Post-Quantum Cryptography: Computational-Hardness Assumptions and Beyond

Thomas Attema[1,2,3], Nicole Gervasoni[1], Michiel Marcus[1], and Gabriele Spini[1,4]

[1]TNO, Cyber Security and Robustness, The Hague, The Netherlands
[2]CWI, Cryptology Group, Amsterdam, The Netherlands
[3]Leiden University, Mathematical Institute, Leiden, The Netherlands
[4]e-mail: gabriele.spini@tno.nl

April 30, 2021

Abstract

The advent of a full-scale quantum computer will severely impact most currently-used cryptographic systems. The most well-known aspect of this impact lies in the computational-hardness assumptions that underpin the security of most current public-key cryptographic systems: a quantum computer can factor integers and compute discrete logarithms in polynomial time, thereby breaking systems based on these problems.

However, simply replacing these problems by other which are (believed to be) impervious even to a quantum computer does not completely solve the issue. Indeed, many security proofs of cryptographic systems are no longer valid in the presence of a quantum-capable attacker; while this does not automatically implies that the affected systems would be broken by a quantum computer, it does raises questions on the exact security guarantees that they can provide.

This overview document aims to analyze all aspects of the impact of quantum computers on cryptographic, by providing an overview of current quantum-hard computational problems (and cryptographic systems based on them), and by presenting the security proofs that are affected by quantum-attackers, detailing what is the current status of research on the topic and what the expected effects on security are.

1 Introduction

In this document, we elaborate on the changes that need to be considered when a quantum adversary arises in the context of cryptography. Research in this area has sparked since the publication of Shor's quantum algorithm [Sho99], which breaks almost all currently-deployed *asymmetric* cryptographic systems. A general misconception is that simply swapping the mathematical problems underlying a cryptographic system for *quantum-safe*[1] mathematical problems is enough to make the system quantum-safe. Unfortunately, this is not true. All mathematical proofs of security have to be interpreted in a certain context and adversarial model. A quantum adversary has properties that are not modeled in classical proofs. Therefore, in addition to using quantum-safe mathematical problems, the proofs themselves often need to be altered as well. These altered proofs often have impact on the security level that is attained against quantum adversaries. Thus there are two ways in which changes need to be made in the presence of a quantum adversary:

1. Find mathematical problems for which there are no efficient quantum algorithms to solve them, and base the security of cryptographic protocols on the hardness of these problems.

[1] quantum-safe problems are problems that are believed to be unsolvable for quantum-polynomial-time algorithms.

2. Reformulate the adversarial models, such that the unique characteristics of a quantum adversary are accounted for, and prove security in these new models.

In addition to Shor's algorithm, there is another quantum algorithm that affects currently-deployed cryptography, but in a less severe way. Grover's algorithm [Gro96] attains quadratic speed-up against *symmetric* cryptography, as opposed to Shor's algorithm, which attains exponential speed-up against *asymmetric* cryptography. As the quadratic speed-up can easily be counteracted by doubling the key-length (e.g., going from AES-128 to AES-256), this algorithm does not *break* symmetric cryptography. This was a widely accepted conclusion, so no further research into quantum adversaries against symmetric primitives was done for a long time. However, the work by Zhandry [Zha12] has renewed interest in quantum cryptanalysis of symmetric primitives.

We start with a general overview of important aspects of provable security. We then focus on asymmetric cryptography in the presence of quantum adversaries. More specifically, we provide an overview of mathematical problems that are conjectured to be hard to solve for both quantum computers and classical computers and afterwards provide an overview of research that reasons about the validity of classical proofs in a quantum setting and possible modifications that are required for these proofs. Lastly, we provide an overview of research that reasons about the security of symmetric primitives in the presence of a quantum adversary.

2 Types of Security Proofs

Reasoning about the security of a cryptographic primitive is not a trivial task. A very naive way to design a cryptographic system would be to go through the following steps:

1. Create a functional design and try to attack it. If the designer cannot find any efficient attacks, the scheme is deployed for use.
2. Wait until someone finds a better attack that breaks the system.
3. Change the system to prevent said attack or change the recommended parameters.
4. Go to step 2.

If no new attack is published in reasonable time, one might assume that the cryptographic scheme is secure, if it can withstand the best known attack. This generally means that it takes in the order of 2^λ operations to break the system with said attack, where λ denotes the computational security parameter.[2] However, it is unclear what "reasonable time" means. We could start with 5 years. However, there are cryptographic systems that were broken after 5 years of silence, such as those used in the PKCS #1 family of standards [Kal98]. PKCS #1 version 1.5 contains a padding protocol for RSA that was standardized in 1993. It was not until 5 years later, in 1998, that a chosen-ciphertext attack was found against PKCS #1 version 1.5 by Bleichenbacher [Ble98]. Therefore, this is not a satisfactory method.

To make more meaningful statements about the security of a scheme, a definition of security needs to be in place. Such a definition should specify how we model an attacker and what the objective of the attacker is. The general aim is to show that an attacker that can break the system, can also solve some well-studied mathematical problem without much extra effort. The security of the system is now *reduced* to the hardness of a mathematical problem. More formally, such a reduction is proved as follows.

- Assume we have a probabilistic polynomial-time algorithm $\mathcal{A}$ that can compromise a certain security goal of the scheme in time T, given certain powers. Here T is polynomial in λ.
- Create an algorithm $\mathcal{B}$ (possibly probabilistic) that, given $\mathcal{A}$, can solve the mathematical problem in time $f(T)$ for some function f.

[2] λ is usually either 128 or 256.

We call this a *security reduction.* If the reduction precisely models the powers of an adversary, such a reduction implies that the adversary must attack either the implementation of the scheme, or the underlying mathematical assumption to compromise the security goal. Ideally, $f(T) \approx T$, in which case it is shown that breaking the cryptographic system takes approximately as much time as solving the mathematical problem. By contrast, if $f(T) \gg T$, then breaking the cryptographic system might be significantly easier than solving the mathematical problem. How close $f(T)$ is to T is referred to as the *tightness* of the reduction. Basing the security of a cryptographic scheme on a non-tight reduction, e.g., $f(T) = T^2$, might result in overly conservative parameter choices and impractical cryptographic protocol instantiations. However, these reductions do show that there is no structural weakness in the cryptographic system.

There are two main types of proofs that are often used to reason about security of cryptographic schemes and protocols that use them. The first type is a *game-based proof* and the second type is a *simulation-based proof.* In game-based proofs, security is defined as the inability of an adversary to win a certain security game, such as distinguishing two ciphertexts. In simulation-based proofs, security is defined as the ability to simulate the behavior of a cryptographic system without knowing its secret input values.

We start by listing a few advantages of simulation-based proofs:

- It is often clear what kind of guarantees the proof provides.
- Simulation-based proofs provide security under sequential composition with arbitrary other protocols, whereas game-based proofs only provide sequential composition of the same proofs.

The latter advantage is important for cryptographic protocols and primitives that are used as building blocks in larger cryptographic systems. Proofs for protocols are therefore generally simulation-based. In comparison, a few advantages of game-based proofs are:

- It is relatively easy to write game-based proofs, since it is well-known how to set up such games and how to connect them to a hard mathematical problem.
- Game-based proofs generally hold in the standard model, which has the weakest assumptions[3].

We elaborate on both types of security proofs below.

2.1 Game-Based Security Proofs

In game-based security proofs, there are two parties: A *challenger* and an *adversary.* There are different game-based notions for public-key encryption schemes (PKE) / key-encapsulation mechanisms (KEM) and signature schemes. We start with an overview of security games for PKE schemes and KEMs and cover security games for signature schemes afterwards.

2.1.1 PKE/KEM Security Games

For PKE schemes and KEMs, there are four desired properties (informal):

- Unbreakability (UB): No attacker can efficiently retrieve the secret key from the public key
- One-wayness (OW): No attacker can efficiently invert the encryption function on a given ciphertext
- Indistinguishability (IND): No attacker can efficiently distinguish encryptions of plaintexts of equal size.
- Non-Malleability (NM): No attacker can efficiently transform ciphertexts into new ciphertexts of which the plaintext is a function over the other plaintexts.

[3] This means that no extra assumptions other than the conjecture regarding the underlying problem are necessary for the proof to hold.

We can model three types of attacks:

- Chosen-Plaintext Attack (CPA): An attacker has access to the public key and records ciphertexts of which the plaintexts need to remain secret. He then chooses a plaintext to decrypt.
- Non-Adaptive Chosen Ciphertext Attack (CCA1): An attacker has access to the public key and decryptions of selected ciphertexts and tries to decrypt chosen ciphertexts.
- Adaptive Chosen-Ciphertext Attack (CCA2): An attacker has access to the public key and decryptions of selected ciphertexts. Additionally, after seeing the ciphertexts the adversary is interested in, it retains the ability to find decryptions of selected ciphertexts, apart from the target ciphertext.

The desired properties are listed in increasing strength. This means that for a given attack model, Non-Malleability implies Indistinguishability, Indistinguishability implies One-wayness and One-wayness implies Unbreakability. Additionally, the attack models are listed in increasing strength, such that CCA2 implies CCA1 and CCA1 implies CPA for a given security property.

As an example, we show the security game for IND-CPA security, which stands for *indistinguishability under chosen plaintext attack*. In this game, there are five stages [KL14].

1. The challenger generates a public-private key pair of the scheme and sends the public key to the adversary.
2. The adversary can use the public key a polynomial number of times[4] to encrypt plaintexts of their choice.
3. The adversary chooses two messages m_0 and m_1 of equal size and sends them to the challenger.
4. The challenger decides to either encrypt m_0 or m_1 and sends the encryption to the adversary.
5. The adversary wins the game if it can correctly guess which message was encrypted.

For IND-CCA1 security, the adversary additionally has access to a decryption oracle in step 2. For IND-CCA2 security, the adversary always has access to a decryption oracle. When IND-CCA is referenced without a number (1 or 2), generally IND-CCA2 is meant. It has been proven that IND-CCA2 and NM-CCA2 are equivalent [BDPR98]. Therefore, a proof for IND-CCA2 is sufficient to prove indistinguishability and non-malleability against adaptive chosen-ciphertext attacks. A security proof using one of these security games generally shows that if the adversary has a non-negligible advantage in distinguishing the two messages, the adversary can solve a mathematical problem that is conjectured to be hard.

2.1.2 Signature Scheme Security Games

For signature schemes, other security notions have been specified. There are three desired properties (informal):

- Unbreakability (UB) - No attacker can efficiently retrieve the secret key from the public key.
- Universal Unforgeability (UUF) - No attacker can efficiently create a valid signature of any message that has not been signed yet.
- Existential Unforgeability (EUF) - No attacker can efficiently create a valid signature of a chosen message that has not been signed yet.

We can model three types of attacks:

- Key-Only Attack (KOA) - An attacker has access to the public key.

[4] Polynomial in the security parameter λ.

- Known-Message Attack (KMA) - An attacker has access to the public key and a number of signatures for known messages.
- Adaptive Chosen-Message Attack (CMA) - An attacker has access to the public key, a number of signatures for known messages, and an oracle that allows the attacker to sign any messages adaptively except the target message. The adaptive part means that the behavior of the attacker can change based on outputs throughout the attack.

Once again, both the desired properties and attack models are listed in increasing strength, such that lower definitions imply all definitions above it. Consider the UUF-CMA security game. In this game, there are five stages [KL14].

1. The challenger generates a public-private key pair of the scheme and sends the public key to the adversary.
2. The adversary can use the public key a polynomial number of times[5] to verify signatures of their choice and use the decryption oracle a polynomial number of times[6] to sign messages of their choice.
3. The challenger sends a challenge message m.
4. The adversary sends a signature σ for m.
5. The adversary wins the game if the verification of σ for m is accepted.

A security proof then shows that, if the provided signature is accepted for the message m with non-negligible probability, some hard problem can be solved.

The security games for public key encryptions schemes and signature schemes have a specific assumption on the powers of the adversary, namely that encryption and decryption (or signing and verification) are modeled in a 'black-box' manner, so the adversary has no access to intermediate states of the scheme. However, through side-channel attacks, an adversary might have access to such states. As side-channel attacks generally only rely on bad implementations, this assumption is widely accepted and implementations are carefully assessed in order to resist side-channel attacks. Nowadays, resistance against IND-CCA2 is the required standard for any serious encryption scheme and EUF-CMA is the standard for signature schemes, both of which are the strongest security notions for the respective cryptographic application [NIS16].

2.2 Simulation-Based Proofs

A different approach to proving certain security aspects are simulation-based proofs. In these types of proofs, two settings are compared: An ideal world and the real world. The proof then shows that these worlds are equal in the eyes of the adversary by showing that any action the adversary initiated in the real world, can be simulated in the ideal world. An example with respect to public-key encryption schemes is Semantic Security against Chosen Plaintext Attacks (SS-CPA).

Informally, perfect semantic security says that whatever an adversary can learn about a message, given its encryption and length, can also be learnt by an adversary that only receives the length of the message. For computational semantic security, we get that the probability that an adversary, who receives an encryption and the length of the message it encrypts, can learn a certain function $f(m)$ over the encrypted message m, is negligibly close to the probability that an adversary, who only receives the length of the encrypted message, can learn $f(m)$. A computational Semantic Security proof under Chosen Plaintext Attack (CPA) therefore has the following setting.

There is a message m that is encrypted to some ciphertext c. Let $\mathcal{A}$ be an algorithm that computes $f(m)$ for some function f over the message, given c, the length of the message $|m|$, and access to an encryption

[5] Polynomial in the security parameter λ.

[6] Polynomial in the security parameter λ.

oracle on arbitrary messages. The proof then has to construct an adversary $\mathcal{A}^*$ with the same computational powers, who only receives $|m|$, such that for all functions f over the message space, we get that

$$|Pr[\mathcal{A}(|m|, c) = f(m)] - Pr[\mathcal{A}^*(|m|) = f(m)]| \leq \epsilon,$$

where ϵ is negligible in the security parameter. Usually, $\mathcal{A}^*$ is constructed by first picking a value for c and then running $\mathcal{A}$ with the chosen c and $|m|$ as a subroutine to achieve such a proof. As the difference in probabilities between an adversary with and an adversary without the ciphertext is negligible, the conclusion is that a negligible amount of information about the plaintext can be extracted from the ciphertext.

In some cases, simulation proofs have been shown to be equivalent to game-based proofs. More specifically, *SS* gives the same guarantees as its counterparts in *IND*, so SS-CPA = IND-CPA [GM82], SS-CCA1 = IND-CCA1 [WSI03] and SS-CCA2 = IND-CCA2 [WSI03]. However, this equivalence does not hold in general. There are many other flavors of simulation proofs as well, such as zero knowledge proofs of knowledge and proofs for protocols. In general, simulation-based proofs give the guarantee that security holds under sequential composition of *arbitrary* protocols, which is important for the security of protocols.

2.3 Proof Models

In the ideal situation, the security of a scheme only depends on some mathematical problem that is conjectured to be hard to solve. If this is the case, we say that it is secure in the *standard model*. Proofs that attain security in the standard model are generally the most robust proofs. All other models have additional assumptions.

A popular alternative model is the *random oracle model* (ROM). We briefly touch upon it here, but a more detailed explanation is given in section 4. In short, the random oracle model assumes that all parties have access to an idealized version of a hash function, called the random oracle. This oracle has certain special properties. The ideal world is a world where there is access to such a random oracle. Even though there is proof that we cannot instantiate a random oracle in the real world [MRH04], the hope is that the proof does not break when a hash function is used instead. At the very least, a proof in the random oracle model provides the confidence that an efficient attack does not abuse a design flaw in the scheme, but a property of the hash function that is used. If such an attack arises, the scheme does not necessarily need to be changed, but a new hash function needs to be used that better emulates the random oracle. The random oracle model is therefore seen as a hybrid model between the rigorous proofs from the standard model and no proof at all. The schemes that can be proven secure in the random oracle model are generally more efficient than those that can be proven secure in the standard model.

The standard model and ROM are the most well-known models, but other models exist, such as the *generic group model* [Sho97], *common refererence string model* [CPS07] and the *public key infrastructure model* [Ped05]. An analysis in the generic group model can answer the question "What is the fastest generic algorithm for breaking a cryptographic hardness assumption?". The common reference string model captures the assumption that a trusted setup exists in which all involved parties get access to the same string taken from a certain distribution. Schemes proven secure in this model are secure given that the setup was performed correctly. The public key infrastructure model assumes that there is some trusted third party called the *certificate authority*, who provides digital certificates.

2.4 Security Under Parallel Composition

As mentioned before, regular game-based and simulation-based proofs provide security under sequential composition with the same protocol (game-based) or arbitrary protocols (simulation-based). However, this does not guarantee that certain protocols are secure when they are initiated in *parallel*. There are two frameworks that guarantee security under parallel composition: the Universal Composability (UC) framework and the Indifferentiability framework. In essence, they both provide the same guarantees, namely that if a larger process initiates another process as a sub-routine, that the behavior of the larger process changes negligibly if the sub-process is exchanged for some ideal functionality. The two frameworks take very different approaches towards formalizing systems and sub-systems, but in principle a proof in one can also be written as a proof in the other. However, the way that processes are formalized does influence how easy it is to write a proof for a certain process. For example, it is quite natural to write a proof for a protocol between different parties in the UC framework, whereas proofs for cryptographic primitives, such as hash functions, can be written more easily in the Indifferentiability framework.

2.4.1 The Universal Composability Framework

The Universal Composability (UC) framework was introduced by Canetti in 2001 [Can01]. If a real sub-protocol behaves like an ideal sub-protocol in the UC framework, then the real sub-protocol can always be substituted for the ideal sub-protocol in the security analysis of the outer protocol, regardless of what this protocol does. This is a very powerful tool to extend proofs for complex protocols. There are two flavors of this protocol: one is statistical UC and the other is computational UC. The first provides statistical security, so attackers can be computationally unbounded, and the latter provides computational security, in which case only probabilistic polynomial-time attackers are considered.

The way this framework is formalized, is by modeling all involved systems as *Interactive Turing Machines* (ITMs), which are abstract models of computation that can simulate any computer algorithm that communicates with other systems (additionally modeled as ITMs). The UC framework guarantees the parallel composability property by introducing an environment ITM, which initiates adversaries (modeled as ITMs), a protocol π between other ITMs, and observes outputs. The goal of a proof in the UC framework is to show that for all possible environments and all possible adversaries, we can find a simulator such that an execution of the environment with the adversary and sub-protocol ρ behaves almost identically[7] to an execution of the environment with the simulator and ideal sub-protocol ϕ. Given the fact that a simulator on the ideal sub-protocol ϕ has less power than the adversary on the real sub-protocol ρ, this shows that any attack on real sub-protocol ρ would also work on sub-protocol ϕ, indicating that the real protocol emulates the ideal protocol.

This framework is very suitable for proving the security of protocols in general, but it is perhaps even more relevant for the area of Multi-Party Computation (MPC). In MPC, protocols are created that should have the same functionality and guarantees that a trusted third party provides, without actual access to a trusted third party. We would therefore like to show that the protocol emulates a trusted third party.

2.4.2 Universal Composability and Quantum Adversaries

With regard to quantum security, theorem 2 of [Unr10] proves that security proofs in the statistical UC framework still hold in a quantum setting. More concretely, if π is a classical protocol that statistically UC-emulates a certain classical functionality F, then π statistically quantum-UC-emulates F. This means that proofs in the statistical UC framework still hold against quantum adversaries, provided that the underlying primitives are quantum-safe. However, it is not generally true that classical statistical indistinguishability implies quantum statistical indistinguishability.

2.4.3 Indifferentiability

The Indifferentiability framework was introduced in 2004 by Maurer et al. [MRH04]. In the Indifferentiability framework, systems are modeled as conditional probability distributions with inputs and outputs. It is still possible to model systems in this framework, because the output of computer algorithm can also be modeled as conditional probability distributions conditioned on the input. To capture the nature of cryptographic systems, the Indifferentiability framework models the input through two channels: a private channel and a public channel.

[7] Formally, we analyze the probabilities that the output of the adversary is 0 or 1 and these probabilities should be negligibly close in both scenarios.

The difference between *indistinguishability* and *indifferentiability* is subtle, but indistinguishability says that for a real system C', an ideal system C, and for every system $\mathcal{D}$, called the distinguisher, the distinguisher behaves almost identically (e.g., the probabilities that it outputs either 0 or 1 are negligibly close) if:

1. The distinguisher has no access to the input interface of C', but observes the output interface of C',
2. The distinguisher has no access to the input interface of C, but observes the output interface of C.

A proof of indistinguishability is enough when we want to substitute ideal system C for C', under the following assumptions:

1. No external party can influence the behaviour of C,
2. No external party has access to the randomness of C.

Such assumptions are acceptable for keyed primitives, as long as the key is not known, because a keyed primitive is essentially a random primitive drawn from a distribution of primitives with deterministic behaviour. However, for other applications such assumptions are not reasonable, which is the case for hash functions which essentially are primitives with fixed behaviour. If we want to use a hash function instead of a random oracle, indistinguishability is not sufficient.

The general description of indifferentiability as given above is still rather vague. Since the indifferentiability framework is generally used to prove implementations of hash functions indifferentiable from random oracles, we explain how to prove a hash function indifferentiable from a random oracle. Generally, hash functions use mechanisms that are based on other primitives. For example, SHA-3 is built using a sponge construction, which uses a compression function as a primitive. In the following, we denote by $\mathcal{C}^{\mathcal{F}}$ the outer construction — e.g., sponge construction — and by $\mathcal{F}$ the ideal version of the inner construction, such as an *ideal compression function*, which is basically a random oracle with a fixed-length input.

In the indifferentiability game, we have a distinguisher $\mathcal{D}$, who needs to distinguish two scenarios. In scenario one, the distinguisher $\mathcal{D}$ is provided with:

- The output of construction $\mathcal{C}^{\mathcal{F}}$ using ideal primitive $\mathcal{F}$.
- The ideal primitive $\mathcal{F}$ and possibly its inverse[8] $\mathcal{F}^{-1}$.

In scenario two, the distinguisher $\mathcal{D}$ is provided with:

- The output of random oracle $\mathcal{H}$.
- A simulator $\mathcal{S}^{\mathcal{H}}$ that simulates the primitive $\mathcal{F}$ (and $\mathcal{F}^{-1}$) based on the random oracle $\mathcal{H}$.

We then need to show that there exists a polynomial-time simulator $\mathcal{S}^{\mathcal{H}}$ such that for all polynomial-time distinguishers $\mathcal{D}$, the probability that $\mathcal{D}$ can distinguish the two scenarios is negligible in the security parameter. More formally, let us assume that $\mathcal{D}$ outputs a bit, where w.l.o.g. it outputs 0 if it thinks it is provided with scenario 1, and 1 if it is provided with scenario 2. We then need to prove that

$$\exists \mathcal{S}^{\mathcal{H}}. \forall \mathcal{D}. |Pr[\mathcal{D}(\mathcal{C}^{\mathcal{F}}, \mathcal{F}/\mathcal{F}^{-1}) = 1] - Pr[\mathcal{D}(\mathcal{H}, \mathcal{S}^{\mathcal{H}}) = 1]| \leq \epsilon,$$

where ϵ is negligible in the security parameter and $\mathcal{S}^{\mathcal{H}}$, $\mathcal{D}$ are polynomial-time.

[8] If the primitive is a symmetric primitive, such as a block cipher, then an efficient inverse exists, which the distinguisher also has access to.

2.4.4 Quantum Indifferentiability

The indifferentiability framework and the proofs built upon it are inherently classical. That is, it is not evident whether classical indifferentiability proofs still hold against quantum adversaries. Classical indifferentiability has been proven for many constructions already. Specifically, the sponge construction used in SHA-3 was shown indifferentiabile from a random oracle in [BDPA08].

The indifferentiability game for hash function from the previous section can easily be extended to the quantum case, by making both $\mathcal{S}^{\mathcal{H}}$ and $\mathcal{D}$ *quantum*-polynomial-time algorithms. In [CETU18], Carstens et al. prove under some quantum-information-theoretical conjecture that the sponge and Feistel constructions are not perfectly quantum-indifferentiable, which are popular constructions for cryptographic primitives, which would mean that SHA-3 is not quantum-indifferentiable from a random oracle. The work by Carstens et al. led to the question whether there are any quantum-indifferentiable constructions of one-way hash functions. In [HY18], the positive is shown. Namely, they show that the Merkle-Damgård construction using the Davies-Meyer compression function is a quantum-indifferentiable one-way hash function. Similar constructions have been used in e.g. MD5, SHA-1 and SHA-2.

Concerning quantum security, it has been shown that SHA-3 is quantum-*indistinguishable* [CHS19]. Both the quantum-indistinguishability and quantum non-indifferentiability results for SHA-3 are very recent, and the question remains whether quantum-indifferentiability is required for quantum-secure hash functions. For certain post-quantum constructions, such as keyed primitives, where an attacker does not have access to the internal building block, quantum-indistinguishability provides the necessary security guarantees. For example, this is the case for the NIST submission SPHINCS+ [BHK$^+$19] when instantiated using the Haraka hash function [KLMR16]. However, cryptographic primitives with proofs in the QROM would require quantum-indifferentiability of the hash functions with a random oracle.

2.4.5 Limitations

It is important to note that both the UC and indifferentiability frameworks have limitations, as illustrated in the work of Coron et al. [RSS11]. They first examine the indifferentiability framework and provide a scheme that is secure in the ROM, but insecure when instantiated with a concrete hash function, even though this hash function is indifferentiable from its ideal functionality: the random oracle. This scheme is a hash-based storage auditing scheme, which can be used when a server stores files and the user wants to verify that the file is present in the database (e.g., the database owner did not throw away random files to save space). The scheme uses an ideal compression function[9] f and when a user wants to verify that their file M is still in the database, they send the challenge C. The database owner then has to provide the response

$$r = f(f(IV, M), C), \tag{1}$$

for some fixed constant string IV (the initialization vector).

The construction as provided in equation 1 was shown to be indifferentiable from a random oracle in [CDMP05]. However, the database owner can cheat by computing $Y = f(IV, M)$ when the document is initially received and computing $f(Y, C)$ as a response to any challenge C in the future. Coron et al. [MRH04] analyzed the proofs and concluded that the indifferentiability claims break for security notions captured by experiments that have multiple, disjoint adversarial stages. This is the case for the hash-based storage auditing scheme. In other words, a proof is multi-stage if an adversary can derive some state S from the input it gets that is smaller than the input itself and can use S to answer challenges. Examples of such experiments are the security notions of deterministic public-key encryption, password-based cryptography, hash function non-malleability and key-dependent message security. Security notions that are not affected are those that involve a single stage with a stateful adversary, such as IND-CPA, IND-CCA and EUF-CMA. Coron et al. additionally show that the same limitations hold for the Universal Composability framework.

[9] This is an idealized version of a compression function, which is a function that takes a fixed-length input and provides an output of smaller length such that it is hard to determine what the input was, given the output. These can be used as building blocks to build hash functions.

2.5 Computer-Aided Verification

Oftentimes, security proofs are written by hand and evaluated by experts. Even though this generally filters out flaws, there are cases where it took a while before certain flaws in proofs for cryptographic schemes or protocols were discovered. For example, an attack [IIMP19] on the ISO standardized blockcipher mode OCB2 [ISO09] was discovered after it was standardized and deployed, even though a security proof was provided. This is just one example that shows that it is hard to write perfectly correct proofs, which motivates efforts to introduce computer-aided proof verification. These verification programs have been around for classical proofs, such as CryptoVerif, a generic mechanism for specifying the security assumptions on cryptographic primitives that generates proofs using consecutive games, and EasyCrypt, a tool set for reasoning about relational properties of probabilistic computations using probabilistic relational Hoare logic[10]. A recent effort by Unruh has resulted in a tool that is closely related to EasyCrypt, which can model quantum adversaries using quantum relational Hoare logic, which further extends probabilistic relation Hoare Logic to be able to reason about relationships between quantum algorithms [Unr19]. Accompanying the same work is a tool that serves as a post-quantum alternative to EasyCrypt.

3 Computational Hardness Assumptions

In this section, we informally summarize the main asymmetric cryptographic protocols that are assumed to withstand quantum adversaries. The protocols are categorized based on the underlying computational hardness assumption.

3.1 Code-based Cryptography

A code-based cryptographic system is a public key system exploiting the problem of decoding a random linear code. The problem of decoding code words from random codes is assumed to be a hard problem, even for quantum adversaries.

In coding theory, an error correcting code (ECC) is used to detect and correct possible communication errors, e.g., bit flipping, over a noisy channel. This is achieved through the addition of redundancy to the original information: *the encoding procedure.* Said redundancy allows the receiver to recover up to a certain number t of faulty bits:

- Alice encodes the original message $\mathbf{m}$ to its redundant equivalent $\mathbf{c}$ and sends it to Bob.
- Bob receives $\mathbf{c} + \mathbf{e}$, where $\mathbf{e}$ represent the error due to the unreliable communication channel.
- If the number of errors is below t, Bob will be able to extract $\mathbf{m}$ from $\mathbf{c} + \mathbf{e}$ through the application of the *decoding procedure.*

In 1978, Robert McEliece proposed the first public key cryptosystem based on Goppa codes [McE78]. The public key of McEliece's cryptosystem is an obfuscated version $\mathbf{G}' = \mathbf{SGP}$ of the generator matrix $\mathbf{G}$ of a *random* error correcting code ($\mathbf{S}$ and $\mathbf{P}$ are scrambling and permutation matrices respectively). The encryption of a plaintext $\mathbf{m}$ is a noisy encoding $\mathbf{c} = \mathbf{mG}' + \mathbf{e}$, where $\mathbf{e}$ is a random vector with Hamming weight t. The private key consists of the knowledge of the code structure, i.e., the components (S, G, P) in the decomposition of the public matrix G'. Using this decomposition, a noisy codeword (ciphertext) can be decoded (decrypted) and the message can be retrieved. The intuition behind this approach is that it is hard to decode a (noisy) code word without knowing the structure (decomposition) of the public key $\mathbf{G}' = \mathbf{SGP}$.

The main advantages of this system are the fast encryption and decryption procedures and of course its quantum resistance. However, to achieve the desired security, it requires a large public key. In [BLP08], the public key sizes for the minimum security level of 80-bit and for the more conservative one of 256-bit are around 60 KB and 958 KB respectively.

[10] Relational Hoare Logic reasons about certain statements throughout two programs. Probabilistic Hoare Logic extends this notion to enable probability claims.

3.1.1 NIST Standardization

The following code-based key encapsulation mechanism (KEM) is a third round finalist of NIST's post-quantum standardization process [AASA$^+$20]:

- Classic McEliece (KEM).

Moreover, the following code-based KEMs and digital signature (DS) advance to the third round as alternate candidates:

- BIKE (KEM);
- HQC (KEM).

3.2 Hash-based Cryptography

A hash function is a one way function mapping an arbitrary length bitstring into a fixed length bitstring. More precisely, cryptographic hash functions satisfy the following properties:

- **Pre-image Resistance**: for a fixed hash value, it should be computationally infeasible to find an input that maps to that hash value;
- **Second Pre-image Resistance**: it is computationally infeasible to find a second distinct input that has the same output as a given input;
- **Collision Resistance**: it is computationally infeasible to find two distinct input values with the same hash value.

These properties of cryptographic hash functions can be used to construct signature schemes. To construct public-key encryption schemes or key encapsulation mechanisms, additional properties (trapdoors) are required.

3.2.1 Lamport Signature Scheme

The first hash-based signature scheme was invented in 1979 by Leslie Lamport. The idea is to generate 2ℓ random numbers $r_{i,b}$ for $1 \leq i \leq \ell$ and $b \in \{0,1\}$, and their corresponding hash values $y_{i,b} = H(r_{i,b})$. The random numbers are kept secret (private key) while their hashes are published (public key).

To sign a message $\mathbf{m} \in \{0,1\}^\ell$ the random values r_{i,m_i} for $1 \leq i \leq \ell$ are revealed. To verify the signature one checks that $H(r_{i,m_i}) = y_{i,m_i}$ for all i. Since the hash function is pre-image resistant, a valid signature can be created only with the knowledge of the secret key.

The first drawback of this scheme is that the public key is very large, i.e., the public key contains 2 hash value for every bit of the message that has to be signed. This drawback can be overcome by using Merkle tree structures, in which case the public key consists of only 1 hash value.

The second drawback of this scheme is that (even when improved by using Merkle trees) is *stateful.* More precisely, every random value $r_{i,b}$ can only be used once. Therefore, users have to keep track of the random values that have been used already, i.e., they have to manage the state of the keys. Managing the states of key pairs is very impractical in many scenarios. Additional techniques exist to derive stateless hash-based signature schemes at the cost of increasing the complexity. As an example SPHINCS is a stateless digital signature scheme that follows this approach.

Hash-based schemes are considered good candidates for PQ, since they only rely on well-studied properties of hash functions. These properties are not affected by Shor's algorithm, and the best achievable speedup with Grover's logarithm is a cube-root speedup on brute force collision search [Ber09]. In addition, the size of the public key is relatively small, around 64 bytes.

3.2.2 MPC in the Head

An alternative approach based on the properties of cryptographic hash functions is presented by the digital signature scheme PICNIC. The private key of this DS is a random bit string x, and the public key consists of a one-way function f and the image $y = f(x)$ of the private key x under this one-way function. A signature on a message m is a non-interactive zero-knowledge proof (ZKP) showing that the signer knows a secret key x such that $f(x) = y$, where the challenge randomness of the ZKP depends on the message m. The zero-knowledge proof system that is used follows the *MPC-in-the-Head* paradigm [IKOS07], i.e., it does not introduce any computational assumptions. In theory, the one-way function f can be sampled from any family of one-way functions. However, to optimize performance the family defined by the LowMC block-cipher is used. For this reason the security of PICNIC depends on the security of the LowMC block-cipher.

3.2.3 NIST Standardization

The following hash-based digital signature schemes are alternate candidates in the third round of NIST's post-quantum standardization process [AASA$^+$20]:

- SPHINCS (DS);
- Picnic (DS).

3.3 Multivariate Cryptography

The basic objects of multivariate cryptography are systems $\mathcal{A}$ of nonlinear (usually quadratic) polynomial equations in several variables over a finite field $\mathbb{F}_q$. The security of this type of scheme is based on the *MQ problem,* i.e., solving the aforementioned system $\mathcal{A}$, which is proven to be NP-hard [GJ79]. In particular, Grover's Algorithm guarantees only a square-root speedup on the exhaustive search of the system solution [SW16].

The public key of a multivariate cryptosystem is a set of multivariate polynomials of degree two, while the secret key is the knowledge of a trapdoor that allows to efficiently compute the system's solution. This trapdoor is usually obtained by building the public key $\mathcal{P}$ of m polynomials in n variables as $\mathcal{P} = \mathcal{T} \circ \mathcal{Q} \circ \mathcal{S}$, where $\mathcal{T}$ and $\mathcal{S}$ are two affine maps and $\mathcal{Q}$ is an easily invertible quadratic map.

In the case of a multivariate encryption scheme, $m \geq n$ must hold in order to ensure each encryption has a unique decryption. The encryption of a message $\mathbf{p}$ is given by $\mathbf{c} = \mathcal{P}(\mathbf{p})$, the decryption is given by $\mathbf{p} = \mathcal{T}^{-1}\left(\mathcal{F}^{-1}\left(\mathcal{S}^{-1}(\mathbf{c})\right)\right)$. Note that many constructions for public-key encryption that have been proposed were broken quickly, because the trapdoor could not effectively be hidden from an attacker. Currently, there are not many multivariate public-key encryption schemes that are considered secure.

In the case of a multivariate signature scheme, $m \leq n$ must hold in order to ensure that one can sign any message. The signature of message p is given by $\mathbf{s} = \mathcal{T}^{-1}\left(\mathcal{F}^{-1}\left(\mathcal{S}^{-1}\left(\mathcal{H}(\mathbf{p})\right)\right)\right)$, where $\mathcal{H}$ is a hash function. The authenticity of $\mathbf{s}$ can be verified by checking that $\mathcal{H}(\mathbf{p})$ is equal to $\mathcal{P}(\mathbf{s})$. Multivariate schemes can be implemented very efficiently and provide the shortest signature among post quantum algorithms. Their main disadvantage is the large size of the public keys, which is typically around 10 to 100 kB [DP17].

3.3.1 NIST Standardization

The following multivariate digital signature scheme is a third round finalist in NIST's post-quantum standardization process [AASA$^+$20]:

- Rainbow (DS).

Moreover, the following multivariate DS advances to the third round as an alternate candidate:

- GeMSS (DS).

3.4 Lattice-based Cryptography

In 1996, Miklós Ajtai proposed the first lattice-based cryptosystem, which relied on the hardness of the *shortest vector problem (SVP),* where one aims to find the shortest element in a lattice. A related lattice problem is the closest vector problem (CVP), where one aims to find the lattice element closest to some target point. Both problems are assumed to be computationally hard to solve, unless one has a "good" representation of a lattice.

A lattice can be represented by a set of basis vectors $\mathbf{b}$. This representation is not unique and the quality of different bases can differ significantly. Given a good basis, many lattice problems can be solved efficiently. However, given only a bad basis, it becomes infeasible to solve these lattice problems. In order to leverage lattices in public-key schemes, the key idea is to use a good lattice basis $\mathbf{b}$ as private key and a bad basis $\tilde{\mathbf{b}}$ of the same lattice as public key.

A basic approach goes as follows. The encryption is performed by first encoding the message as a lattice point $\mathbf{m}$. This can be done by using the public basis $\tilde{\mathbf{b}}$. Subsequently a random error is added to that lattice point to obtain the cipher text $\mathbf{c}$. The error should be small enough such that the closest lattice point to $\mathbf{c}$ is $\mathbf{m}$. Decryption is finding the lattice point closest to $\mathbf{c}$, i.e., solving the closest vector problem. Decryption can be performed efficiently by using the private basis $\mathbf{b}$, but is infeasible for anyone with only knowledge of the public basis $\tilde{\mathbf{b}}$.

Another lattice problem that is used for constructing cryptosystems is the *learning with errors* (LWE) problem. The (decisional) LWE problem asks to distinguish between random elements of the form $(A, \mathbf{v}) \in \mathbb{Z}^{m\times n} \times \mathbb{Z}^m$, where A and $\mathbf{v}$ are both sampled uniformly at random, and random elements of the form $(A, A\mathbf{s} + \mathbf{e})$, where A is sampled uniformly at random, $\mathbf{s} \in \mathbb{Z}^n$ is short element and $\mathbf{e}$ is sampled from some error distribution. In 2005, Regev introduced the first LWE-based cryptographic scheme and showed that an efficient algorithm for the LWE problem implies an efficient algorithm for the (approximate) SVP problem.

3.4.1 NIST Standardization

The following lattice-based key encapsulation mechanisms and digital signature schemes are third-round finalists of NIST's post-quantum standardization process [AASA$^+$20]:

- CRYSTALS-KYBER (KEM);
- NTRU (KEM);
- SABER (KEM);
- CRYSTALS-DILITHIUM (DS);
- Falcon (DS).

Moreover, the following lattice-based KEMs are alternate candidates for the third round of NIST's post-quantum standardization process:

- FrodoKEM (KEM);
- NTRU Prime (KEM).

3.5 Supersingular Elliptic-curve Isogeny Cryptography

Elliptic-curve cryptography (ECC) was born in 1985 by the ideas of Neal Koblitz [Kob87] and Victor Miller [Mil85]. These early cryptosystems relied on the group operations within a single elliptic curve and allow for efficient variants of the Diffie-Hellman key exchange protocol. Unfortunately they are only secure against classical adversaries.

In 2006, Rostovtsev and Stolbunov [RS06] introduced the idea of using isogenies (specific type of maps between two curves) between different elliptic curves to construct cryptographic protocols. However in 2010, Childs, Jao and Soukharev found a subexponential quantum attack for this scheme [CJS10].

In 2011, Jao and De Feo introduced the idea of performing a key exchange protocol leveraging isogenies between supersingular elliptic-curves [JF11]. These specific curves have a particular structure which makes them resilient to the Childs-Jao-Soukharev attack.

The key exchange protocol, which is based on random walks in an isogeny graph, is an instantiation of the Diffie-Hellman protocol and it goes as follows. The two participants, Alice and Bob, start from the same common curve E_0, and take a (secret) random walk to some curves E_A, E_B. After publishing their respective curves, Alice starts a new walk from E_B, while Bob starts from E_A. By repeating the same secret steps, they both eventually arrive on a shared secret curve E_S, only known to them.

Isogenies are the only post-quantum approach that enables a Diffie-Hellman like key exchange, the supersingular isogeny Diffie-Hellman (SIDH) key exchange. The main advantage of the SIDH key exchange is its small keys. However, it computational complexity is one or two orders of magnitude larger than other post-quantum primitives. We can formally state the security of SIDH as a hardness assumption on a problem called the *supersingular decisional Diffie-Hellman* (SSDDH) assumption. The best known algorithms for SSDDH have exponential complexity, even on a quantum computer [Feo17].

Key exchange is not the only public-key protocol that can be derived from isogeny graphs. It is in fact possible to derive a public-key encryption protocol similar to El Gamal from either the Rostovtsev-Stolbunov protocol or SIDH.

3.5.1 NIST Standardization

The following isogeny-based key encapsulation mechanism is an alternate candidate in the third round of NIST's post-quantum standardization process [AASA$^+$20]:

- SIKE (KEM).

4 The Quantum Random Oracle Model

A subtle problem arises in cryptography when considering quantum-capable adversaries: security proofs may no longer hold. Namely, some complex cryptographic protocols, which rely on other sub-protocols as building blocks, are proven to be secure only in a special model known as *Random Oracle Model* or *ROM*. When a quantum-capable adversary is considered, proofs in this model are sometimes no longer valid, even if the underlying cryptographic building blocks rely on quantum-safe hardness assumptions (cf. Section 3).

In this section, we explain this problem, as well as its practical impact.

4.1 The Random Oracle Model

In cryptology, security proofs use mathematical arguments to demonstrate that protocols are secure, possibly under the assumption that a given computational problem is intractable. A somewhat controversial [KM15] family of proof models *cryptographic hash functions,* an essential building block for many cryptographic protocols, as mathematical constructions known as *random oracles.* Proofs of this type are "valid in the *Random Oracle Model* or *ROM*", and are somewhat controversial for the following reasons.

- On one hand, a random oracle is *not* an accurate representation of a hash function. In fact, there exists artificial protocols which are proven to be secure in the ROM, but which are provably *in*secure when a hash function is used instead of the ROM, for all possible hash functions [CGH98].
- On the other hand, proofs in the ROM work very well in practice: counterexamples such as the one referenced above remain artificial and of no impact on concrete protocols. Additionally, constructions that are proven secure in the ROM are often very efficient.

Despite the downside expressed in the first point above, the ROM has been widely successful, and many protocols of daily use (e.g., some RSA implementations [CJNP02]) rely on it.

4.2 Quantum Access to the ROM

A security proof should model a real situation as accurately as possible; hence, when considering a quantum-capable adversary, one should assume that it has quantum access to the building blocks in its possession. In particular, if the cryptographic protocol under scrutiny uses a hash function H, then it should be assumed that the adversary can query the hash function in superposition, i.e., that they can obtain the quantum state $\sum |x\rangle|H(x)\rangle$ for any superposition $\sum |x\rangle$ of input values in a single evaluation of the hash function.

Translated to the ROM, this means that such an attacker should be able to query the oracle once with a superposition $\sum |x\rangle$ of input values and obtain $\sum |x\rangle|H(x)\rangle$ (H here being the random oracle). Such a setting is called *Quantum-accessible Random Oracle Model*, or simply *Quantum Random Oracle Model (Q-ROM)*.

The problematic introduced by quantum computing is that security proofs that hold in the ROM do *not* automatically hold in the Q-ROM [BDF$^+$11]. In fact, there are protocols that are provably secure in the ROM but provably *in*secure in the Q-ROM [BDF$^+$11]; however, much like the theory-vs-practice dichotomy of the ROM, such counterexamples are completely artificial, and it is generally believed that a concrete protocol that is secure in the ROM will remain secure in the Q-ROM (if based on quantum-safe hardness assumptions), although with no formal proof in this sense.

We detail in the following subsections some typical scenarios where the Q-ROM plays a role, and what the impact of these considerations is.

4.3 The Fiat-Shamir Transformation

The *Fiat-Shamir transformation* [FS86] turns an interactive Proof of Knowledge (PoK) of a certain form (i.e., a public-coin PoK) into a non-interactive one. The security of the PoK obtained in this way can be proven in a black-box fashion in the ROM under the assumption that the original interactive PoK proof is secure. However, the success probability of an adversary trying to break the non-interactive PoK is a factor q larger than an adversary trying to break the interactive PoK, where q is the amount of queries that the adversary is allowed to make. The Fiat-Shamir transformation is notably used to construct digital signature schemes [GMO16, CDG$^+$17].

The classical security proof of the Fiat-Shamir transform does not directly hold in the Q-ROM. While other, less efficient transforms have been introduced that do carry security over to the Q-ROM [Unr12], a recent result [LZ19, DFMS19] shows that the Fiat-Shamir transform *does* preserve security even in the Q-ROM, albeit with a degradation of the tightness of the reduction. In the Q-ROM case, the probability that an adversary is successful is a factor $(2q + 1)^2$ larger for the non-interactive case [DFM20]. This means that, in practice, Q-ROM Fiat-Shamir reductions result in larger parameters than ROM Fiat-Shamir reductions, e.g., a larger challenge set used in the interactive protocol or the number of parallel repetitions of the interactive protocol should be approximately twice as large. Altogether, digital signature schemes based on quantum-hard computational problems that are proven safe in the ROM (e.g., FISH [CDG$^+$17]) remain provably secure in the Q-ROM, and are thus “fully” provably quantum-safe.

4.4 The Fujisaki-Okamoto Transformation

A typical approach for designing public key encryption and key-encapsulation schemes is to design an IND-CPA secure scheme, which is in general much easier than designing an IND-CCA secure scheme directly, and apply a generic transformation to achieve IND-CCA security. The Fujisaki-Okamoto transform is one of such transforms that turn an IND-CPA secure scheme into an IND-CCA secure scheme. However, these transformations were originally proven secure in the ROM, and security in the Q-ROM is not automatically implied. As the Fujisaki-Okamoto transform is widely applied, we assess its Q-ROM security in this section.

In [FO99] Fujisaki and Okamoto designed a generic transformation to convert an IND-CPA secure public key encryption scheme into an IND-CCA secure public key encryption scheme. A number of variations of the original FO transform, achieving IND-CCA secure schemes with different properties, have been constructed, e.g., [HHK17]. Here, we describe the original approach of Fujisaki and Okamoto. They construct an IND-CCA secure scheme combining:

- An IND-CPA public key encryption scheme $(\mathrm{Enc}_{\mathsf{pk}}^{\mathsf{asym}}, \mathrm{Dec}_{\mathsf{sk}}^{\mathsf{asym}})$ for public private key-pair $(\mathsf{pk}, \mathsf{sk})$;
- A deterministic symmetric encryption scheme $(\mathrm{Enc}_{k}^{\mathsf{sym}}, \mathrm{Dec}_{k}^{\mathsf{sym}})$ for encryption key k;
- Hash functions H_1 and H_2.

Using these ingredients the IND-CCA encryption function is defined as follows:

$$\mathrm{Enc}_{\mathsf{pk}}^{FO}(m; r) = \Big(\mathrm{Enc}_{\mathsf{pk}}^{\mathsf{asym}}\,(r; H_1(r, m))\,, \mathrm{Enc}_{H_2(r)}^{\mathsf{sym}}\,(m)\Big)\,,$$

where m is the message and r is the encryption randomness. For a ciphertext $C = (C_1, C_2)$ and private key sk, the decryption algorithm is defined as follows:

$$\mathrm{Dec}_{\mathsf{sk}}^{FO}(C_1, C_2) = \begin{cases} m, & \text{if} \quad C_1 = \mathrm{Enc}_{\mathsf{pk}}^{\mathsf{asym}}(r; H_1(r, m)), \\ \perp, & \text{otherwise}, \end{cases}$$

where $r = \mathrm{Dec}_{\mathsf{sk}}^{\mathsf{asym}}(C_1)$, $k = H_2(r)$ and $m = \mathrm{Dec}_{k}^{\mathsf{sym}}(C_1)$.

The resulting public-key encryption protocol $(\mathrm{Enc}_{\mathsf{pk}}^{FO}, \mathrm{Dec}_{\mathsf{sk}}^{FO})$ is IND-CCA secure in the random oracle model (ROM), i.e., when the hash functions H_1 and H_2 are modeled as random oracles. However, the security reduction for the FO scheme is not tight. More precisely, the success probability of an adversary is a factor q larger for the FO scheme than it is for the original IND-CPA secure scheme $(\mathrm{Enc}_{\mathsf{pk}}^{\mathsf{asym}}, \mathrm{Dec}_{\mathsf{sk}}^{\mathsf{asym}})$, where q is the number of random oracle queries the adversary is allowed to make.

Unfortunately, the proof techniques to prove the IND-CCA security of the FO transform in the ROM cannot easily be translated into proof techniques for the QROM. This is due to some issues that arise with quantum computation. Specifically, a common proof strategy in the ROM for FO transforms is to show that, if the evaluation of a random oracle on a given input can be distinguished from a uniformly random value, then the adversary must have queried the oracle on that input already. The adversary therefore knows the input and must have broken the security of the asymmetric encryption scheme that is used in the FO transform. Since quantum adversaries can access the random oracle in superposition, it is not immediately clear how this proof technique would work in the QROM.

In [TU15], a generalization of this notion was given that extends to quantum adversaries, which they called *one-way to hiding*. Intuitively, the one-way to hiding property says that for some random oracle $\mathcal{H}$ and uniformly random value y, the behavior of all possible quantum-polynomial-time adversaries $\mathcal{A}$ on either inputs $(x, \mathcal{H}(x))$ or (x, y) differs negligibly. More specifically, for all possible quantum-polynomial-time adversaries $\mathcal{A}$, we consider the difference between the probability that the adversary outputs 1 on input $(x, \mathcal{H}(x))$ and the probability the adversary outputs 1 on input (x, y). In [TU15], they show that this probability difference can be upper bounded by $2 \cdot q \cdot \sqrt{\epsilon}$, where q is the number of queries to the random oracle and ϵ is the probability that measuring the query register of the quantum adversary, after at most q quantum queries to the random oracle, reveals x.

Ideally, this bound would be just ϵ, because the reduction would then be tight. However, we see a linear non-tightness in the multiplicative factor q and a quadratic non-tightness in the square root around ϵ. This means that based on this analysis, the security parameters for the underlying asymmetric scheme would have to be increased to get the same security level for the FO transformed construction. Therefore, any improvement on the bound of the one-way to hiding property immediately improves the efficiency of the IND-CCA scheme produced by the FO transform. In recent years, there have been several efforts to move this bound towards ϵ. The way that these analyses get improved bounds is by setting extra assumptions on the underlying asymmetric scheme. These works attain bounds of

- $2 \cdot \sqrt{q \cdot \epsilon}$ [AHU19];
- $2 \cdot \sqrt{\epsilon}$ [BHH+19];

- $4 \cdot q \cdot \epsilon$ [KSS+20].

We now illustrate why this bound is relevant and how it is used. Most FO transformations can be thought of as consisting of 2 phases. In the **first phase**, a non-deterministic asymmetric encryption scheme is first turned into a deterministic encryption scheme through a *derandomization* process. This is generally done by using a hash function on the plaintext to generate the required randomness to encrypt the plaintext. Therefore, no randomness is sampled anymore and the scheme is deterministic. More formally, let pk be the public key, sk be the secret key, m be the plaintext message, r be encryption randomness and H be a hash function. If the original non-deterministic asymmetric encryption scheme has an encryption function $ENC_{pk}(m, r)$, and a decryption function $DEC_{sk}(c)$, then we can construct a new encryption scheme with encryption function $ENC'_{pk}(m)$ and decryption function $DEC'_{sk}(c)$.

Generally, this is the point where the one-way to hiding bound is crucial for security, because the input to the hash function contains the plaintext. Assuming this bound is taken into account, we can then find parameters for the original scheme, such that the new deterministic scheme has an appropriate security level for the appropriate security notion. What this security notion exactly is, depends on the second phase of the FO transform. In other words, the way that the second phase of the FO transform is designed, impacts the necessary security notion on the intermediate deterministic encryption scheme. Similarly, the properties of the intermediate scheme influence which transformation can be applied in the second phase.

In the **second phase**, the encryption function ENC' and decryption function DEC' are used to construct a Key Encapsulation Mechanism (KEM). There are different approaches to this with subtle modifications. Each modification has implications for the assumptions on the intermediate encryption scheme necessary to achieve IND-CCA security for the resulting KEM. The general structure of turning the intermediate encryption scheme into a KEM is as follows:

Encapsulation:

1. A uniformly random plaintext m is sampled.
2. $c \leftarrow ENC'(m)$. One modification would also add $H'(m)$ to the ciphertext, for hash function H'.
3. k is generated using a hash function H (independent of H'). Different approaches use different elements as input to the hash function. For example, you can calculate $k \leftarrow H(m)$ or $k \leftarrow H(m, c)$.

Decapsulation:

1. $m' \leftarrow DEC'(c)$.
2. A check is done. Different approaches do this differently. You can for example check whether DEC' returns $\perp$ or try to re-encrypt the message by retrieving the used randomness and checking whether the ciphertexts are the same. If the encapsulation was supposed to add $H'(m)$ in step 2, then the check also includes checking whether $H'(m)$ is equal to $H'(m')$.
3. If the check fails, a certain rejection procedure is done. This can be returning $\perp$ or returning a pseudorandom value.
4. k is reconstructed depending on how it was generated during encapsulation.

For example, [SXY18] show that for a *perfectly correct* intermediate encryption scheme with one additional requirement[11], the security bound on the resulting IND-CCA KEM is tight with the intermediate encryption scheme. In other words, the advantage against the security of the intermediate scheme is equal to the advantage against the IND-CCA security of the resulting KEM. This specifically requires encapsulation to generate k using only m as input for the hash function, requires re-encryption in the decapsulation check, and

[11] The extra requirement is that it should be possible to efficiently create elements that are not in the set of valid ciphertexts that are hard to distinguish from valid ciphertexts.

requires a false check to return a pseudorandom value. If all these requirements are met, the only security loss therefore stems from the first phase, where the one-way to hiding property incurs a security loss.

However, most post-quantum encryption schemes are not perfectly correct. There are two recent works that show how an intermediate scheme that is not perfectly correct can be turned into an IND-CCA KEM. Both works require the intermediate encryption scheme to be one-way, such that an attacker has a negligible advantage ϵ_{OW} to invert the encryption function without the secret key. Additionally, it is required that for the intermediate scheme,

1. (informal) it is hard to find a valid ciphertext that decrypts incorrectly, and
2. ENC' has to be injective with probability $1 - \mu$ for a negligible μ.

The respective works then show how to turn these intermediate schemes into a KEM. The tightness results are:

- ([BHH$^+$19]) An attacker has an advantage of $\sqrt{\epsilon_{OW}}$ against the IND-CCA security of the KEM.
- ([KSS$^+$20]) An attacker has an advantage of $q \cdot \epsilon_{OW}$ against the IND-CCA security of the KEM, where q is the number of oracle queries.

This shows that it is generally hard to get a tight transformation into a KEM for non-perfectly correct intermediate encryption schemes. Combining the results from both phases of the FO transformation, we have the following results for FO transformations on δ-correct asymmetric encryption schemes. There are six possible FO transformations and almost all of them implement the second phase differently. For details on that, we refer to the respective papers. We note that the way that the second phase is implemented can affect the size of the ciphertext.

In table 1, the requirements and references to these transformations are given. The *Security Notion* denotes the required security notion of the original (non-deterministic) asymmetric encryption scheme, the *Aditional Requirement* denotes the additional requirement on this scheme for the FO transformation to be IND-CCA, the *Security Loss* denotes the tightness of the reduction, where ϵ denotes the advantage against the security property of the scheme, and *Reference* contains a reference to the paper that introduced the transform. Note that the security loss is provided without constants to maintain readability. For the exact security loss, we refer to the respective papers.

We now explain the additional requirements.

- *Disjoint Simulatability.* A scheme attains Disjoint Simulatability if it is possible to efficiently create elements that are not in the set of valid ciphertexts that are hard to distinguish from valid ciphertexts.
- *Puncturability.* A scheme attains Puncturability, if one element from the message space can be left out to implicitly attain DS.
- *Injectivity.* A scheme attains μ-injectivity if the derived deterministic scheme is injective with probability $1 - \mu$ for a negligible μ.

Table 1: FO transformations for δ-correct asymmetric encryption schemes

Security Notion	Additional Requirement	Security Loss	Reference
IND-OW	-	$q \cdot \sqrt{\epsilon_{OW}} + q \cdot \sqrt{\delta}$	[JZC$^+$18], [JZM19]
IND-CPA	-	$\sqrt{q \cdot \epsilon_{CPA}} + q \cdot \sqrt{\delta}$	[JZM19]
IND-CPA	Disjoint Simulatability	$\sqrt{q \cdot \epsilon_{CPA}} + \epsilon_{DS} + q \cdot \sqrt{\delta}$	[HKSU20]
IND-CPA	Puncturability	$\sqrt{q \cdot \epsilon_{CPA}} + q \cdot \sqrt{\delta}$	[HKSU20]
IND-CPA	μ-injectivity	$\sqrt{q \cdot \epsilon_{CPA}} + q \cdot \sqrt{\delta}$	[BHH$^+$19]
IND-CPA	μ-injectivity	$q^2 \cdot \epsilon_{CPA} + q^2 \cdot \delta$	[KSS$^+$20]

These results can be directly used to find appropriate parameters that attain the right security level for the KEM resulting from the FO transformation. Once again, assume that an adversary has an advantage of ϵ against the security notion of the scheme (Either IND-OW or IND-CPA) and that the scheme is δ-correct. Here q denotes the number of queries an adversary is allowed to make to the random oracle. We now list the maximum value of ϵ against the security notion that is required to obtain an IND-CCA KEM for which the advantage against the IND-CCA security is at most ϵ_{CCA}.

- If the scheme is IND-OW, then using [JZC+18] or [JZM19],

$$\epsilon_{OW} = \mathcal{O}\left(\frac{(\epsilon_{CCA} - q \cdot \sqrt{\delta})^2}{q^2}\right).$$

- If the scheme is IND-CPA, then using [JZM19],

$$\epsilon_{CPA} = \mathcal{O}\left(\frac{(\epsilon_{CCA} - q \cdot \sqrt{\delta})^2}{q}\right).$$

- If the scheme is IND-CPA and *Disjoint Simulatable*, then using [HKSU20],

$$\epsilon_{CPA} = \mathcal{O}\left(\frac{(\epsilon_{CCA} - \epsilon_{DS} - q \cdot \sqrt{\delta})^2}{q}\right),$$

  where ϵ_{DS} is the advantage against the Disjoint Simulatability.

- If the scheme is IND-CPA and *Puncturable*, then using [HKSU20],

$$\epsilon_{CPA} = \mathcal{O}\left(\frac{(\epsilon_{CCA} - q \cdot \sqrt{\delta})^2}{q}\right).$$

- If the scheme is IND-CPA and μ*-injective*, then using [BHH+19] ,

$$\epsilon_{CPA} = \mathcal{O}\left(\frac{(\epsilon_{CCA} - q \cdot \sqrt{\delta})^2}{q}\right)$$

  and using [KSS+20],

$$\epsilon_{CPA} = \mathcal{O}\left(\frac{\epsilon_{CCA}}{q^2} - \delta\right).$$

We note that this overview might make it look like extra requirements sometimes do not give any advantages, but the constants become smaller in those cases, which are not shown in the complexity overview.

5 The Rewinding Problem

The rewinding problem is an issue of similar nature to the random-oracle problem described in Section 4. Namely, several security proofs for a class of cryptographic protocols knows as *Zero-Knowledge proof systems* are no longer valid in a quantum setting, due to the difficulty of transposing a mathematical proof technique known as *rewinding.* Whether this lack of a formal proof does lead to concrete security risks is still under investigation by researchers.

In this section, we describe the issue and its potential impact on security of existing classical protocols.

5.1 Zero-Knowledge Proof Systems

A *Zero-Knowledge Proof System,* often shortened to *ZK,* involves two entities: a *prover* $\mathcal{P}$ and a *verifier* $\mathcal{V}$. Intuitively, the prover holds a (non-secret) value s, and wishes to convince the verifier that s enjoys a particular property; namely, the prover claims to also know a secret value w, known as a *witness,* such that s and w satisfy a particular relation. A ZK system allows the prover to convince the verifier of the veracity of their claim, but without revealing w. For a concrete example, one can think of s as a ciphertext, and assume the prover claims to know the underlying plaintext but does not wish to disclose it with the verifier: a ZK system would then allow him to provide such a proof, where the witness w is in this case the underlying plaintext (and randomness used for the encryption) and the relation is simply $s = \mathrm{Enc}(w)$.

We are here interested in a particular, and highly popular, special family of ZK systems known as *Sigma protocols.* Protocols in this family adhere to the following three-step approach:

1. the prover computes a value a, based on their (supposed) knowledge of s and w, and sends this value to the verifier.

2. the verifier, upon receiving a, and possibly based on s, computes a *"challenge"* value c and sends it back to the prover.

3. the prover computes a value z based on s, w and c, and sends it to the verifier.

The verifier, based on all elements that they have received and computed, then either "accepts" (that the prover holds a witness) or "rejects".

Typically, we require that for a ZK system at least three properties hold (with high probability); in informal terms:

1. *(Correctness or Completeness)* If the prover does indeed know a witness w for s, and if both prover and verifier follow the instructions of the protocol, then the verifier will accept.

2. *(Soundness)* If the prover does *not* hold a witness w, then the verifier will reject with overwhelming probability.

3. *(Zero-Knowledge)* The verifier gains no information on w as a result of the protocol.

The quantum-world issue lies here in a popular strategy to prove soundness. The way that soundness is generally proven, is by showing that if a dishonest prover can convince the verifier that they know w with non-negligible probability, that the value of w can be extracted using oracle access to the prover. This means that the steps a dishonest prover would undergo to convince the verifier that they know w can be used to obtain w, which would break a certain security assumption. The crux is in the fact that such a proof assumes that the extraction algorithm has access to snapshots of the state that the prover is in throughout the protocol. If we want a *quantum proof of knowledge*, then such snapshots should be quantumly available and accessible, but that is impossible for two reasons:

1. The *no-cloning theorem* [WZ82] states that quantum information cannot be copied. When we take snapshots of a certain state of the prover, we implicitly copy the state and save it for later access, but this theorem prohibits that. This means that the extraction algorithm does not work in the quantum setting.

2. When a quantum state is measured, it collapses to a classical state, which 'destroys' information. Such a measurement might be necessary for certain interactions with simulated machines as part of the proof, which would destroy information that could be necessary later on.

However, Unruh [Unr12] showed that classical proofs of knowledge can be quantum proofs of knowledge, if the protocol additionally attains the property of *strict soundness.* Informally, such a property says that for a given a and c as described above, the value z is uniquely defined. This essentially ensures that z itself does not contain a lot of information, so measuring z does not disturb the quantum state too much. In turn, this makes it possible to apply a quantum rewinding technique. This property, together with Correctness and Soundness, makes it a quantum proof of knowledge. However, these properties are not enough to prove that the protocol is quantum-computationally zero-knowledge. Unruh [Unr12] shows how it is possible to create a quantum-computationally zero-knowledge quantum proof of knowledge, using the NP-complete problem of Hamiltonian cycles, under the assumption that quantum 1-1 one-way functions exist[12]. As the problem of Hamiltonian cycles is NP-complete, any NP-relation can be reduced to the Hamiltonian cycle problem, so the proposed protocol can be extended to prove any relation in NP.

6 Commitment Schemes

Commitment schemes are interactive two-party protocols that consist of two phases and two corresponding algorithms: *commit* and *verify.*

1. Commit Phase: Party A wants to commit to a message m. He computes $(c, u) \leftarrow \mathit{commit}(m)$, where c is the commitment and u denotes some *opening information.*

2. Reveal phase: Party A sends m and u to party B. Party B then verifies that $\mathit{verify}(m, u, c) = 1$.

Commitment schemes should have two properties: binding and hiding. The binding property says that given c, party A cannot find m', u' such that $\mathit{verify}(m', u', c) = 1$. The hiding property says that given c, it is infeasible to determine m. A commitment scheme can be information-theoretically binding, in which case there is only one pair m, u that is accepted for commitment c. Alternatively, it can be computationally binding, such that there exist multiple pairs m', u' that are accepted, but they are hard to compute.

The hiding property can be either information-theoretical, statistical, or computational. Information-theoretical hiding means that all accepting pairs m', u' are equally likely. If it is statistically hiding, all accepting pairs m', u' are almost equally likely. If it is computationally hiding, then it is computationally infeasible to find different accepting pairs m', u'.

It was proven in [May97] that no commitment scheme can be both information-theoretically binding and information-theoretically/statistically hiding. Intuitively, this is because, if a commitment is information-theoretically binding, there is only one pair m, u such that $\mathit{verify}(m, u, c)$ returns 1. We therefore can exhaustively try all m', u' to figure out which message was committed to. Similarly, if a commitment scheme is information-theoretically hiding, then there should be multiple m', u' such that $\mathit{verify}(m', u', c)$ returns 1. Then we can exhaustively search for other m', u' that are accepted.

Since we are interested in the post-quantum security of classical commitment schemes, we will assume that all c, m and u are classical. A commitment scheme is computationally binding, if for all quantum-polynomial-time algorithms $\mathcal{A}$:

$$Pr[\mathit{check} = \mathit{check}' = 1 \wedge m \neq m' : (m, u, m', u', c) \leftarrow \mathcal{A}, \mathit{check} \leftarrow \mathit{verify}(m, u, c), \mathit{check}' \leftarrow \mathit{verify}(m', u', c)] \leq \epsilon,$$

where ϵ is negligible in the security parameter. This property is sufficient against classical adversaries, but, unfortunately, the work of [ARU14] shows that computational binding is not enough for a commitment scheme that wants to achieve post-quantum security. More specifically, they show the existence of a commitment scheme that attains computational binding, yet there is a quantum-polynomial-time adversary $\mathcal{A}'$ who, given some commitment c, can find opening information u' for any requested message m' in the message space. This seems to be in contradiction with the binding property, but this is not the case. The adversary against this scheme never finds two pairs $(m, u), (m', u')$ that are both accepting openings for the commitment. The algorithm only finds one pair, but the message can be chosen after the commitment has been made.

[12] In the same work, Unruh makes two proposals based on hash functions and block ciphers respectively. If these are quantum pseudo-random functions, the construction is a quantum 1-1 one-way function.

More specifically, the adversary $\mathcal{A}'$ can sample pairs $(y, |y\rangle)$, where the y is classical and $|y\rangle$ is a superposition of 'accepting' information for y. We do not go into detail on the commitment scheme, but the idea is that multiple y values are used for the commitment and afterwards any m can be chosen. A version of Grover's algorithm can then be run on the superposition $|y\rangle$ for each y to find valid accepting information for y that is consistent with the message m. This in turn is valid opening information for the commitment scheme. As the state has collapsed, no correct opening information can be found for any other $m' \neq m$.

One solution is to only use information-theoretically binding commitment schemes (which will have to be statistically/computationally hiding), but this generally results in larger commitments and inefficient commitment schemes. Other alternatives introduce new binding properties that can solve the aforementioned problem, but they all have certain disadvantages that are undesired for practical and efficient commitment schemes. An overview of these can be found in [Unr16]. Additionally, in [Unr16], Unruh introduces a new binding property called *collapse-binding*, which holds under parallel composition and does not suffer from the same disadvantages as the other propositions. More specifically, this definition works well with quantum-rewinding, does not conflict with information-theoretical/statistical hiding the way that information-theoretical binding would and allows for short commitments. Unruh therefore claims that collapse-binding commitments in the quantum setting are similar to computationally binding commitments in the classical setting.

To explain the property of collapse-binding, we first explain the setup that is used in the definition. Let S, U and M be quantum registers. Given a commitment c, let M and U contain a superposition

$$\sum_{m,u:verify(m,u,c)=1} \alpha_{m,u} |m, u\rangle ,$$

over all possible messages that commit to c, where u is the respective opening information and α denotes the amplitude. The register S contains the rest of the quantum state.

We now consider two settings. In setting 1, we apply a projection operator V_c that checks whether the combined superposition in M and U is a superposition of messages m and opening information u such that each of those is accepting for commitment c. The output is then the registers S, M and U. It is clear that if V_c returns 1 and M contains a superposition over one message m, then the commitment is information-theoretically binding.

In setting 2, we also apply V_c, but we apply an extra operator, M_{check}, which depends on V_c. If V_c returns 0, then M_{check} does nothing, but if V_c returns 1, then M_{check} measures register M in the computational basis. This means that registers M and U collapse, if they are a correct superposition. The registers S, M and U are then provided as output.

Let $\mathcal{A}$ be a quantum algorithm that outputs a classical commitment c and quantum registers S, M and U after applying either setting 1 or setting 2. Let $\mathcal{B}$ be an algorithm that outputs some classical bit b based on output registers S, M and U provided by $\mathcal{A}$. Algorithm $\mathcal{B}$ does not know whether we are in setting 1 or setting 2. Let us now consider the behavior of $\mathcal{B}$. Unruh concludes that if and only if the probability $Pr[b = 1]$ is identical regardless of the setting for all quantum-polynomial-time algorithms $\mathcal{A}$ and $\mathcal{B}$, then the commitment is information-theoretically binding. Intuitively, if a commitment c is not information-theoretically binding, then there is some non-trivial super-position in register M that collapses in setting 2, but not in setting 1. The setting therefore influences the behavior of $\mathcal{B}$ and therefore changes $Pr[b = 1]$.

The definition of *collapse-binding* is then that $Pr[b = 1]$ in setting 1 is negligibly close to $Pr[b = 1]$ in setting 2 for all quantum-polynomial-time algorithms $\mathcal{A}$ and $\mathcal{B}$. This is a relaxed version of information-theoretical binding and essentially means that the superposition in register M is close to a trivial superposition, which is sufficient for computational security. In the same work, Unruh shows that collapse-binding commitment schemes can be constructed using hash functions. More specifically, he shows that collision-resistance of the hash function is too weak a property to obtain a collapse-binding commitment scheme and introduces the notion of *collapsing hash functions*, which can be used to obtain a collapse-binding commitment scheme.

Collapsing hash functions are defined as follows. We once again have two scenarios. Let H be the hash function in question. In both scenarios, we have a quantum-polynomial-time algorithm $\mathcal{A}$, which for some hash output h creates a superposition of all x such that $H(x) = h$. In other words, for a certain hash output of H, $\mathcal{A}$ outputs a superposition of all pre-images. In scenario 1, the superposition is measured before it is sent to quantum-polynomial-time algorithm $\mathcal{B}$ and in scenario 2 it is sent without measuring first. Algorithm $\mathcal{B}$ then outputs a classical bit b based on the provided quantum state. The hash function H is collapse-binding, if for all quantum-polynomial-time algorithms $\mathcal{A}$ and $\mathcal{B}$, the difference in probabilities that $\mathcal{B}$ outputs 1 in each scenario is negligible.

A collapsing hash function can then naturally be used to create collapse-binding commitment schemes. An example of a collapsing hash construction is the Merkle-Damgård construction using a collapsing compression function, such as SHA-2. In conclusion, if post-quantum security for classical commitment schemes is desired, it is important to make sure it is collapse-binding and statistically hiding, or information-theoretically binding and computationally hiding.

7 Symmetric Primitives

In the previous sections, we have focused on asymmetric primitives. This has largely been the focus of post-quantum security analyses, since Shor's quantum algorithm breaks current asymmetric primitives in polynomial time, whereas symmetric primitives were believed to still be secure, albeit at a cost of half the bit security on account of Grover's algorithm. However, we saw that underlying mathematical problems were not the only consideration in the post-quantum security of the asymmetric primitives. Specifically, a lot of additional research has gone into understanding the post-quantum security of hash functions, and a lot of the constructions underlying hash functions are used in symmetric primitives. This raises the question whether we need to redefine security notions for symmetric primitives as well.

Symmetric schemes and primitives do not rely on trapdoor systems the way that asymmetric cryptographic schemes do. Therefore, the main indication of security of a symmetric primitive comes from the area of cryptanalysis. It is therefore important that the security of symmetric primitives be re-established based on the powers of a quantum adversary. In recent years, there has been more research in quantum cryptanalysis of symmetric cryptographic primitives. The work that sparked interest in quantum cryptanalysis was on account of Zhandry [Zha12], who provided the first analysis of quantum-secure pseudo-random functions (QPRF). He provided two models to reason about the powers of a quantum adversary for QPRFs, which apply to all symmetric primitives in general.

1. *Standard Security* [13]: a quantum adversary can do local quantum computations, but input to and output from the primitive in question is purely classical.
2. *Quantum Security*: a quantum adversary has quantum access to the primitive in question, such that a quantum state (e.g., a superposition) can be provided as input and the output is a quantum state as well.

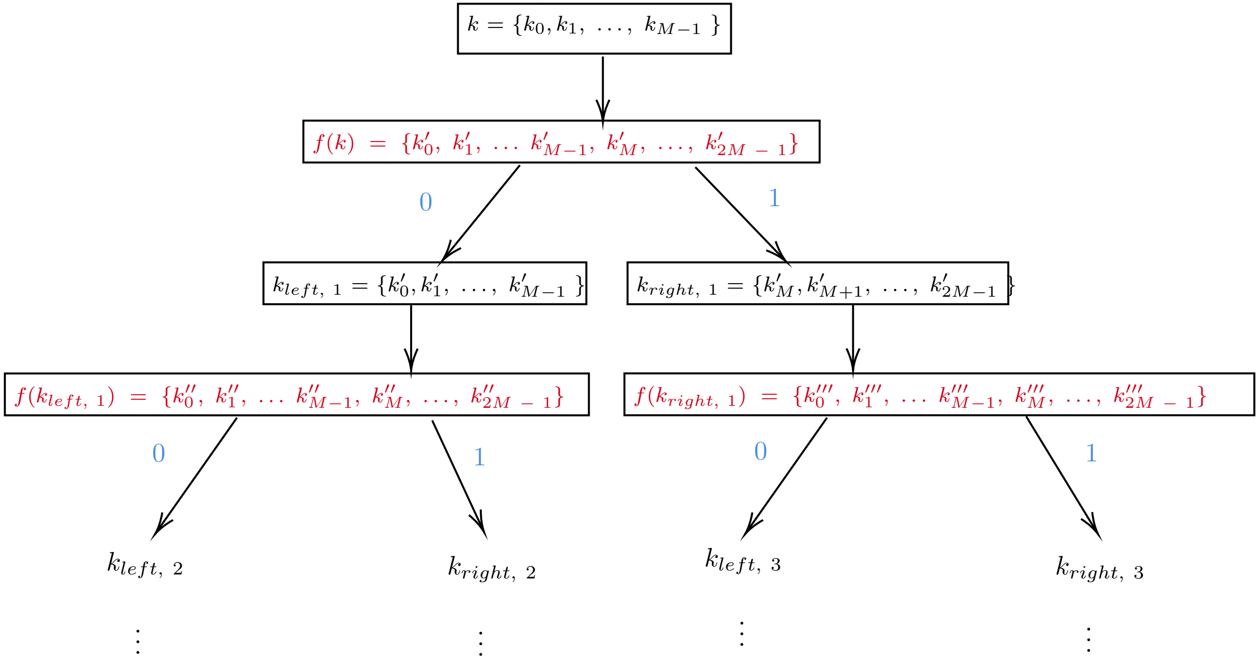
Even though the quantum security gives a lot of power to the adversary, which might not directly be applicable to all practical situations, it captures a wider class of attackers. For example, if a quantum internet becomes wide-spread, this class of attackers becomes more prominent. The conservative long-term approach is therefore to use symmetric primitives that attain quantum security, but for the foreseeable future, standard security is the most realistic.

7.1 Pseudo-Random Functions and Message Authentication Codes

In [Zha12], Zhandry notes that classical proofs of existing PRFs based on pseudo-random generators used reasoning that does not apply to quantum adversaries. Simply put, the classical proofs used the argument that a classical adversary can only call the PRF a polynomial number of times, which evaluates a polynomial number of 'internal states', even though the PRFs have an exponential number of internal states. For

[13] Do not confuse this with the standard model introduced in section 2.3

example, in [GGM84], Goldreich et al. create a PRF using a keyed length-doubling pseudo-random generator and construct a binary tree. The first node is the key itself. Then the edge between this node and the left child-node is assigned the value 0 and the edge between the node and its right child-node is assigned value 1. Then the pseudo-random generator is applied to the value in the node to obtain a string that is twice as large as the string in the node itself. The left half is assigned to the left child node (with edge value 0) and the right half is assigned to the right child node (with edge value 1). This is done recursively.


$k = \{k_0, k_1, \ldots, k_{M-1}\}$
$f(k) = \{k'_0, k'_1, \ldots k'_{M-1}, k'_M, \ldots, k'_{2M-1}\}$
0
1
$k_{left,1} = \{k'_0, k'_1, \ldots, k'_{M-1}\}$
$k_{right,1} = \{k'_M, k'_{M+1}, \ldots, k'_{2M-1}\}$
$f(k_{left,1}) = \{k''_0, k''_1, \ldots k''_{M-1}, k''_M, \ldots, k''_{2M-1}\}$
$f(k_{right,1}) = \{k'''_0, k'''_1, \ldots k'''_{M-1}, k'''_M, \ldots, k'''_{2M-1}\}$
0
1
0
1
$k_{left,2}$
$k_{right,2}$
$k_{left,3}$
$k_{right,3}$
⋮
⋮
⋮
⋮

Whenever the PRF is called on an input b, it decomposes the value into bits b_0 through b_N for some N. Then it starts at the first node, it traverses down the edge with value b_0, at the next node it traverses down the edge with value b_1, and so on. The value at the leaf node is then provided as output.

Each call to the PRF only visits a polynomial number of nodes on each level. Clearly, there are an exponential number of nodes. This argument can then be used to construct a polynomial-time adversary $\mathcal{B}$, who can distinguish the underlying pseudo-random generator from random, given a polynomial-time adversary $\mathcal{A}$ who can distinguish the PRF from random. More specifically, $\mathcal{B}$ succeeds with polynomially smaller success probability than $\mathcal{A}$, so it still runs in polynomial-time, if it wants to achieve the same success probability.

Zhandry notes that many other PRFs have security proofs with similar arguments. However, quantum adversaries could access the PRFs in superposition, possibly accessing all (exponential number of) nodes at the same time with one query. Now the adversary $\mathcal{B}$ constructed from $\mathcal{A}$ succeeds with exponentially smaller probability. To solve this gap, Zhandry provides quantum-security proofs for PRFs based on pseudo-random generators [GGM84], pseudorandom synthesizers [NR95] or lattices [BPR12] . Additionally, Zhandry proves that if secure PRFs exist, then there are standard-secure PRFs that are not QPRFs. Zhandry specifically shows that certain standard-secure PRFs can be turned into PRFs with a hidden period, which can be extracted using Simon's algorithm by quantum adversaries, but not by classical adversaries. They are therefore not QPRFs. In other words, there are PRFs that are indistinguishable from random, if a quantum adversary has classical access to the PRF, but it is distinguishable from random if the adversary has quantum access to the PRF, so even though the three PRFs for which Zhandry provides alternative proofs turned out to be quantum-secure, it does not generally hold that all standard-secure PRFs are quantum-secure. Some negative results are already known, namely PRFs based on three-round Feistel cipher are prone to quantum distinguishing attacks [KM10] and PRfs based on the Even-Mansour cipher are also prone to quantum distinguishing attacks [KM12].

The quantum-security results for PRFs have direct consequences for other cryptographic applications. For example, in[BZ13], Boneh and Zhandry show that quantum-secure PRFs are quantum-secure message authentication codes (MACs). More specifically, they are existentially unforgeable under *quantum* chosen-message attacks. However, not all MAC constructions are quantum-secure. Noteably, Kaplan et al. [KLLN16a] show that Simon's algorithm can be used to break standardized modes of operation such as CBC-MAC, PMAC and GMAC in the quantum security model. These are based on block ciphers and are still broken if the underlying block cipher is quantum-secure.

7.2 Symmetric Cryptosystems

For symmetric cryptosystems, no massive improvements to classical cryptanalysis are known in the standard security mode, except for the speed-up given by Grover's algorithm. However, Grover's algorithm cannot always be applied directly to speed up attacks on symmetric cryptographics systems, so turning the best classical attack into a quantum attack is not always the best strategy [KLLN16b]. According to [BNS19], AES is largely unaffected by quantum attacks, except for Grover's algorithm, even if the quantum security model is assumed.

References

[AASA$^+$20] Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David Cooper, Quynh Dang, John Kelsey, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, Daniel Smith-Tone, and Yi-Kai Liu. Nistir 8309: Status report on the second round of the nist post-quantum cryptography standardization process. *NIST, Tech. Rep., July*, 2020.

[AHU19] Andris Ambainis, Mike Hamburg, and Dominique Unruh. Quantum security proofs using semi-classical oracles. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II*, volume 11693 of *Lecture Notes in Computer Science*, pages 269–295. Springer, 2019.

[ARU14] Andris Ambainis, Ansis Rosmanis, and Dominique Unruh. Quantum attacks on classical proof systems - the hardness of quantum rewinding. *IACR Cryptology ePrint Archive*, 2014:296, 2014.

[BDF$^+$11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, pages 41–69, 2011.

[BDPA08] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. On the indifferentiability of the sponge construction. In Nigel P. Smart, editor, *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*, volume 4965 of *Lecture Notes in Computer Science*, pages 181–197. Springer, 2008.

[BDPR98] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In Hugo Krawczyk, editor, *Advances in Cryptology — CRYPTO '98*, Lecture Notes in Computer Science, pages 26–45, Berlin, Heidelberg, 1998. Springer.

[Ber09] Daniel J Bernstein. Cost analysis of hash collisions: Will quantum computers make sharcs obsolete. *SHARCS*, 9:105, 2009.

[BHH$^+$19] Nina Bindel, Mike Hamburg, Kathrin Hövelmanns, Andreas Hülsing, and Edoardo Persichetti. Tighter proofs of CCA security in the quantum random oracle model. In Dennis Hofheinz and Alon Rosen, editors, *Theory of Cryptography - 17th International Conference, TCC 2019, Nuremberg, Germany, December 1-5, 2019, Proceedings, Part II*, volume 11892 of *Lecture Notes in Computer Science*, pages 61–90. Springer, 2019.

[BHK$^+$19] Daniel J. Bernstein, Andreas Hülsing, Stefan Kölbl, Ruben Niederhagen, Joost Rijneveld, and Peter Schwabe. The SPHINCS $^+$ Signature Framework. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 2129–2146, London United Kingdom, November 2019. ACM.

[Ble98] Daniel Bleichenbacher. Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1. In *CRYPTO*, volume 1462 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 1998.

[BLP08] Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Attacking and defending the mceliece cryptosystem. *IACR Cryptology ePrint Archive*, 2008:318, 2008.

[BNS19] Xavier Bonnetain, María Naya-Plasencia, and André Schrottenloher. Quantum security analysis of AES. *IACR Trans. Symmetric Cryptol.*, 2019(2):55–93, 2019.

[BPR12] Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, volume 7237 of *Lecture Notes in Computer Science*, pages 719–737. Springer, 2012.

[BZ13] Dan Boneh and Mark Zhandry. Quantum-secure message authentication codes. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, volume 7881 of *Lecture Notes in Computer Science*, pages 592–608. Springer, 2013.

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

[CDG$^+$17] Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, and Greg Zaverucha. Post-quantum zero-knowledge and signatures from symmetric-key primitives. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 1825–1842, 2017.

[CDMP05] Jean-Sébastien Coron, Yevgeniy Dodis, Cécile Malinaud, and Prashant Puniya. Merkle-Damgård Revisited: How to Construct a Hash Function. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Gerhard Weikum, and Victor Shoup, editors, *Advances in Cryptology – CRYPTO 2005*, volume 3621, pages 430–448. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. Series Title: Lecture Notes in Computer Science.

[CETU18] Tore Vincent Carstens, Ehsan Ebrahimi, Gelo Noel Tabia, and Dominique Unruh. On quantum indifferentiability. *IACR Cryptol. ePrint Arch.*, 2018:257, 2018.

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

[CHS19] Jan Czajkowski, Andreas Hülsing, and Christian Schaffner. Quantum indistinguishability of random sponges. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II*, volume 11693 of *Lecture Notes in Computer Science*, pages 296–325. Springer, 2019.

[CJNP02] Jean-Sébastien Coron, Marc Joye, David Naccache, and Pascal Paillier. Universal padding schemes for RSA. In Moti Yung, editor, *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, volume 2442 of *Lecture Notes in Computer Science*, pages 226–241. Springer, 2002.

[CJS10] Andrew M. Childs, David Jao, and Vladimir Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *CoRR*, abs/1012.4019, 2010.

[CPS07] Ran Canetti, Rafael Pass, and Abhi Shelat. Cryptography from sunspots: How to use an imperfect reference string. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*, pages 249–259. IEEE Computer Society, 2007.

[DFM20] Jelle Don, Serge Fehr, and Christian Majenz. The measure-and-reprogram technique 2.0: Multi-round fiat-shamir and more. In *CRYPTO (3)*, volume 12172 of *Lecture Notes in Computer Science*, pages 602–631. Springer, 2020.

[DFMS19] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Security of the fiat-shamir transformation in the quantum random-oracle model. *CoRR*, abs/1902.07556, 2019.

[DP17] Jintai Ding and Albrecht Petzoldt. Current state of multivariate cryptography. *IEEE Secur. Priv.*, 15(4):28–36, 2017.

[Feo17] Luca De Feo. Mathematics of isogeny based cryptography. *CoRR*, abs/1711.04062, 2017.

[FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 537–554. Springer, 1999.

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986.

[GGM84] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions (extended abstract). In *25th Annual Symposium on Foundations of Computer Science, West Palm Beach, Florida, USA, 24-26 October 1984*, pages 464–479. IEEE Computer Society, 1984.

[GJ79] M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979.

[GM82] Shafi Goldwasser and Silvio Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In Harry R. Lewis, Barbara B. Simons, Walter A. Burkhard, and Lawrence H. Landweber, editors, *Proceedings of the 14th Annual ACM Symposium on*

Theory of Computing, May 5-7, 1982, San Francisco, California, USA, pages 365–377. ACM, 1982.

[GMO16] Irene Giacomelli, Jesper Madsen, and Claudio Orlandi. Zkboo: Faster zero-knowledge for boolean circuits. In *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016.*, pages 1069–1083, 2016.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 212–219. ACM, 1996.

[HHK17] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the fujisaki-okamoto transformation. *IACR Cryptol. ePrint Arch.*, 2017:604, 2017.

[HKSU20] Kathrin Hövelmanns, Eike Kiltz, Sven Schäge, and Dominique Unruh. Generic authenticated key exchange in the quantum random oracle model. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *Public-Key Cryptography - PKC 2020 - 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, Edinburgh, UK, May 4-7, 2020, Proceedings, Part II*, volume 12111 of *Lecture Notes in Computer Science*, pages 389–422. Springer, 2020.

[HY18] Akinori Hosoyamada and Kan Yasuda. Building quantum-one-way functions from block ciphers: Davies-meyer and merkle-damgård constructions. In Thomas Peyrin and Steven D. Galbraith, editors, *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part I*, volume 11272 of *Lecture Notes in Computer Science*, pages 275–304. Springer, 2018.

[IIMP19] Akiko Inoue, Tetsu Iwata, Kazuhiko Minematsu, and Bertram Poettering. Cryptanalysis of OCB2: attacks on authenticity and confidentiality. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part I*, volume 11692 of *Lecture Notes in Computer Science*, pages 3–31. Springer, 2019.

[IKOS07] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge from secure multiparty computation. In *STOC*, pages 21–30. ACM, 2007.

[ISO09] Information Technology – Security techniques – Authenticated encryption. Standard, International Organization for Standardization, Geneva, CH, 2009.

[JF11] David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *IACR Cryptology ePrint Archive*, 2011:506, 2011.

[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 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part III*, volume 10993 of *Lecture Notes in Computer Science*, pages 96–125. Springer, 2018.

[JZM19] Haodong Jiang, Zhenfeng Zhang, and Zhi Ma. Key encapsulation mechanism with explicit rejection in the quantum random oracle model. In Dongdai Lin and Kazue Sako, editors, *Public-Key Cryptography - PKC 2019 - 22nd IACR International Conference on Practice and Theory of Public-Key Cryptography, Beijing, China, April 14-17, 2019, Proceedings, Part II*, volume 11443 of *Lecture Notes in Computer Science*, pages 618–645. Springer, 2019.

[Kal98] Burt Kaliski. PKCS #1: RSA encryption version 1.5. *RFC*, 2313:1–19, 1998.

[KL14] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography, Second Edition*. CRC Press, 2014.

[KLLN16a] Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia. Breaking symmetric cryptosystems using quantum period finding. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pages 207–237. Springer, 2016.

[KLLN16b] Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia. Quantum differential and linear cryptanalysis. *IACR Trans. Symmetric Cryptol.*, 2016(1):71–94, 2016.

[KLMR16] Stefan Kölbl, Martin M. Lauridsen, Florian Mendel, and Christian Rechberger. Haraka v2 - efficient short-input hashing for post-quantum applications. *IACR Trans. Symmetric Cryptol.*, 2016(2):1–29, 2016.

[KM10] Hidenori Kuwakado and Masakatu Morii. Quantum distinguisher between the 3-round feistel cipher and the random permutation. In *IEEE International Symposium on Information Theory, ISIT 2010, June 13-18, 2010, Austin, Texas, USA, Proceedings*, pages 2682–2685. IEEE, 2010.

[KM12] Hidenori Kuwakado and Masakatu Morii. Security on the quantum-type even-mansour cipher. In *Proceedings of the International Symposium on Information Theory and its Applications, ISITA 2012, Honolulu, HI, USA, October 28-31, 2012*, pages 312–316. IEEE, 2012.

[KM15] Neal Koblitz and Alfred J. Menezes. The random oracle model: a twenty-year retrospective. *Des. Codes Cryptography*, 77(2-3):587–610, 2015.

[Kob87] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48(177):203–209, January 1987.

[KSS$^+$20] Veronika Kuchta, Amin Sakzad, Damien Stehlé, Ron Steinfeld, and Shifeng Sun. Measure-rewind-measure: Tighter quantum random oracle model proofs for one-way to hiding and CCA security. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part III*, volume 12107 of *Lecture Notes in Computer Science*, pages 703–728. Springer, 2020.

[LZ19] Qipeng Liu and Mark Zhandry. Revisiting post-quantum fiat-shamir. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II*, volume 11693 of *Lecture Notes in Computer Science*, pages 326–355. Springer, 2019.

[May97] Dominic Mayers. Unconditionally secure quantum bit commitment is impossible. *Physical Review Letters*, 78(17):3414–3417, April 1997.

[McE78] Robert J. McEliece. A Public-Key Cryptosystem Based on Algebraic Coding Theory. *JPL DSN Progress Report*, 44, 1978.

[Mil85] Victor S. Miller. Use of elliptic curves in cryptography. In Hugh C. Williams, editor, *Advances in Cryptology - CRYPTO '85, Santa Barbara, California, USA, August 18-22, 1985, Proceedings*, volume 218 of *Lecture Notes in Computer Science*, pages 417–426. Springer, 1985.

[MRH04] Ueli M. Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In Moni Naor, editor, *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings*, volume 2951 of *Lecture Notes in Computer Science*, pages 21–39. Springer, 2004.

[NIS16] NIST. Submission requirements and evaluation criteria for the post-quantum cryptography standardization process, 2016.

[NR95] Moni Naor and Omer Reingold. Synthesizers and their application to the parallel construction of pseudo-random functions. *Electron. Colloquium Comput. Complex.*, 2(45), 1995.

[Ped05] Torben P. Pedersen. PKIX - public key infrastructure (X.509). In Henk C. A. van Tilborg, editor, *Encyclopedia of Cryptography and Security*. Springer, 2005.

[RS06] Alexander Rostovtsev and Anton Stolbunov. Public-key cryptosystem based on isogenies. *IACR Cryptology ePrint Archive*, 2006:145, 2006.

[RSS11] Thomas Ristenpart, Hovav Shacham, and Thomas Shrimpton. Careful with composition: Limitations of the indifferentiability framework. In Kenneth G. Paterson, editor, *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, volume 6632 of *Lecture Notes in Computer Science*, pages 487–506. Springer, 2011.

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceeding*, volume 1233 of *Lecture Notes in Computer Science*, pages 256–266. Springer, 1997.

[Sho99] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev.*, 41(2):303–332, 1999.

[SW16] Peter Schwabe and Bas Westerbaan. Solving binary *MQ* with grover's algorithm. In Claude Carlet, M. Anwar Hasan, and Vishal Saraswat, editors, *Security, Privacy, and Applied Cryptography Engineering - 6th International Conference, SPACE 2016, Hyderabad, India, December 14-18, 2016, Proceedings*, volume 10076 of *Lecture Notes in Computer Science*, pages 303–322. Springer, 2016.

[SXY18] Tsunekazu Saito, Keita Xagawa, and Takashi Yamakawa. Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part III*, volume 10822 of *Lecture Notes in Computer Science*, pages 520–551. Springer, 2018.

[TU15] Ehsan Ebrahimi Targhi and Dominique Unruh. Quantum security of the Fujisaki-Okamoto and OAEP Transforms. *IACR Cryptol. ePrint Arch.*, 2015:1210, 2015.

[Unr10] Dominique Unruh. Universally composable quantum multi-party computation. In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*, pages 486–505. Springer, 2010.

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

[Unr16] Dominique Unruh. Computationally binding quantum commitments. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 497–527. Springer, 2016.

[Unr19] Dominique Unruh. Quantum relational hoare logic. *Proc. ACM Program. Lang.*, 3(POPL):33:1–33:31, 2019.

[WSI03] Yodai Watanabe, Junji Shikata, and Hideki Imai. Equivalence between semantic security and indistinguishability against chosen ciphertext attacks. In *Public Key Cryptography*, volume 2567 of *Lecture Notes in Computer Science*, pages 71–84. Springer, 2003.

[WZ82] W. K. Wootters and W. H. Zurek. A single quantum cannot be cloned. *Nature*, 299(5886):802–803, 1982.

[Zha12] Mark Zhandry. How to construct quantum random functions. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 679–687. IEEE Computer Society, 2012.