit geval lineair (niet gebruik makend van 'terugspoelen') en on-the-fly (tijdens de uitvoering van het protocol en zonder het te verstoren). Terugspoelen veroorzaakt vaak een reductieverlies (omdat de tegenstander twee keer moet slagen) en uiteraard veroorzaakt een verstoring in de toestand van de tegenstander ook een verlies. Indien mogelijk heeft online extractie dus de voorkeur.

Onze eerste belangrijke toepassing is de zogenaamde ‘commit-and-open protocollen’. Deze protocollen vormen een subklasse van sigma-protocollen, waarbij het eerste bericht een verzameling commitments is, en de challenge bepaalt welke subset hiervan de prover moet openen in het derde bericht. De tweede belangrijke toepassing is de Fujisaki-Okamoto transformatie, die ten grondslag ligt aan veel KEM’s in de bekende NIST post-quantum competitie. We geven het eerste volledige post-kwantum veiligheidsbewijs van de FO transformatie.

Tenslotte laten we in hoofdstuk 5 opnieuw online extractability zien voor commit-and-open protocollen, maar nu van de Fiat-Shamir getransformeerde niet-interactieve versie ervan. Hoewel de technieken uit hoofdstuk 3 en 4 gecombineerd zouden kunnen worden om een veiligheidsreductie voor zulke protocollen te krijgen, zou deze strategie niet resulteren in *online* extractie vanwege de verstoring die wordt veroorzaakt door de meting in de measure-and-reprogram techniek, waardoor er een $(2q+1)^2$ multiplicatief verlies optreedt voor de succeskans van de reductie.

Onze veiligheidsreductie in dit hoofdstuk is optimaal: Als een prover een geldig bewijs uitvoert, slaagt de online-extractor, behalve met een kleine kans, die verband houdt met collision- en preimage-aanvallen op de betrokken hashfuncties. Ons resultaat geldt ook voor een variant van de Fiat-Shamir transformatie waar een digitaal handtekeningsysteem uitkomt. Het maakt daardoor voor het eerst een multiplicatief optimale veiligheidsreductie in de QROM mogelijk voor bijvoorbeeld de digitale handtekeningsystemen gebaseerd op het MPC-in-the-head paradigma, zoals Picnic, Banquet en Rainier.

Acknowledgements

I would like to thank my advisors Prof.dr. Serge Fehr and Prof.dr. Ronald Cramer for their wonderful guidance over the past few years.

Serge, my promotor, has been a great example for me in many ways. His exceptional talent for clear scientific communication, an undervalued trait in the mathematical sciences, was inspiring to observe. Serge always pushed me to find the simplest and most elegant solution, and to refrain from cluttering up my proofs with unnecessary complications. As a supervisor, Serge is literally always available for feedback and discussions, and I have often felt blessed when comparing my amount of supervision hours with the average PhD student. Working with him has been a great pleasure.

Ronald, my second promotor, is the one who landed me the job, and I thank him for taking that leap of faith with a master student who's grades were not above average. Ronald cheerfully leads his group and makes you feel part of his tribe. I want to especially thank him for giving me the opportunity to experience a more applied side of cryptographic research over the past year.

Christian Schaffner deserves a mention here, because this thesis would not exist if he had not introduced me to the science of cryptography during my master of logic programme. If scientific output was measured by all the work produced by oneself and that of students inspired by one's educational enthusiasm, Chris would be top of the list.

Yu-Hsuan Huang, my PhD would not have been the same without you coming all the way from Taiwan just to have endless discussions with me on mind-boggling details of the QROM or other cryptographic niches. Thank you for being so persistent and resourceful in getting to the bottom of things, forcing me to think hard about every little aspect of the work we were doing together.

A big thanks to dr. Anne Broadbent, dr. Nicholas Spooner and Prof.dr. Dominique Unruh for taking part in the reading committee and providing me with valuable feedback.

CWI is not a person, but definitely an institute to be thankful for. And of course it is all the people at CWI that make it the great place it is. From the warm welcome by Bikkie, when I spent long days in the CWI library writing my master thesis, to the many inspiring and fun colleagues in the Crypto group (including our emphatic secretary Suzanne!), thank you for making CWI the coolest place for research I know of.

Finally, I am intensely grateful for the love and support that I receive everyday from my family, friends and partner. Thank you for bringing me all the joy that you do, and for helping me more than you probably think.

Curriculum Vitae

Jelle Don was born in Amsterdam, The Netherlands, on January 30, 1990. He obtained his high school diploma from the Montessori Lyceum Amsterdam in 2008.

After partially completing a bachelor's programme in mathematics, Jelle obtained his bachelor's degree in artificial intelligence from Utrecht University in 2014. He then continued his studies at the Institute for Logic, Language and Computation in Amsterdam (University of Amsterdam). In 2018, he obtained a master's degree in logic, with a focus on theoretical computer science.

Jelle started working as a PhD-student in the Cryptology Group of the Centrum Wiskunde & Informatica (CWI) in 2019, under the supervision of Prof.dr. Serge Fehr. Since 2023, he combines his position at CWI with a one day per week volunteer job as the chair of housing cooperative de Torteltuin.

Publications

[DFMS19] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. "Security of the Fiat-Shamir Transformation in the Quantum Random-Oracle Model". In: *Advances in Cryptology – CRYPTO 2019*. Ed. by Alexandra Boldyreva and Daniele Micciancio. Cham: Springer International Publishing, 2019, pp. 356–383.

[DFM20] Jelle Don, Serge Fehr, and Christian Majenz. "The Measure-and-Reprogram Technique 2.0: Multi-round Fiat-Shamir and More". In: *Advances in Cryptology – CRYPTO 2020*. Ed. by Daniele Micciancio and Thomas Ristenpart. Cham: Springer International Publishing, 2020, pp. 602–631.

[DFMS22a] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. "Online-Extractability in the Quantum Random-Oracle Model". In: *Advances in Cryptology – EUROCRYPT 2022*. Ed. by Orr

Dunkelman and Stefan Dziembowski. Cham: Springer International Publishing, 2022, pp. 677–706.

[DFH22] Jelle Don, Serge Fehr, and Yu-Hsuan Huang. “Adaptive Versus Static Multi-oracle Algorithms, and Quantum Security of a Split-Key PRF”. In: *Theory of Cryptography.* Ed. by Eike Kiltz and Vinod Vaikuntanathan. Cham: Springer Nature Switzerland, 2022, pp. 33–51. ISBN: 978-3-031-22318-1.

[DFMS22b] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. “Efficient NIZKs and Signatures from Commit-and-Open Protocols in the QROM”. In: *Advances in Cryptology – CRYPTO 2022.* Ed. by Yevgeniy Dodis and Thomas Shrimpton. Cham: Springer Nature Switzerland, 2022, pp. 729–757.

[BBD+23] Manuel Barbosa, Gilles Barthe, Christian Doczkal, Jelle Don, Serge Fehr, Benjamin Grégoire, Yu-Hsuan Huang, Andreas Hülsing, Yi Lee, and Xiaodi Wu. “Fixing and Mechanizing the Security Proof of Fiat-Shamir with Aborts and Dilithium”. In: *Advances in Cryptology – CRYPTO 2023.* Ed. by Helena Handschuh and Anna Lysyanskaya. Cham: Springer Nature Switzerland, 2023, pp. 358–389. ISBN: 978-3-031-38554-4.

[DFHS23] Jelle Don, Serge Fehr, Yu-Hsuan Huang, and Patrick Struck. *On the (In)Security of the BUFF Transform.* Cryptology ePrint Archive, Paper 2023/1634. 2023. URL: https://eprint.iacr.org/2023/1634.

KNOWLEDGE EXTRACTION IN THE QUANTUM RANDOM-ORACLE MODEL

The aim of this thesis is to present novel techniques for proving cryptographic schemes secure against quantum adversaries. Most results are within the context of an idealized model called the ‘quantum random-oracle model’. A particular challenge is to extract some piece of knowledge an adversary possesses just from its interaction with an oracle, while mitigating the effects of the collapse of the wave function caused by the observation of a quantum state.