PhD-FSTM-2023-056
The Faculty of Science, Technology and Medicine

DISSERTATION

Defence held on 19/07/2023 in Esch-sur-Alzette

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN INFORMATIQUE

by

Jeroen Thomas VAN WIER
Born on 13th of August 1995 in Amsterdam (Netherlands)

DENIABILITY, PLAINTEXT-AWARENESS, AND NON-MALLEABILITY IN THE QUANTUM AND POST-QUANTUM SETTING

Dissertation defence committee

Prof Dr Peter Y.A. RYAN, dissertation supervisor
Professor, Université du Luxembourg

Prof Dr Marcus VÖLP, Chairman
Associate Professor, Université du Luxembourg

Prof Dr Boris ŠKORIC
Associate Professor, Eindhoven University of Technology

Dr Peter RØNNE
Postdoctoral Researcher, Université de Lorraine

Dr Ehsan EBRAHIMI, Vice Chairman
Postdoctoral Researcher, Université du Luxembourg

Doctoral Thesis — University of Luxembourg

Doctorat en Informatique

SnT
securityandtrust.lu

Fonds National de la Recherche Luxembourg

uni.lu
Université du Luxembourg

Deniability, Plaintext-Awareness, and Non-Malleability in the Quantum and Post-Quantum Setting

Jeroen Thomas VAN WIER

September 18, 2023

Supervisors:
Prof. Dr. Peter Y.A. Ryan
Dr. Peter Rønne
Dr. Ehsan Ebrahimi

Abstract

Secure communication plays an important role in our everyday life, from the messages we send our friends to online access to our banking. In fact, we can hardly imagine a world without it. With quantum computers on the rise, it is critical for us to consider what security might look like in the future. Can we rely on the principles we use today? Or should we adapt them? This thesis asks exactly those questions. We will look at both the *quantum setting*, where we consider communication between quantum computers, and the *post-quantum setting*, where we consider communication between classical computers in the presence of adversaries with quantum computers. In this thesis, we will consider security questions centred around misleading others, by considering to what extent the exchange of secrets can be denied, misconstructed, or modified. We do this by exploring three security principles.

Firstly, we consider *deniability* for quantum key exchange, which describes the ability to generate secure keys without leaving evidence. As quantum key exchange can be performed without a fully-fledged quantum computer, using basic quantum-capable machines, this concept is already close to becoming a reality. We explore the setting of public-key authenticated quantum key exchange, and define a simulation-based notion of deniability. We show how this notion can be achieved through an adapted form of BB84, using post-quantum secure strong designated-verifier signature schemes.

Secondly, we consider *plaintext-awareness*, which addresses the security of a scheme by looking at the ability of an adversary to generate ciphertexts without knowing the plaintext. Here two settings are considered. Firstly, the post-quantum setting, in which we formalize three different plaintext-awareness notions in the superposition access model, show their achievability and the relations between them, as well as in which settings they can imply ciphertext indistinguishability. Next, the quantum setting, in which we adapt the same three plaintext-awareness notions to a setting where quantum computers are communicating with each other, and we again show achievability and relations with ciphertext indistinguishability.

Lastly, we consider *non-malleability*, which protects a message from attacks that alter the underlying plaintext. Overcoming the notorious “recording barrier” known from generalizing other integrity-like security notions to quantum encryption, we generalize one of the equivalent classical definitions, comparison-based non-malleability, to the quantum setting and show how this new definition can be fulfilled. We also show its equivalence to the classical definition when restricted to a post-quantum setting.

Contents

Acknowledgements

Throughout the four-year journey of writing this thesis, I have been lucky to be supported by so many great people who have each had an impact on this work either directly or indirectly.

I would like to thank my mentor and guide *Peter Rønne*, who has directly supported me from the beginning of my journey till the end and has been a beacon of hope in trying times.

In addition to this, I would like to thank the other members of my CET and thesis committees, *Peter Ryan*, *Elham Kashefi*, *Ehsan Ebrahimi*, *Marcus Völp*, and *Boris Škorić* for their feedback and involvement in my research, as well as for creating the research project that I was so lucky to be a part of.

Without hesitation, I extend the same gratitude to *Arash Atashpendar*, *Christian Majenz*, *Christian Schaffner*, and other co-authors mentioned above whom I have had the pleasure to collaborate with on many interesting topics.

I thank my amazing colleagues from the APSIA group for their wonderful discussions on the most random topics, which were a much-needed distraction at times. In particular, I owe much of my mental health to *Aditya*, who has often shown understanding and listened in trying times.

I thank my parents, *Ron and Carla*, for their eternal and unconditional support and love, always empowering me to seek new paths in life.

Furthermore, I would like to thank all the volunteers of the Erasmus Student Network that have impacted me over the years. They made me feel at home when I first arrived. They have allowed me to have a direct positive impact on society in times when I feared my thesis might not. They have brought forward my natural capacity to seek improvement and inspire others to do the same. It would be impossible to list them all on this page, but I thank *Alessio* for being a mentor and friend to me and I thank *Diane, Cody, and Carolina* for being the best board, and the best friends, I could have ever wished for, always believing in me, and spending hundreds of hours together to achieve our common dreams.

Lastly, I am grateful to be supported by the Luxembourg National Research Fund (FNR), under the joint CORE project Q-CoDe (CORE17/IS/11689058/Q-CoDe/Ryan).

Popular Summary

In our world, the battle for information security is ever-evolving, a struggle between those who seek to break and those who seek to build. Like any battle, it is crucial to think ahead and to learn from past mistakes. Many great scientists have constructed security that is used by us every day when you load your favourite website, when you send a message from your phone, and when you download this thesis. This started long ago. In the times of Romans, messages were secured by cyphers, which relied on the secrecy of the method used to hide information. This did not last forever, as more and more discovered how to break this form of information hiding. Thus, the battle began, with new methods being constructed and broken over time. Nowadays the state-of-the-art methods to protect oneself rely not on the enemy not knowing the security procedure but on the secrecy of the keys used to encrypt a message. But another assumption lies at the heart of modern-day encryption, one under threat by new developments. At the base of all security lie assumptions that we, as researchers, think are impossible for computers to solve quickly, problems that are tested and tried by modern-day computers to be extremely difficult to break. But what if we are wrong? Or a new type of computer is constructed that can solve these problems? This is the threat that quantum computers pose to information secrecy as we know it.

Of course, the threat has not yet arrived. Even optimistic estimates say that quantum computers powerful enough to break the security we use are still decades away. But as always, we are thinking ahead, for the consequences of being unprepared would be catastrophic. This thesis explores the possibility of these new quantum computers, what havoc they could wreak on certain aspects of security, how we can be prepared, and how we can turn the power of these new computers to our advantage to instead secure information.

Security comes in many forms and shapes, depending on the protection you are looking for. In our analysis of what is and is not secure, we often use a three-party system: Alice wants to send a message to Bob, and Eve is eavesdropping on their communication, looking to do something evil to it. Most commonly, security refers to *indistinguishability*, which essentially says that Eve is not able to read the message that Alice is sending to Bob. But many more types of security are applied in everyday life. In this thesis, we look at three different types of security. Firstly, *Deniability*, where we want to make such that even when Eve is the one delivering the message from Alice to Bob, she cannot prove that she did so. In other words, Alice and Bob can deny ever having sent a message. Secondly, *plaintext-awareness*, which means that it is only possible to make an encrypted message if you know which message you are encrypting. Lastly, *non-malleability*, which protects Alice and Bob from having their message changed by Eve. Without this, it is sometimes possible for Eve to change a message even if she cannot read it.

CHAPTER 1

Introduction

Securely communicating messages has been an ongoing struggle for thousands of years. For almost all of this time, secure communication was limited to systems you could write on paper. With the onset of radio, special devices were built to improve the effectiveness of security. But the real adoption of secure communication came with the widespread availability of computers. Nowadays, communicating securely is as easy as opening an app, and is available to most of the people on earth. But throughout history, we have seen the game of cat and mouse unfold. Codes on paper broken by paper. Encryption by machines of radio broken by more complex machines. And now, a new type of computer, the quantum computer, threatens to break the communication schemes we use today[Sho94, Sho97].

In this thesis, we will explore both the opportunities and dangers that quantum computers bring to modern-day security. With security, it is critical to think ahead and prepare for the dangers that might be coming in the near future. This is why top security institutions such as the National Institute of Science and Technology (NIST) of the USA have launched initiatives looking for the best security available that protects against the threat of quantum computers [SN17].

Luckily, the threat of quantum computers is still imaginary for the moment. Quantum computers are expected not to be able to break the most commonly used encryption schemes for a few more decades. But it is important to think ahead in this. Imagine a database of sensitive data is leaked, for example with state secrets. Even if it is encrypted and no computer can currently break it, even a future threat poses a risk to it. If an attacker can store it until technology evolves to break the security, some of the now-discovered secrets might still be relevant.

This thesis aims to be a little step in this process of thinking ahead, considering various forms of security and how they would be affected by quantum computers. Moreover, we look at how we can use quantum computers to enhance our security, or even use them to securely communicate in completely new ways. In the grand scheme of things, most of the work presented in this thesis will remain theoretical for the foreseeable future. However, I hope it will inspire others to build and improve upon it, and when the day comes that an implementation becomes viable, this thesis may have been a small step in that implementation.

1.1 The Definitions of Security

To understand the impact of quantum computers on security, it is essential to establish what form of security we are considering. Oftentimes, this starts with an intuition of what a hypothetical attacker, or *adversary*, should and should not be able to do. For example, you might want it to be impossible for an adversary to read an encrypted message or even for an adversary to distinguish whether two encrypted messages encrypt the same message. Ensuring that encrypted messages, or *ciphertexts*, have this property creates a scheme that satisfies ciphertext indistinguishability.

However, many more forms of security exist. After establishing an intuition, the next step is to formulate a technical statement that captures this intuition, so that the security of schemes can be proven, in a mathematical sense, to be equivalent to a problem that is hard for computers to solve. This process of translating an intuition to computer science concepts is what we tackle in this work, by looking at existing definitions of security concepts for classical (non-quantum) computers and adapting them to our desired setting. However, often a literal translation, inserting the word ‘quantum’ in front of every term, does not yield the desired result. Due to the inherent properties of quantum mechanics, in particular, the impossibility of cloning quantum data, a more creative solution is needed.

In this thesis, we look at forms of security dealing with the misleading or modification of ciphertexts, rather than the secrecy of the message. We will do this by looking at three distinct security properties. In the following Subsections, we will briefly explain the intuition behind the notions we will explore, though each notion is explained in more detail in the chapters that cover them.

1.1.1 Deniability

Firstly, we consider *deniability* for quantum key exchange. Imagine two parties, Alice and Bob, who wish to exchange a secret message. To use any form of security, they must first establish a key to use for encrypting their messages. To exchange such a key, they can use the power of quantum mechanics to their advantage, through a process called Quantum Key Exchange (QKE, sometimes also called Quantum Key Distribution). With very basic quantum computing resources and a bit of classical cryptography, they are able to securely establish a key in such a way that no eavesdropper Eve can learn what the key is.

However, doing this leaves behind evidence of the interaction. For the process to work, classical messages sent between Alice and Bob need to be digitally authenticated. While these signatures do not give away the contents of the key, they do irrefutably prove that Alice and Bob were involved in this process. Sometimes, this can be a problem, and you would prefer there to be no proof of the interaction. This is what deniability, in this setting, would guarantee. We establish a form of security that allows Alice and Bob to establish a key with all the security that QKE offers, with an eavesdropper Eve reading all classical and quantum communication between Bob and Alice. After they have established the key, Eve is left with no proof of the interaction; all the data that Eve has gathered is indistinguishable from data that she could have generated herself.

1.1.2 Plaintext-Awareness

The second security concept we explore is *plaintext-awareness*. This concept covers a more nuanced form of security, that intuitively protects against attackers creating encryptions of random messages. In this setting, we challenge an adversary to produce a valid encrypted message without using the key. If they are not able to do so, it shows that it is only possible to create encryptions by using the encrypting algorithm of a scheme, which means that any party creating a ciphertext is aware of the plaintext it encrypts.

This property is important because it shows that it is not possible to randomly generate ciphertexts, which could mislead systems they are sent to into thinking the sender knows the right keys. Using plaintext awareness, we can also prove a higher level of ciphertext indistinguishability of schemes that possess the property, making it a useful tool when proving the security of schemes. Interestingly, we were able to study this security property in both the quantum and post-quantum case, which allowed us to highlight the similarities and differences between the settings.

1.1.3 Non-Malleability

Lastly, we consider how to protect encrypted data from being changed, through the security property of *non-malleability*. In some schemes, it can be possible to structurally modify encrypted data even without being able to decrypt the ciphertext. Non-malleability intends to secure against this kind of attack. To evaluate this form of security, we give an adversary an encrypted message and challenge them to change it, and afterwards, we evaluate the success of the adversary by comparing the output of the adversary with the original ciphertext. In the quantum setting, this presents a natural challenge as we cannot both give the state to the adversary and keep it to compare later. We show how to overcome this challenge through clever construction of the state given to the adversary.

1.2 Overview of the Thesis

This thesis is built from the work presented in four separate papers. Each paper is discussed in a chapter dedicated to it including an introduction to the topic. To prepare the reader for these topics, we present the common preliminaries of the topics in Chapter 2. Each chapter will present the chapter-specific preliminaries, and then continue with the main topics of the paper that it covers, concluding with a discussion on future work in that field.

We will start with an overview of the model and notation we use to describe security concepts, then give a brief overview of security concepts used in classical cryptography. Afterwards, we give an overview of the basics of quantum computing and highlight the abstractions that we use in our security definitions. Using these preliminaries, we will then tackle the three core topics of this thesis.

1.2.1 Deniability

Firstly, we will tackle deniability for quantum key exchange through two chapters.

In Chapter 3 we look at designated verifier signatures, and construct a security property called 'sender-privacy' which captures the notion that it is impossible for a third party to deduce from a signature who was the person that signed it which takes into account a multi-party setting. Schemes with this property are called Strong Designated Verifier Signature (SDVS) schemes.

Definition 1.1 (Simplified)**.** *A designated verifier signature scheme is n-party sender-private if an adversary interacting with n parties with oracle access to the signing, simulating and verification procedures has negligible advantage in deducing who the signing party of a challenge signature is. Here the challenge signing party is always one of two fixed parties known to the adversary.*

We also present a number of alternate definitions that could be considered and show they are equivalent. Creating a new security definition of this form creates the challenge of showing which existing and future schemes satisfy it. To simplify this process, we present two major theorems that reduce this challenge significantly by showing the common cases where sender-privacy definitions from other literature are equivalent to the one we presented, allowing schemes that satisfy these other definitions to automatically satisfy our definition.

Theorem 1.2 (Simplified)**.** *For any scheme that is strongly unforgeable, the definitions of sender-privacy with or without access to a verifications oracle are equivalent.*

Theorem 1.3 (Simplified)**.** *For any scheme that is strongly unforgeable and computationally non-transferable, the definitions of* 2*-party sender-privacy and* n*-party sender-privacy are equivalent.*

We build upon this in Chapter 4, where we use SDVS schemes to build a deniable version of BB84 [BB84], one of the most famous QKE protocols. We start off by defining what deniability means in this context, where we focus on participation deniability. This allows two parties to exchange a key without leaving any identifiable evidence of the interaction.

Definition 1.4 (simplified)**.** *A* QKE *scheme is* deniable *if for any adversary* $\mathcal{M}$ *there exists a simulator* $\mathsf{SIM}_{\mathcal{M}}$ *that receives the same inputs, which creates a state indistinguishable by quantum computers from the internal state of* $\mathcal{M}$*, where* $\mathsf{SIM}_{\mathcal{M}}$ *does not interact with the honest parties that* $\mathcal{M}$ *interacts with.*

We show a concrete implementation of such a deniable scheme and define it in an established model for QKD[MSU13] to show its security.

Theorem 1.5 (simplified)**.** *A public-key authenticated version of the BB84 protocol using a post-quantum secure Strong Designated Verifier Signature Scheme is deniable.*

1.2.2 Plaintext-Awareness

The second security concept we explore is plaintext-awareness (PA), which we discuss in Chapter 5. This notion captures the intuition that one should only be able to create a ciphertext by encrypting a plaintext, rather than generating a ciphertext randomly. First, we consider the post-quantum setting and create six different definitions of plaintext-awareness, corresponding to the three different types of PA in the literature considered in two different oracle settings.

Definition 1.6 (informal)**.** *A classical public-key encryption scheme is* $\mathsf{pqPA\text{-}C_{dec}}$ *if no quantum adversary can distinguish between a decryption algorithm and a plaintext extractor through classical oracle access. Here, a plaintext extractor is an algorithm that, using the internal state of the adversary, produces the plaintext corresponding ciphertext without access to the corresponding secret key.*

The three variants of this definition that we present differ through the type of access they have to the oracle, in line with existing literature.

- In the PA0 setting, the adversary may make one query to the oracle.
- In the PA1 setting, the adversary may make multiple queries to the oracle.

- In the PA2 setting, the access to the oracle is the same as in PA1, but the adversary can also eavesdrop ciphertexts that it may not query to the oracle.

The $\mathsf{C}_{\mathsf{dec}}$ setting focuses on having classical access to the oracle, where each query is a classical bitstring. We also explore the setting where this access is quantum instead, allowing the adversary to make superposition queries to the oracle.

Definition 1.7 (informal)**.** *A classical public-key encryption scheme is* $\mathsf{pqPA}\text{-}\mathsf{Q}_{\mathsf{dec}}$ *if no quantum adversary can distinguish between a decryption algorithm and a plaintext extractor through quantum oracle access.*

We show the relationship between all these six notions and their satisfiability. Furthermore, we consider the classical result that IND-CPA and PA together imply forms of IND-CCA, and lift this result to the post-quantum setting.

Theorem 1.8 (informal)**.** *Any classical public-key encryption scheme that is IND-qCPA secure and* $\mathsf{pqPA2}\text{-}\mathsf{Q}_{\mathsf{dec}}$ *secure is also IND-qCCA secure.*

Afterwards, still in Chapter 5, we consider the notion of plaintext-awareness for communication between quantum computers and define the three different notions of PA in this setting. Here, we focus strongly on the issue that comes up in the PA2 setting, where eavesdropped ciphertexts cannot be queried to the decryption oracle. This presents an inherent challenge, where it is necessary to record a list of eavesdropped states that are given to the adversary, which is hindered by the no-cloning properties of quantum states. We overcome this by creating a counterfactual definition, similar to the QIND-CCA2 definition presented in [AGM18].

Also in the quantum setting, we show the relations between the notion, and satisfiability, and explore the conditions that imply QIND-CCA notions.

1.2.3 Non-Malleability

Lastly, in Chapter 6, we explore the notion of non-malleability in the setting of communication between quantum computers. This notion captures the intuition that an attacker cannot structurally change an encrypted message. It derives its importance from the fact that signature schemes do not exist for quantum states [AGM21], thus non-malleability forms the strongest notion of integrity in the public-key setting. We define a notion of non-malleability based on the classical notion of comparison-based non-malleability known from literature. In comparison-based non-malleability, an adversary is challenged to modify a given ciphertext such that the modification implements a structural change in the plaintext. The adversary is then challenged to describe the change it implemented on the plaintext through a mathematical relation which should hold between the original plaintext and the decryption of the modified ciphertext.

Here we deal again with the same problem as we did in the setting of plaintext-awareness, where a quantum state must both be given to an adversary and kept for later comparison. Here we solve this problem using a new approach, where the challenge state is produced twice by distilling the sampling algorithm to a unitary operation. The relationship is then implemented as a measurement on the combined space of both states.

We then go on to show the relationship between our definition, when restricted to a post-quantum setting, and the original comparison-based non-malleability notion.

Theorem 1.9 (informal)**.** *When restricted to the post-quantum setting,* QCNM *and* CNM *are equivalent.*

Lastly, we show the satisfiability of our definition.

Theorem 1.10 (informal)**.** *Using a comparison-based non-malleable classical public-key scheme and a non-malleable quantum symmetric-key scheme it is possible to construct a* QCNM *scheme via quantum-classical hybrid encryption.*

CHAPTER 2

Preliminaries

The work discussed in this thesis is built upon the research of many great scientists. In this chapter, we will give an overview and introduction of the core concepts used throughout the different chapters. For chapter-specific background knowledge, each chapter will include its own section that builds upon this one. As with any work, it is impossible to give an overview of all research from the beginning of computer science. For this chapter, we will assume the reader has knowledge of the construction of Turing machines, knows the basics of Shannon information theory, and possesses mathematical maturity.

The chapter is broken down into two sections. In Section 2.2, we will introduce all elements related to quantum computing and its cryptographic implications. To build up to this, we will introduce any relevant classical security concepts in Section 2.1.

2.1 Classical Security Concepts

Many of the security concepts developed for quantum computers over the last years are built on the classical concepts that mirror them. Often, this presents an interesting challenge when common techniques in the classical setting do not translate well into the quantum setting. In this section, we will introduce the classical security concepts that we aim to bring to the quantum setting. In the next section, we will elaborate on the challenges one faces in this process.

2.1.1 Notation

As computers evolve, they become more and more powerful and can execute more and more instructions in the same amount of time. To ensure that the security concepts that are created today can be adapted to the security needs of the world of tomorrow, it is common practice to introduce a *security parameter* that is directly linked to the security of a scheme. In general, the execution time and space of algorithms that are part of a security scheme should be polynomial, in which case security can be defined as the absence of a possible attack that executes in time polynomial to the security parameter. We denote with $\kappa \in \mathbb{N}$ the security parameter of a scheme and implicitly assume that any algorithm that is part of a scheme is given input 1^κ, i.e. the string of κ 1's, in addition to its specified inputs. We implicitly assume that all adversaries are probabilistic polynomial-time Turing machines (PPT) when talking about classical adversaries.

We use the following shorthands throughout this work:

- $[n]$ denotes the set $\{0, \ldots, n\}$
- $\varepsilon \leq \text{negl}(n)$ denotes that a function ε is negligible in n, which means that for every polynomial p there exists $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$ it holds that $\varepsilon(n) < \frac{1}{p(n)}$
- $\perp$ is reserved as an error symbol, and implicitly added to the output space of algorithms
- $\boldsymbol{v}$ denotes a vector of values and v_i denotes the i-th value of this vector
- $\log(x)$ denotes the base-2 logarithm of x

In the rest of the work, we will refer to probabilistic Turing machines as *algorithms*. This is done as a layer of abstraction, but also to indicate that the underlying mathematical model is not crucial to the definitions. The security concepts defined throughout this work could be adapted to fit any model for computation that has a measure of execution time. The notation $y \leftarrow A(x)$ is used to denote the execution of an algorithm A on input x with output y. We use the $\leftarrow$ symbol instead of the $=$ symbol to highlight that often the algorithm is probabilistic and repeated execution on the same input can give different outputs. This randomness is achieved through a random input r, which is only explicitly mentioned where needed by writing $y \leftarrow A(x; r)$.

2.1.2 Encryption Schemes

At the basis of security lies the principle of encryption schemes. They are used to take an unencrypted message, known as a *plaintext*, and translate it into an encrypted message, the *ciphertext*. To do this, a user of a scheme makes use of a *key*, a bitstring that is only known to them and the person they intend to send the message to.

Definition 2.1. *A* symmetric-key encryption scheme (SKES) *is a triple* $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$*, where*

- KeyGen*: Produces a key k.*
- $\mathsf{Enc}_k(m) := \mathsf{Enc}(k, m)$*: Upon input of a key k and a message m, produces a ciphertext.*
- $\mathsf{Dec}_k(m) := \mathsf{Dec}(k, c)$*: Upon input of a key k and a ciphertext c, produces a plaintext.*

These algorithms must satisfy the property that $\mathsf{Dec}_k \circ \mathsf{Enc}_k$ *is statistically indistinguishable from the identity map* id *for all* $k \leftarrow \mathsf{KeyGen}(1^\kappa)$*.*

This definition of a security scheme captures the case where both the sender and receiver use the same key. However, often times a different form of encryption is used, where each party has a *private key* (or *secret key*) sk and a *public key* pk. Together, they form a keypair. Each party is expected to share their public key but keep their private key to themselves. This public key can then be used for encrypting messages that one wants to send to the owner of the corresponding private key.

Definition 2.2. *A* public-key encryption scheme (PKES) *is a triple* $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$*, where*

- KeyGen*: Produces a keypair* $(\mathsf{pk}, \mathsf{sk})$*.*

- $\mathsf{Enc}_{\mathsf{pk}}(m) := \mathsf{Enc}(\mathsf{pk}, m)$*: Upon input of a public key* pk *and a message m, produces a ciphertext.*

- $\mathsf{Dec}_{\mathsf{sk}}(m) := \mathsf{Dec}(\mathsf{sk}, c)$*: Upon input of a private key* sk *and a ciphertext c, produces a plaintext.*

These algorithms must satisfy the property that $\mathsf{Dec}_{\mathsf{sk}} \circ \mathsf{Enc}_{\mathsf{pk}}$ *is statistically indistinguishable from the identity map* id *for all* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa)$*.*

2.1.3 Signature Schemes

Signature schemes are used throughout classical security to authenticate messages, i.e. to verify that they are indeed sent by the correct party and not a malicious one. Basic signature schemes operate in a setting where each party has a public and a private key, and all parties know beforehand what the public keys of the other parties are. Anyone holding a private key can use it to Sign a message, producing a *signature* of this message. Using the public key of the party that signed the message, anyone who wants to can verify that the signature indeed matches both the message and that it was produced by the correct party.

Definition 2.3. *A* signature scheme *is a tuple* $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *of* PPT *algorithms such that:*

- Setup*: Produces the public parameters of a scheme,* params*. It is implicitly assumed that these parameters are passed to the following algorithms.*

- KeyGen*: Produces a keypair* $(\mathsf{pk}, \mathsf{sk})$*.*

- $\mathsf{Sign}(\mathsf{sk}_S, \mathsf{pk}_S, m)$*: Upon input of a sender's keypair and a message m, produces a signature if all keys are valid and $\perp$ otherwise.*

- $\mathsf{Verify}(\mathsf{pk}_S, m, \sigma)$*: Upon input of a sender's public key, a message m, and a signature σ, outputs the validity of σ (a boolean value) if all keys are valid and $\perp$ otherwise.*

In general, it is desired that signatures can be verified by anyone holding the public key of the signer. However, this also creates sometimes undesirable evidence of a party's involvement in the execution of a scheme. The same property that proves the authenticity of a message m belonging to a party P, also proves that P was involved in the creation of this signature. In case one would like to avoid this problem, it is possible to use designated verifier signature schemes. In these schemes, the intended party to verify the message is indicated at the time of creating the signature and the signature is constructed such that its authenticity can only be validated by the intended recipient. Furthermore, the verifier can simulate the creation of signatures with them as the intended recipient, which are indistinguishable from real signatures.

Definition 2.4. *A* designated verifier signature scheme (DVS scheme) *is a tuple* $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Simulate})$ *of* PPT *algorithms such that:*

- Setup*: Produces the public parameters of a scheme,* params*. It is implicitly assumed that these parameters are passed to the following algorithms.*

- KeyGen*: Produces a keypair* $(\mathsf{pk}, \mathsf{sk})$*.*

- $\mathsf{Sign}_{S\to V}(m) := \mathsf{Sign}(\mathsf{sk}_S, \mathsf{pk}_S, \mathsf{pk}_V, m)$*: Upon input of a sender's keypair, a verifier's public key, and a message m, produces a signature σ if all keys are valid and $\perp$ otherwise.*
- $\mathsf{Verify}_{S\to V}(m,\sigma) := \mathsf{Verify}(\mathsf{sk}_V, \mathsf{pk}_V, \mathsf{pk}_S, m, \sigma)$*: Upon input of a verifier's keypair, a sender's public key, a message m, and a signature σ, outputs the validity of σ (a boolean value) if all keys are valid and $\perp$ otherwise.*
- $\mathsf{Simulate}_{S\to V}(m) := \mathsf{Simulate}(\mathsf{sk}_V, \mathsf{pk}_V, \mathsf{pk}_S, m)$*: Upon input of a verifier's keypair, a sender's public key, and a message m, produces a simulated signature σ'.*

2.2 Quantum Computation

In this section, we will discuss the basics of quantum computing and its relation to cryptography. While we present a basic overview and explain our notation, we refer to [Wat18] for a complete and detailed overview of quantum computing.

2.2.1 Quantum States

In this thesis, we will model quantum states as mathematical objects without considering their physical implementation. The simplest class of states we will consider is the class of pure quantum states, which are modelled as vectors in a complex space.

We denote vectors $|\psi\rangle = (\psi_1, \psi_2, \cdots, \psi_n) \in \mathbb{C}^n$ using bra-ket notation. Here vectors are denoted as $|\phi\rangle$, and $\langle\phi|$ is used to denote the Hermitian transpose of that vector. The inner product of two vectors $|\phi\rangle$ and $|\psi\rangle$ is defined as $\langle\phi|\psi\rangle = \sum_i \psi_i^* \phi_i$ where ψ_i^* is the complex conjugate of ψ_i. This inner product can also be seen as the matrix multiplication of $\langle\phi|$ and $|\psi\rangle$, which is the inspiration for the notation. The norm of a matrix M is defined as $\|M\| = \mathrm{Tr}\left[\sqrt{M^\dagger M}\right]$. The n-dimensional Hilbert space $\mathcal{H}$ is the complex vector space $\mathbb{C}^n$ with the inner product defined above and is referred to as the *quantum system.* For two quantum systems $\mathcal{H}_1$ and $\mathcal{H}_2$, the composition of them is defined by the tensor product, denoted $\mathcal{H}_1 \otimes \mathcal{H}_2$. We use $|\mathcal{H}|$ to denote the dimension of a quantum system. In this thesis, we will only consider finite-dimensional Hilbert spaces.

In this model, a *pure quantum state* $|\psi\rangle$ is a vector $|\psi\rangle$ in $\mathcal{H}$ with norm 1. To explicitly denote that a state $|\phi\rangle$ is an element of a quantum system $\mathcal{H}_A$, we use a grey superscript $|\phi\rangle^A$. Quantum states can be composed using the tensor product to form a state in the composite quantum system. However, not all states in the composite system can be written as a tensor product. When a state $|\phi\rangle^{AB}$ in a composite system $\mathcal{H}_A \otimes \mathcal{H}_B$ cannot be described as the tensor product of two states in the systems $\mathcal{H}_A$ and $\mathcal{H}_B$ respectively, then we say $|\phi\rangle$ is an *entangled state*.

However, to describe a quantum state as a vector, it is critical that one knows the entire space that the state lives in, meaning there is no uncertainty about which state the system is in. In many cases, it is not possible to describe a state like this because a part of the system is unknown or the system is in an uncertain state. To describe those cases, we generalize the definition of quantum states.

In this general model, quantum states are described by *density matrices*, which are positive semi-definite Hermitian matrices with trace 1. The set of density matrices on a Hilbert space $\mathcal{H}_A$ is denoted by $\mathcal{D}(\mathcal{H}_A)$. In this model, a pure state $|\phi\rangle$ is instead modelled as $|\phi\rangle\langle\phi|$, though the notation may be used interchangingly.

To simplify, we define a few states that are commonly used throughout this thesis:

- The *classical states* are pure states that represent classical bitstrings. For single bits, they are $|0\rangle = (1, 0) \in \mathcal{C}^2$ and $|1\rangle = (0, 1) \in \mathcal{C}^2$. Any bitstring $\boldsymbol{b} \in \{0, 1\}^n$ can then be respresented as a state $|\boldsymbol{b}\rangle = |b_1\rangle \otimes \ldots \otimes |b_n\rangle$;
- The *maximally entangled state* over two systems $\mathcal{H}_A$ and $\mathcal{H}_B$ of dimension $\mathcal{C}^{2^n}$ is denoted $|\phi^+\rangle^{AB} = \frac{1}{\sqrt{|A|}} \sum\limits_{\boldsymbol{x} \in \{0,1\}^n} (|\boldsymbol{x}\rangle \otimes |\boldsymbol{x}\rangle)$;
- The *maximally mixed state* is defined as $\tau^A = \frac{\mathbb{I}}{|A|}$, which denotes a quantum state in a completely unknown state.

2.2.2 Quantum Operations

Having defined what the state of a quantum system is, we now explore the operations that one can perform on this state. The first operation that one can perform is a *unitary operation*, which corresponds to a physical operation on a state without measuring. A unitary operation over $\mathcal{H}_{\mathcal{A}}$ is a matrix U^A such that $UU^\dagger = U^\dagger U = \mathbb{I}$ where $U^\dagger$ is the Hermitian transpose of U and $\mathbb{I}$ is the identity operator over $\mathcal{H}$. A unitary U^A and be applied to a state ρ^A resulting in the state $U\rho U^\dagger$. Unitary matrices can be combined using the tensor product to form a unitary matrix on a composite quantum system. When a unitary U^A is applied to a state ρ^{AB}, i.e. a state in a quantum system composed of both $\mathcal{H}_A$ and another system $\mathcal{H}_B$, then we implicitly apply the identity operation to the $\mathcal{H}_B$ system. This means we can write $U^A \rho^{AB} U^{\dagger A}$ to mean $(U^A \otimes \mathbb{I}^B)\rho^{AB}(U^{\dagger A} \otimes \mathbb{I}^B)$.

Note that it is physically impossible to directly inspect the exact state that a quantum system is in. To gather information about the state of a system, one must perform a measurement. An orthogonal projection P over $\mathcal{H}$ is a linear transformation such that $P^2 = P = P^\dagger$. A *projective measurement* on a Hilbert space is defined as a family of non-zero projectors $\{P_0, \ldots P_n\}$ that are pairwise orthogonal. For a state ρ, the output of a measurement is randomly selected from $[n]$, where i is selected with probability $\mathrm{Tr}\left[P_i \rho\right]$. When the outcome of a projective measurement is i, the state *collapses* to $\frac{P_i \rho P_i}{\mathrm{Tr}[P_i \rho]}$.

An example of such a measurement is the *computational basis measurement*, which projects a state onto a classical state. On a quantum system $\mathcal{H}_A$ of dimension $\mathcal{C}^{2^n}$, this measurement is defined as $\{P_b = |\boldsymbol{b}\rangle \langle \boldsymbol{b}| \mid \boldsymbol{b} \in \{0, 1\}^n\}$.

Using a number of auxiliary systems, one can extend projective measurements to a more general type of measurement, called a *positive operator valued measure (POVM) measurement.* Such a measure is described by a set $\{M_i \mid i \in \mathbb{N}\}$ such that all M_i are positive semi-definite matrices and $\sum_i M_i = \mathbb{I}$. As with the projective measurement, the probability of an outcome i is $\mathrm{Tr}\left[M_i \rho\right]$. However, the post-measurement state depends on the exact implementation of the POVM measurement, which is often not specified.

In a real-world setting, a quantum computer can only implement a few basic unitary matrices and measurements as direct operations, and more complex operations are built as a *quantum circuit*, which is a combination of these basic building blocks. Any quantum circuit can be described by a Completely Positive Trace-Perserving map, also known as a *quantum channel.* Similarly to the norm we defined on matrices, we define the *induced trace norm* of any operator $\Gamma : \mathcal{D}(\mathcal{H}_A) \to \mathcal{D}(\mathcal{H}_B)$ as

$$\left\|\Gamma\right\|_1 = \max_\rho \left\|\Gamma(\rho)\right\|_1,$$

where the maximum is taken over all density matrices $\rho \in \mathcal{D}(\mathcal{H}_A)$. However, this norm proves not to be satisfactory when quantifying the distinguishability of quantum channels, as it disregards the possibility of side information. To better deal with side information, we will use the diamond norm when dealing with quantum channels. The *diamond norm* of an operator $\Gamma : \mathcal{B}(\mathcal{H}_A) \to \mathcal{B}(\mathcal{H}_B)$ is defined as

$$\|\Gamma\|_\diamond = \left\|\Gamma \otimes \mathbb{I}^{A'}\right\|_1,$$

where $\mathcal{H}_{A'}$ is some Hilbert space of equal dimension as $\mathcal{H}_A$.

To deal with states of varying sizes, we define an abstraction called a *quantum register.* A quantum register R stores a state $\rho \in \mathcal{D}(\mathcal{H}_R)$. In notation, registers are used much in the same way as variables in the classical setting. When dealing with states of different sizes, a register can also refer to a family of registers of various sizes, allowing the register to essentially change in size. Registers may be appended, i.e. the register AB holds a state $\rho \in \mathcal{D}(\mathcal{H}_A \otimes \mathcal{H}_B)$.

A *quantum algorithm* is a uniform family of quantum circuits, allowing for varying input state sizes. If a quantum algorithm A takes a register R as input and outputs a register O, we denote this $O \leftarrow A(R)$. To clarify the input-output registers of such an algorithm we use the notation $A^{R \to O}$ where it is needed.

The amount of direct operations required to implement a quantum algorithm is referred to as the runtime of that algorithm. We define the class of *quantum polynomial time (*QPT*)* algorithms as the class of quantum algorithms where the runtime scales at most polynomially in the size of the input register.

2.2.3 Quantum Cryptography

In the quantum setting, we define security schemes in a very similar way to the classical setting, however with quantum registers to hold the plaintext and ciphertext.

Definition 2.5. *A* symmetric-key quantum encryption scheme (SKQES) *is a triple* (KeyGen, Enc, Dec)*, where*

- KeyGen *is a* QPT *algorithm that given a security parameter $\kappa \in \mathbb{N}$ outputs a key k,*
- Enc *is a* QPT *algorithm which takes as input a classical key k and a quantum state in register M and outputs a quantum state in register C,*
- Dec *is a* QPT *algorithm which takes as input a classical key k and a quantum state in register C and outputs a quantum state in register M or $|\bot\rangle\langle\bot|^{\bot}$,*

such that $\left\|\mathsf{Dec}_k \circ \mathsf{Enc}_k - \mathrm{id}^{M \to M \oplus \bot}\right\|_\diamond \leq \mathrm{negl}(\kappa)$ *for all* $k \leftarrow \mathsf{KeyGen}(1^\kappa)$*.*

It is implicit that $|M| \leq |C| \leq 2^{q(\kappa)}$ for some polynomial q. Furthermore, we only consider *fixed-length* schemes, which means $|M|$ is a fixed function of κ. Lastly, we adopt the convention that every honest party applies the measurement $\{|\bot\rangle\langle\bot|, \mathbb{I} - |\bot\rangle\langle\bot|\}$ after running Dec, and denote with $\mathsf{Dec}_k(C) \neq \bot$ the event that this measurement did not measure $|\bot\rangle\langle\bot|$ and thus produced a valid plaintext. Because of this convention, we often state that the output space of Dec is $\mathcal{D}(\mathcal{H}_M)$ although it is technically $\mathcal{D}(\mathcal{H}_M \oplus \mathcal{H}_\bot)$, where $\mathcal{H}_\bot = \mathbb{C}|\bot\rangle$.

Definition 2.6. *A* public-key quantum encryption scheme (PKQES) *is a triple* (KeyGen, Enc, Dec)*, where*

- KeyGen *is a* QPT *algorithm that given a security parameter* $\kappa \in \mathbb{N}$ *outputs a keypair* $(\mathsf{pk}, \mathsf{sk})$*,*
- Enc *is a* QPT *algorithm which takes as input a classical key* pk *and a quantum state in register* M *and outputs a quantum state in register* C*,*
- Dec *is a* QPT *algorithm which takes as input a classical key* sk *and a quantum state in register* C *and outputs a quantum state in register* M *or* $|\bot\rangle\langle\bot|^{\bot}$*,*

such that $\left\|\mathsf{Dec}_{\mathsf{sk}} \circ \mathsf{Enc}_{\mathsf{pk}} - \mathrm{id}^{M \to M\oplus\bot}\right\|_{\diamond} \leq \mathrm{negl}(\kappa)$ *for all* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^{\kappa})$*.*

For the public-key setting, we make the same assumption on the error behaviour of $\mathsf{Dec}_{\mathsf{sk}}$ and the fixed length of the plaintext and ciphertext spaces as we made in the symmetric-key setting.

When taking security principles from the classical world to the quantum world, it is often much easier to implement them in the symmetric-key setting and afterwards bring them to the public-key setting by using a post-quantum scheme with the desired properties. To do this, we make use of a quantum-classical hybrid construction like used in [AGM18, AM17, AGM21]. The idea is similar to the classical technique of hybrid encryption. We construct a PKQES by encrypting each plaintext with a quantum symmetric-key scheme and encrypting the key using a classical PKES.

Construction 2.7. Let $\Pi^{\mathsf{Qu}} = (\mathsf{KeyGen}^{\mathsf{Qu}}, \mathsf{Enc}^{\mathsf{Qu}}, \mathsf{Dec}^{\mathsf{Qu}})$ be a (Quantum) SKQES and $\Pi^{\mathsf{Cl}} = (\mathsf{KeyGen}^{\mathsf{Cl}}, \mathsf{Enc}^{\mathsf{Cl}}, \mathsf{Dec}^{\mathsf{Cl}})$ a (Classical) PKES. We define the hybrid scheme $\Pi^{\mathsf{Hyb}}[\Pi^{\mathsf{Qu}}, \Pi^{\mathsf{Cl}}] = (\mathsf{KeyGen}^{\mathsf{Hyb}}, \mathsf{Enc}^{\mathsf{Hyb}}\mathsf{Dec}^{\mathsf{Hyb}})$ as follows. We set $\mathsf{KeyGen}^{\mathsf{Hyb}} = \mathsf{KeyGen}^{\mathsf{Cl}}$. The encryption algorithm $\mathsf{Enc}^{\mathsf{Hyb}}_{\mathsf{pk}}$, on input quantum register X,

1. generates a key $k \leftarrow \mathsf{KeyGen}^{\mathsf{Qu}}(1^{n(\mathsf{pk})})$, and
2. outputs the pair $(\mathsf{Enc}^{\mathsf{Qu}}_k(X), \mathsf{Enc}^{\mathsf{Cl}}_{\mathsf{pk}}(k))$ to register C, where the first element of the pair is a quantum state and the second a classical state.

Decryption is done in the obvious way, by first decrypting the second part of the ciphertext using $\mathsf{Dec}^{\mathsf{Cl}}_{\mathsf{sk}}$ to obtain the one-time key k', and then decrypting the first part using $\mathsf{Dec}^{\mathsf{Qu}}_{k'}$.

Often more knowledge is required of the construction of a quantum encryption scheme to be able to analyse its impact on a state. For this, we make use of a general characterization of quantum encryption schemes, which we will need in both our plaintext-awareness and non-malleability definitions.

Lemma 2.8 (Lemma 5.7 in [AGM21])**.**
Let $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *be a PKQES, then* Enc *and* Dec *have the following form, for all* $k := (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}$*:*

$$\left\|\mathsf{Enc}_k - V_k((\cdot)^M \otimes \sigma_k^T)V_k^{\dagger}\right\|_{\diamond} \leq \varepsilon$$

$$\left\|\mathsf{Dec}_k(V_k P^T_{\sigma_k}(V_k^{\dagger}(\cdot)^C V_k)P^T_{\sigma_k} V_k^{\dagger}) - \mathrm{Tr}_T\left[P^T_{\sigma_k}(V_k^{\dagger}(\cdot)^C V_k)P^T_{\sigma_k}\right]\right\|_{\diamond} \leq \varepsilon.$$

Here T *is a register such that* $C \cong MT$*,* σ_k *is a state on register* T*,* V_k *is a unitary* $\varepsilon \leq \mathrm{negl}(\kappa)$*. Furthermore,* P_{σ_k} *is an orthogonal projectors such that* $\|P_{\sigma_k}\sigma_k P_{\sigma_k} - \sigma_k\|_{\diamond} \leq \varepsilon$ *and* $\bar{P}_{\sigma_k} = \mathbb{I} - P_{\sigma_k}$*.*

Furthermore, for every k *there exists a probability distribution* p_k *and a family of quantum states* $|\psi_{k,r}\rangle^T$ *such that* Enc_k *is* ε*-close to the following algorithm:*

1. *sample* $r \xleftarrow{p_k} \{0,1\}^{\log|T|}$;

2. *apply the map* $\mathsf{Enc}_{k;r}(X^M) = V_k(X^M \otimes \psi_{k,r}^T)V_k^\dagger$.

Note that while it is proven that every scheme can be described in this way, it is not currently known that this decomposition is always efficient.

Condition 2.9 (Condition 5.9 in [AGM21])**.** Let Π be a PKQE, and let $p_k, |\psi_{k,r}\rangle$ and V_k be as in Lemma 2.8. We say Π satisfies Condition 2.9 if there exist QPTs for (i.) sampling from p_k, (ii.) preparing $|\psi_{k,r}\rangle$, and (iii.) implementing V_k on inputs of the form $\rho \otimes |\psi_{k,r}\rangle\langle\psi_{k,r}|$, and this holds for all but a negligible fraction of k and r.

In this thesis, we restrict ourselves to schemes that satisfy Condition 2.9. We are not currently aware of any schemes or constructions where this condition does not hold, but there is also no proof that an efficient decomposition always exists.

CHAPTER 3

Strong Designated-Verifier Signature Schemes: Sender Privacy in the Multi-Party Setting

This chapter is based on the paper "On SDVS Sender Privacy in the Multi-Party Setting" [Wie21].

Strong designated verifier signature schemes rely on sender-privacy to hide the identity of the creator of a signature to all but the intended recipient. This property can be invaluable in, for example, the context of deniability, where the identity of a party should not be deducible from the communication sent during protocol execution. In this work, we explore the technical definition of sender-privacy and extend it from a 2-party setting to an n-party setting. Afterwards, we show in which cases this extension provides stronger security and in which cases it does not.

3.1 Introduction

Digital signatures have many useful applications in our everyday lives, from message authentication to software updates. In many cases, they provide a publicly verifiable way of proving the authenticity of a message. However, sometimes it is desired to prove authenticity only to the intended receiver, or designated verifier, of a message. Designated verifier signature (DVS) schemes were constructed for this reason, to allow for the signing of a message in such a way that the receiver would be fully convinced of its authenticity, but to third-party observers, the validity of the signature could be denied. Strong designated verifier signature (SDVS) schemes are the refinement of this idea, with the additional restraint that no one but the creator and the designated verifier should be able to deduce from a signature who was the creator. While this concept has been studied extensively and is interpreted intuitively in the same way by many, the technical definitions for the property separating DVS schemes from SDVS schemes, known as sender-privacy, vary. In this work we analyze and generalize the definitions in current literature and aim to provide a universally applicable way to define this property, particularly focusing on the n-party setting. Furthermore, we prove that our general form of sender-privacy can be achieved by combining weaker forms of sender-privacy with non-transferability or unforgeability.

3.1.1 Related Work

Chaum and van Antwerpen first introduced undeniable signatures in [CA90], which required interaction between the signer and verifier. In 1996 this requirement was removed by Chaum [Cha96] and by Jakobsson et al. [JSI96] separately, who introduced designated verifier signatures. These formal definitions were later refined by Saeednia et al. [SKM04]. Rivest et al. introduced ring signatures in [RST01], which can be interpreted as DVS when a ring size of 2 is used, although not SDVS.

An important step was made when Laguillaumie and Vergnaud formalised *sender-privacy*, the property separating DVS from SDVS, in [LV05]. The notion of SDVS was further refined to Identity-Based SDVS by Susilo et al. [SZM04], where all private keys are issued using a master secret key (i.e. central authority). For this setting, sender-privacy was later formalized in a game-based manner by Huang et al. [Hua+06].

3.1.2 Summary of Contributions

The main objective of the chapter is to present a definition of sender-privacy that is usable in the multi-party setting without requiring additional proofs. This will be critical when we apply it in Chapter 4. To achieve this, we first survey the definitions of sender-privacy that are present in the literature in Section 3.2. Here we also show how the definition of designated verifier signatures got split into non-transferability and sender-privacy.

In Section 3.3 we show how the definitions from the literature differ from each other and what their potential shortcomings are. Here we also present the multi-party sender-privacy definition and go deeper into which oracles are used.

Definition 3.1 (Simplified)**.** *A designated verifier signature scheme is n-party sender-private if an adversary interacting with n parties with oracle access to the signing, simulating and verification procedures has negligible advantage in deducing who the signing party of a challenge signature is. Here the challenge signing party is always one of two fixed parties known to the adversary.*

In Section 3.4 we show that a number of alternative definitions are equivalent to the one above.

Definition 3.2 (Simplified)**.** *A designated verifier signature scheme is n-party random-challenge sender-private if an adversary interacting with n parties with oracle access to the signing, simulating and verification procedures has negligible advantage in deducing who the signing party of a challenge signature is. Here the challenge signing party is always one of the n parties known to the adversary.*

Definition 3.3 (Simplified)**.** *A designated verifier signature scheme is n-party adverserial-challenge sender-private if an adversary interacting with n parties with oracle access to the signing, simulating and verification procedures has negligible advantage in deducing who the signing party of a challenge signature is. Here the challenge signing party is always one of two parties chosen by adversary.*

Theorem 3.4. *Definitions 3.14, 3.15, and 3.16 are equivalent up to polynomial differences in the advantages.*

In order to integrate our definition with the current literature, we present in Section 3.5 some settings in which definitions with different oracles or numbers of parties coincide.

Theorem 3.5 (Simplified)**.** *For any scheme that is strongly unforgeable, the definitions of sender-privacy with or without access to a verifications oracle are equivalent.*

Theorem 3.6 (Simplified)**.** *For any scheme that is strongly unforgeable and computationally non-transferable, the definitions of* 2*-party sender-privacy and* n*-party sender-privacy are equivalent.*

3.2 Preliminaries

3.2.1 Historical Definitions

The original definitions for strong verifier designation are a combination of what we currently distinguish as *non-transferability* and *sender-privacy.* The following definitions are the initial attempts at defining strong verifier designation, and in their respective papers, they are accompanied by definitions for (non-strong) verifier designation, which are very much in line with the intuition behind non-transferability.

Definition 3.7 ([JSI96])**.** *Let* $(\mathcal{P}_A, \mathcal{P}_B)$ *be a protocol for Alice to prove the truth of the statement* Ω *to Bob. We say that Bob is a* JSI strong designated verifier *if, for any protocol* $(\mathcal{P}_A, \mathcal{P}_B, \mathcal{P}_C, \mathcal{P}_D)$ *involving Alice, Bob, Cindy, and Dave, by which Dave proves the truth of some statement* θ *to Cindy, there is another protocol* $(\mathcal{P}_C, \mathcal{P}'_D)$ *such that Dave can perform the calculations of* $\mathcal{P}'_D$*, and Cindy cannot distinguish transcripts of* $(\mathcal{P}_A, \mathcal{P}_B, \mathcal{P}_C, \mathcal{P}_D)$ *from those of* $(\mathcal{P}_C, \mathcal{P}'_D)$*.*

In the above definition, the intuition is that Alice proves a statement to Bob, e.g. the authenticity of a given message. Dave observes this interaction and tries to prove this observation to Cindy. However, strong designation in this sense prevents him from doing so, as any proof he could present to Cindy is indistinguishable (to Cindy) from a simulated proof.

Definition 3.8 ([SKM04])**.** *Let* $\mathcal{P}(A, B)$ *be a protocol for Alice to prove the truth of the statement* Ω *to Bob. We say that* $\mathcal{P}(A, B)$ *is a* SKM strong designated verifier proof *if anyone can produce identically distributed transcripts that are indistinguishable from those of* $\mathcal{P}(A, B)$ *for everybody, except Bob.*

In later work, we see the definition for strong verifier designation split. Non-transferability captures the notion that the verifier can produce signatures from anyone designated to himself, thus ensuring that no signature provides proof of signer-verifier interaction for third parties. Sender-privacy adds to this that, from a signature, one cannot deduce the sender, thus allowing no third-party observer to use a signature to plausibly deduce that an interaction between two parties happened.

Definition 3.9. *A* DVS *scheme* $\Pi = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Simulate})$ *is* computationally non-transferable *if for any adversary* $\mathcal{A}$*,*

$$\mathsf{Adv}^{\mathsf{NT}}_{\Pi,\mathcal{A}}(\kappa) = \Pr_{b \in \{0,1\}}\left[\mathsf{G}^{\mathsf{NT}}_{\Pi,\mathcal{A}}(\kappa, b) = b\right] - \frac{1}{2} \leq \mathrm{negl}(\kappa),$$

where the game $\mathsf{G}^{\mathsf{NT}}_{\Pi,\mathcal{A}}$ *is defined as follows:*

Definition 3.10. *A* DVS $\Pi = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Simulate})$ *is* statistically non-transferable *if for all* S*,* V*, and* m*,* $\mathsf{Sign}_{S \to V}(m)$ *and* $\mathsf{Simulate}_{S \to V}(m)$ *are statistically indistinguishable distributions.*

Game 1: $\mathsf{G}^{\mathsf{NT}}_{\Pi,\mathcal{A}}(\kappa, b)$

```
$\mathsf{params} \leftarrow \mathsf{Setup}$
$(\mathsf{pk}_S, \mathsf{sk}_S) \leftarrow \mathsf{KeyGen}$
$(\mathsf{pk}_V, \mathsf{sk}_V) \leftarrow \mathsf{KeyGen}$
$(m^*, \mathsf{state}) \leftarrow \mathcal{A}(1, \mathsf{params}, \mathsf{pk}_S, \mathsf{sk}_S, \mathsf{pk}_V, \mathsf{sk}_V)$
if $b = 0$ then
    $\sigma^* = \mathsf{Sign}(\mathsf{sk}_S, \mathsf{pk}_S, \mathsf{pk}_V, m^*)$
else
    $\sigma^* = \mathsf{Simulate}(\mathsf{sk}_V, \mathsf{pk}_V, \mathsf{pk}_S, m^*)$
$b' \leftarrow \mathcal{A}(2, \mathsf{state}, \sigma^*)$
Output $b'$
```

For sender-privacy, many slightly different definitions are presented in the literature. Many follow the form of Game 2, but with different oracles presented to the adversary. Note that this game is a generalized definition designed to be instantiated with a set of oracles $\mathcal{O}$ to form the specific definitions found in the literature. Besides the oracles, the game takes as parameters the security parameter κ, the number of parties n, and the challenge party index c. For each $i \in [n]$, party i is denoted P_i. P_n is designated as the verifier for the challenge. In much of the literature, this game is played with 3 parties: S_0, S_1, and V, who would here correspond with P_0, P_1, and P_2 respectively in the $n = 2$ setting.

Game 2: $\mathsf{G}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A},\mathcal{O}}(\kappa, n, c)$, the generalized game for sender-privacy.

```
$\mathsf{params} \leftarrow \mathsf{Setup}$
$(\mathsf{pk}_{P_0}, \mathsf{sk}_{P_0}) \leftarrow \mathsf{KeyGen}; \ldots; (\mathsf{pk}_{P_n}, \mathsf{sk}_{P_n}) \leftarrow \mathsf{KeyGen}$
$(m^*, \mathsf{state}) \leftarrow \mathcal{A}^{\mathcal{O}^{(1)}_{sign}, \mathcal{O}^{(1)}_{veri}, \mathcal{O}^{(1)}_{sim}}(1, \mathsf{params}, \mathsf{pk}_{P_0}, \ldots, \mathsf{pk}_{P_n})$
$\sigma^* = \mathsf{Sign}_{P_c \to P_n}(m^*)$
$c' \leftarrow \mathcal{A}^{\mathcal{O}^{(2)}_{sign}, \mathcal{O}^{(2)}_{veri}, \mathcal{O}^{(2)}_{sim}}(2, \mathsf{state}, \sigma^*)$
Output $c'$
```

Definition 3.11 ([Hua+06])**.** *A* DVS $\Pi = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Simulate})$ *is a* Hua-strong DVS *if it is statistically non-transferable and for any* PPT *adversary* $\mathcal{A}$*,*

$$\mathsf{Adv}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A}}(\kappa) = \Pr_{c \leftarrow \{0,1\}}\left[\mathsf{G}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A}}(\kappa, 2, c) = c\right] - \frac{1}{2} \leq \operatorname{negl}(\kappa),$$

where $\mathsf{G}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A}}$ *is played with the following oracles:*

- $\mathcal{O}^{(1)}_{sign}$*: Upon input* (m_i, d_i) *returns* $\mathsf{Sign}_{P_{d_i} \to P_2}(m_i)$ *if* $d_i \in \{0, 1\}$ *and* $\bot$ *otherwise.*
- $\mathcal{O}^{(2)}_{sign}$*: Upon input* (m_i, d_i) *returns* $\mathsf{Sign}_{P_{d_i} \to P_2}(m_i)$ *if* $d_i \in \{0, 1\}$ *and* $m_i \neq m^*$*,and* $\bot$ *otherwise.*
- $\mathcal{O}^{(1)}_{veri}$*: Upon input* (σ_i, m_i, d_i) *returns* $\mathsf{Verify}_{P_{d_i} \to P_2}(m_i)$ *if* $d_i \in \{0, 1\}$ *and* $\bot$ *otherwise.*

- $\mathcal{O}^{(2)}_{veri}$: *Upon input* (σ_i, m_i, d_i) *returns* $\mathsf{Verify}_{P_{d_i}\rightarrow P_2}(m_i)$ *if* $d_i \in \{0,1\}$, $\sigma_i \neq \sigma^*$, *and* $m_i \neq m^*$,*and* $\perp$ *otherwise.*
- $\mathcal{O}^{(1)}_{sim} = \mathcal{O}^{(2)}_{sim} = \emptyset$

In [Hua+06], Huang et al. define signer-privacy for identity-based-SDVS, a similar type of DVS where all keypairs are issued by a central authority. Here, they allow signing queries from any party to any party, and the adversary is allowed to choose the two signer and the verifier parties. We explore this option for SDVS in Definition 3.16.

3.3 Bringing Sender-Privacy to the Multi-Party Setting

Sender-privacy is meant to provide security in the setting where an eavesdropping adversary is trying to detect the identity of the sender of a signature. In the previously presented definitions, this is modelled by a coin flip between two senders, with a fixed verifier. This way of defining sender-privacy is similar to key-privacy in public-key cryptography [Bel+01]. The key difference here is that public-key ciphertexts are only related to one keypair, the receivers. However, designated verifier signatures are bound to two parties, the signer and the designated verifier. This creates the problem that the naive way of defining sender-privacy does not cover any attacks that require multiple parties. In key-privacy, any adversary requiring n parties for their attack can perform this attack in the two-party setting by simulating the other $n-2$ parties themself. However, in the case of SDVS schemes, this is not necessarily possible. The adversary could be unable to create signatures signed by one of the two challenge parties with their simulated parties as the verifier, as is depicted in Figure 3.1. In particular if one does not have statistical non-transferability, this might pose a problem. For this reason, we explicitly shape our definition for the multi-party setting. We explore settings where this is a non-issue in Section 3.5.

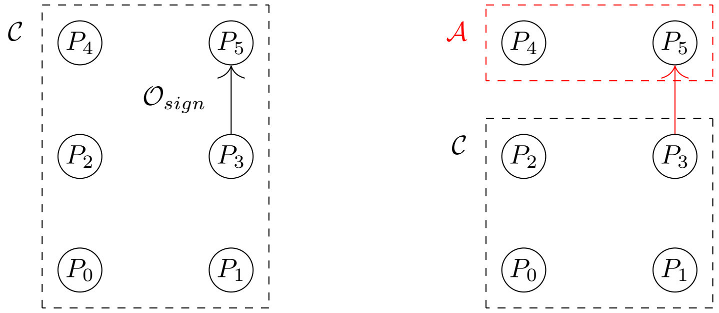

Figure 3.1: Left: a 6-party setting where the adversary requests a signature using an oracle, Right: a 4-party setting where the adversary simulates another 2 parties but is now unable to obtain the same signature as on the left.

3.3.1 Oracles

Many different interpretations exist in the literature of what oracles the adversary should be given access to. The key choices here are whether (1) a simulation oracle should be provided, (2) a verification oracle should be provided, and (3) whether the adversary should still have access to the oracles after the challenge has been issued. Whereas the precise attacker model might depend on the context and our framework allows us to capture this, we here choose to focus on the strongest level of security, by providing the adversary with as much as possible without trivially breaking the challenge.

Definition 3.12. *For any n, let the* standard n-sender $\mathsf{SendPriv}$-oracles *denote:*

- $\mathcal{O}^{(1)}_{sign} = \mathcal{O}^{(2)}_{sign}$*: Upon input* (m_i, s, v) *returns* $\sigma_i := \mathsf{Sign}_{P_s \to P_v}(m_i)$ *if* $s, v \in [n]$ *and* $\perp$ *otherwise.*
- $\mathcal{O}^{(1)}_{sim} = \mathcal{O}^{(2)}_{sim}$*: Upon input* (m_i, s, v) *returns* $\sigma_i := \mathsf{Simulate}_{P_s \to P_v}(m_i)$ *if* $s, v \in [n]$ *and* $\perp$ *otherwise.*
- $\mathcal{O}^{(1)}_{veri}$*: Upon input* (m_i, σ_i, s, v) *returns* $\mathsf{Verify}_{P_s \to P_v}(m_i, \sigma_i)$ *if* $s, v \in [n]$ *and* $\perp$ *otherwise.*
- $\mathcal{O}^{(2)}_{veri}$*: Upon input* (m_i, σ_i, s, v) *returns* $\mathsf{Verify}_{P_s \to P_v}(m_i, \sigma_i)$ *if* $s, v \in [n]$ *and* $\sigma_i \neq \sigma^*$*, and* $\perp$ *otherwise.*

Note that the oracles make use of an implicit ordering of the parties. This makes no difference in any real-world application, but for constructing proofs we also define a set of oracles that allows this ordering to be hidden by a permutation.

Definition 3.13. *For any set of oracles for* $\mathsf{G}^{\mathsf{SendPriv}}$ *and any permutation* π *define the permuted oracles as follows, where* $b \in \{0, 1\}$*:*

- $\mathcal{O}^{(\pi,b)}_{sign}$*: On input* (m_i, s, v) *output* $\mathcal{O}^{(b)}_{sign}(m_i, \pi(s), \pi(v))$
- $\mathcal{O}^{(\pi,b)}_{sim}$*: On input* (m_i, s, v) *output* $\mathcal{O}^{(b)}_{sim}(m_i, \pi(s), \pi(v))$
- $\mathcal{O}^{(\pi,b)}_{veri}$*: On input* (m_i, σ_i, s, v) *output* $\mathcal{O}^{(b)}_{veri}(m_i, \sigma_i, \pi(s), \pi(v))$

3.3.2 Definition

Taking all these things into consideration, we can now craft a definition of sender-privacy. This definition is more in line with current research in ID-based-SDVS research such as [Hua+11].

Definition 3.14. *A* DVS *scheme* Π *is* n-party sender-private with respect to $\mathcal{O}$ *if for any adversary* $\mathcal{A}$*,*

$$\mathsf{Adv}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A},\mathcal{O}}(\kappa, n) = \Pr_{c \leftarrow \{0,1\}} \left[\mathsf{G}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A}}(\kappa, n, c) = c\right] - \frac{1}{2} \leq \mathrm{negl}(\kappa).$$

A DVS *scheme is* n-party sender-private *if it is n-party sender-private with respect to the standard n-sender* $\mathsf{SendPriv}$*-oracles.*

3.4 Alternative Definitions

In this section, we look at possible alternative definitions that one could consider equally valid generalizations of the 2-party setting to the n-party setting. For example, in the 2-party setting, we pick the challenge uniformly at random between the two possible senders, thus one could consider picking uniformly at random from n senders in the n-party setting.

Definition 3.15. *A* DVS *scheme is* n-party random-challenge sender-private with respect to $\mathcal{O}$ *if for any adversary* $\mathcal{A}$*,*

$$\mathsf{Adv}^{nr\mathsf{SendPriv}}_{\Pi,\mathcal{A},\mathcal{O}}(\kappa, n) = \Pr_{c \leftarrow [n-1]} \left[\mathsf{G}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A},\mathcal{O}}(\kappa, n, c) = c\right] - \frac{1}{n} \leq \mathrm{negl}(\kappa).$$

A DVS *scheme is* n-party random-challenge sender-private *if it is n-party random-challenge sender-private with respect to the standard n-sender* SendPriv*-oracles.*

Furthermore, one could strengthen the definition even more by allowing the adversary to choose which two senders the challenge is chosen from and which party is the verifier.

Game 3: $\mathsf{G}^{\mathsf{ChosenSendPriv}}_{\Pi,\mathcal{A},\mathcal{O}}(\kappa, n, c)$

```
$\mathsf{params} \leftarrow \mathsf{Setup}$
$(\mathsf{pk}_{P_0}, \mathsf{sk}_{P_0}) \leftarrow \mathsf{KeyGen}; \ldots; (\mathsf{pk}_{P_n}, \mathsf{sk}_{P_n}) \leftarrow \mathsf{KeyGen}$
$(m^*, s_0, s_1, r, \mathsf{state}) \leftarrow \mathcal{A}^{\mathcal{O}^{(1)}_{sign}, \mathcal{O}^{(1)}_{veri}, \mathcal{O}^{(1)}_{sim}}(1, \mathsf{params}, \mathsf{pk}_{P_0}, \ldots, \mathsf{pk}_{P_n})$
$\sigma^* = \mathsf{Sign}_{P_{s_c} \to P_r}(m^*)$
$c' \leftarrow \mathcal{A}^{\mathcal{O}^{(2)}_{sign}, \mathcal{O}^{(2)}_{veri}, \mathcal{O}^{(2)}_{sim}}(2, \mathsf{state}, \sigma^*)$
Output $c'$
```

Definition 3.16. *A* DVS *scheme is* n-party adversarial-challenge sender-private with respect to $\mathcal{O}$ *if for any adversary* $\mathcal{A}$*,*

$$\mathsf{Adv}^{\mathsf{ChosenSendPriv}}_{\Pi,\mathcal{A},\mathcal{O}}(\kappa, n) = \Pr_{c \leftarrow \{0,1\}} \left[\mathsf{G}^{\mathsf{ChosenSendPriv}}_{\Pi,\mathcal{A}}(\kappa, n, c) = c\right] - \frac{1}{2} \leq \mathrm{negl}(\kappa).$$

A DVS *scheme is* n-party adversarial-challenge sender-private *if it is n-party adversarial-challenge sender-private with respect to the standard n-sender* SendPriv*-oracles.*

3.4.1 Relations

As one might expect, the above-defined alternative definitions relate strongly to the main definition, Definition 3.14. In fact, in this section, we show that they are equivalent up to polynomial differences in the advantages.

For the universally random challenge, this can be done by simply only considering the cases where the challenge is P_0 or P_1, which will be the case 2 out of n times, giving us a loss in the advantage of a factor $\frac{2}{n}$.

Theorem 3.17. *For any adversary* $\mathcal{A}$*,* DVS *scheme* Π*, and set of oracles* $\mathcal{O}$*,*

$$\frac{2}{n} \cdot \mathsf{Adv}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A},\mathcal{O}}(\kappa, n) \leq \mathsf{Adv}^{nr\mathsf{SendPriv}}_{\Pi,\mathcal{A},\mathcal{O}}(\kappa, n).$$

Proof.

$$
\begin{aligned}
&\mathsf{Adv}^{nr\mathsf{SendPriv}}_{\Pi,\mathcal{A}}(\kappa, n)\\
&= \Pr_{c \leftarrow [n-1]}\left[\mathsf{G}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A}}(\kappa, n, c) = c\right] - \frac{1}{n}\\
&= \frac{2}{n}\Pr_{c \leftarrow [1]}\left[\mathsf{G}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A}}(\kappa, n, c) = c\right] + \frac{n-2}{n}\Pr_{c \leftarrow [2,n-1]}\left[\mathsf{G}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A}}(\kappa, n, c) = c\right] - \frac{1}{n}\\
&= \frac{2}{n}\left(\Pr_{c \leftarrow [1]}\left[\mathsf{G}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A}}(\kappa, n, c) = c\right] - \frac{1}{2}\right) + \frac{n-2}{n}\Pr_{c \leftarrow [2,n-1]}\left[\mathsf{G}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A}}(\kappa, n, c) = c\right]\\
&= \frac{2}{n}\cdot\mathsf{Adv}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A}}(\kappa) + \frac{n-2}{n}\Pr_{c \leftarrow [2,n-1]}\left[\mathsf{G}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A}}(\kappa, n, c) = c\right]\\
&\geq \frac{2}{n}\cdot\mathsf{Adv}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A},\mathcal{O}}(\kappa, n),
\end{aligned}
$$

where $[2, n-1] = \{2, \ldots, n-1\}$. □

Theorem 3.18. *For any adversary $\mathcal{A}$, set of oracles $\mathcal{O}$ and* DVS *scheme Π, there exists an adversary $\mathcal{B}$ such that*

$$\frac{1}{2}\mathsf{Adv}^{nr\mathsf{SendPriv}}_{\Pi,\mathcal{A},\mathcal{O}}(\kappa, n) \leq \mathsf{Adv}^{\mathsf{SendPriv}}_{\Pi,\mathcal{B},\mathcal{O}}(\kappa, n).$$

Proof. Here, we omit the subscripts Π and $\mathcal{O}$ for Adv and G for simplicity. Let $\mathcal{B}$ be defined as in Games 4 and 5. The permutation is used here to hide the indexation of the parties from the adversary. Note that applying a permutation π in this fashion is equivalent to generating the keypairs in the order $\pi^{-1}(0)\ldots\pi^{-1}(n)$ and since these are i.i.d. samples the order of their generation does not affect the winning probability of $\mathcal{A}$. However, it guarantees that the winning probability of $\mathcal{A}$ is the same for every c. Note that here we use $\Pr_\pi$ to indicate the uniform probability over all $\pi : [n] \mapsto [n]$ such that

Game 4: $\mathcal{B}^{\mathcal{O}^{(1)}_{sign},\mathcal{O}^{(1)}_{veri},\mathcal{O}^{(1)}_{sim}}(1, \mathsf{params}, \mathsf{pk}_{P_0}, \ldots, \mathsf{pk}_{P_n})$

```
Pick a random permutation $\pi : [n] \mapsto [n]$ such that $\pi(n) = n$
$(m^*, \mathsf{state}) \leftarrow \mathcal{A}^{\mathcal{O}^{(\pi,1)}_{sign},\mathcal{O}^{(\pi,1)}_{veri},\mathcal{O}^{(\pi,1)}_{sim}}(1, \mathsf{params}, \mathsf{pk}_{P_{\pi(0)}}, \ldots, \mathsf{pk}_{P_{\pi(n)}})$
Output $(m^*, (\pi, \mathsf{state}))$
```

Game 5: $\mathcal{B}^{\mathcal{O}^{(2)}_{sign},\mathcal{O}^{(2)}_{veri},\mathcal{O}^{(2)}_{sim}}(2, \mathsf{state}', \sigma^*)$

```
Parse $\mathsf{state}'$ as $(\pi, \mathsf{state})$
$c' \leftarrow \mathcal{A}^{\mathcal{O}^{(\pi,2)}_{sign},\mathcal{O}^{(\pi,2)}_{veri},\mathcal{O}^{(\pi,2)}_{sim}}(2, \mathsf{state}, \sigma^*)$
if $\pi(c') \in \{0, 1\}$ then
    Output $\pi(c')$
else
    Output 0
```

$\pi(n) = n$.

$$\begin{aligned}
&\mathsf{Adv}^{\mathsf{SendPriv}}_{\mathcal{B}}(\kappa, n) \\
&= \Pr_{c \leftarrow [1]} \left[\mathsf{G}^{\mathsf{SendPriv}}_{\mathcal{B}}(\kappa, n, c) = c\right] - \frac{1}{2} \\
&= \Pr_{c \leftarrow [1], \pi} \left[\mathsf{G}^{\mathsf{SendPriv}}_{\mathcal{A}}(\kappa, n, \pi^{-1}(c)) = \pi^{-1}(c)\right] \\
&\qquad + \frac{1}{2} \Pr_{\pi} \left[\mathsf{G}^{\mathsf{SendPriv}}_{\mathcal{A}}(\kappa, n, \pi^{-1}(0)) \notin \{\pi^{-1}(0), \pi^{-1}(1)\}\right] - \frac{1}{2} \\
&= \Pr_{c \leftarrow [n-1]} \left[\mathsf{G}^{\mathsf{SendPriv}}_{\mathcal{A}}(\kappa, n, c) = c\right] \\
&\qquad - \frac{1}{2} \Pr_{\pi} \left[\mathsf{G}^{\mathsf{SendPriv}}_{\mathcal{A}}(\kappa, n, \pi^{-1}(0)) \in \{\pi^{-1}(0), \pi^{-1}(1)\}\right] \\
&= \frac{1}{2} \Pr_{c \leftarrow [n-1]} \left[\mathsf{G}^{\mathsf{SendPriv}}_{\mathcal{A}}(\kappa, n, c) = c\right] - \frac{1}{2} \Pr_{\pi} \left[\mathsf{G}^{\mathsf{SendPriv}}_{\mathcal{A}}(\kappa, n, \pi^{-1}(0)) = \pi^{-1}(1)\right] \\
&= \frac{1}{2} \left(\Pr_{c \leftarrow [n-1]} \left[\mathsf{G}^{\mathsf{SendPriv}}_{\mathcal{A}}(\kappa, n, c) = c\right] - \frac{1}{n-1} \Pr_{c \leftarrow [n-1]} \left[\mathsf{G}^{\mathsf{SendPriv}}_{\mathcal{A}}(\kappa, n, c) \neq c\right] \right) \\
&= \frac{1}{2} \left(\frac{n}{n-1} \Pr_{c \leftarrow [n-1]} \left[\mathsf{G}^{\mathsf{SendPriv}}_{\mathcal{A}}(\kappa, n, c) = c\right] - \frac{1}{n-1} \right) \\
&= \frac{n}{2(n-1)} \mathsf{Adv}^{nr\mathsf{SendPriv}}_{\mathcal{A}}(\kappa, n) \geq \frac{1}{2} \mathsf{Adv}^{nr\mathsf{SendPriv}}_{\mathcal{A}}(\kappa, n)
\end{aligned}$$

□

Combining Theorem 3.17 and Theorem 3.18, we see that the advantages for fixed-challenge and random-challenge only differ by at most a linear factor. Thus these definitions are equivalent when considering negligible advantages.

Corollary 3.19. *For any $n \in \mathbb{N}$, an* SDVS *scheme is n-party random-challenge sender-private if and only if it is n-party sender-private.*

For adversarially-chosen challenges, we could try to simply consider only the cases where the adversary chooses P_0 and P_1 as the challenge senders and P_n as the challenge verifier. However, an adversary could be crafted to never choose this exact combination of parties. Thus, we hide the indexation of the parties under a random permutation. This is done only for the proof and has no impact on the actual definition, as all parties' keypairs are i.i.d. samples. Since the adversary does not know this permutation, the chance of them picking these parties is in the order of n^{-3} and thus a loss of this order is incurred in the advantage.

Theorem 3.20. *For any adversary $\mathcal{A}$ and set of oracles $\mathcal{O}$, there exists an adversary $\mathcal{B}$ such that*

$$\frac{2}{n^3 - n} \cdot \mathsf{Adv}^{\mathsf{ChosenSendPriv}}_{\Pi, \mathcal{A}, \mathcal{O}}(\kappa, n) \leq \mathsf{Adv}^{\mathsf{SendPriv}}_{\Pi, \mathcal{B}, \mathcal{O}}(\kappa, n)$$

Proof. Fix $\mathcal{A}$. Let $\mathcal{B}$ be defined as in Game 6 and Game 7.

The permutation is used here to hide the indexation of the parties from the adversary. Note that applying a permutation π in this fashion is equivalent to generating the keypairs in the order $\pi^{-1}(0) \dots \pi^{-1}(n)$ and since these are i.i.d. samples the order of their generation does not affect the winning probability of $\mathcal{A}$. When playing game $\mathsf{G}^{\mathsf{SendPriv}}_{\Pi, \mathcal{B}}$, we can now distinguish two cases:

Game 6: $\mathcal{B}^{\mathcal{O}^{(1)}_{sign},\mathcal{O}^{(1)}_{veri},\mathcal{O}^{(1)}_{sim}}(1, \mathsf{params}, \mathsf{pk}_{P_0}, \dots, \mathsf{pk}_{P_n})$

```
Pick a random permutation $\pi : [n] \mapsto [n]$
$(m^*, s_0, s_1, r, \mathsf{state}) \leftarrow \mathcal{A}^{\mathcal{O}^{(\pi,1)}_{sign},\mathcal{O}^{(\pi,1)}_{veri},\mathcal{O}^{(\pi,1)}_{sim}}(1, \mathsf{params}, \mathsf{pk}_{P_{\pi(0)}}, \dots, \mathsf{pk}_{P_{\pi(n)}})$
if $\pi(s_0) = 0 \wedge \pi(s_1) = 1 \wedge \pi(r) = n$ then
    Output $(m^*, (0, \mathsf{state}))$
else if $\pi(s_0) = 1 \wedge \pi(s_1) = 0 \wedge \pi(r) = n$ then
    Output $(m^*, (1, \mathsf{state}))$
else
    Output $(m^*, (2, \mathsf{state}))$
```

Game 7: $\mathcal{B}^{\mathcal{O}^{(2)}_{sign},\mathcal{O}^{(2)}_{veri},\mathcal{O}^{(2)}_{sim}}(2, \mathsf{state}', \sigma^*)$

```
Parse $\mathsf{state}'$ as $(b, \mathsf{state})$
$c' \leftarrow \mathcal{A}^{\mathcal{O}^{(2)}_{sign},\mathcal{O}^{(2)}_{veri},\mathcal{O}^{(2)}_{sim}}(2, \mathsf{state}, \sigma^*)$
if $b = 0$ then
    Output $c'$
else if $b = 1$ then
    Output $1 - c'$
else
    $c'' \leftarrow \{0, 1\}$
    Output $c''$
```

1. $\{\pi(s_0), \pi(s_1)\} = \{0, 1\}$ and $\pi(r) = n$. Since π is random and unknown to $\mathcal{A}$, this happens with probability $\frac{2(n-2)!}{(n+1)!}$. In this case, $\mathcal{A}$ has chosen P_0 and P_1 as the possible signers and P_n as the verifier, making $\mathsf{G}^{\mathsf{ChosenSendPriv}}_{\Pi,\mathcal{A}}$ and $\mathsf{G}^{\mathsf{SendPriv}}_{\Pi,\mathcal{B}}$ equivalent.

2. Otherwise, $\mathcal{A}$ has chosen different signers or verifiers, in which case $\mathsf{G}^{\mathsf{SendPriv}}_{\Pi,\mathcal{B}}$ becomes equivalent to a random coin flip, with probability $\frac{1}{2}$ of guessing c.

Combining this, we get that

$$\Pr_{c \leftarrow \{0,1\}}\left[\mathsf{G}^{\mathsf{SendPriv}}_{\Pi,\mathcal{B},\mathcal{O}}(\kappa, n, c) = c\right] =$$

$$\frac{2(n-2)!}{(n+1)!} \Pr_{c \leftarrow \{0,1\}}\left[\mathsf{G}^{\mathsf{ChosenSendPriv}}_{\Pi,\mathcal{A},\mathcal{O}}(\kappa, n, c) = c\right] + \left(1 - \frac{2(n-2)!}{(n+1)!}\right)\frac{1}{2}.$$

Thus,

$$\mathsf{Adv}^{\mathsf{SendPriv}}_{\Pi,\mathcal{B},\mathcal{O}}(\kappa, n) = \frac{2(n-2)!}{(n+1)!} \cdot \mathsf{Adv}^{\mathsf{ChosenSendPriv}}_{\Pi,\mathcal{A},\mathcal{O}}(\kappa, n).$$

□

Corollary 3.21. *For any $n \in \mathbb{N}$, an* SDVS *scheme is n-party adversarial-challenge sender-private if and only if it is n-party sender-private.*

Proof. Theorem 3.20 shows that if a scheme is sender-private then it is also adversarial-challenge sender-private since the advantage differs by a factor $\mathcal{O}(n^3)$. The other direction is trivial, as any adversary for sender-privacy can trivially be transformed into an adversary for adversarial-challenge sender-privacy, always outputting $s_0 = 0, s_1 = 1, r = n$, which gives both adversaries the exact same winning probability. □

3.5 Alternative Oracles

In this section we show that one can use other properties of SDVS schemes, e.g. non-transferability and unforgeability, to provide equally strong sender-privacy while giving the adversary weaker oracles. This allows us to more easily prove that existing schemes satisfy our definition. Note that in this section we only consider the cases where the security advantages are negligible. First, we will focus on the verification oracle, showing that they can be removed without impacting the quality of the security when the scheme is unforgeable. Then, we show that the number of parties can be limited to 3 ($n = 2$) when a scheme is both unforgeable and non-transferable.

Definition 3.22. *A* DVS *scheme* $\Pi = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Simulate})$ *is* n-party strongly-unforgeable with respect to $\mathcal{O}$ *if for any adversary* $\mathcal{A}$*,*

$$\mathsf{Adv}^{\mathsf{UF}}_{\Pi,\mathcal{A},\mathcal{O}}(\kappa, n) = \Pr\left[\mathsf{G}^{\mathsf{UF}}_{\Pi,\mathcal{A},\mathcal{O}}(\kappa, n) = \top\right] \leq \mathrm{negl}(\kappa),$$

where the game $\mathsf{G}^{\mathsf{UF}}_{\Pi,\mathcal{A},\mathcal{O}}$ *is defined in Game 8. A* DVS *scheme is* n-party strongly-unforgeable *if it is n-party strongly-unforgeable with respect to* $\mathcal{O}^{(1)}_{sign}$, $\mathcal{O}^{(1)}_{sim}$, $\mathcal{O}^{(1)}_{veri}$ *from the standard n-sender* SendPriv*-oracles.*

Game 8: $\mathsf{G}^{\mathsf{UF}}_{\Pi,\mathcal{A},\mathcal{O}}(\kappa, n)$

```
params ← Setup
$(\mathsf{pk}_{P_0}, \mathsf{sk}_{P_0}) \leftarrow \mathsf{KeyGen}; \ldots; (\mathsf{pk}_{P_n}, \mathsf{sk}_{P_n}) \leftarrow \mathsf{KeyGen}$
$(m^*, \sigma^*, s, v) \leftarrow \mathcal{A}^{\mathcal{O}_{sign}, \mathcal{O}_{veri}, \mathcal{O}_{sim}}(\mathsf{params}, \mathsf{pk}_{P_0}, \ldots, \mathsf{pk}_{P_n})$
if $\mathsf{Verify}_{P_s \to P_v}(m^*, \sigma^*) = 1$ and $\forall i : \sigma^* \neq \sigma_i$ then
    Output $\top$.
else
    Output $\bot$.
```

Theorem 3.23. *Let* $n \in \mathbb{N}$ *and* $\mathcal{O} = \{\mathcal{O}^{(1)}_{sign}, \mathcal{O}^{(2)}_{sign}, \mathcal{O}^{(1)}_{sim}, \mathcal{O}^{(2)}_{sim}, \mathcal{O}^{(1)}_{veri}, \mathcal{O}^{(2)}_{veri}\}$ *be the n-sender standard oracles. Any* DVS *scheme that is n-party sender-private with respect to* $\mathcal{O}' = \{\mathcal{O}^{(1)}_{sign}, \mathcal{O}^{(2)}_{sign}, \mathcal{O}^{(1)}_{sim}, \mathcal{O}^{(2)}_{sim}, \mathcal{O}'^{(1)}_{veri} = \emptyset, \mathcal{O}'^{(2)}_{veri} = \emptyset\}$ *and strongly unforgeable is n-party sender-private (with respect to $\mathcal{O}$).*

Proof. Fix $n \in \mathbb{N}$. Suppose DVS scheme Π is n-party sender-private with respect to $\mathcal{O}' = \{\mathcal{O}^{(1)}_{sign}, \mathcal{O}^{(2)}_{sign}, \mathcal{O}^{(1)}_{sim}, \mathcal{O}^{(2)}_{sim}, \mathcal{O}'^{(1)}_{veri} = \emptyset, \mathcal{O}'^{(2)}_{veri} = \emptyset\}$ and strongly unforgeable, but not n-party sender-private with respect to $\mathcal{O}$. Then there exists an adversary $\mathcal{A}$ such that $\mathsf{Adv}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A},\mathcal{O}}(\kappa) \not\leq \mathrm{negl}(\kappa)$. Let $\mathcal{A}'$ be $\mathcal{A}$, except every query $\mathcal{O}^{(b)}_{veri}(m_i, \sigma_i, s, v)$ is replaced with $\top$ if (m_i, σ_i) was the result of a signing or simulating oracle query and $\bot$ otherwise. Since $\mathcal{A}'$ no longer uses the verification oracles, we have $\mathsf{Adv}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A}',\mathcal{O}} =$

$\mathsf{Adv}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A}',\mathcal{O}'} \leq \text{negl}(\kappa)$, i.e. $\mathcal{A}'$ has the same advantage with respect to $\mathcal{O}$ and $\mathcal{O}'$, as they only differ in the verification oracles.

Now consider the adversary $\mathcal{B}$, who intends to create a forged signature. $\mathcal{B}$ runs $\mathcal{A}$, recording all signing and simulating queries. Whenever $\mathcal{A}$ makes a verification query for a valid signature that was not the result of a signing or simulating query, $\mathcal{B}$ outputs this signature and halts. Note that the only difference in the behaviour of $\mathcal{A}$ and $\mathcal{A}'$ can occur when $\mathcal{A}$ makes such a query. Since the difference between $\mathsf{Adv}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A}',\mathcal{O}}$ and $\mathsf{Adv}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A},\mathcal{O}}$ is more than negligible, we have that such a query occurs with more than negligible probability, giving $\mathcal{B}$ a more than negligible probability of constructing a forgery. This contradicts the fact that Π is strongly unforgeable. □

Theorem 3.24. *Any* DVS *scheme* Π *that is* 2*-party sender-private, strongly unforgeable, and computationally non-transferable is* n*-party sender-private for any* $n \geq 2$*.*

Proof. Suppose a DVS scheme Π is 2-party sender-private, strongly unforgeable, and computationally non-transferable. Assume towards a contradiction that Π is not n-party sender-private for some fixed $n > 2$. By Theorem 3.23, this means Π is also not n-party sender-private with respect to

$$\mathcal{O}' = \{\mathcal{O}^{(1)}_{sign}, \mathcal{O}^{(2)}_{sign}, \mathcal{O}^{(1)}_{sim}, \mathcal{O}^{(2)}_{sim}, \mathcal{O}^{'(1)}_{veri} = \emptyset, \mathcal{O}^{'(2)}_{veri} = \emptyset\}.$$

Thus, there exists and adversary $\mathcal{A}$ such that $\mathsf{Adv}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A},\mathcal{O}'}(\kappa, n) \not\leq \text{negl}(\kappa)$.

Let $\mathcal{A}'(1, \mathsf{params}, \mathsf{pk}_{P_0}, \mathsf{pk}_{P_1}, \mathsf{pk}_{P_2})$ be as follows: First, sample $n-2$ keypairs $(\mathsf{sk}'_{P_2}, \mathsf{pk}'_{P_2}) \ldots (\mathsf{sk}'_{P_{n-1}}, \mathsf{pk}'_{P_{n-1}})$ representing parties $P'_2 \ldots P'_{n-1}$ and set $P'_0 = P_0$, $P'_1 = P_1$, $P'_n = P_2$. Then, run $\mathcal{A}$ with the oracles $\mathcal{O}''$ defined as follows, with $b = 1, 2$:

- $\mathcal{O}^{''(b)}_{veri} = \emptyset$.
- $\mathcal{O}^{''(b)}_{sign}(m_i, s, v)$:
  - If $s, v \in \{0, 1, n\}$, return $\mathcal{O}^{(b)}_{sign}(m_i, \max(2, s), \max(2, v))$.
  - If $s \in \{2, \ldots, n-1\}$ and $v \in [n]$, return $\mathsf{Sign}_{P'_s \to P'_v}(m_i)$.
  - If $s \in \{0, 1, n\}$ and $v \in \{2, \ldots, n-1\}$, return $\mathsf{Simulate}_{P'_s \to P'_v}(m_i)$.
  - Else, return $\bot$.
- $\mathcal{O}^{''(b)}_{sim}(m_i, s, v)$:
  - If $s, v \in \{0, 1, n\}$, return $\mathcal{O}^{(b)}_{sim}(m_i, \max(2, s), \max(2, v))$.
  - If $v \in \{2, \ldots, n-1\}$ and $s \in [n]$, return $\mathsf{Simulate}_{P'_s \to P'_v}(m_i)$.
  - If $v \in \{0, 1, n\}$ and $s \in \{2, \ldots, n-1\}$, return $\mathsf{Sign}_{P'_s \to P'_v}(m_i)$.
  - Else, return $\bot$.

Note that these oracles make use of the fact that one can simulate or sign a signature without, respectively, the sender's or verifier's secret key. Thus we circumvent the issue mentioned in Section 3.3. In the oracles, max is used here to map n to 2, as n and 2 are the challenge verifiers in the n- and 2-party respectively.

Since Π is 2-party sender-private, we have $\mathsf{Adv}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A}',\mathcal{O}''}(\kappa, 2) \leq \text{negl}(\kappa)$. When we replace all oracle calls by their respective functionality, then in the end $\mathsf{G}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A},\mathcal{O}'}(\kappa, n, c)$ and $\mathsf{G}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A}',\mathcal{O}''}(\kappa, 2, c)$ differ, up to relabeling of the parties, only in one way : some Sign executions in $\mathsf{G}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A},\mathcal{O}'}(\kappa,n,c)$ have been replaced by $\mathsf{Simulate}$ in $\mathsf{G}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A}',\mathcal{O}''}(\kappa,2,c)$ and vice versa. Suppose $i \in \mathbb{N}$ such replacements have been made, then for $0 \leq j \leq i$ let $\mathsf{G}_j(\kappa,c)$ be $\mathsf{G}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A},\mathcal{O}'}(\kappa,n,c)$ with only the first j such replacements made, which means $\mathsf{G}_0(\kappa,c) = \mathsf{G}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A},\mathcal{O}'}(\kappa,n,c)$ and $\mathsf{G}_i(\kappa,c) = \mathsf{G}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A}',\mathcal{O}''}(\kappa,2,c)$. Since, by construction, $\Pr[\mathsf{G}_0(\kappa,c)=c] - \frac{1}{2} \not\leq \mathrm{negl}(\kappa)$ and $\Pr[\mathsf{G}_i(\kappa,c)=c] - \frac{1}{2} \leq \mathrm{negl}(\kappa)$, we can fix a lowest k such that $\Pr[\mathsf{G}_k(\kappa,c)=c] - \frac{1}{2} \not\leq \mathrm{negl}(\kappa)$ and $\Pr[\mathsf{G}_{k+1}(\kappa,c)=c] - \frac{1}{2} \leq \mathrm{negl}(\kappa)$. G_k and G_{k+1} differ only in one replacement. Without loss of generality, assume one $\mathsf{Sign}_{P_s \to P_v}(m)$ was replaced by $\mathsf{Simulate}_{P_s \to P_v}(m)$

Now define an adversary $\mathcal{B}$ for G^{NT} as follows: $\mathcal{B}(1, \mathsf{params}, \mathsf{pk}_S, \mathsf{sk}_S, \mathsf{pk}_V, \mathsf{sk}_V)$ picks a $c \in \{0,1\}$ and runs $\mathsf{G}_k(c,\kappa)$, replacing pk_s with pk_S, sk_s with sk_S, pk_v with pk_V, and sk_v with sk_V. This replacement is only a relabeling. The execution of G_k is stopped at the one difference with G_{k+1}, then outputs $(m, (\mathsf{state}, c))$, where state is the current state of G_k and m the message in the replaced Sign. $\mathcal{B}(2, (\mathsf{state}, c), \sigma)$ then continues the execution of G_k with σ as the result of the replaced Sign until G_k outputs c'. $\mathcal{B}$ then outputs 0 if $c = c'$ and 1 otherwise.

Note that in $\mathsf{G}^{\mathsf{NT}}_{\Pi,\mathcal{B}}(0,\kappa)$, i.e. the case where a Sign is used in the non-transferability game, $\mathcal{B}$ plays $\mathsf{G}_k(c,\kappa)$ and in $\mathsf{G}^{\mathsf{NT}}_{\Pi,\mathcal{B}}(1,\kappa)$, $\mathcal{B}$ plays $\mathsf{G}_{k+1}(c,\kappa)$. Thus we have that

$$\Pr_b\left[\mathsf{G}^{\mathsf{NT}}_{\Pi,\mathcal{B}}(b,\kappa) = b\right] = \frac{1}{2}\Pr_c\left[\mathsf{G}_k(c,\kappa) = c\right] + \frac{1}{2}\Pr_c\left[\mathsf{G}_{k+1}(c,\kappa) \neq c\right].$$

This directly implies that

$$\mathsf{Adv}^{\mathsf{NT}}_{\Pi,\mathcal{B}}(\kappa,n) = \frac{1}{2}\left(\Pr_c\left[\mathsf{G}_k(c,\kappa) = c\right] - \Pr_c\left[\mathsf{G}_{k+1}(c,\kappa) = c\right]\right) \not\leq \mathrm{negl}(\kappa).$$

This contradicts our assumption that Π is computationally non-transferable, thus Π must be n-party sender-private. □

3.6 Conclusion

In this chapter, we provided a way of defining sender-privacy in the n-party setting that is novel for DVS schemes, a generalization of existing definitions and in line with definitions for other types of schemes in the multi-party setting, in particular, ID-based SDVS schemes. We explored the effects of choosing the challenge differently and observed that this induces only polynomial differences in the advantage the adversary has. Furthermore, we showed how other properties of a SDVS scheme can be used to boost the sender-privacy of a scheme from an alternative definition to our definition. In particular, we have proven that under the assumption of strong unforgeability and computational non-transferability a 2-party sender-private scheme is n-party sender-private. The proven relations are important since the SDVS schemes are often meant to be employed in an n-party setting and we give sufficient conditions for this to be secure.

We would like to stress that the objective of this chapter is to formulate sender-privacy in such a way that it can be invoked in proofs that require this property and deal with a multi-party setting. Our last theorem shows that in almost all schemes the two-party sender-privacy will extend to the multi-party setting, as most schemes are unforgeable and non-transferable. As such we do not provide separating examples of schemes that satisfy one definition but not another, as any such case would be extremely artificial. Instead, the definitions in this work and their equivalence should be used to simplify proofs where the sender-privacy property is used, both in classical and quantum use cases.

CHAPTER 4

Deniable Public-Key Authenticated Quantum Key Exchange

This chapter is based on the paper "Deniable Public-Key Authenticated Quantum Key Exchange" [WAR].

In this work, we explore the notion of deniability in public-key authenticated quantum key exchange (QKE), which allows two parties to establish a shared secret key without leaving any evidence that would bind a session to either party. The deniability property is expressed in terms of being able to simulate the transcripts of a protocol. The ability to deny a message or action has applications ranging from secure messaging to secure e-voting and whistle-blowing. While quite well-established in classical cryptography, it remains largely unexplored in the quantum setting. Here, we first present a natural extension of classical definitions in the simulation paradigm to the setting of quantum computation and formalize the requirements for a deniable QKE scheme. We then prove that the BB84 variant of QKE, when authenticated using a strong designated verifier signature scheme, satisfies deniability and, finally, we propose a concrete instantiation.

4.1 Introduction

Among the wide variety of anticipated cybersecurity challenges, the possibility of the emergence of scalable quantum computers poses a serious threat to our current information security infrastructure and has been receiving increasingly more attention from the information security community over the past few decades. While quantum computing would have its advantages, Shor's algorithm [Sho94] for efficiently computing discrete logarithms and performing integer factorization showed that quantum computing is a double-edged sword as it can be equally damaging when used for the purpose of compromising public-key (PK) cryptosystems that guarantee the security of today's modern communication systems.

These concerns are perhaps best exemplified by recent advances that have prompted calls by the National Security Agency (NSA) for transitioning to post-quantum (PQ) secure cryptosystems and the call for PQ secure proposals, initiated by the National Institute of Standards and Technology (NIST) as part of a standardization process for post-quantum algorithms [SN17].

On the other hand, Quantum Key Exchange (QKE), provides security without relying on computational assumptions as in PQ key exchange protocols, but at the cost of developing new infrastructure to support quantum channels. Whereas such quantum communication infrastructures for a long time were mostly of academic interest, both terrestrial and satellite networks are now being deployed in practice and planned at large scale, see e.g. [Che+21, Eur].

Deniability constitutes a subtle and fundamental concept in cryptography that has many applications ranging from secure messaging (e.g., the Signal protocol) to coercion-resistance in secure e-voting to deniable transmission and storage in the context of data breaches. On a more fundamental and theoretical level, deniability shares an intimate connection with incoercible secure multi-party computation [CG96]. Yet, it has received very little attention in the quantum setting and thus presents a wealth of open questions.

Attempts at providing security against quantum adversaries can be broken down into two classes, namely those that largely rely on classical constructions that are conjectured to be quantum-secure, often classified as post-quantum cryptography, e.g., lattice-based cryptography, and those that make use of quantum information processing and thus fall in the realm of quantum cryptography, such as quantum key exchange. In both cases, and perhaps more surprisingly in the context of quantum cryptography, the notion of deniability has been largely neglected to the extent that there exist only a few works on this topic in the literature [Bea02, Ata+18, Ata19].

In this work, we focus on deniability in public-key authenticated QKE in the simulation paradigm wherein a scheme is considered to be deniable if its transcripts can be simulated. This becomes relevant in a setting with two parties A and B, in which one of the parties is dishonest (i.e., the adversary $\mathcal{M}$) and the goal is to prevent either one from proving to a judge that they exchanged a key with a specific party in a given session. Now, if the transcript obtained by $\mathcal{M}$ could have been simulated without having access to the honest party's secret key, the resulting evidence cannot convincingly associate a specific party with a given session. Note that in the case of deniable key exchange, not only the communication but also the resulting session secret should be simulatable [DRGK06].

The particular choice of considering public-key authentication for QKE is motivated by the following observations. As already pointed out in the seminal work of Di Raimondo et al. [DRGK06] on deniable authenticated key exchange for classical schemes, deniability for symmetric key exchange protocols in the simulation paradigm is trivially satisfied. Secondly, to cope with the criticism that unconditionally secure QKE requires pre-shared symmetric keys for authentication, a problem that scales quadratically with the number of connected users, the idea of using public-key authentication algorithms for performing QKE with everlasting security had been considered for quite a while until its security was formally proved by Mosca et al. in [MSU13]. For a detailed analysis of PK-authenticated QKE, we refer the reader to [IM11].

While PK-authenticated QKE solves the problem of pre-shared keys, the signatures also introduce non-repudiation. In this chapter, we demonstrate how a deniable QKE can be constructed in the PK setting, in order to regain the deniability from the symmetric setting, by authenticating via quantum-safe strong designated verifier signatures (SDVS), e.g. obtained from lattices as in [NJ16], thus being potentially quantum secure. This implies that the resulting deniable QKE scheme would provide everlasting security, i.e., unless the adversary breaks the authentication during a limited window of attack, the derived shared secret key retains information-theoretic security. Note that due to a unique property of QKE, namely that of non-attributability [IM11] (i.e., the final secret key being completely independent of the classical communication and the initial pre-shared key), the simulatability of the classical communication and that of the secret key itself can be considered separately. The latter follows from the inherent properties of QKE and, to establish the deniability of our solution, it thus suffices to show that the transcript of the authentication can be simulated, i.e., the authentication is deniable.

4.1.0.1 Related Work

Compared to classical cryptography, deniability remains largely unexplored in the context of quantum and post-quantum cryptography. More specifically, in a paper by Beaver [Bea02] focusing on a setting motivated by an earlier work by Canetti et al. [Can+97] on deniable encryption, it is mainly argued that QKE protocols are not necessarily deniable. In a related work [Ata+18], Atashpendar et al. revisit Beaver's analysis and formalize the problem of coercer-deniability in terms of the indistinguishability of coercer views, which considers a scenario wherein the adversary can demand that the honest parties reveal their private randomness in order to verify whether or not their revealed secret key is real or fake. They also establish a link between covert quantum communication and deniability, as well as a relation between entanglement distillation and information-theoretic deniability.

However, [Ata+18] concludes with a number of open questions, including an analysis of public-key authenticated QKE in the simulation paradigm, which is the focus of our work.

The work of Canetti et al. [Can+97] led to a long series of works on deniability for various cryptographic primitives, including a formalization of deniability for authenticated key exchange in the simulation paradigm by Di Raimondo et al. [DRGK06], which in turn was an extension of the definitional work of Dwork et al. [DNS04] on deniable authentication in the context of zero-knowledge proofs. We refer the reader to [Ata+18, Ata19] and references therein for more details on deniability in cryptography.

4.1.0.2 Contributions

We adopt the security framework for authenticated QKE given in [MSU13] and adapt the classical definition of deniable AKEs [DRGK06] to the quantum setting for public-key authenticated QKE and formulate it in terms of the simulatability of protocol transcripts in a game-based setting.

We prove in Theorem 4.7 that a public-key authenticated QKE protocol satisfies deniability when authenticated using an SDVS with non-transferability against quantum adversaries. We also propose the first concrete instance of a deniable PK-authenticated QKE, which is a BB84 variant whose deniability follows as a corollary of Theorem 4.7.

4.2 Preliminaries

4.2.0.1 Strong Designated Verifier Signatures (SDVS)

The classical communication in authenticated QKE poses a challenge for deniability because a receiver must be able to verify that a message came from the correct sender, but this task must be impossible for any eavesdropper. Note that we focus explicitly on the setting of public-key authentication, which presents the problem that a standard signature, verifiable by anyone with the signer's public key, would prove the involvement of the signer. In the symmetric-key setting, this problem would be trivially solved, as any signature can only be verified by the signer and the intended recipient, and either party can create the same signatures. To achieve these same properties in the public-key setting, we make use of strong designated verifier signatures presented in chapter 3.

In this chapter, we will make use of the SDVS scheme proposed in [NJ16], called SUSDVS, which satisfies the desired sender-privacy notion under a number of hardness assumptions on lattices. For the interested reader, we present the assumptions here, based on the Short Integer Solutions (SIS) and the Inhomogenous Short Integer Solutions (ISIS) problems. The assumptions are conjectured to hold in the presence of quantum computers. For the full scheme and all the details, we refer to [NJ16]. The following (simplified) parameters are used:

Definition 4.1. *Given a uniformly random matrix $A \in \mathbb{Z}_q^{n \times m}$ and a syndrome $\boldsymbol{u} \in \mathbb{Z}_q^n$, the $\mathsf{ISIS}_{q,m,\beta}$ problem is to find a nonzero vector $\boldsymbol{v} \in \mathbb{Z}^m$ such that $A\boldsymbol{v} = \boldsymbol{u} \pmod q$ and $\|\boldsymbol{v}\| \leq \beta$. The $\mathsf{SIS}_{q,m,\beta}$ problem is the $\mathsf{ISIS}_{q,m,\beta}$ problem with syndrome $\boldsymbol{u} = \boldsymbol{0}$.*

- $|\mathsf{msg}|$ is the length of the message being signed,
- κ is the security parameter,
- $h = O(\log \kappa)$
- $m = O(\kappa h)$,
- $q = \mathsf{poly}(\kappa)$ is a sufficiently large number,
- $l \leq (p-1)\kappa$, where p is the smallest prime dividing q, and
- $s = O(\sqrt{\kappa l h}) \cdot \omega(\sqrt{\log \kappa})^2$ a sufficiently large parameter.

Assumption 4.2. The $\mathsf{SIS}_{q,m,\beta}$ problem is hard for sufficiently large q and β, where $q = \sqrt{(|\mathsf{msg}| + 4ms^2)\kappa} + \omega(\sqrt{\log \kappa})$ and $\beta = \sqrt{|\mathsf{msg}| + 4ms^2}$. The $\mathsf{SIS}_{q,m,\beta}$ problem and $\mathsf{ISIS}_{q,m,\beta}$ problem are hard for sufficiently large $q = O(l^{3/2}\kappa^3 \log^{5/2} \kappa) \cdot \omega(\sqrt{\log \kappa})^6$, $m = O(\kappa \log q)$, and $\beta = s\sqrt{2m}O(l\kappa^{3/2}k^{3/2}) \cdot \omega(\sqrt{\log \kappa})^3$.

Furthermore, we have the following assumption on the hardness of distinguishing lattices, where q is prime. Here $\mathcal{D}_{\Lambda_{\boldsymbol{w}}^{\perp}(A),s}$ denotes the distribution of sampling from $\{\boldsymbol{z} \in \mathbb{Z}^{\kappa} \mid A\boldsymbol{z} = \boldsymbol{w} (\text{mod } q)\}$ according to a Gaussian distribution.

Assumption 4.3 (Assumption 2.1 in [NJ16])**.** Let $m_1, m_2 = O(\kappa \log q)$, A, R uniform random matrices from $\mathbb{Z}_q^{\kappa \times m_1}$, $C_0, \ldots, C_l$ uniform random matrices from $\mathbb{Z}_q^{\kappa \times m_2}$, $\boldsymbol{w}$ a fixed vector from $\mathbb{Z}_q^{\kappa}$, $\mu \in \{0,1\}^l$ a secret bitstring, and $C_\mu = C_0 + \sum_{j=1}^{l} \mu_j C_j$, then it is hard to distinguish between $\mathcal{D}_{\Lambda_{\boldsymbol{w}}^{\perp}(A|C_\mu),s}$ and $\mathcal{D}_{\Lambda_{\boldsymbol{w}}^{\perp}(R|C_\mu),s}$ without any information on μ.

4.2.0.2 BB84

In Algorithm 9, we describe the *BB84* protocol, the most well-known QKE variant due to Bennett and Brassard [BB84]. We use the well-established formalism based on error-correcting codes, used by Shor and Preskill [SP00]. Let $C_1[n, k_1]$ and $C_2[n, k_2]$ be two classical linear binary codes encoding k_1 and k_2 bits in n bits such that $\{0\} \subset C_2 \subset C_1 \subset \mathbf{F}_2^n$ where $\mathbf{F}_2^n$ is the binary vector space on n bits. A mapping of vectors $v \in C_1$ to a set of basis states (codewords) for the Calderbank-Shor-Steane (CSS) [CS96, Ste96] code subspace is given by: $v \mapsto (1/\sqrt{|C_2|}) \sum_{w \in C_2} |v + w\rangle$. Due to the irrelevance of phase errors and their decoupling from bit flips in CSS codes, Alice can send $|v\rangle$ along with classical error-correction information $u+v$ where $u, v \in \mathbf{F}_2^n$ and $u \in C_1$, such that Bob can decode to a codeword in C_1 from $(v+\epsilon)-(u+v)$ where ϵ is an error codeword, with the final key being the coset leader of $u+C_2$.

Game 9: BB84 for an n-bit key with protection against δn bit errors

```
Alice generates two random bit strings $a, b \in \{0,1\}^{(4+\delta)n}$, encodes $a_i$ into $|\psi_i\rangle$ in basis $(+)$ if $b_i = 0$ and in $(\times)$ otherwise, and $\forall i \in [1, |a|]$ sends $|\psi_i\rangle$ to Bob.
Bob generates a random bit string $b' \in \{0,1\}^{(4+\delta)n}$ and upon receiving the qubits, measures $|\psi_i\rangle$ in $(+)$ or $(\times)$ according to $b'_i$ to obtain $a'_i$.
Alice announces $b$ and Bob discards $a'_i$ where $b_i \neq b'_i$, ending up with at least $2n$ bits with high probability.
Alice picks a set $p$ of $2n$ bits at random from $a$, and a set $q$ containing $n$ elements of $p$ chosen as check bits at random. Let $v = p \setminus q$.
Alice and Bob compare their check bits and abort if the error exceeds a predefined threshold.
Alice announces $u+v$, where $v$ is the string of the remaining non-check bits, and $u$ is a random codeword in $C_1$.
Bob subtracts $u+v$ from his code qubits, $v+\epsilon$, and corrects the result, $u+\epsilon$, to a codeword in $C_1$.
Alice and Bob use the coset of $u+C_2$ as their final secret key of length $n$.
```

4.3 The Quantum Key Distribution Model

We use the QKD model from [MSU13] to model the combination of classical and quantum communication, for which we provide a brief overview here. Each *party* in this model has access to both a classical and a quantum Turing machine, connected by a private tape. Furthermore, the classical machine has access to a private randomness tape and both machines can communicate to other parties over public tapes. Two or more parties may execute a *protocol*, which is specified as a series of subroutines. Each subroutine is triggered by an *activation* over one of the tapes.

In this appendix we elaborate on the [MSU13] model. The classical Turing machine can receive the following activations:

- $\mathsf{SendC}(\Psi, \mathsf{msg})$: The Turing machine resumes the *session* with identifier Ψ using msg as input. Ψ may also be a vector of session identifiers, where it is clear from context which one belongs to the receiving party and which to other parties.
- $\mathsf{SendC}(\mathsf{params}, \mathsf{pid})$: When SendC is received without a session identifier it indicates the start of a new protocol execution with public parameters params.
- $\mathsf{Q2C}(\mathsf{msg})$: This activation indicates a classical output of the quantum Turing machine and activates the classical Turing machine with the most recent session.

The quantum Turing machine has the following activations:

- $\mathsf{SendQ}(\rho)$: The quantum Turing machine activates with as input the state ρ.
- $\mathsf{C2Q}(\mathsf{msg})$: The quantum Turing machine is activated by the classical Turing machine with message msg.

We use Ψ to denote *ephemeral* variables, which are variables that are bound to a session. After each activation, the Turing machines may send activations over their respective public channel and the private channel between them. At the end of a session, the classical Turing machine of both parties outputs four values:

- sk, the shared secret key established during this session, or $\perp$ if execution failed.
- pid, the identifier of the other party involved in this session.
- A vector $\boldsymbol{v} = (\boldsymbol{v_0}, \dots)$, where each $\boldsymbol{v_i}$ is a vector of labels of values.
- A vector $\boldsymbol{u} = (\boldsymbol{u_0}, \dots)$, where each $\boldsymbol{u_i}$ is a vector of labels of values.

A protocol is *correct* if, when all messages are delivered without changes or reordering, both parties output the same key sk and the same vector $\boldsymbol{v}$. Each classical value Ψ_d has a label $\ell(\Psi_d)$ and an adversary can partner a value by issuing $\mathsf{Partner}(\ell(\Psi_d))$ to learn the value corresponding to the label. An adversary can also partner a session Ψ, learning the value sk if it has been output. Note that if an adversary learns a value without partnering (through public communication, for example), this value remains *unpartnered.* A session Ψ is *fresh* as long as every $\boldsymbol{v_i}$ contains at least (the label of) one value that the adversary has not partnered and the adversary has not partnered Ψ or any session Ψ' with the same $\boldsymbol{v}$ and sk and, *at the time of output*, there is least one value in each $\boldsymbol{u_i}$ with which the adversary has not partnered. This signifies the main difference between $\boldsymbol{v}$ and $\boldsymbol{u}$: values in $\boldsymbol{u}$ pose no security risk if revealed after the key has been established, but values in $\boldsymbol{v}$ do.

4.3.1 BB84 in This Model

In Algorithm 10 and 11 we present, respectively, the initiator and responder roles in the [BB84] QKD protocol, following the [MSU13] model.

4.3.1.1 Modeling the Adversary

In our work, the main objective of the adversary is to prove the involvement of a party in a key exchange protocol, e.g. to prove that A talked to B. The model, as presented in [MSU13], was mainly used for an eavesdropping adversary, but in our case, we will consider the adversary to always be the initiator (A) or responder (B) in a protocol. The reason for this is that if no adversary $\mathcal{M}$ can prove that A talked to $\mathcal{M}$, then surely $\mathcal{M}$ can also not prove that A talked to B. This argument is also made in [DRGK06], although we provide some alternate views on this in the discussion.

Concretely, this means we model the adversary as a quantum and classical Turing machine, who can perform the QKE protocol with any number of *honest* parties P_i. The adversary can be the initiator or responder in any of these interactions. For any adversary $\mathcal{M}$, we write $\mathsf{View}_{\mathcal{M}}$ to mean the complete contents of $\mathcal{M}$'s memory at the end of execution, including all keys that were established with the other parties.

4.4 Deniability

We first provide a natural extension of the classical definition of deniability given in [DRGK06] to the quantum setting, by making both the adversary and the distinguisher a QPT algorithm. In the following definition, the adversary $\mathcal{M}$ is given access to the public

Game 10: Σ_I

```
    Upon Activation: $\mathsf{SendC}(\mathsf{start}, \mathsf{initiator}, R)$
1.1 Create a new session $\Psi^I$ with responder identifier $R$
1.2 Read $n_1$ random bits $\Psi^I_{dIR}$
1.3 Read $n_1$ random bits $\Psi^I_{bI}$
1.4 Send activation $\mathsf{C2Q}(\Psi^I_{dIR}, \Psi^I_{bI}, R)$
1.5 Send activation $\mathsf{SendC}(\Psi^I, \mathsf{start}, \mathsf{responder}, I)$ to $R$
    Upon Activation: $\mathsf{C2Q}(\Psi^I_{dIR}, \Psi^I_{bI}, R)$
2.1 Prepare $\rho$ to be the bitwise encoding of $\Psi^I_{dIR}$ in the $(+)$ or $(\times)$ basis if the
      corresponding bit of $\Psi^I_{bI}$ is 0 or 1 respectively
2.2 Send activation $\mathsf{SendQ}(\rho)$ to $R$
    Upon Activation: $\mathsf{SendC}(\Psi^I, \Psi^R, \Psi^R_{bR})$
3.1 Discard all bit positions from $\Psi^I_{dIR}$ for which $\Psi^I_{bI}$ is not equal to $\Psi^R_{bR}$; Let $n_2$
      denote the amount of bits left in $\Psi^I_{dIR}$
3.2 Read $n_2$ random bits $\Psi^I_{indIR}$; Let $\Psi^I_{chkIR}$ be the substring of $\Psi^I_{dIR}$ for which the
      bits of $\Psi^I_{indIR}$ are 1 and $\Psi^I_{kIR}$ the substring for which they are 0; Let $n_3$ be the
      length of $\Psi^I_{kIR}$
3.3 Send activation $\mathsf{SendC}(\Psi^I, \Psi^R, \Psi^I_{bI}, \Psi^I_{indIR}, \Psi^I_{chkIR})$ to $R$
    Upon Activation: $\mathsf{SendC}(\Psi^I, \Psi^R, \varepsilon, \sigma_R)$
4.1 Read random bits $\Psi^I_F$ to construct a 2-universal hash function $F$ and compute
      $F' \leftarrow F(\Psi^I_{kIR})$
4.2 Read random bits $\Psi^I_{P,G}$ to construct a 2-universal hash function $G$ and a random
      permutation $P$ and compute $\Psi^I_{skIR} \leftarrow G(P(\Psi^I_{kIR}))$
4.3 Compute $\sigma_I \leftarrow \mathsf{Sign}(\Psi_I, \Psi_R, \Psi^I_{bI}, \Psi^I_{indIR}, \Psi^I_{chkIR}, F, F', P, G, I)$
4.4 Send activation $\mathsf{SendC}(\Psi^I, \Psi^R, F, F', P, G, \sigma_I)$ to $R$
4.5 Abort if $\mathsf{Verify}(\sigma_R, (\Psi^I, \Psi^R, \Psi^R_{bR}, \varepsilon, R))$ fails
4.6 Output $(\mathsf{sk} = \Psi^I_{skIR}, pid = R, \boldsymbol{v} =$
      $(\ell(\Psi^I_{dIR}), \ell(\Psi^R_{dIR}), \ell(\Psi^I_{bI}), \ell(\Psi^R_{bR}), \ell(\Psi^I_F), \ell(\Psi^I_{P,G})), \boldsymbol{u} = (\mathsf{sk}_I))$
```

keys of an arbitrary amount of honest parties, with whom $\mathcal{M}$ can interact. $\mathcal{M}$ is also given an auxiliary input from the set AUX. The simulator is given all the same inputs as $\mathcal{M}$, including the same classical randomness, but cannot interact with the honest parties.

Definition 4.4. *A* QKE *scheme* $(\mathsf{AKG}, \Sigma_I, \Sigma_R)$ *is* deniable *w.r.t.* AUX *if for any* QPT *adversary* $\mathcal{M}$ *there exists a* QPT *simulator* $\mathsf{SIM}_{\mathcal{M}}$ *s.t.*

$$\forall \kappa \in \mathbb{N}, aux \in \mathsf{AUX} : \mathcal{Real}(\kappa, aux) \approx_q \mathcal{Sim}(\kappa, aux),$$

where

$$\mathcal{Real}(\kappa, aux) = [(\mathsf{sk}_i, \mathsf{pk}_i) \leftarrow \mathsf{AKG}(1^\kappa); (aux, \mathbf{pk}, \mathsf{View}_{\mathcal{M}}(\mathbf{pk}, aux))]$$
$$\mathcal{Sim}(\kappa, aux) = [(\mathsf{sk}_i, \mathsf{pk}_i) \leftarrow \mathsf{AKG}(1^\kappa); (aux, \mathbf{pk}, \mathsf{SIM}_{\mathcal{M}}(\mathbf{pk}, aux))].$$

Definition 4.5. *Given a public-key signature scheme* $(\mathsf{AKG}, \mathsf{Sign}, \mathsf{Verify})$*, we define the* QKE *scheme* $\mathsf{AuthBB} := (\mathsf{AKG}, \Sigma_I, \Sigma_R)$ *(authenticated BB84), where* Σ_I *and* Σ_R *are as in Algorithm 10 and Algorithm 11 respectively. This is the implementation of BB84 as described before, in the above-presented model and using the public-key signature scheme for the classical authentication.*

Game 11: Σ_R

```
    Upon Activation: $\mathsf{SendC}(\Psi^I, \mathsf{start}, \mathsf{responder}, R)$
1.1 Create a new session $\Psi^R$ with initiator identifier I
1.2 Read $n_1$ random bits $\Psi^R_{bR}$
1.3 Send activation $\mathsf{C2Q}(\Psi^R_{bR})$
    Upon Activation: $\mathsf{C2Q}(\Psi^R_{bR})$ combined with $\mathsf{SendQ}(\rho)$
2.1 Set $\Psi^R_{dIR}$ to be the qubit-wise measurement of $\rho$ in the $(+)$ or $(\times)$ basis if the
      corresponding bit of $\Psi^R_{bR}$ is 0 or 1 respectively
2.2 Send activation $\mathsf{Q2C}(\Psi^R_{dIR})$
    Upon Activation: $\mathsf{Q2C}(\Psi^R_{dIR})$
3.1 Send activation $\mathsf{SendC}(\Psi^I, \Psi^R, \Psi^R_{bR})$ to $I$
    Upon Activation: $\mathsf{SendC}(\Psi^I, \Psi^R, \Psi^I_{bI}, \Psi^I_{indIR}, \Psi^I_{chkIR})$
4.1 Discard all bit positions from $\Psi^R_{dIR}$ for which $\Psi^I_{bI}$ is not equal to $\Psi^R_{bR}$
4.2 Let $\Psi^R_{chkIR}$ be the substring of $\Psi^R_{dIR}$ for which the bits of $\Psi^I_{indIR}$ are 1 and $\Psi^R_{kIR}$
      the substring for which they are 0
4.3 Let $\varepsilon$ be the proportion of bits of $\Psi^I_{chkIR}$ that do not match $\Psi^R_{chkIR}$; abort if
      $\varepsilon > \delta$, where $\delta$ is the error rate parameter
4.4 Compute $\sigma_R \leftarrow \mathsf{Sign}(\Psi^I, \Psi^R, \Psi^R_{bR}, \varepsilon, R)$
4.5 Send activation $\mathsf{SendC}(\Psi^I, \Psi^R, \varepsilon, \sigma_R)$ to $I$
    Upon Activation: $\mathsf{SendC}(\Psi^I, \Psi^R, F, F', P, G, \sigma_I)$
5.1 Abort if $\mathsf{Verify}(\sigma_I, (\Psi_I, \Psi_R, \Psi^I_{bI}, \Psi^I_{indIR}, \Psi^I_{chkIR}, F, F', P, G, I))$ fails
5.2 Use $F$ and $F'$ to correct $\Psi^R_{kIR}$ to $\Psi^R_{kIR'}$
5.3 Compute $\Psi^R_{skIR} \leftarrow G(P(\Psi^R_{kIR'}))$
5.4 Output $(\mathsf{sk} = \Psi^R_{skIR}, pid = I, \boldsymbol{v} =$
      $(\ell(\Psi^I_{dIR}), \ell(\Psi^R_{dIR}), \ell(\Psi^I_{bI}), \ell(\Psi^R_{bR}), \ell(\Psi^I_F), \ell(\Psi^I_{P,G})), \boldsymbol{u} = (\mathsf{sk}_R))$
```

We restate the definition of deniability, in order to relate deniability to the properties of SDVS, which are presented in a game-based setting.

Definition 4.6 (Restatement of Definition 4.4 with $\mathsf{AUX} = \{0\}$)**.** *A* QKE *scheme* $\Pi = (\mathsf{AKG}, \Sigma_I, \Sigma_R)$ *is* deniable *if for any* QPT *adversary* $\mathcal{M}$ *there exists a* QPT *simulator* $\mathsf{SIM}_{\mathcal{M}}$ *that does not interact with any party s.t. no* QPT *distinguisher* $\mathcal{F}$ *can achieve non-negligible advantage* $\mathsf{Adv}^{\mathsf{Den}}_{\Pi,\mathcal{F},\mathcal{M},\mathsf{SIM}_{\mathcal{M}}}(\kappa, n)$*, which is the advantage in winning the game* $\mathsf{G}^{\mathsf{Den}}_{\Pi,\mathcal{F},\mathcal{M},\mathsf{SIM}_{\mathcal{M}}}$.

Game 12: $\mathsf{G}^{\mathsf{Den}}_{\Pi,\mathcal{F},\mathcal{M},\mathsf{SIM}_{\mathcal{M}}}(\kappa, n, b)$

```
$(\mathsf{pk}_{P_0}, \mathsf{sk}_{P_0}) \leftarrow \mathsf{AKG}; \ldots; (\mathsf{pk}_{P_{n-1}}, \mathsf{sk}_{P_{n-1}}) \leftarrow \mathsf{AKG}$
Let $\mathbf{pk} = \mathsf{pk}_{P_0} \ldots \mathsf{pk}_{P_{n-1}}$
if $b = 0$ then
    $b' \leftarrow \mathcal{F}(\mathsf{View}_{\mathcal{M}}(\mathbf{pk}), \mathbf{pk})$
else
    $b' \leftarrow \mathcal{F}(\mathsf{SIM}_{\mathcal{M}}(\mathbf{pk}), \mathbf{pk})$
Output $b'$
```

The advantage of $\mathcal{F}$ in this game is defined as:

$$\mathsf{Adv}^{\mathsf{Den}}_{\Pi,\mathcal{F},\mathcal{M},\mathsf{SIM}_{\mathcal{M}}}(\kappa, n) := \Pr_{b \leftarrow \{0,1\}} \left[\mathsf{G}^{\mathsf{Den}}_{\Pi,\mathcal{F},\mathcal{M},\mathsf{SIM}_{\mathcal{M}}}(\kappa, n, b) = b\right] - \frac{1}{2}$$

4.4.1 Deniable PK-Authenticated BB84

In the following theorem, we provide a concrete scheme that satisfies our version of deniability, however we emphasize that the precise schemes chosen for this (in particular SUSDVS) are simply examples and not critical to the satisfiability of the definition, another possibility could be the scheme presented in [STW12].

Theorem 4.7. AuthBB *with authentication scheme* Π_{SDVS} *is deniable if* Π_{SDVS} *is an* SDVS *with non-transferability against quantum adversaries.*

Proof. Fix an arbitrary $\mathcal{M}$. This adversary $\mathcal{M}$ can generate many different keypairs to perform protocol sessions with the honest parties, or even use public keys for which they do not know the private key. However, it is not the goal of the adversary to impersonate or trick any of the honest parties, but simply to convince a third party that they interacted with one of the honest parties. Thus, w.l.o.g. we assume that, for each session, the adversary either uses a keypair $(\mathsf{pk}_{\mathcal{M}}, \mathsf{sk}_{\mathcal{M}})$ for which $\mathcal{M}$ knows the secret key or uses $\mathsf{pk}'_{\mathcal{M}}$ for which $\mathcal{M}$ does not know the private key. $\mathsf{SIM}_{\mathcal{M}}$ simulates $\mathcal{M}$ and all P_i, except:

(*) For each P_i, generate a keypair $(\mathsf{pk}'_{P_i}, \mathsf{sk}'_{P_i})$.

(1) Each call of $\mathsf{Sign}_{P_i \to \mathcal{M}}$ is replaced with $\mathsf{Simulate}_{P_i \to \mathcal{M}}$.

(2) Each call of $\mathsf{Sign}(\mathsf{sk}_{P_i}, \mathsf{pk}_{P_i}, \mathsf{pk}'_{\mathcal{M}}, x)$ is replaced with $\mathsf{Sign}(\mathsf{sk}'_{P_i}, \mathsf{pk}'_{P_i}, \mathsf{pk}'_{\mathcal{M}}, x)$.

(3) Each call of Verify is replaced with $\top$.

By definition, each P_i performs only honest executions of the protocol, thus only using its private key in Sign and Verify. Furthermore, each Sign using sk_{P_i} is replaced with either a Sign using sk'_{P_i} or a Simulate, which does not make use of sk_{P_i}. Each Verify is replaced with a static $\top$, which also does not use sk_{P_i}. This means $\mathsf{SIM}_{\mathcal{M}}$ can run on input **pk**, i.e. simulate each party P_i without the knowledge of sk_{P_i}.

First, we show that change (1) is undetectable by an adversary. Define $P_i^{(1)}$ to be the simulation of P_i after modification (1) and $\mathsf{SIM}^{(1)}_{\mathcal{M}}$ to be the simulation of $\mathcal{M}$ interacting with $P_i^{(1)}$ instead of P_i. Let Π be AuthBB using Π_{SDVS}. For any fixed distinguisher $\mathcal{F}$, let $\mathcal{H}_{start}$ be the $b = 0$ instance of $\mathsf{G}^{\mathsf{Den}}_{\Pi,\mathcal{F},\mathcal{M},\mathsf{SIM}^{(1)}_{\mathcal{M}}}$ and $\mathcal{H}_{end}$ be the $b = 1$ instance. Let $\mathcal{H}_0, \ldots, \mathcal{H}_m$ be a series of hybrids such that $\mathcal{H}_0 = \mathcal{H}_{start}$, $\mathcal{H}_m = \mathcal{H}_{end}$ and each step $\mathcal{H}_k \to \mathcal{H}_{k+1}$ replaces one $\mathsf{Sign}_{P_i \to \mathcal{M}}(x)$ with $\mathsf{Simulate}_{P_i \to \mathcal{M}}(x)$.

Suppose, for some fixed k, there exists a QPT distinguisher $\mathcal{D}$ that can distinguish between $\mathcal{H}_k$ and $\mathcal{H}_{k+1}$, where one $\mathsf{Sign}_{P_i \to \mathcal{M}}(x)$ is replaced in the step $\mathcal{H}_k \to \mathcal{H}_{k+1}$. We use this distinguisher to build an adversary $\mathcal{A}$ that breaks the non-transferability of Π_{SDVS}, as follows:

- $\mathcal{A}$ receives $(1, \mathsf{params}, \mathsf{pk}_S, \mathsf{sk}_S, \mathsf{pk}_V, \mathsf{sk}_V)$.
- $\mathcal{A}$ runs $\mathcal{H}_k$, but replaces $(\mathsf{pk}_{P_i}, \mathsf{sk}_{P_i}, \mathsf{pk}_{\mathcal{M}}, \mathsf{sk}_{\mathcal{M}})$ with $(\mathsf{pk}_S, \mathsf{sk}_S, \mathsf{pk}_V, \mathsf{sk}_V)$ before running $\mathcal{F}$.

- $\mathcal{A}$ stops $\mathcal{H}_k$ before the replaced $\mathsf{Sign}_{P_i \to \mathcal{M}}(x)$ call and outputs (x, state), where state is the state of $\mathcal{A}$ at this point.
- $\mathcal{A}$ receives $(2, \mathsf{state}, \sigma^*)$ and restores from state.
- $\mathcal{A}$ replaces $\mathsf{Sign}_{P_i \to \mathcal{M}}(x)$ with σ^* and continues running $\mathcal{H}_k$.
- $\mathcal{A}$ runs $\mathcal{D}$ and outputs what $\mathcal{D}$ outputs.

Observe that in the $b = 0$ case of $\mathsf{G}^{\mathsf{NT}}_{\Pi_{\mathsf{SDVS}},\mathcal{A}}$, the Sign is replaced by $\mathsf{Sign}_{S \to V}(x)$ and in the $b = 1$ case it is replaced by $\mathsf{Simulate}_{S \to V}(x)$. Furthermore, observe that the insertion of the $(\mathsf{pk}_S, \mathsf{sk}_S, \mathsf{pk}_V, \mathsf{sk}_V)$ keypairs is only a relabeling of the keypairs, but ensures that the $b = 0$ case of $\mathsf{G}^{\mathsf{NT}}_{\Pi_{\mathsf{SDVS}},\mathcal{A}}$ is equal to $\mathcal{H}_k$ and the $b = 1$ case equal to $\mathcal{H}_{k+1}$, thus the distinguishing probability of $\mathcal{D}$ is the same as the winning probability of $\mathcal{A}$ in the $\mathsf{G}^{\mathsf{NT}}_{\Pi_{\mathsf{SDVS}},\mathcal{A}}$ game, which would imply that $\mathsf{Adv}^{\mathsf{NT}}_{\Pi_{\mathsf{SDVS}},\mathcal{A}}$ is non-negligible. Since this is a contradiction, it must be the case that no such $\mathcal{D}$ exists.

For modification (2), the argument is similar. Suppose, in a chain of hybrids similar to the one above, there are two hybrids $\mathcal{H}$ and $\mathcal{H}'$, where $\mathcal{H}'$ is the result of replacing one call of $\mathsf{Sign}(\mathsf{sk}_{P_i}, \mathsf{pk}_{P_i}, \mathsf{pk}'_{\mathcal{M}}, x)$ with $\mathsf{Sign}(\mathsf{sk}'_{P_i}, \mathsf{pk}'_{P_i}, \mathsf{pk}'_{\mathcal{M}}, x)$ in $\mathcal{H}$ and some QPT distinguisher $\mathcal{D}$ can distinguish between them. We use this distinguisher to build an adversary $\mathcal{B}$ that breaks the $n+2$-party sender-privacy of Π_{SDVS}, as follows:

- $\mathcal{B}$ receives $(1, \mathsf{params}, \mathsf{pk}_0, \ldots, \mathsf{pk}_{n+1})$.
- $\mathcal{B}$ runs $\mathcal{H}$, but replaces $(\mathsf{pk}_{P_i}, \mathsf{pk}'_{P_i}, \mathsf{pk}_{P_0}, \ldots, \mathsf{pk}_{P_{i-1}}, \mathsf{pk}_{P_{i+1}}, \ldots, \mathsf{pk}_{P_{n-1}}, \mathsf{pk}'_{\mathcal{M}})$ with $(\mathsf{pk}_0, \ldots, \mathsf{pk}_{n+1})$ before running $\mathcal{F}$. All Sign, $\mathsf{Simulate}$ and Verify calls involving some P_j are performed by oracle calls.
- $\mathcal{B}$ stops $\mathcal{H}$ before the $\mathsf{Sign}(\mathsf{sk}_{P_i}, \mathsf{pk}_{P_i}, \mathsf{pk}'_{\mathcal{M}}, x)$ call that will be replaced in $\mathcal{H}'$ and outputs (x, state), where state is the state of $\mathcal{A}$ at this point.
- $\mathcal{B}$ receives $(2, \mathsf{state}, \sigma^*)$ and restores from state.
- $\mathcal{B}$ replaces $\mathsf{Sign}(\mathsf{sk}_{P_i}, \mathsf{pk}_{P_i}, \mathsf{pk}'_{\mathcal{M}}, x)$ with σ^* and continues running $\mathcal{H}$.
- $\mathcal{B}$ runs $\mathcal{D}$ and outputs what $\mathcal{D}$ outputs.

Observe that in the $b = 0$ case of $\mathsf{G}^{\mathsf{SendPriv}}_{\Pi_{\mathsf{SDVS}},\mathcal{B}}$, Sign is replaced by $\mathsf{Sign}(\mathsf{sk}_{P_i}, \mathsf{pk}_{P_i}, \mathsf{pk}'_{\mathcal{M}}, x)$ and in the $b = 1$ case it is replaced by $\mathsf{Sign}(\mathsf{sk}'_{P_i}, \mathsf{pk}'_{P_i}, \mathsf{pk}'_{\mathcal{M}}, x)$. Furthermore, observe that the replacement of the public keys for all honest parties and pk'_{P_i} is simply a relabeling since they were all honestly generated. The only public key that is not honestly generated was $\mathsf{pk}'_{\mathcal{M}}$, however since the adversary, by definition, does not know the corresponding private key the replacement is undetectable to the adversary. The replacements of the keys ensures that the $b = 0$ case of $\mathsf{G}^{\mathsf{SendPriv}}_{\Pi_{\mathsf{SDVS}},\mathcal{B}}$ is equal to $\mathcal{H}$ and the $b = 1$ case equal to $\mathcal{H}'$, thus the distinguishing probability of $\mathcal{D}$ is the same as the winning probability of $\mathcal{B}$ in the $\mathsf{G}^{\mathsf{SendPriv}}_{\Pi_{\mathsf{SDVS}},\mathcal{B}}$ game, which would imply that $\mathsf{Adv}^{\mathsf{SendPriv}}_{\Pi_{\mathsf{SDVS}},\mathcal{B}}$ is non-negligible. Since this is a contradiction, it must be the case that no such $\mathcal{D}$ exists.

For modification (3), observe that both the $\mathsf{initiator}$ and the $\mathsf{responder}$ perform their verification after having done all their communication. This means that it is impossible for $\mathcal{M}$ to prove to a third party whether the key exchange was accepted or rejected by P_i. Modification (3) ensures that, for the simulator, even invalid signatures are accepted by the simulated honest parties, but this change cannot alter the behaviour of the simulated adversary as the adversary cannot detect this change. In fact, for each session, the simulated honest party could stop their execution after the last message is sent since the rest of the execution is private and does not influence future sessions.

□

Corollary 4.8. *Under Assumptions 4.2 and 4.3,* SUSDVS *(from [NJ16]) is an* SDVS *with non-transferability against quantum adversaries, thus* AuthBB *using* SUSDVS *is deniable in the standard model.*

4.4.2 Eavesdropping on Interactions Between Honest Parties

In Theorem 4.7 we assume that the honest parties only perform QKE sessions with the adversary, arguing that the adversary has no more power as a third-party observer than she has as one of the participants. This assumption was also made in [DRGK06] and we consider it fundamentally sound. However, one can consider what happens if we relax it and give the adversary the ability to force two honest parties to perform a QKE session. The reason that this setting is interesting, is that the simulator is no longer able to create a signature between two honest parties, as doing so requires the private key of either of the honest parties. E.g. if two-party ring signatures were used for authentication, then when Alice and Bob communicate the adversary can prove that at least one of them was present, which would defy deniability.

To solve this problem, one can use the sender-privacy property of an SDVS scheme. The simulator simply generates a keypair for each simulated honest party and uses this to sign any messages, still designating the verifier by their original public key. Since all eavesdropped sessions are between honest parties, the simulator can skip the verification of these signatures. The sender-privacy property ensures that no third party can distinguish between these incorrect signatures and any correct ones the adversary might have collected.

4.5 Conclusion

While the work presented here provides a firm basis for deniability in the quantum setting, some obvious open problems remain. Firstly, our protocol delays all authentication until the end. This is done to stop the adversary from intentionally sending an invalid signature to cause an abort, as the simulator would not be able to perform the verification when simulating the honest parties. However, this intentional abort can only be caused by the behaviour of the adversary, which the simulator knows. Thus, intuitively deniability should be achievable without this modification.

Furthermore, in the case of QKE, there is inherent independence between the classical communication and the established key, meaning that the classical part of the transcript contains no information about the established key [MSU13]. This leads us to conjecture that using any deniable public-key authentication might be enough to create a deniable QKE protocol.

Finally, we limit ourselves to the case where $\mathsf{AUX} = \{0\}$ for simplicity, however, we conjecture that this restriction is not necessary and that the provided proof extends to any AUX.

CHAPTER 5

(Post-) Quantum Plaintext-Awareness

This chapter is based on the paper “Post-quantum Plaintext-Awareness” [EW22] published at PQCrypto 2022, which I wrote together with Ehsan Ebrahimi.

In this chapter, we formalize the plaintext-awareness notion in the superposition access model in which a quantum adversary may implement the encryption oracle in a quantum device and make superposition queries to the decryption oracle. Due to various possible ways an adversary can access the decryption oracles, we present six security definitions to capture the plaintext-awareness notion with respect to each way of access. We study the relationships between these definitions and present various implications and non-implications.

Classically, the strongest plaintext-awareness notion (PA2) accompanied by the indistinguishability under chosen-plaintext attack (IND-CPA) notion yields the indistinguishability under chosen-ciphertext attack (IND-CCA) notion. We show that the PA2 notion is not sufficient to show the above relation when targeting the IND-qCCA notion (Boneh-Zhandry definition, Crypto 2013). However, our proposed post-quantum PA2 notion with superposition decryption queries fulfils this implication.

5.1 Introduction

Plaintext-awareness is the property of a public-key encryption scheme that guarantees the only way to feasibly create a ciphertext is using the encryption algorithm, similar to the unforgeability notion for symmetric-key schemes. This property guarantees that the creator of a ciphertext knows the corresponding plaintext, even without knowing the secret key. This becomes a powerful tool when constructing proofs of other security properties, as it effectively negates the need to provide the adversary with a decryption oracle. For example, plaintext-awareness allows us to boost security from IND-CPA to IND-CCA since the only difference between these security properties is the availability of a decryption oracle to the adversary. Plaintext-awareness is also a useful property in the setting of deniability, where one would often like a process between two parties to be simulatable by either party. Plaintext-awareness steps in here and guarantees that any ciphertext created in this simulation can be decrypted without the need for a secret key, as the plaintext is known by the ciphertext-creating party and can be extracted from the simulation. Lastly, plaintext-awareness can provide useful insight into why a scheme does or does not achieve a certain level of security. Clearly, it is a property that one would intuitively like to satisfy, as the natural way of creating ciphertexts is to use the encryption algorithm. When another way to craft ciphertext is available, i.e. when a scheme is not plaintext-aware, this might indicate a gap in security.

The plaintext-awareness notion was first introduced in the random oracle model by Bellare and Rogaway [BR95]. Vaguely speaking, their definition of plaintext-awareness implies the existence of an extractor algorithm which, given access to the random oracle queries, is able to decrypt any ciphertext outputted by the adversary. The main motivation to define this notion was to show the security of Optimal Asymmetric Encryption Padding (OAEP).

The definition in [BR95] does not take into account the possibility of eavesdropping the communication by the adversary. Subsequently in [Bel+98], a stronger definition of plaintext-awareness was introduced in the random oracle model. In [Bel+98], the adversary is able to eavesdrop some valid ciphertexts (through an oracle) and the extractor, given access to these ciphertexts and the random oracle queries made by the adversary, should be able to decrypt any ciphertext outputted by the adversary.

The first attempt to define a plaintext-awareness notion in the standard model was in [HLM03], but, it needs to access a trusted third party. Later, Bellare and Palacio defined three levels of plaintext-awareness notions in the standard model (PA0, PA1, PA2) without the use of a third party [BP04]. In addition, they study the relations between these notions and IND-CCA notions.

The PA+1 notion, which lies between PA1 and PA2, was introduced by Dent [Den06]. Dent showed that an encryption scheme that is PA+1 and "simulatable" is PA2. Then he showed that the Cramer-Shoup encryption scheme is PA+1 and simulatable and therefore it satisfies the PA2 notion. This result is extended in the journal version [BD14]. A symmetric-key version of plaintext-awareness was considered in [And+14].

In this chapter, we investigate the plaintext-awareness notion in the quantum setting. This includes adopting the plaintext-awareness notion to the superposition setting in which a quantum adversary is attacking a classical public-key encryption scheme.

5.1.1 Motivation

The plaintext-awareness notion is a strong security notion for public-key encryption schemes, which guarantees that the adversary is not able to generate a valid ciphertext without knowing the corresponding plaintext (called PA1). If we consider the possibility of eavesdropping the communication for the adversary, a stronger notion is considered. Namely, an adversary with the ability to eavesdrop on the communication is not able to generate a valid ciphertext without knowing the corresponding plaintext unless it obtains this ciphertext through eavesdropping (called PA2).

Since the advent of quantum algorithms that break some classical computational problems [Sho97], there has been extensive research to construct post-quantum secure public-key encryption schemes[1]. This line of work varies from constructing public-key encryption schemes from quantum-hard assumptions [McE78, Reg05] to considering stronger security notions for public-key encryption schemes [BZ13b]. (For instance, the IND-qCCA notion introduced in [BZ13b] in which a quantum adversary has superposition access to the decryption oracle.)

Traditionally, the PA2 plaintext-awareness notion, accompanied by the IND-CPA notion, is used to prove IND-CCA security. If we use public-key encryption schemes based on quantum-hard assumptions, we will get the same result in the presence of a quantum

[1] Along with NIST competition to standardize the post-quantum public-key encryption schemes.

adversary as well (PA2 + IND-CPA implies IND-CCA in the presence of a quantum adversary). However, if one wants to achieve a stronger level of security (e.g. IND-qCCA security), the classical PA2 notion is not sufficient. In fact, we show that Classical PA2 + IND-qCPA does not imply IND-qCCA security (see Corollary 5.23.). Therefore, we need to formalize a stronger plaintext-awareness notion to achieve a security level of type IND-qCCA for public-key encryption schemes.

In addition, a post-quantum plaintext-awareness notion is used in a high-level manner in some existing security proofs in the literature without giving any formal treatment of the notion. For instance in [Ebr22], to show IND-qCCA security of plain OAEP transform in the quantum random oracle model, the adversary's inability in producing a valid ciphertext (without executing the encryption oracle or eavesdropping the communication) is crucial in the transition from Game 4 to Game 5 in their security proof. Note that this step will not hold with a classical PA2 notion since the adversary attacking in the sense of the IND-qCCA notion has superposition access to the decryption oracle. However, in the classical PA2 notion, the adversary can only make classical decryption queries in order to generate a valid ciphertext. Formalizing a post-quantum plaintext-awareness notion will lead to more formal and accessible IND-qCCA security proofs. And currently, such a notion is not available in the literature.

And last but not least, a quantum adversary on input pk can implement the encryption oracle in his quantum device. So it is natural and necessary to investigate the effect of this stronger access to the encryption oracle on the plaintext-awareness notion. Currently, it is unknown if superposition access to the encryption oracle renders public-key encryption schemes not-plaintext-aware or it does not give a noticeable advantage to the ciphertext-creator adversary.

The overall conclusion is that formalizing and investigating the plaintext-awareness notion in the quantum setting seems a natural and necessary extension given the facts that: 1) a quantum adversary can have quantum access to the encryption oracle and the effect of this access to PA notions is unknown, 2) available plaintext-awareness notions are not sufficient to conclude stronger security notions like the IND-qCCA notion, 3) some post-quantum security proofs rely on post-quantum plaintext-awareness notions in a high-level argument without any formal definition for PA notions in the quantum setting, etc.

5.1.2 Challenges and Our Contribution

Intuitively, we say a scheme is (classically) plaintext-aware if for any (ciphertext-creator) adversary $\mathcal{A}$, there exists a (plaintext-extractor) algorithm $\mathcal{A}^*$ that, when given access to the "view" of $\mathcal{A}$, is able to answer the decryption queries outputted by $\mathcal{A}$.

In the quantum setting, a quantum adversary on input pk can implement the encryption oracle in its quantum device, or equivalently, the adversary can run the encryption oracle in superposition. At first glance, it seems that the plaintext-awareness notion might not be possible to achieve when the adversary can execute the encryption oracle in superposition. Hypothetically, assume that an adversary $\mathcal{A}$ is able to access the encryption oracle by the "minimal-query model" [Kas+02], that is $|m\rangle \to |\mathsf{Enc}(m;r)\rangle$ (where r is a classical value chosen uniformly at random from the randomness space), without using any ancillary registers. In this model, the adversary is able to generate a valid ciphertext without knowing its corresponding plaintext. Namely, the adversary queries the uniform superposition of all messages, $\sum |m\rangle$, to get the superposition of corresponding ciphertexts, $\sum |\mathsf{Enc}(m;r)\rangle$. Now if the adversary measures the state $\sum |\mathsf{Enc}(m;r)\rangle$, the result is a random valid ciphertext for which the algorithm $\mathcal{A}^*$ might not be able to decrypt.

Even though the minimal query model has been studied in many works [Kas+02, GHS16, Car+21, GKS21], it is not a canonical quantum access model. For private-key encryption schemes, the implementation of this query model requires some ancillary quantum registers and a decryption query. In the public-key setting, the query model can be implemented for some public-key encryption schemes without knowledge of the secret key but with access to an ancillary register containing the randomness needed for the encryption [GKS21]. These encryption schemes are called "recoverable public-key encryption schemes" in [GKS21]. Note that this implementation of the minimal query model requires an ancillary register to store the randomness, that is, $|r,m\rangle \rightarrow |r,\mathsf{Enc}(m;r)\rangle$. Measuring the quantum state after the query fixes a randomness r and $c := \mathsf{Enc}(m;r)$ and using this randomness r, $\mathcal{A}^*$ is able to recover m from c, that is, the adversary knows the corresponding plaintext of c and the attack sketched above does not work for this implementation.

In this chapter, we consider the "standard query model" and not the minimal query model to formulate superposition access to the encryption oracle. For any classical function f, the standard way to implement this function in a quantum computer is $\mathbb{U}_f : |x,y\rangle \rightarrow |x,y \oplus f(x)\rangle$. So for an encryption oracle $\mathsf{Enc}_{\mathsf{pk}}$, we consider $\mathbb{U}_{\mathsf{Enc}_{\mathsf{pk}}} : |m,r,c\rangle \rightarrow |m,r,c \oplus \mathsf{Enc}_{\mathsf{pk}}(m;r)\rangle$. Clearly, this transformation is a unitary and an involution. In the above, we briefly discussed that available implementations of the minimal query model require some ancillary registers along with either a decryption query or access to the randomness register. Even though there is no implementation of the minimal query model without using ancillary registers ($|m\rangle \rightarrow |\mathsf{Enc}_{\mathsf{pk}}(m;r)\rangle$) and it might not be possible at all to implement the minimal query model without the use of ancillary registers (since a quantum operation is a unitary but the size of the ciphertext space is usually bigger than the size of the plaintext space and the operation $|m\rangle \rightarrow |\mathsf{Enc}_{\mathsf{pk}}(m;r)\rangle$ might not be a unitary), we give an argument below why it is not reasonable to consider the query model $|m\rangle \rightarrow |\mathsf{Enc}_{\mathsf{pk}}(m;r)\rangle$ to define plaintext-awareness notions.

5.1.3 Philosophical reasoning.

Note that in the public-key setting, the encryption oracle can be implemented in the standard way, so any effort conducted by the adversary to implement the query model $|m\rangle \rightarrow |\mathsf{Enc}_{\mathsf{pk}}(m;r)\rangle$ instead of implementing the encryption oracle as a standard query might be considered an intentional effort to forget the corresponding plaintext that is encrypted. Considering it from a different angle, let us consider this classical scenario in which the classical adversary encrypts a message m to obtain the ciphertext $c := \mathsf{Enc}(m;r)$, then it permanently deletes m from its memory. Now, the adversary possesses a ciphertext c without knowing its corresponding plaintext. We argue that any effort by the adversary to implement the query model $|m\rangle \rightarrow |\mathsf{Enc}_{\mathsf{pk}}(m;r)\rangle$ lies in the "encrypt-then-forget" argument sketched above.

In addition, we need to propose a notion that captures the vague intuition that we established above: "a valid ciphertext that is not the output of a superposition execution of the encryption oracle". Note that a superposition query to the encryption oracle can contain an exponential number of ciphertexts and thus we cannot argue that the output of the adversary is not in this superposition of ciphertexts.

	$\mathsf{pqPA2\text{-}Q_{dec}}$	$\mathsf{pqPA2\text{-}C_{dec}}$	$\mathsf{pqPA1\text{-}Q_{dec}}$	$\mathsf{pqPA1\text{-}C_{dec}}$	$\mathsf{pqPA0\text{-}Q_{dec}}$	$\mathsf{pqPA0\text{-}C_{dec}}$
$\mathsf{pqPA2\text{-}Q_{dec}}$		$\Rightarrow^{Theorem\ 5.14}$	$\Rightarrow^{Theorem\ 5.15}$	$\Rightarrow$	$\Rightarrow$	$\Rightarrow$
$\mathsf{pqPA2\text{-}C_{dec}}$	$\not\Rightarrow^{Theorem\ 5.17}$		$\Rightarrow$	$\Rightarrow^{Theorem\ 5.14}$	$\not\Rightarrow^{Corollary\ 5.19}$	$\Rightarrow$
$\mathsf{pqPA1\text{-}Q_{dec}}$	$\not\Rightarrow$	$\not\Rightarrow^{Theorem\ 5.20}$		$\Rightarrow^{Theorem\ 5.14}$	$\Rightarrow^{Theorem\ 5.16}$	$\Rightarrow$
$\mathsf{pqPA1\text{-}C_{dec}}$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow^{Theorem\ 5.17}$		$\not\Rightarrow$	$\Rightarrow^{Theorem\ 5.16}$
$\mathsf{pqPA0\text{-}Q_{dec}}$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow^{Theorem\ 5.21}$		$\Rightarrow^{Theorem\ 5.14}$
$\mathsf{pqPA0\text{-}C_{dec}}$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow$	$\not\Rightarrow^{Corollary\ 5.18}$	

Table 5.1: Implications and separations between definitions. An arrow in row n, column m indicates whether n implies or does not imply m. The superscript number next to an arrow indicates the number of the corresponding theorem. Arrows without a superscript follow by transitivity.

5.1.4 Our Contribution

In the superposition setting (when a classical public-key encryption scheme is attacked by a quantum adversary), we present various definitions. These definitions vary with respect to the following criteria:

- Number of decryption queries: one or many.
- Type of decryption queries: classical or quantum.
- Possibility of eavesdropping some ciphertexts.

Then, we study the relationship between these notions. Table 5.1 summarizes these notions and their relations with each other. In the abbreviation of notions, pq stands for post-quantum, $\mathsf{C_{dec}}$ stands for classical decryption queries, $\mathsf{Q_{dec}}$ stands for quantum decryption queries, PA0 is a notion with one decryption query and without the possibility of eavesdropping, PA1 is a notion with many decryption queries and without the possibility of eavesdropping and PA2 is a notion with many decryption queries and the possibility of eavesdropping. So for example, $\mathsf{pqPA1\text{-}Q_{dec}}$ is a notion in which the adversary is allowed to make many quantum decryption queries but is not allowed to eavesdrop ciphertexts.

Our notions are an adaptation of classical PA0, PA1, and PA2 notions in the standard model [BP04] to the quantum setting. Vaguely speaking, a public-key encryption scheme is plaintext-aware with respect to a class of adversaries if for any adversary $\mathcal{A}$ in the class, there exists a plaintext-extractor algorithm $\mathcal{A}^*$ that given access to the view of $\mathcal{A}$ is able to simulate the decryption algorithm without using the secret key. Classically, given access to the view of $\mathcal{A}$ is formalized by given $\mathcal{A}^*$ the access to the coin tosses of $\mathcal{A}$. In our setting, the adversaries are QPT algorithms and are able to generate randomness by doing some quantum operations. For instance, applying Hadamard to $|0\rangle$ and measuring the result in the computational basis gives a random bit. To formalize our notions, we give $\mathcal{A}^*$ access to the internal quantum registers of $\mathcal{A}$.

For instance, we say a public-key encryption scheme is $\mathsf{pqPA1\text{-}Q_{dec}}$ if for any QPT ciphertext-creator adversary $\mathcal{A}$ that makes quantum queries to the decryption oracle, there exists a QPT plaintext-extractor algorithm $\mathcal{A}^*$ that given access to the internal registers of $\mathcal{A}$ can simulate the decryption queries. In more detail, the execution of $\mathcal{A}$ querying the decryption oracle is indistinguishable from the execution of $\mathcal{A}$ querying $\mathcal{A}^*$ for any QPT distinguisher $\mathcal{D}$.

For PA2 notions, the possibility of eavesdropping the communication is given to $\mathcal{A}$ by classical access to a randomized algorithm $\mathcal{P}$ (called a plaintext-creator) that upon receiving a query from $\mathcal{A}$ generates a message, encrypts it and sends the ciphertext to $\mathcal{A}$. (Since in the post-quantum setting the honest parties use the classical public-key encryption schemes to communicate, we do not consider the possibility of eavesdropping a superposition of ciphertexts in this chapter.) Note that $\mathcal{A}^*$ does not have any access to the internal quantum registers of $\mathcal{P}$, so it might not be able to decrypt a ciphertext obtained from $\mathcal{P}$. The list of these ciphertexts is given to both the decryption oracle and $\mathcal{A}^*$ to return $\perp$ when one of these ciphertexts is submitted as a decryption query.

5.1.5 Organization

We present some preliminaries in Section 5.2. In Section 5.3, we define six possible definitions for the plaintext-awareness notion in the post-quantum setting. Section 5.4 discusses the relationships between notions. Finally, we discuss the achievability of our notions in Section 5.5.

5.2 Preliminaries

Any classical function $f : X \to Y$ can be implemented as a unitary operator $\mathbb{U}_f$ in a quantum computer where $\mathbb{U}_f : |x, y\rangle \to |x, y \oplus f(x)\rangle$ and it is clear that $\mathbb{U}_f^\dagger = \mathbb{U}_f$. A quantum adversary has standard oracle access to a classical function f if it can query the unitary $\mathbb{U}_f$.

5.2.1 Definitions

We define a strong quantum-secure pseudo-random permutation as a permutation that is indistinguishable from a random permutation when the quantum adversary has superposition access to the permutation and its inverse.

Definition 5.1. *We say a permutation P a strong quantum-secure pseudo-random permutation if for any* QPT *adversary $\mathcal{A}$,*

$$|\Pr\big[b = 1 : b \leftarrow \mathcal{A}^{\mathbb{U}_P, \mathbb{U}_{P^{-1}}}\big] - \Pr\big[b = 1 : b \leftarrow \mathcal{A}^{\mathbb{U}_\pi, \mathbb{U}_{\pi^{-1}}}\big]| \leq \mathrm{negl}(\kappa),$$

where π is a truly random permutation and κ is the security parameter.

5.2.2 Commitment Scheme

In the following, we define a commitment scheme.

Definition 5.2 (Commitment Scheme)**.** *A commitment scheme consists of three polynomial algorithms* Gen, Com *and* Ver *described below.*

- *The key generating algorithm* Gen *that on the input of the security parameter 1^κ returns a public-key pk_{com}.*
- *The commitment algorithm* Com *on the inputs pk_{com} and a message m chooses a randomness r and returns $c := \mathrm{Com}(\mathsf{pk}_{com}, m; r)$ and the corresponding opening information ω.*
- *The verification algorithm* Ver *on the inputs pk_{com}, c, ω and m, either accepts $(b = 1)$ or rejects $(b = 0)$.*

The scheme has the correctness property, that is, the verification algorithm returns 1 *with the probability* 1 *if* c, ω *are the output of* Com*:*

$$\Pr[b = 1 : \mathsf{pk}_{com} \leftarrow \mathrm{Gen}(1^\kappa), (c, \omega) \leftarrow \mathrm{Com}(\mathsf{pk}_{com}, m), b \leftarrow \mathrm{Ver}(\mathsf{pk}_{com}, c, \omega, m)] = 1.$$

We define the hiding and binding properties of a commitment scheme against a QPT adversary.

Definition 5.3. *We say a commitment scheme* $(\mathrm{Gen}(1^\kappa), \mathrm{Com}, \mathrm{Ver})$ *is computationally hiding if for any* $\mathsf{pk}_{com} \leftarrow \mathrm{Gen}(1^\kappa)$*, for any two messages* m_1, m_2 *and for any* QPT *distinguisher* $\mathcal{D}$

$$|\Pr\big[\mathcal{D}(\mathsf{pk}_{com}, c_1) = 1 : (c_1, \omega_1) \leftarrow \mathrm{Com}_{\mathsf{pk}_{com}}(m_1)\big] - \Pr\big[\mathcal{D}(\mathsf{pk}_{com}, c_2) = 1 : (c_2, \omega_2) \leftarrow \mathrm{Com}_{\mathsf{pk}_{com}}(m_2)\big]| \leq \mathrm{negl}(\kappa).$$

Definition 5.4. *A commitment scheme* $(\mathrm{Gen}(1^\kappa), \mathrm{Com}, \mathrm{Ver})$ *is computationally binding if for any commitment c, and any* QPT *adversary* $\mathcal{A}$

$$|\Pr\ [\mathrm{Ver}(\mathsf{pk}_{com}, c, m_1, \omega_1) = 1 \wedge \mathrm{Ver}(\mathsf{pk}_{com}, c, m_2, \omega_2) = 1 \wedge m_1 \neq m_2 : \mathsf{pk}_{com} \leftarrow \mathrm{Gen}(1^\kappa), (m_1, \omega_1, m_2, \omega_2) \leftarrow \mathcal{A}(c, \mathsf{pk}_{\mathrm{Com}})]| \leq \mathrm{negl}(\kappa).$$

Note that these properties are achievable, for instance, the commitment scheme in [Jai+12] fulfills these properties under certain computational assumptions.

We define a one-way public-key encryption scheme below. This is the minimum security requirement for an encryption scheme. This is needed for separation theorems between PA notions to exclude trivial encryption schemes, for example, the identity encryption scheme that is defined as $\mathsf{Enc}_{\mathsf{pk}}(m) = m$, which are plaintext-aware with respect to any reasonable definition.

Definition 5.5. *We say a public-key encryption scheme* $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is one-way if for any* QPT *adversary* $\mathcal{A}$

$$\Pr\Big[\mathcal{A}(\mathsf{pk}, c) = m : (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa), m \overset{\$}{\leftarrow} M, c \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m)\Big] \leq \mathrm{negl}(\kappa),$$

where M *is the message space. It is implicitly assumed* $|M| = poly(\kappa)$*.*

5.2.2.1 IND-qCPA and IND-qCCA.

Here, we define a quantum IND-CPA and quantum IND-CCA notion used in this chapter. Note that a quantum adversary can implement a public-key encryption algorithm in its quantum device since pk is public. To define IND-qCPA and IND-qCCA notions, we need to determine whether the challenge queries and decryption queries are classical or quantum. There are many quantum IND-CPA notions available in the literature [Car+21] that include definitions with classical challenge queries and quantum challenge queries, on the other hand, there is only one definite quantum IND-CCA notion (called IND-qCCA) available in the literature that only allows classical challenge queries [BZ13b][2] the weakest quantum IND-CPA notion which, accompanied by our quantum PA2 notion, implies the IND-qCCA notion. We follow the definitions proposed in [BZ13b] by Boneh and Zhandry in this chapter.

[2] There are some very recent research to define a quantum IND-CCA notion with quantum challenge queries (for instance [CEV22, GKS21]).

Definition 5.6. *We say an encryption scheme* Enc *is IND-qCPA secure if the following two games are indistinguishable for any* QPT *adversary* $\mathcal{X}$*.*

Game 0: $G^{qCPA}_{\mathcal{X},0}$

$$\begin{array}{rl} (\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^{\kappa}), & m_0, m_1 \leftarrow \mathcal{X}(\mathsf{pk}), \\ \mathsf{Enc}(m_0; r_0) \leftarrow \mathit{Challenger}(m_0, m_1), & b \leftarrow \mathcal{X}(\mathsf{pk}, \mathsf{Enc}(m_0; r_0)) \end{array}$$

Game 1: $G^{qCPA}_{\mathcal{X},1}$

$$\begin{array}{rl} (\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^{\kappa}), & m_0, m_1 \leftarrow \mathcal{X}(\mathsf{pk}), \\ \mathsf{Enc}(m_1; r_1) \leftarrow \mathit{Challenger}(m_0, m_1), & b \leftarrow \mathcal{X}(\mathsf{pk}, \mathsf{Enc}(m_1; r_1)) \end{array}$$

In other words, $|\Pr\Big[G^{qCPA}_{\mathcal{X},0} = 1\Big] - \Pr\Big[G^{qCPA}_{\mathcal{X},1} = 1\Big]| \leq \text{negl}(\kappa)$ *for any* QPT *adversary* $\mathcal{X}$*.*

5.2.2.2 IND-qCCA

Here, a quantum adversary can query the encryption and decryption oracle on a superposition of inputs but the challenge queries are classical. Let **List** be the list of ciphertexts obtained during the challenge phase. We say **List** is defined if at least one challenge query has been executed. We define a decryption algorithm $\mathsf{Dec}'_{(\mathsf{sk},\mathbf{List})}$ as follows:

$$\mathsf{Dec}'_{(\mathsf{sk},\mathbf{List})}(c) = \begin{cases} \perp & \text{if } \mathbf{List} \text{ is defined and } c \in \mathbf{List} \\ \mathsf{Dec}_{\mathsf{sk}}(c) & \text{otherwise} \end{cases}.$$

Definition 5.7. *We say an encryption scheme* Enc *is IND-qCCA secure if the following two games are indistinguishable for any* QPT *adversary* $\mathcal{X}$*.*

Game 0: $G^{qCCA}_{\mathcal{X},0}$

$$\begin{array}{rl} (\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^{\kappa}), & m_0, m_1 \leftarrow \mathcal{X}^{\mathbb{U}_{\mathsf{Dec}'_{(\mathsf{sk},\mathbf{List})}}}(\mathsf{pk}), \\ \mathsf{Enc}(m_0; r_0) \leftarrow \mathit{Challenger}(m_0, m_1), & b \leftarrow \mathcal{X}^{\mathbb{U}_{\mathsf{Dec}'_{(\mathsf{sk},\mathbf{List})}}}(\mathsf{pk}, \mathsf{Enc}(m_0; r_0)) \end{array}$$

Game 1: $G^{qCCA}_{\mathcal{X},1}$

$$\begin{array}{rl} (\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^{\kappa}), & m_0, m_1 \leftarrow \mathcal{X}^{\mathbb{U}_{\mathsf{Dec}'_{(\mathsf{sk},\mathbf{List})}}}(\mathsf{pk}), \\ \mathsf{Enc}(m_1; r_1) \leftarrow \mathit{Challenger}(m_0, m_1), & b \leftarrow \mathcal{X}^{\mathbb{U}_{\mathsf{Dec}'_{(\mathsf{sk},\mathbf{List})}}}(\mathsf{pk}, \mathsf{Enc}(m_1; r_1)) \end{array}$$

In other words, $|\Pr\Big[G^{qCCA}_{\mathcal{X},0} = 1\Big] - \Pr\Big[G^{qCCA}_{\mathcal{X},1}\Big]| \leq \text{negl}(\kappa)$ *for any* QPT *adversary* $\mathcal{X}$*.*

5.3 Post-quantum Plaintext-awareness

In this section, we define plaintext-awareness for classical encryption schemes in the presence of a quantum adversary. Let Q_{int} indicate the internal registers of the ciphertext-creator adversary $\mathcal{A}$. Note that Q_{int} includes the input, output and some ancillary registers of $\mathcal{A}$.

5.3.1 Post-quantum PA0, PA1

There are two possible cases to define PA0 and PA1. Namely, either $\mathcal{A}$'s goal is to output a classical ciphertext without knowing its corresponding plaintext or its goal is to output a superposition of ciphertexts where the corresponding quantum plaintext is unknown to $\mathcal{A}$. In the formulation of these two possible cases, the access to the decryption oracle will differ. Namely, either the adversary $\mathcal{A}$ has classical access to the decryption oracle or it has superposition access to the decryption oracle. In other words, to say that $\mathcal{A}$ is not able to output a valid classical (quantum) ciphertext unless it executes the encryption algorithm, there should be an algorithm $\mathcal{A}^*$ that can respond to classical (quantum) decryption queries given the internal registers of $\mathcal{A}$. That is, any valid ciphertext known to $\mathcal{A}$ can be decrypted if $\mathcal{A}^*$ has access to the internal register of $\mathcal{A}$.

5.3.1.1 Classical decryption queries

We define the definition using two games. In the real game, $\mathcal{A}$ given pk has access to the decryption oracle. In the fake game, the decryption queries will be answered with an algorithm $\mathcal{A}^*$ that has access to the internal register of $\mathcal{A}$. In both games, $\mathcal{A}$ outputs a quantum state in the end. We say a public-key encryption scheme is plaintext-aware if, for any QPT adversary $\mathcal{A}$, there exists a QPT algorithm $\mathcal{A}^*$ such that the output of these two games is indistinguishable for any QPT distinguisher $\mathcal{D}$. Without loss of generality, we assume that the output of $\mathcal{D}$ is determined with a computational basis measurement. This computational indistinguishability definition for quantum states is common in the literature, for instance in Definition 1 in [Kaw+05].

Game $G^{\mathsf{pqPA1\text{-}C_{dec}}}_{real}$. In this game, the ciphertext-creator adversary $\mathcal{A}$ given pk has classical access to the decryption oracle. At the end, $\mathcal{A}$ outputs a quantum state.

Game $G^{\mathsf{pqPA1\text{-}C_{dec}}}_{real}$

$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^{\kappa})$, $\rho_{\kappa} \leftarrow \mathcal{A}^{\mathsf{Dec}_{\mathsf{sk}}}(\mathsf{pk})$

Game $G^{\mathsf{pqPA1\text{-}C_{dec}}}_{fake}$. In this game, $\mathcal{A}$'s decryption queries will be responded by a plaintext-extractor algorithm $\mathcal{A}^*$. Here, $\mathcal{A}^*$ has access to pk and the internal register of $\mathcal{A}$. At the end, $\mathcal{A}$ outputs a quantum state.

Game $G^{\mathsf{pqPA1\text{-}C_{dec}}}_{fake}$

$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^{\kappa})$, $\rho_{\kappa} \leftarrow \mathcal{A}^{\mathcal{A}^*(\mathsf{pk},\mathsf{Q}_{int})}(\mathsf{pk})$

Definition 5.8 ($\mathsf{pqPA1\text{-}C_{dec}}$). *We say a public-key encryption scheme* Enc *is* $\mathsf{pqPA1\text{-}C_{dec}}$ *plaintext-aware if for any* QPT *ciphertext-creator* $\mathcal{A}$*, there exists a* QPT *plaintext-extractor* $\mathcal{A}^*$ *such that for all* QPT *distinguishing algorithms* $\mathcal{D}$*, the advantage of* $\mathcal{D}$ *in distinguishing* $G^{\mathsf{pqPA1\text{-}C_{dec}}}_{real}$ *and* $G^{\mathsf{pqPA1\text{-}C_{dec}}}_{fake}$ *is negligible as a function of the security parameter:*

$$\begin{aligned}\mathsf{Adv}_{\mathcal{D},\mathcal{A}} = |\Pr\Big[\mathcal{D}(\rho_\kappa) = 1 : \rho_\kappa \leftarrow G^{\mathsf{pqPA1\text{-}C_{dec}}}_{real}\Big] - \\ \Pr\Big[\mathcal{D}(\rho_\kappa) = 1 : \rho_\kappa \leftarrow G^{\mathsf{pqPA1\text{-}C_{dec}}}_{fake}\Big]| \leq \operatorname{negl}(\kappa),\end{aligned}$$

where the output of $\mathcal{D}$ *is determined with the computational basis measurement.*

Definition 5.9 ($\mathsf{pqPA0\text{-}C_{dec}}$)**.** *This is defined similarly to* $\mathsf{pqPA1\text{-}C_{dec}}$ *except the adversary* $\mathcal{A}$ *is allowed to make only one decryption query.*

5.3.1.2 Superposition decryption queries.

In this subsection, we define plaintext-awareness definition when the adversary $\mathcal{A}$ has superposition access to the decryption oracle. Similar to the above definition, we define this notion using two games.

Game $G_{real}^{\mathsf{pqPA1\text{-}Q_{dec}}}$. In this game, the ciphertext-creator adversary $\mathcal{A}$ given pk has quantum access to the decryption oracle. At the end, $\mathcal{A}$ outputs a quantum state.

Game $G_{real}^{\mathsf{pqPA1\text{-}Q_{dec}}}$

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa)$, $\rho_\kappa \leftarrow \mathcal{A}^{\mathbb{U}_{\mathsf{Dec}_{\mathsf{sk}}}}(\mathsf{pk})$

Game $G_{fake}^{\mathsf{pqPA1\text{-}Q_{dec}}}$. In this game, $\mathcal{A}$'s quantum decryption queries will be responded to by a plaintext-extractor algorithm $\mathcal{A}^*$. Here, $\mathcal{A}^*$ has access to pk and the internal register of $\mathcal{A}$. At the end, $\mathcal{A}$ outputs a quantum state.

Game $G_{fake}^{\mathsf{pqPA1\text{-}Q_{dec}}}$

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa)$, $\rho_\kappa \leftarrow \mathcal{A}^{\mathcal{A}^*(\mathsf{pk}, \mathsf{Q}_{int})}(\mathsf{pk})$

Definition 5.10 ($\mathsf{pqPA1\text{-}Q_{dec}}$)**.** *We say a public-key encryption scheme* Enc *is* $\mathsf{pqPA1\text{-}Q_{dec}}$ *plaintext-aware if for any* QPT *ciphertext-creator* $\mathcal{A}$*, there exists a* QPT *plaintext-extractor* $\mathcal{A}^*$ *such that for all* QPT *distinguishing algorithms* $\mathcal{D}$*, the advantage of* $\mathcal{D}$ *in distinguishing* $G_{real}^{\mathsf{pqPA1\text{-}Q_{dec}}}$ *and* $G_{fake}^{\mathsf{pqPA1\text{-}Q_{dec}}}$ *is negligible as a function of the security parameter:*

$$\begin{aligned}\mathsf{Adv}_{\mathcal{D},\mathcal{A}} = |\Pr\Big[\mathcal{D}(\rho_\kappa) = 1 : \rho_\kappa \leftarrow G_{real}^{\mathsf{pqPA1\text{-}Q_{dec}}}\Big] - \\ \Pr\Big[\mathcal{D}(\rho_\kappa) = 1 : \rho_\kappa \leftarrow G_{fake}^{\mathsf{pqPA1\text{-}Q_{dec}}}\Big]| \leq \operatorname{negl}(\kappa),\end{aligned}$$

where the output of $\mathcal{D}$ *is determined with the computational basis measurement.*

Definition 5.11 ($\mathsf{pqPA0\text{-}Q_{dec}}$)**.** *This is defined similarly to* $\mathsf{pqPA1\text{-}Q_{dec}}$ *except the adversary* $\mathcal{A}$ *is allowed to make only one decryption query.*

5.3.2 Post-quantum PA2

In $\mathsf{pqPA0}$ and $\mathsf{pqPA1}$ definitions, it has not been considered that the adversary may be able to eavesdrop some ciphertexts and use them to generate new ciphertexts without knowing their corresponding plaintexts. There are two scenarios for the eavesdropping:

- The adversary may eavesdrop some classical ciphertexts.
- The adversary may obtain some superposition of ciphertexts.

Note that in the post-quantum setting, the honest parties are using their classical devices to communicate. So the assumption that the adversary may be able to eavesdrop some superposition of ciphertexts seems too exotic and we do not analyse it in this chapter.

A possible plaintext-awareness definition that considers superposition eavesdropping may be difficult to define due to the no-cloning theorem. For instance, if we follow the above formalism, the plaintext-creator adversary $\mathcal{P}$ upon receiving the input and output registers Q_{inp} and Q_{out} from $\mathcal{A}$, can apply a random unitary to Q_{inp}, then applies the encryption unitary and sends both registers back to the adversary. But now it is not clear how one can handle decryption queries. More specifically, the superposition ciphertexts that have been created by calling $\mathcal{P}$ can not be recorded in general and if one of them is submitted as a decryption query, in the real game, the decryption oracle will return the corresponding superposition of messages but in the fake game, $\mathcal{A}^*$ is not able to return the corresponding superposition of messages without access to the internal register of $\mathcal{P}$. Note that if $\mathcal{A}^*$ is able to decrypt those queries without access to the internal register of $\mathcal{P}$ and the secret key, it renders the encryption scheme insecure.

The possibility for eavesdropping is granted to the adversary by a randomized algorithm $\mathcal{P}$ (called the plaintext-creator). Here, $\mathcal{P}$ upon receiving a query from $\mathcal{A}$ outputs the encryption of a message of its choosing to $\mathcal{A}$. Additionally, $\mathcal{P}$ adds m and its corresponding ciphertext to a **List**.

Similar to $\mathsf{pqPA0}$ and $\mathsf{pqPA1}$, we consider two possible goals for the adversary $\mathcal{A}$: outputting a classical ciphertext without knowing its corresponding plaintext or a superposition of ciphertexts without knowing its corresponding superposition of plaintexts.

Recall that $\mathsf{Dec}'_{(\mathsf{sk},\mathbf{List})}$ is defined as:

$$\mathsf{Dec}'_{(\mathsf{sk},\mathbf{List})}(c) = \begin{cases} \bot & \text{if } \mathbf{List} \text{ is defined and } c \in \mathbf{List} \\ \mathsf{Dec}_{\mathsf{sk}}(c) & \text{otherwise} \end{cases}.$$

5.3.2.1 Classical decryption queries

In this subsection, we define plaintext-awareness when the adversary $\mathcal{A}$ has classical access to a plaintext creator algorithm $\mathcal{P}$ and the decryption oracle $\mathsf{Dec}'_{(\mathsf{sk},\mathbf{List})}$. Similarly, we define the notion using two games.

Game $G_{real}^{\mathsf{pqPA2\text{-}C_{dec}}}$. In this game, the ciphertext-creator adversary $\mathcal{A}$ given pk has oracle access to $\mathcal{P}$. It has classical access to the decryption oracle $\mathsf{Dec}'_{(\mathsf{sk},\mathbf{List})}$. At the end, $\mathcal{A}$ outputs a quantum state.

Game $G_{real}^{\mathsf{pqPA2\text{-}C_{dec}}}$

$$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa),\ \rho_\kappa \leftarrow \mathcal{A}^{\mathcal{P},\mathsf{Dec}'_{(\mathsf{sk},\mathbf{List})}}(\mathsf{pk})$$

Game $G_{fake}^{\mathsf{pqPA2\text{-}C_{dec}}}$. In this game, $\mathcal{A}$'s decryption queries will be responded to by a plaintext-extractor algorithm $\mathcal{A}^*$. Here, $\mathcal{A}^*$ given pk has access to **List** and the internal register of $\mathcal{A}$. At the end, $\mathcal{A}$ outputs a quantum state.

Game $G_{fake}^{\mathsf{pqPA2\text{-}C_{dec}}}$

$$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa),\ \rho_\kappa \leftarrow \mathcal{A}^{\mathcal{P},\mathcal{A}^*(\mathsf{pk},\mathbf{List},\mathrm{Q}_{int})}(\mathsf{pk})$$

Definition 5.12 ($\mathsf{pqPA2\text{-}C_{dec}}$)**.** *We say a public-key encryption scheme* Enc *is* $\mathsf{pqPA2\text{-}C_{dec}}$ *plaintext-aware if for any* QPT *ciphertext-creator* $\mathcal{A}$*, there exists a* QPT *plaintext-extractor* $\mathcal{A}^*$ *such that for any* QPT *plaintext-creator* $\mathcal{P}$ *and any* QPT *distinguishing algorithms* $\mathcal{D}$*,* *the advantage of $\mathcal{D}$ in distinguishing $G_{real}^{\mathsf{pqPA2\text{-}C_{dec}}}$ and $G_{fake}^{\mathsf{pqPA2\text{-}C_{dec}}}$ is negligible as a function of the security parameter:*

$$\begin{aligned}\mathsf{Adv}_{\mathcal{D},\mathcal{A}} = |\Pr\Big[\mathcal{D}(\rho_\kappa) = 1 : \rho_\kappa \leftarrow G_{real}^{\mathsf{pqPA2\text{-}C_{dec}}}\Big] - \\ \Pr\Big[\mathcal{D}(\rho_\kappa) = 1 : \rho_\kappa \leftarrow G_{fake}^{\mathsf{pqPA2\text{-}C_{dec}}}\Big]| \leq \mathrm{negl}(\kappa),\end{aligned}$$

where the output of $\mathcal{D}$ is determined with the computational basis measurement.

5.3.3 Superposition decryption queries.

In this subsection, we define plaintext-awareness when the adversary $\mathcal{A}$ has classical access to a plaintext creator algorithm $\mathcal{P}$ and superposition access to the decryption oracle $\mathsf{Dec}'_{(\mathsf{sk},\mathbf{List})}$. Similarly, we define the notion using two games.

Game $G_{real}^{\mathsf{pqPA2\text{-}Q_{dec}}}$. In this game, the ciphertext-creator adversary $\mathcal{A}$ given pk has oracle access to $\mathcal{P}$ and superposition access to the decryption oracle. At the end, $\mathcal{A}$ outputs a quantum state.

Game $G_{real}^{\mathsf{pqPA2\text{-}Q_{dec}}}$

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa)$, $\rho_\kappa \leftarrow \mathcal{A}^{\mathcal{P},\mathbb{U}_{\mathsf{Dec}'_{(\mathsf{sk},\mathbf{List})}}}(\mathsf{pk})$

Game $G_{fake}^{\mathsf{pqPA2\text{-}Q_{dec}}}$. In this game, $\mathcal{A}$'s decryption queries will be responded to by a plaintext-extractor algorithm $\mathcal{A}^*$. Here, $\mathcal{A}^*$ given pk has access to **List** and the internal register of $\mathcal{A}$. At the end, $\mathcal{A}$ outputs a quantum state.

Game $G_{fake}^{\mathsf{pqPA2\text{-}Q_{dec}}}$

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa)$, $\rho_\kappa \leftarrow \mathcal{A}^{\mathcal{P},\mathcal{A}^*(\mathsf{pk},\mathbf{List},\mathsf{Q}_{int})}(\mathsf{pk})$

Definition 5.13 ($\mathsf{pqPA2\text{-}Q_{dec}}$)**.** *We say a public-key encryption scheme* Enc *is* $\mathsf{pqPA2\text{-}Q_{dec}}$ *plaintext-aware if for any* QPT *ciphertext-creator $\mathcal{A}$, there exists a* QPT *plaintext-extractor $\mathcal{A}^*$ such that that for any* QPT *plaintext-extractor $\mathcal{P}$ and any* QPT *distinguishing algorithms $\mathcal{D}$, the advantage of $\mathcal{D}$ in distinguishing $G_{real}^{\mathsf{pqPA2\text{-}Q_{dec}}}$ and $G_{fake}^{\mathsf{pqPA2\text{-}Q_{dec}}}$ is negligible as a function of the security parameter:*

$$\begin{aligned}\mathsf{Adv}_{\mathcal{D},\mathcal{A}} = |\Pr\Big[\mathcal{D}(\rho_\kappa) = 1 : \rho_\kappa \leftarrow G_{real}^{\mathsf{pqPA2\text{-}Q_{dec}}}\Big] - \\ \Pr\Big[\mathcal{D}(\rho_\kappa) = 1 : \rho_\kappa \leftarrow G_{fake}^{\mathsf{pqPA2\text{-}Q_{dec}}}\Big]| \leq \mathrm{negl}(\kappa),\end{aligned}$$

where the output of $\mathcal{D}$ is determined with the computational basis measurement.

5.4 Relationships Between Notions

In this section, we study the relations between different PA notions defined in this chapter. In addition, we show that the $\mathsf{pqPA2\text{-}Q_{dec}}$ plaintext-awareness notion defined in this chapter along with IND-qCPA security implies IND-qCCA security.

5.4.1 Relationships between PA notions

5.4.1.1 Implications.

First, we show the implications between the notions. Clearly, $\mathsf{pqPA}i\text{-}\mathsf{Q}_{\mathsf{dec}}$ plaintext-awareness implies $\mathsf{pqPA}i\text{-}\mathsf{C}_{\mathsf{dec}}$ plaintext-awareness for $i = 0, 1, 2$. The reason is the existence of a plaintext-extractor algorithm $\mathcal{A}^*$ for an adversary $\mathcal{A}$ that makes superposition queries to the decryption oracle is enough to prove $\mathsf{pqPA}i\text{-}\mathsf{C}_{\mathsf{dec}}$ plaintext-awareness. In other words, the algorithm $\mathcal{A}^*$ is a plaintext extractor for an adversary attacking in the sense of $\mathsf{pqPA}i\text{-}\mathsf{C}_{\mathsf{dec}}$.

Theorem 5.14. *For any $i = 0, 1, 2$, a public-key encryption scheme* Enc *that is* $\mathsf{pqPA}i\text{-}\mathsf{Q}_{\mathsf{dec}}$ *plaintext-aware, it is* $\mathsf{pqPA}i\text{-}\mathsf{C}_{\mathsf{dec}}$ *plaintext-aware.*

Below, we investigate the relations between PAi notions for different i.

Theorem 5.15. *If an encryption scheme is* $\mathsf{pqPA2}\text{-}Qu$ *aware then it is* $\mathsf{pqPA1}\text{-}Qu$ *aware when* $Qu \in \{\mathsf{C}_{\mathsf{dec}}, \mathsf{Q}_{\mathsf{dec}}\}$.

Proof. The proof is straightforward because an adversary $\mathcal{A}$ that breaks $\mathsf{pqPA1}\text{-}Qu$ awareness can be run to break $\mathsf{pqPA2}\text{-}Qu$ awareness. In more detail, the reduction adversary $\mathcal{B}$ runs $\mathcal{A}$ and simulates $\mathcal{A}$'s decryption queries using its decryption oracle. (Note that the reduction adversary $\mathcal{B}$ does not use the possibility of querying the plaintext-creator and breaks the $\mathsf{pqPA2}\text{-}Qu$ awareness notion.) □

Theorem 5.16. *If an encryption scheme is* $\mathsf{pqPA1}\text{-}Qu$ *aware then it is* $\mathsf{pqPA0}\text{-}Qu$ *aware when* $Qu \in \{\mathsf{C}_{\mathsf{dec}}, \mathsf{Q}_{\mathsf{dec}}\}$.

Proof. The proof is obvious since the only difference between $PA1$ and $PA0$ notions are the number of decryption queries, which is polynomially many queries and one query, respectively. □

5.4.1.2 Non-implications.

The rest of this subsection shows non-implications (i.e. separations) between notions. Note that in order to exclude the trivial encryption schemes that are plaintext-aware with respect to all definitions (for instance, the identity encryption), we add a security requirement (one-wayness or IND-qCPA security) for encryption in the separation theorems.

Below, we show that $\mathsf{pqPA}i\text{-}\mathsf{Q}_{\mathsf{dec}}$ is strictly stronger than $\mathsf{pqPA}i\text{-}\mathsf{C}_{\mathsf{dec}}$ for $i = 1, 2$. The high-level idea is to take an encryption scheme that is $\mathsf{pqPA}i\text{-}\mathsf{C}_{\mathsf{dec}}$ plaintext-aware and modifies its decryption algorithm in a way that remains $\mathsf{pqPA}i\text{-}\mathsf{C}_{\mathsf{dec}}$ plaintext-aware but it leaks a valid ciphertext to the $\mathsf{pqPA}i\text{-}\mathsf{Q}_{\mathsf{dec}}$ adversary.

Theorem 5.17. *A one-way* $\mathsf{pqPA}i\text{-}\mathsf{C}_{\mathsf{dec}}$ *plaintext-aware public-key encryption scheme is not necessarily* $\mathsf{pqPA}i\text{-}\mathsf{Q}_{\mathsf{dec}}$ *plaintext-aware for* $i = 1, 2$.

Proof. Let $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme that is $\mathsf{pqPA}i\text{-}\mathsf{C}_{\mathsf{dec}}$ plaintext-aware. Let $\{0,1\}^n$ be the ciphertext space of Π. Let the ciphertext c_v be generated by choosing a random message m and a randomness r and computing $\mathsf{Enc}(m; r)$. We modify Π to a new encryption scheme $\Pi' = (\mathsf{KeyGen}', \mathsf{Enc}', \mathsf{Dec}')$. The algorithm KeyGen' runs KeyGen to get $(\mathsf{pk}, \mathsf{sk})$, it outputs a key pk_{com} for a computationally hiding and binding commitment scheme $(\mathrm{Com}, \mathrm{Ver})$, and it chooses a random periodic function f on c_v. (That is for any $x \in \{0,1\}^n$, $f(x \oplus c_v) = f(x)$.) It returns the pair

$(\mathsf{pk}', \mathsf{sk}') = ((\mathsf{pk}, \mathsf{pk}_{com}), (\mathsf{sk}, f))$ and the commitment value $c_{com} = \mathrm{Com}(\mathsf{pk}_{com}, c_v)$ with the corresponding opening ω. For any message m in the message-space of Π, $\mathsf{Enc}'_{\mathsf{pk}'}(m) = \mathsf{Enc}_{\mathsf{pk}}(m)|| \perp$. The new decryption algorithm $\mathsf{Dec}'_{(\mathsf{sk},f)}$ takes as input a ciphertext from $(\{0,1\}^n \cup \perp) \times (\{0,1\}^n \cup \perp)$ and operates as the following:

$$\mathsf{Dec}'_{(\mathsf{sk},f,r)}(c_1, c_2) = \begin{cases} \mathsf{Dec}_{\mathsf{sk}}(c_1) & \text{if } c_1 \neq \perp \text{ and } c_2 = \perp \\ \mathsf{Dec}_{\mathsf{sk}}(c_v) \| r \| \omega & \text{if } c_1 = \perp \text{ and } c_2 = c_v \\ \perp & \text{if } c_1 = \perp \text{ and } c_2 \neq c_v \\ f(c_2) & \text{otherwise} \end{cases}.$$

Since $\mathsf{Dec}'_{\mathsf{sk}'}(\mathsf{Enc}'_{\mathsf{pk}'}(m)) = \mathsf{Dec}_{\mathsf{sk}}(\mathsf{Enc}_{\mathsf{pk}}(m))$, Π' satisfies the correctness property. It is clear that Π' is one-way since Π is one-way. We show that Π' is $\mathsf{pqPA1}$-$\mathsf{C_{dec}}$ plaintext-aware. Let $\mathcal{A}^*$ be the QPT plaintext-extractor algorithm for Π. We construct a QPT plaintext-extractor algorithm $\mathcal{A}'^*$ for Π'. Namely, $\mathcal{A}'^*$ chooses a random function f' with the same domain and co-domain as f and for any (c_1, c_2) operates as follows:

$$\mathcal{A}'^*(c_1, c_2) = \begin{cases} \mathcal{A}^*(c_1) & \text{if } c_1 \neq \perp \text{ and } c_2 = \perp \\ \perp & \text{if } c_1 = \perp \\ f'(c_2) & \text{otherwise} \end{cases}.$$

Note that an adversary with classical access to the decryption oracle is not able to get c_v. In addition, the commitment scheme is computationally hiding and c_{com} reveals c_v only with a negligible probability. Therefore, the decryption query $(\perp, c_v)$ will be submitted with a negligible probability. Since for a polynomial-time adversary with classical access to f and f', these two functions are indistinguishable, $\mathcal{A}'^*$ is a successful polynomial-time plaintext-extractor algorithm for Π'.

However, an adversary $\mathcal{A}$ with superposition access to Dec', can choose a random ciphertext c' from $\{0,1\}^n$ and queries $|c'\rangle \otimes \sum_{c \in \{0,1\}^n} \frac{1}{\sqrt{n}} |c\rangle$ to Dec'. Therefore, the adversary can employ Simon's quantum algorithm [Sim97]to obtain c_v and break $\mathsf{pqPA}i$-$\mathsf{Q_{dec}}$ plaintext-awareness. In more detail, $\mathcal{A}$ submits $(\perp, c_v)$ as a decryption query. After getting a response $m' \| r' \| \omega'$, it checks if $c_v = \mathsf{Enc}(m'; r')$ and $\mathrm{Ver}(\mathsf{pk}_{com}, c_{com}, c_v, \omega') = 1$. If both equalities hold, it returns 1, otherwise, it returns 0.

In the real case, the Dec' returns $\mathsf{Dec}_{\mathsf{sk}}(c_v) \| r$ and $\mathcal{A}$ outputs 1 with a high probability, namely the probability of Simon's algorithm succeeding. However, in the fake game, since Enc is one-way and the commitment scheme is computationally binding, there is no QPT algorithm $\mathcal{A}^*$ that can simulate an answer to the decryption query $(\perp, c_v)$ such that both equalities above hold with a non-negligible probability. So $\mathcal{A}$ returns 1 with a negligible probability in this case. Consequently, a distinguisher that returns the output of $\mathcal{A}$ can distinguish between the real game and the fake game with a non-negligible probability. □

We can use a similar trick to show that $\mathsf{pqPA0}$-$\mathsf{Q_{dec}}$ is strictly stronger than $\mathsf{pqPA0}$-$\mathsf{C_{dec}}$. Since Simon's algorithm needs a polynomial number of queries to extract c_v but in the $\mathsf{pqPA0}$-$\mathsf{C_{dec}}$ notion the adversary is only allowed to make a single query, we need to modify Dec' a bit further. Namely, we expand the ciphertext space and define Dec'' as the following:

$$\mathsf{Dec}''(c_1, c_2, \cdots, c_m) = \begin{cases} \mathsf{Dec}(c_1) & \text{if } c_1 \neq \perp \text{ and } c_2 = \cdots = c_m = \perp \\ \mathsf{Dec}_{\mathsf{sk}}(c_v) \| r \| \omega & \text{if } c_1 = c_3 = \cdots = c_m = \perp \text{ and } c_2 = c_v \\ \perp & \text{if } c_1 = \perp \text{ and } c_2 \neq c_v \\ f(c_2) || \cdots || f(c_m) & \text{otherwise} \end{cases}.$$

The adversary queries

$$|c\rangle \otimes \sum_{c\in\{0,1\}^n} \frac{1}{\sqrt{n}} |c\rangle \otimes \cdots \otimes \sum_{c\in\{0,1\}^n} \frac{1}{\sqrt{n}} |c\rangle$$

to Dec'' to extract c_v. (Note that m is big enough that Simon's algorithm returns c_v with a high probability.)

Corollary 5.18. *A one-way* $\mathsf{pqPA0\text{-}C_{dec}}$ *plaintext-aware public-key encryption scheme* Enc *is not necessarily* $\mathsf{pqPA0\text{-}Q_{dec}}$ *plaintext-aware.*

Therefore, we can conclude that even the strongest plaintext-awareness notion with classical decryption queries will not imply the weakest plaintext-awareness notion with quantum decryption queries.

Corollary 5.19. *A one-way* $\mathsf{pqPA2\text{-}C_{dec}}$ *plaintext-aware public-key encryption scheme* Enc *is not necessarily* $\mathsf{pqPA0\text{-}Q_{dec}}$ *plaintext-aware.*

Proof. The proof is similar to the proof of Corollary 5.18. □

In the theorem below, we show that an adversary with the ability to eavesdrop some ciphertexts is strictly stronger than an adversary without this ability. Namely, we show that an encryption scheme that is $\mathsf{pqPA1\text{-}Q_{dec}}$ plaintext-aware is not necessarily $\mathsf{pqPA2\text{-}C_{dec}}$ plaintext-aware. The high-level idea to show this claim is to design an encryption scheme that is malleable on the last bit, however, this malleability does not change the corresponding plaintext. In other words, if we flip the last bit of any ciphertext, we will get a valid ciphertext, but, without any change on the corresponding plaintext. A PA1 adversary is not able to use this malleability since this does not change the plaintext inside of the ciphertext. However, an PA2 adversary can obtain a valid ciphertext (c, b) by eavesdropping and change it to a new ciphertext $(c, b \oplus 1)$ where its corresponding plaintext is unknown to the adversary.

Theorem 5.20. *A public-key encryption scheme that is* $\mathsf{pqPA1\text{-}Q_{dec}}$ *plaintext-aware and IND-qCPA secure, it is not necessarily* $\mathsf{pqPA2\text{-}C_{dec}}$ *plaintext-aware.*

Proof. Let $\Pi = (\mathsf{Enc}, \mathsf{Dec}, \mathsf{KeyGen})$ be a $\mathsf{pqPA1\text{-}Q_{dec}}$ plaintext-aware. We construct the following encryption scheme Π':

- $\mathsf{KeyGen}' = \mathsf{KeyGen}$
- $\mathsf{Enc}'(m) = \mathsf{Enc}(m)\|0$
- $\mathsf{Dec}'(c\|b) = \mathsf{Dec}(c)$, where $b \in \{0, 1\}$

The IND-qCPA security of Π' is obtained easily by the IND-qCPA security of Π. We show that Π' is also $\mathsf{pqPA1\text{-}Q_{dec}}$ plaintext-aware. Let $\mathcal{A}$ be an adversary that attacks Π' in the sense of $\mathsf{pqPA1\text{-}Q_{dec}}$. We construct an adversary $\mathcal{B}$ that attacks Π. The adversary $\mathcal{B}$ runs $\mathcal{A}$ and answers its decryption queries as follows. Let Q_c, Q_b be the input quantum registers for c, b respectively. Let Q_{out} be the output quantum register. The adversary $\mathcal{B}$ upon receiving Q_c, Q_b, Q_{out} registers from $\mathcal{A}$, it forwards Q_c, Q_{out} registers to its decryption oracle. After getting back $\mathbb{U}_{\mathsf{Dec}}(Q_c Q_{out})$ from its decryption oracle, it sends all three registers to $\mathcal{A}$. It is clear that the decryption queries are simulated perfectly for $\mathcal{A}$. Since

Π is $\mathsf{pqPA1\text{-}Q_{dec}}$ plaintext-aware, there exists a plaintext-extractor algorithm $\mathcal{B}^*$ for $\mathcal{B}$. Now from $\mathcal{B}^*$, one can construct an extractor $\mathcal{A}^*$ for $\mathcal{A}$. Namely, $\mathcal{A}^*(c\|b) := \mathcal{B}^*(c)$.

However, Π' is not $\mathsf{pqPA2\text{-}C_{dec}}$ plaintext-aware. Let $\mathcal{A}$ be an adversary that sends two messages $m_0 := 0^n$ and $m_1 := 1^n$ as a query to its plaintext-creator $\mathcal{P}$. Upon receiving a ciphertext $(c\|0)$ from $\mathcal{P}$, it sends $(c\|1)$ as a decryption query. If the answer is 0^n, it returns 0, otherwise it returns 1. Consider a plaintext-creator algorithm $\mathcal{P}_b$ that upon receiving a query m_0, m_1, it sends m_b to Enc. Then, it forwards $(c_b\|0) := \mathsf{Enc}(m_b)$ to the adversary. Let $\mathcal{D}$ be a distinguisher that returns the output of $\mathcal{A}$. Proof by contrary, let's assume that Π' is $\mathsf{pqPA2\text{-}C_{dec}}$ plaintext-aware. Then, there exists a plaintext-extractor algorithm $\mathcal{A}^*$ that works for $(\mathcal{A}, \mathcal{P}_0, \mathcal{D})$ and $(\mathcal{A}, \mathcal{P}_1, \mathcal{D})$. That is,

$$G_{real}^{\mathsf{pqPA2\text{-}C_{dec}}}(\mathcal{A}, \mathbb{U}_{\mathsf{Dec}'}, \mathcal{P}_0, \mathcal{D}) \cong G_{fake}^{\mathsf{pqPA2\text{-}C_{dec}}}(\mathcal{A}, \mathcal{A}^*, \mathcal{P}_0, \mathcal{D})$$

and

$$G_{real}^{\mathsf{pqPA2\text{-}C_{dec}}}(\mathcal{A}, \mathbb{U}_{\mathsf{Dec}'}, \mathcal{P}_1, \mathcal{D}) \cong G_{fake}^{\mathsf{pqPA2\text{-}C_{dec}}}(\mathcal{A}, \mathcal{A}^*, \mathcal{P}_1, \mathcal{D})$$

It is clear that in the real case, $\mathcal{D}$ returns 0 with the probability 1 when $\mathcal{A}$ interacts with $\mathcal{P}_0$ and it returns 1 with the probability 1 when $\mathcal{A}$ interacts with $\mathcal{P}_1$. So these two games are distinguishable. Consequently, in the fake game, $\mathcal{A}$'s interaction with $\mathcal{P}_0$ is distinguishable from its interaction with $\mathcal{P}_1$. And this is a contradiction to the IND-qCPA security of Π'. Namely, an adversary $\mathcal{B}$ that runs $\mathcal{A}$ and answers its decryption queries with $\mathcal{A}^*$ and its queries to a plaintext creator with Π''s challenger can break IND-qCPA security of Π'. □

In the following theorem, we show that a one-way public-key encryption scheme that is plaintext-aware against adversaries that make a single quantum decryption query is not necessarily plaintext-aware against adversaries that make many classical decryption queries. The high-level idea is that the decryption oracle partially reveals a valid ciphertext in each query. In more details, we write a valid ciphertext c_v as XOR of two random values $c_v^{(1)}$ and $c_v^{(2)}$, that is $c_v = c_v^{(1)} \oplus c_v^{(2)}$. Then the decryption oracle reveals one of $c_v^{(1)}$ or $c_v^{(2)}$ randomly in each query. Obviously, the adversary with a single query is able to get one of $c_v^{(1)}$ or $c_v^{(2)}$ and that does not give any useful information. On the other hand, the adversary with many decryption queries is able to obtain c_v.

Theorem 5.21. *A one-way* $\mathsf{pqPA0\text{-}Q_{dec}}$ *plaintext-aware public-key encryption scheme is not necessarily* $\mathsf{pqPA1\text{-}C_{dec}}$ *plaintext-aware.*

Proof. Let $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a $\mathsf{pqPA0\text{-}Q_{dec}}$ plaintext-aware encryption scheme. Let c_v be a ciphertext that is generated by choosing a random message m and a randomness r and computing $\mathsf{Enc}(m; r)$. Let $c_v^{(1)}$ and $c_v^{(2)}$ be two random elements such that $c_v = c_v^{(1)} \oplus c_v^{(2)}$. We construct an encryption scheme $\Pi' = (\mathsf{KeyGen}', \mathsf{Enc}', \mathsf{Dec}')$. The algorithm KeyGen' runs KeyGen to get $(\mathsf{pk}, \mathsf{sk})$ and it outputs a key pk_{com} for a computationally hiding and binding commitment scheme (Com, Ver). That is, the outputs of KeyGen' are $((\mathsf{pk}, \mathsf{pk}_{com}), \mathsf{sk})$ and the commitment value $c_{com} = \mathrm{Com}(\mathsf{pk}_{com}, c_v)$ with the corresponding opening ω. Note that a QPT adversary is not able to compute c_v from c_{com} with a non-negligible probability since the commitment scheme is computationally hiding. Let $\omega = \omega^{(1)} \oplus \omega^{(2)}$ for random values $\omega^{(1)}$ and $\omega^{(2)}$. For any message m, $\mathsf{Enc}'(m) = \mathsf{Enc}(m)\|0$. Dec' is a probabilistic algorithm and is defined as:

$$\mathsf{Dec}'(c\|b) = \begin{cases} \mathsf{Dec}(c) & \text{if } \mathsf{Dec}(c) \neq \perp \text{ or } b = 0 \\ c_v^{(i)} \| r \| \omega^{(i)} \text{ for a random i} \in \{0, 1\} & \text{if } b = 1 \text{ and } \mathsf{Dec}(c) = \perp \end{cases}.$$

It is clear that $\mathsf{Dec}'(\mathsf{Enc}'(m)) = m$ with the probability 1. We make a convention that for any bit string x, $x \oplus \perp = x$. We show that Π' is $\mathsf{pqPA0\text{-}Q_{dec}}$ plaintext-aware. Let $\mathcal{A}$ be an adversary that attacks Π' in the sense of $\mathsf{pqPA0\text{-}Q_{dec}}$. From $\mathcal{A}$, we construct an adversary $\mathcal{B}$ that attacks Π in the sense of $\mathsf{pqPA0\text{-}Q_{dec}}$. The adversary $\mathcal{B}$ runs $\mathcal{A}$ and answers to its decryption query as follows. Let Q_c, Q_b be the quantum input registers to store the c-part and the b-part of the ciphertext, respectively. Let Q_{out} be a register to store the output. The adversary $\mathcal{B}$ upon receiving these three registers Q_c, Q_b, Q_{out}, it forwards Q_c, Q_{out} to its decryption oracle. After getting $\mathbb{U}_{\mathsf{Dec}}(Q_c Q_{out})$ back from its decryption oracle, it applies a control operator $\mathbb{U}_{cnt}$ on Q_c, Q_b, Q_{out}. The unitary $\mathbb{U}_{cnt}$ XORs a classical random value $c' \| r' \| \omega'$ to the Q_{out} register if $b = 1$ and $\mathsf{Dec}(c) = \perp$. Otherwise, $\mathbb{U}_{cnt}$ is identity. It is clear that the decryption query is simulated perfectly. Since Π is $\mathsf{pqPA0\text{-}Q_{dec}}$, there exists a successful plaintext-extractor $\mathcal{B}^*$ for $\mathcal{B}$. Now we construct a successful plaintext-extractor for $\mathcal{A}$. Namely,

$$\mathcal{A}^*(c\|b) = \begin{cases} \mathcal{B}^*(c) & \text{if } \mathcal{B}^*(c) \neq \perp \text{ or } b = 0 \\ c'\|r'\|\omega' & \text{if } b = 1 \text{ and } \mathcal{B}^*(c) = \perp \end{cases},$$

where c', r' and ω' are random values.

The encryption scheme Π' is not $\mathsf{pqPA1\text{-}C_{dec}}$ aware since an adversary $\mathcal{A}$ is able to obtain c_v, ω, and the corresponding randomness r. It then sends c_v as a decryption query to get m'. Then it outputs 1 if $c_v = \mathsf{Enc}(m'; r)$ and $\mathrm{Ver}(\mathsf{pk}_{com}, c_{com}, c_v, \omega) = 1$. Otherwise, it returns 0. It is clear that in the real case, $\mathcal{A}$ outputs 1 with the probability 1. However, in the fake game, $\mathcal{A}$ outputs 0 with a non-negligible probability since Π is one-way and the commitment scheme is computationally binding. □

5.4.2 Relation with IND-qCCA

First, we show that IND-qCPA security and $\mathsf{pqPA2\text{-}C_{dec}}$ plaintext-awareness notions are not enough to conclude IND-qCCA security. The proof technique is similar to the proof of Theorem 5.17.

Theorem 5.22. *A public-key encryption scheme* Enc *that is* $\mathsf{pqPA2\text{-}C_{dec}}$ *plaintext-aware and IND-qCPA secure is not necessarily IND-qCCA secure.*

Proof. Let Enc with the decryption algorithm Dec be a public-key encryption scheme that is $\mathsf{pqPA2\text{-}C_{dec}}$ plaintext-aware and IND-qCPA. Let $\{0,1\}^n$ is the ciphertext space of Enc. We modify Dec to a new decryption algorithm Dec' in which it takes as input a ciphertext from $\{0,1\}^n \times \{0,1\}^n$ and operates as the following:

$$\mathsf{Dec}'(c_1, c_2) = \begin{cases} \mathsf{Dec}(c_1) \| \perp & \text{if } \mathsf{Dec}(c_1) \neq \perp \\ \perp \| f(c_2) & \text{otherwise} \end{cases},$$

where f is a periodic function on the secret key sk. (That is for any $x \in \{0,1\}^n$, $f(x \oplus \mathsf{sk}) = f(x)$.) It is clear that Enc remains $\mathsf{pqPA1\text{-}C_{dec}}$ plaintext-aware and IND-qCPA secure with this modification to Dec since exponential classical decryption queries are needed to recover sk. However, an adversary with superposition access to Dec', can choose a random ciphertext c' from $\{0,1\}^n$ and queries $|c'\rangle \otimes \sum_{c \in \{0,1\}^n} \frac{1}{\sqrt{n}} |c\rangle$ to Dec'. Since Enc is $\mathsf{pqPA}i\text{-}\mathsf{C_{dec}}$ plaintext-aware, with overwhelming probability $\mathsf{Dec}(c') = \perp$. Therefore, the adversary can employ Simon's quantum algorithm [Sim97] to obtain sk and breaks IND-qCCA security. □

Since $\mathsf{pqPA2\text{-}C_{dec}}$ plaintext-awareness notion implies classical PA2 notion, we can conclude that PA2 + IND-qCPA notion does not imply IND-qCCA security.

Corollary 5.23. *A public-key encryption scheme* Enc *that is PA2 plaintext-aware and IND-qCPA secure is not necessarily IND-qCCA secure.*

In the theorem below, we show that a plaintext-awareness notion that allows quantum decryption queries, namely the $\mathsf{pqPA2\text{-}Q_{dec}}$ notion, along with the IND-qCPA notion is enough to imply IND-qCCA security.

Theorem 5.24. *Any public-key encryption scheme* Enc *that is* $\mathsf{pqPA2\text{-}Q_{dec}}$ *plaintext-aware and IND-qCPA secure is IND-qCCA secure.*

Proof. Let $\mathcal{X}$ be a QPT adversary that attacks the encryption scheme Enc in the sense of IND-qCCA. We start with IND-qCCA game with the challenge bit $b = 0$ (G_0^{qCCA}) and reach the IND-qCCA game with the challenge bit 1 (G_1^{qCCA}) by introducing intermediate games that are in a negligible distance.

Game 0: G_0^{qCCA}

$$\begin{array}{rl} (\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa), & m_0, m_1 \leftarrow \mathcal{X}^{\mathbb{U}_{\mathsf{Dec}'_{(\mathsf{sk},\mathbf{List})}}}(\mathsf{pk}), \\ \mathsf{Enc}(m_0;r_0) \leftarrow \text{Challenger}(m_0,m_1), & b \leftarrow \mathcal{X}^{\mathbb{U}_{\mathsf{Dec}'_{(\mathsf{sk},\mathbf{List})}}}(\mathsf{pk},\mathsf{Enc}(m_0;r_0)) \end{array}$$

Let $\mathcal{P}_0$ be a plaintext-creator that upon receiving a query of type m_0, m_1 chooses a randomness r_0 and returns $\mathsf{Enc}(m_0, r_0)$. We replace the challenger in $G_{b=0}^{qCCA}$ with $\mathcal{P}_0$ to reach Game 1.

Game 1: $G_{real}^{\mathsf{pqPA2\text{-}Q_{dec}}}$ *with* $\mathcal{P}_0$

$$\begin{array}{rl} (\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa), & m_0, m_1 \leftarrow \mathcal{X}^{\mathbb{U}_{\mathsf{Dec}'_{(\mathsf{sk},\mathbf{List})}}}(\mathsf{pk}), \\ \mathsf{Enc}(m_0;r_0) \leftarrow \mathcal{P}_0(m_0,m_1), & b \leftarrow \mathcal{X}^{\mathbb{U}_{\mathsf{Dec}'_{(\mathsf{sk},\mathbf{List})}}}(\mathsf{pk},\mathsf{Enc}(m_0;r_0)) \end{array}$$

It is obvious that Game 0 and Game 1 are indistinguishable.

Since Enc is $\mathsf{pqPA2\text{-}Q_{dec}}$ aware there exists a successful ciphertext extractor $\mathcal{A}^*$ for $\mathcal{X}$. Let Q_{int} be the internal register of $\mathcal{X}$. In Game 2, we replace the decryption oracle with $\mathcal{A}^*$.

Game 2: $G_{fake}^{\mathsf{pqPA2\text{-}Q_{dec}}}$ *with* $\mathcal{P}_0$

$$\begin{array}{rl} (\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa), & m_0, m_1 \leftarrow \mathcal{X}^{\mathcal{A}^*(\mathsf{pk},\mathrm{Q}_{int})}, \\ \mathsf{Enc}(m_0;r_0) \leftarrow \mathcal{P}_0, & b \leftarrow \mathcal{X}^{\mathcal{A}^*(\mathsf{pk},\mathbf{List},\mathrm{Q}_{int})}(\mathsf{pk},\mathsf{Enc}(m_0;r_0)) \end{array}$$

Since $\mathcal{A}^*$ is a successful ciphertext extractor for $\mathcal{X}$, Game 1 and Game 2 are indistinguishable.

Let $\mathcal{P}_1$ be a plaintext-creator algorithm that upon receiving a query of type m_0, m_1 chooses randomness r_1 and returns $\mathsf{Enc}(m_1;r_1)$. We replace $\mathcal{P}_0$ with $\mathcal{P}_1$ in Game 2 to reach Game 3.

Game 3: $G^{\mathsf{pqPA2\text{-}Q_{dec}}}_{fake}$ *with* $\mathcal{P}_1$

$$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa), \qquad m_0, m_1 \leftarrow \mathcal{X}^{\mathcal{A}^*(\mathsf{pk},\mathsf{Q}_{int})},$$
$$\mathsf{Enc}(m_1; r_1) \leftarrow \mathcal{P}_1, \qquad b \leftarrow \mathcal{X}^{\mathcal{A}^*(\mathsf{pk},\mathbf{List},\mathsf{Q}_{int})}(\mathsf{pk}, \mathsf{Enc}(m_1; r_1))$$

Since Enc is IND-qCPA secure, Game 2 and Game 3 are indistinguishable. In more detail, let us assume there is a distinguisher $\mathcal{D}$ with a non-negligible advantage for these two games. Now $\mathcal{Y} = (\mathcal{X}, \mathcal{A}^*, \mathcal{D})$ is an adversary to break IND-qCPA security of Enc that is a contradiction.

In Game 4, we replace $\mathcal{A}^*$ with the decryption oracle.

Game 4

$$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa), \qquad m_0, m_1 \leftarrow \mathcal{X}^{\mathbb{U}_{\mathsf{Dec}'(\mathsf{sk},\mathbf{List})}}(\mathsf{pk}),$$
$$\mathsf{Enc}(m_1; r_1) \leftarrow \mathcal{P}_1(m_0, m_1), \quad b \leftarrow \mathcal{X}^{\mathbb{U}_{\mathsf{Dec}'(\mathsf{sk},\mathbf{List})}}(\mathsf{pk}, \mathsf{Enc}(m_1; r_1))$$

Since $\mathcal{A}^*$ is a successful plaintext-extractor for $\mathcal{X}$, these two games are indistinguishable.

Finally, we replace $\mathcal{P}_1$ with the challenger in Game 5 to reach G_1^{qCCA}.

Game 5: G_1^{qCCA}

$$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa), \qquad m_0, m_1 \leftarrow \mathcal{X}^{\mathbb{U}_{\mathsf{Dec}'(\mathsf{sk},\mathbf{List})}}(\mathsf{pk}),$$
$$\mathsf{Enc}(m_1; r_1) \leftarrow \text{Challenger}(m_0, m_1), \quad b \leftarrow \mathcal{X}^{\mathbb{U}_{\mathsf{Dec}'(\mathsf{sk},\mathbf{List})}}(\mathsf{pk}, \mathsf{Enc}(m_1; r_1))$$

It is clear that Game 4 and Game 5 are indistinguishable. And this finishes the proof. □

5.5 Achievability

In this section, we lift a public-key encryption scheme that is PA2 plaintext-aware against a quantum adversary (PA2 notion with classical decryption) to an encryption scheme that is $\mathsf{pqPA2\text{-}Q_{dec}}$.

Let $\Pi^{asy} = (\mathsf{KeyGen}^{asy}, \mathsf{Enc}^{asy}, \mathsf{Dec}^{asy})$ be a public-key encryption scheme that is PA2 plaintext-aware against QPT adversaries. We construct a public-key encryption scheme $\Pi^{hyb} = (\mathsf{KeyGen}^{hyb}, \mathsf{Enc}^{hyb}, \mathsf{Dec}^{hyb})$ and show that it is $\mathsf{pqPA2\text{-}Q_{dec}}$. The encryption scheme Π^{hyb} is defined as :

- The algorithm KeyGen^{hyb} on input of the security parameter κ runs $\mathsf{KeyGen}^{asy}(\kappa)$ and returns its output $(\mathsf{pk}, \mathsf{sk})$.

- For any message $m \in \{0,1\}^n$, the algorithm Enc^{hyb} chooses a randomness r and returns $\mathsf{Enc}^{asy}_{\mathsf{pk}}(r)||qPRP_r(m||0^k)$ where $qPRP$ is a strong quantum-secure pseudorandom permutation and k depends on the security parameter κ.

- For any ciphertext (c_1, c_2), Dec^{hyb} first decrypts c_1 using sk, if the output is $\perp$, it returns $\perp$. Otherwise, it uses the output as the key for $qPRP$ to decrypt c_2. If the k_1 least significant bits of the outcome is not 0, it returns $\perp$, otherwise it returns the n most significant bits of the outcome.

$$\mathsf{Dec}^{hyb}(c_1, c_2) = \begin{cases} \bot & \text{if } \mathsf{Dec}^{asy}_{\mathsf{sk}}(c_1) = \bot \\ \bot & \text{if } [qPRP^{-1}_{\mathsf{Dec}^{asy}_{\mathsf{sk}}(c_1)}(c_2)]_k \neq 0^k \\ [qPRP^{-1}_{\mathsf{Dec}^{asy}_{\mathsf{sk}}(c_1)}(c_2)]^n & \text{otherwise} \end{cases}.$$

Theorem 5.25. *Under the assumption of the existence of a quantum one-way function, the public-key encryption scheme* $\Pi^{hyb} = (\mathsf{KeyGen}^{hyb}, \mathsf{Enc}^{hyb}, \mathsf{Dec}^{hyb})$ *described above is* $\mathsf{pqPA2\text{-}Q_{dec}}$.

Proof. Let $\mathcal{A}$ be an adversary that attacks Π^{hyb} in the sense of $\mathsf{pqPA2\text{-}Q_{dec}}$. We construct an adversary $\mathcal{B}$ that attacks Π^{asy} in the sense of PA2. Let $\mathcal{P}_B$ be a plaintext-creator adversary that upon receiving a query, chooses a randomness r and sends it to the encryption oracle Π^{asy} to receive $\mathsf{Enc}^{asy}_{\mathsf{pk}}(r)$. Then it sends $\mathsf{Enc}^{asy}_{\mathsf{pk}}(r)$ to the ciphertex-creator adversary. The adversary $\mathcal{B}$ runs $\mathcal{A}$ and answers to the decryption queries as follows. When $\mathcal{A}$ makes a decryption query $\sum_{c_2} \alpha_{c_2} |c_1\rangle |c_2\rangle$, the adversary $\mathcal{B}$ forwards only the first part of the ciphertext (c_1) to its oracle. (Note that c_1 is a classical value and it is not entangled with the rest of the query. So forwarding the c_1-part does not disturb the decryption query.) If its oracle on input c_1 returns $\bot$, $\mathcal{B}$ returns $\bot$. Otherwise, if its oracle on input c_1 returns r ($\neq\bot$), $\mathcal{B}$ uses r as the key for $qPRP$ to decrypt the c_2-part. Note that if the k_1 least significant bits of $qPRP_r^{-1}(c_2)$ is not zero, the output of $\mathcal{B}$ will be $\bot$. Otherwise, the output will be the n most significant bits of $qPRP_r^{-1}(c_2)$. When $\mathcal{A}$ makes a query m to its plaintext-creator $\mathcal{P}_A$, $\mathcal{B}$ makes a query to $\mathcal{P}_B$ to receive the ciphertext c_1. Then it sends $(c_1, \pi(m||0^k))$ to $\mathcal{A}$ where π is a random permutation. Since Π^{asy} is PA2, there exists a ciphertex extractor $\mathcal{B}^*$ for $\mathcal{B}$.

Now we consider the ciphertex extractor $\mathbb{U}_{\mathcal{A}_1^*}$ where for any (c_1, c_2),

$$\mathcal{A}_1^*(c_1, c_2) = \begin{cases} \bot & \text{if } \mathcal{B}^*(c_1) = \bot \\ \bot & \text{if } [qPRP^{-1}_{\mathcal{B}^*(c_1)}(c_2)]_k \neq 0^k \\ [qPRP^{-1}_{\mathcal{B}^*(c_1)}(c_2)]^n & \text{otherwise} \end{cases}.$$

We show that $\mathbb{U}_{\mathcal{A}_1^*}$ is a successful plaintext-extractor for $\mathcal{A}$ in the following.

Game 0: We start with $G^{\mathsf{pqPA2\text{-}Q_{dec}}}_{real}$ that is run by a plaintext-creator $\mathcal{P}_A$ and a distinguisher $\mathcal{D}$.
Game 1: We change the plaintex creator $\mathcal{P}_A$ to a new plaintext-creator $\mathcal{P}'_B$ that upon receiving a query m runs $\mathcal{P}_B$ to obtain c_1, then it chooses a random permutation π and returns $(c_1, \pi(m||0^k))$. We show that these two games are indistinguishable. An observation is that the first part of the $\mathcal{P}_A$'s output (c_1) is independent of $\mathcal{P}_A$ since it is the encryption of a random string that is chosen by the encryption algorithm. In other words, the c_1-part is generated exactly the same by $\mathcal{P}_A$ and $\mathcal{P}'_B$. The indistinguishability of the c_2-part holds as well since a quantum-secure pseudo-random permutation is indistinguishable from a random permutation.
Game 2: In this game, the decryption queries will be answered by $\mathbb{U}_{\mathcal{A}_1^*}$. An observation is that $\mathcal{A}_1^*$ is indistinguishable from Dec^{hyb} because $\mathcal{B}^*$ is indistinguishable from $\mathsf{Dec}^{asy}_{\mathsf{sk}}$ (the rest of $\mathcal{A}_1^*$ and Dec^{hyb} are the same). Therefore, these two games remain indistinguishable. In other words, these two games are indistinguishable because $\mathcal{B}^*$ is a successful plaintext-extractor for $\mathcal{B}$.

Game 3: In the last game, we replace the plaintext-creator $\mathcal{P}'_B$ with $\mathcal{P}_A$. The same reasoning as Game 0,1 shows that Game 2 and Game 3 are indistinguishable and this finishes the proof. □

5.5.1 OAEP transform

The main motivation to present the first definition for plaintext-awareness notion [BR95] was to show the security of Optimal Asymmetric Encryption Padding (OAEP). Even though our definitions for PA notions are in the standard model, we argue that these definitions apply to the random oracle model as well because queries to the random oracles are a part of the internal register of the adversary. We briefly explain why we think OAEP is $\mathsf{pqPA1}$-$\mathsf{Q_{dec}}$ plaintext-aware. We take this from a recent work on the IND-qCCA security of OAEP transform [Ebr22]. There, Ebrahimi started with the actual decryption algorithm $\mathbb{U}_{\mathsf{Dec}}$ and introduced a sequence of indistinguishable decryption algorithms to construct a decryption algorithm $\mathbb{U}_{\mathsf{Dec}^{(4)}}$ that does not use the secret key. (Since the queries to the random oracles are quantum, Zhandry's compressed oracle technique [Zha19] has been used in [Ebr22].) This decryption algorithm $\mathbb{U}_{\mathsf{Dec}^{(4)}}$ can be invoked by a plaintext-extractor adversary $\mathcal{A}^*$ in the fake game. The indistinguishably of $\mathbb{U}_{\mathsf{Dec}}$ and $\mathbb{U}_{\mathsf{Dec}^{(4)}}$ gives us the $\mathsf{pqPA1}$-$\mathsf{Q_{dec}}$ plaintext-awareness. However, whether OAEP is $\mathsf{pqPA2}$-$\mathsf{Q_{dec}}$ plaintext-aware or not is an open question. The reason is the random oracle queries that are submitted by a plaintext-creator $\mathcal{P}$ are not accessible by $\mathcal{A}^*$. So $\mathbb{U}_{\mathsf{Dec}^{(4)}}$ sketched above is not able to decrypt a ciphertext that is obtained by indirect (for instance by a malleability of a ciphertext obtained from $\mathcal{P}$) use of these random oracle queries.

5.6 Quantum Plaintext-awareness

In the previous sections, we discussed plaintext-awareness for classical schemes in a quantum setting. In this section, we consider quantum encryption schemes instead, showing how we can define and achieve plaintext-awareness in the quantum setting. We will establish the same variants as in previous sections.

Definition 5.26 (QPA1)**.** *Let* G^{QPA1}_{real} *and* G^{QPA1}_{fake} *be* $G^{pqPA1-C_{dec}}_{real}$ *and* $G^{pqPA1-C_{dec}}_{fake}$ *respectively, except with quantum oracles* $\mathcal{A}^*$ *and* Dec.

We say a quantum public-key encryption scheme Enc *is* $\mathsf{QPA1}$ *plaintext aware if for any quantum polynomial-time ciphertext creators* $\mathcal{A}$*, there exists a quantum polynomial-time plaintext extractor* $\mathcal{A}^*$ *such that for all quantum polynomial-time distinguishing algorithms* $\mathcal{D}$*, the advantage of* $\mathcal{D}$ *in distinguishing* G^{QPA1}_{real} *and* G^{QPA1}_{fake} *is negligible as a function of the security parameter:*

$$\mathsf{Adv}_{\mathcal{D},\mathcal{A}} = |\Pr\Big[\mathcal{D}(\rho_\kappa) = 1 : \rho_\kappa \leftarrow G^{\mathsf{QPA1}}_{real}\Big] - \Pr\Big[\mathcal{D}(\rho_\kappa) = 1 : \rho_\kappa \leftarrow G^{\mathsf{QPA1}}_{fake}\Big]| \leq \mathrm{negl}(\kappa),$$

where the output of $\mathcal{D}$ *is determined with the computational basis measurement.*

We use the hybrid construction from Construction 2.7 to construct a PKQES that satisfies $\mathsf{QPA1}$. Intuitively, it is clear why plaintext-awareness of the PKES directly translates to plaintext-awareness of the hybrid scheme; if a plaintext extractor exists that can extract the symmetric key k from the second part of the hybrid ciphertext, then this key can trivially be used to decrypt the first part, thus extracting the quantum plaintext.

Theorem 5.27. *Let $\Pi^{\mathsf{Cl}} = (\mathsf{KeyGen}^{\mathsf{Cl}}, \mathsf{Enc}^{\mathsf{Cl}}, \mathsf{Dec}^{\mathsf{Cl}})$ be a $pqPA1 - Q_{dec}$ plaintext aware PKES and $\Pi^{\mathsf{Qu}} = (\mathsf{KeyGen}^{\mathsf{Qu}}, \mathsf{Enc}^{\mathsf{Qu}}, \mathsf{Dec}^{\mathsf{Qu}})$ any SKQES. Then $\Pi^{\mathsf{Hyb}}[\Pi^{\mathsf{Qu}}, \Pi^{\mathsf{Cl}}]$ is a $PA1$ plaintext aware PKQES.*

Proof. Let $\mathcal{A}^{\mathcal{O}}$ be an arbitrary quantum polynomial-time ciphertext creator for $\Pi^{\mathsf{Hyb}}[\Pi^{\mathsf{Qu}}, \Pi^{\mathsf{Cl}}]$ with access to the quantum oracle $\mathcal{O} \in \{\mathcal{A}^*, \mathsf{Dec}_{\mathsf{sk}}\}$ (where $\mathcal{A}^*$ is defined later). Define $\mathcal{B}^{\mathcal{O}'}$ as $\mathcal{A}$, but with every oracle call replaced by the following procedure:

1. Upon input ρ, call $\mathcal{O}'$ to decrypt the second part of ρ.

2. Decrypt and then return the first part of ρ using $\mathsf{Dec}^{\mathsf{Qu}}$.

Note that by construction, when setting $\mathcal{O}' = \mathbf{U}_{\mathsf{Dec}^{\mathsf{Cl}}_{sk}}$, this procedure is equivalent to calling $\mathsf{Dec}^{\mathsf{Hyb}}(\rho)$. Furthermore, since Π^{Cl} is $pqPA1 - Q_{dec}$ plaintext-aware, there exists a plaintext extractor $\mathcal{B}^*$ such that for every $\mathcal{D}$, $\mathsf{Adv}_{\mathcal{D},\mathcal{B}} \leq neg(\kappa)$, where Adv is as in the definition of $pqPA1 - Q_{dec}$.

Now define $\mathcal{A}^*$ to be the above procedure with $\mathcal{O}' = \mathcal{B}^*$. This means that, by construction, playing G^{QPA1}_{fake} using $\mathcal{A}$ and $\mathcal{A}^*$ is the same as playing $G^{pqPA1-Q_{dec}}_{fake}$ with $\mathcal{B}$ and $\mathcal{B}^*$. Furthermore, we observed earlier that playing G^{QPA1}_{real} using $\mathcal{A}$ and $\mathsf{Dec}^{\mathsf{Hyb}}_{\mathsf{sk}}$ is the same as playing $G^{pqPA1-Q_{dec}}_{real}$ with $\mathcal{B}$ and $\mathbf{U}_{\mathsf{Dec}^{\mathsf{Cl}}_{sk}}$. It follows that for any $\mathcal{D}$, $\mathsf{Adv}_{\mathcal{D},\mathcal{A}} = \mathsf{Adv}_{\mathcal{D},\mathcal{B}} \leq \mathrm{negl}(\kappa)$. □

Similar to the classical case, plaintext awareness can also be used in the quantum setting to boost IND-CPA security to IND-CCA security.

Definition 5.28 ($QIND$,[Ala+16])**.** *A PKQES is QIND-CPA secure if for every QPT adversary $\mathcal{A} = (\mathcal{M}, \mathcal{D})$ we have:*

$$\left\| \Pr\left[\mathcal{D}\{(\mathsf{Enc}_{\mathsf{pk}} \otimes \mathbb{I})\rho^{ME}\} = 1\right] - \Pr\left[\mathcal{D}\{(\mathsf{Enc}_{\mathsf{pk}} \otimes \mathbb{I}) |0\rangle \langle 0|^{M} \otimes \rho^{E}\} = 1\right] \right\| \leq \mathrm{negl}(\kappa)$$

where $\rho^{ME} \leftarrow \mathcal{M}(\mathsf{pk})$, $\rho^{E} = \mathrm{Tr}_M\left[\rho^{ME}\right]$ and probability is taken over $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}$ and the internal randomness of Enc and $\mathcal{A}$. A PKQES is QIND-CCA1 secure if in the QIND-CPA definition, $\mathcal{M}$ is given oracle access to $\mathsf{Dec}_{\mathsf{sk}}$.

Theorem 5.29. *Any PKQES Π that is QIND-CPA secure and PA1 plaintext aware is QIND-CCA1 secure.*

Proof. Let $\mathcal{A} = (\mathcal{D}, \mathcal{M}^{\mathsf{Dec}_{\mathsf{sk}}})$ be an arbitrary QIND-CCA1 adversary for Π. Consider $\mathcal{M}$ as a ciphertext creator for QPA1. Since Π is QPA1 plaintext aware, there exists a plaintex extractor $\mathcal{M}^*$ such that $\rho^{ME} \leftarrow \mathcal{M}^{\mathsf{Dec}_{\mathsf{sk}}}(pk)$ and $\rho'^{ME} \leftarrow \mathcal{M}^{\mathcal{M}^*(\mathsf{pk}, Q_{int})}$ cannot be distinguished by any QPT distinguisher. In particular, they cannot be distinguished by the following distinguishers $\mathcal{D}'_b(\rho^{ME})$:

- if $b = 1$: replace the M register with $|0\rangle \langle 0|^{M}$
- $C \leftarrow \mathsf{Enc}_{\mathsf{pk}}(M)$
- Output $\mathcal{D}(CE)$

Consider $(\mathcal{D}, \mathcal{M}')$ as a QIND-CPA adversary for Π, where $\mathcal{M}'$ is $\mathcal{M}$ except all decryption queries are simulated using $\mathcal{M}^*$. Then we have

$$\begin{aligned}
\Pr\left[\mathcal{D}\{(\mathsf{Enc}_{\mathsf{pk}} \otimes \mathbb{I})\rho^{ME}\} = 1\right] &= \Pr\left[\mathcal{D}_0'(\rho^{ME}) = 1\right] \\
&\approx \Pr\left[\mathcal{D}_0'({\rho'}^{ME}) = 1\right] \\
&= \Pr\left[\mathcal{D}\{(\mathsf{Enc}_{\mathsf{pk}} \otimes \mathbb{I}){\rho'}^{ME}\} = 1\right] \\
&\approx \Pr\left[\mathcal{D}\{(\mathsf{Enc}_{\mathsf{pk}} \otimes \mathbb{I})(|0\rangle\langle 0|^{M} \otimes {\rho'}^{E})\} = 1\right] \\
&= \Pr\left[\mathcal{D}_1'({\rho'}^{ME}) = 1\right] \\
&\approx \Pr\left[\mathcal{D}_1'(\rho^{ME}) = 1\right] \\
&= \Pr\left[\mathcal{D}\{(\mathsf{Enc}_{\mathsf{pk}} \otimes \mathbb{I})(|0\rangle\langle 0|^{M} \otimes \rho^{E})\} = 1\right]
\end{aligned}$$

where $A \approx B$ means $\|A - B\| \leq \text{negl}(\kappa)$. This means $\| \Pr\left[\mathcal{D}\{(\mathsf{Enc}_{\mathsf{pk}} \otimes \mathbb{I})\rho^{ME}\} = 1\right] - \Pr\left[\mathcal{D}\{(\mathsf{Enc}_{\mathsf{pk}} \otimes \mathbb{I})(|0\rangle\langle 0|^{M} \otimes \rho^{E})\} = 1\right] \| \leq \text{negl}(\kappa)$ and thus Π is QIND-CCA1 secure. □

5.6.1 Quantum PA2

While the definition of QPA1 is a natural extension of the pqPA1 definitions, the PA2 case does not present such a natural extension. The reason for this is mostly the **List** construct used to record ciphertexts that the adversary has eavesdropped. Of course, it is not possible to maintain copies of arbitrarily produced ciphertexts, but some efforts have been made to facilitate this. In [AGM18] a method of recording is presented where the challenger simply encrypts half of an entangled pair $|\phi^+\rangle$, and uses a projective measurement on the space of this single state to test authenticity. This method is further refined in [AM17] to allow an adversary to provide an arbitrary (unitary inducing a) message space. However, both these methods only work for recording a single state, and the recording of multiple, possibly entangled, states remains an open issue.

Another way of implementing a decryption oracle that works on every ciphertext except the challenge one is also presented in [AGM18]. This method is used in the QIND-CCA2 definition and works in a counterfactual way. In a first game, instead of restricting the oracle, the adversary is simply given access to a decryption oracle (that works on all states), with the objective of distinguishing an adversarially prepared state from a random state. In a second game, the adversary always gets the encryption of half of an entangled state from the challenger and is caught cheating if it queries this state to the decryption oracle. The idea behind this is that if the adversary would query the challenge ciphertext to the decryption oracle in the first game, it would also do so in the second game and get caught. Thus if one subtracts the probability of getting caught cheating from the winning probability of the adversary in the first game, one obtains a probability that an adversary can distinguish a real challenge ciphertext from a random one without querying it to the decryption oracle.

We will use this second form of recording to define a version of QPA2. In the definition below, the 'test' game is where the adversary is challenged to distinguish a decryption oracle from a plaintext-extractor in the same way an adversary would be challenged to distinguish between the real and fake games in the PA1 definition. To keep the adversary from cheating, a 'cheat' game is used to determine the probability that the adversary will try to decrypt a ciphertext created by the plaintext creator. This 'cheat' game will make use of a list of entangled pairs to catch a cheating adversary. A critical part of "recording" such a list of ciphertexts in this way is to characterize the encryption as a unitary map. This way, one can start with a known state, transform it into a ciphertext, but one is able to undo this process. If one uses half of an entangled state as the input to this process, the authenticity of the state after undoing the encryption can then be verified by a measurement on the space spanned by the original entangled pair.

Using the decomposition from Lemma 2.8, we define QPA2.

```
Game $G^{\mathsf{QPA2\text{-}Test}}_{\mathcal{D},\mathcal{P},\Pi,\mathcal{A}}(b)$
    $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa)$
    If $b = 0$: $\rho_\kappa \leftarrow \mathcal{A}^{\mathcal{P}, \mathsf{Dec}_{\mathsf{sk}}}(\mathsf{pk})$
    If $b = 1$: $\rho_\kappa \leftarrow \mathcal{A}^{\mathcal{P}, \mathcal{A}^*(\mathsf{pk}, Q_{int})}(\mathsf{pk})$
    Output $\mathcal{D}(\rho_\kappa)$
```

```
Game $G^{\mathsf{QPA2\text{-}Cheat}}_{\Pi,\mathcal{A}}(b)$
    $k = (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa)$; $\textbf{List} = \emptyset$
    $\mathcal{P}'(\cdot)$
        $MM' \leftarrow |\phi^+\rangle^{MM'}$
        $r \xleftarrow{p_k} \{0,1\}^{\log|T|}$
        $C \leftarrow \mathsf{Enc}_{k;r}(M)$
        Add $(r, M')$ to $\textbf{List}$
        Output $C$
    $D_k(C)$
        $MT \leftarrow V_k^\dagger C V_k$
        Measure $\{0 : P^T_{\sigma_k}, 1 : \bar{P}^T_{\sigma_k}\}$
        If the outcome is 1, apply $C \leftarrow V_k(MT)V_k^\dagger$ and return $\mathsf{Dec}_k(C)$
        For each pair $(r, M')$ in $\textbf{List}$:
        1.    Measure $\{0 : |\psi_{k,r}\rangle\langle\psi_{k,r}|, 1 : \mathbb{I} - |\psi_{k,r}\rangle\langle\psi_{k,r}|\}$ on $T$
        2.    If the outcome is 0, measure $\{0 : \Pi^+, 1 : \mathbb{I} - \Pi^+\}$ on $MM'$
        3.    If the outcome is again 0, output $\mathsf{cheat}$
        Return $M$
    $\rho_\kappa \leftarrow \mathcal{A}^{\mathcal{P}', D_k}(\mathsf{pk})$
    If a query to $D_k$ ouputs $\mathsf{cheat}$ or $b = 0$ output $\mathsf{cheat}$
    Else output $\mathsf{win}$
```

Definition 5.30 (QPA2). *We say a quantum public-key encryption scheme* Enc *is* QPA2 *plaintext aware if for any quantum polynomial-time ciphertext creators $\mathcal{A}$, there exists a quantum polynomial-time plaintext extractor $\mathcal{A}^*$ such that for any quantum polynomial-time plaintext creator $\mathcal{P}$ and any quantum polynomial-time distinguishing algorithms $\mathcal{D}$, the advantage of $\mathcal{D}$ in distinguishing $G^{\mathsf{QPA2\text{-}Test}}_{\mathcal{D},\mathcal{P},\Pi,\mathcal{A}}$ and $G^{\mathsf{QPA2\text{-}Cheat}}_{\Pi,\mathcal{A}}$ is negligible as a function* *of the security parameter:*

$$\mathsf{Adv}_{\mathcal{D},\mathcal{A}} = \Pr_b\left[b = b' \mid b' \leftarrow G^{\mathsf{QPA2\text{-}Test}}_{\mathcal{D},\mathcal{P},\Pi,\mathcal{A}}(b)\right] - \Pr_b\left[\mathsf{cheat} \leftarrow G^{\mathsf{QPA2\text{-}Cheat}}_{\Pi,\mathcal{A}}(b)\right] \leq \mathrm{negl}(\kappa)$$

Note that the absence of the absolute value here is intentional. Indeed, the adversary can always query the plaintext creator once and immediately query the decryption oracle on this input to achieve a cheating probability of 1.

5.6.1.1 Separation between QPA1 and QPA2

The separation between QPA1 and QPA2 follows in a similar fashion as in the post-quantum case.

Theorem 5.31. *A PKQES that is* QPA1 *plaintext-aware and QIND-CPA secure is not necessarily* QPA2 *plaintext aware.*

Proof. Let $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a QPA1 plaintext-aware and QIND-CPA secure PKQES. We define $\Pi' = (\mathsf{KeyGen}', \mathsf{Enc}', \mathsf{Dec}')$ as follows:

- $\mathsf{KeyGen}' = \mathsf{KeyGen}$
- $\mathsf{Enc}'(M) = \mathsf{Enc}(M)^C \otimes |0\rangle\langle 0|^B$
- $\mathsf{Dec}'(CB) = \mathsf{Dec}(C)$ (B is traced out)

It is trivial to see that Π' is still QIND-CPA secure, as $\mathcal{D}$ simply gets an additional $|0\rangle\langle 0|^B$ as input but otherwise receives the same input. To see that Π' is still QPA1 plaintext-aware, consider an arbitrary adversary $\mathcal{B}$ for the QPA1 security game with Π'. We construct an adversary $\mathcal{A}$ for Π that does the exact same thing as $\mathcal{B}$, except traces out the B register before any decryption query. Since Π is QPA1, there exists a plaintext extractor $\mathcal{A}^*$ for $\mathcal{A}$. Let $\mathcal{B}^*(CB) = \mathcal{A}^*(C)$, then $\mathcal{B}^*$ is a plaintext extractor for $\mathcal{B}$. Since $\mathcal{B}$ was arbitrary, this means Π' is also QPA1.

We show, however, that Π' is not QPA2. Assume towards a contradiction that Π' is QPA2. Let $\mathcal{A}^*$ be the plaintext extractor for the $\mathcal{A}^{\mathcal{O}}$ adversary that does the following, with oracle $\mathcal{O} \in \{\mathcal{A}^*, \mathsf{Dec}_{\mathsf{sk}}, D_k\}$:

- Prepares $\left|0^{\log|M|}\right\rangle$ in M.
- Runs $CB \leftarrow \mathcal{P}(M)$
- Performs a Pauli-X gate on B, which transforms $|0\rangle\langle 0|$ into $|1\rangle\langle 1|$
- Runs $M' \leftarrow \mathcal{O}(CB)$
- Outputs M'

Note that because of the Pauli X gate, the state queried to the oracle is orthogonal to the one received from the plaintext creator. This means that D_k in the $G^{\mathsf{QPA2\text{-}Cheat}}_{\Pi',\mathcal{A}}(b)$ game measures 1 on step 1 of the for loop each time and thus outputs cheat only with negligible probability. This means

$$\Pr_b\left[\mathsf{cheat} \leftarrow G^{\mathsf{QPA2\text{-}Cheat}}_{\Pi',\mathcal{A}}(b)\right] - \frac{1}{2} \leq \mathrm{negl}(\kappa) \tag{5.1}$$

For $b \in \{0,1\}$, let $\mathcal{P}_b(\cdot) = \mathsf{Enc}_{\mathsf{pk}}(\left|b^{\log|M|}\right\rangle)$, i.e. $\mathcal{P}_0$ is the plaintext creator that outputs an encryption of the classical state of the string of all 0 bits, regardless of input, and likewise for $\mathcal{P}_1$. Let $\mathcal{D}_0$ be the distinguisher that measures its input state in the computational basis and outputs 0 if the measurement is the all 0 string and 1 otherwise and let $\mathcal{D}_1 := 1 - \mathcal{D}_0$, i.e. the complement. Note that for an approximately correct Π, in all but negligible cases the game $G^{\mathsf{QPA2\text{-}Test}}_{\mathcal{D}_0,\mathcal{P}_0,\Pi',\mathcal{A}}(0)$ outputs 0, as all that is done is encrypting and then decrypting an all 0 state, then measuring it in the computational basis. The same is true for the $G^{\mathsf{QPA2\text{-}Test}}_{\mathcal{D}_1,\mathcal{P}_1,\Pi',\mathcal{A}}(0)$ game. This means that, for $d \in \{0,1\}$,

$$1 - \Pr\left[G^{\mathsf{QPA2\text{-}Test}}_{\mathcal{D}_d,\mathcal{P}_d,\Pi',\mathcal{A}}(0) = 0\right] \leq \text{negl}(\kappa) \tag{5.2}$$

Lastly, we construct for $b \in \{0,1\}$ the QIND-CPA adversaries $\mathcal{A}_{\mathsf{IND},b} = (\mathcal{M}_{\mathsf{IND}}, \mathcal{D}_{\mathsf{IND},b})$ with the same M register as Π' and no E register. $\mathcal{M}_{\mathsf{IND}}$ simply prepares a $\left|1^{\log|M|}\right\rangle$ state. $\mathcal{D}_{\mathsf{IND},b}$ does the following, on input register CB:

- Run $M' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{IND}},\mathcal{A}^*}$, where $\mathcal{O}_{\mathsf{IND}}$ is the oracle that provides CB on its first query and nothing afterwards.

- Run $b' \leftarrow \mathcal{D}_b(M')$

- Output b'

Note that, by construction, running $\mathcal{D}_{\mathsf{IND},b}$ on the all 0 input is equivalent to running $\mathcal{D}_b(\mathcal{A}^{\mathcal{P}_0,\mathcal{A}^*})$ and similarly running $\mathcal{D}_{\mathsf{IND},b}$ on the all 1 input is equivalent to $\mathcal{D}_b(\mathcal{A}^{\mathcal{P}_1,\mathcal{A}^*})$. Since $\mathcal{D}_0$ and $\mathcal{D}_1$ are complementary, we have

$$\Pr\left[\mathcal{D}_{\mathsf{IND},1}\{\mathsf{Enc}_{\mathsf{pk}}(\rho^M)\} = 1\right] = 1 - \Pr\left[\mathcal{D}_{\mathsf{IND},0}\{\mathsf{Enc}_{\mathsf{pk}}(\rho^M)\} = 1\right].$$

This means there exists a $d \in \{0,1\}$ such that $\Pr\left[\mathcal{D}_{\mathsf{IND},d}\{\mathsf{Enc}_{\mathsf{pk}}(\rho^M)\} = 1\right] \geq \frac{1}{2}$. By QIND-CPA, we also have that, for the same d, $\Pr\left[\mathcal{D}_{\mathsf{IND},d}\{\mathsf{Enc}_{\mathsf{pk}}(|0\rangle\langle 0|^M)\} = 1\right] \geq \frac{1}{2} - \varepsilon$, where $\varepsilon \leq \text{negl}(\kappa)$. This means we have either $\Pr\left[\mathcal{D}_1(\mathcal{A}^{\mathcal{P}_1,\mathcal{A}^*}) = 1\right] \geq \frac{1}{2}$ (in the $d = 1$ case) or $\Pr\left[\mathcal{D}_0(\mathcal{A}^{\mathcal{P}_0,\mathcal{A}^*}) = 1\right] \geq \frac{1}{2} - \varepsilon$ (in the $d = 0$ case). I.e.

$$\frac{1}{2} - \Pr\left[G^{\mathsf{QPA2\text{-}Test}}_{\mathcal{D}_d,\mathcal{P}_d,\Pi',\mathcal{A}}(1) = 1\right] \leq \text{negl}(\kappa) \tag{5.3}$$

Using this, we can conclude

$$\begin{aligned}
\mathsf{Adv}_{\mathcal{D}_d,\mathcal{A}} &= \Pr_b\left[b = b' \mid b' \leftarrow G^{\mathsf{QPA2\text{-}Test}}_{\mathcal{D}_d,\mathcal{P}_d,\Pi',\mathcal{A}}(b)\right] - (\frac{1}{2} + \varepsilon_1) && \text{By (5.1)}\\
&= \frac{1}{2}\Pr\left[1 \leftarrow G^{\mathsf{QPA2\text{-}Test}}_{\mathcal{D}_d,\mathcal{P}_d,\Pi',\mathcal{A}}(1)\right] + (\frac{1}{2} - \varepsilon_2) - (\frac{1}{2} + \varepsilon_1) && \text{By (5.2)}\\
&\geq \frac{1}{2}(\frac{1}{2} - \varepsilon_3) + (\frac{1}{2} - \varepsilon_2) - (\frac{1}{2} + \varepsilon_1) && \text{By (5.3)}\\
&= \frac{1}{4} - (\frac{\varepsilon_3}{2} + \varepsilon_2 + \varepsilon_1) \nleq \text{negl}(\kappa),
\end{aligned}$$

where $\varepsilon_1, \varepsilon_2, \varepsilon_3 \leq \text{negl}(\kappa)$. Since $\mathcal{A}^*$ was arbitrary, this means Π' is not QPA2. □

5.6.1.2 Relation between QPA2 and QIND-CCA2

Like with QPA1, QPA2 can be used to boost the indistinguishability of a scheme from QIND-CPA to QIND-CCA2 in a sense, although some shortcomings arise.

```
Game $G^{\mathsf{QCCA2\text{-}Test}}_{\Pi,\mathcal{A}}(b)$
    $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa)$
    $MS \leftarrow \mathcal{A}_1^{\mathsf{Dec}_{\mathsf{sk}}}(\mathsf{pk})$
    If $b = 0$: $C \leftarrow \mathsf{Enc}_{\mathsf{pk}}(M)$
    If $b = 1$: $C \leftarrow \mathsf{Enc}_{\mathsf{pk}}(\tau^M)$
    $b' \leftarrow \mathcal{A}_2^{\mathsf{Dec}_{\mathsf{sk}}}(CS, \mathsf{pk})$
    Output $b'$
```

```
Game $G^{\mathsf{QCCA2\text{-}Cheat}}_{\Pi,\mathcal{A}}(b)$
    $k = (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa)$
    $MS \leftarrow \mathcal{A}_1^{\mathsf{Dec}_{\mathsf{sk}}}(\mathsf{pk})$
    $M'M'' \leftarrow |\phi^+\rangle^{M'M''}$
    $r \xleftarrow{p_k} \{0,1\}^{\log|T|}$
    $C \leftarrow \mathsf{Enc}_{k;r}(M)$
    Store $(r, M'')$
    $D_k(C)$
        $MT \leftarrow V_k^\dagger C V_k$
        Measure $\{0 : P^T_{\sigma_k}, 1 : \bar{P}^T_{\sigma_k}\}$
        If the outcome is 1, apply $C \leftarrow V_k(MT)V_k^\dagger$ and return $\mathsf{Dec}_{\mathsf{sk}}(C)$
        For the stored $(r, M'')$:
        1.    Measure $\{0 : |\psi_{k,r}\rangle\langle\psi_{k,r}|, 1 : \mathbb{I} - |\psi_{k,r}\rangle\langle\psi_{k,r}|\}$ on $T$
        2.    If the outcome is 0, measure $\{0 : \Pi^+, 1 : \mathbb{I} - \Pi^+\}$ on $MM''$
        3.    If the outcome is again 0, output cheat
        If no cheat was output, return $M$
    $b' \leftarrow \mathcal{A}_2^{D_k}(CS, \mathsf{pk})$
    If a query to $D_k$ ouputs cheat or $b = 0$ output cheat
    Else output win
```

Definition 5.32 (QIND-CCA2, [AGM18])**.** *A PKQES* Π *is* QIND-CCA2 *if, for all* QPT *adversaries* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*,*

$$\Pr_b\left[b' = b \mid b' \leftarrow G^{\mathsf{QCCA2\text{-}Test}}_{\Pi,\mathcal{A}}(b)\right] - \Pr_b\left[G^{\mathsf{QCCA2\text{-}Cheat}}_{\Pi,\mathcal{A}}(b) \to \mathsf{cheat}\right] \leq \mathrm{negl}(\kappa).$$

Like in the definition of QPA2, the omission of the absolute value in the definition of QIND-CCA2 is intentional. The key observation here is that when limiting $|\mathbf{List}| = 1$, the D_k oracles in the QPA2 and QIND-CCA2 cheat games are identical. For QPA2 adversaries who make exactly one query to $\mathcal{P}$, the cheat detection games are completely identical.

Unfortunately, here is where a shortcoming of our definition presents itself. The natural way to prove QCCA2 security is to use QPA2 to replace the decryption oracle with the plaintext extractor, invoke QIND-CPA security since the adversary is now operating without decryption oracles, and then replace the extractor with the decryption oracle again. However, with our counterfactual definition, this process incurs a penalty in the tightness equal to the cheating amount of the adversary, that is, the probability that D_k in $G^{\mathsf{QCCA2\text{-}Cheat}}_{\Pi,\mathcal{A}}$ outputs cheat.

Note that the probability that an adversary is caught cheating is completely controlled by the adversary themselves, as they control whether they query a state from the plaintext creator to the plaintext extractor. Thus, we conjecture that this result can be extended to all adversaries, by arguing that for any adversary there is an adversary that achieves the same QCCA2 advantage while only getting caught cheating with negligible probability. The proof of this we leave for future work.

Theorem 5.33. *Any PKQES* Π *that is* QIND-CPA *secure and* QPA2 *plaintext-aware is also* QIND-CCA2 *secure against adversaries* $\mathcal{A}$ *for which*

$$\Pr_b\left[G^{\mathsf{QCCA2\text{-}Cheat}}_{\Pi,\mathcal{A}}(b) \to \mathsf{cheat}\right] - \frac{1}{2} \leq \mathrm{negl}(\kappa).$$

Proof. The basic idea of boosting CPA to CCA2 security is to simulate the decryption queries with a plaintext extractor, thus allowing an adversary to achieve the same advantage with or without decryption oracles. To do this, we start with a PKQES Π that is QIND-CPA secure and QPA2 plaintext-aware.

Let $\mathcal{A}_{\mathsf{QCCA2}} = (\mathcal{A}_1, \mathcal{A}_2)$ be an arbitrary QIND-CCA2 adversary for Π. We construct the following QPA2 adverary $\mathcal{A}^{\mathcal{P},\mathcal{O}}_{\mathsf{QPA2}}$ for Π, with oracle access to $\mathcal{O} \in \{\mathsf{Dec_{sk}}, \mathcal{A}^*, D_k\}$:

```
$\mathcal{A}^{\mathcal{P},\mathcal{O}}_{\mathsf{QPA2}}(\mathsf{pk})$
    $MS \leftarrow \mathcal{A}_1^{\mathcal{O}}(\mathsf{pk})$
    $C \leftarrow \mathcal{P}(M)$
    $b' \leftarrow \mathcal{A}_2^{\mathcal{O}}(CS, \mathsf{pk})$
    Output $|b'\rangle$
```

Since Π is QPA2, there exists $\mathcal{A}^*$ that satisfies the QPA2 requirements for all plaintext creators $\mathcal{P}$ and distinguishers $\mathcal{D}$. In particular, define $\mathcal{P}_0(M) = \mathsf{Enc_{pk}}(M)$, $\mathcal{P}_1(\cdot) = \mathsf{Enc_{pk}}(\tau^M)$. Also let $\mathcal{D}_0(\rho)$ be the distinguisher that measures ρ in the computational basis and outputs 0 if the result is 0 and 1 otherwise, and let $\mathcal{D}_1(\rho) = 1 - \mathcal{D}_0(\rho)$, i.e. the bit flipped output of $\mathcal{D}_0$.

Note that $\mathcal{A}_{\mathsf{QPA2}}$ makes only one query to $\mathcal{P}$ and thus limits the size of **List** to 1. This means that $G^{\mathsf{QPA2\text{-}Cheat}}_{\Pi,\mathcal{A}_{\mathsf{QPA2}}}(b)$ and $G^{\mathsf{QCCA2\text{-}Cheat}}_{\Pi,\mathcal{A}_{\mathsf{QCCA2}}}(b)$ are almost identical for $b \in \{0,1\}$. The only difference is that $\mathcal{A}_{\mathsf{QPA2}}$ encapsulates the output bit b' in a quantum state, but since this output is disregarded this does not influence the output of the games. Thus we have:

$$P_{\mathsf{cheat}} := \Pr_b\left[G^{\mathsf{QPA2\text{-}Cheat}}_{\Pi,\mathcal{A}_{\mathsf{QPA2}}}(b) \to \mathsf{cheat}\right] = \Pr_b\left[G^{\mathsf{QCCA2\text{-}Cheat}}_{\Pi,\mathcal{A}_{\mathsf{QCCA2}}}(b) \to \mathsf{cheat}\right] \tag{5.4}$$

Furthermore, by construction, we have that running the $b = 0$ case of the QPA2-Test game with D_0 and $\mathcal{P}_d$ ($d \in \{0,1\}$) is equivalent to running the $b = d$ case of the QCCA2-Test game, with the only difference being that the output bit b' is directly output in the QCCA2 game and in the QPA2 game is wrapped in a classical state and then immediately measured, which does not influence the output. Note however that we are running the $b = 0$ case of QPA2, which means that when $d = 1$ the same game is run and the same b' is output, but in the QPA2 this means $b \neq b'$ whenever in the QCCA2 case there is $d = b'$. We can fix this discrepancy by running the QPA2-Test game with D_d instead of

D_0, in which case the game outputs 0 if and only if the QCCA2-Test game outputs $b' = d$. This gives us the following equivalence for any $b \in \{0, 1\}$:

$$P_b^{\text{test0}} := \Pr\left[0 \leftarrow G^{\text{QPA2-Test}}_{D_b,\mathcal{P}_b,\Pi,\mathcal{A}_{\text{QPA2}}}(0)\right] = \Pr\left[b' = b \mid b' \leftarrow G^{\text{QCCA2-Test}}_{\Pi,\mathcal{A}_{\text{QCCA2}}}(b)\right] \tag{5.5}$$

For the next part, consider that the $b = 1$ case of the QPA2-Test game is run without any oracle access to $\mathsf{Dec}_{\mathsf{sk}}$. Since $\mathcal{A}_{\text{QPA2}}$ only requests one state from $\mathcal{P}$, it is trivial to see that QIND-CPA security implies that $\mathcal{A}^{\mathcal{P}_0,\mathcal{A}^*}_{\text{QPA2}}$ and $\mathcal{A}^{\mathcal{P}_1,\mathcal{A}^*}_{\text{QPA2}}$ produce states that cannot be distinguished from each other by any QPT distinguisher with more than negligible probability. This means we have, for $d \in \{0, 1\}$:

$$\left\|\Pr\left[1 \leftarrow G^{\text{QPA2-Test}}_{D_d,\mathcal{P}_0,\Pi,\mathcal{A}_{\text{QPA2}}}(1)\right] - \Pr\left[1 \leftarrow G^{\text{QPA2-Test}}_{D_d,\mathcal{P}_1,\Pi,\mathcal{A}_{\text{QPA2}}}(1)\right]\right\| \leq \text{negl}(\kappa) \tag{5.6}$$

Next, we remember that D_0 and D_1 are eachothers complement, therefore we have for any $d \in \{0, 1\}$:

$$\Pr\left[1 \leftarrow G^{\text{QPA2-Test}}_{D_0,\mathcal{P}_d,\Pi,\mathcal{A}_{\text{QPA2}}}(1)\right] = 1 - \Pr\left[1 \leftarrow G^{\text{QPA2-Test}}_{D_1,\mathcal{P}_d,\Pi,\mathcal{A}_{\text{QPA2}}}(1)\right] \tag{5.7}$$

For $b \in \{0, 1\}$, let $P_b^{\text{test1}} := \Pr\left[1 \leftarrow G^{\text{QPA2-Test}}_{D_b,\mathcal{P}_b,\Pi,\mathcal{A}_{\text{QPA2}}}(1)\right]$. Then we have, combining equation 5.6 and 5.7:

$$\left\|P_1^{\text{test1}} - (1 - P_0^{\text{test1}})\right\| \leq \text{negl}(\kappa) \tag{5.8}$$

Furthermore, since Π is QPA2 plaintext-aware, we have for $b \in \{0, 1\}$:

$$\left\|\frac{1}{2}(P_b^{\text{test0}} + P_b^{\text{test1}}) - P_{\text{cheat}}\right\| \leq \text{negl}(\kappa) \tag{5.9}$$

Lastly, we combine all the results. For simplicity, we will write $x \approx y$ if $\|x - y\| \leq \text{negl}(\kappa)$, which defines an equivalence relation.

$$\begin{aligned}
P_{\text{cheat}} &\approx \frac{1}{2}P_1^{\text{test0}} + \frac{1}{2}P_1^{\text{test1}} && \text{By 5.9, } b = 1 \\
&\approx \frac{1}{2}P_1^{\text{test0}} + \frac{1}{2} - \frac{1}{2}P_0^{\text{test1}} && \text{By 5.8} \\
&\approx \frac{1}{2}P_1^{\text{test0}} + \frac{1}{2} - \left(P_{\text{cheat}} - \frac{1}{2}P_0^{\text{test0}}\right) && \text{By 5.9, } b = 0 \\
\Rightarrow P_{\text{cheat}} - \frac{1}{2} &\approx \frac{1}{2}P_1^{\text{test0}} + \frac{1}{2}P_0^{\text{test0}} - P_{\text{cheat}}
\end{aligned}$$

We can combine this with Equation 5.2 to say that the QIND-CCA2 advantage of $\mathcal{A}_{\text{QCCA2}}$ is equal to $\frac{1}{2}P_1^{\text{test0}} + \frac{1}{2}P_0^{\text{test0}} - P_{\text{cheat}}$, which is equal to $P_{\text{cheat}} - \frac{1}{2} + \varepsilon$ with $\varepsilon \leq \text{negl}(\kappa)$. Combining this with the assumption that $P_{\text{cheat}} - \frac{1}{2} \leq \text{negl}(\kappa)$, this implies that Π is QIND-CCA2 secure. □

5.7 Conclusion

In this chapter, we discussed various forms of plaintext-awareness in the post-quantum and quantum setting. Importantly, we showed for each variation how it can be satisfied and what the relationship is between the presented notion of plaintext-awareness and indistinguishability. Furthermore, we presented the relations between all the post-quantum notions, as well as the hierarchy of quantum notions.

However, some open questions remain in the field of plaintext-awareness in the settings we discussed. Firstly, it remains an open question whether OAEP is $\mathsf{pqPA2Q_{dec}}$, as discussed at the end of Section 5.5.1. Furthermore, the achievability of the $\mathsf{QPA2}$ notion we presented remains an open question as well. We believe that a hybrid construction will achieve the notion similar to how $\mathsf{QPA1}$ was achieved, however, we were not able to prove this.

CHAPTER 6

Non-malleability for Quantum Public-key Encryption

This chapter is based on the paper "Non-malleability for quantum public-key encryption" [MSW19], which I wrote together with Christian Schaffner and Christian Majenz.

Non-malleability is an important security property for public-key encryption (PKE). Its significance is due to the fundamental unachievability of integrity and authenticity guarantees in this setting, rendering it the strongest integrity-like property achievable using only PKE, without digital signatures. In this work, we generalize this notion to the setting of *quantum* public-key encryption. Overcoming the notorious "recording barrier" known from generalizing other integrity-like security notions to quantum encryption, we generalize one of the equivalent classical definitions, comparison-based non-malleability, and show how it can be fulfilled. In addition, we explore one-time non-malleability notions for symmetric-key encryption from the literature by defining plaintext and ciphertext variants and by characterizing their relation.

6.1 Introduction

The development of quantum information processing technology has accelerated recently, with many large public and private players investing heavily [Wal18]. A future where communication networks include at least some high-capacity quantum channels and fault-tolerant quantum computers seems therefore more and more likely. How will we secure communication over the resulting "quantum internet" [WEH18]? One approach is to rely on features inherent to quantum theory to get unconditional security, e.g. by using teleportation. Such methods are, however, a far cry from the classical standard internet cryptography in terms of efficiency, as they require interaction. A different and more efficient approach is to generalize modern private- and public-key cryptography to the quantum realm.

In this chapter, we focus on the notion of non-malleability, which captures the idea that an encrypted message cannot be altered by a third party in a structured manner. This form of security does not inherently prevent a third party from learning the message that was encrypted and in the world of classical (non-quantum) computers, these notions are considered separate. When considering quantum computers we will see that an inherent connection between these two notions exists, but it is still valuable to consider both.

Non-malleability, first introduced by Dolev, Dwork and Naor [DDN03], derives its importance in quantum cryptography from the fact that it is the strongest integrity-like notion that is achievable using public-key encryption only. By this, we mean that one cannot implement digital signatures or any other publicly-verifiable form of authentication on quantum states [AGM21]. The aim of this work is to generalize the notion of non-malleability to public-key encryption of quantum data.

A recent attack that exemplifies the relevance of the concept of non-malleability is the "efail"-attack on the PGP protocol for confidential and authenticated e-mail communication [Pod+18]. The attack demonstrates a flaw in the symmetric-key phase of the PGP protocol which allows a possible attacker to insert a text of her own choosing into an encrypted message, which in turn exploits the behaviour of the program used to receive the e-mail. This kind of attack, where an attacker is not directly able to learn the message yet is still able to modify it, is exactly what non-malleable encryption secures against.

The classical notion of non-malleability is based on the notion of related plaintexts. For a non-malleable encryption scheme, it should, roughly speaking, be hard for an adversary to transform an encryption of a message m into a *different* ciphertext that decrypts to a *related* message m'. Here, "related" just means that the adversary has some control over the transformation that is applied to the plaintext underlying the ciphertext he attacks. Generalizing this notion to the quantum case is complicated by the quantum no-cloning theorem: After a message has been encrypted and modified by the adversary and subsequently decrypted, it cannot be compared with the result anymore. In addition, it cannot be checked in a straightforward manner whether the adversary has indeed modified the ciphertext.

In this work, we overcome these obstacles. The key ideas are the following. In the classical security game, an adversary is first asked to submit a distribution from which a plaintext is sampled. In the quantum setting, any message sampling procedure can be implemented by first performing a unitary quantum computation, and then discarding the contents of an auxiliary register. Instead of discarding this register, we view it as an extra record that is created along with the message. This extra record is then used instead of the original plaintext for evaluating the quantum analogue of a relation. The test whether the adversary has indeed modified the ciphertext is performed by running the sampling- and encryption computations backwards on the attacked ciphertext. If the ciphertext was *not* modified, this returns the registers into their initial blank state, which can be detected.

We establish confidence in the new security notion by showing that it becomes equivalent to the classical notion when restricted to the post-quantum setting, i.e. to classical PKE schemes and classical plaintexts and ciphertexts. We also show how to satisfy the new security notion using a classical-quantum hybrid construction.

6.1.1 Related Work

Non-malleability has been studied extensively in the classical setting, see [BS99, PsV06] and references therein. In quantum cryptography, non-malleability has been, to our knowledge, the subject of only two earlier works [ABW09, AM17], which were only concerned with one-time security for symmetric-key encryption.

Quantum public-key encryption has been studied in [BJ15, Ala+16] with respect to confidentiality.

Problems due to quantum no-cloning and the destructive nature of quantum measurement similar to the ones we face in this work have arisen before in the literature. In particular, devising security notions for quantum encryption where the classical security definition requires copying and comparing plaintexts or ciphertexts [AGM18, AGM21], as well as in some quantum attack models for classical cryptography [BZ13a, BZ13b, Ala+20] requires tackling similar obstacles. Another important case where the generalization of classical techniques is complicated by the mentioned features of quantum theory is that of rewinding and reprogramming [Unr12, Wat18, Don+19].

6.1.2 Summary of Contributions

We propose a definition for public-key quantum non-malleability in a computational setting, by adapting the classical definition for comparison-based non-malleability found in [BS99], a real-vs-ideal definition. In the following, we describe informally what main challenges the generalization of the classical security experiments (the real and the ideal one) to the quantum setting poses, and how we resolved them.

In the first step in the classical security experiments, the adversary submits a probability distribution p over messages. In both experiments, a plaintext from this distribution is sampled, encrypted and sent to the adversary. The adversary now has the opportunity to manipulate (or "malleate") the ciphertext with the goal that the output decrypts to a *related* plaintext.[1] The relation according to which the plaintexts are related, is supplied by the adversary. Of course, there are examples of relations that allow for easy creation of a ciphertext that decrypts to a related plaintext, like e.g. the trivial relation where any plaintext is related to any other plaintext. To not credit an adversary with a break for fulfilling such a relation, her success in two experiments is compared: in the real world, the relation is evaluated on the initial and final plaintexts, but in the ideal world, it is evaluated on the final plaintext and a plaintext that is independently sampled from p.

Attempting a naive quantum generalization, we face two main challenges: How does the challenger ensure that the ciphertext he received from the adversary is actually modified? And how does he evaluate a relation on the input plaintext and the decrypted one? Both questions are complicated by the fact that quantum information cannot be copied. The first question has a rather elegant solution. Instead of asking the adversary to provide a distribution of messages, we ask her to provide a state preparation circuit, a strict generalization of the former. Such a state preparation circuit starts from a blank register and prepares a quantum state on the plaintext register and an auxiliary register. But quantum operations are reversible, which means that to test whether the plaintext has changed after encryption, attack and decryption,[2] we can run preparation backwards and measure whether we got back a blank register. If so, the ciphertext was not changed, and the candidate manipulated plaintext is discarded. If not, we run preparation forward again, recovering (the actually changed part of) the adversary's candidate malleation.

The second question is solved by exploiting the fact that in the quantum setting, any message-sampling procedure can be implemented by first performing a unitary quantum computation, and then discarding the contents of an auxiliary register. We can therefore ask the adversary to provide a state-sampling unitary and store the auxiliary register as a record indicating which plaintext has been created. After proceeding with the experiment as in the classical case and using the modification test as described above, instead of

[1] In the actual experiments, the adversary is allowed to transform the ciphertext into many attempted manipulated ciphertexts. In this informal description, we simplify as no significant additional technical challenges arise from the generalization.

[2] Here we have to undo encryption in a different way as decryption is an irreversible process in general, see Section 6.3 for details.

checking whether the original and the attacked plaintext are related, we can now, in the real-world case, check whether the attacked plaintext is related to the record. In the ideal world, the record is replaced with an independently created one.

Definition 6.1 (QCNM, informal)**.** *A scheme is quantum comparison-based non-malleable (*QCNM*) if no adversary can achieve a better than negligible advantage in distinguishing the real and ideal versions of the quantum comparison-based non-malleability experiment described above.*

We go on to show that QCNM is a consistent generalization of CNM.

Theorem 6.2 (6.10, informal)**.** *When restricted to the post-quantum setting,* QCNM *and* CNM *are equivalent.*

Finally, we show that a QCNM scheme can be constructed from a CiNM scheme. The quantum-classical hybrid construction, which was extensively studied in [AGM21] with respect to confidentiality and integrity, is obtained by encrypting every plaintext with a symmetric-key, one-time secure quantum encryption scheme and a fresh key, and then encrypting that key with a non-malleable classical public-key encryption scheme and appending it to the ciphertext.

Theorem 6.3 (6.13, informal)**.** *Using a* CNM *classical scheme and a* NM *quantum scheme it is possible to construct a* QCNM *scheme via quantum-classical hybrid encryption.*

6.2 Preliminaries

In this section, we will briefly introduce the classical definitions that this chapter is built on, as well as the existing notion of non-malleability for quantum symmetric-key communication. We will build upon the classical definitions of non-malleability [BS99] and the existing quantum definitions of non-malleability [ABW09, AM17].

Game 13: CNM-Real

```
Input  : $\Pi, \mathcal{A}, \kappa$
Output: $b \in \{0,1\}$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa)$
$(M, s) \leftarrow \mathcal{A}_1(\mathsf{pk})$
$x \leftarrow M$
$y \leftarrow \mathsf{Enc}_{\mathsf{pk}}(x)$
$(R, \mathbf{y}) \leftarrow \mathcal{A}_2(s, y)$
$\mathbf{x} \leftarrow \mathsf{Dec}_{\mathsf{sk}}(\mathbf{y})$
Output 1 iff $(y \notin \mathbf{y}) \wedge R(x, \mathbf{x})$
```

Game 14: CNM-Ideal

```
Input  : $\Pi, \mathcal{A}, \kappa$
Output: $b \in \{0,1\}$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa)$
$(M, s) \leftarrow \mathcal{A}_1(\mathsf{pk})$
$x, \tilde{x} \leftarrow M$
$\tilde{y} \leftarrow \mathsf{Enc}_{\mathsf{pk}}(\tilde{x})$
$(R, \tilde{\mathbf{y}}) \leftarrow \mathcal{A}_2(s, \tilde{y})$
$\tilde{\mathbf{x}} \leftarrow \mathsf{Dec}_{\mathsf{sk}}(\tilde{\mathbf{y}})$
Output 1 iff $(\tilde{y} \notin \tilde{\mathbf{y}}) \wedge R(x, \tilde{\mathbf{x}})$
```

Definition 6.4 (Definition 2 in [BS99] (CNM-CPA))**.** *A* PKES Π *is* comparison-based non-malleable for chosen-plaintext attacks (CNM) *if for any adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ it holds that*

$$\Pr\left[\mathsf{CNM\text{-}Real}(\Pi, \mathcal{A}, \kappa) = 1\right] - \Pr\left[\mathsf{CNM\text{-}Ideal}(\Pi, \mathcal{A}, \kappa) = 1\right] \leq \mathrm{negl}(\kappa),$$

if $\mathcal{A}$ is such that:

- *$\mathcal{A}_1$ and $\mathcal{A}_2$ are PPT*

- $\mathcal{A}_1$ *outputs a valid message space* M *which can be sampled by a PPT algorithm*
- $\mathcal{A}_2$ *outputs a relation* R *computable by a PPT algorithm*
- $\mathcal{A}_2$ *outputs a vector* $\mathbf{y}$ *such that* $\perp \notin \mathsf{Dec}_{\mathsf{sk}}(\mathbf{y})$

For comparison-based non-malleability, we consider adversaries that are split into two stages, where each stage is a probabilistic algorithm. The first stage takes as input the public key and produces a message distribution, which is (a description of) a probabilistic algorithm that produces a plaintext. The second stage takes as input one ciphertext of a plaintext produced by this algorithm and produces a vector of ciphertexts and a relation R. The goal of the adversary is to construct R in such a way that R holds between the original plaintext and the (element-wise) decryption of the produced ciphertext vector, but not between another plaintext which is sampled independently from the message distribution and the decryption of this same vector. If an adversary can achieve this relation to hold with non-negligible probability, then intuitively the adversary was able to structurally change an encrypted message, which would indicate that the scheme is malleable.

In the existing literature on non-malleability in the quantum setting, the approach taken is quite different from the notion described above. Here, the focus is put on unconditional one-time security notions of symmetric-key non-malleability and authentication. In this setting, a notion of non-malleability was first introduced in [ABW09], which defines non-malleability as a condition on the effective map of an arbitrary attack. The *effective map* of an attack $\Lambda_A^{CB\to C\hat{B}}$ is defined as $\tilde{\Lambda}_A^{MB\to M\hat{B}} = \mathop{\mathbb{E}}_{k\leftarrow \mathsf{KeyGen}(1^\kappa)} \mathsf{Dec}_k \circ \Lambda_A \circ \mathsf{Enc}_k$, and can be thought of as the average effect of an attack on the plaintext level.

The main idea of this definition is that a ciphertext cannot be meaningfully transformed into the ciphertext of another message, which means that the effective map of any attack is either identity, in case no transformation is applied or a map $\langle\rho\rangle$, that replaces the ciphertext by a fixed one. Note that this way of defining non-malleability can also be satisfied by a scheme which has the property that an attacker can transform a ciphertext into another ciphertext of the same message. In other words, the non-malleability is only enforced on the plaintext level, which means it is a form of *plaintext non-malleability.* The classical notions discussed in the previous section do not allow for attacks that map an encrypted message to a different encryption of the same message. This restriction means non-malleability is enforced on the ciphertext level and thus these classical notions define forms of *ciphertext non-malleability.*

This effective-map-based way of describing non-malleability was continued in [AM17], where an insufficiency of the previous definition was demonstrated and a new definition was given. Their definition is given in terms of the mutual information between the plaintext and the side information collected by the attacker, which allows them to use quantum-information-theoretic tools to analyze the definition. Their definition also considers more powerful adversaries and protects against a “plaintext-injection” attack, where the adversary can send any chosen plaintext to the receiver instead of the intended message. One of the main results in their paper is a characterization theorem, which we will consider as the definition instead.

Definition 6.5 (Theorem 4.4 in [AM17])**.** *A* SKQES $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is* ε-non-malleable (ε-NM) *if, for any attack* $\Lambda_A^{CB\to C\hat{B}}$*, its effective map*

$\tilde{\Lambda}_A^{MB\to M\hat{B}}$ *is such that*

$$\left\| \tilde{\Lambda}_A - \left(\mathrm{id}^M \otimes \Lambda_1^{B\to\hat{B}} + \frac{1}{|C|^2 - 1} \left(|C|^2 \langle \mathsf{Dec}_K(\tau^C) \rangle - \mathrm{id} \right)^M \otimes \Lambda_2^{B\to\hat{B}} \right) \right\|_\diamond \le \varepsilon$$

where

$$\Lambda_1 = \mathrm{Tr}_{CC'} \left[\phi^{+CC'} \Lambda_A(\phi^{+CC'} \otimes (\cdot)) \right] \qquad \textit{and}$$
$$\Lambda_2 = \mathrm{Tr}_{CC'} \left[(\mathbb{I}^{CC'} - \phi^{+CC'}) \Lambda_A(\phi^{+CC'} \otimes (\cdot)) \right].$$

A SKQES *is* non-malleable (NM) *if it is* ε-NM *for some* $\varepsilon \le \mathrm{negl}(\kappa)$.

In the symmetric-key setting, one can also consider the notion of authentication. A scheme satisfying this notion not only prevents an attacker from meaningfully transforming ciphertexts but any attempt to do so can also be detected by the receiving party. In [DNS12] a definition is given for this notion, which we adapt slightly to use the diamond norm instead of the trace norm.

Definition 6.6 (Definition 2.2 in [DNS12]). *A* SKQES Π *is* ε-DNS authenticating (ε-DNS) *if, for any attack* $\Lambda_A^{CB\to C\hat{B}}$*, its effective map* $\tilde{\Lambda}_A^{MB\to M\hat{B}}$ *is such that*

$$\left\| \tilde{\Lambda}_A - \left(\mathrm{id}^M \otimes \Lambda_{acc}^{B\to\hat{B}} + \langle |\bot\rangle \langle \bot| \rangle \otimes \Lambda_{rej}^{B\to\hat{B}} \right) \right\|_\diamond \le \varepsilon,$$

for some CPTNI *maps* $\Lambda_{acc}, \Lambda_{rej}$ *such that* $\Lambda_{acc} + \Lambda_{rej}$ *is* CPTP.

A SKQES $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is* DNS authenticating (DNS) *if it is* $\varepsilon -$ DNS *for some* $\varepsilon \le \mathrm{negl}(\kappa)$.

It is shown in [AM17] that a NM scheme can be modified to a scheme that is DNS authenticating by appending a tag to the encoded plaintext.

6.3 Non-Malleability for Quantum PKE

In this section, we will define a notion of many-time non-malleability for quantum public-key encryption, quantum comparison-based non-malleability (QCNM), as a quantum analogue of the classical notion of comparison-based non-malleability (CNM) introduced in [BS99]. We first analyze CNM with the goal of finding appropriate quantum analogues of each of its components.

The message distribution M in the CNM definition allows an adversary to select messages that she thinks might produce ciphertexts that can be modified in a structural way. This choice is given because the total plaintext space is exponentially large, thus if one picks a message completely at random and only a few of them can be modified into related ciphertexts, then the winning probability is negligible despite the scheme being insecure. In the quantum representation of this message space, we consider the following requirements:

1. As mentioned earlier, the quantum no-cloning theorem prevents copying the plaintext after sampling it for future reference. In order to check the relation in the last step of CNM, we require that two related states are produced, one of which will be kept by the challenger and the other encrypted and used by the adversary.

2. It must not be possible for the adversary to correlate herself with either of the produced messages. This is to prevent the adversary from influencing the second copy of the state later on. For example, consider the case where the adversary produces the state $\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$, where the first two qubits are the two copies of the message and the last is kept by the adversary. The adversary can then measure her qubit, collapsing the superposition and informing her in which (classical) state the second copy now is. This allows her to trivially construct a relation between her output and the second copy.

In order to satisfy requirement (1), we have chosen to represent M by a unitary U^{MRP} such that $U|0\rangle$ is a purification of the message distribution, where the message resides in M, the second (reference) state in R, and P is used for the purification[3]. This purification register allows the adversary to implement any QPT quantum channel on MR, with U being a Stinespring dilation of that channel. Note that this does not allow the adversary to give an auxiliary input to the message sampling algorithm, which is crucial in avoiding the trivial attack presented in requirement (2). If any input state can be passed to the message sampling algorithm, this can be used by the adversary to build an entangled state between their registers and the plaintext, and collapsing this state after encryption of the message would be a malleability attack. In this setting, the adversary is thus able to implement any message state sampling channel that starts with no input as a unitary U. The first part of the quantum adversary, $\mathcal{A}_1$, produces this unitary in the form of a (classical description of a) circuit along with some side information S to be passed on to the next stage. We denote this process by $(U, S) \leftarrow \mathcal{A}_1(\mathsf{pk})$.

For the QCNM definition we define two experiments, similar to the CNM definition. In the following paragraph, we describe how the different elements of the CNM experiment are instantiated in the quantum case. The appropriate quantum notion of a relation R on plaintexts is given by a POVM element $0 \le E^{MR} \le \mathbb{I}$. The two registers MR are considered to contain related states if an application of the measurement $\{E, \mathbb{I}-E\}$ returns the outcome corresponding to E. Of course, this POVM is provided by the adversary in the form of a circuit and must hence be efficient. The quantum analogue of the vector $\mathbf{y}$ is given by a collection of registers $\mathbf{C} = C_1 \ldots C_m$, where m is at most polynomial in n, the security parameter of the considered scheme, and each C_i satisfies $M_iT_i = C_i \cong C = MT$. The quantum analogue of the vector $\mathbf{x}$ is similarly given as $\mathbf{M} = M_1 \ldots M_m$. Observe that any PKQES can also be seen as a SKQES, with keys of the form $k = (pk, sk)$, which allows us to use Lemma 2.8[4]. For any PKQES $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ with security parameter κ, let $\{V_k \mid k = (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa)\}$, $t = \log |T|$, $\{\psi_{k,r} \mid k = (pk, sk) \leftarrow \mathsf{KeyGen}(1^\kappa), r \in \{0,1\}^t\}$ and $\{p_k \mid k = (pk, sk) \leftarrow \mathsf{KeyGen}(1^\kappa)\}$ be as in Lemma 2.8 in the QCNM experiments.

Lastly, we define the unitary U_{prep} combining the preparation of the message state and the encryption of its part in register M. This means a check similar to the $y \notin \mathbf{y}$ check in the CNM experiments can be implemented by sequentially undoing U_{prep} on all combinations C_iRP and then measuring whether the result is $|0\rangle\langle 0|$, which is only the case if C_i contained part of $U_{prep}|0\rangle$, which is the original ciphertext given to the adversary.

We are now ready to define the real and ideal experiments for quantum comparison-based non-malleability.

[3] A purification is a quantum register that is similar to the "garbage" register in reversible computation.

[4] This characterization could also be invoked with $k = \mathsf{pk}$, however, the resulting encryption unitary is then (likely) not efficiently implementable.

Game 15: QCNM-Real

```
   Input  : $\Pi, \mathcal{A}, \kappa$
   Output: $b \in \{0,1\}$
1  $k = (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa)$
2  $(U^{MRP}, S) \leftarrow \mathcal{A}_1(\mathsf{pk})$
3  $r \xleftarrow{p_k} \{0,1\}^t$
4  Construct $U_\psi^T$ such that $U_\psi^T |0\rangle^T = |\psi_{k,r}\rangle^T$
5  Construct $U_{prep}^{MTRP} = V_k^{MT}(U^{MRP} \otimes U_\psi^T)$
6  Prepare $U_{prep} |0\rangle \langle 0| U_{prep}^\dagger$ in $MTRP$
7  $(\mathbf{C}, E) \leftarrow \mathcal{A}_2(MT, S)$
8  for $i = 1, \ldots, |\mathbf{C}|$ do
9      Perform $U_{prep}^\dagger$ on $C_iRP$
10     Measure $\{|0\rangle \langle 0|, \mathbb{I} - |0\rangle \langle 0|\}$ on $C_iRP$ with outcome $b$
11     if $b = 0$ then
12         Output 0
13     Perform $U_{prep}$ on $C_iRP$
14 $\mathbf{M} \leftarrow \mathsf{Dec}_{\mathsf{sk}}(\mathbf{C})$
15 Measure $\{E, \mathbb{I} - E\}$ on $R\mathbf{M}$ with outcome $e$
16 Output $e$
```

Game 16: QCNM-Ideal

```
   Input  : $\Pi, \mathcal{A}, \kappa$
   Output: $b \in \{0,1\}$
1  Run lines 1-14 of Experiment QCNM-Real
15 Prepare $U |0\rangle \langle 0| U^\dagger$ in $\tilde{M}\tilde{R}\tilde{P}$
16 Measure $\{E, \mathbb{I} - E\}$ on $\tilde{R}\mathbf{M}$ with outcome $e$
17 Output $e$
```

A PKQES is now defined to be QCNM-secure, if no adversary can achieve a higher success probability in the experiment QCNM-Real than in QCNM-Ideal.

Definition 6.7. *A* PKQES Π *is* quantum comparison-based non-malleable (QCNM) *if for any QPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *it holds that*

$$\Pr[\mathsf{QCNM\text{-}Real}(\Pi, \mathcal{A}, \kappa) = 1] - \Pr[\mathsf{QCNM\text{-}Ideal}(\Pi, \mathcal{A}, \kappa) = 1] \leq \mathrm{negl}(\kappa),$$

if $\mathcal{A}$ *such that:*

- $\mathcal{A}_1$ *outputs a valid unitary* U *which can be implemented by a QPT algorithm,*
- $\mathcal{A}_2$ *outputs a POVM element* E *which can be implemented by a QPT algorithm,*
- $\mathcal{A}_2$ *outputs a vector of registers* $\mathbf{C}$ *such that* $\perp \notin \mathsf{Dec}_{\mathsf{sk}}(\mathbf{C})$.

6.3.1 Relation Between QCNM and CNM

In order to compare QCNM to CNM, we consider both definitions modified for quantum adversaries and encryption schemes that have classical input and output but can perform quantum computation. In the case that a quantum state is sent to such a post-quantum algorithm, it is first measured in the computational basis to obtain a classical input.

Game 17: $\mathsf{QCNM\text{-}Real}_{PQ}$

```
   Input  : $\Pi, \mathcal{A}, \kappa$
   Output: $b \in \{0,1\}$
1  $k = (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa)$
2  $(U^{MRP}, S) \leftarrow \mathcal{A}_1(\mathsf{pk})$
3  $r \xleftarrow{p_k} \{0,1\}^t$
4  Construct $U_\psi^T$ such that $U_\psi^T \left|0\right\rangle^T = \left|\psi_{k,r}\right\rangle^T$
5  Construct $U_{prep}^{MTRP} = V_k^{MT}(U^{MRP} \otimes U_\psi^T)$
6  Prepare $U_{prep} \left|0\right\rangle \left\langle 0\right| U_{prep}^\dagger$ in $MTRP$
7  Measure $MTR$ in the computational basis with outcome $y^{MT} z^R$
8  $(\mathbf{C}, E) \leftarrow \mathcal{A}_2(MT, S)$
9  for $i = 1, \ldots, |\mathbf{C}|$ do
10     Measure $\{\left|y\right\rangle\left\langle y\right|, \mathbb{I} - \left|y\right\rangle\left\langle y\right|\}$ on $C_i$ with outcome $b$
11     if $b = y$ then
12         Output 0
13 $\mathbf{M} \leftarrow \mathsf{Dec}_{sk}(\mathbf{C})$
14 Measure $\{E, \mathbb{I} - E\}$ on $R\mathbf{M}$ with outcome $e$
15 Output $e$
```

Game 18: $\mathsf{QCNM\text{-}Ideal}_{PQ}$

```
   Input  : $\Pi, \mathcal{A}, \kappa$
   Output: $b \in \{0,1\}$
1  Run lines 1-13 of Experiment $\mathsf{QCNM\text{-}Ideal}_{PQ}$
14 Prepare $U \left|0\right\rangle \left\langle 0\right| U^\dagger$ in $\tilde{M}\tilde{R}\tilde{P}$
15 Measure $\tilde{M}\tilde{R}$ in the computational basis
16 Measure $\{E, \mathbb{I} - E\}$ on $\tilde{R}\mathbf{M}$ with outcome $e$
17 Output $e$
```

We consider the above experiments to be the post-quantum version of the QCNM experiments. The main modification is the measurement in Step 7, which enforces the requirement that $\mathcal{A}_2$ only takes classical input. The modification of Steps 9 through 12 is made because the measurement in Step 7 disturbs the state in an irreversible fashion, thus performing $U_{prep}^\dagger$ no longer inverts the sampling/encryption process. Lastly, in the Ideal setting, Step 15 is added to mimic the effect that Step 7 would have on M'.

Definition 6.8. *A* PKQES Π *is* post-quantum comparison-based non-malleable (QCNM_{PQ}) *if for any adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *it holds that*

$$\Pr\left[\mathsf{QCNM\text{-}Real}_{PQ}(\Pi, \mathcal{A}, \kappa) = 1\right] - \Pr\left[\mathsf{QCNM\text{-}Ideal}_{PQ}(\Pi, \mathcal{A}, \kappa) = 1\right] \leq \mathrm{negl}(\kappa),$$

if $\mathcal{A}$ *and* Π *are such that:*

- $\mathcal{A}_1$ *and* $\mathcal{A}_2$ *are QPT and output only classical states,*
- $\mathcal{A}_1$ *outputs a valid unitary* U *which can be implemented by a QPT algorithm,*
- $\mathcal{A}_2$ *outputs a POVM element* E *which implementable by a QPT algorithm, and*

- *$\mathcal{A}_2$ outputs a vector of registers $\mathbf{C}$ such that $\bot \notin \mathsf{Dec}_{\mathsf{sk}}(\mathbf{C})$.*

Sampling of the message by the challenger is now done by not only applying U to $|0\rangle$, but in addition also measuring in the computational basis. Similarly, we define a post-quantum version of CNM.

Definition 6.9. *A PKQES Π is* comparison-based non-malleable against post-quantum adversaries (CNM_{PQ}) *if for any QPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ it holds that*

$$\Pr\left[\mathsf{CNM\text{-}Real}(\Pi, \mathcal{A}, \kappa) = 1\right] - \Pr\left[\mathsf{CNM\text{-}Ideal}(\Pi, \mathcal{A}, \kappa) = 1\right] \leq \mathrm{negl}(\kappa),$$

if Π and $\mathcal{A}$ are such that:

- *$\mathcal{A}_1$ and $\mathcal{A}_2$ are QPT and output classical strings,*
- *$\mathcal{A}_1$ outputs a valid QPT algorithm M which produces classical strings,*
- *$\mathcal{A}_2$ outputs a QPT algorithm R,*
- *$\mathcal{A}_2$ outputs a vector $\mathbf{y}$ such that $\bot \notin \mathsf{Dec}_{\mathsf{sk}}(\mathbf{y})$.*

The only difference between CNM and CNM_{PQ} is that the latter allows the encryption scheme, adversary and any algorithms produced by the adversary to use a quantum computer. Furthermore, the relation R has become probabilistic, but since it is used only once there is no difference between using a probabilistic relation or picking a deterministic relation at random. Observe that CNM_{PQ} is simply a stronger requirement than CNM since it requires security against a strict superset of adversaries, and thus trivially implies CNM.

Theorem 6.10. *A PKQES Π is QCNM_{PQ} if and only if Π is CNM_{PQ}.*

Proof. For the $\Rightarrow$ direction, let Π be an arbitrary PKQES which is QCNM_{PQ}-secure and let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an arbitrary quantum adversary intended to perform the CNM_{PQ} experiments. Assume that Π is such that Enc and Dec take only classical input and produce only classical output. Define $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ as follows:
$\mathcal{B}_1(\mathsf{pk})$:

```
$(M, s) \leftarrow \mathcal{A}_1(\mathsf{pk})$
Let $p_M(x)$ be the probability that $x \leftarrow M$, then construct $U$ such that
  $U|0\rangle^{MM'P} = \frac{1}{|R|}\sum_{r \in R}|M(r)M(r)r\rangle = \sum_{x \leftarrow M}\sqrt{p_M(x)}\,|xx\phi_x\rangle^{MM'P}$, where
  $M' \cong M$ is the reference register, $R$ is the set of possible input for $M$ and $\phi_x$ is
  the uniform superposition over all $|r\rangle$ such that $x \leftarrow M(r)$.
Output $(U, |s\rangle\langle s|)$
```

$\mathcal{B}_2(|s\rangle\langle s|^{S}, |y\rangle\langle y|^{MT})$:

```
$(R, \mathbf{y}) \leftarrow \mathcal{A}_2(y, s)$
Construct $E = \sum_{i,\mathbf{j}} R(i, \mathbf{j})\,|i\mathbf{j}\rangle\langle i\mathbf{j}|$
Output $(E, |\mathbf{y}\rangle\langle\mathbf{y}|^{C_1 \ldots C_m})$
```

Observe that the definition of $\mathsf{QCNM\text{-}Real}_{PQ}(\Pi, \mathcal{B}, \kappa)$, after some simplification, yields

```
1  $k = (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa)$
2  $(M, s) \leftarrow \mathcal{A}_1(\mathsf{pk})$
3  Let $p_M(x)$ be the probability that $x \leftarrow M$, then construct $U$ such that
   $U|0\rangle^{MM'P} = \sum_{x \leftarrow M} \sqrt{p_M(x)}\,|xx\phi_x\rangle^{MM'P}$
4  $r \xleftarrow{p_k} \{0,1\}^t$
5  Construct $U_\psi^T$ such that $U_\psi^T|0\rangle^T = |\psi_{k,r}\rangle^T$
6  Construct $U_{prep}^{MTM'P} = V_k^{MT}(U^{MM'P} \otimes U_\psi^T)$
7  Prepare $U_{prep}|0\rangle\langle 0|U_{prep}^\dagger$ in $MTM'P$
8  Measure $MTM'$ in the computational basis with outcome $yz$
9  $(R, \mathbf{y}) \leftarrow \mathcal{A}_2(y, s)$
10 Construct $E = \sum_{i,\mathbf{j}} R(i,\mathbf{j})\,|i\mathbf{j}\rangle\langle i\mathbf{j}|$
11 Prepare $|\mathbf{y}\rangle\langle\mathbf{y}|$ in $\mathbf{C}$
12 for $i = 1, \ldots, |\mathbf{C}|$ do
13     Measure $\{|y\rangle\langle y|, \mathbb{I} - |y\rangle\langle y|\}$ on $C_i$ with outcome $b$
14     if $b = y$ then
15         Output 0
16 $\mathbf{M} \leftarrow \mathsf{Dec}_{\mathsf{sk}}(\mathbf{C})$
17 Measure $\{E, \mathbb{I} - E\}$ on $M'\mathbf{M}$ with outcome $e$
18 Output $e$
```

Here Steps 3,5,6,7 and 8 together simply execute $x \leftarrow M; y \leftarrow \mathsf{Enc}_{k;r}(x)$. Furthermore, if $y \in \mathbf{y}$ then some C_i contains $|y\rangle\langle y|$, which will guarantee the output to be y in Step 13. Conversely if $y \notin \mathbf{y}$, then all C_i contain some state orthogonal to $|y\rangle\langle y|$ and thus Step 13 has 0 probability of outputting y in this case, thus Step 13 effectively implements the $y \notin \mathbf{y}$ check. Lastly, note that E is a projective measurement which projects onto the space spanned by all $|i\mathbf{j}\rangle$ such that $R(i,\mathbf{j})$, which means that Step 17 outputs 1 iff $R(x,\mathbf{x})$, where x is stored in M' and $\mathbf{x}$ in $\mathbf{M}$. We conclude that $\mathsf{QCNM\text{-}Real}_{PQ}(\Pi, \mathcal{B}, \kappa)$ produces the same random variable as $\mathsf{CNM\text{-}Real}(\Pi, \mathcal{A}, \kappa)$. By similar reasoning the same is true for the Ideal case, with the additional observation that preparing $U|0\rangle$ in $\tilde{M}\tilde{M}'\tilde{P}$ and measuring $\tilde{M}$ in the computational basis with result $\tilde{x}$ is equivalent to $\tilde{x} \leftarrow M$ and collapses $\tilde{M}'$ to $\tilde{x}$. It follows that Π is CNM_{PQ}.

For the $\Leftarrow$ direction, let Π be an arbitrary PKQES fulfilling CNM_{PQ} and let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an arbitrary classical adversary on this scheme intended to perform the QCNM_{PQ} experiments. Define $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ as follows:
$\mathcal{B}_1(\mathsf{pk})$:

```
1  $(U, |s\rangle\langle s|^S) \leftarrow \mathcal{A}_1(\mathsf{pk})$
2  Prepare $U|0\rangle$ twice, in $M_0R_0P_0$ and $M_1R_1P_1$
3  Measure $M_0R_0M_1R_1$ in the computational basis with outcome $m_0z_0m_1z_1$
4  Construct $M$ to be the uniform distribution over $\{m_0, m_1\}$
5  Output $(M, sm_0z_0m_1z_1)$
```

$\mathcal{B}_2(s', y)$:

```
Parse $s'$ as $sm_0z_0m_1z_1$
$(E, \mathbf{y}) \leftarrow \mathcal{A}_2(|s\rangle\langle s|^S, |y\rangle\langle y|^{MT})$
Construct $R(x, \mathbf{x})$ to be
    Find $i$ such that $m_i = x$
    prepare $|z_i\mathbf{x}\rangle\langle z_i\mathbf{x}|$ in $R'\mathbf{M}$
    measure $\{E, \mathbb{I} - E\}$ on $R'\mathbf{M}$, output 1 iff the outcome is $E$
Output $(R, |\mathbf{y}\rangle\langle\mathbf{y}|)$
```

Observe that the definition of $\mathsf{CNM\text{-}Real}_{PQ}(\Pi, \mathcal{B}, \kappa)$, after some simplification, yields

```
1  $k = (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa)$
2  $(U, |s\rangle\langle s|^S) \leftarrow \mathcal{A}_1(\mathsf{pk})$
3  Prepare $U|0\rangle$ in $M_0R_0P_0$
4  Prepare $U|0\rangle$ in $M_1R_1P_1$
5  Measure $M_0R_0M_1R_1$ in the computational basis with outcome $m_0z_0m_1z_1$
6  Pick $i \leftarrow \{0, 1\}$
7  $y \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m_i)$
8  $(E, |\mathbf{y}\rangle\langle\mathbf{y}|) \leftarrow \mathcal{A}_2(|s\rangle\langle s|^S, |y\rangle\langle y|^{MT})$
9  $\mathbf{x} \leftarrow \mathsf{Dec}_{\mathsf{sk}}(\mathbf{y})$
10 if $y \in \mathbf{y}$ then
11     Output 0
12 Find $j$ such that $m_j = m_i$
13 Prepare $|z_j\mathbf{x}\rangle\langle z_j\mathbf{x}|$ in $R\mathbf{M}$
14 Measure $\{E, \mathbb{I} - E\}$ on $R\mathbf{M}$ with outcome $e$
15 Output $e$
```

Similarly, the $\mathsf{CNM\text{-}Ideal}_{PQ}(\Pi, \mathcal{B}, \kappa)$ yields the same Experiment except with line 12 replaced with "Pick $j \leftarrow \{0, 1\}$". Note that Step 7, the encrypting, is not performed by U_{prep} but simply by Enc and that Step 10 simply checks $y \in \mathbf{y}$ instead of the loop that we earlier argued to be equivalent. Additionally, the measurement in Steps 3 and 4 are equivalent to measuring the ciphertext after encryption (as is done in QCNM), because it is assumed that encryption, and thus V_k, maps classical states to classical states.

Note that w.l.o.g. we can assume that $m_0 \neq m_1$, since if this is not the case then the Real and Ideal cases are equivalent and thus the adversary has no hope of winning. This makes that the $\mathsf{CNM\text{-}Real}_{PQ}(\Pi, \mathcal{B}, \kappa)$ and $\mathsf{QCNM\text{-}Real}_{PQ}(\Pi, \mathcal{A}, \kappa)$ are equivalent given the observations in the previous paragraph. Furthermore, when $i = j$ in the $\mathsf{CNM\text{-}Ideal}$ case then it is equivalent to the $\mathsf{CNM\text{-}Real}$ case. When $i \neq j$, the $\mathsf{CNM\text{-}Ideal}_{PQ}(\Pi, \mathcal{B}, \kappa)$ and $\mathsf{QCNM\text{-}Ideal}_{PQ}(\Pi, \mathcal{A}, n)$ experiments are equivalent. Thus the advantage of $\mathcal{B}$ in CNM is half the advantage of $\mathcal{A}$ in QCNM, which implies that Π is QCNM_{PQ}. □

Note that we argued earlier that, for any PKES, being CNM_{PQ} trivially implies being CNM, thus we derive the following corollary.

Corollary 6.11. *Any* QCNM_{PQ} PKES *is* CNM.

6.3.2 A QCNM Secure Scheme

In this section we show how QCNM-security can be achieved using a quantum-classical hybrid construction like the ones used in [AGM18, AGM21], that we presented in Construction 2.7. We continue by proving that if Π^{Qu} is unitary and secure according to

NM, and Π^{Cl} to be CNM, then $\Pi^{\mathsf{Hyb}}[\Pi^{\mathsf{Qu}}, \Pi^{\mathsf{Cl}}]$ is QCNM.To prove QCNM security of the classical-quantum hybrid scheme, we need the following lemma.

Lemma 6.12. *Let $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a* SKQES*, let $\ell \in \mathbb{N}$, let $\mathbf{C} = C_1 \dots C_\ell \cong C^\ell$ and $\mathbf{M} = M_1 \dots M_\ell \cong M^\ell$ be vectors of registers, let $\Lambda^{C\to\mathbf{C}}$ be a* CPTP *map, and set*

$$\tilde{\Lambda}^{M\to\mathbf{M}} = \mathop{\mathbb{E}}_{k\leftarrow\mathsf{KeyGen}(1^n)}\left[(\mathsf{Dec}_k)^{\otimes\ell} \circ \Lambda \circ \mathsf{Enc}_k\right].$$

If SKQES *is* CiNM *secure, then for some p_0 and $\{\sigma_i\}_i$ we have that*

$$\tilde{\Lambda}^{M\to\mathbf{M}} = \sum_{i=1}^{\ell} p_i \,\mathrm{id}^{M\to M_i} \otimes \sigma_i^{\mathbf{M}_{-i}} + p_0\langle\sigma_0^{\mathbf{M}}\rangle,$$

where q_i is the probability that is equal to Λ_1 from Definition 6.5 applied to the attack map $\mathrm{Tr}_{\mathbf{C}_{-i}} \circ\Lambda$ and $\mathbf{M}_{-i} = M_1 \dots M_{i-1}M_{i+1} \dots M_\ell$ and $p_i = q_i - \frac{1}{|C|^2-1}(1-q_i)$.

Proof. For fixed i, consider the attack $\Lambda' = \mathrm{Tr}_{\mathbf{C}_{-i}} \circ\Lambda$ on Π and observe that its effective map satisfies $\tilde{\Lambda}' = \mathbb{E}_k\left[\mathsf{Dec}_k \circ \mathrm{Tr}_{\mathbf{C}_{-i}} \circ\Lambda \circ \mathsf{Enc}_k\right] = \mathrm{Tr}_{\mathbf{M}_{-i}} \circ\tilde{\Lambda}$. Because Π is NM, we have $\tilde{\Lambda}' = p_i \,\mathrm{id} + (1-p_i)\langle\mathsf{Dec}_K(\tau)\rangle$ and thus

$$\mathrm{Tr}_{\mathbf{M}_{-i}} \circ\tilde{\Lambda}(\phi^{+MM'}) = p_i\phi^{+M_iM'} + (1-p_i)\mathsf{Dec}_K(\tau) \otimes \tau^{M'}. \tag{6.1}$$

Consider the state $\tilde{\Lambda}(\phi^{+MM'})$. Because $\phi^{+M_iM'}$ is a pure state, we know that $\tilde{\Lambda}(\phi^{+MM'})$ is a convex combination of terms of the form $p_i\phi^{+M_iM'} \otimes \sigma_{\mathbf{M}_{-i}}$ and a term $p_0\sigma_0^{\mathbf{M}} \otimes \tau^{M'}$, i.e.

$$\tilde{\Lambda}(\phi^{+MM'}) = \sum_{i=1}^{\ell} p_i\phi^{+M_iM'} \otimes \sigma_i^{\mathbf{M}_{-i}} + p_0\sigma_0^{\mathbf{M}} \otimes \tau^{M'}. \tag{6.2}$$

By the Choi-Jamiolkowski isomorphism[Jam72, Cho75] this means that

$$\tilde{\Lambda}^{M\to\mathbf{M}} = \sum_{i=1}^{\ell} p_i \,\mathrm{id}^{M\to M_i} \otimes \sigma_i^{\mathbf{M}_{-i}} + p_0\langle\sigma_0^{\mathbf{M}}\rangle.$$

Using Equation (6.1), we get in addition that all single-system marginals of σ_i, $i = 0, ..., \ell$ are equal to $\mathsf{Dec}_K(\tau)$. □

Note that a similar statement can be proven for attack maps $\Lambda^{MB\to\mathbf{M}\tilde{B}}$ with side information, but we only need the above statement in Theorem 6.13.

Using this lemma, we can show that the hybrid scheme is indeed QCNM-secure.

Theorem 6.13. *Let $\Pi^{\mathsf{Qu}} = (\mathsf{KeyGen}^{\mathsf{Qu}}, \mathsf{Enc}^{\mathsf{Qu}}, \mathsf{Dec}^{\mathsf{Qu}})$ be a* NM *secure* SKQES *with unitary encryption and decryption map, and $\Pi^{\mathsf{Cl}} = (\mathsf{KeyGen}^{\mathsf{Cl}}, \mathsf{Enc}^{\mathsf{Cl}}, \mathsf{Dec}^{\mathsf{Cl}})$ a CNM_{PQ} secure* PKES*. Then $\Pi^{\mathsf{Hyb}}[\Pi^{\mathsf{Qu}}, \Pi^{\mathsf{Cl}}]$ is* QCNM*.*

Proof. We begin by defining modified versions of the two experiments used in defining QCNM, sckQCNM-Real and sckQCNM-Ideal (for **s**poofed **c**lassical **k**ey). These two experiments are defined exactly as the experiments QCNM-Real and QCNM-Ideal, except for the following modifications:

1. When creating the ciphertext register C that is handed to the adversary, its classical part c is produced by encrypting a fresh, independently sampled one-time key $k' \leftarrow \mathsf{KeyGen}^{\mathsf{Qu}}$. The pair (c, k) is stored (k being the key used for encryption with $\mathsf{Enc}^{\mathsf{Qu}}$.)

2. The test whether the ciphertext was modified by the adversary is done by first checking whether the classical part c' is equal to c. If it is not, the ciphertext was modified and no further test of the quantum part is necessary. If $c' = c$, the modification check from the games QCNM-Real and QCNM-Ideal is applied, using the stored one-time key k. Note that this is equivalent to the check mandated for the QCNM experiments.

3. Before decrypting any ciphertext, the challenger checks whether its classical part is equal to c. If not, he proceeds with decryption, otherwise, he just decrypts the quantum ciphertext with $\mathsf{Dec}_k^{\mathsf{Qu}}$.

Let $\mathcal{A}$ be a QCNM-adversary against Π^{Hyb}. Recall that it was proven in [BS99] that CNM is equivalent to IND-parCCA2, indistinguishability under parallel chosen ciphertext attacks. In this attack model, after receiving the challenge ciphertext, the adversary is allowed to submit one tuple of ciphertexts that are decrypted in case none of them is equal to the challenge ciphertext. Define the following IND-parCCA2 adversary $\mathcal{A}'$ against Π^{Cl}. $\mathcal{A}'$ simulates the $\mathsf{QCNM\text{-}Real}(\Pi^{\mathsf{Hyb}}, \mathcal{A}, \kappa)$-experiment. When the QCNM-Real challenger is supposed to encrypt a plaintext to be sent to $\mathcal{A}$, $\mathcal{A}'$ sends $m_0 = k$ and $m_1 = k'$ as challenge plaintexts to the IND-parCCA2 challenger, where $k, k' \leftarrow \mathsf{KeyGen}^{\mathsf{Qu}}$, and k is used to encrypt the quantum plaintext. After storing a copy of the resulting classical ciphertext c and the one-time key k, $\mathcal{A}'$ continues to simulate $\mathsf{QCNM\text{-}Real}(\Pi^{\mathsf{Hyb}}, \mathcal{A}, \kappa)$ but using the mixed quantum-classical modification check from the spoofed classical key experiments defined above. Decryption is done using the parCCA2 oracle, except for the ciphertexts with classical part c, which are just decrypted using the stored one-time key k. Now $\mathcal{A}'$ outputs the result of the simulated experiment $\mathsf{QCNM\text{-}Real}(\Pi^{\mathsf{Hyb}}, \mathcal{A}, \kappa)$.

Now observe that if the IND-parCCA2 challenger's bit comes up $b = 0$, $\mathcal{A}'$ faithfully simulated the experiment $\mathsf{QCNM\text{-}Real}(\Pi^{\mathsf{Hyb}}, \mathcal{A}, \kappa)$, while the case $b = 1$ results in a simulation of $\mathsf{sckQCNM\text{-}Real}(\Pi^{\mathsf{Hyb}}, \mathcal{A}, \kappa)$. Therefore, the IND-parCCA2 security of Π^{Cl} implies that the games QCNM-Real and sckQCNM-Real have the same result, up to negligible difference.

We can also define an IND-parCCA2 adversary $\mathcal{A}''$ against Π^{Cl} in the same way as $\mathcal{A}'$, but this time using the QCNM-Ideal experiments. This implies analogously that the experiments QCNM-Ideal and sckQCNM-Ideal also have the same result, up to negligible difference.

Finally, what is left to prove is that the experiments sckQCNM-Real and sckQCNM-Ideal have the same outcome due to the NM security of Π^{Qu}. If the classical part of the ciphertext has been modified, Dec_k is never applied. By the fact that the scheme Π^{Qu} is IND secure [AM17], $\mathbf{M}$ is independent of (i.e. in a product state with) R, i.e. $\mathbf{M}R$ and $\mathbf{M}\tilde{R}$ have the same state. Therefore, sckQCNM-Real and sckQCNM-Ideal have the same outcome. For the remaining case of $c' = c$, note that the modification test in lines 8 through 13 of Experiments 15 and 16 are identical, and that the application of $\left(V^M\right)^\dagger$ (T is trivial for unitary encryption) is equal to decryption. We can hence decrypt all ciphertexts before the modification test in the experiments sckQCNM-Real and sckQCNM-Ideal (line 9 in experiments QCNM-Real and QCNM-Ideal), and replace U_{prep} by U. It follows that the rest of the experiment after decryption does not depend on the one-time key k anymore. Hence the experiment has the form of a multi-decryption attack on the scheme Π^{Qu}, i.e.

where one ciphertext (the one that $\mathcal{A}_2$ receives as input) is mapped to many ciphertexts (the ones in $\mathbf{C}$) and are subsequently decrypted. We can therefore apply Lemma 6.12 to conclude that the modification test outputs 0 unless $\mathbf{M}$ is in product with R, in which case $\mathbf{M}R$ and $\mathbf{M}\tilde{R}$ have the same state. sckQCNM-Real and sckQCNM-Ideal, therefore, have the same outcome. □

6.4 Conclusion

After providing the first definition of non-malleability for quantum public-key encryption and showing how to fulfil it, and providing a comprehensive taxonomy of one-time security notions in the symmetric-key case, our work leaves a number of interesting open questions.

While our proposed definition of QCNM provides a natural extension of CNM to the quantum setting, a number of alternative but equivalent definitions of classical non-malleability exist, such as simulation-based non-malleability as defined in [BS99]. Besides the natural question of whether QCNM truly captures non-malleability, one might want to consider quantum versions of other classical notions of non-malleability and the relations between them. Furthermore, a symmetric-key version of QCNM could be explored, which we suspect to be distinct from computational versions of existing symmetric-key notions [MSW19] due to the mismatch of the way side information is handled.

CHAPTER 7

Conclusion

In this thesis, we were able to shed light on some of the problems all of us face in the world of emerging quantum technologies, and the opportunities we have ahead of ourselves. Overall, this thesis brought together various primitives in the quantum and post-quantum worlds and highlighted their similarities when tackling their implementation. We highlighted a number of open questions for each of the topics. Furthermore, the natural open question is which other primitives can be brought to the quantum setting using similar techniques as used in this thesis, and what could their implementation in the quantum setting imply for the overall security landscape.

In the sections below, we highlight the biggest results of the three concepts we tackled.

7.1 Deniability

To build towards deniable QKE, we first discussed a more complete form of sender-privacy that enhances the use of strong designated verifier signature schemes in both the classical and quantum settings. Here, the strongest results were those boosting existing 2-party sender-privacy to the n-party setting through Theorem 3.24 and Theorem 3.23. This shows that our definition of sender-privacy, which was designed to be more easily applicable in constructing proofs while encompassing more edge cases, is implied by a combination of common properties.

Theorem 7.1 (Theorem 3.24)**.** *Any* DVS *scheme* Π *that is* 2*-party sender-private, strongly unforgeable, and computationally non-transferable is* n*-party sender-private for any* $n \geq 2$*.*

Theorem 7.2 (Theorem 3.23)**.** *Let* $n \in \mathbb{N}$ *and* $\mathcal{O} = \{\mathcal{O}^{(1)}_{sign}, \mathcal{O}^{(2)}_{sign}, \mathcal{O}^{(1)}_{sim}, \mathcal{O}^{(2)}_{sim}, \mathcal{O}^{(1)}_{veri}, \mathcal{O}^{(2)}_{veri}\}$ *be the* n*-sender standard oracles. Any* DVS *scheme that is* n*-party sender-private with respect to* $\mathcal{O}' = \{\mathcal{O}^{(1)}_{sign}, \mathcal{O}^{(2)}_{sign}, \mathcal{O}^{(1)}_{sim}, \mathcal{O}^{(2)}_{sim}, \mathcal{O}'^{(1)}_{veri} = \emptyset, \mathcal{O}'^{(2)}_{veri} = \emptyset\}$ *and strongly unforgeable is* n*-party sender-private (with respect to* $\mathcal{O}$*).*

We were directly able to showcase the usefulness of these enhancements by using our form of sender-privacy as a building block for participation-deniable quantum key exchange. Moreover, the scheme we construct is an instantiation of the (slightly modified) BB84 scheme, which is one of the most well-studied QKE schemes.

Theorem 7.3 (Theorem 4.7)**.** AuthBB *with authentication scheme* Π_{SDVS} *is deniable if* Π_{SDVS} *is an* SDVS *with non-transferability against quantum adversaries.*

Corollary 7.4 (Corollary 4.8). *Under Assumptions 4.2 and 4.3,* SUSDVS *(from [NJ16]) is an* SDVS *with non-transferability against quantum adversaries, thus* AuthBB *using* SUSDVS *is deniable in the standard model.*

7.2 Plaintext-Awareness

We went on to study the concept of plaintext-awareness, bringing it to the post-quantum setting. Here, we consider the various challenges that occur in a setting where recording queries for future use is not possible in the classical sense. We defined various notions of PA in this setting and showed their satisfiability, as well as numerous relations between them. A key result here is the proof of the expected relation between IND-qCPA, $\mathsf{pqPA2}$ and IND-qCCA.

Theorem 7.5 (Theorem 5.24). *Any public-key encryption scheme* Enc *that is* $\mathsf{pqPA2}\text{-}\mathsf{Q}_{\mathsf{dec}}$ *plaintext-aware and IND-qCPA secure is IND-qCCA secure.*

After this, we move to the setting of communication between quantum computers, and explore the same questions in this setting, obtaining novel notions of plaintext-awareness in this context that mirrors the classical setting. Here we again show the expected relations between our presented notions, as well as the way in which $\mathsf{QPA2}$ can be used to achieve $\mathsf{QIND\text{-}CCA2}$ security.

Theorem 7.6 (Theorem 5.33). *Any PKQES* Π *that is* $\mathsf{QIND\text{-}CPA}$ *secure and* $\mathsf{QPA2}$ *plaintext-aware is also* $\mathsf{QIND\text{-}CCA2}$ *secure against adversaries* $\mathcal{A}$ *for which*

$$\Pr_b\left[G^{\mathsf{QCCA2\text{-}Cheat}}_{\Pi,\mathcal{A}}(b) \to \mathsf{cheat}\right] - \frac{1}{2} \leq \mathrm{negl}(\kappa).$$

7.3 Non-Malleability

Lastly, we use our newly-acquired familiarity with the recording barrier in the quantum setting to tackle non-malleability for public-key communication between quantum computers. This provides us with an integrity primitive in a setting where digital signatures are not possible. We show how our notion relates to the classical equivalents. The key result we present in this context is the satisfiability of the newly constructed QCNM notion.

Theorem 7.7 (Theorem 6.13). *Let* $\Pi^{\mathsf{Qu}} = (\mathsf{KeyGen}^{\mathsf{Qu}}, \mathsf{Enc}^{\mathsf{Qu}}, \mathsf{Dec}^{\mathsf{Qu}})$ *be a* NM *secure* SKQES *with unitary encryption and decryption map, and* $\Pi^{\mathsf{Cl}} = (\mathsf{KeyGen}^{\mathsf{Cl}}, \mathsf{Enc}^{\mathsf{Cl}}, \mathsf{Dec}^{\mathsf{Cl}})$ *a* CNM_{PQ} *secure* PKES*. Then* $\Pi^{\mathsf{Hyb}}[\Pi^{\mathsf{Qu}}, \Pi^{\mathsf{Cl}}]$ *is* QCNM*.*

Jeroen van Wier is supported by the Luxembourg National Research Fund (FNR), under the joint CORE project Q-CoDe (CORE17/IS/11689058/Q-CoDe/Ryan).

Bibliography

[ABW09] Andris Ambainis, Jan Bouda, and Andreas Winter. "Nonmalleable encryption of quantum information". In: *Journal of Mathematical Physics* 50.4 (2009), p. 042106. DOI: 10.1063/1.3094756.

[AGM18] Gorjan Alagic, Tommaso Gagliardoni, and Christian Majenz. "Unforgeable Quantum Encryption". In: *Advances in Cryptology — EUROCRYPT 2018*. Cham: Springer International Publishing, 2018, pp. 489–519. ISBN: 978-3-319-78372-7. DOI: 10.1007/978-3-319-78372-7_16.

[AGM21] Gorjan Alagic, Tommaso Gagliardoni, and Christian Majenz. "Can you sign a quantum state?" In: *Quantum* 5 (2021), p. 603. ISSN: 2521-327X. DOI: 10.22331/q-2021-12-16-603.

[Ala+16] Gorjan Alagic et al. "Computational Security of Quantum Encryption". In: *Information Theoretic Security*. Ed. by Anderson C.A. Nascimento and Paulo Barreto. Cham: Springer International Publishing, 2016, pp. 47–71. ISBN: 978-3-319-49175-2. DOI: 10.1007/978-3-319-49175-2_3.

[Ala+20] Gorjan Alagic et al. "Quantum-Access-Secure Message Authentication via Blind-Unforgeability". In: *Advances in Cryptology – EUROCRYPT 2020*. Ed. by Anne Canteaut and Yuval Ishai. Cham: Springer International Publishing, 2020, pp. 788–817. ISBN: 978-3-030-45727-3. DOI: 10.1007/978-3-030-45727-3_27.

[AM17] Gorjan Alagic and Christian Majenz. "Quantum Non-malleability and Authentication". In: *Advances in Cryptology – CRYPTO 2017*. Ed. by Jonathan Katz and Hovav Shacham. Cham: Springer International Publishing, 2017, pp. 310–341. ISBN: 978-3-319-63715-0. DOI: 10.1007/978-3-319-63715-0_11.

[And+14] Elena Andreeva et al. "How to Securely Release Unverified Plaintext in Authenticated Encryption". In: *Advances in Cryptology — ASIACRYPT 2014*. Vol. 8873. Springer, 2014, pp. 105–125. DOI: 10.1007/978-3-662-45611-8_6.

[Ata+18] Arash Atashpendar et al. "Revisiting Deniability in Quantum Key Exchange". In: *Secure IT Systems, 23rd Nordic Conference, NordSec 2018*. Cham: Springer International Publishing, 2018, pp. 104–120. ISBN: 978-3-030-03638-6. DOI: 10.1007/978-3-030-03638-6_7.

[Ata19] Arash Atashpendar. "From Information Theory Puzzles in Deletion Channels to Deniability in Quantum Cryptography". PhD thesis. University of Luxembourg, Luxembourg, 2019. URL: https://arxiv.org/pdf/2003.11663.pdf.

[BB84] Charles H Bennett and Gilles Brassard. "Quantum cryptography: public key distribution and coin tossing". In: *Conf. on Computers, Systems and Signal Processing (India, Dec. 1984)*. 1984, pp. 175–9. DOI: 10.1016/j.tcs.2014.05.025.

[BD14] James Birkett and Alexander W. Dent. "Security Models and Proof Strategies for Plaintext-Aware Encryption". In: *J. Cryptol.* 27.1 (2014), pp. 139–180. DOI: 10.1007/s00145-012-9141-6.

[Bea02] Donald Beaver. "On Deniability in Quantum Key Exchange". In: *Advances in Cryptology — EUROCRYPT 2002*. Springer Berlin Heidelberg, 2002, pp. 352–367. ISBN: 978-3-540-46035-0. DOI: 10.1007/3-540-46035-7_23.

[Bel+01] Mihir Bellare et al. "Key-Privacy in Public-Key Encryption". In: *Advances in Cryptology — ASIACRYPT 2001*. Ed. by Colin Boyd. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 566–582. ISBN: 978-3-540-45682-7. DOI: 10.1007/3-540-45682-1_33.

[Bel+98] Mihir Bellare et al. "Relations Among Notions of Security for Public-Key Encryption Schemes". In: *Advances in Cryptology — CRYPTO 1998*. Vol. 1462. Springer, 1998, pp. 26–45. DOI: 10.1007/BFb0055718.

[BJ15] Anne Broadbent and Stacey Jeffery. "Quantum Homomorphic Encryption for Circuits of Low T-gate Complexity". In: *Advances in Cryptology – CRYPTO 2015*. Ed. by Rosario Gennaro and Matthew Robshaw. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 609–629. ISBN: 978-3-662-48000-7. DOI: 10.1007/978-3-662-48000-7_30.

[BP04] Mihir Bellare and Adriana Palacio. "Towards Plaintext-Aware Public-Key Encryption Without Random Oracles". In: *Advances in Cryptology — ASIACRYPT 2004*. Vol. 3329. Lecture Notes in Computer Science. Springer, 2004, pp. 48–62. DOI: 10.1007/978-3-540-30539-2_4.

[BR95] Mihir Bellare and Phillip Rogaway. "Optimal asymmetric encryption". In: *Advances in Cryptology — EUROCRYPT'94*. Ed. by Alfredo De Santis. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 92–111. ISBN: 978-3-540-44717-7. DOI: 10.1007/BFb0053428.

[BS99] Mihir Bellare and Amit Sahai. "Non-malleable Encryption: Equivalence between Two Notions, and an Indistinguishability-Based Characterization". In: *Advances in Cryptology – CRYPTO '99*. Ed. by Michael Wiener. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 519–536. ISBN: 978-3-540-48405-9. DOI: 10.1007/3-540-48405-1_33.

[BZ13a] Dan Boneh and Mark Zhandry. "Quantum-Secure Message Authentication Codes". In: *Advances in Cryptology – EUROCRYPT 2013*. Ed. by Thomas Johansson and Phong Q. Nguyen. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 592–608. ISBN: 978-3-642-38348-9. DOI: 10.1007/978-3-642-38348-9_35.

[BZ13b] Dan Boneh and Mark Zhandry. "Secure Signatures and Chosen Ciphertext Security in a Quantum Computing World". In: *Advances in Cryptology – CRYPTO 2013*. Ed. by Ran Canetti and Juan A. Garay. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 361–379. ISBN: 978-3-642-40084-1. DOI: 10.1007/978-3-642-40084-1_21.

[CA90] David Chaum and Hans van Antwerpen. “Undeniable Signatures”. In: *Advances in Cryptology — CRYPTO’ 89 Proceedings*. Ed. by Gilles Brassard. New York, NY: Springer New York, 1990, pp. 212–216. ISBN: 978-0-387-34805-6. DOI: 10.1007/0-387-34805-0_20.

[Can+97] Ran Canetti et al. “Deniable Encryption”. In: *Advances in Cryptology — CRYPTO ’97*. Springer Berlin Heidelberg, 1997, pp. 90–104. ISBN: 978-3-540-69528-8. DOI: 10.1007/BFb0052229.

[Car+21] Tore Vincent Carstens et al. “Relationships Between Quantum IND-CPA Notions”. In: *TCC 2021,* vol. 13042. Springer, 2021, pp. 240–272. DOI: 10.1007/978-3-030-90459-3_9.

[CEV22] Céline Chevalier, Ehsan Ebrahimi, and Quoc-Huy Vu. “On Security Notions for Encryption in a Quantum World”. In: *Progress in Cryptology – INDOCRYPT 2022*. Ed. by Takanori Isobe and Santanu Sarkar. Cham: Springer International Publishing, 2022, pp. 592–613. ISBN: 978-3-031-22912-1. DOI: 10.1007/978-3-031-22912-1_26.

[CG96] R. Canetti and R. Gennaro. “Incoercible multiparty computation”. In: *Proceedings of 37th Conference on Foundations of Computer Science*. Oct. 1996, pp. 504–513. DOI: 10.1109/SFCS.1996.548509.

[Cha96] David Chaum. *Private signature and proof systems*. US Patent No. 5,493,614. Feb. 1996.

[Che+21] Yu-Ao Chen et al. “An integrated space-to-ground quantum communication network over 4,600 kilometres”. In: *Nature* 589.7841 (2021), pp. 214–219. DOI: 10.1038/s41586-020-03093-8.

[Cho75] Man-Duen Choi. “Completely positive linear maps on complex matrices”. In: *Linear algebra and its applications* 10.3 (1975), pp. 285–290. DOI: 10.1016/0024-3795(75)90075-0.

[CS96] A Robert Calderbank and Peter W Shor. “Good quantum error-correcting codes exist”. In: *Physical Review A* 54.2 (1996), p. 1098. DOI: 10.1103/PhysRevA.54.1098.

[DDN03] D. Dolev, C. Dwork, and M. Naor. “Nonmalleable Cryptography”. In: *SIAM Review* 45.4 (2003), pp. 727–784. DOI: 10.1137/S0036144503429856.

[Den06] Alexander W. Dent. “The Cramer-Shoup Encryption Scheme Is Plaintext Aware in the Standard Model”. In: *Advances in Cryptology — EUROCRYPT 2006*. Vol. 4004. Springer, 2006, pp. 289–307. DOI: 10.1007/11761679_18.

[DNS04] Cynthia Dwork, Moni Naor, and Amit Sahai. “Concurrent Zero-knowledge”. In: *J. ACM* 51.6 (Nov. 2004), pp. 851–898. ISSN: 0004-5411. DOI: 10.1145/1039488.1039489.

[DNS12] Frédéric Dupuis, Jesper Buus Nielsen, and Louis Salvail. “Actively Secure Two-Party Evaluation of Any Quantum Operation”. In: *Advances in Cryptology – CRYPTO 2012*. Ed. by Reihaneh Safavi-Naini and Ran Canetti. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 794–811. ISBN: 978-3-642-32009-5. DOI: 10.1007/978-3-642-32009-5_46.

[Don+19] Jelle Don et al. “Security of the Fiat-Shamir Transformation in the Quantum Random-Oracle Model”. In: *Advances in Cryptology – CRYPTO 2019*. Ed. by Alexandra Boldyreva and Daniele Micciancio. Cham: Springer International Publishing, 2019, pp. 356–383. ISBN: 978-3-030-26951-7. DOI: 10.1007/978-3-030-26951-7_13.

[DRGK06] Mario Di Raimondo, Rosario Gennaro, and Hugo Krawczyk. “Deniable Authentication and Key Exchange”. In: *Proceedings of the 13th ACM Conference on Computer and Communications Security*. CCS ’06. Alexandria, Virginia, USA: ACM, 2006, pp. 400–409. ISBN: 1-59593-518-5. DOI: 10.1145/1180405.1180454.

[Ebr22] Ehsan Ebrahimi. “Post-quantum Security of Plain OAEP Transform”. In: *PKC 2022*. Vol. 13177. Springer, 2022, pp. 34–51. DOI: 10.1007/978-3-030-97121-2_2.

[Eur] *European Quantum Communication Infrastructure (EuroQCI) | Shaping Europe’s digital future*. https://digital-strategy.ec.europa.eu/en/policies/european-quantum-communication-infrastructure-euroqci. (Accessed on 09/07/2021).

[EW22] Ehsan Ebrahimi and Jeroen van Wier. “Post-quantum Plaintext-Awareness”. In: *Post-Quantum Cryptography*. Ed. by Jung Hee Cheon and Thomas Johansson. Cham: Springer International Publishing, 2022, pp. 260–285. ISBN: 978-3-031-17234-2. DOI: 10.1007/978-3-031-17234-2_13.

[GHS16] Tommaso Gagliardoni, Andreas Hülsing, and Christian Schaffner. “Semantic Security and Indistinguishability in the Quantum World”. In: *Advances in Cryptology — CRYPTO 2016*. Vol. 9816. Springer, 2016, pp. 60–89. DOI: 10.1007/978-3-662-53015-3_3.

[GKS21] Tommaso Gagliardoni, Juliane Krämer, and Patrick Struck. “Quantum Indistinguishability for Public Key Encryption”. In: *PQCrypto 2021*. Vol. 12841. Springer, 2021, pp. 463–482. DOI: 10.1007/978-3-030-81293-5_24.

[HLM03] Jonathan Herzog, Moses D. Liskov, and Silvio Micali. “Plaintext Awareness via Key Registration”. In: *Advances in Cryptology — CRYPTO 2003*. Ed. by Dan Boneh. Vol. 2729. Springer, 2003, pp. 548–564. DOI: 10.1007/978-3-540-45146-4_32.

[Hua+06] Xinyi Huang et al. “Short (Identity-Based) Strong Designated Verifier Signature Schemes”. In: *Information Security Practice and Experience*. Ed. by Kefei Chen et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 214–225. ISBN: 978-3-540-33058-5. DOI: 10.1007/11689522_20.

[Hua+11] Qiong Huang et al. “Identity-based strong designated verifier signature revisited”. In: *Journal of Systems and Software* 84.1 (2011). Information Networking and Software Services, pp. 120–129. ISSN: 0164-1212. DOI: 10.1016/j.jss.2010.08.057.

[IM11] Lawrence M. Ioannou and Michele Mosca. “A New Spin on Quantum Cryptography: Avoiding Trapdoors and Embracing Public Keys”. In: *Post-Quantum Cryptography*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 255–274. ISBN: 978-3-642-25405-5. DOI: 10.1007/978-3-642-25405-5_17.

[Jai+12] Abhishek Jain et al. “Commitments and Efficient Zero-Knowledge Proofs from Learning Parity with Noise”. In: *Advances in Cryptology — ASIACRYPT 2012*. Vol. 7658. Lecture Notes in Computer Science. Springer, 2012, pp. 663–680. DOI: 10.1007/978-3-642-34961-4_40.

[Jam72] Andrzej Jamiołkowski. “Linear transformations which preserve trace and positive semidefiniteness of operators”. In: *Reports on Mathematical Physics* 3.4 (1972), pp. 275–278. DOI: 10.1016/0034-4877(72)90011-0.

[JSI96] Markus Jakobsson, Kazue Sako, and Russell Impagliazzo. “Designated Verifier Proofs and Their Applications”. In: *Advances in Cryptology — EUROCRYPT ’96*. Ed. by Ueli Maurer. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 143–154. ISBN: 978-3-540-68339-1. DOI: 10.1007/3-540-68339-9_13.

[Kas+02] Elham Kashefi et al. “Comparison of quantum oracles”. In: *Phys. Rev. A* 65 (5 2002), p. 050304. DOI: 10.1103/PhysRevA.65.050304.

[Kaw+05] Akinori Kawachi et al. “Computational Indistinguishability Between Quantum States and Its Cryptographic Application”. In: *Advances in Cryptology — EUROCRYPT 2005*. Vol. 3494. Springer, 2005, pp. 268–284. DOI: 10.1007/11426639_16.

[LV05] Fabien Laguillaumie and Damien Vergnaud. “Designated Verifier Signatures: Anonymity and Efficient Construction from Any Bilinear Map”. In: *Security in Communication Networks*. Ed. by Carlo Blundo and Stelvio Cimato. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 105–119. ISBN: 978-3-540-30598-9. DOI: 10.1007/978-3-540-30598-9_8.

[McE78] Robert J McEliece. “A public-key cryptosystem based on algebraic coding theory”. In: *The Deep Space Network Progress Report* 42-44 (1978), pp. 114–116. URL: https://tmo.jpl.nasa.gov/progress_report2/42-44/44N.PDF.

[MSU13] Michele Mosca, Douglas Stebila, and Berkant Ustaoğlu. “Quantum Key Distribution in the Classical Authenticated Key Exchange Framework”. In: *Post-Quantum Cryptography*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 136–154. ISBN: 978-3-642-38616-9. DOI: 10.1007/978-3-642-38616-9_9.

[MSW19] Christian Majenz, Christian Schaffner, and Jeroen van Wier. *Non-Malleability for Quantum Public-Key Encryption*. 2019. arXiv: 1905.05490.

[NJ16] Geontae Noh and Ik Rae Jeong. “Strong designated verifier signature scheme from lattices in the standard model”. In: *Security and Communication Networks* 9.18 (2016), pp. 6202–6214. DOI: 10.1002/sec.1766.

[Pod+18] Damian Poddebniak et al. “Efail: Breaking S/MIME and OpenPGP Email Encryption using Exfiltration Channels”. In: *27th USENIX Security Symposium (USENIX Security 18)*. Baltimore, MD: USENIX Association, 2018, pp. 549–566. ISBN: 978-1-931971-46-1. URL: https://www.usenix.org/conference/usenixsecurity18/presentation/poddebniak.

[PsV06] Rafael Pass, abhi shelat, and Vinod Vaikuntanathan. “Construction of a Non-malleable Encryption Scheme from Any Semantically Secure One”. In: *Advances in Cryptology – CRYPTO 2006*. Ed. by Cynthia Dwork. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 271–289. ISBN: 978-3-540-37433-6. DOI: 10.1007/11818175_16.

[Reg05] Oded Regev. “On lattices, learning with errors, random linear codes, and cryptography”. In: *ACM Symposium on Theory of Computing, 2005*. ACM, 2005, pp. 84–93. DOI: 10.1145/1060590.1060603.

[RST01] Ronald L. Rivest, Adi Shamir, and Yael Tauman. “How to Leak a Secret”. In: *Advances in Cryptology — ASIACRYPT 2001*. Ed. by Colin Boyd. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 552–565. ISBN: 978-3-540-45682-7. DOI: 10.1007/3-540-45682-1_32.

[Sho94] Peter W Shor. “Algorithms for quantum computation: discrete logarithms and factoring”. In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. Nov. 1994, pp. 124–134. DOI: 10.1109/SFCS.1994.365700.

[Sho97] Peter W. Shor. “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer”. In: *SIAM J. Comput.* 26.5 (1997), pp. 1484–1509. DOI: 10.1137/S0097539795293172.

[Sim97] Daniel R. Simon. “On the Power of Quantum Computation”. In: *SIAM J. Comput.* 26.5 (1997), pp. 1474–1483. DOI: 10.1137/S0097539796298637.

[SKM04] Shahrokh Saeednia, Steve Kremer, and Olivier Markowitch. “An Efficient Strong Designated Verifier Signature Scheme”. In: *Information Security and Cryptology - ICISC 2003*. Ed. by Jong-In Lim and Dong-Hoon Lee. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 40–54. ISBN: 978-3-540-24691-6. DOI: 10.1007/978-3-540-24691-6_4.

[SN17] National Institute of Standards and Technology (NIST). *Post-Quantum Cryptography Standardization*. https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization. [Online; accessed 22-July-2019]. 2017.

[SP00] Peter W Shor and John Preskill. “Simple proof of security of the BB84 quantum key distribution protocol”. In: *Physical review letters* 85.2 (2000), p. 441. DOI: 10.1103/PhysRevLett.85.441.

[Ste96] Andrew Steane. “Multiple-particle interference and quantum error correction”. In: *Proc. R. Soc. Lond. A* 452.1954 (1996), pp. 2551–2577. DOI: 10.1098/rspa.1996.0136.

[STW12] X. Sun, H. Tian, and Y. Wang. “Toward Quantum-Resistant Strong Designated Verifier Signature from Isogenies”. In: *2012 Fourth International Conference on Intelligent Networking and Collaborative Systems*. Sept. 2012, pp. 292–296. DOI: 10.1109/iNCoS.2012.70.

[SZM04] Willy Susilo, Fangguo Zhang, and Yi Mu. “Identity-Based Strong Designated Verifier Signature Schemes”. In: *Information Security and Privacy*. Ed. by Huaxiong Wang, Josef Pieprzyk, and Vijay Varadharajan. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 313–324. ISBN: 978-3-540-27800-9. DOI: 10.1007/978-3-540-27800-9_27.

[Unr12] Dominique Unruh. “Quantum Proofs of Knowledge”. In: *Advances in Cryptology – EUROCRYPT 2012*. Ed. by David Pointcheval and Thomas Johansson. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 135–152. ISBN: 978-3-642-29011-4. DOI: 10.1007/978-3-642-29011-4_10.

[Wal18] Nicholas Wallace. *"EU runs to catch up as governments pledge more cash for quantum computing"*. Ed. by Richard L. Hudson. Nov. 8, 2018. URL: https://web.archive.org/web/20181108205334/https://sciencebusiness.net/news/eu-runs-catch-governments-pledge-more-cash-quantum-computing.

[WAR] Jeroen van Wier, Arash Atashpendar, and Peter Roenne. *Deniable Public-Key Authenticated Quantum Key Exchange*. URL: http://hdl.handle.net/10993/51736.

[Wat18] John Watrous. *The Theory of Quantum Information*. Cambridge University Press, 2018. URL: https://cs.uwaterloo.ca/~watrous/TQI/TQI.pdf.

[WEH18] Stephanie Wehner, David Elkouss, and Ronald Hanson. "Quantum internet: A vision for the road ahead". In: *Science* 362.6412 (2018). ISSN: 0036-8075. DOI: 10.1126/science.aam9288.

[Wie21] Jeroen van Wier. *On SDVS Sender Privacy In The Multi-Party Setting*. 2021. arXiv: 2107.06119.

[Zha19] Mark Zhandry. "How to Record Quantum Queries, and Applications to Quantum Indifferentiability". In: *Advances in Cryptology — CRYPTO 2019*. Vol. 11693. Springer, 2019, pp. 239–268. DOI: 10.1007/978-3-030-26951-7_9.