A Survey on Security Reductions in Post-Quantum Cryptography

Thomas Attema[1,4], Ronald Cramer[1,2], Serge Fehr[1,2],
Yu-Hsuan Huang*[,3], Bor de Kock[4], and Jana Sotáková[4]

[1]CWI, Cryptology Group, Amsterdam, The Netherlands
[2]Leiden University, Mathematical Institute, Leiden, The Netherlands
[3]Max Planck Institute for Security and Privacy, Germany
[4]TNO, Applied Cryptography & Quantum Applications, The Hague, The Netherlands

April 30, 2026

Abstract

It is obviously necessary that the security of post-quantum cryptographic schemes is based on computational problems that are hard to solve even with a quantum computer (unlike, e.g., factoring). Examples of such computational problems appear in the theory of lattices or in coding theory. However, this is not sufficient: also the *security proof*, which comes in the form of an algorithmic reduction that turns any hypothetical attacker into an algorithm that solves the considered hard computational problem, needs to be valid when considering quantum computing as the model of computation.

In this work, we provide an overview of the hurdles one typically encounters when proving the security of post-quantum cryptographic schemes, and we elaborate on some of the mathematical techniques that have been developed in order to overcome these hurdles (to some extent). We also discuss the caveat that even when a security proof can be established (by reducing the security to a quantum-hard computational problem), the reduction often suffers from a larger *reduction loss*, compared to when proving classical security, which negatively affects the *concrete* security.

In the second part of this work, we offer a survey of the respective reduction losses in (1) generic transformations that are often used in the design of cryptographic schemes (like the Fiat–Shamir and Fujisaki–Okamoto transformations), and (2) some concrete cryptographic schemes (with a focus on those standardized by NIST), when considering classical and when considering post-quantum security.

Finally, we consider the notion of *bit security*, the standard measure of the concrete security of a cryptographic scheme (or of the hardness of an underlying computational problem). A natural question is how the bit security is affected by the different reduction losses we encountered. Surprisingly, we observe that a better or worse reduction (in terms of the reduction loss) is not always reflected as such in the bit security. We explain this phenomenon by the fact that the bit security is oblivious to the actual advantage–time function, and instead considers a worst-case behavior of that function. Thus, by exploiting the actual advantage–time function there is potential to get more accurate (i.e., less conservative) estimates for the concrete security.

*This work was primarily conducted while Yu-Hsuan Huang was employed at CWI Amsterdam.

This work was funded by the Netherlands National Communication Security Authority (NL-NCSA), part of the Dutch General Intelligence and Security Service (AIVD).

Contents

1 Introduction **3**

2 Provable Security **4**
2.1 Defining Security . . . 4
2.2 Security Reductions . . . 5
2.3 Tightness and the Reduction Loss . . . 6
2.4 Idealized Models . . . 7
2.5 Security Models and Proof Strategies . . . 8

3 Technical Preliminary **9**
3.1 Public-Key Encryption . . . 9
3.2 Key Encapsulation . . . 11
3.3 Digital Signatures . . . 12

4 Quantum Attacks and Provable Security **12**
4.1 The Computational Assumption . . . 13
4.2 The Security Reduction . . . 13
4.2.1 Rewinding . . . 14
4.2.2 Superposition Access to the Random Oracle . . . 14
4.2.3 New Security Definitions . . . 14
4.3 Q1 and Q2 Security Models . . . 15

5 Proof Techniques in the (Q)ROM **15**
5.1 Lazy Sampling and the Compressed Oracle Technique . . . 15
5.2 Adaptive Programming . . . 17
5.3 Measure and Reprogram . . . 18

6 Quantum Security of Generic Design Principles **21**
6.1 Fujisaki–Okamoto (FO) . . . 22
6.2 Fiat–Shamir (FS) . . . 24
6.3 Fiat–Shamir with Aborts . . . 27

7 Quantum Security of Concrete Schemes **28**
7.1 Kyber . . . 28
7.2 HQC . . . 30
7.3 FrodoKEM . . . 31
7.4 Dilithium . . . 32
7.5 Falcon . . . 33
7.6 SPHINCS$^+$ . . . 34

8 Reduction Loss versus Bit Security **34**

9 Afterthoughts and Open Ends **36**
9.1 Non-ROM Schemes . . . 37
9.2 Beyond Standard Security . . . 37
9.3 Hybrid-Combiners . . . 38

1 Introduction

Background. The impact of the quantum computer on the security of cryptographic schemes has been a popular research topic for years, ever since it became clear that the algorithms by Shor [Sho94] and Grover [Gro96] could be used to break or decrease the security of most cryptographic schemes in mainstream use. The search for *quantum-resistant* or *post-quantum* cryptography (PQC) was accelerated by the United States National Institute for Standards in Technology (NIST), which in 2016 initiated a standardization process for key encapsulation methods (KEM) and digital signature algorithms (DSA) [NIS16]. This standardization process — often referred to as *the NIST competition* — led to 69 submitted candidate algorithms from the industrial and academic cryptographic community. In 2022 NIST announced the intention to standardize four algorithms [NIS22], followed by the release of the first standard documents in 2024 [NIS24b, NIS24a, NIS24c]. Besides these new standards the competition has lead to the development of many other candidate schemes, implementations, cryptanalysis results and other research in the broader area of PQC.

The security of a cryptographic scheme is established by basing the security on a well-defined — and hopefully well-studied — computational problem that is believed to be hard to solve. Ideally, one can rigorously prove that any hypothetical attacker against the scheme gives rise to an algorithm that solves the computational problem, which then contradicts the assumed hardness of the latter. Thus, such a *security reduction*, also referred to as a *security proof*, shows that it is not only *sufficient* to solve the considered hard computational problem in order to break the scheme, but it is also *necessary*. This ensures that there is no overlooked weakness in the design of the scheme, and so no shortcut in breaking the scheme that circumvents solving the considered hard problem.

In order to offer quantum-resistance, the security of a post-quantum cryptographic scheme must thus be based on a computational problem that is hard to solve also by means of quantum algorithms. However, this is not the end of the story: also the security reduction, as well as its analysis, must be sound when the model of computation is extended to quantum computing. In certain cases, the reduction and its analysis are pretty much independent on the details of the model of computation, and so carry over directly to the case of quantum; but in other cases they do not. In particular for schemes that are designed to be particularly efficient, like the schemes from the NIST competition, the security reduction is typically affected (since those schemes are mostly analyzed in the random oracle model).

Next to the *existence* of a security reduction is the question of the *quality* of the reduction. Namely, in a security reduction, when turning a (hypothetical) attacker against the scheme into an algorithm for the considered computational problem, there may be a loss in success probability and/or an increase in run time, which is referred to as *reduction loss*. The reduction loss directly affects the concrete security of a scheme, and a large reduction loss may require the scheme to be instantiated with larger parameters. Consequently, the reduction loss may also affect the scheme's efficiency, and the goal is therefore to have this loss as small as possible.

Contribution. This article provides a survey on how the provable security of cryptographic schemes is affected when introducing quantum attackers, i.e., attackers that may be equipped with a quantum computer capable of running quantum algorithms. Rather than zooming in on the computational problems that are suitable for post-quantum cryptography, we focus on the security reductions.

In the first part, we discuss more broadly on how security reductions may be affected by quantum attacks, and what are the typical technical obstacles that one encounters when trying to lift a security reduction to the setting where the attacker may be quantum. A particularly important example is the case of the random oracle model (ROM), where a cryptographic hash function is treated as an idealized object, forcing the attacker to make queries to an oracle in order to "compute" the hash of any input. In the case of quantum attackers, these queries may then be in quantum-superposition, making typical arguments in the security reductions break down.

This discussion is then followed by an exposition of a selection of techniques that have been developed over the last years, in order to deal with the obstacles caused by the ROM. Concretely, we give a high-level exposition of Zhandry's *compressed oracle* technique [Zha19] and of *adaptive reprogramming* by [GHHM21], and, given

that this is quite self-contained, we show the *measure-and-reprogram* technique from [DFMS19, DFM20] in technical detail. While powerful techniques are available nowadays, whether a security proof in the ROM can be lifted to the case of quantum attackers, and what is the impact on the reduction loss, very much remains a case-by-case study.

In the second part of this survey, we compare the respective reduction losses when considering classical and quantum attacks, for several concrete cases. Specifically, we offer this comparison for the Fujisako–Okamoto transformation and the Fiat–Shamir transformation, and we offer this comparison for the KEMs and digital signature schemes chosen by the NIST for standardization. Not surprisingly, in most of the cases, the reduction loss is worse when considering quantum attacks, sometimes considerably worse.

Finally, we consider the notion of *bit security*, which compresses the advantage–time function into a single parameter that captures the best achievable advantage–time ratio for the attacker. A natural question is therefore how reduction losses translate into losses in bit security, and how different types of reduction losses impact the resulting security guarantees. Surprisingly, it turns out that a worse reduction loss (with a clearly worse advantage-versus-time behavior) is not necessarily reflected by a larger loss in bit security. We work out the loss in bit security for several typical reductions, and observe that a poorer reduction loss may not be reflected as such when considering the bit security (only). Thus, the notion of bit security can obscure how reduction losses actually affect security.

We explain this phenomenon by the fact that the bit security is oblivious to the actual advantage–time function, and instead considers a worst-case behavior of that function. Thus, by exploiting the actual advantage–time function there is potential to get more accurate (i.e., less conservative) estimates for the concrete security.

Organization of the Paper The first sections of this survey provide a general introduction to (quantum) security proofs, while the later sections are more technical and assume a stronger background in cryptography. Section 2 reviews the foundations of provable security, including security models and standard proof techniques such as game-based arguments. Section 3 introduces the required technical preliminaries and formal security notions. Section 4 examines the impact of quantum computation on cryptographic security and the extent to which classical proofs extend to the quantum setting. In Section 5 surveys general methodologies for adapting classical proofs to the quantum setting, together with their inherent limitations. Section 6 and Section 7 analyze the implications of quantum computation for standard design paradigms and for concrete constructions. Section 8 relates provable security and reduction loss to the concrete notion of bit security. Finally, Section 9 concludes with further remarks on non-ROM proofs, reduction loss versus bit security, and security beyond the standard model.

2 Provable Security

A cryptographic scheme is designed to achieve a specific security property. For example, it should be infeasible for an attacker to read a secret message, forge a signature or figure out the secret key. More precisely, it should take an attacker an infeasible amount of computational resources to break the scheme. To achieve this, cryptography relies on *computationally hard problems*, namely mathematical problems for which no efficient, i.e., polynomial time, algorithm can succeed with non-negligible[1] success probability. Hence, equivalently, a computational problem is considered hard if any algorithm that solves it with non-negligible success probability must use super-polynomial (e.g., sub-exponential or exponential) computational resources in the security parameter λ.

2.1 Defining Security

To prove the security of a cryptographic scheme, we must first formally define what it means for the scheme to be secure. This is typically done via a *security game*. A security game specifies the goal of an adversary

[1]Non-negligible as a function of the security parameter λ used to generate problem instances. More precisely, problem instances are generated by some efficient probabilistic algorithm on input 1^λ and are therefore of size polynomial in λ.

together with the resources it may use to achieve this goal. The security of a scheme is then quantified by the maximum success probability that any adversary can achieve in this game.

For example, in the case of a digital signature scheme, the adversary's objective is to forge a valid signature. Further, for an encryption scheme it is commonly required that an adversary cannot distinguish between the encryptions of two messages of its choice.

Note that, in the case of an encryption scheme, an adversary can always guess which message corresponds to a given ciphertext, yielding a success probability of 1/2. Hence, we cannot hope for a negligible success probability.

For this reason, the figure of merit we are actually interested in is the attacker's *advantage* over some baseline, e.g., random guessing. The advantage takes a value between 0 and 1. It may be a success probability, like the probability of forging a signature, or it may be a distinguishing advantage, like for distinguishing two encryptions. We denote the advantage of an attacker $\mathcal{A}$ breaking security property sec against scheme Π as

$$\mathbf{Adv}^{\mathsf{sec}}_{\Pi}(\mathcal{A}) \ .$$

If the security property sec is clear from context, we may simply write $\mathbf{Adv}_{\Pi}(\mathcal{A})$. For the scheme to be secure, we require the advantage to be small for any (computationally bounded) $\mathcal{A}$. More formally, as a function of the security parameter λ, the advantage is required to be negligible for any $\mathcal{A}$.

Considering the encryption scheme example again, where the attacker $\mathcal{A}$ must distinguish two encryptions of different messages. The advantage then measures how much better the attacker performs than random guessing, and can be computed as

$$\mathbf{Adv}^{\mathsf{sec}}_{\Pi}(\mathcal{A}) := \left| \Pr[\mathcal{A} \text{ guesses correctly}] - \frac{1}{2} \right| \ .$$

For this hypothetical scheme, the desired security requirement is

$$\mathbf{Adv}^{\mathsf{sec}}_{\Pi}(\mathcal{A}) \leq \mathsf{negl}(\lambda)$$

for any $\mathcal{A}$, i.e., the scheme is considered secure when all attackers cannot do significantly better than a random guess.

2.2 Security Reductions

In general, we do not know how to unconditionally prove the hardness of breaking cryptographic schemes. As a consequence, modern cryptography necessarily relies on computational hardness assumptions. However, not all assumptions are equally convincing. Some problems have been studied extensively over many years without finding efficient algorithms, whereas others are new and less well understood. In general, assumptions based on long-standing and well-analyzed problems are considered more reliable.

Accordingly, cryptographic security proofs relate the difficulty of breaking a scheme's security properties to the hardness of solving *well-understood* computational problems. More precisely, by showing that breaking a scheme would imply an efficient solution to a given problem, the security of the scheme is *reduced* to the assumed hardness of that problem.

Concretely, the proof (or reduction) demonstrates that any probabilistic polynomial-time algorithm that breaks Π with non-negligible success probability can be transformed into a probabilistic polynomial-time algorithm that solves the problem X with non-negligible success probability. The construction of a security proof of this form roughly follows the following steps (adapted from [KL07]):

1. Assume there exists an efficient (probabilistic polynomial-time) attacker $\mathcal{A}$ with advantage $\mathbf{Adv}^{\mathsf{sec}}_{\Pi}(\mathcal{A})$ in attacking (the security game of) scheme Π.

2. Construct an efficient algorithm $\mathcal{B}^{\mathcal{A}}$ to solve a computational problem X using $\mathcal{A}$ as a black-box subroutine, i.e., $\mathcal{B}^{\mathcal{A}}$ may interact with $\mathcal{A}$ but it knows nothing about 'how' $\mathcal{A}$ works. When solving problem X, algorithm $\mathcal{B}^{\mathcal{A}}$ may run $\mathcal{A}$ while simulating Π. This simulation is such that:

(a) The view of $\mathcal{A}$ in the simulated experiment is indistinguishable from its view in a real execution of Π.

(b) The algorithm $\mathcal{B}^{\mathcal{A}}$ runs in probabilistic polynomial time, i.e., it is efficient.

(c) There exists a function $f\colon [0,1] \to [0,1]$ such that the advantage of $\mathcal{B}^{\mathcal{A}}$ in solving X satisfies

$$\mathbf{Adv}_X(\mathcal{B}^{\mathcal{A}}) \geq f\big(\mathbf{Adv}^{\mathsf{sec}}_{\Pi}(\mathcal{A})\big)\,,$$

where $f(x)$ is non-negligible whenever x is non-negligible in the security parameter λ.

3. Therefore, if $\mathcal{A}$ runs in probabilistic polynomial time and has non-negligible advantage in breaking Π, then also $\mathcal{B}^{\mathcal{A}}$ runs in probabilistic polynomial time and has non-negligible advantage in solving X.

4. This contradicts the assumed hardness of problem X. We conclude that no efficient attacker can break Π with non-negligible success probability.

Informally, what this concretely implies is that if the proof is correct and we think we found an attack against our scheme, either that attack is wrong, or our hardness assumption is incorrect. Given that we aim to base these hardness assumptions on mathematical problems that have been studied for many decades, the latter would typically require a significant scientific breakthrough.

Assuming that the function f is invertible, the above advantage inequality is typically rewritten as

$$\mathbf{Adv}^{\mathsf{sec}}_{\Pi}(\mathcal{A}) \leq f^{-1}\big(\mathbf{Adv}_X(\mathcal{B}^{\mathcal{A}})\big)\,. \tag{1}$$

This form reflects the goal of security reductions, namely to relate the security of a cryptographic scheme Π to the hardness of a computational problem X. For well-understood problems X, there are often widely believed conjectured upper bounds on the advantage achievable by any efficient algorithm for solving X. The above inequality then translates such conjectured bounds into corresponding bounds on the advantage of any efficient adversary attacking Π.

Later in this work we introduce quantum attackers, who have access to a quantum computer (and play by the laws of quantum mechanics) in addition to the capabilities above. We will denote these attackers by $\mathcal{A}_*$ to emphasize their quantum capabilities.

Remark 1. *Security reductions are not only used to relate cryptographic schemes to their underlying mathematical assumptions, but they also appear between different computational problems and between different schemes. In particular, the security of one scheme can sometimes be reduced to the security of another scheme, or the hardness of one problem can be reduced to the hardness of another problem. Such reductions often form chains, providing a modular and composable approach to analyzing security. For example, the security of the Diffie-Hellman key exchange in a group can be reduced to the hardness of the Group Diffie-Hellman (GDH) problem, which in turn reduces to the Decisional Diffie-Hellman (DDH) problem, itself shown to be equivalent to the Computational Diffie-Hellman (CDH) problem [BCP03].*

2.3 Tightness and the Reduction Loss

So far, we mainly cared about showing that if the underlying computational problem X is *asymptotically* hard then the considered scheme Π is *asymptotically* secure. Thus, from the security reduction, which turns an attacker $\mathcal{A}$ against Π into a algorithm $\mathcal{B}^{\mathcal{A}}$ for solving X, we mainly required that if the running time $\mathit{Time}(\mathcal{A})$ of $\mathcal{A}$ is polynomial and the advantage $\mathbf{Adv}^{\mathsf{sec}}_{\Pi}(\mathcal{A})$ is non-negligible (in the security parameter λ), then the same should hold for $\mathit{Time}(\mathcal{B}^{\mathcal{A}})$ and $\mathbf{Adv}_X(\mathcal{B}^{\mathcal{A}})$.

In practice, however, it is important to consider the *concrete* hardness of breaking a scheme, i.e., to estimate the computational resources required to break a scheme with a specific advantage. This is done by upper bounding $\mathbf{Adv}^{\mathsf{sec}}_{\Pi}(\mathcal{A})$ as a function of $\mathbf{Adv}_X(\mathcal{B}^{\mathcal{A}})$, as in Equation (1), while requiring that $\mathit{Time}(\mathcal{B}^{\mathcal{A}}) \approx \mathit{Time}(\mathcal{A})$, i.e., $\mathcal{B}^{\mathcal{A}}$ has pretty much the same running time as $\mathcal{A}$. More precisely, we write $\mathit{Time}(\mathcal{B}^{\mathcal{A}}) \approx \mathit{Time}(\mathcal{A})$ if the running times of $\mathcal{A}$ and $\mathcal{B}^{\mathcal{A}}$ differ by at most a constant factor. Since an

algorithm's advantage depends inherently on its running time, it is only meaningful to compare advantages of algorithms whose running times are comparable.

The tighter the function relating the two advantages, the tighter the reduction, i.e., less security "gets lost" in the reduction. Concretely, if $\mathbf{Adv}^{\mathsf{sec}}_{\Pi}(\mathcal{A})$ is upper bounded by a linear function in $\mathbf{Adv}_X(\mathcal{B}^{\mathcal{A}})$, e.g., $\mathbf{Adv}^{\mathsf{sec}}_{\Pi}(\mathcal{A}) \leq 4 \cdot \mathbf{Adv}_X(\mathcal{B}^{\mathcal{A}})$, then the reduction is called *tight*. In this case, the resources needed to break the scheme Π very closely match up with the resources necessary to solve the computational problem X, for which, ideally, one has an estimate on the resources necessary. On the other hand, if $\mathbf{Adv}^{\mathsf{sec}}_{\Pi}(\mathcal{A})$ is upper bounded by a function of $\mathbf{Adv}_X(\mathcal{B}^{\mathcal{A}})$ that is super-linear (in the range 0 to 1), for instance $\mathbf{Adv}^{\mathsf{sec}}_{\Pi}(\mathcal{A}) \leq \sqrt{\mathbf{Adv}_X(\mathcal{B}^{\mathcal{A}})}$, then the reduction is called *non-tight* or *loose*. In such a case, while asymptotic security of Π is still ensured, the concrete resources needed to break the scheme may be considerably less than for solving the underlying hard problem X. This leads to weaker concrete security. Tight reductions are preferable because they ensure that the security of the scheme closely reflects the hardness of the underlying problem.

We emphasize that tightness of a reduction should not be confused with the tightness of the inequality that bounds $\mathbf{Adv}^{\mathsf{sec}}_{\Pi}(\mathcal{A})$ in terms of $\mathbf{Adv}_X(\mathcal{B}^{\mathcal{A}})$; it may well be that a reduction is non-tight, i.e., that $\mathbf{Adv}^{\mathsf{sec}}_{\Pi}(\mathcal{A})$ is bounded by a super-linear function in $\mathbf{Adv}_X(\mathcal{B}^{\mathcal{A}})$, but that this inequality is tight in that it cannot be improved.

Sometimes, the bound on $\mathbf{Adv}^{\mathsf{sec}}_{\Pi}(\mathcal{A})$ may involve additional additive terms beyond the term involving $\mathbf{Adv}_X(\mathcal{B}^{\mathcal{A}})$, for instance terms involving parameters of the scheme, and it may not always be clear which of the additive terms is the dominating one.

We repeat that whenever we compare advantages, like $\mathbf{Adv}^{\mathsf{sec}}_{\Pi}(\mathcal{A})$ with $\mathbf{Adv}_X(\mathcal{B}^{\mathcal{A}})$, it is always understood that we consider $\mathcal{A}$ and $\mathcal{B}^{\mathcal{A}}$ with $\mathit{Time}(\mathcal{B}^{\mathcal{A}}) \approx \mathit{Time}(\mathcal{A})$.

2.4 Idealized Models

In cryptographic proofs and reductions, we often make use of idealized models, in which certain components of a cryptographic scheme are treated as ideal primitives. This abstraction enables simpler and more modular reasoning about security. For example, a security proof may assume that a hash function behaves as a truly random function (or is the only way a formal reasoning is possible at all for a given scheme). In practice, however, the idealized component are instantiated by a concrete algorithm, e.g., the random function is replaced by a concrete hash function.

If a scheme that is provably secure in an idealized model is found to be vulnerable when instantiated in practice, this indicates that the concrete instantiation deviates from the ideal behavior. Some cryptographic schemes are currently only known to admit security proofs within specific idealized models. The *standard model* refers to the situation in which no idealized assumptions are made. Below, we describe several common idealized models.

The Random Oracle Model (ROM). In the random oracle model, which was popularized by Bellare and Rogaway [BR93], we treat a cryptographic hash function, that is used in the considered cryptographic scheme, as a truly random function $H : \{0,1\}^* \to \{0,1\}^\lambda$ (referred to as a *random oracle*), which (honest and dishonest) parties need to query in order to compute $H(x)$ for any x. This can then be exploited in security reductions, because now the reduction gets to see the values x for which the attacker "computes" $H(x)$. The expectation is that security of a cryptographic scheme in the ROM implies security when a "good enough" cryptographic hash function is used. However, it is known that *in general* this random oracle methodology is false [CGH98, Bar01, GK03, BBH$^+$19, KRS25], i.e., there exist (mostly artificial) schemes that are proven secure in the ROM but are insecure *for any* concrete hash function.

Despite this negative theoretical result, practical experience since the introduction of the random oracle heuristic has been largely positive. In practice, cryptographic schemes that are proven secure in the random oracle model are widely believed to remain secure when instantiated with standard cryptographic hash functions, provided they are not deliberately engineered to bo insecure when instantiated in practice.

The random permutation model. Similarly to the ROM, in the *random permutation model*, we model a publicly known and efficiently computable permutation that is "random looking" (in an informal way) by a truly random permutation, which needs to be queried in order to evaluate the permutation (or its inverse) on any input. A block cipher can be idealized as a family of random permutations. Therefore, the random permutation model is often used to prove the security of cryptographic constructions based on block ciphers.

The generic group model. The generic group model (GGM) [Sho97b, Mau05] applies to cryptographic schemes that rely on certain groups, such as the multiplicative group of a finite field or the group of points on an elliptic curve. In the GGM, the group is idealized in a way that prevents the attacker from exploiting the concrete representation of group elements.

Formally, each group element is encoded as a uniformly random string, and the attacker must query an oracle access to perform the group operation and its inverse. These oracles ensure consistency with the group law, while hiding all structural information about the underlying representation.

The underlying heuristic is that, for suitably chosen concrete groups, security proven in the generic group model continues to hold when the abstract group operations are instantiated with actual group representations.

2.5 Security Models and Proof Strategies

There are several conventions for writing and structuring the security proof of cryptographic schemes, depending on the setting and what exactly is to be proven. In this work we will mostly make use of *game-based* security proofs. This section is primarily based on [Sho04], to which we also defer the reader for a more extensive tutorial on using game-based proofs.

Game-based security proofs. As set out in the first part of this section, we often argue about the security of a certain cryptographic scheme by relating it to the hardness of a problem known (or assumed to be) practically unbreakable. In proofs we can structure this argument as a series of *games* played by an attacker. Informally we often define Game 0 as the security property we are trying to prove: the attacker has for example won this game if they manage to break the security of the scheme, such as distinguishing between two values that should be indistinguishable from each other, or authenticating themselves as one of the honest participants.

Example (UF-CMA). Let us consider the security of a digital signature scheme. A standard security notion in this setting is *existential unforgeability under chosen-message attack*, or UF-CMA. An adversary $\mathcal{A}$ is said to succeed in a UF-CMA attack if it produces a valid signature on a message of its own choosing. We denote this event by $\mathsf{UF\text{-}CMA}^{\mathcal{A}}$.

A signature scheme $\mathcal{S}$ is called UF-CMA-secure if the probability that any probabilistic polynomial-time adversary succeeds is negligible. Formally, we define the advantage of $\mathcal{A}$ against $\mathcal{S}$ as

$$\mathbf{Adv}^{\mathsf{UF\text{-}CMA}}_{\mathcal{S}}(\mathcal{A}) \;:=\; \Pr\left[\mathsf{UF\text{-}CMA}^{\mathcal{A}}\right],$$

and, for λ denoting the security parameter, require that

$$\mathbf{Adv}^{\mathsf{UF\text{-}CMA}}_{\mathcal{S}}(\mathcal{A}) \leq \mathsf{negl}(\lambda)\,.$$

More precisely, UF-CMA security is defined via a security game between a challenger and the adversary. When we formally define UF-CMA in Section 3.3, we do so using this game-based formulation: the event $\mathsf{UF\text{-}CMA}^{\mathcal{A}}$ is said to occur if the adversary $\mathcal{A}$ *wins* the UF-CMA game as described in Figure 5.

After defining this first Game 0 describing the security of the scheme, a series of other games $1, \ldots n-1$ follows, for each of which the winning probability is related to that of the game before. Finally, the winning probability of Game n is then the desired security level of the scheme — e.g., the scheme is secure if we can prove an attacker can only win the game with negligible advantage (as defined in Section 2.1).

Some common techniques to make a transition from Game i to Game $i+1$ include:

1. proving that the games are statistically indistinguishable;
2. proving that the games are the same unless some failure event F occurs (but the probability of F occurring is negligible);
3. a bridging step that does not change the games a lot, but rewrites some of the computations to prepare for a transition of the first or second kind.

The resulting *series of games* now forms the desired security proof — making it clear how the security claim relates to the hardness assumption the protocol is based on.

The Universal Composability model (UC). The UC-model [Can01] is a different approach to proving the security of schemes, where the focus is on proving that protocols are secure also when they are composed (combined) with other protocols, or with multiple instantiations of themselves. In this model the system that is to be proven is first described as an *ideal functionality* $\mathcal{F}$, where an omnipotent, trusted third party performs all computations. Protocols P are then designed to emulate the behavior of $\mathcal{F}$, that is, a protocol is correct if it can fulfill the same function without actually needing this third party. In UC-terms we say P then *UC-emulates* $\mathcal{F}$. The composability of schemes proven in the UC-model can be an advantage in some settings due to the strong security guarantees it provides and the flexibility of the resulting protocols; a downside is that UC proofs are not suitable for all applications and that some things are considered hard to model in the UC, even though they are easy in other models. Additionally, UC proofs are less common and less standardized than game-based proofs, and — perhaps due to this — considered esoteric and hard to read by some in the community.

3 Technical Preliminary

In this section, we present the formal security definition for public-key encryption, key encapsulation and digital signature schemes.

3.1 Public-Key Encryption

Public-key encryption (PKE) is a technique to encrypt a message using a *public-key*, while decryption is only feasible with the help of the corresponding *secret-key*. Thus, everyone who knows the public key can encrypt, but only the legitimate receiver receiver with the right secret key can decrypt. A crucial feature is that it is sufficient that the public-key is communicated to the sender in an *authenticated* manner (i.e., so that the sender is confident that the given public-key belongs to the intended receiver of the message), it does not have to be communicated in a private way (since knowing the public-key alone does not help to decrypt).

Formally, a PKE scheme $\mathcal{E} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ is specified by three PPT algorithms – the key-generation KGen, the encryption Enc and the decryption Dec. The key-generation KGen takes the security parameter λ as input[2] and it produces a secret-/public-key pair: $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}(1^\lambda)$, and Enc and Dec perform the en- and decryption, using pk and sk, respectively.

Correctness requires that

$$\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)) = m$$

holds (except with negligible probability) for every message m and for $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}(1^\lambda)$, i.e., an honestly generated ciphertext $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$ is (almost) always correctly decrypted.

Regarding security, informally, from a PKE we want that it should be hard to decrypt a ciphertext without knowing the secret key. In its simplest form, this is captured by requiring that for a randomly chosen message m^*, no (computationally bounded) attacker can recover m^* from its encryption $c^* = \mathsf{Enc}(\mathsf{pk}, m^*)$ when only given pk but not sk. This is called *one-wayness*, or $\mathsf{OW\text{-}CPA}$ security.

[2]For technical reasons, λ is is given in unary notation, i.e., as $1^\lambda = 11\cdots1$, as input.

OW-CPA security is often not sufficient (because often in applications the message m is not random). A stronger notion is *indistinguishability* (a.k.a. *semantic security*, or more specifically IND-CPA), which essentially requires that for *any* two messages m_0^*, m_1^* it should be hard to distinguish $\mathsf{Enc}(\mathsf{pk}, m_0^*)$ from $\mathsf{Enc}(\mathsf{pk}, m_1^*)$ for any (computationally bounded) attacker.

History has shown that the aforementioned notion of semantic security is still insufficient for practical applications. In 1998, Daniel Bleichenbacher proposed an attack [Ble98] that breaks practical use cases of RSA PKCS #1, a cryptographic standard that was widely deployed at the time, also in the Secure Socket Layer (SSL) protocol. Bleichenbacher's attack does not violate the IND-CPA security of the considered scheme. Instead, it crucially exploits that, in "real life situations," an attacker might be able to fool the owner of the secret key to decrypt some "harmless" ciphertexts. This has motivated a stronger sense of security, where indistinguishability holds even when the attacker gets hold of a decryption oracle that decrypts any ciphertext, except the one the attacker wants to break. This is formally captured by the following security definition, referred to IND-CCA security as defined below.[3]

Definition 1. *We say that a PKE scheme $\mathcal{E} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ is* IND-CCA*, if for every PPT oracle algorithm $\mathcal{A}$, which we refer to as an* IND-CCA *attacker, the quantity*

$$\mathbf{Adv}_{\mathcal{E}}^{\mathsf{IND\text{-}CCA}}(\mathcal{A}) := \left| \Pr\left[\mathsf{IND\text{-}CCA}^{\mathcal{A}}(0)\right] - \Pr\left[\mathsf{IND\text{-}CCA}^{\mathcal{A}}(1)\right] \right| \leq \mathsf{negl}(\lambda)$$

is negligibly small, where the security game $\mathsf{IND\text{-}CCA}^{\mathcal{A}}$ is formally specified in Figure 1.
The weaker variant where $\mathcal{A}$ has no access to the decryption oracle $\mathsf{Dec}(\mathsf{sk}, \cdot)$ is called IND *security.*

$\mathsf{IND\text{-}CCA}^{\mathcal{A}}(b)$
1: $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}$
2: $(m_0^*, m_1^*, \mathsf{st}) \leftarrow \mathcal{A}^{\mathsf{Dec}(\mathsf{sk}, \cdot)}(\mathsf{pk})$
3: $c^* \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b^*)$
4: **return** $\left[\mathcal{A}^{\mathsf{Dec}^{\neq c^*}(\mathsf{sk}, \cdot)}(c^*, \mathsf{st}) = 1\right]$

$\mathsf{Dec}^{\neq c^*}(\mathsf{sk}, c)$
1: **if** $c = c^*$ **return** $\perp$
2: **return** $\mathsf{Dec}(\mathsf{sk}, c)$

Figure 1: The IND-CCA game for PKE.

In the formal IND-CCA game in Figure 1, the attacker $\mathcal{A}$ gets the public-key pk as input, and it is given access to a decryption oracle that decrypts any message that $\mathcal{A}$ asks to be decrypted. In a first step, $\mathcal{A}$ then outputs two messages m_0^* and m_1^*, of which one is encrypted, which the ciphertext c^* given to $\mathcal{A}$, who then has to find out which of the two was encrypted. For that step, $\mathcal{A}$ again has access to the decryption oracle, but it is not allowed to ask c^* to be decrypted.

Take for (non)example the plain Elgamal encryption scheme. The public-secret key pair $(\mathsf{sk}, \mathsf{pk})$ is of the form $\mathsf{sk} = a$ and $\mathsf{pk} = g^a$, where $(a, g) \in \mathbb{Z}_{>0} \times G$ for a cryptographic group G that is generated by g. To encrypt a message (which we treat as a group element) $m \in G$, the ciphertext is produced via

$$(c_0, c_1) := (g^r, m \cdot \mathsf{pk}^r) \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$$

for a randomly chosen number r from a sufficiently large interval, and to decrypt the ciphertext $(c_0, c_1) \in G^2$, simply restore the message via computing

$$c_1 / c_0^{\mathsf{sk}} \leftarrow \mathsf{Dec}(\mathsf{sk}, (c_0, c_1)).$$

Under the decisional Diffie-Hellman assumption, this scheme satisfies IND-CPA. However, there is a simple attack breaking the IND-CCA security. Indeed, given the encryption $c^* := (c_0, c_1) \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$ of a message m, the attacker can query $\mathsf{Dec}^{\neq c^*}$ in order to compute $m' := \mathsf{Dec}^{\neq c^*}\left(\mathsf{sk}, c_0^{-1}, c_1^{-1}\right) \cdot \mathsf{Dec}^{\neq c^*}\left(\mathsf{sk}, c_0^2, c_1^2\right)$, which coincides with the message m with certainty.

[3]The mnemonics is as follows. Encryptions of any two (adversarially chosen) messages should be INDistinguishable, even when facing a Chosen-Ciphertext Attack.

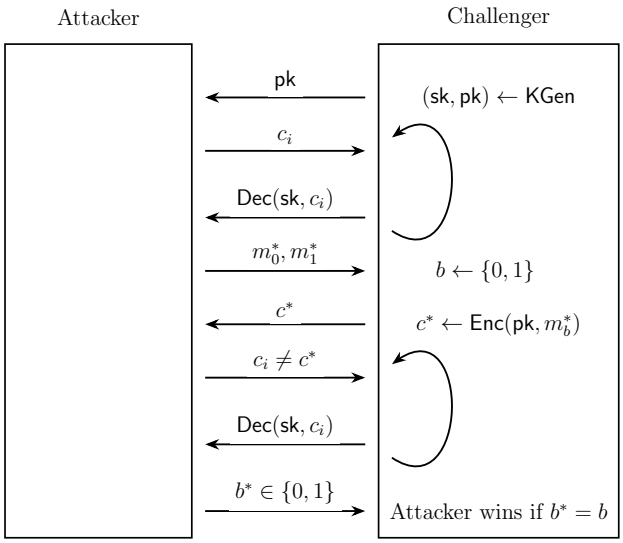


Figure 2: The IND-CCA game for PKE, illustrated.

3.2 Key Encapsulation

Given that a PKE is often used to securely communicate a random symmetric-encryption key, a slightly weaker functionality is sufficient then, where the to-be-encrypted message is not given as input to the encryption algorithm, but produced by the algorithm. This is then referred to as a key encapsulation mechanism (KEM).

Formally, a KEM $\mathcal{E} = (\mathsf{KGen}, \mathsf{Encaps}, \mathsf{Decaps})$ consists of three PPT algorithms – the key-generation KGen, the (session-key) encapsulation Encaps, and the decapsulation Decaps. Here, correctness means that $\mathsf{Decaps}(\mathsf{sk}, c) = k$ almost with certainty for $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}(1^\lambda)$ and $(k, c) \leftarrow \mathsf{Encaps}(\mathsf{pk})$. Security requires that the encapsulated key cannot be distinguished from a fresh random key (even when given access to a decapsulation oracle).

Definition 2. *We say that a KEM* $\mathcal{E} = (\mathsf{KGen}, \mathsf{Encaps}, \mathsf{Decaps})$ *is* IND-CCA*, if for every PPT oracle algorithm* $\mathcal{A}$*, which we call an* IND-CCA *attacker, the quantity*

$$\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathcal{E}}(\mathcal{A}) := \left| \Pr\left[\mathsf{IND\text{-}CCA}^{\mathcal{A}}(0)\right] - \Pr\left[\mathsf{IND\text{-}CCA}^{\mathcal{A}}(1)\right] \right| \leq \mathsf{negl}(\lambda)$$

is negligibly small, where $\mathcal{K}$ *denotes the set of all possible session keys, and the security game* $\mathsf{IND\text{-}CCA}^{\mathcal{A}}$ *is formally specified in Figure 3.*

$\mathsf{IND\text{-}CCA}^{\mathcal{A}}(b)$
1: $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}$
2: $(k_0, c^*) \leftarrow \mathsf{Encaps}(\mathsf{pk})$
3: $k_1 \leftarrow \mathcal{K}$
4: **return** $\left[\mathcal{A}^{\mathsf{Decaps}^{\neq c^*}(\mathsf{sk}, \cdot)}(k_b, c^*) = 1\right]$

$\mathsf{Decaps}^{\neq c^*}(\mathsf{sk}, c)$:
1: **if** $c = c^*$ **return** $\perp$
2: **return** $\mathsf{Decaps}(\mathsf{sk}, c)$

Figure 3: The IND-CCA game for KEM.

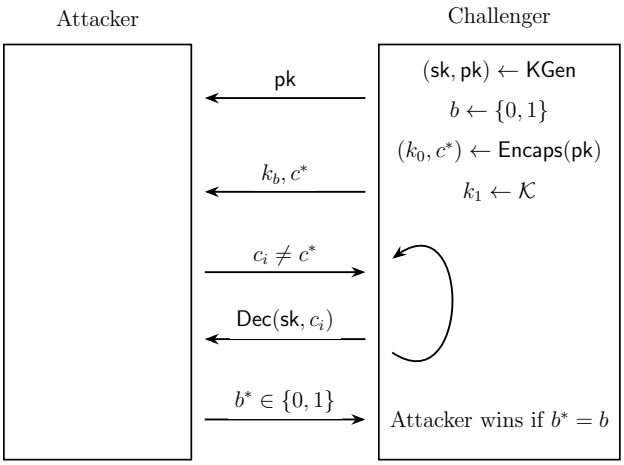


Figure 4: The IND-CCA game for KEM, illustrated.

3.3 Digital Signatures

A signature scheme $\mathcal{S} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ consists of three PPT algorithms – the key generation KGen, the signing Sign and the verification Vrfy. Correctness requires that an honestly produced signature is accepted, i.e.,

$$\mathsf{Vrfy}(\mathsf{pk}, \mathsf{Sign}(\mathsf{sk}, m)) = 1$$

almost with certainty for every message m and for $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}(1^\lambda)$. Security requires that, without knowing the secret key, it is infeasible to find a valid signature for a message of the attacker's choice, even when given access to a signature oracle that signs any message except the one chosen by the attacker.

Definition 3. *We say that a signature scheme $\mathcal{S} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ is $\mathsf{UF\text{-}CMA}$, if for every PPT oracle algorithm $\mathcal{A}$, which we refer to as an $\mathsf{UF\text{-}CMA}$ attacker, the probability*

$$\mathbf{Adv}^{\mathsf{UF\text{-}CMA}}_{\mathcal{S}}(\mathcal{A}) := \Pr\left[\mathsf{UF\text{-}CMA}^{\mathcal{A}}\right] \leq \mathsf{negl}(\lambda)$$

is negligibly small, where the security game $\mathsf{UF\text{-}CMA}^{\mathcal{A}}$ is formally specified as in Figure 5.
The weaker variant where $\mathcal{A}$ has no access to the signing oracle $\mathsf{Sign}(\mathsf{sk}, \cdot)$ is called UF (or $\mathsf{UF\text{-}NMA}$) security.

$\mathsf{UF\text{-}CMA}^{\mathcal{A}}$
1: $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}$
2: $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sign}(\mathsf{sk}, \cdot)}(\mathsf{pk})$
3: **return** $\mathsf{Vrfy}(\mathsf{pk}, m^*, \sigma^*) \ \wedge \ \forall i: \ m^* \neq m_i$
4: $\{m_i$ is the ith query made by $\mathcal{A}$ to $\mathsf{Sign}(\mathsf{sk}, \cdot)\}$

Figure 5: The $\mathsf{UF\text{-}CMA}$ game.

4 Quantum Attacks and Provable Security

The (so far hypothetical) possibility of doing quantum computations affects the above described provable-security paradigm in the following two ways.

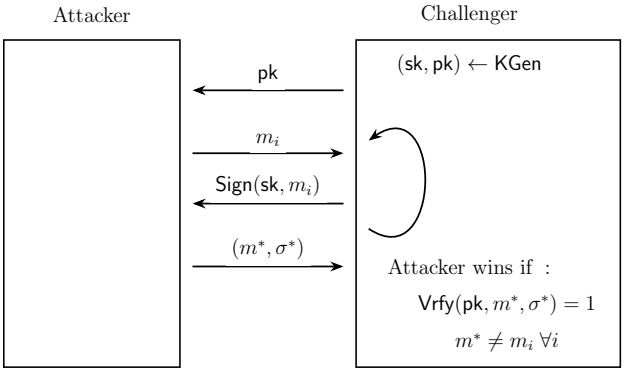


Figure 6: The UF-CMA game, illustrated.

4.1 The Computational Assumption

The classical choices for the underlying computational problem X (integer-factorization, and the problem of computing discrete logarithms in certain groups like elliptic curves) are known to *not* be hard to solve by means of a quantum computer. Indeed, these computational problems can be efficiently solved by Shor's quantum algorithm [Sho94, Sho97a]. This not only makes the statement *if X is hard to solve then* Π *is secure* void, all these schemes can actually be easily broken with the help of a quantum computer, by means of Shor's quantum algorithm.

This issue is countered by replacing these computational problems by new ones that are (believed to be) hard to solve with a quantum computer. The community has identified a few promising candidates (in different variations) for such new computational problems that are, on the one hand, believed to be quantum-hard and, on the other hand, have enough structure in order to admit the design of cryptographic schemes. Notable examples are lattice problems, decoding of random codes, solving multi-variate nonlinear systems of equations, and computing elliptic-curve isogenies.

These computational problems tend to be less clear-cut than, say, factoring and computing discrete logarithms: the specific problem statement X involves various options and parameters that one can set. For instance, in the context of lattices one can consider arbitrary lattices or lattices with additional structure (e.g., where the lattice forms an ideal in a polynomial quotient ring), and one can vary the dimension of the lattice as well as the field size, and more. Such choices affect (i) the flexibility of using X in order to design a cryptographic scheme Π, (ii) the efficiently of executing Π, and (iii) the computational hardness of X. Not so surprising, (i) and (ii) compete against (iii): setting the options and parameters so that X admits a richer class of constructions and/or more efficient schemes, typically decreases the computational hardness of X. Thus, much of the current research into this direction is towards scrutinizing the computational hardness of the different flavors of problem X obtained by considering different choices of options and parameters, finding better contractions for the scheme Π so that a more conservative choice of the parameters of X have less negative impact on, say, the efficiency of Π, and eventually to get the right balance so that X is sufficiently hard while Π is as efficient as possible.

4.2 The Security Reduction

A more subtle aspect is the following. Even when X is replaced by a computational problem that is (believed) to be hard in the light of a quantum computer, the original proof of the statement *if the problem X is hard to solve then the scheme* Π *is hard to break* may not be sound anymore when switching to quantum algorithms as the model of computation that defines what it means for something to be (computationally) "hard". There are cases where the proof is not affected and remains valid, but there are multiple reasons for why a security

proof may — and often does — become invalid, which we briefly discuss below. In the latter case, one often runs into deep technical questions about the behavior of quantum information when trying to salvage the proof. Also, even if the security statement can be salvaged via a new, quantum-sound proof, the reductions loss is typically worse.

4.2.1 Rewinding

One possible reason is that in the security reduction, which turns any hypothetical successful attacker $\mathcal{A}_*$ attacking Π into an efficient algorithm $\mathcal{A}'_*$ for solving X, the attacker $\mathcal{A}_*$ may be *reset and restarted*, either from the beginning of the attack or from some point in time during the attack (this is referred to as *rewinding*). However, by the laws of quantum mechanics, this is not possible anymore — at least not immediately — when $\mathcal{A}_*$ is a quantum algorithm. Essentially, this is due to the no-cloning theorem, which implies that the internal state of $\mathcal{A}_*$ cannot be copied, and the irreversibility of quantum measurements, which means that in general one cannot undo operations done by $\mathcal{A}_*$. This is called the *rewinding barrier*.

4.2.2 Superposition Access to the Random Oracle

In Section 2.4 we introduced the Random Oracle Model (ROM), and how it treats cryptographic hash functions as an idealized *oracle* that the attacker needs to query in order to "compute" the hash value of any input.

In the security reduction, which turns a successful attacker $\mathcal{A}_*$ attacking Π into an efficient algorithm $\mathcal{A}'_*$ for solving X, it is then typically so that inspecting the queries that $\mathcal{A}_*$ makes for "computing" hash values provides valuable information that makes the reduction, i.e., the construction of $\mathcal{A}'_*$, go through. However, when considering a quantum attacker, which, by the laws of quantum mechanics, can compute anything in quantum superposition, his queries to the oracle may then be "in superposition" as well; one the speaks of the *quantum* random oracle model (QROM). Formally, this means that instead of oracle access to the (uniformly random) function H, $\mathcal{A}_*$ is given oracle access to the quantum operation (formally, the unitary map) $U_H : |x\rangle\,|y\rangle \mapsto |x\rangle\,|y + H(x)\rangle$. But then, again by the laws of quantum mechanics, these queries cannot be inspected and recorded anymore without disturbing the superposition. This is sometimes referred to as the *recording barrier*. Similar issues occur in other idealized models, like the *ideal-cipher model* or the *random-permutation model*.

4.2.3 New Security Definitions

At times, the formal *definition* of what it means to "break" a cryptographic scheme Π needs to be revised in the context of a quantum attacker. This would obviously also affects the *proof* of a scheme being secure.

A notable example is the definition of the collision resistance of a hash function H, which classically demands that it is computationally hard to find x and y with $H(x) = H(y)$. In the classical (i.e., non-quantum) setting, this captures well that a hash value h acts as a fingerprint of a *unique* digital document x. Indeed, consider an attacker $\mathcal{A}_*$ that produces a hash value h (in a possibly malicious way, i.e., not necessary by hash a document) along with some state information st (kept by $\mathcal{A}_*$). Then, there is a *single* document x that can be revealed by $\mathcal{A}_*$ to match h; namely, if it was possible to find (by means of two respective (classical) computations with input st) *one or another* document, say x or y, that both hash to the same hash value h, i.e., $H(x) = h$ and $H(y) = h$, then, by means of doing both computations, one can find *both* documents x and y, contradicting the hardness of finding x and y with $H(x) = H(y)$.

However, somewhat surprisingly at first glance, this reasoning fails to hold when we allow quantum computations. The reason is that then, being able to compute *one or the other* document by means of running one or the other quantum computation (on input st) does *not* imply anymore being able to compute *both* (and so there is no contradiction anymore). Namely, the state information st may be in the form of a quantum state, and so, by no-cloning, the state st cannot be copied and hence is not available as input to both computations. Thus, either computation is possible on input st, but not both.

A similar issue occurs in the context of the binding property of commitment schemes, and, less obviously, in the context of so-called Σ-protocols, both being sophisticated cryptographic primitives that play a crucial role in modern cryptography.

4.3 Q1 and Q2 Security Models

Section 3 provided formal security definitions for public-key encryption, key encapsulation, and digital signatures. The security of these primitives is defined in terms of security games, as described in Section 2.5, and these definitions are largely agnostic to whether the attacker is classical or quantum (in contrast to the cases mentioned in Section 4.2.3).

Given that in these security games, the attacker is allowed to query a signing or a decryption oracle on messages/ciphertexts of his choice, similar to ROM versus QROM case one might be tempted to allow a quantum attacker to query the given oracle *in superposition*. And, as a matter of fact, there is a line of research that considers this *Q2 security model*, which indeed gives rise to challenging research problems.

However, from a practical standpoint, the Q2 model is clearly unrealistic. In an actual attack, in order to get, say, a signature on a message under the public key it wants to attack, the attacker needs to interact with a classical device (which has the secret key stored), and it is unrealistic to assume that this device behaves quantumly.[4] This is in contrast to the ROM case, where the random oracle models a *publicly-known* hash function that the attacker can locally compute, and so a quantum attacker can perform the computation on a quantum computer, giving him the ability to compute the hash function in superposition. So, giving superposition access to the random oracle is well justified.

For this reason, in the standard security definition the attacker, even when quantum, can query the signing (or decryption or decapsulation) oracle only classically. Sometimes, in order to make the distinction, one then speaks of the *Q1 security model*.

5 Proof Techniques in the (Q)ROM

We show here some typical technical barriers that one encounters when trying to lift classical security proofs in the random oracle model to the setting of a quantum attacker (i.e., to the quantum random oracle model), and we show how these barriers have been removed (to some extent) by sophisticated techniques. The list presented here is by no means conclusive; there are barriers and solutions not discussed here, and there are still barriers that have no (satisfying) solution yet.

5.1 Lazy Sampling and the Compressed Oracle Technique

The Classical Case. Recall the random oracle model, discussed in Section 2.4. Here a cryptographic hash function with domain $\mathcal{X}$ and codomain $\mathcal{Y}$ is modeled as a uniformly random function $H : \mathcal{X} \to \mathcal{Y}$, to which the attacker $\mathcal{A}$ (and any other algorithm) have query access, only. We may write $\mathcal{A}^H$ to emphasize $\mathcal{A}$'s access to H. In a security reduction, which runs a hypothetical successful attacker $\mathcal{A}$ internally, it is then up to the reduction to answer the queries of $\mathcal{A}$ to H. Since the reduction needs to be efficient (and so cannot sample a uniformly random function H), this is typically done by *lazy sampling*: Whenever $\mathcal{A}$ queries an input $x \in \mathcal{X}$ that it has not queried yet, the reduction samples a uniformly random $y \leftarrow \mathcal{Y}$ and answers the query with $H(x) := y$; when $\mathcal{A}$ queries an input $x \in \mathcal{X}$ that it has already queried, the reduction answers consistently with the earlier defined value for $H(x)$.

The fact that in the random oracle model the reduction controls the interaction between the attacker $\mathcal{A}$ and the (simulated) random oracle gives the reduction powerful leverage. For instance, the reduction now "knows" for which inputs x the attacker has learned the hash value $H(x)$ and for which ones not.[5] More

[4] Sometimes it is argued that if the device is a smart card in the hands of the attacker, and the attacker cools down the smart card to close to absolute zero, then one cannot exclude that some quantum effects appear — but no results indicating the possibility of signing messages in superposition exist.

[5] This is in contrast to the standard model, where $\mathcal{A}$ can compute the considered hash function (say SHA3) internally, and possibly in an obfuscated way, and so the reduction then has not clue about which hash values $\mathcal{A}$ knows.

precisely, in the analysis of the reduction one can make statements depending on the queries made by $\mathcal{A}$. E.g., one can exploit that if a certain x has not been queried (yet) then the hash $H(x)$ of x is like a fresh uniformly random value to $\mathcal{A}$.

More formally, when simulating the random oracle using lazy sampling, the reduction builds up a *partial* function $H : \mathcal{X} \to \mathcal{Y} \cup \{\perp\}$, with $H(x) = \perp$ meaning that x has not yet been queried and so the value of $H(x)$ has not been assigned yet in the laze-sampling procedure, and therefore the "true hash" of x is a freshly random value in $\mathcal{Y}$. By construction, at any point in time if no more that q queries were made to the random oracle so far, then the support $\mathsf{supp}(H) := \{x \in \mathcal{X} \mid H(x) \neq \perp\}$ is of size at most q.

As a toy example, we prove that finding a preimage under H is hard in the random oracle model, i.e., when $H : \mathcal{X} \to \mathcal{Y}$ is modeled as a random oracle.[6] For that purpose, consider an attacker/algorithm $\mathcal{A}$, whose goal is to find $x \in \mathcal{X}$ such that $H(x) = y$ for a given choice of $y \in \mathcal{Y}$. Let's say that $\mathcal{A}$ makes q queries to the random oracle. Answering the queries via lazy sampling, for every such query x_i by $\mathcal{A}$ the hash $H(x_i)$ is determined by choosing a fresh random value (unless $\mathcal{A}$ has already queried x_i, then we answer consistently). Hence $\Pr[H(x_i) = y] = 1/|\mathcal{Y}|$ for every single query x_i, and so, by union bound, $H(x_i) \neq y$ for all queries except with probability $q/|\mathcal{Y}|$. However, in case of this (very likely) event, the best $\mathcal{A}$ can do is output an x that is different to all the queries x_i, and hope that $H(x) = y$; this though again happens with probability $1/|\mathcal{Y}|$ only. Thus, overall, the success probability of $\mathcal{A}$ is no more than $(q+1)/|\mathcal{Y}|$.

Via a similar line of reasoning one can show that it is hard to find a collision in H. As long as there is no collision in the hash values that $\mathcal{A}$ has queried, it is almost impossible for $\mathcal{A}$ to output a collision. However, here, since the potential number of collisions among the queries grows quadratically in q, i.e., there are $\frac{1}{2}q(q-1)$ candidate pairs (x_i, x_j), the overall success probability is in the order of $q^2/|\mathcal{Y}|$, which is in line with the famous "birthday bound".

The Post-Quantum Case. Switching to the quantum setting, where $\mathcal{A}_*$ is a quantum algorithm that may query the random oracle in superposition, i.e., has access to the unitary $U_H : |x\rangle|y\rangle \mapsto |x\rangle|y + H(x)\rangle$, the above does not work anymore — at least not in an obvious way. To start with, it is not even clear that the reduction can efficiently simulate $\mathcal{A}_*$'s access to the unitary U_H. For instance, by querying the random oracle on the (to be normalized) state $\sum_x |x\rangle|0\rangle$, the attacker can obtain a superposition $\sum_x |x\rangle|H(x)\rangle$ of all hash-input-output pairs. Per-se, it is not clear how to provide such a superposition without first sampling (the uniformly random function) H, which is not feasible (unless $\mathcal{X}$ and $\mathcal{Y}$ are very small).

The so-called compressed-oracle technique, introduced by Zhandry [Zha19] provides the means to salvage some of the features of the above classical lazy-sampling idea. Explaining the technique in detail is beyond the scope of this report (it involves purifying the random choice of H and switching back-and-forth between the computational and the Fourier basis); we mainly sketch some of its features — and the caveat.

Similarly to the lazy-sampling case, where a partial function H is maintained with support size that is bounded by the number of queries made so far, the simulation of the quantum random oracle using the compressed-oracle technique works by maintaining a quantum state that is a *superposition* $|\Delta\rangle = \sum_H \alpha_H |H\rangle$ of partial functions H with $|\mathsf{supp}(H)| \leq$ the number of queries made so far.[7] Furthermore, for any $x \in \mathcal{X}$, the reduction can perform a measurement in the Fourier basis on $|\Delta\rangle$ to check whether "$H(x) \stackrel{?}{=} \perp$". If the measurement outcome is "yes, $H(x) = \perp$" then this is to be interpreted as that $\mathcal{A}_*$ has not queried x yet and so the hash of x remains random and unknown to $\mathcal{A}_*$. Indeed, the actual hash of x can be obtained by means of a measurement in the computational basis, and given the above outcome to the former measurement, the part of the state to be measured here has collapsed to a uniform superposition of all possible $y \in \mathcal{Y}$, and so the measurement will be a fresh random $y \in \mathcal{Y}$.

Thus, in the above sense, the compressed oracle technique seems to offer the same functionality as lazy sampling. One big difference however is that the above quantum measurements *disturb* the state $|\Delta\rangle$. This

[6] This proof is somewhat atypical since we are proving something about the hash function H, while normally one would prove something about a cryptographic scheme that uses H as a subroutine. For this reason, the proof is not a reduction to a computationally hard problem, but it is an unconditional proof (in the ROM). Still, it nicely demonstrates a typical way of how the random oracle model is exploited.

[7] Technically, this is not fully correct. The state $|\Delta\rangle$ will be entangled with the internal state of $\mathcal{A}_*$ and so is a *mixed* state (and not a *pure* state), *supported* by such H's with bounded support size.

disturbance may then also influence $\mathcal{A}_*$; in the worst case $\mathcal{A}_*$ aborts and so becomess useless to the reduction.

For example, we cannot exclude that $\mathcal{A}_*$ makes a certain superposition query and then "uncomputes" the query again by means of another query. However, if in-between these two queries $|\Delta\rangle$ is measured, then the uncomputing will uncompute properly, due to the disturbance caused by the measurement. This can be detected by $\mathcal{A}_*$, and we cannot exclude that $\mathcal{A}_*$ then aborts.

Therefore, one needs to carefully analyze these measurements in the considered context and argue that they remain unnoticed to $\mathcal{A}_*$, or more generally control the impact they have. This typically also comes with a quantitative effect, in making the reduction less tight than in the classical case, where the information on which hashes $\mathcal{A}_*$ knows comes "for free".

In [CFHL21] a framework is presented that abstracts away the quantum-technical aspects of the compressed oracle technique and offers the means to lift certain proofs in the random oracle model to the quantum setting. Applied to the two toy examples discussed above, one can then derive in a rather straightforward way the quantum counter parts for the bounds for preimage finding and collision finding, which are of order $q^2/|\mathcal{Y}|$ and $q^3/|\mathcal{Y}|$, respectively, in line with the quadratic Grover speedup and the known quantum improvement for collision finding. However, most of the time, classical proofs cannot be lifted in a straightforward manner.

5.2 Adaptive Programming

The Classical Case. Another useful feature of the reduction simulating the random oracle is that it can select the hash values. When doing lazy sampling, by default all the hash values (that need to be provided) are sampled uniformly random from the codomain $\mathcal{Y}$; however, in certain cases, it is convenient to *redefine* during the course of the reduction the hash $H(x)$ for a particular x to be some strategically chosen value y.

This often comes in handy when the reduction needs to simulate signing queries, when facing a UF-CMA attacker attacking a signature scheme. Instead of computing the signature honestly (which would require knowledge of the secret key, which the reduction often does not have), the reduction can produce the signature by "cheating" with the random oracle, i.e., by reprogramming H in a suitable way. For example, if the signature for a message m is given by a pair (r, s) with r randomly chosen "salt" and

$$s = f^{-1}(H(m\|r))\,,$$

where, say, f is an trapdoor one-way permutation (i.e., inverting it requires knowledge of the secret key), the reduction can simulate such a signature by choosing a random r and s and redefining $H(m\|r)$ to be

$$H(m\|r) := f(s)\,,$$

indeed, it then holds that $s = f^{-1}(H(m\|r))$. By being able to simulate signing queries (in this way), we can reduce UF-CMA to UF-NMA security, which is typically significantly easier to prove.

As long as the newly assigned value y for $H(x)$ is again uniformly random (which for instance is the case in the above example), $\mathcal{A}$ can only notice this *reprogramming* of H if it has queried H on x before the reprogramming; indeed, in this case it will notice the inconsistency when querying H on x again (and may then abort and so become useless for the reduction). Thus, as long as one can argue that it is unlikely that $\mathcal{A}$ has queried x yet, the reprogramming of H remains unnoticed by $\mathcal{A}$. This holds trivially if the (re)programming takes place at the very beginning of the execution of $\mathcal{A}$ (and so $\mathcal{A}$ has made no queries yet); the more interesting case though is when x is adaptively chosen *during* the course of the execution, which is why one speaks of *adaptive* reprogramming in general. In this case, we need that x is hard to guess (for $\mathcal{A}$); indeed, if x is so that $\Pr[x = x^\circ] \leq \epsilon$ for all x° (think of $\epsilon \approx 0$), then by a simply union-bound argument, after q queries $\mathcal{A}$ has queried x with probability at most $q\epsilon$. That is why in the above example $m\|r$ is hashed, instead of only m; this makes $x = m\|r$ hard to guess. Again, the reasoning exploits that we can make (probabilistic) statements about the queries made by $\mathcal{A}$, like that whether $\mathcal{A}$ has made a query to x.

The Post-Quantum Case. Again, the above reasoning does not easily carry over. Indeed, it is not even well defined to say that $\mathcal{A}_*$ has not made a query to a particular x, when $\mathcal{A}_*$ makes queries in superposition.

Again, some non-trivial technique comes to the rescue, showing that conceptually the same holds, but with a worse loss. This comes in the form of what is called the *adaptive reprogramming lemma*, which ensures that if x is hard to guess then the reprogramming still remains unnoticed to $\mathcal{A}_*$, but with a square-root loss in the quantitative statement.

Lemma 1 (A paraphrase of [GHHM21, Theorem 1])**.** *Let $\epsilon > 0$, and let $\{\mathcal{D}_i\}_{i\in\mathcal{I}}$ be a family of distributions over a finite set $\mathcal{X}$, with the property that $\Pr[x = x^\circ] \le \epsilon$ for all $i \in \mathcal{I}$ and $x^\circ \in \mathcal{X}$, with x sampled from $\mathcal{D}_i$. Let $\mathcal{A}_*$ be an oracle algorithm that makes at most q quantum queries to the random oracle H, and one classical query to a yet-to-be-specified oracle. Then, for O_0 and O_1 as specified in Figure 7, we have*

$$\mathsf{SD}\left(\mathcal{A}_*^{H,O_0}, \mathcal{A}_*^{H,O_1}\right) \le \sqrt{q\epsilon} + q\epsilon/2 \ ,$$

where SD *denotes the statistical distance between the outputs of the two algorithms.*

$O_0(i)$:	$O_1(i)$:
1: $x \leftarrow \mathcal{D}_i$	1: $x \leftarrow \mathcal{D}_i$
2: $y := H(x)$	2: $H(x) := y \leftarrow \mathcal{Y}$
3: **return** (x, y)	3: **return** (x, y)

Figure 7: Reprogramming oracles O_0 and O_1.

We sketch the proof, which uses the compressed oracle technique discussed above.

Proof sketch. We consider a run of $\mathcal{A}_*$ up to the point where it makes a query to O_0 or O_1, respectively, where we simulate the random oracle by using the compressed oracle technique. Then, for the purpose of the argument, we measure the state $|\Delta\rangle$ of the compressed oracle to check whether "$H(x) \stackrel{?}{=} \perp$", which x being the value that is sampled in the execution of O_0 or O_1. By the promise on the state $|\Delta\rangle$ of being a superposition of (actually, being supported by) $|H\rangle$'s with $|\,\mathsf{supp}(H)\,| \le q$, and by the high-entropy condition on $\mathcal{D}_i$, the probability that the measurement outcome is negative is at most $q\epsilon$.[8] It then follows from the Gentle-Measurement Lemma[9] that the collapsed state after the measurement is $(\sqrt{q\epsilon} + q\epsilon/2)$-close to the original, unmeasured state. Furthermore, given an affirmative measurement outcome, the actual hash value of x is obtained by measuring a uniform superposition of all values in the codomain, and thus there is no difference at all between O_0 (where this superposition is measured to determine $H(x)$) or O_1 (where $H(x)$ is reprogrammed to a freshly chosen uniformly random value). □

5.3 Measure and Reprogram

Here, we consider the following abstract problem. Let $\mathcal{A}$ be an attacker with access to the random oracle H, aiming to output a pair (x, z) such that the triple $(x, H(x), z)$ satisfies a given predicate $V(x, H(x), z)$. The reader familiar with the Fiat–Shamir transformation (see Section 6.2) should recognize the setting. Our goal is to relate the probability of $\mathcal{A}$ succeeding to produce such a pair with the probability of winning the following *interactive* game: produce x (in an arbitrary way), receive a random challenge Θ, and then produce z such that $V(x, \Theta, z)$ is satisfied.

It is clear that if the interactive game is easy to win then it is easy for $\mathcal{A}$ to produce (x, z) as required: produce x, get $\Theta = H(x)$ by querying the random oracle, and the produce z as in the interactive game. The interesting question is the other direction: if the interactive game is hard to win, is the challenge for $\mathcal{A}$ hard as well? Clearly, if there is a strategy for the interactive game that succeeds with probability ε (and for

[8] In more detail, for the purpose of the argument, we can measure "$H(x') \stackrel{?}{=} \perp$" for *all* x', and for all but q choices of x' the outcome will be affirmative (due to the support condition), confirming that $H(x') = \perp$, and so it will also hold for x except with probability $q\epsilon$.

[9] The Gentle-Measurement Lemma states, in a quantitative way, that if the outcome of a (projective) measurement is almost certain then the measurement disturbs the state only by little.

which x has high entropy), but through repeated trying we get an $\mathcal{A}$ that succeeds with probability $q\varepsilon$ when making q queries to the random oracle — but is this the best $\mathcal{A}$ can do?

More formally, one would like to show that for every $\mathcal{A}$ there exists a strategy $\mathcal{S}$ for the interactive game such that

$$\Pr\big[V(x,\Theta,z) \,:\, (x,z) \leftarrow \langle \mathcal{S},\Theta\rangle\big] \gtrsim \frac{1}{q}\Pr\big[V(x,H(x),z) \,:\, (x,z)\leftarrow \mathcal{A}^H\big]\,,$$

where $\langle \mathcal{S},\Theta\rangle$ is to be understood as a run of the interactive game with random challenge Θ, and q is an arbitrary but fixed upper bound on the number of queries that $\mathcal{A}$ makes to the random oracle.

The Classical Case. In case of a classical $\mathcal{A}$, i.e., when $\mathcal{A}$ makes classical queries (let us denote them by $x_1,\ldots,x_q$ below) to the random oracle, the reduction is rather straightforward. Indeed, $\mathcal{S}$ then works as follows. It chooses a random index $i\in\{1,\ldots,q+1\}$ and runs $\mathcal{A}$ (while answering $\mathcal{A}$'s random oracle queries using lazy sampling) until $\mathcal{A}$ makes the i-th query to the random oracle (where $i=q+1$ then means that $\mathcal{S}$ runs $\mathcal{A}$ until the end). If $i=q+1$ then $\mathcal{S}$ simply collects $\mathcal{A}$'s output (x,z) and first outputs x, and then, after receiving Θ, it also outputs z. Otherwise, $\mathcal{S}$ proceeds as follows: it outputs $\mathcal{A}$'s *query* x_i, and after receiving Θ, it reprograms the random oracle at the point x_i to Θ, answers $\mathcal{A}$ with this reprogrammed value, runs $\mathcal{A}$ to the very end to get $\mathcal{A}$'s output (x,z), and then outputs z.

Let $t\in\{1,\ldots,q+1\}$ be so that $x_t=x$ and $x_j\neq x$ for all $j<t$ (with the understanding here that $x_{q+1}=x$). The crucial observation is that if $i=t$, then the reprogramming is consistent with standard lazy sampling, i.e., with answering a fresh query (to x_t) by a random value (Θ). Thus, from $\mathcal{A}$'s perspective, it is just an ordinary run. Furthermore, if $i=t$ then $x_i=x_t=x$. Thus,

$$\begin{aligned}\Pr\big[V(x_i,\Theta,z)\big] &\geq \Pr\big[i=t\big]\Pr\big[V(x_i,\Theta,z)\,\big|\,i=t\big] = \frac{1}{q+1}\Pr\big[V(x_i,\Theta,z)\,\big|\,i=t\big]\\ &= \frac{1}{q+1}\Pr\big[V(x,H(x),z)\,\big|\,i=t\big] = \frac{1}{q+1}\Pr\big[V(x,H(x),z)\big]\,.\end{aligned}$$

Hence, indeed,

$$\Pr\big[V(x,\Theta,z) \,:\, (x,z) \leftarrow \langle \mathcal{S},\Theta\rangle\big] \gtrsim \frac{1}{q}\Pr\big[V(x,H(x),z) \,:\, (x,z)\leftarrow \mathcal{A}^H\big]\,.$$

The Post-Quantum Case. When considering an attacker $\mathcal{A}_*$ that makes quantum queries to the random oracle, then one faces multiple hurdles. A first observation is that the analysis from the classical case does not carry over: with the queries being quantum now, the random variable t, which is meant to capture the first query made to x, is not properly defined (or does not lead to the same implications).

Also, the design of $\mathcal{S}$ raises concerns. Given that the queries are now quantum, but $\mathcal{S}$ still needs to output a classical x, if we want to follow the design from the classical case then $\mathcal{S}$ will have to *measure* the i-th query made by $\mathcal{A}_*$. This is clearly possible but raises the question whether $\mathcal{A}_*$ would notice that the query got measured, and then, say, abort (and so $\mathcal{S}$ will not get hold of z).

As a matter of fact, at first glance this looks concerning. There is a particular strategy for $\mathcal{A}_*$ to make a query to the random oracle and then, by means of a second query, to "uncompute" the first, so that, by checking if the uncomputing has worked, $\mathcal{A}_*$ can then find out if one of the two queries got measured. Thus, if $\mathcal{A}_*$ is running this strategy then it *will* detect when a query is measured, and may then indeed abort.

On the positive side, when inspecting this particular strategy we see that, while learning whether a query got measured or not, $\mathcal{A}_*$ will then not learn anything about the random oracle H — informally, that is because $\mathcal{A}_*$ uncomputes everything it learns about H. Thus, intuitively, we expect that, in order to be successful in finding (x,z) such that $V(x,H(x),z)$ is satisfied (while still trying to detect if we measure one of the queries), it either needs do at least one query without uncomputing it — but then it is not going to notice if we measure that very query — or the predicate V does not much depend on H — and then there is a good chance that $V(x,\Theta,z)$ is satisfied when $V(x,H(x),z)$ is.

The above intuition suggests that the construction of $\mathcal{S}$, subject to measuring the query, should actually still work. However, the intuition is geared by a particular strategy for $\mathcal{A}_*$ and it remains unclear on how

to formally argue that $\mathcal{S}$ indeed works properly for *any* $\mathcal{A}_*$. The following is a modified construction of $\mathcal{S}$, where we can offer no intuitive argument for why the modification is necessary or helpful, beyond that the mathematics then works out in our analysis.

Like in the classical case, $\mathcal{S}$ runs $\mathcal{A}_*$ while simulating the random oracle H to answer the queries made by $\mathcal{A}_*$ (here, the queries are quantum, so this is already non-trivial, but there are ways to do it, e.g., using the compressed oracle technique from Section 5.1). Also, as in the classical case, $\mathcal{S}$ samples an index $i \in \{1, \ldots, q+1\}$, which specifies the query from which $\mathcal{S}$ will get its output (for the first stage). However, there are three crucial differences:

1. The sampling of i is slightly biased in that each of the q actual queries is picked with probability $\frac{2}{2q+1}$, while the final output is picked with probability $\Pr[i = q+1] = \frac{1}{2q+1}$,
2. $\mathcal{S}$ gets its output for the first stage by *measuring* the i-th query made by $\mathcal{A}_*$, and
3. this i-th query is then answered *either* using the original H *or* the reprogrammed oracle where $H(x) := \Theta$, *with the choice made at random*, while all remaining queries are answered using the reprogrammed oracle.

In particular the last point is quite mysterious: why it is helpful to sometimes wait with the reprogramming? Unfortunately, we do not have a good explanation; we can "only" show that it works. Concretely, we will prove below that this construction of $\mathcal{S}$ satisfies

$$\Pr\big[x = x_\circ \wedge V(x, \Theta, z) \,:\, (x,z) \leftarrow \langle \mathcal{S}, \Theta \rangle\big] \geq \frac{1}{(2q+1)^2} \Pr\big[x = x_\circ \wedge V(x, H(x), z) \,:\, (x,z) \leftarrow \mathcal{A}_*^H\big]$$

for every fixed choice of $x_\circ$, and so, by summing over all choices of $x_\circ$, the same inequality (with the same $1/(2q+1)^2$ factor) holds also with the event $x = x_\circ$ removed. Thus, there is a quadratic loss in the bound compared to the classical case (and also to what the above intuition might suggest).

For the propose of the proof, we introduce the following notation. For $0 \leq i \leq j \leq q$ we set

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

to be the unitary map that describes the quantum evolution that the state, on which $\mathcal{A}_*$ is acting, is undergoing, from query $i+1$ onward to right before query $j+1$ (by convention $\mathcal{A}^H_{*,i\to j} = \mathbb{1}$ if $j \leq i$). We write $|\phi_0\rangle$ for $\mathcal{A}_*$'s initial state, so that

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

is then the state of $\mathcal{A}_*$ after the i-th step but right before the $(i+1)$-st query, with $\big|\phi_q^H\big\rangle$ then the final state of $\mathcal{A}_*$, given an arbitrary but fixed choice of $H : \mathcal{X} \to \mathcal{Y}$. Finally, for fixed $x_\circ \in \mathcal{X}$ and $\Theta \in \mathcal{Y}$, we define the reprogrammed function $H{*}\Theta x_\circ : \mathcal{X} \to \mathcal{Y}$ that coincides with H on $\mathcal{X} \setminus \{x_\circ\}$ but maps $x_\circ$ to Θ, and we let $X = |x_\circ\rangle\langle x_\circ|$ be the projector into the span of $|x_\circ\rangle$. Informally, X checks equality to $x_\circ$.

With this notation at hand, by elementary quantum information science, it holds that for any fixed choices of H and Θ

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

and

$$\Pr\big[x = x_\circ \wedge V(x, \Theta, z) \,:\, (x,z) \leftarrow \langle \mathcal{S}, \Theta \rangle\big] = \mathbb{E}_{i,b}\left[\big\| G_{x_\circ}^{\Theta} \big(\mathcal{A}^{H*\Theta x_\circ}_{*,i+b\to q}\big)\big(\mathcal{A}^H_{*,i\to i+b}\big) X \big|\phi_i^H\big\rangle \big\|_2^2\right],$$

where $G_{x_\circ}^{\Theta} = (|x_\circ\rangle\langle x_\circ| \otimes \Pi_{x_\circ,\Theta})$ is a projector that checks the equality $x = x_\circ$ and whether $V(x, \Theta, z)$ is satisfied.

The term for the second probability is harder to digest, but the reader should note that the index i determines the query that is measured (and checked to be equal to $x_\circ$), and the bit b decides whether H is reprogrammed now or from the next query onwards — very much in line with how $\mathcal{S}$ works.

Towards proving the claimed relationship, by inserting a resolution of the identity and exploiting that

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

which holds since $\mathbb{1}-X$ projects away from the only point where H and $H*\Theta x_\circ$ differ, we can write

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

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

$$\begin{aligned}
\left\|G^{\Theta}_{x_\circ}\left(\mathcal{A}^{H*\Theta x_\circ}_{*,i\to q}\right)\left|\phi^H_i\right\rangle\right\|_2 \le{}& \left\|G^{\Theta}_{x_\circ}\left(\mathcal{A}^{H*\Theta x_\circ}_{*,i+1\to q}\right)\left|\phi^H_{i+1}\right\rangle\right\|_2 + \left\|G^{\Theta}_{x_\circ}\left(\mathcal{A}^{H*\Theta x_\circ}_{*,i\to q}\right)X\left|\phi^H_i\right\rangle\right\|_2 \\
&+ \left\|G^{\Theta}_{x_\circ}\left(\mathcal{A}^{H*\Theta x_\circ}_{*,i+1\to q}\right)\left(\mathcal{A}^{H}_{*,\to i+1}\right)X\left|\phi^H_i\right\rangle\right\|_2 .
\end{aligned}$$

Summing up the respective sides of the inequality over $i = 0,\ldots,q-1$, and exploiting that many terms cancel out, we get

$$\left\|G^{\Theta}_{x_\circ}\left|\phi^{H*\Theta x_\circ}_q\right\rangle\right\|_2 \;\le\; \left\|G^{\Theta}_{x_\circ}\left|\phi^H_q\right\rangle\right\|_2 + \sum_{\substack{0\le i<q\\ b\in\{0,1\}}} \left\|G^{\Theta}_{x_\circ}\left(\mathcal{A}^{H*\Theta x_\circ}_{*,i+b\to q}\right)\left(\mathcal{A}^{H}_{*,i\to i+b}\right)X\left|\phi^H_i\right\rangle\right\|_2 .$$

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

$$\frac{\left\|G^{\Theta}_{x_\circ}\left|\phi^{H*\Theta x_\circ}_q\right\rangle\right\|_2^2}{2q+1} \le \left\|G^{\Theta}_{x_\circ}\left|\phi^H_q\right\rangle\right\|_2^2 + \sum_{\substack{0\le i<q\\ b\in\{0,1\}}} \left\|G^{\Theta}_{x_\circ}\left(\mathcal{A}^{H*\Theta x_\circ}_{*,i+b\to q}\right)\left(\mathcal{A}^{H}_{*,i\to i+b}\right)X\left|\phi^H_i\right\rangle\right\|_2^2 .$$

So, noting that we can write $\left\|G^{\Theta}_{x_\circ}\left|\phi^H_q\right\rangle\right\|_2^2$ as $\left\|G^{\Theta}_{x_\circ}\left(\mathcal{A}^{H*\Theta x_\circ}_{*,i+b\to q}\right)\left(\mathcal{A}^{H}_{*,i\to i+b}\right)X\left|\phi^H_i\right\rangle\right\|_2^2$ with $i=q$ and $b=0$,

$$\frac{\left\|G^{\Theta}_{x_\circ}\left|\phi^{H*\Theta x_\circ}_q\right\rangle\right\|_2^2}{(2q+1)^2} \;\le\; \mathbb{E}_{i,b}\left[\left\|G^{\Theta}_{x_\circ}\left(\mathcal{A}^{H*\Theta x_\circ}_{*,i+b\to q}\right)\left(\mathcal{A}^{H}_{*,i\to i+b}\right)X\left|\phi^H_i\right\rangle\right\|_2^2\right] .$$

This holds for any fixed choice of H and Θ. The claimed inequality now follows by observing that for random H and Θ, the function $H*\Theta x_\circ$ is just a random function again, and so on average over the choices of H and Θ there is no difference between $\left\|G^{\Theta}_{x_\circ}\left|\phi^{H*\Theta x_\circ}_q\right\rangle\right\|_2^2$ and $\left\|G^{H(x_\circ)}_{x_\circ}\left|\phi^H_q\right\rangle\right\|_2^2$.

6 Quantum Security of Generic Design Principles

In this section, we analyze several frequently used techniques in constructing cryptographic schemes, and discuss the security loss in the reductions to the underlying primitives.

Constants, Simplifications, Aesthetics. We point out that the difference between classical and quantum bounds found in the literature is not always exclusively due to the quantum aspects. Sometimes, the quantum proof introduces an additional trick that was not exploited in the classical bound, e.g., to improve a constant. Sometimes, the resulting formulas are simplified by leaving out non-dominating factors.

Oracles. The reduction losses presented in this section depend on the number of queries the attacker makes to certain oracles. Specifically, the parameters of interest are the number of random oracle queries q_{RO} (where the random oracle models the hash function used in the construction), the number of decryption queries q_D, and the number of signing queries q_S. We recall that in case of a quantum attacker, we consider the standard Q1 security model, where the decryption queries and the signing queries remain classical, but where the q_{RO} random oracle queries may be quantum.

Therefore, in the Q1 security model we consider, the q_D decryption queries and the q_S signing queries are always classical, whereas the q_{RO} random oracle queries may be quantum. See Section 4.3 for a brief discussion of the security model under consideration.

Square-Root Loss. In reductions involving quantum attackers, square roots often emerge in the formulas. However, as advantages (and success probabilities) are all numbers $0 \leq \epsilon \leq 1$, so $\epsilon^2 \leq \epsilon \leq \sqrt{\epsilon}$. Therefore, a square root in the reduction makes the reduction loss *larger*, and the reduction *worse*. Such a square-root increase in the advantage is generally considered rather bad, as it reduces the bit security (in the worst case) by a factor 2; see the discussion on this in Section 7. However, the precise effect strongly depends on the shape of the advantage as a function of the running time. For instance, for many computational problems the best algorithm takes some time T to solve the problem (with high probability), and for any lower running time the advantage is essentially zero. In case of such an approximate step function, the square-root has almost no effect.

6.1 Fujisaki–Okamoto (FO)

The Fujisaki–Okamoto (FO) transformation serves the purpose of enhancing the security of a weakly secure, i.e., OW-CPA or IND-CPA secure, public-key encryption (PKE) scheme, to an IND-CCA secure key-encapsulation mechanism (KEM), see Section 3 for formal definitions of PKEs and KEMs. The FO transformation involves two cryptographic hash functions G and H, which are modeled in the analysis as random oracles.

The transformation is specified in Figure 8. For encapsulation, a random message m (think of it as a "raw key") is encrypted by the PKE scheme $\mathcal{E}$ using $G(m)$ as the randomness for encryption, and the KEM session key K is then given by the hash $H(m)$. For decapsulation, the ciphertext is decrypted with the corresponding secret key to recover m, and then m is *re-encrypted* with the same randomness $G(m)$ and it is *checked* that this gives back the original ciphertext. If this is the case then the decapsulation outputs the KEM session key $K = H(m)$, otherwise it aborts, i.e., outputs some dummy symbol $\perp$.

Informally, the purpose of this re-encryption check is to make it infeasible to produce a valid ciphertext without knowing the message, which, again informally, makes decapsulation queries useless in the CCA security game.

$\mathsf{Encaps}(\mathsf{pk})$	$\mathsf{Decaps}(\mathsf{sk}, c)$
1: $m \leftarrow \mathcal{M}$	1: $m \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$
2: $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m; G(m))$	2: **if** $\mathsf{Enc}(\mathsf{pk}, m; G(m)) \neq c$
3: $K := H(m)$	3: **return** $\perp$
4: **return** (c, K)	4: **return** $H(m)$

Figure 8: The (plain) FO transformation, turning a PKE scheme $\mathcal{E} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ into a KEM $\mathsf{FO}[\mathcal{E}, H, G] = (\mathsf{KGen}, \mathsf{Encaps}, \mathsf{Decaps})$, where H and G are cryptographic hash functions.

For the formal security analysis of the FO transformation, two additional properties of the original PKE scheme $\mathcal{E}$ are relevant: δ-correctness, and γ-spreadness. Roughly speaking, the parameter δ bounds the probability of a decryption error, and γ captures the amount of randomness (i.e., min-entropy) in a ciphertext. As we can see from the bounds below, we want δ to be small and γ to be large. We refer the interested reader to [HHK17] for a more formal introduction to these properties/parameters. Also, here and below, we

denote by $\mathcal{M}$ the message space of the original PKE scheme $\mathcal{E}$, i.e., the space from which the "raw key" m is sampled.

Theorem 2. *Let $\mathcal{E}$ be a γ-spread and δ-correct PKE, and $\mathsf{FO}[\mathcal{E}, H, G]$ be the KEM obtained by applying the plain FO transform, where H and G are modeled as random oracles. Let $\mathcal{A}$ be an $\mathsf{IND\text{-}CCA}$ attacker making at most $q_D \geq 1$ decapsulation queries and at most q_{RO} queries to H and G in total. In the case where $\mathcal{A}$ can only make classical queries to H and G, there exist similarly efficient $\mathsf{IND\text{-}CPA}$ and $\mathsf{OW\text{-}CPA}$ attackers $\mathcal{B}$ and $\mathcal{C}$, respectively, such that, as shown in [HHK17, Section 3.3], we have*

$$\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{FO}[\mathcal{E},H,G]}(\mathcal{A}) \leq 2(q_{RO}+q_D)\delta + q_{RO}2^{-\gamma} + \begin{cases} \frac{2q_{RO}+1}{|\mathcal{M}|} + 3\mathbf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{E}}(\mathcal{B}) \\ (q_{RO}+1)\mathbf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathcal{E}}(\mathcal{C}). \end{cases}$$

In the case where $\mathcal{A}$ can make quantum queries to H and G in general, which we denote by $\mathcal{A}_* = \mathcal{A}$, there exists similarly efficient attackers $\mathcal{B}_*, \mathcal{C}_*$ making quantum queries to H and G such that, as shown in [HM24, Theorem 1] and [DFMS22, Theorem 4], we furthermore have

$$\begin{aligned} &\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{FO}[\mathcal{E},H,G]}(\mathcal{A}_*) \\ &\quad\leq \begin{cases} 10(q_{RO}+1)^2\delta + 24q_D(q_{RO}+4q_D)\cdot 2^{-\gamma/2} + \frac{8(q_{RO}+q_D)}{\sqrt{|\mathcal{M}|}} + 4\sqrt{(q_{RO}+q_D)\cdot \mathbf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{E}}(\mathcal{B}_*)} \\ 24(q_{RO}+2q_D)^2\sqrt{\delta} + 24\sqrt{(q_{RO}+2q_D)^3 q_D}\,2^{-\gamma/4} + 2(q_{RO}+2q_D)\sqrt{\mathbf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathcal{E}}(\mathcal{C}_*)}\,. \end{cases} \end{aligned}$$

Immediately, one can see a clear difference between classical and quantum bounds. In the quantum case, there is a square-root enclosing the $\mathsf{IND\text{-}CPA/CCA}$ advantage, which is not present in the classical case.

In the literature and in practical schemes, different variants of the *plain* FO transformation can be found. One prominent variant is the *implicit-rejection* variant (used for instance in *Kyber*, *HQC*, or *FrodoKEM*), which modifies the re-encryption check: in case the ciphertext comparison fails, the decapsulation does not abort but instead outputs a pseudorandom value $PRF(m)$. We denote this variant by $\mathsf{FO}^{\not\perp}[\mathcal{E}, H, G, PRF]$. As shown in [HHK17, Section 3.3], [JZC+18, Theorem 2] and [JZM19, Theorem 1], one can obtain the following classical and quantum bounds for $\mathcal{A}$ and $\mathcal{A}_*$ denoting classical and quantum $\mathsf{IND\text{-}CCA}$ attackers, respectively

$$\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{FO}^{\not\perp}[\mathcal{E},H,G,\mathsf{PRF}]}(\mathcal{A}) \leq 2(q_{RO}+q_D)\delta + \mathbf{Adv}^{\mathsf{PRF}}(\mathcal{D}) + \begin{cases} \frac{2q_{RO}+1}{|\mathcal{M}|} + 3\mathbf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{E}}(\mathcal{B}) \\ (q_{RO}+1)\mathbf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathcal{E}}(\mathcal{C})\,, \end{cases} \tag{2}$$

$$\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{FO}^{\not\perp}[\mathcal{E},H,G,\mathsf{PRF}]}(\mathcal{A}_*) \leq 4q_{RO}\sqrt{\delta} + \mathbf{Adv}^{\mathsf{PRF}}(\mathcal{D}_*) + \begin{cases} 2\sqrt{(q_{RO}+1)\mathbf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{E}}(\mathcal{B}_*) + 2\frac{(q_{RO}+1)^2}{|\mathcal{M}|}} \\ 2q_{RO}\sqrt{\mathbf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathcal{E}}(\mathcal{C}_*)}\,, \end{cases} \tag{3}$$

for $\mathcal{B}, \mathcal{B}_*, \mathcal{C}, \mathcal{C}_*$ as in Theorem 2, and $\mathcal{D}, \mathcal{D}_*$ attackers for the pseudorandom function (PRF) used in the implicit rejection. Similar to the plain FO, there is also a square-root loss occurring in the quantum case.

One motivation of using implicit-rejection is to trade the γ-spreadness assumption of the original PKE $\mathcal{E}$ with the pseudorandomness of PRF. Indeed, the reduction loss for the implicit-rejection variant no-longer depend on the parameter γ, but now with an additive loss being the PRF advantage.

Another variant of the FO transformation (which can be combined with explicit or implicit rejection) is obtained by setting $K := H(m, c)$, i.e., as the hash of the message *and the ciphertext*. For instance, *Kyber* originally adopted this modification, but the NIST standardized version *ML-KEM* instead hashes the message with the public (encapsulation) key. Furthermore, some variant of the transformation comes with an additional key confirmation hash attached to the ciphertext [HHK17].

Some of these variants were introduced and considered in order to facilitate a quantum security proof (or to get a better reduction), but the latest results on the plain FO transform, provided by [DFMS22] and given above make some of these variants obsolete from a theoretical perspective. However, practical schemes occasionally make modifications to the FO transform in their designs, despite making security claims based on the standard versions. Therefore, care is needed in reading security proofs *in the wild*.

6.2 Fiat–Shamir (FS)

An interactive proof is an interactive protocol between a prover and a verifier, in which the prover aims to convince the verifier of the validity of a statement, typically by demonstrating knowledge of a (secret) witness. Certain interactive proofs can be used as identification schemes. Because these schemes are interactive, they generally require multiple rounds of communication between the prover and the verifier.

The Fiat–Shamir (FS) transformation allows specific classes of interactive proofs to be converted into non-interactive ones. This approach has become a common design principle for constructing digital signature algorithms (DSAs).

In this section, we explain how the Fiat–Shamir transformation works and outline its security properties. As before, we highlight the differences between the classical and quantum security analyses. While the Fiat–Shamir transformation applies to interactive protocols with an arbitrary number of rounds, we restrict the discussion to three-round interactive proofs, also known as Sigma protocols.

Sigma protocols. We begin by explaining what a Sigma protocol is. Let $\mathcal{X}$ and $\mathcal{W}$ be finite sets, and let D be an efficiently sampleable distribution over a relation $R \subseteq \mathcal{X} \times \mathcal{W}$. If $(x, w) \in R$, the element w is called a witness for the statement x. A Sigma protocol $\Sigma = (\mathcal{P}, \mathcal{V})$ for the relation R is a three-move interactive protocol that allows a prover $\mathcal{P}$ to convince a verifier $\mathcal{V}$ that it knows a (secret) witness w for a public statement x. A Sigma protocol is somewhat meaningless if a witness w can be computed efficiently from a statement x. For this reason, we only consider relations for which it is computationally hard to find w given (only) x.

In the first move, the prover, on input (x, w), sends a commitment a to the verifier. In the second move, the verifier samples a challenge c uniformly at random from some challenge set. In the third move, the prover sends a response z. Based on the statement x together with the transcript (a, c, z), the verifier decides whether to accept the prover's claim.

We assume that a Sigma protocol Σ satisfies the following three security properties:

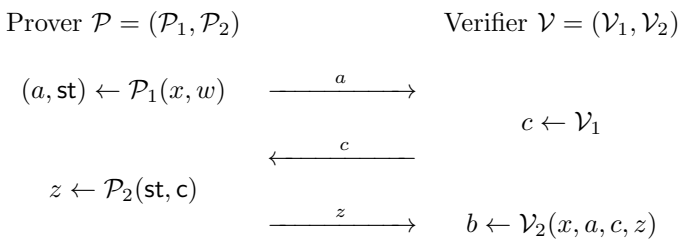
- *Completeness*: An honest prover always convinces an honest verifier.
- *Soundness*: A dishonest prover succeeds in convincing the verifier only with negligible probability. More precisely, for every polynomial-time algorithm $\mathcal{P}^* = (\mathcal{P}_1^*, \mathcal{P}_2^*)$ given x (but not w), it is hard to convince the verifier in Σ, i.e.,

$$\mathbf{Adv}_D^{\Sigma}(\mathcal{A}) := \Pr\left[\mathcal{V}_2(x, a, c, z) = 1 \;\middle|\; \begin{array}{c}(x, w) \leftarrow D\ , \quad (a, \mathsf{st}) \leftarrow \mathcal{P}_1^*(x) \\ c \leftarrow \mathcal{V}_1\ , \quad z \leftarrow \mathcal{P}_2^*(\mathsf{st}, c)\end{array}\right] \leq \mathsf{negl}(\lambda)\ .$$

- *(Honest-Verifier) Zero-Knowledge (HVZK)*: An honest verifier learns nothing beyond the fact that the prover knows a witness, i.e., there exists a polynomial-time algorithm Sim_{Σ} such that the statistical distance ζ, between honestly generated transcripts (x, a, c, z) and simulated transcripts $\mathsf{Sim}_{\Sigma}(x)$, is negligibly small. This ensures that a transcript, and later a digital signature, does not reveal any information about the secret witness.

We further assume that, on average over the choice of $(x, w) \leftarrow D$, the message a is of high min-entropy, i.e., $\mathsf{guess}(a \mid x, w) := \max_f \Pr[a = f(x, w)] \leq \epsilon$ (where the maximization is taken over all functions f) is negligibly small. This technical assumption is needed for the UF-CMA to UF-NMA security reduction. More precisely, it ensures that the attacker queries H on the same message a twice only with negligible probability. Such an event would otherwise invalidate the reduction argument.

Fiat–Shamir transformation. The Fiat–Shamir transformation, using a hash function H, can be used to convert the interactive Σ into a non-interactive digital signature scheme by replacing the verifier's challenge with a hash value. Following this principle, the public key is a statement x for which a witness w exists, and the private key is the witness w. These keys are generated via the distribution D over the relation R. A signature on a message m consists of an accepting transcript (a, c, z) for the statement x, where the challenge



KGen	$\mathsf{Sign}(\mathsf{pk}, m)$	$\mathsf{Vrfy}(\mathsf{pk}, \mathsf{sk}, \sigma)$
1: $(x, w) \leftarrow D$	1: $(a, \mathsf{st}) \leftarrow \mathcal{P}_1(x, w)$	1: $c := H(\mathsf{pk}, a, m)$
2: $\mathsf{sk} := (x, w)$	2: $c := H(\mathsf{pk}, a, m)$	2: **return** $\mathcal{V}_2(x, a, c, z)$
3: $\mathsf{pk} := x$	3: $z \leftarrow \mathcal{P}_2(\mathsf{st}, c)$	
4: **return** $(\mathsf{sk}, \mathsf{pk})$	4: **return** $\sigma := (a, z)$	

Figure 9: The Sigma protocol Σ, and the Fiat–Shamir signature scheme $\mathsf{FS}[\Sigma, D, H] = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$.

is computed as $c = H(x, a, m)$. Figure 9 provides the formal description of both the Sigma protocol and the Fiat–Shamir signature scheme constructed from it.

Security proofs for Fiat–Shamir signatures are typically carried out in two steps. First, UF-CMA-security is reduced to UF-NMA-security under the HVZK assumption of the Sigma protocol. Second, UF-NMA-security is reduced to the security of the underlying Sigma protocol.

We begin with the reduction from UF-CMA-security to UF-NMA-security. The reduction loss depends on the parameters q_{RO} and q_S, denoting the number of (classical) queries the attacker makes to the random oracle (modeling the hash function H) and the number of (quantum) queries to the signing oracle, respectively. The following bounds are known for classical and quantum attackers. Recall that ϵ and ζ are determined by the properties of the underlying Sigma protocol, and should be negligible.

The reductions show that, for any classical attacker $\mathcal{A}$ and any quantum attacker $\mathcal{A}_*$, there exist corresponding classical and quantum attackers $\mathcal{B}$ and $\mathcal{B}_*$ with similar running times such that

$$\mathbf{Adv}^{\mathsf{UF\text{-}CMA}}_{\mathcal{S}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{UF\text{-}NMA}}_{\mathcal{S}}(\mathcal{B}) + \frac{q_S(q_S+1)\epsilon}{2} + q_S(q_{RO}+1)\epsilon + q_S\zeta\,, \qquad \text{(Adapted from [FFH25])}$$

$$\mathbf{Adv}^{\mathsf{UF\text{-}CMA}}_{\mathcal{S}}(\mathcal{A}_*) \leq \mathbf{Adv}^{\mathsf{UF\text{-}NMA}}_{\mathcal{S}}(\mathcal{B}_*) + 3q_S\sqrt{(q_{RO}+1)\epsilon} + q_S\zeta\,. \qquad \text{(Adapted from [FFH25])}$$

In the quantum setting, the square-root dependence leads to a larger reduction loss. Apart from this difference, however, the additive losses remain qualitatively similar. Note that, in the quantum reduction, the additive term $q_S(q_S+1)\epsilon/2$ that appears in the classical reduction can be absorbed into the subsequent term, using the fact that

$$q_S^2\epsilon \leq \sqrt{q_S^2\epsilon} = q_S\sqrt{\epsilon} \leq q_S\sqrt{(q_{RO}+1)\epsilon}\,.$$

This does increase the associated constant, but recall that such constants can be somewhat arbitrary anyway. It is to allow slightly larger constants in exchange for more elegant, or simplified, expressions. Moreover, changes in constants when transitioning from the classical to the quantum setting may simply reflect minor differences in the proof technique or additional optimizations, rather than any inherently quantum effect.

We now turn to the reduction of UF-NMA-security to the security of the underlying Σ-protocol. For any classical attacker $\mathcal{B}$ and any quantum attacker $\mathcal{B}_*$, there exist corresponding classical and quantum attackers $\mathcal{C}$ and $\mathcal{C}_*$ with similar running times such that

$$\mathbf{Adv}^{\mathsf{UF\text{-}NMA}}_{\mathcal{S}}(\mathcal{B}) \leq (q_{RO}+1)\cdot\mathbf{Adv}^{\Sigma}_{D}(\mathcal{C})\,, \qquad \text{(Folklore)}$$

$$\mathbf{Adv}^{\mathsf{UF\text{-}NMA}}_{\mathcal{S}}(\mathcal{B}_*) \leq (2q_{RO}+1)^2\cdot\mathbf{Adv}^{\Sigma}_{D}(\mathcal{C}_*)\,. \qquad \text{([DFM20])}$$

Hence, the classical reduction incurs a multiplicative loss that grows linearly in q_{RO}, whereas the quantum reduction incurs a multiplicative loss that is quadratic in q_{RO}.

The final step is to bound the advantage $\mathbf{Adv}_D^{\Sigma}$ of breaking Σ in terms of the advantage of solving the underlying computational problem. More precisely, we reduce the task of finding a witness w for a statement x from an honestly generated instance $(x, w) \leftarrow D$ to that of breaking Σ. The corresponding reduction requires *rewinding* the interactive Sigma protocol, and the resulting loss depends critically on the concrete soundness properties of the protocol. For this reason, we suppress certain non-dominating additive reduction losses that depend on the underlying Σ-protocol.

Further, we assume that the Sigma protocol satisfies 2-special-soundness, meaning that the reduction must rewind only once. This property is satisfied by many Sigma protocols used in digital signature constructions (see Remark 2 for the more general case).

The reduction yields that for any classical attacker $\mathcal{C}$ and any quantum attacker $\mathcal{C}_*$, there exist corresponding classical and quantum attackers $\mathcal{D}$ and $\mathcal{D}_*$ with comparable running time such that

$$\mathbf{Adv}_D^{\Sigma}(\mathcal{C}) \lesssim \sqrt{\Pr_{(x,w)\leftarrow D}[\,\mathcal{D}(x) = w\,]}\,, \tag{Folklore}$$

$$\mathbf{Adv}_D^{\Sigma}(\mathcal{C}_*) \lesssim \sqrt[3]{\Pr_{(x,w)\leftarrow D}[\,\mathcal{D}_*(x) = w\,]}\,, \tag{[Unr12]}$$

where $\lesssim$ hides some non-dominating additive reduction loss.

Combining the three reductions, we obtain the following classical and quantum bounds:

$$\mathbf{Adv}_{\mathcal{S}}^{\mathsf{UF\text{-}CMA}}(\mathcal{A}) \lesssim (q_{RO}+1)\sqrt{\Pr_{(x,w)\leftarrow D}[\,\mathcal{D}(x) = w\,]} + \frac{q_S(q_S+1)\epsilon}{2} + q_S(q_{RO}+1)\epsilon + q_S\zeta\,,$$

$$\mathbf{Adv}_{\mathcal{S}}^{\mathsf{UF\text{-}CMA}}(\mathcal{A}_*) \lesssim (2q_{RO}+1)^2\sqrt[3]{\Pr_{(x,w)\leftarrow D}[\,\mathcal{D}_*(x) = w\,]} + 3q_S\sqrt{(q_{RO}+1)\epsilon} + q_S\zeta\,.$$

Remark 2 (k-Special-Soundness)**.** *If the underlying Sigma protocol satisfies the more general notion of k-special-soundness for some $k \in \mathbb{N}$, then the corresponding reductions are required to rewind $k-1$ times and the bounds take the following form:*

$$\mathbf{Adv}_D^{\Sigma}(\mathcal{C}) \lesssim \sqrt[k]{\Pr_{(x,w)\leftarrow D}[\,\mathcal{D}(x) = w\,]}\,, \tag{Folklore}$$

$$\mathbf{Adv}_D^{\Sigma}(\mathcal{C}_*) \lesssim \sqrt[2k-1]{\Pr_{(x,w)\leftarrow D}[\,\mathcal{D}_*(x) = w\,]}\,. \tag{[DFMS19]}$$

Thus, the reduction incurs a larger loss as k increases.

Remark 3 (Forking Lemma)**.** *Classically, a shortcut via the forking lemma exists, which reduces the reduction loss. The forking lemma reduces UF-NMA security directly to the hardness of the underlying computational problem, thereby bypassing the intermediate reduction to the security of the Sigma protocol. This avoids a multiplicative loss of $\sqrt{q_{RO}+1}$ in the reduction. More precisely, we obtain the bound*

$$\mathbf{Adv}_{\mathcal{S}}^{\mathsf{UF\text{-}NMA}}(\mathcal{B}) \lesssim \sqrt{(q_{RO}+1)\Pr_{(x,w)\leftarrow D}[\,\mathcal{D}(x) = w\,]}\,.$$

Currently, no quantum counterpart of the forking lemma is known.

Remark 4 (Unique Responses)**.** *For the quantum bounds in this (and the next) subsection to hold, the underlying Σ-protocol must satisfy an additional technical requirement. It must either have* unique responses *or, more relaxed, satisfy a so-called* collapsing *property. We refer the reader to [DFMS19] for further details.*

6.3 Fiat–Shamir with Aborts

Many post-quantum digital signature schemes follow the same Fiat–Shamir design paradigm. A crucial difference, however, is that in some of these schemes (e.g., *ML-DSA*) the underlying interactive proof system admits a non-zero *abort probability*. More precisely, an honest prover may abort the protocol after receiving the verifier's challenge, in which case it outputs $z = \perp$. This abort mechanism is essential to ensure that protocol transcripts leak no information about the prover's secret witness. In particular, their distribution must remain independent of the witness.

As in the classical setting, the verifier's challenges can be replaced by evaluations of a hash function, thereby rendering the protocol non-interactive. The key difference with the traditional Fiat–Shamir transformation is that the non-interactive prover repeatedly samples fresh randomness until it produces a transcript that does not abort. This transformation is known as *Fiat–Shamir with Aborts* (FSwA).

The resulting signature scheme $\mathsf{FSwA}[\Sigma, D, H]$ is summarized in Figure 10, where H denotes a cryptographic hash function.

KGen	$\mathsf{Sign}(\mathsf{pk}, m)$	$\mathsf{Vrfy}(\mathsf{pk}, \mathsf{sk}, \sigma)$
1: $(x, w) \leftarrow D$	1: **repeat**	1: $c := H(\mathsf{pk}, a, m)$
2: $\mathsf{sk} := (x, w)$	2: $(a, \mathsf{st}) \leftarrow \mathcal{P}_1(x, w)$	2: **return** $\mathcal{V}_2(x, a, c, z)$
3: $\mathsf{pk} := x$	3: $c := H(\mathsf{pk}, a, m)$	
4: **return** $(\mathsf{sk}, \mathsf{pk})$	4: $z \leftarrow \mathcal{P}_2(\mathsf{st}, c)$	
	5: **until** $z \neq \perp$	
	6: **return** $\sigma := (a, z)$	

Figure 10: The Fiat–Shamir with aborts signature scheme $\mathsf{FSwA}[\Sigma, D, H]$.

A subtle but important nuance arises here. Typically, only the distribution of *non-aborting* transcripts is independent of the secret witness. Hence, an aborting transcript may in fact leak information about the witness. Interactive proofs of this form therefore satisfy a weaker zero-knowledge property, known as *accepting* honest-verifier zero-knowledge (*accepting* HVZK), rather than full HVZK. Since, in the non-interactive setting, the prover outputs only non-aborting transcripts, this weaker property is sufficient for most applications. The shift from the plain Fiat–Shamir transformation to FSwA, however, affects the security reduction. In particular, the CMA-to-NMA reduction must account for the abort probability, whereas the remaining two reductions are unaffected by the fact that the protocol may abort.

We now discuss in more detail the CMA-to-NMA reduction for FSwA. Let Σ_{ab} be a Σ-protocol with abort probability $p = \Pr[z = \perp]$. As before, we assume that the prover's first message a has high min-entropy, i.e

$$\mathsf{guess}(a \mid x, w) := \max_f \Pr[a = f(x, w)] \leq \epsilon$$

is negligible. Moreover, we assume that Σ_{ab} satisfies *accepting* HVZK. More precisely, there exists a polynomial-time simulator $\mathsf{Sim}_{\Sigma_{\mathsf{ab}}}$ such that the statistical distance ζ between honestly generated *accepting* transcripts (x, a, c, z) and simulated transcripts $\mathsf{Sim}_{\Sigma_{\mathsf{ab}}}(x)$ is negligible. In other words,

$$\mathsf{SD}((x, a, c, z), \mathsf{Sim}_{\Sigma_{\mathsf{ab}}}(x) \mid z \neq \perp) = \zeta$$

is negligibly small.

The FSwA CMA-to-NMA reductions then show that, for any classical attacker $\mathcal{A}$ and any quantum attacker $\mathcal{A}_*$, there exist corresponding classical and quantum attackers $\mathcal{B}$ and $\mathcal{B}_*$ with comparable running times such that

$$\mathbf{Adv}^{\mathsf{UF\text{-}CMA}}_{\mathcal{S}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{UF\text{-}NMA}}_{\mathcal{S}}(\mathcal{B}) + \frac{q_S(q_S+1)\epsilon}{2(1-p)} + \frac{2q_S(q_{RO}+1)\epsilon}{1-p} + q_S\zeta\ , \qquad ([\mathrm{BBD}^+23])$$

$$\mathbf{Adv}^{\mathsf{UF\text{-}CMA}}_{\mathcal{S}}(\mathcal{A}_*) \leq \mathbf{Adv}^{\mathsf{UF\text{-}NMA}}_{\mathcal{S}}(\mathcal{B}_*) + \frac{8q_S\sqrt{(q_{RO}+1)\epsilon}}{1-p} + q_S\zeta\ . \qquad ([\mathrm{FFH25}])$$

Up to small constant factors, the only difference from the ordinary Fiat–Shamir reduction is the additional multiplicative term $1/(1-p)$. This factor arises from the expected number of attempts required for the prover to obtain an accepting transcript (and hence a valid signature).

Remark 5 (Hash-and-Sign)**.** *Another common design principle for digital signature algorithms is the* Hash-and-Sign *(HS) paradigm. Analogous to Fiat–Shamir with Aborts (FSwA), the Hash-and-Sign variant in which the signing procedure is repeated until a valid signature is produced is known as* Hash-and-Sign with Retry *(HSwR).*

Signatures obtained through this paradigm are subject to security reductions similar to those for the Fiat–Shamir (FS) transformation. In fact, a Hash-and-Sign signature scheme can be viewed as the Fiat–Shamir transformation applied to a particular Σ-protocol. This observation allows the CMA-*to*-NMA *reductions developed for FS and FSwA to be carried over directly to HS and HSwR, respectively.*

In practice, however, the reduction from UF-NMA *security of HS/HSwR to the underlying computational problem may be more straightforward. Consequently, the more elaborate reductions required in the FS and FSwA settings can sometimes be avoided in the HS and HSwR case.*

7 Quantum Security of Concrete Schemes

In this section, we discuss the security justification of several concrete schemes. The list includes KEMs — *Kyber*, *HQC*, *FrodoKEM*, as well as digital signature schemes — *Dilithium*, *Falcon* $^+$ (the updated version of *Falcon*), and *SPHINCS*$^+$. For all of these schemes, analyses are performed in the random oracle model, where the underlying cryptographic hash functions being used are modeled as random oracles. We therefore denote by q_{RO} as the maximum number of RO queries that the considered attacker is allowed make.

For efficiency and other reasons, in many of these schemes, the randomness that goes into the key-generation and the encapsulation/signing procedure is produced by choosing a random seed and expanding it using a suitable cryptographic function. In the security analyses of the schemes, this aspect is sometimes explicitly treated either by assuming the expansion function to be a pseudo-random function or by modeling it as a random oracle, and sometimes it is ignored. For simplicity and in order to have a uniform exposition, we leave this randomness expansion implicit. This means that the bounds we present apply when all the randomness is truly random; to compensate for this simplification, a small additive term should be added to the bounds, either a term $\mathbf{Adv}^{\mathsf{PRF}}(\mathcal{C})$ assuming the expansion function to be a pseudo-random function, or a small statistical term when modeling it as a random oracle.

Also here, q_{RO} will denote the number of random-oracle queries the considered attacker may do (which may be quantum queries when considering a quantum attacker), and q_D and q_S refer to the number of decapsulation and signing queries, respectively (which are classical no matter what).

Many of the reductions we discuss below bound the relevant advantage by means of several additive terms. The different terms are often of a somewhat different nature, and typically not equally relevant; though it is not always clear if there is a single dominating term. We present these bound with a somewhat systematic ordering of the terms, and so that, as far as we understand, the (potentially) dominating terms are at the beginning of the expression and the less relevant ones at the end.

7.1 Kyber

The Scheme. Here we consider the KEM *Kyber*, which was selected by the NIST as one of the post-quantum standards and standardized with minor changes as *ML-KEM* in FIPS 203. *Kyber* is a lattice-based schemes and its security relies on the *module learning with error* problem (MLWE).

Design Principle. The scheme *Kyber* is obtained by considering a CPA-secure version *CPA-Kyber*, and then applying (a non-standard variant of) the FO transformation.

Simplified, the CPA-version *CPA-Kyber* works as follows. The public key consists of a matrix A and of a vector t over some specific coeficient ring, such that $t \approx As$ (i.e., $t = As$ plus some small noise), where s is

the secret key. The encryption of a message m is given by a pair (u, v), computed such that

$$u \approx A^T r \qquad \text{and} \qquad v \approx t^T r + Encode(m)$$

for a suitable encoding that allows for error correction. Then, clearly,

$$s^T u - v \approx s^T A^T r - t^T r + Encode(m) \approx Encode(m)$$

and so the message m can be recovered by doing error correction. The security relies on a variant of the Learning with Errors problem (LWE): knowing A, recover the secret s from the noisy product $t \approx As$. For increased efficiency, the coefficient ring has some polynomial structure, and the matrix A is replaced with a small array of matrices, giving rise to the *Module Learning with Errors* problem (MLWE).

Because of the size of the vectors and matrices involved, *Kyber* replaces the sampling of randomness with a smaller random seed, which is expanded with a pseudorandom function.

The non-standard variant of the FO variant that is applied to obtain *Kyber* uses implicit rejection (using a pseudorandom function) and extracts the key from the message-ciphertext pair, but that also uses some non-standard nested hashing, and involves a hash of the public-key as well. Note that *ML-KEM* (as standardized by NIST) no longer hashes the ciphertext to obtain the session key but instead hashes the message m with the (hash of) the encapsulation key. However, this section follows the security analysis of the Round-3 Kyber submission [SAB+20]. The original motivation to change the FO transform in Kyber was to tie the ciphertext to a public-key to avoid *multi-target* attacks (against many different users); the modifications also allows for more advanced security properties of KEMs(see Section 9.2).

Security Reductions. As shown in [SAB+20, Theorem 1], the CPA security of the underlying scheme *CPA-Kyber* tightly reduces to the hardness of MLWE, in that

$$\mathbf{Adv}^{\mathsf{IND\text{-}CPA}}_{CPA\text{-}Kyber}(\mathcal{A}) \leq 2\mathbf{Adv}^{\mathsf{MLWE}}(\mathcal{B}) \tag{4}$$

for attackers with similar running times. Here and in the remainder, this is to be understood that for every $\mathcal{A}$ there exists an $\mathcal{B}$ with similar running times as $\mathcal{A}$, and for which the above inequality holds.

[SAB+20] is not very explicit about this, but we understand it that the bound is claimed to hold for *classical* and *quantum* attackers.

Moving to CCA security, things get more subtle. Given the use of the FO transform, one could simply apply the generic result for the FO transformation from Section 6.1, making necessary adjustments for the non-standard variant used in *Kyber*.

Instead, a direct analysis of *Kyber* is provided, i.e., a direct reduction of the security of *Kyber* to the hardness of MLWE. It turns out that in the case of a *classical* attacker, this saves the linear (in q_{RO}) security loss that we have in the bound in Section 6 . Concretely, the following bound is given in [SAB+20, Theorem 2]

$$\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{Kyber}(\mathcal{A}) \leq 2\mathbf{Adv}^{\mathsf{MLWE}}(\mathcal{B}) + \mathbf{Adv}^{\mathsf{PRF}}(\mathcal{C}) + 4q_{RO}\delta \tag{5}$$

for attackers with similar running times. Here, the decryption error probability δ of *CPA-Kyber* can be estimated heuristically; the $\mathbf{Adv}^{\mathsf{PRF}}$ comes from the fact that the implicit rejection uses a pseudo-random function.

For *quantum* attackers, as shown in [SAB+20, Theorem 3], the following holds

$$\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{Kyber}(\mathcal{A}_*) \leq 4q_{RO}\sqrt{\mathbf{Adv}^{\mathsf{MLWE}}(\mathcal{B}_*)} + \mathbf{Adv}^{\mathsf{PRF}}(\mathcal{C}_*) + 8q_{RO}^2\delta\,. \tag{6}$$

Neglecting the fact that *Kyber* uses a tweaked version of FO, the generic better bound [JZM19, Theorem 3] would give us

$$\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{Kyber}(\mathcal{A}_*) \leq \sqrt{8(q_{RO}+1)\mathbf{Adv}^{\mathsf{MLWE}}(\mathcal{B}_*) + \frac{2(q_{RO}+1)^2}{|\,\mathcal{M}\,|}} + 4q_{RO}\sqrt{\delta} + \frac{2q_{RO}}{\sqrt{|\,\mathcal{M}\,|}}\,. \tag{7}$$

To avoid the suboptimal square-root loss in the quantum setting, it is also possible to reduce the quantum security (more) tightly to a non-standard assumption. Specifically, as per [SAB+20, Section 4.3.2], it is reduced to the pseudorandomness of a deterministic variant *DKyber* of *Kyber* as

$$\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{Kyber}(\mathcal{A}_*) \leq 2\mathbf{Adv}^{\mathsf{MLWE}}(\mathcal{B}_*) + \mathbf{Adv}^{\mathsf{PRF}}(\mathcal{C}_*) + \mathbf{Adv}^{\mathsf{PR}}_{DKyber}(\mathcal{D}_*) + 8q^2_{RO}\delta$$

for quantum attackers with similar running times.

Brief Discussion. The classical bound (5) is obtained by applying the generic FO result (2) to (4), seemingly ignoring the tweaks in the considered variant of FO); similarly for the quantum bound (6). While the classical reduction (5) is very much tight (up to a small constant factor and a small additive term), the quantum reduction (6) suffers from a square-root loss and a linear-in-q_{RO} factor (plus a worse additive term), while (7) is slightly better, moving the q_{RO} factor into the square-root. Tightness (without any square-root) can be re-established at the cost of introducing and relying on a non-standard and somewhat ad-hoc computational assumption.

We note that some bounds contain a term $\mathbf{Adv}^{\mathsf{PRF}}$, while other bounds have a statistical term that depends on the size of the message space $\mathcal{M}$ of *CPA-Kyber*; this discrepancy comes from the fact that in some cases the hash function that is used for the implicit rejection in the considered FO transform is assumed to be a pseudorandom function, while in other cases it is modeled as a random oracle.

7.2 HQC

The Scheme. Here we consider the KEM *HQC*, which was selected by the NIST for standardization as a backup for *ML-KEM* (which is considered as the main algorithm for general key establishment). *HQC* will be standardized as *HQC-KEM* in the upcoming FIPS 207 standard. The abbreviation *HQC* stands for "Hamming Quasi-Cyclic", and the security of the scheme relies on the hardness of a computational problem in coding theory, on two variants of the *Decisional Quasi-Cyclic Syndrome Decoding* (DQCSD) problem, both with an additional constraint on the *parity* of the syndrome, and one with a truncation on the syndrome vector. We denote the two variants by n-DQCSD-P, and n-DQCSD-PT, respectively, where the parameter n specifies the *code index* of the underlying code.

Design Principle. The scheme is obtained by considering the CPA-secure version *CPA-HQC*, and then applying (a non-standard variant of) the FO transformation to obtain *HQC*.

Simplified, the CPA-version *CPA-HQC* has a very similar syntax as *CPA-Kyber*, but now with objects (and operations) from coding theory (which we do not spell out). The public key consists of code h and a noise codeword s such that $s \approx h \cdot y$ for some sparse y, with y the secret key. The encryption of a message m is given by a pair (u, v), computed such that

$$u \approx h \cdot r \qquad \text{and} \qquad v \approx s \cdot r + Encode(m)$$

for a suitable encoding that allows for error correction. It follows that

$$u \cdot y - v \approx h \cdot r \cdot y - h \cdot y \cdot r + Encode(m) \approx Encode(m)$$

and so m can be recovered by doing error correction.

Again, the actual scheme *CPA-HQC* uses a random seed expansion to produce the randomness. The full-fledged *HQC* is obtained by applying a salted variant of FO transform with implicit rejection (plus other minor tweaks, e.g., again involving a hash of the public-key).

Security Reductions. As shown in [AAB+22], the IND-CPA security of *CPA-HQC* tightly reduces to 2-DQCSD-P and 3-DQCSD-PT as mentioned above, specifically:

$$\mathbf{Adv}^{\mathsf{IND\text{-}CPA}}_{CPA\text{-}HQC}(\mathcal{A}) \leq 2\left(\mathbf{Adv}^{\text{2-DQCSD-P}}(\mathcal{B}) + \mathbf{Adv}^{\text{3-DQCSD-PT}}(\mathcal{C})\right) ,$$

considering *classical* and *quantum* attackers.

The specification [AAB+22] only considers IND-CCA security of *HQC* for a *classical* attacker, concluding the following bound by means of applying a generic FO result (seemingly neglecting the use of the non-standard FO transform):

$$\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{HQC}(\mathcal{A}) \leq 2\left(\mathbf{Adv}^{\text{2-DQCSD-P}}(\mathcal{B}) + \mathbf{Adv}^{\text{3-DQCSD-PT}}(\mathcal{C})\right) + (q_{RO}+q_D)\delta + \frac{3q_{RO}+2^{-\ell}}{|\,\mathcal{M}\,|}\ . \tag{8}$$

The parameter ℓ specifies the length of the salt in the considered salted FO transform, δ is the correctness parameter of *CPA-HQC*, and $\mathcal{M}$ is the (large) message space of *CPA-HQC*. We note that, for the above bound, the hash function used for the implicit rejection in the considered FO transform is modeled as a random oracle, which then gives rise to the two terms that depend on $|\mathcal{M}|$.

No explicit security proof for a *quantum* attacker is provided in the literature, but applying the generic FO bound described in (3) from [JZM19, Theorem 1] (neglecting the fact that a slightly tweaked version is used), one expects to obtain

$$\begin{aligned}&\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{HQC}(\mathcal{A}_*)\\ &\quad\leq 2\sqrt{(q_{RO}+1)\left(\mathbf{Adv}^{\text{2-DQCSD-P}}(\mathcal{B}_*) + \mathbf{Adv}^{\text{3-DQCSD-PT}}(\mathcal{C}_*)\right) + \frac{2(q_{RO}+1)^2}{|\,\mathcal{M}\,|}} + 4q_{RO}\sqrt{\delta} + \frac{2q_{RO}}{\sqrt{|\,\mathcal{M}\,|}}\ .\end{aligned}$$

Brief Discussion The purpose of the salt in the deployed FO transform is unclear: choosing $\ell = 0$ in the above bound on $\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{HQC}(\mathcal{A})$ has no noticeable effect. Nothing better is known for the quantum security than applying the generic FO result (3), again neglecting the tweaks to the FO transform. This generic result incurs a square-root loss and linear-in-$\sqrt{q_{RO}}$ factor.

7.3 FrodoKEM

The Scheme. Here we consider the KEM *FrodoKEM*, which was one of the the round-three candidates in the NIST PQC competition. Eventually, it was not selected for standardization by NIST, however, it is being standardized by ISO. It is based on the hardness of learning with errors (LWE) on *unstructured* lattices and so is recommended by various European authorities (like BSI or ANSSI) as a more conservative alternative to *Kyber*.

Design Principle. The CPA-version (*CPA-Frodo*) is designed similarly to *CPA-Kyber*, but with the considered coefficient ring being integers modulo q, denoted $\mathbb{Z}/q\mathbb{Z}$. In case the encoding of a plaintext does not fit into $\mathbb{Z}/q$, it is split into a vector of $\mathbb{Z}/q\mathbb{Z}$ elements and each component is masked/encrypted separately.

To obtain IND-CCA security, *FrodoKEM* applies a variant of the FO transform that involves nested hashing and implicit rejection.

Security of Reductions. As shown in [NAB+20, Section 5], the IND-CPA security of *FrodoKEM* tightly reduces to the hardness of LWE over the ring $\mathbb{Z}/q\mathbb{Z}$; more formally, it reduces to the hardness of two version of LWE, which we denote by LWE and LWE′, with different concrete parameter choices. The following bound applies to both quantum and classical attackers

$$\mathbf{Adv}^{\mathsf{IND\text{-}CPA}}_{\textit{CPA-Frodo}}(\mathcal{A}) \leq 8\mathbf{Adv}^{\mathsf{LWE}}(\mathcal{B}) + 8\mathbf{Adv}^{\mathsf{LWE}'}(\mathcal{C}) + 2^{-512}\ .$$

Moreover, the LWE assumption used here can be further reduced to a worst-case lattice assumption[10] called *bounded distance decoding with discrete Gaussian samples* (BDDwDGS).

[10]This is contrary to the typical case. In cryptography, hardness assumptions are typically *average-case*, requiring most instances being hard to solve. On the other hand, *worst-case* hardness only requires one instance being hard to solve.

As for the IND-CCA security, similar to the case of *Kyber* and *HQC*, δ-correctness is additionally required. For any classical (resp. quantum) IND-CCA attacker $\mathcal{A}$ (resp. $\mathcal{A}_*$), [NAB+20, Table 3 and Section 5] tells us that

$$\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{FrodoKEM}(\mathcal{A}) \leq \left(\left(3\mathbf{Adv}^{\mathsf{IND\text{-}CPA}}_{CPA\text{-}Frodo}(\mathcal{B}) + q_{RO}\delta + \frac{2q_{RO}+1}{|\mathcal{M}|}\right) C_\alpha\right)^{1-1/\alpha} + \frac{q_{RO}}{|\mathcal{M}|} ,$$

$$\mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{FrodoKEM}(\mathcal{A}_*) \leq 2\sqrt{q_{RO}\cdot\mathbf{Adv}^{\mathsf{IND\text{-}CPA}}_{CPA\text{-}Frodo}(\mathcal{B}_*) + \frac{2(q_{RO}+1)^2}{|\mathcal{M}|}} + 4q_{RO}\sqrt{\delta} + \frac{2q_{RO}}{|\mathcal{M}|} ,$$

where $C_\alpha \leq 3.12$ and $200 \leq \alpha \leq 1000$ are concrete quantities calculated dependent on the parameters of the scheme (which are different for the different NIST levels).

Brief Discussion. The classical bound is of an unusual form; this comes from a so-called Rényi-divergence analysis. It is worthwhile to note that for the considered choices of α the exponent $1-1/\alpha$ above is very close to 1. On the other hand, the quantum bound comes with a square-root loss, which is inherited from generic results.

7.4 Dilithium

The Scheme. Here we consider the signature scheme *Dilithium*, which was selected by the NIST as one of the post-quantum standards and standardized in FIPS 204 under the name *ML-DSA*. *Dilithium* is lattice-based; concretely, the security relies on MLWE and (variants of) MSIS.

Design Principle. The scheme *Dilithium* follows the Fiat–Shamir with aborts (FSwA) design principle.

The underlying Sigma protocol Σ-*Dilithium*, simplified, works as follows. The public-key (Prover and Verifier's input) consists of a matrix A and a vector $t = As_1 + s_2$ over a (suitably structured) coefficient ring, where (s_1, s_2) is the secret key (Prover's input). Prover first sends over $\mathsf{highBits}(w)$, where $w = Ay$ for a random short vector y, and highBits take a subset of bits of its input w. Verifier sends back a small random element c of the ring. Prover then responds with a vector $z := y + c \cdot s_1$ if its coefficients are sufficiently small, and certain condition relevant to the correctness is satisfied, and otherwise aborts. Finally, if Prover has not aborted, Verifier checks whether z is sufficiently short, and whether it can reconstruct $\mathsf{highBits}(w)$ from $Az - ct$ in a specific manner.

Additionally, for the sake of efficiency, the challenge c instead of $\mathsf{highBits}(w)$ is included in the signature.

Security of Reductions. The security of *Dilithium* follows a modular analysis in two steps — reducing the UF-CMA security to UF-NMA, and then proving the UF-NMA security (based on computational assumptions), both in the random oracle model.

The UF-CMA to UF-NMA reduction of *Dilithium* is achieved via applying the generic results in [BBD+23, FFH25]. Formally, the obtained security reduction is of the following form

$$\mathbf{Adv}^{\mathsf{UF\text{-}CMA}}_{Dilithium}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{UF\text{-}NMA}}_{Dilithium}(\mathcal{B}) + \frac{2q_S(q_{RO}+1)\epsilon}{1-p} + \frac{q_S(q_S+1)\epsilon}{2(1-p)^2} + \delta , \tag{9}$$

$$\mathbf{Adv}^{\mathsf{UF\text{-}CMA}}_{Dilithium}(\mathcal{A}_*) \leq \mathbf{Adv}^{\mathsf{UF\text{-}NMA}}_{Dilithium}(\mathcal{B}_*) + \frac{8q_S\sqrt{(q_{RO}+1)\epsilon}}{1-p} + \delta , \tag{10}$$

where p, ϵ, δ are quantities that are computed dependent on the concrete parameters of the scheme and satisfy $0.44 \leq p \leq 0.81$ and $5 \cdot 10^{-499} \leq \epsilon \leq 10^{-132}$ and $2^{-99} \leq \delta \leq 2^{-641}$. Concretely, p is the (worst-case) abort probability and ϵ the (average-case) min-entropy of $\mathsf{highBits}(w)$, and δ is a parameter that can be chosen freely so as to optimize p and ϵ by means of avoiding certain undesirable choises of the key.

As for the UF-NMA property, it is then reduced to the following two problems: MLWE and its vairant SelfTargetMSIS. The following reduction applies both classically and quantum

$$\mathbf{Adv}^{\mathsf{UF\text{-}NMA}}_{Dilithium}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{MLWE}}(\mathcal{B}) + \mathbf{Adv}^{\mathsf{SelfTargetMSIS}}(\mathcal{C}) \ . \tag{11}$$

Since SelfTargetMSIS is a novel computational problem, further justification of its hardness, which we gather below, is desirable. Classically, [KLS18] mentioned that there is a reduction using the forking lemma with $\mathbf{Adv}^{\mathsf{SelfTargetMSIS}}(\mathcal{A}) \lesssim \sqrt{q_{RO}\mathbf{Adv}^{\mathsf{MSIS}}(\mathcal{B})}$, which can then be further improved to $\mathbf{Adv}^{\mathsf{SelfTargetMSIS}}(\mathcal{A}) \lesssim q_{RO}\mathbf{Adv}^{\mathsf{MSIS}}(\mathcal{B})$.[11] For quantum attackers on the other hand, one can reduce the hardness of self-target MSIS to the standard MLWE, with $\mathbf{Adv}^{\mathsf{SelfTargetMSIS}}(\mathcal{A}_*) \lesssim q_{RO}^2\sqrt{\mathbf{Adv}^{\mathsf{MLWE}}(\mathcal{B}_*)}$ (cf. [JMW24, Theorem 2]). With q_{RO} outside of the square-root, the loss is then bigger in the quantum case.

Brief Discussion. In (10), we again see the square-root loss appears in the quantum case. However, this square-root loss has less-prominent effect compared to the case of FO — indeed, the square-root encloses a less dominating additive term that is proportional to ϵ. When choosing concrete parameters, one can make ϵ arbitrarily small via adding salts to the underlying Sigma protocol, at the expense of slightly increasing the signature size, and the randomness complexity.

The hardness of MLWE is rather standard, and SelfTargetMSIS is believed no easier than MSIS on the cryptanalytic-side. This suggests that the aforementioned quantum reduction between the two problems is non-tight.

7.5 Falcon

The Scheme. Here we consider the signature scheme *Falcon*, which was selected by the NIST as one of the post-quantum standards and will be standardized as the *FN-DSA* scheme in FIPS 206. *Falcon* is lattice-based; concretely, it is based on the hardness of the *ring inhomogeneous short integer solution* ($\mathcal{R}$-ISIS).

Design Principle. *Falcon* follows the hash-and-sign design principle. Specifically, it follows the variant put forward by [GPV08] that is based a *preimage sampleable trapdoor function*.

Simplified, the public key of *Falcon* contains of a matrix A over a suitable ring, and the secret key contains a trapdoor td, given which, it is efficient to sample small vectors s and e such that $y = As + e$ for every y. To sign a message m, simply sample such short vectors s, e and a salt r such that $As + e = H(m, r)$ for a hash function H, and produce (r, s, e) as the signature.

In the real *Falcon* scheme, there is a slight chance that the above procedure fails, in which case the vectors s, e has to be resampled until success.

Recently, [GJK24] consider and analyzed a slightly updated version called *Falcon*$^+$, and mentioned that this update has been incorporated into the latest implementation. Crucially, it introduces a tweak that, upon each retry, not only s, e, but also the salt r is being resampled.

Security Reductions. The current provable side of *Falcon* is rather unsatisfactory. The original *Falcon* does not have any meaningful provable security guarantee, classically nor quantum. On the other hand, for *Falcon*$^+$ there is a classical security proof, but the quantum security proof is currently still open.

Falcon$^+$ follows a tailored security analysis. Its classical security is based on a multi-instance version of $\mathcal{R}$-ISIS, which we call n-$\mathcal{R}$-ISIS.[12] The (classical) UF-CMA security proof detailed in [GJK24] gives the following bound

$$\mathbf{Adv}^{\mathsf{UF\text{-}CMA}}_{Falcon^+}(\mathcal{A}) \leq \left(16 \cdot \left(16 \cdot \mathbf{Adv}^{(q_{RO}+1)\text{-}\mathcal{R}\text{-}\mathsf{ISIS}}(\mathcal{B})\right)^{1-1/\alpha_p}\right)^{1-1/\alpha_u} + \sum_{0\leq i\leq q_S}\binom{C_s}{i}p^{C_s-i}(1-p)^i + \frac{q_S(C_s+q_{RO})}{2^{320}} \ ,$$

[11] This is from footnote 7 in [KLS18], which we believe contains a typo. Here we write what we think is correct.

[12] As its name suggests, in the n-$\mathcal{R}$-ISIS problem, an algorithm is given n instances of $\mathcal{R}$-ISIS, and tasked to solve at least one of the instances on its choice.

where $p, r_u, r_p, \alpha_u, \alpha_p$ are quantities depending on the concrete parameters of the scheme; the values satisfy the bounds $C_s \leq q_S(1 + 2^{-14})$, and $72 \leq \alpha_p \leq 584$, and $71 \leq \alpha_u \leq 583$.

Brief Discussion. Similar to the case of *FrodoKEM*, the classical bound is of an unusual form; this again comes from a Rényi-divergence analysis.

On the cryptanalytic side, n-$\mathcal{R}$-ISIS appears to be as hard as $\mathcal{R}$-ISIS. On the provable side, the hardness of n-$\mathcal{R}$-ISIS can be further reduced to that of the standard $\mathcal{R}$-ISIS with $\mathbf{Adv}^{n\text{-}\mathcal{R}\text{-ISIS}}(\mathcal{A}) \lesssim n \cdot \mathbf{Adv}^{\mathcal{R}\text{-ISIS}}(\mathcal{B})$. However, applying this reduction would result in a less tight, and a potentially meaningless bound.

7.6 SPHINCS$^+$

The Scheme. Here we consider the hash-based signature scheme *SPHINCS*$^+$, which was selected by the NIST as one of the post-quantum standards and standardized as the *SLH-DSA* scheme in FIPS 205.

Design Principle. *SPHINCS*$^+$ is a purely hash-based signature scheme. It relies on a Merkle tree construction, where the root acts as the public key of the signature scheme, and each leave is (the hash of) the public key of (a variant of) the hash-based Winternitz one-time signature scheme (*WOTS*$^+$). To sign a message, a leave in the tree is picked pseudorandomly (determined by the message), and then the key-pair associated to this leave is used to sign the message (and the corresponding authentication path to the root is provided). Since constructing the entire tree, as needed to compute the public-key, would be infeasible, the tree is actually organized as a Merkle tree of Merkle trees, where each lower-level tree is attached to a node of the parent tree by a signature (using the key pair associated to the node) on the root of the lower-level tree. This allows to produce the lower-level trees on the fly.

Security Reductions. *SPHINCS*$^+$ is one of the special cases that is provably secure in the plain model, i.e., without any idealization. The following security reduction is from [HK22, Theorem 3]. *SPHINCS*$^+$ can be built from a single cryptographic hash function, however, this hash function is used to instantiate several different function families $PRF, PRF_{\mathsf{msg}}, H_{\mathsf{msg}}, F, H, Th_k, Th_l$. The problems below relate to the collision and pre-image resistance properties of these functions: the *interleaved-target-subset-resilience* (ITSR); and variants for the *single function, multi-target* setting: *undetectability* (SM-UD), *pre-image resistance* (SM-PRE), *target-collision* (SM-TCR), and *decisional second-preimage resistance* (SM-DSPR).

The reduction is then of the following form for both classical and quantum attackers

$$\begin{aligned}\mathbf{Adv}^{\mathsf{UF\text{-}CMA}}_{SPHINCS^+}(\mathcal{A}) \leq\ & \mathbf{Adv}^{\mathsf{PRF}}_{PRF}(\mathcal{B}) + \mathbf{Adv}^{\mathsf{PRF}}_{PRF_{\mathsf{msg}}}(\mathcal{C}) + \mathbf{Adv}^{\mathsf{ITSR}}_{H_{\mathsf{msg}}}(\mathcal{D}) + 16 \cdot \mathbf{Adv}^{\mathsf{SM\text{-}UD}}_{F}(\mathcal{E}) \\ & + \mathbf{Adv}^{\mathsf{SM\text{-}TCR}}_{F}(\mathcal{F}) + \mathbf{Adv}^{\mathsf{SM\text{-}PRE}}_{F}(\mathcal{G}) + \mathbf{Adv}^{\mathsf{SM\text{-}TCR}}_{H}(\mathcal{H}) + \mathbf{Adv}^{\mathsf{SM\text{-}TCR}}_{Th_k}(\mathcal{I}) \\ & + \mathbf{Adv}^{\mathsf{SM\text{-}TCR}}_{Th_l}(\mathcal{J}) + 3 \cdot \mathbf{Adv}^{\mathsf{SM\text{-}TCR}}_{F}(\mathcal{K}) + \mathbf{Adv}^{\mathsf{SM\text{-}DSPR}}_{F}(\mathcal{L})\ .\end{aligned}$$

Brief Discussion. There is no reduction loss in the sense that the overall advantage is bounded by the sum of advantages against these problems, with only a constant-factor blow-up.

Note that the proof presented in the specification [HBD$^+$20, Theorem 9.1] is flawed, and this flaw is later fixed by [HK22, Theorem 3].

8 Reduction Loss versus Bit Security

A formal security proof gives confidence in the soundness of the design of the considered cryptographic scheme. The concrete reduction loss additionally specifies how the parameters for the considered scheme need to be chosen so that breaking the scheme provably contradicts the computational hardness assumption for instances that are (believed to be) *hard in practice*. For example, while factoring is believed to be a hard problem (for classical computers), factoring the specific instance 15 is easy. This in particular means that if the reduction

loss is large then, in principle, the parameters of the scheme would have to be chosen correspondingly large in order to provide a meaningful security guarantee.[13]

In the context of cryptanalysis, the hardness of breaking a cryptographic scheme (e.g., against known attacks), or the hardness of a computational problem, is often expressed in terms of the *bit security*. Formally, the *bit security* of a cryptographic scheme Π or computational problem X is defined as the largest $\kappa \in \mathbb{N}$ such that for any attacker/algorithm $\mathcal{A}$

$$\frac{\mathbf{Adv}(\mathcal{A})}{\mathit{Time}(\mathcal{A})} \leq 2^{-\kappa}\,,$$

where $\mathit{Time}(\mathcal{A})$ is the (possibly expected) running time of $\mathcal{A}$. For example, for the NIST PQC levels 1, 3 and 5, a respective bit security of 128, 192 and 256 is requested for classical attackers (and somewhat smaller number of quantum attackers). We note that in the literature, bit security is often simplified to $\kappa = \log(T)$ with T the running time of the best algorithm that solves the considered problem (almost) with certainty. However, it is silently understood that in that case there is no advantage/running-time trade-off that performs better, i.e., that has a better ratio than $1/T$, which is the case for the commonly used cryptographic schemes.

Note that if the advantage of an algorithm for solving the considered computational problem does not depend on the choice of the problem instance, then the advantage can be amplified to become arbitrarily close to 1 while preserving the same advantage-to-time ratio, simply by rerunning the algorithm. This observation motivates the above simplification. However, this assumption is not always (well) justified, in particular when considering the security of a cryptographic scheme. While we certainly *want* that there are no "weak keys", it is in general not clear that the advantage of the best attacker is key-independent, and per-se it cannot be excluded if the obtained bound on the advantage–time curve is concave.

Below, we translate the effect of different reduction losses from provable security to the bit security. In other words, when the bit security of the underlying computational problem X is estimated to be κ, what can we then conclude on the bit security of the scheme, based on the proven security reduction? We note that this may then well be lower than what is *believed* to be the bit security of the scheme, judged by the known attacks.

Clearly, for a tight reduction $\mathbf{Adv}_{\Pi}(\mathcal{A}) \leq \mathbf{Adv}_{X}(\mathcal{B})$, the bit security remains the same.[14] For a reduction that is tight up to a constant factor, the bit security drops by a few bits. E.g., if $\mathbf{Adv}_{\Pi}(\mathcal{A}) \leq 8\mathbf{Adv}_{X}(\mathcal{B})$ then the bit security κ of X becomes bit security (of at least) $\kappa - 3$ of Π, with the 3 reflecting the factor 2^3 gain in advantage.

For a reduction like $\mathbf{Adv}_{\Pi}(\mathcal{A}) \leq q_{RO}\mathbf{Adv}_{X}(\mathcal{B})$, with a linear loss in the number of random oracle queries (which may be as large as the running time), we obtain

$$\begin{aligned}\frac{\mathbf{Adv}_{\Pi}(\mathcal{A})}{\mathit{Time}(\mathcal{A})} &\leq \min\left(\frac{1}{\mathit{Time}(\mathcal{B})}, \frac{\mathit{Time}(\mathcal{B})\mathbf{Adv}_{X}(\mathcal{B})}{\mathit{Time}(\mathcal{B})}\right)\\ &\leq \min\left(\frac{1}{\mathit{Time}(\mathcal{B})}, \mathit{Time}(\mathcal{B})2^{-\kappa}\right) \leq 2^{-\kappa/2}\,.\end{aligned} \tag{12}$$

Thus, the bit security κ becomes $\kappa/2$, i.e., we lose half the bit security. This for instance applies to the folklore UF-NMA security of the Fiat–Shamir transformation. Interestingly, this is the same loss in bit security as for a reduction of the form $\mathbf{Adv}_{\Pi}(\mathcal{A}) \leq \sqrt{\mathbf{Adv}_{X}(\mathcal{B})}$, where we deduce that

$$\frac{\mathbf{Adv}_{\Pi}(\mathcal{A})}{\mathit{Time}(\mathcal{A})} \leq \frac{\sqrt{\mathbf{Adv}_{X}(\mathcal{B})}}{\mathit{Time}(\mathcal{B})} = \sqrt{\frac{\mathbf{Adv}_{X}(\mathcal{B})}{\mathit{Time}(\mathcal{B})}}\frac{1}{\sqrt{\mathit{Time}(\mathcal{B})}} \leq \frac{2^{-\kappa/2}}{\sqrt{\mathit{Time}(\mathcal{B})}} \leq 2^{-\kappa/2}\,. \tag{13}$$

At this point we would like to stress that even though in both derivations Equation (12) and Equation (13) the final two inequalities are individually tight for a certain choice of $\mathit{Time}(\mathcal{B})$, they are not necessarily

[13] However, this would render the scheme impractical. Thus, it is not uncommon that parameters of the scheme are not chosen in line with the reduction loss, but by considering the best known attack against the scheme. As a matter of fact, for typical schemes the best known attack works by breaking the specific instance specified by the scheme (even though a lossy reduction potentially admits better attacks).

[14] For simplicity, we assume here and below the inequalities between the advantages to hold for attackers $\mathcal{A}$ and $\mathcal{B}$ with *equal* running times, i.e. $\mathit{Time}(\mathcal{A}) = \mathit{Time}(\mathcal{B})$.

tight for the same choice. Thus, in typical cases, the above derived drops in bit security are actually overly conservative. In order to get the precise bit security, as much as can be concluded from the reduction loss, one would have to consider the precise relation between $\mathit{Time}(\mathcal{B})$ and $\mathbf{Adv}_X(\mathcal{B})$; simply having the largest ratio is not sufficient for this purpose.

As far as we understand, this caveat is also the reason for the following oddity. For a reduction like $\mathbf{Adv}_\Pi(\mathcal{A}) \le \sqrt{q_{RO}\mathbf{Adv}_X(\mathcal{B})}$, which is obviously worse than the above, we obtain the same loss in bit security. Namely,

$$\frac{\mathbf{Adv}_\Pi(\mathcal{A})}{\mathit{Time}(\mathcal{A})} \le \min\left(\frac{1}{\mathit{Time}(\mathcal{B})}, \frac{\sqrt{\mathit{Time}(\mathcal{B})\mathbf{Adv}_X(\mathcal{B})}}{\mathit{Time}(\mathcal{B})}\right) \le \min\left(\frac{1}{\mathit{Time}(\mathcal{B})}, 2^{-\kappa/2}\right) \le 2^{-\kappa/2}\,.$$

Finally, the even worse reduction $\mathbf{Adv}_\Pi(\mathcal{A}) \le q_{RO}\sqrt{\mathbf{Adv}_X(\mathcal{B})}$ leads to

$$\frac{\mathbf{Adv}_\Pi(\mathcal{A})}{\mathit{Time}(\mathcal{A})} \le \min\left(\frac{1}{\mathit{Time}(\mathcal{B})}, \frac{\mathit{Time}(\mathcal{B})\sqrt{\mathbf{Adv}_X(\mathcal{B})}}{\mathit{Time}(\mathcal{B})}\right) \le \min\left(\frac{1}{\mathit{Time}(\mathcal{B})}, \sqrt{\mathit{Time}(\mathcal{B})}\,2^{-\kappa/2}\right) \le 2^{-\kappa/3}\,,$$

which again is not tight in general and only captures the worst possible loss in bit security.

In many of the bounds in Section 7, we encounter additional *additive* terms in the reduction losses, terms like $q_{RO}\delta$ or $q_{RO}^2\delta$, with δ being a parameter of the scheme. Clearly, when a bound involves several additive terms, the security is essentially determined by the *largest* of the terms. However, while parameters of the scheme can typically be explicitly computed or well approximated, terms involving $\mathbf{Adv}_X$ can at best be estimated based on best-known algorithms.

It follows that the effect of such an additive term on the bit security is pretty much as follows. For a bound on $\mathbf{Adv}_\Pi(\mathcal{A})$ as considered above but with an additive term $q_{RO}\delta$, the bit security essentially becomes the *minimum* of what is derived above and $\log(1/\delta)$. For example, if $\mathbf{Adv}_\Pi(\mathcal{A}) \le \mathbf{Adv}_X(\mathcal{B}) + q_{RO}\delta$ then

$$\frac{\mathbf{Adv}_\Pi(\mathcal{A})}{\mathit{Time}(\mathcal{A})} \le \frac{\mathbf{Adv}_X(\mathcal{B}) + \mathit{Time}(\mathcal{B})\delta}{\mathit{Time}(\mathcal{B})} \le 2^{-\kappa} + \delta\,,$$

and similarly if the relation between $\mathbf{Adv}_\Pi(\mathcal{A})$ and $\mathbf{Adv}_X(\mathcal{B})$ is differently. Thus, it is necessary that $\log(1/\delta)$ is larger than the required bit security, since otherwise the additive term will become the weak spot. For an additive term $q_{RO}^2\delta$, the bit security becomes the *minimum* of what is derived above and $\frac{1}{2}\log(1/\delta)$. For example, if $\mathbf{Adv}_\Pi(\mathcal{A}) \le \mathbf{Adv}_X(\mathcal{B}) + q_{RO}^2\delta$ then

$$\begin{aligned}\frac{\mathbf{Adv}_\Pi(\mathcal{A})}{\mathit{Time}(\mathcal{A})} &\le \min\left(\frac{1}{\mathit{Time}(\mathcal{B})}, \frac{\mathbf{Adv}_X(\mathcal{B}) + \mathit{Time}(\mathcal{B})^2\delta}{\mathit{Time}(\mathcal{B})}\right) \le \min\left(\frac{1}{\mathit{Time}(\mathcal{B})}, 2^{-\kappa} + \mathit{Time}(\mathcal{B})\delta\right)\\ &\le 2^{-\kappa} + \min\left(\frac{1}{\mathit{Time}(\mathcal{B})}, \mathit{Time}(\mathcal{B})\delta\right) \le 2^{-\kappa} + \sqrt{\delta}\,.\end{aligned}$$

Thus, $\log(1/\delta)$ needs to be at least twice the required bit security.

The above shows how to translate reduction losses into losses in bit security, and how different types of reduction losses impact the resulting security guarantees. However, it turns out that this translation does not always reflect that one reduction is worse than another, and it may suggest a too large bit-security loss. Therefore, a more fine-grained analysis that takes into account the full advantage function is expected to show a different — better — behavior in general. Thus, a direction for future work is to investigate the impact of reduction losses on the concrete security of specific schemes while explicitly taking the full advantage function into account.

9 Afterthoughts and Open Ends

This section collects several afterthoughts and open ends that arise from our discussions. These topics are closely related to our analysis but were either beyond the scope of the present work or naturally emerge as follow-up questions.

9.1 Non-ROM Schemes

Almost all of the schemes considered and analyzed in this document offer a security proof only in the ROM (the exception is *SPHINCS*$^+$). And, indeed, while a decade or two ago this was considered a significant drawback, it has become commonly accepted — if not the standard — to have a security proof in the ROM "only". While there is certainly strong motivation due to the efficiency of such schemes, even on the pure research front there is no push at all towards post-quantum secure cryptographic schemes that do not rely on the random oracle heuristic. In contrast, when relying on computational hardness assumptions like discrete-log or RSA-related ones, which are known to be easy to solve with a quantum computer, reasonably practical encryption and digital signatures schemes are known that have provable security in the plain model (i.e., without relying on the ROM heuristic) — but these are trivially insecure against quantum attacks due to the underlying assumptions.

As mentioned in Section 2.4, a security proof in the ROM provides (strong) evidence for the security of the actual scheme when the ROM is instantiated with a "good" cryptographic hash function, but there is no guarantee. There exists (mostly contrived) examples of cryptographic schemes that are provably secure in the ROM, but totally insecure no matter what hash function is used to instantiate the ROM.

Even though there is no strong concern, certainly from an academic perspective it would be desirable to have alternatives that do not rely on the ROM. But also from a practical perspective, one cannot exclude new, and possibly more concerning, failures of the ROM methodology occurring in the future, which could then affect the general trust in the methodology, and then having a plan B available would certainly be comforting. Schemes that avoid the ROM would then also avoid the difficulties of dealing with a quantum attacker that can make superposition queries to the ROM. Hence, in typical cases, the classical security proof would then carry over, without any additional loss in the reduction.

Cramer and Shoup showed that *hash proof systems* imply IND-CCA-secure public-key encryption [CS02]. Recent research also shows that hash proof systems imply qCCA security [AM24], that is, IND-CCA security against attackers that are allowed to make *quantum* decryption queries. Classically, hash proof systems are easy to construct from discrete log assumptions. Recently, hash proof systems were constructed from *effective cryptographic group actions* [ADMP20], positing that such effective cryptographic group actions can be constructed from isogeny assumptions and thus opening the possibility to construct IND-CCA-secure schemes from post-quantum assumptions. Such a group action was exhibited a few years later [PR23], giving IND-CCA- and qCCA-secure PKE. The practical performance of these schemes is alas not competitive with the other KEM schemes considered in this note.

9.2 Beyond Standard Security

We considered the standard security notions for encryption/encapsulation schemes and for digital signature schemes, formally defined in Section 3. In practice, cryptographic schemes are sometimes used beyond the originally intended purpose, with the risk then being that the considered, and possibly proven, security property is not sufficient. For this reason, the NIST call for additional post-quantum signatures explicitly recommended to incorporate "additional desirable security properties" (like *non-resignability* or *exclusive ownership*). Due to this NIST recommendation, the so-called *BUFF transform* has recently received quite some attention, since it had originally promised to provide these additional security properties for digital signature schemes almost for free — but then it has been shown that the situation is actually much more subtle [DFHS24, DFH$^+$24, FHK26].[15] The purpose of these additional security properties is, e.g., to "bind" signatures to the original public-key. For instance, informally, non-resignability requires that it is hard to turn a signature for an unknown message into a new signature for the same (unknown) message but now for a different public key (even when knowing the corresponding secret key). If signatures are (ab)used in order to provide some kind of "token" (for documents that are not publicly known), non-resignability then ensures that an attacker cannot steal a token and turn it into a token that is then accepted as being his. The BUFF transform achieves this (up to the mentioned subtleties) generically by signing the hash of the message and the public-key.

[15] The subtleties go beyond the scope of this document.

We did not look into these additional security properties in this document. However, we would like to point out that all three KEMs we considered in this document use a non-standard variant of the FO transform that involves hashing the public-key as well. The original motivation for Kyber was to avoid *multi-target* attacks; in effect this binds the session key to the public key (as well as to the ciphertext in case of Kyber). Recently, formal definitions of stronger security notions for KEMs have been developed, allowing for instance for implicit authentication using certain KEMs [CDM24].

9.3 Hybrid-Combiners

In this text, we have considered several post-quantum secure encryption/encapsulation and digital signature schemes. For post-quantum schemes that have been standardized, there is consensus about their security and they are suitable for use in practice. However, some of these schemes are fairly new and especially their implementation security has not undergone the same rigorous research and test of time as the classical schemes have. And even classical schemes like ECC have taken several iterations to decide on secure parameters and instances as the knowledge developed. Therefore, it is likely that at least in some cases *hybrid combiners* will be used to achieve post-quantum security while retaining the guarantees of classical security given by the conventional schemes. Indeed, the use of a combiner ensures that both, the conventional scheme as well as the new (believed-to-be) quantum-secure scheme, need to be successfully broken in order to break the combined scheme.

Combiners for signature schemes are quite straightforward: signing the message or document twice, once with one and once with the other scheme, works well enough for most use cases. However, the problem of constructing combiners for signatures that preserve more advanced security notions is not settled yet.

Constructing combiners for KEMs (or encryption schemes) is more subtle. Naively, one might expect that one can simply add (i.e., xor) the two encapsulated keys, but this is not sufficient to get CCA security. More complicated key-combining functions are necessary, like *split-key PRFs*. In particular when the goal is to combine a conventional with a post-quantum secure scheme, we also want the key-combining function to offer post-quantum security.

References

[AAB+22] Carlos Aguilar-Melchor, Nicolas Aragon, Slim Bettaieb, Loïc Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Edoardo Persichetti, Gilles Zémor, Jurjen Bos, Arnaud Dion, Jerome Lacan, Jean-Marc Robert, and Pascal Veron. HQC. Technical report, National Institute of Standards and Technology, 2022. available at `https://csrc.nist.gov/Projects/post-quantum-cryptography/round-4-submissions`.

[ADMP20] Navid Alamati, Luca De Feo, Hart Montgomery, and Sikhar Patranabis. Cryptographic group actions and applications. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020, Part II*, volume 12492 of *Lecture Notes in Computer Science*, pages 411–439, Daejeon, South Korea, December 7–11, 2020. Springer, Cham, Switzerland.

[AM24] Navid Alamati and Varun Maram. Quantum CCA-secure PKE, revisited. In Qiang Tang and Vanessa Teague, editors, *PKC 2024: 27th International Conference on Theory and Practice of Public Key Cryptography, Part IV*, volume 14604 of *Lecture Notes in Computer Science*, pages 193–226, Sydney, NSW, Australia, April 15–17, 2024. Springer, Cham, Switzerland.

[Bar01] Boaz Barak. How to go beyond the black-box simulation barrier. In *42nd Annual Symposium on Foundations of Computer Science*, pages 106–115, Las Vegas, NV, USA, October 14–17, 2001. IEEE Computer Society Press.

[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 Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology – CRYPTO 2023, Part V*, volume 14085 of *Lecture Notes in Computer Science*, pages 358–389, Santa Barbara, CA, USA, August 20–24, 2023. Springer, Cham, Switzerland.

[BBH+19] James Bartusek, Liron Bronfman, Justin Holmgren, Fermi Ma, and Ron D. Rothblum. On the (in)security of kilian-based SNARGs. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019: 17th Theory of Cryptography Conference, Part II*, volume 11892 of *Lecture Notes in Computer Science*, pages 522–551, Nuremberg, Germany, December 1–5, 2019. Springer, Cham, Switzerland.

[BCP03] Emmanuel Bresson, Olivier Chevassut, and David Pointcheval. The group Diffie-Hellman problems. In Kaisa Nyberg and Howard M. Heys, editors, *SAC 2002: 9th Annual International Workshop on Selected Areas in Cryptography*, volume 2595 of *Lecture Notes in Computer Science*, pages 325–338, St. John's, Newfoundland, Canada, August 15–16, 2003. Springer Berlin Heidelberg, Germany.

[Ble98] Daniel Bleichenbacher. Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1. In Hugo Krawczyk, editor, *Advances in Cryptology – CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 1–12, Santa Barbara, CA, USA, August 23–27, 1998. Springer Berlin Heidelberg, Germany.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *ACM CCS 93: 1st Conference on Computer and Communications Security*, pages 62–73, Fairfax, Virginia, USA, November 3–5, 1993. ACM Press.

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd Annual Symposium on Foundations of Computer Science*, pages 136–145, Las Vegas, NV, USA, October 14–17, 2001. IEEE Computer Society Press.

[CDM24] Cas Cremers, Alexander Dax, and Niklas Medinger. Keeping up with the KEMs: Stronger security notions for KEMs and automated analysis of KEM-based protocols. In Bo Luo, Xiaojing Liao, Jun Xu, Engin Kirda, and David Lie, editors, *ACM CCS 2024: 31st Conference on Computer and Communications Security*, pages 1046–1060, Salt Lake City, UT, USA, October 14–18, 2024. ACM Press.

[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 Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021, Part II*, volume 12697 of *Lecture Notes in Computer Science*, pages 598–629, Zagreb, Croatia, October 17–21, 2021. Springer, Cham, Switzerland.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *30th Annual ACM Symposium on Theory of Computing*, pages 209–218, Dallas, TX, USA, May 23–26, 1998. ACM Press.

[CS02] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars R. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 45–64, Amsterdam, The Netherlands, April 28 – May 2, 2002. Springer Berlin Heidelberg, Germany.

[DFH+24] Jelle Don, Serge Fehr, Yu-Hsuan Huang, Jyun-Jie Liao, and Patrick Struck. Hide-and-seek and the non-resignability of the BUFF transform. In Elette Boyle and Mohammad Mahmoody, editors, *TCC 2024: 22nd Theory of Cryptography Conference, Part III*, volume 15366 of *Lecture Notes in Computer Science*, pages 347–370, Milan, Italy, December 2–6, 2024. Springer, Cham, Switzerland.

[DFHS24] Jelle Don, Serge Fehr, Yu-Hsuan Huang, and Patrick Struck. On the (in)security of the BUFF transform. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology – CRYPTO 2024, Part I*, volume 14920 of *Lecture Notes in Computer Science*, pages 246–275, Santa Barbara, CA, USA, August 18–22, 2024. Springer, Cham, Switzerland.

[DFM20] Jelle Don, Serge Fehr, and Christian Majenz. The measure-and-reprogram technique 2.0: Multi-round fiat-shamir and more. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020, Part III*, volume 12172 of *Lecture Notes in Computer Science*, pages 602–631, Santa Barbara, CA, USA, August 17–21, 2020. Springer, Cham, Switzerland.

[DFMS19] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Security of the Fiat-Shamir transformation in the quantum random-oracle model. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part II*, volume 11693 of *Lecture Notes in Computer Science*, pages 356–383, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Cham, Switzerland.

[DFMS22] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Online-extractability in the quantum random-oracle model. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology – EUROCRYPT 2022, Part III*, volume 13277 of *Lecture Notes in Computer Science*, pages 677–706, Trondheim, Norway, May 30 – June 3, 2022. Springer, Cham, Switzerland.

[FFH25] Pouria Fallahpour, Serge Fehr, and Yu-Hsuan Huang. Tighter quantum security for Fiat-Shamir-with-aborts and hash-and-sign-with-retry signatures. Cryptology ePrint Archive, Paper 2025/985, 2025.

[FHK26] Serge Fehr, Yu-Hsuan Huang, and Julia Kastner. Sandwich BUFF: Achieving non-resignability using iterative hash functions. In Benny Applebaum and Huijia (Rachel) Lin, editors, *Theory of Cryptography*, pages 235–265, Cham, 2026. Springer Nature Switzerland.

[GHHM21] Alex B. Grilo, Kathrin Hövelmanns, Andreas Hülsing, and Christian Majenz. Tight adaptive reprogramming in the QROM. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2021, Part I*, volume 13090 of *Lecture Notes in Computer Science*, pages 637–667, Singapore, December 6–10, 2021. Springer, Cham, Switzerland.

[GJK24] Phillip Gajland, Jonas Janneck, and Eike Kiltz. A closer look at falcon. Cryptology ePrint Archive, Report 2024/1769, 2024.

[GK03] Shafi Goldwasser and Yael Tauman Kalai. On the (in)security of the Fiat-Shamir paradigm. In *44th Annual Symposium on Foundations of Computer Science*, pages 102–115, Cambridge, MA, USA, October 11–14, 2003. IEEE Computer Society Press.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th Annual ACM Symposium on Theory of Computing*, pages 197–206, Victoria, BC, Canada, May 17–20, 2008. ACM Press.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *28th Annual ACM Symposium on Theory of Computing*, pages 212–219, Philadephia, PA, USA, May 22–24, 1996. ACM Press.

[HBD+20] Andreas Hülsing, Daniel J. Bernstein, Christoph Dobraunig, Maria Eichlseder, Scott Fluhrer, Stefan-Lukas Gazdag, Panos Kampanakis, Stefan Kölbl, Tanja Lange, Martin M. Lauridsen, Florian Mendel, Ruben Niederhagen, Christian Rechberger, Joost Rijneveld, Peter Schwabe, Jean-Philippe Aumasson, Bas Westerbaan, and Ward Beullens. SPHINCS$^+$. Technical report, National Institute of Standards and Technology, 2020. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions`.

[HHK17] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017: 15th Theory of Cryptography Conference, Part I*, volume 10677 of *Lecture Notes in Computer Science*, pages 341–371, Baltimore, MD, USA, November 12–15, 2017. Springer, Cham, Switzerland.

[HK22] Andreas Hülsing and Mikhail A. Kudinov. Recovering the tight security proof of SPHINCS$^+$. In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology – ASIACRYPT 2022, Part IV*, volume 13794 of *Lecture Notes in Computer Science*, pages 3–33, Taipei, Taiwan, December 5–9, 2022. Springer, Cham, Switzerland.

[HM24] Kathrin Hövelmanns and Christian Majenz. A note on failing gracefully: Completing the picture for explicitly rejecting Fujisaki-Okamoto transforms using worst-case correctness. In Markku-Juhani Saarinen and Daniel Smith-Tone, editors, *Post-Quantum Cryptography - 15th International Workshop, PQCrypto 2024, Part II*, pages 245–265, Oxford, UK, June 12–14, 2024. Springer, Cham, Switzerland.

[JMW24] Kelsey A. Jackson, Carl A. Miller, and Daochen Wang. Evaluating the security of CRYSTALS-Dilithium in the quantum random oracle model. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology – EUROCRYPT 2024, Part VI*, volume 14656 of *Lecture Notes in Computer Science*, pages 418–446, Zurich, Switzerland, May 26–30, 2024. Springer, Cham, Switzerland.

[JZC+18] Haodong Jiang, Zhenfeng Zhang, Long Chen, Hong Wang, and Zhi Ma. IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part III*, volume 10993 of *Lecture Notes in Computer Science*, pages 96–125, Santa Barbara, CA, USA, August 19–23, 2018. Springer, Cham, Switzerland.

[JZM19] Haodong Jiang, Zhenfeng Zhang, and Zhi Ma. Tighter security proofs for generic key encapsulation mechanism in the quantum random oracle model. In Jintai Ding and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019*, pages 227–248, Chongqing, China, May 8–10, 2019. Springer, Cham, Switzerland.

[KL07] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. Chapman and Hall/CRC Press, first edition, 2007.

[KLS18] Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner. A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part III*, volume 10822 of *Lecture Notes in Computer Science*, pages 552–586, Tel Aviv, Israel, April 29 – May 3, 2018. Springer, Cham, Switzerland.

[KRS25] Dmitry Khovratovich, Ron D. Rothblum, and Lev Soukhanov. How to prove false statements: Practical attacks on fiat-shamir. In Yael Tauman Kalai and Seny F. Kamara, editors, *Advances in Cryptology – CRYPTO 2025, Part VI*, volume 16005 of *Lecture Notes in Computer Science*, pages 3–26, Santa Barbara, CA, USA, August 17–21, 2025. Springer, Cham, Switzerland.

[Mau05] Ueli M. Maurer. Abstract models of computation in cryptography (invited paper). In Nigel P. Smart, editor, *10th IMA International Conference on Cryptography and Coding*, volume 3796 of *Lecture Notes in Computer Science*, pages 1–12, Cirencester, UK, December 19–21, 2005. Springer Berlin Heidelberg, Germany.

[NAB+20] Michael Naehrig, Erdem Alkim, Joppe Bos, Léo Ducas, Karen Easterbrook, Brian LaMacchia, Patrick Longa, Ilya Mironov, Valeria Nikolaenko, Christopher Peikert, Ananth Raghunathan,

and Douglas Stebila. FrodoKEM. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions.

[NIS16] Stateless Hash-Based Digital Signature Standard. https://csrc.nist.gov/News/2016/Public-Key-Post-Quantum-Cryptographic-Algorithms, December 2016.

[NIS22] Stateless Hash-Based Digital Signature Standard. https://csrc.nist.gov/News/2022/pqc-candidates-to-be-standardized-and-round-4, July 2022.

[NIS24a] Module-Lattice-Based Digital Signature Standard. National Institute of Standards and Technology, NIST FIPS PUB 204, U.S. Department of Commerce, August 2024.

[NIS24b] Module-Lattice-Based Key-Encapsulation Mechanism Standard. National Institute of Standards and Technology, NIST FIPS PUB 203, U.S. Department of Commerce, August 2024.

[NIS24c] Stateless Hash-Based Digital Signature Standard. National Institute of Standards and Technology, NIST FIPS PUB 205, U.S. Department of Commerce, August 2024.

[PR23] Aurel Page and Damien Robert. Introducing clapoti(s): Evaluating the isogeny class group action in polynomial time. Cryptology ePrint Archive, Report 2023/1766, 2023.

[SAB+20] Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, and Damien Stehlé. CRYSTALS-KYBER. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions.

[Sho94] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th Annual Symposium on Foundations of Computer Science*, pages 124–134, Santa Fe, NM, USA, November 20–22, 1994. IEEE Computer Society Press.

[Sho97a] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

[Sho97b] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *Advances in Cryptology – EUROCRYPT'97*, volume 1233 of *Lecture Notes in Computer Science*, pages 256–266, Konstanz, Germany, May 11–15, 1997. Springer Berlin Heidelberg, Germany.

[Sho04] Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. Cryptology ePrint Archive, Report 2004/332, 2004.

[Unr12] Dominique Unruh. Quantum proofs of knowledge. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 135–152, Cambridge, UK, April 15–19, 2012. Springer Berlin Heidelberg, Germany.

[Zha19] Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part II*, volume 11693 of *Lecture Notes in Computer Science*, pages 239–268, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Cham, Switzerland.