Facing and Exploiting the Quantum Wave in Computing: New Security Definitions and Cryptographic Constructions

Quoc-Huy Vu

Université Paris - Panthéon - Assas

Ecole doctorale d'Économie, Gestion, Information et Communication

Thèse de doctorat en Informatique

Préparée à l'École normale supérieure

Soutenue le 01 Février 2023

Thèse de doctorat Février/ 2023

Facing and Exploiting the Quantum Wave in Computing: New Security Definitions and Cryptographic Constructions

Quoc-Huy VU

Sous la direction de Céline Chevalier

Membres du jury :

Céline Chevalier Université Panthéon-Assas	*Directeur de thèse*
Gorjan Alagic University of Maryland	*Rapporteur*
Prabhanjan Ananth University of California, Santa Barbara	*Rapporteur*
Anne Broadbent University of Ottawa	*Examinateur*
Elham Kashefi Sorbonne Université	*Examinateur*
Hieu Phan Télécom Paris	*Examinateur*
Olivier Blazy École Polytechnique	*Examinateur*
David Pointcheval École Normale Supérieure	*Examinateur*

Avertissement

L'université n'entend donner aucune approbation ni improbation aux opinions émises dans cette thèse ; ces opinions doivent être considérées comme propres à leur auteur.

Résumé

La cryptographie moderne a un ennemi de taille à l'horizon : la montée inévitable des ordinateurs quantiques. Cependant, cette même puissance de calcul permettrait également de trouver des solutions sur des tâches cryptographiques qui sont tout simplement impossibles à réaliser avec la technologie actuelle. Dans cette thèse, nous mettons les pieds dans un univers où le quantique est omniprésent en y présentant notamment deux principales contributions.

Nous mettons en avant à la fois des nouveaux modèles et de nouvelles analyses de sécurité pour deux primitives cryptographiques : les chiffrements et les preuves à divulgation nulle de connaissance non interactives. Les définitions usuelles de sécurité de ces primitives requièrent intrinsèquement la capacité d'enregistrer et de comparer des chaînes classiques. Cependant, les tâches d'enregistrement et de comparaison sont extrêmement difficiles dans le monde quantique en raison du principe d'incertitude. Nous proposons deux alternatives afin de surmonter cette barrière. De plus, nos notions de sécurité sont les premières à prendre pleinement en compte les attaques quantiques dans lesquelles les attaquants peuvent interagir avec les utilisateurs finaux sur des canaux quantiques.

D'autre part, nous montrons que la disponibilité des ordinateurs quantiques se révèle être également à l'avantage des cryptographes, même lorsque les utilisateurs finaux n'utilisent que des communications classiques. En particulier, nous présentons un protocole interactif entre une Alice classique et un Bob quantique. Ce dispositif permet à Alice d'envoyer un état quantique caché non clonable à Bob par des canaux classiques. En outre, cet état quantique non clonable établit une forte propriété dite de monogamie de l'intrication, qui décrit les limites de la force des corrélations multipartites quantiques. Enfin, nous appliquons notre protocole et nous donnons les premiers schémas semi-quantiques de protection contre la copie.

Mots clés : Cryptographie quantique, Modèles de sécurité, Chiffrement, Preuves à divulgation nulle de connaissance, Cryptographie non clonable.

Abstract

Modern cryptography has a major foe on the horizon: the inevitable rise of quantum computers. However, the same computing power will also unlock solutions to cryptographic tasks that are simply impossible to achieve with the current technology. This thesis sets foot in a ubiquitous quantum world, where everyone will be running quantum computers, with two main contributions.

Firstly, we put forth new security models and security analyses for two cryptographic primitives: encryption and non-interactive zero- knowledge proofs. Classical security definitions of these primitives inherently require the ability to record and compare classical strings. However, the tasks of recording and comparing are highly non-trivial in the quantum setting, due to the quantum uncertainty principle. We propose two different ways to overcome this recording barrier. Our security notions are the first to fully capture quantum attacks in which the codebreakers can interact with the end-users over quantum channels.

Secondly, we show that the availability of quantum computers turns out to be also the advantage of codemakers, even when the end- users only use classical communication. In particular, we exhibit an interactive protocol between a classical Alice and a quantum Bob which allows Alice to send a hidden unclonable quantum state to Bob through classical channels. Furthermore, the constructed unclonable quantum state establishes a strong monogamy-of-entanglement property, which describes the limitations on the strength of quantum multipartite correlations. We further apply our protocol to quantum copy-protection and give the first semi-quantum copy-protection schemes.

Keywords: Quantum Cryptography, Security Models, Encryption, Zero-Knowledge Proofs, Unclonable Cryptography.

Contents

Chapter

1 Introduction

Chapter content

Cryptography, the science dedicated to studying the protection of information, has become prevalent within a few decades. Protection has multiple meanings, in which the most basic protection mechanism is achieved by means of encryption. Encryption schemes are further divided into two categories: *symmetric* encryption and *public-key* encryption. The latter concept was originally put forth by Diffie and Hellman [DH76], and since then, has become ubiquitous. In practice, for example, with the Transport Layer Security (TLS) protocol, which has widespread use on the Internet, hybrid approaches are used, with public-key cryptography establishing a shared secret key to be used to encrypt the messages with a symmetric encryption. Beyond these basic notions, many other advanced cryptographic primitives have also been proposed to address more complex requirements of protection. Unfortunately, the security of all currently widely-used cryptosystems is threatened by the potential advent of *full-scaled quantum computers*.

1.1 History of Quantum Computing

The field of quantum computing originated in the 1980s as a subfield of quantum physics. Some physicists and mathematicians, including notably Richard Feynman [Fey82] and Yuri Manin [Man80], had remarked that the task of simulating quantum mechanical systems would pose a challenge to classical computers, and proposed the idea of quantum models of computation. In 1985, David Deutsch formalized this idea under the notion of *universal quantum Turing machine* [Deu85], and raised the question whether quantum computers might have a strict advantage over classical computers at solving problems. A few early quantum algorithms were developed: for example, Deutsch-Jozsa's [DJ92], and the theory of *quantum complexity* appeared [BV93]. Soon after, David Simon showed that a quantum computer could achieve an exponential speedup in solving an idealized version of the problem of finding the period of function [Sim94]. Though Simon's problem had no obvious applications, it inspired Peter Shor [Sho99] who formulated an efficient quantum algorithm for computing discrete logarithms and factoring large numbers. This breakthrough did not only challenge the strong Church-Turing thesis, but also the widely-used cryptosystems based on these (classically) difficult problems.

1.2 Cryptography Meets Quantum Computers

On the one hand, Shor's discovery, and its obvious implications for cryptanalysis, caused interest in quantum computing to skyrocket. *(Post-)Quantum security* of *classical* cryptosystems was born with the prospect of finding the strengths and weaknesses of this new attacker, thereby finding new secure designs for an era of quantum computers.

On the other hand, quantum information can arguably be used as an advantage for *designing* cryptosystems. If we go back a bit further, the relationship between quantum information and cryptography is almost half-a-century old: in a remarkable 1968 manuscript that first pioneered the idea of *conjugate coding* [Wie83], Stephen Wiesner also proposed a scheme for *quantum money*, a cryptographic primitive that would be unimaginable without quantum mechanics. Indeed, this paper is "arguably the foundational document of the entire field of quantum information science".[1] Wiesner's ideas have led to the first scheme for quantum key distribution [BB84] – what we now call *BB84*, and created a new branch of cryptography: *quantum cryptography* (also called *unclonable cryptography*). These bewildering possibilities in the quantum world are only possible thanks to one simple but deep difference between classical and quantum information, that is, classical information can in principle always be copied and quantum information cannot. This principle is called the *quantum no-cloning theorem* [WZ82], formulated as:

There is no procedure $|\psi\rangle \mapsto |\psi\rangle \otimes |\psi\rangle$*, for an arbitrary quantum state* $|\psi\rangle$*.*

In short, while quantum information opens up the cryptographic landscape to allow functionalities that do not exist classically, the availability of quantum computing to cryptography also hands us unique challenges and limitations, some of them are seemingly trivial in the classical setting.[2] In a few decades, quantum information and quantum computation have both become: (i) a bogeyman of classical cryptography; and (ii) a friend of quantum cryptography. These two directions of studying quantum security and quantum cryptography are therefore the main motivation of our work.[3]

1.2.1 Quantum Security of Classical Cryptosystems

As a starting point, let us exemplify *classical* security notions for encryption. The security of encryption schemes is defined formally as a game between an adversary that tries to win, that is, to trigger a particular event, or learn some particular information (for instance, the adversary wins if it can recover the encrypted message only knowing the public information), and a challenger that interacts with the adversary. The game specifies which messages are sent by the challenger depending on the adversary's behavior, and the winning condition for the adversary. The security game is defined in such a way that the adversary's capabilities encompass all possible attacks that could reasonably occur in a real-life scenario.

Defining security is a challenging task that has prompted fundamental research papers, such as [GM84], which defined the notions of semantic security and indistinguishability-based security for encryption. Before the age of quantum computers, many fundamental

[1] Famously, this paper was so far ahead of its time, before quantum information became an area of study. It was rejected once and took nearly 15 years to get published [Aar].

[2] Examples include copying and comparing data.

[3] This explains the title of our manuscript.

results were proven secure in a model where attackers are limited to efficient classical computation. In the age of quantum computers, the security model changes dramatically: the adversary can always perform *local* quantum computations, but the way it *accesses* the data plays an important role, leading to the following scenarios:

- **Post-Quantum Security.** This security level is against an adversary who wants to attack a classical cryptosystem by using a quantum computer they have in their basement, that is, the adversary can perform offline quantum computations, but it can only access the private oracles (e.g., decryption, authentication) through *classical* queries. For instance, in the security game for encryption, it means that the communication between the adversary and the challenger is completely classical.

  The prototypical example is simply downloading a public key for RSA off the Internet and then running Shor's algorithm on it to recover the secret key. Another example is the security of schemes requiring the use of public hash functions, which is usually captured in the quantum random oracle model [BDFL+11].

  This attack setting is certainly the most meaningful. It will be applicable to all classical cryptosystems once a scalable quantum architecture is built, but it also concerns today's encrypted traffic, if an attacker can record it and wait patiently for a quantum computer to appear.

- **Quantum Security.** This second category is very different from the first. In this setting, the adversary is significantly stronger: besides having a quantum computer in their basement, they also have the ability to interface *quantumly* with portions of the cryptosystem that involve our private key. For instance, in the security game for encryption, the adversary can ask for encryption of messages, or decryption of ciphertexts of its choice *in superposition*. This corresponds to the *quantum query model* in the literature on provable quantum security, for instance in [Zha12a; BZ13a; BZ13b; DFNS14; KLLN16a].

  Unlike the case of public hash functions, there is no clear generic mechanism for how an adversary could gain this kind of access. While it is not yet clear whether there are settings in which this model is directly, practically relevant, there is arguably no question about the desirability of quantum security for several reasons:

  - **Theoretical interest.** The quantum security of cryptographic primitives (beyond random oracles) is an active area of research with many exciting results, such as quantum secure pseudorandom functions [Zha12a], digital signatures [BZ13a; BZ13b; AMRS20], encryption schemes [BZ13b; GHS16], zero-knowledge proofs [BKS21]. These primitives have also led to unexpected applications, such as pseudorandom quantum states [JLS18; AQY22; MY22].
  - **Composability.** This security model ensures that classical cryptosystems retain their security even if executed on a quantum computer, possibly in complex environments or protocols where composition should be taken into account.
  - **Security proof concerns.** Security reductions might require more that just post-quantum security of underlying primitives. An example is the need of quantum-secure pseudorandom functions in order to simulate a quantum random oracle. Another example is the case of code obfuscation (see [SW14]), in which the

obfuscated program will be run on the adversary's quantum computers. For example if the adversary is given a public encryption key which is generated by hardcoding a symmetric key into an obfuscated encryption program, in which the adversary has access to a full description of the oracle with a secret key and can implement a quantum embedding of it.

- **Device-independent security.** This model ensures security in a world where even end-users are using quantum devices, and hence can potentially interact with the adversary using quantum communication. Even if the end-users are not running full-fledged quantum computers, their devices may exhibit some quantum effects. For example, modern processors have reached the point where quantum tunneling is an important consideration [Zha15]. Other settings include the case where a quantum computer is used to run a classical algorithm, but an adversary manages to have control over the measurement devices of the victim. In this setting, classicalization is burden on hardware designers, and it mounts to a *hardware* assumption, which, from the security point of view, is undesirable.
- **Security in the worst-case.** In this model, we achieve security in the worst-case scenario where the adversary is controlling the hardware. So it can be thought of as a more conservative security notion, since proving security of a construction with respect to a stronger security notion will give more confidence in using it.

Thus, an important goal towards understanding whether cryptographic protocols will resist quantum attacks is

Question 1: *Upgrade security models for classical cryptosystems to handle quantum attackers.*

Terminology. These two security models are also referred to as the *Q1* (for post-quantum security), and *Q2* (for quantum security) models in literature ([KLLN16a; KLLN16b; HS18a; HS18b]).

1.2.2 Unclonable Cryptography

On the other hand, properties of quantum mechanics have enabled the emergence of quantum cryptographic protocols achieving important goals which are proven to be impossible classically, with credits given to the no-cloning principle. Looking beyond the notion of quantum money [Wie83], quantum encoding can further achieve richer levels of applicability, which defines a hierarchy of "unclonable" objects corresponding to different variants of the no-cloning principle [AGKZ20; BJLP+21].

Wiesner's quantum money [Wie83] is a direct application of the no-cloning principle in cryptography, where the adversary obtains truly unknown quantum states. Public-key quantum money [AC12] can be seen as a strengthening, where no-cloning still holds even for parties that have the ability to verify the state. Quantum lightning [Zha19b] is then a further strengthening, where no-cloning holds even for parties that devised the original state themselves. Tamper-evident encryption [Got03], unclonable encryption [BL20] and certified deletion [BI20; HMNY21; HMNY22] apply no-cloning to information, meaning

that there is some underlying data that can be decoded, but there are limitations on the possibility of copying this data while it is encoded. And yet another extremely strong variant of the no-cloning principle is where the unclonable state has been endowed with some *functionality*. The functionality level of this hierarchy was first discussed in terms of *quantum copy-protection* by Aaronson [Aar09]: here, a quantum encoding allows the evaluation of a function on a chosen input, but in a way that the number of *simultaneous* evaluations is limited. A related concept to quantum copy-protection is the notion of secure software leasing [AL21; BJLP+21; KNY21]: where the unclonable state allows evaluation of a circuit, while also enabling the originator to verify that the software is *returned*. There are several other application-specific notions, also at the functionality level of the hierarchy, including: single-decryptor encryption [CLLZ21], which can be seen as quantum copy-protection of the decryption algorithm, and tokenized digital signatures [BS17; AGKZ20], which is a quantum one-time program [BGS13] for digital signatures.

By standard definition, these quantum primitives can be seen as a two-party protocol requiring quantum communication to transfer the quantumly encoded program between parties, and of course, local quantum computation from both parties. Ideally, for both theoretical and practical reasons, we might want to minimize the required model and use local quantum computation and only classical communication. With only classical communication, however, these notions become unusable. This then leads to the following natural question

Question 2: *Achieve quantum unclonable functionalities with classical communication.*

Terminology. In this thesis, we will use the term *quantum protocols* to mean cryptographic protocols requiring quantum computation and quantum communication, and *semi-quantum protocols* to mean cryptographic protocols requiring quantum computation and classical communication.[a]

[a] This is called *hybrid quantum cryptography* in [AGKZ20].

1.3 Contributions of the Thesis

Part I Quantum Security. There has been towards this goal extensive research works that consider this scenario of quantum superposition attacks for different classical cryptographic constructions such as random oracles [BDFL+11], pseudorandom functions [Zha12a], encryption [BZ13b; GHS16] and signatures [BZ13a; AMRS20] and give corresponding new security definitions. With this wide program, in this thesis, we focus, in particular, on two basic cryptographic primitives: encryption and non-interactive zero-knowledge.

On Security Notions for Encryption in a Quantum World [CEV20]. The results of this paper are presented in Chapter 3. Indistinguishability against adaptive chosen-ciphertext attacks (IND-CCA2) is usually considered the most desirable security notion for classical encryption. The security of quantum-secure classical encryption has first been studied by Boneh and Zhandry [BZ13b], but they

restricted the adversary to classical challenge queries, which makes the indistinguishability only hold for classical messages (denoted as IND-qCCA2). We extend their work by giving the first security notions for fully quantum indistinguishability under quantum adaptive chosen-ciphertext attacks, where the indistinguishability holds for superposition of plaintexts (denoted as qIND-qCCA2). We then show that our notions are strictly stronger than previous notions with classical challenge queries, and achieve composability as the classical definitions. We also provide constructions satisfying these security notions. For the symmetric-key setting, our construction follows the classical Encrypt-then-MAC paradigm, in which we use a pseudorandom function in the role of the MAC scheme. For the public-key setting, we propose a compiler that unconditionally lifts any secure encryption scheme in the sense of [BZ13b] to an encryption scheme secure in the sense of our notions. In fact, our feasibility results show that quantum security can be achieved for free, without any new assumptions rather than ones needed for post-quantum security. Finally, we show that in the quantum setting, one-bit encryption schemes are necessary and sufficient to build many-bit encryption schemes, similar to the classical setting [Ms09].

This paper has been presented at QCrypt in 2020, and published in the proceedings of the conference INDOCRYPT in 2022.

Quantum-Simulation-Sound Non-Interactive Zero-Knowledge: Definitions, Constructions and Applications [ACEM+22].

The results of this paper are presented in Chapter 4. Non-interactive zero-knowledge (NIZK) proof systems have very numerous applications in modern cryptographic protocols. We say that a NIZK proof system is simulation-sound (SS) if an adversary cannot provide a convincing proof for a false statement, even after receiving a polynomial number of simulated proofs of (possibly false) statements. In this paper, we present a quantum simulation-soundness definition that allows superposition access to the simulator. We give a separation result between post-quantum and quantum security of SSNIZK, and prove that both Sahai's construction for SSNIZK [DDOP+01; Sah01] (in the common reference string model) and the Fiat-Shamir transformation [FS87] (in the quantum random oracle model) can be made quantumly-simulation-sound. As an application, this allows us to prove quantum security of the Naor-Yung construction for achieving CCA encryption schemes from CPA encryption scheme and simulation-sound NIZKs. As a side result, we introduce a new notion of quantum-query advantage functions, which could be used as a general framework to show classical/quantum security separation for other cryptographic primitives.

Part II Quantum Cryptography. Quantum cryptography is known for enabling functionalities that are unattainable using classical information alone. Perhaps the most striking example of quantum primitives is the notion of *quantum copy-protection*, introduced by Aaronson [Aar09]. Informally, quantum copy-protection allows for a program to be encoded in a quantum state in such a way that the program can be evaluated, but not copied. Unfortunately, this usually comes at the cost of needing quantum power from every party in the protocol, while arguably a more realistic scenario would be a network of classical clients, classically interacting with a quantum server. An emerging field of "dequantizing" quantum cryptographic protocols has shown that it is possible to use local

quantum computation and classical communication to obtain cryptographic constructions which are otherwise classically impossible [BCMV+18; Mah18b; AGKZ20; RS20; HMNY21; KNY21; GMP22; Shm22a; Shm22b]. We continue this research direction in the second part of the thesis.

Semi-Quantum Copy-Protection and More [CHV22].

The results of this paper are presented in Chapter 5. In this paper, we focus on copy-protection, which is a quantum primitive that allows a program to be evaluated, but not copied, and has shown interest especially due to its links to other unclonable cryptographic primitives. Our main contribution is to show how to dequantize existing quantum copy-protection from hidden coset states: we give a construction for *classically-instructed remote state preparation for coset states*, based on the existence of indistinguishability obfuscation for classical circuits and the Learning With Errors [Reg05] problem. Our protocol is a multi-round protocol between classical Alice and quantum polynomial-time Bob that allows Alice to delegate the construction of hidden coset states to Bob. Furthermore, Alice knows the description of the constructed coset states (which reside on Bob's device), while Bob himself does not, and no-cloning also applies to these states. Hence, the situation at the end of this protocol is equivalent to one where Alice sent hidden coset states to Bob, allowing us to dequantize existing quantum copy-protection from coset states in a generic and modular way. To broaden the applicability of our semi-quantum protocol, we also present in this work a copy-protection for point functions in the plain model, to which our dequantizer could be applied. In fact, our copy-protection scheme is almost identical to that for pseudorandom functions given in [CLLZ21]. We observe that by making few modifications to their proof, we obtain a copy-protection scheme for point functions with a non-trivial challenge distribution in the security definition. We note that before this paper, no copy-protection scheme for point functions in the plain model with negligible security was known.

Other contributions. The recent NIST call for post-quantum encryption and signature schemes has revived the interest for designing *post-quantum* advanced cryptographic protocols such as oblivious transfer and non-interactive zero-knowledge proof systems. We give below a brief description of our contributions in this direction, which are not included in this manuscript.

Post-Quantum UC-Secure Oblivious Transfer in the Standard Model with Adaptive Corruptions [BCV19].

We describe in this paper an oblivious transfer (OT) scheme which is post-quantum, universal-composability-secure, and deals with adaptive corruptions assuming reliable erasures. Since the seminal result of Kilian [Kil88], oblivious transfer has proven to be a fundamental primitive in cryptography. In such a scheme, a user is able to gain access to an element owned by a server, without learning more than this single element, and without the server learning which element the user has accessed. This primitive has received a lot of study in the literature, among which very few schemes are based on lattices. To the best of our knowledge, this is the first post-quantum OT scheme with such a high level of security. Our methodology relies on the generic construction of [BC15]. In order to instantiate the necessary

building blocks, we replace the use of the smooth projective hash functions (SPHFs) construction of [KV09] by a chameleon hash function, an IND-CCA2 encryption scheme and an SPHF construction from [BBDQ18]. This allows us to give an SPHF-friendly commitment scheme based on the Learning with Errors problem [Reg05], which can be seen as a side contribution of the paper. Furthermore, we propose concrete parameters and an implementation of our scheme.

This paper has been published in the proceedings of the conference ARES in 2019.

zkSNARKs from Codes with Rank Metrics [DMV22].

Zero-knowledge succinct arguments of knowledge (zkSNARKs) are non-interactive proof systems enabling efficient privacy-preserving proofs of membership for NP languages. A large body of work has studied candidate constructions that are secure against quantum attackers, which are based on either lattice assumptions, or post-quantum collision-resistant hash functions. In this paper, we propose a code-based zkSNARK scheme, whose security is based on the Rank Support Learning (RSL) problem, a variant of the random linear code decoding problem in the rank metric. Our construction follows the general framework of Gennaro *et al.* [GMNO18], which is based on Square Span Programs (SSPs). Due to the fundamental differences between the hardness assumptions, our proof of security cannot apply the techniques from the lattice-based constructions, and indeed, it distinguishes itself by the use of techniques from coding theory. We also provide the scheme with a set of concrete parameters.

Road-map. The rest of this thesis is organized as follows. In Chapter 2, we introduce the notation, the relevant background on quantum information and computation, as well as notions of cryptographic primitives that will be used throughout this thesis. In Chapter 3, we present our definitions of quantum indistinguishability for encryption. Then, in Chapter 4, we present our notions of quantum simulation-sound non-interactive zero-knowledge. Finally, in Chapter 5, we exhibit our work on copy-protection. We also give in the appendices several additional results about tokenized digital signatures (Appendix A) and password-based key exchange protocols (Appendix B) which have been done during the preparation of this thesis.

Chapter

2 Preliminaries

This preliminary chapter aims to fix the notations and to recall the notions we will use throughout this thesis.

Chapter content

2.1 Notation

Throughout this thesis, λ denotes the security parameter. The notation $\mathsf{negl}(\lambda)$ denotes any function f such that $f(\lambda) = \lambda^{-\omega(1)}$, and $\mathsf{poly}(\lambda)$ denotes any function f such that $f(\lambda) = \mathcal{O}(\lambda^c)$ for some $c > 0$. Ω denotes some fixed finite alphabet with $0 \in \Omega$, we usually think of Ω as $\{0,1\}$. For $a, b \in \mathbb{R}$, $[a,b] \coloneqq \{\, x \in \mathbb{R} \mid a \leq x \leq b \,\}$ and $[\![a,b]\!] \coloneqq \{\, x \in \mathbb{Z} \mid a \leq x \leq b \,\}$ will denote the closed real and integer intervals with endpoints a and b. With an abuse of notation, we will write $[\![n]\!]$ as shorthand for $[\![0, n-1]\!]$. For a string $q \coloneqq q_1 \dots q_n \in \Omega^n$ of arbitrary length $n \geq 0$, the Hamming weight of q is defined as the number of non-zero entries in q: $\mathrm{wt}(q) \coloneqq |\{\, i \in [\![1,n]\!] \mid q_i \neq 0 \,\}|$. We also use the

notion of the relative Hamming weight of q, defined as $\omega(q) \coloneqq \mathrm{wt}(q)/n$. By convention, the relative Hamming weight of the empty string $\perp$ is set to $\omega(\perp) \coloneqq 0$. For a set $I = \{i_1, \ldots, i_\ell\} \subseteq [\![1, n]\!]$ and a n-bit string $x \in \Omega^n$, we write $x|_I \coloneqq x_{i_1} \cdots x_{i_\ell}$, and $\overline{I}$ as the complement $\overline{I} \coloneqq [\![1, n]\!] \setminus I$ of I.

When sampling uniformly at random a value a from a set $\mathcal{U}$, we employ the notation $a \xleftarrow{\$} \mathcal{U}$. When sampling a value a from a probabilistic algorithm $\mathcal{A}$, we employ the notation $a \leftarrow \mathcal{A}$. Let $|\cdot|$ denote either the length of a string, or the cardinal of a finite set, or the absolute value. By PPT we mean a polynomial-time non-uniform family of probabilistic circuits, and by QPT we mean a polynomial-time family of quantum circuits. For a probabilistic algorithm f, we write $f(x; r)$ to denote the computation of f on input x with randomness r drawn uniformly at random. We sometimes omit the randomness and just write $f(x)$. Finally, let $\delta_{x,x'}$ denote the Kronecker delta function of x and x'.

Sub-exponential Security. A system is sub-exponentially secure if there is an adversary of (quantum) polynomial size that breaks the system with sub-exponentially small probability. Typically, this probability is upper bounded by $2^{-\lambda^\varepsilon}$, for some constant $0 < \varepsilon < 1$.

2.2 Quantum Information and Computation

We use $\mathcal{H}$ to denote an arbitrary finite-dimensional Hilbert space, and use indices to differentiate between distinct spaces. We let $|\phi\rangle$ denote an arbitrary pure quantum state, and $|x\rangle$ denote an element in the standard (computational) basis. A mixed state will be denoted by lowercase Greek letters, e.g., ρ. The map $\mathrm{Tr} : \mathcal{L}(\mathcal{H}) \to \mathbb{C}$ denotes the trace, and $\mathrm{Tr}_B : \mathcal{L}(\mathcal{H}_A \otimes \mathcal{H}_B) \to \mathcal{L}(\mathcal{H}_A)$ is the partial trace over subsystem B. $\mathrm{Pos}(\mathcal{H})$ denotes the set of positive semidefinite operators on $\mathcal{H}$, and $\mathcal{D}(\mathcal{H}) \coloneqq \{A \in \mathrm{Pos}(\mathcal{H}) \mid \mathrm{Tr}[A] = 1\}$ is the set of density matrices on $\mathcal{H}$.

The single qubit Pauli operators are $\sigma_X \coloneqq \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ and $\sigma_Z \coloneqq \left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$. The Hadamard basis states are written as $|(-)^b\rangle \coloneqq \frac{1}{\sqrt{2}}(|0\rangle + (-1)^b |1\rangle)$. We also sometimes write $|+\rangle$ for $|(-)^0\rangle$ and $|-\rangle$ for $|(-)^1\rangle$.

A pure state $|\phi\rangle$ can be manipulated by performing a unitary transformation U to the state $|\phi\rangle$, which we denote $U|\phi\rangle$. The identity on a n-bit quantum system is denoted $\mathcal{I}_n$. Given two quantum systems A, B, with corresponding Hilbert spaces $\mathcal{H}_A, \mathcal{H}_B$, let $|\phi\rangle \coloneqq |\phi_0, \phi_1\rangle$ be a state of the joint system. We write $U_A|\phi\rangle$ to denote that we act with U on register A, and with identity $\mathcal{I}$ on register B, and we write U_{AB} to denote that we act with U on both registers A, B simultaneously, that is $U_{AB} = U_A \otimes U_B$.

An observable on $\mathcal{H}$ is a Hermitian linear operator on $\mathcal{H}$. A binary observable is an observable that only has eigenvalues in $\{-1, 1\}$. For a binary observable O and $b \in \{0, 1\}$, we denote by $O^{(b)}$ the projector onto the $(-1)^b$-eigenspace of O. For any procedure which takes a quantum state as input and produces a bit (or more generally an integer) as output, e.g., by measuring the input state, we denote the probability distribution over outputs b on input state ψ by $\Pr[\, b \,|\, \psi \,]$.

2.2.1 Quantum Computation

Quantum gates. We refer to the following well-known unitary gates:

- *Pauli gates:* $\mathsf{X} : |a\rangle \mapsto |1-a\rangle$, $\mathsf{Z} : |a\rangle \mapsto (-1)^a |a\rangle$ and $\mathsf{Y} \coloneqq i\mathsf{XZ}$, for each $a \in \{0,1\}$.
- *Hadamard gate:* $\mathsf{H} : |a\rangle \mapsto \frac{1}{\sqrt{2}} |0\rangle + \frac{(-1)^a}{\sqrt{2}} |1\rangle$, for each $a \in \{0,1\}$.
- *Rotation gates:* $\mathsf{R}_\phi : |a\rangle \mapsto e^{ia\phi} |a\rangle$, for each $a \in \{0,1\}$. We obtain the T gate where $\phi = \frac{\pi}{4}$, the phase gate P where $\phi = \frac{\pi}{2}$.
- *Controlled gates:* for any k-qubit unitary quantum gate U, we define the controlled-U as: $\mathsf{Ctrl\text{-}U} : |a\rangle |x\rangle \mapsto |a\rangle \mathsf{U}^a |x\rangle$, for each $a \in \{0,1\}$ and $x \in \{0,1\}^k$. In particular, we write the controlled-NOT gate as $\mathsf{CNOT} : |a\rangle |b\rangle \mapsto |a\rangle |b \oplus a\rangle$.
- *Toffoli gates:* $\mathsf{CCNOT} : |a,b,c\rangle \mapsto |a,b,c \oplus (a \cdot b)\rangle$ for each $(a,b,c) \in \{0,1\}^3$.

Quantum Fourier transform. Let Q be a n-bit quantum system over $\mathbb{Z}_q$ for some integer q. The Quantum Fourier transform QFT performs the following operation efficiently:

$$\mathsf{QFT} |x\rangle \coloneqq \frac{1}{\sqrt{q^n}} \sum_{y \in \{0,1\}^n} \omega_q^{x \cdot y} |y\rangle ,$$

where $\omega_q \coloneqq \exp(\frac{2\pi i}{q})$, and $x \cdot y$ denotes the dot product. In this thesis, we usually consider $q = 2$, so that $\omega_q = (-1)$.

Oracle access to an interactive quantum machine. We say that a quantum algorithm $\mathcal{A}$ has oracle access to an interactive quantum machine M (and we write this as $\mathcal{A}^M$ if $\mathcal{A}$ can only make classical queries to M, or $\mathcal{A}^{|M\rangle}$ to emphasize that M is a quantum machine and that oracle access includes the ability to make queries in superposition and apply the inverse of M) to mean the following. Besides the security parameter and its own classical input x, we allow $\mathcal{A}$ to execute the quantum circuit U specifying M, and its inverse (recall that these act on the internal register and on the network register of M). Moreover, we allow $\mathcal{A}$ to provide and read messages from M (formally, we allow $\mathcal{A}$ to act freely on the network register). We do not allow $\mathcal{A}$ to act on the internal register of M, except via U or its inverse.

Given a function $f : \mathcal{X} \to \mathcal{Y}$, we model a quantum-accessible oracle $\mathcal{O}$ for f as a unitary transformation $\mathcal{O}_f$ as follows.

- The *standard* oracle model: $\mathcal{O}_f$ acts on three registers X, Y, Z with the property that $\mathcal{O}_f : |x,y,0\rangle \mapsto |x, y \oplus f(x), 0\rangle$, where $\oplus$ is some involutive group operation.
- The *minimal* oracle model: $\mathcal{O}_f$ acts on two registers X, Y with the property that $\mathcal{O}_f : |x,0\rangle \mapsto |f(x),0\rangle$. This model is implementable if and only if f is a bijective function.

Randomness. If an oracle $\mathcal{O}$ implements a classical randomized algorithm, there are several choices for how the randomness is used in each query if the oracle is queried in superposition. One option is to choose fresh randomness for each message in the superposition. Another option is to choose a single randomness value for each query, and generate output in the superposition with that randomness. We note that there is a simple transformation that converts an oracle requiring independent randomness for every message into a scheme that is secure when a single randomness value is used for an

entire query: for each query, choose a fresh random key k for a quantum pseudorandom function (QPRF) (see Definition 2.7). This will be the single per-query randomness value. Each message m in the superposition will be answered using randomness obtained by applying the QPRF to m using the key k. From the adversary's point of view, this is indistinguishable from choosing independent randomness for each message. Indeed, Zhandry [Zha12b] shows that we can replace the QPRF with a function drawn from a pairwise independent function family, which allows us to achieve perfect simulability. For this reason, requiring global randomness per query does not change the oracle from the adversary's point of view, but greatly simplifies its implementation. In this work, we choose the second approach and all randomized oracles are implemented this way.

2.2.2 Efficiency in the Quantum Setting

Definition 2.1 — Efficiency

Efficient unitaries: a family of unitaries $\{U_\lambda \in \mathcal{U}(\mathcal{H}_\lambda)\}_{\lambda\in\mathbb{N}}$ is efficient if there exists a (classical) polynomial-time Turing machine M that, on input 1^λ, outputs a description of a circuit (with a fixed gate set) that implements the unitary.

Efficient isometries: a family of isometries $\{V_\lambda : \mathcal{H}_{A_\lambda} \to \mathcal{H}_{B_\lambda}\}_{\lambda\in\mathbb{N}}$ is efficient if there exists an efficient family of unitaries $\{U_\lambda \in \mathcal{U}(\mathcal{H}_{B_\lambda})\}_{\lambda\in\mathbb{N}}$ such that $V_\lambda = U_\lambda(\mathcal{I}_{A_\lambda} \otimes |0_{k(\lambda)}\rangle)$, where $k(\lambda) = \dim(\mathcal{H}_{B_\lambda}) - \dim(\mathcal{H}_{A_\lambda})$.

Efficient observables: a family of binary observables $\{Z_\lambda : \mathrm{Herm}(\mathcal{H}_{A_\lambda})\}_{\lambda\in\mathbb{N}}$ is efficient if there exists a family of Hilbert spaces $\mathcal{H}_{B_\lambda}$ with $\dim(\mathcal{H}_{B_\lambda}) = \mathrm{poly}(\lambda)$, and a family of efficient unitaries $\{U_\lambda \in \mathcal{U}(\mathcal{H}_{A_\lambda} \otimes \mathcal{H}_{B_\lambda})\}_{\lambda\in\mathbb{N}}$ such that for any $|\psi\rangle_A \in \mathcal{H}_A$:

$$U^\dagger(\sigma_Z \otimes \mathcal{I})U_\lambda(|\psi\rangle_A |0\rangle_B) = (Z_\lambda |\psi\rangle_A) \otimes |0\rangle_B . \tag{2.1}$$

Efficient measurements: a family of measurements $\{M_\lambda = \{M_\lambda^{(i)} \in \mathcal{L}(\mathcal{H}_{A_\lambda})\}_{i\in\mathcal{A}}\}_{\lambda\in\mathbb{N}}$ is efficient if the isometry

$$|\psi\rangle \mapsto \sum_{i\in\mathcal{A}} |i\rangle \otimes M_\lambda^{(i)} |\psi\rangle \tag{2.2}$$

is efficient.

2.2.3 Distance Measures

Definition 2.2 — Norms

Let $A \in \mathcal{L}(\mathcal{H})$ with singular values $\lambda_1, \ldots, \lambda_n \geq 0$. Then, the trace norm is defined as

$$\|A\|_1 := \sum_i \lambda_i .$$

Definition 2.3 — Trace Distance

For two quantum states $\rho, \sigma \in \mathrm{Pos}(\mathcal{H})$, the trace distance between them is

$$\Delta(\rho, \sigma) := \frac{1}{2}\|\rho - \sigma\|_1.$$

Definition 2.4 — Approximate Equality, [MV21, Definition 2.8 and Definition 2.14]

We overload the symbol "$\approx$" in the following ways (leaving the dependence on the security parameter implicit in the quantities on the left):

Complex numbers: For $a, b \in \mathbb{C}$ we define:

$$a \approx_\varepsilon b \iff |a - b| = O(\varepsilon) + \mathsf{negl}(\lambda)\,.$$

Operators: For $A, B \in \mathcal{L}(\mathcal{H})$, we define:

$$A \approx_\varepsilon B \iff \|A - B\|_1^2 = O(\varepsilon) + \mathsf{negl}(\lambda)\,.$$

(We will most frequently use this for (possibly subnormalised) quantum states $A, B \in \mathrm{Pos}(\mathcal{H})$.)

Operators on a state: For $A, B \in \mathcal{L}(\mathcal{H})$ and $\psi \in \mathrm{Pos}(\mathcal{H})$, we define:

$$A \approx_{\varepsilon,\psi} B \iff \mathrm{Tr}\left[(A - B)^\dagger (A - B)\psi\right] = O(\varepsilon) + \mathsf{negl}(\lambda)\,.$$

Computationally indistinguishable states: For two (families of not necessarily normalised) states $\psi, \psi' \in \mathrm{Pos}(\mathcal{H})$ which are computationally indistinguishable up to δ (i.e., no QPT distinguisher has advantage exceeding δ in distinguishing ψ from ψ'[a]), we write:

$$\psi \overset{c}{\approx}_\delta \psi'\,.$$

We can also define computational indistinguishability with respect to *non-uniform* QPT algorithms with *quantum advice*, denoted by $\mathcal{A} := \{\mathcal{A}_\lambda, \phi_\lambda\}_{\lambda \in \mathbb{N}}$, where each $\mathcal{A}_\lambda$ is the classical description of a $\mathrm{poly}(\lambda)$-size quantum circuit, and ϕ_λ is some (not necessarily efficiently computable) non-uniform $\mathrm{poly}(\lambda)$-qubit quantum advice. In this thesis, we implicitly consider computational indistinguishability with respect to non-uniform QPT adversaries with quantum advice, unless stated explicitly otherwise.

If we write $\approx_0$, we mean that the quantities are negligibly close. All asymptotic statements are understood to be in the limits $\varepsilon \to 0$ and $\lambda \to \infty$.

[a] A distinguisher $\mathcal{D}$ is a completely positive and trace-preserving map from the input state to a classical single-qubit state (i.e. a distribution over $\{0, 1\}$). The distinguishability is the trace distance between $\mathcal{D}(\psi)$ and $\mathcal{D}(\psi')$.

We include a copy of some technical lemmas on state-dependent operator relations using computational indistinguishability from [MV21] below for the reader's convenience.

Properties of the State-Dependent Distance

A feature of the state-dependent distance is that if two operators are close in the state-dependent distance, we can replace one operator by the other *acting on either side of the state*.

Lemma 2.1 (Replacement lemma [MV21, Lemma 2.21]). *Let $\psi \in \mathrm{Pos}(\mathcal{H})$, and $A, B, C \in \mathcal{L}(\mathcal{H})$. If $A \approx_{\varepsilon,\psi} B$ and $\|C\|_\infty = \mathcal{O}(1)$, then*

$$\mathrm{Tr}[CA\psi] \approx_{\varepsilon^{1/2}} \mathrm{Tr}[CB\psi] \,, \tag{2.3}$$

$$\mathrm{Tr}[AC\psi] \approx_{\varepsilon^{1/2}} \mathrm{Tr}[BC\psi] \,. \tag{2.4}$$

Lemma 2.2 ([MV21, Lemma 2.22]). *Let $A, B \in \mathcal{L}(\mathcal{H})$ be linear operators, $C \in \mathcal{L}(\mathcal{H})$ a linear operator with constant operator norm, and $\psi \in \mathrm{Pos}(\mathcal{H})$ with $\mathrm{Tr}[\psi] \leq 1$. Then, the following holds:*

$$A \approx_{\varepsilon,\psi} B \implies A\,\psi\,C \approx_\varepsilon B\,\psi\,C \quad \text{and} \quad C\,\psi\,A^\dagger \approx_\varepsilon C\,\psi\,B^\dagger \,. \tag{2.5}$$

The following lemma allows us to replace computationally indistinguishable states with one another in the state-dependent distance. This means that if two states are computationally indistinguishable and a state-dependent operator relation holds for one of the states, we can "lift" this relation to the other state, provided the operators are efficient.

Lemma 2.3 (Lifting lemma [MV21, Lemma 2.25]). *Let $\mathcal{H}, \mathcal{H}'$ Hilbert spaces with $\dim(\mathcal{H}') \geq \dim(\mathcal{H})$. Let $\psi, \psi' \in \mathcal{D}(\mathcal{H}')$ such that $\psi \overset{c}{\approx}_\delta \psi'$. Let A be an efficient binary observable on $\mathcal{H}$, B an efficient binary observable on $\mathcal{H}'$, and $V : \mathcal{H} \to \mathcal{H}'$ an efficient isometry. Then:*

$$VAV^\dagger \approx_{\varepsilon,\,\psi} B \implies VAV^\dagger \approx_{\varepsilon^{1/4}+\delta,\psi'} B \,. \tag{2.6}$$

Finally, we recall some further miscellaneous properties of the state-dependent distance.

Lemma 2.4 ([MV21, Lemma 2.18]). *Let $\psi_i \in \mathrm{Pos}(\mathcal{H})$ for $i \in \{1, \ldots, n\}$ with constant n, and $A, B \in \mathcal{L}(\mathcal{H})$. Define $\psi = \sum_i \psi_i$. Then:*

$$\forall i \in [\![1, n]\!] : \; A \approx_{\varepsilon,\psi_i} B \text{ iff } A \approx_{\varepsilon,\psi} B \tag{2.7}$$

Lemma 2.5 ([MV21, Lemma 2.24]). *Let $\mathcal{H}_1, \mathcal{H}_2$ be Hilbert spaces with $\dim(\mathcal{H}_1) \leq \dim(\mathcal{H}_2)$ and $V : \mathcal{H}_1 \to \mathcal{H}_2$ an isometry. Let A and B be binary observables on $\mathcal{H}_1$ and $\mathcal{H}_2$, respectively, $\psi \in \mathrm{Pos}(\mathcal{H}_1)$, and $\varepsilon \geq 0$. Then for any $b \in \{0, 1\}$:*

$$V^\dagger B V \approx_{\varepsilon,\psi} A \implies V^\dagger B^{(b)} V \approx_{\varepsilon,\psi} A^{(b)} \,, \tag{2.8}$$

$$B \approx_{\varepsilon, V\psi V^\dagger} VAV^\dagger \implies B^{(b)} \approx_{\varepsilon, V\psi V^\dagger} VA^{(b)}V^\dagger \,. \tag{2.9}$$

2.2.4 Quantum Random Oracle Model

A *random oracle* is a function $H : \{0,1\}^m \to \{0,1\}^n$ sampled from $\mathsf{Funcs}[m,n]$, the uniform distribution over functions from $\{0,1\}^m$ to $\{0,1\}^n$. The standard method to encode a random function H as a quantum operation is the unitary matrix $\mathcal{O}_H$, which acts as $|x,y\rangle \mapsto |x, y \oplus H(x)\rangle$. Another way to do it is to use the *compressed random oracle* CStO formalism of Zhandry [Zha19a].

For more details on the description of the compressed random oracle, we refer the reader to [Zha19a; DFMS22]. For the purposes of this thesis, we will only use the fact that the compressed random oracle CStO is a certain unitary matrix, indistinguishable from a real random oracle. We give here some technical lemmas that are used later.

We denote $\mathfrak{D} = \otimes_{x\in\mathcal{X}} \mathfrak{D}_x$ be the compressed random oracle registers, which corresponds to its database (denoted as D). The state space of $\mathfrak{D}_x$ is generated with vectors $|y\rangle$ for $y \in \mathcal{Y} \cup \{\perp\}$. The initial state of the register $\mathfrak{D}$ is $\otimes_{x\in\mathcal{X}} |\perp\rangle$. For a fixed relation $R \subset \mathcal{X} \times \mathcal{Y}$, Γ_R is the maximum number of y's that fulfill the relation R where the maximum is taken over all $x \in \mathcal{X}$:

$$\Gamma_R = \max_{x\in\mathcal{X}} |\{y \in \mathcal{Y} | (x,y) \in R\}|.$$

We define a projector $\Pi^x_{\mathfrak{D}_x}$ that checks if the register $\mathfrak{D}_x$ contains a value $y \neq \perp$ such that $(x,y) \in R$:

$$\Pi^x_{\mathfrak{D}_x} := \sum_{y:(x,y)\in R} |y\rangle\langle y|_{\mathfrak{D}_x}.$$

Let $\bar{\Pi}^x_{\mathfrak{D}_x} = \mathcal{I}_{\mathfrak{D}_x} - \Pi^x_{\mathfrak{D}_x}$. We define the measurement $\mathbb{M}$ to be the set of projectors $\{\Sigma^x\}_{x\in X\cup\{\perp\}}$ where

$$\Sigma^x := \bigotimes_{x'<x} \bar{\Pi}^{x'}_{\mathfrak{D}_{x'}} \otimes \Pi^x_{\mathfrak{D}_x} \text{ for } x \in X \text{ and } \Sigma^\perp := \mathcal{I} - \sum_x \Sigma^x. \tag{2.10}$$

Informally, the measurement $\mathbb{M}$ checks for the smallest x for which $\mathfrak{D}_x$ contains a value $y \neq \perp$ such that $(x,y) \in R$. If no register $\mathfrak{D}_x$ contains a value $y \neq \perp$ such that $(x,y) \in R$, the outcome of $\mathbb{M}$ is $\perp$. We define a purified measurement $\mathbb{M}_{\mathfrak{D}P}$ corresponding to $\mathbb{M}$ that XORs the outcome of the measurement to an ancillary register:

$$\mathbb{M}_{\mathfrak{D}P} |\phi, z\rangle_{\mathfrak{D}P} \to \sum_{x\in X\cup\{\perp\}} \Sigma^x |\phi\rangle_{\mathfrak{D}} |z \oplus x\rangle_P .$$

The following lemma states that the compressed random oracle and $\mathbb{M}_{\mathfrak{D}P}$ almost commute if Γ_R is small proportional to the size of $\mathcal{Y}$.

Lemma 2.6 ([DFMS22, Theorem 3.1]). *For any relation R and Γ_R defined above, the commutator $[\mathsf{CStO}, \mathbb{M}_{\mathfrak{D}P}]$ is bounded as follows:*

$$\|[\mathsf{CStO}, \mathbb{M}_{\mathfrak{D}P}]\| \leq 8 \cdot 2^{-n/2} \sqrt{2\Gamma_R}.$$

The following lemma says that the output of the adversary when making queries to a random oracle is identical to the one obtained by measuring the compressed database, except with negligible probability.

Lemma 2.7 ([Zha19a, Lemma 5]). *Let p be the probability that an adversary making queries to a random oracle $H : \{0,1\}^m \to \{0,1\}^n$ and outputting a tuple $(\vec{a}, \vec{b}, c)$ such that $|\vec{a}| = |\vec{b}| = k$ and $H(a_i) = b_i$ for each $i \in [\![k]\!]$. Let R be a collection of such tuples. Now consider running the adversary with the compressed oracle, and we measure the database D after the adversary procedures its output. Let p' be the probability that there exists a tuple $(\vec{a}', \vec{b}', c') \in R$ such that $D(a'_i) = b'_i$ for each $i \in [\![k]\!]$. Then $\sqrt{p} \leq \sqrt{p'} + \sqrt{k/2^n}$.*

2.2.5 Sampling in a Quantum Population

In this section, we describe a generic framework presented in [BF10] for analyzing cut-and-choose strategies applied to quantum states.

Classical Sampling Stratiegies

Let $q := (q_1, \ldots, q_n) \in \Omega^n$ be a string of length n. We consider the problem of estimating the relative Hamming weight of a substring $\omega(q|_{\bar{t}})$ by only looking at the substring $q|_t$ of q, for a subset $t \subset [\![1, n]\!]$. We consider sampling strategies $\Psi := (P_T, P_S, f)$, where P_T is an (independently sampled) distribution over subsets $t \subseteq [\![1, n]\!]$, P_S is a distribution over seeds $s \in S$, and $f : \{(t, v) : t \subset [\![1, n]\!], v \in \Omega^t\} \times S \to \mathbb{R}$ is a function that takes the subset t, the substring v, and a seed s, and outputs an estimate for the relative Hamming weight of the remaining string. For a fixed subset t, seed s, and a parameter δ, define $B^\delta_{t,s}(\Psi) \subseteq \Omega^n$ as

$$B^\delta_{t,s} := \{b \in \Omega^n : |\omega(b|_{\bar{t}}) - f(t, b|_t, s)| < \delta\}.$$

Then we define the *classical error probability* of strategy Ψ as follows.

Definition 2.5 — Classical Error Probability

The classical error probability of a sampling strategy $\Psi := (P_T, P_S, f)$ is defined as the following value, parameterized by $0 < \delta < 1$:

$$\varepsilon^\delta_{\text{classical}}(\Psi) := \max_{q \in \Omega^n} \Pr_{t \leftarrow P_T, s \leftarrow P_S} \left[q \notin B^\delta_{t,s}(\Psi) \right].$$

Quantum Sampling Strategies

Now, let $A := A_1, \ldots, A_n$ be an n-partite quantum system where the state space of each system A_i equals $\mathcal{H}_{A_i} = \mathbb{C}^d$ with $d = |\Omega|$, and let $\{|a\rangle\}_{a \in \Omega}$ be a fixed orthonormal basis of $\mathbb{C}^d$. A may be entangled with another system E, and we write the purified state on A and E as $|\psi\rangle_{AE}$. We consider the problem of testing whether the state on A is close to the all-zero reference state $|0\rangle_{A_1} \ldots |0\rangle_{A_n}$. There is a natural way to apply any sampling strategy $\Psi = (P_T, P_S, f)$ to this setting: sample t, s according to P_T, P_S, measure subsystems A_i for $i \in [\![1, t]\!]$ in basis $\{|a\rangle\}_a$ to observe $q|_t \in \Omega^{|t|}$, and compute an estimate $f(t, q|_t, s)$.

In order to analyze the effect of this strategy, we first consider the mixed state on registers T (holding the subset t), S (holding the seed s), and A, E that results from

sampling t and s according to $P_{TS} \coloneqq P_T P_S$

$$\rho_{TSAE} \coloneqq \sum_{t,s} P_{TS}(t,s) \left|t,s\right\rangle \left\langle t,s\right|_{TS} \otimes \left|\psi\right\rangle \left\langle \psi\right|_{AE}.$$

Next, we compare this state to an *ideal* state, parameterized by $0 < \delta < 1$, of the form

$$\widetilde{\rho}_{TSAE} \coloneqq \sum_{t,s} P_{TS}(t,s) \left|t,s\right\rangle \left\langle t,s\right|_{TS} \otimes \left|\widetilde{\psi}^{ts}\right\rangle \left\langle \widetilde{\psi}^{ts}\right|_{AE} \text{ with } \left|\psi^{ts}\right\rangle_{AE} \in \mathsf{span}\left(B^{\delta}_{t,s}\right) \otimes \mathcal{H}_E,$$

where

$$\mathsf{span}\left(B^{\delta}_{t,s}\right) \coloneqq \mathsf{span}\left(\{\left|b\right\rangle : b \in B^{\delta}_{t,s}\}\right) = \mathsf{span}\left(\{\left|b\right\rangle : \left|\omega(b|_{\bar{t}}) - f(t, b|_t, s)\right| < \delta\}\right).$$

That is, $\widetilde{\rho}_{TSAE}$ is a state such that it holds *with certainty* that the state on registers $A|_{\bar{t}}E$, after having measured $A|_t$ and observing $q|_t$, is in a superposition of states with relative Hamming weight δ-close to $f(t, q|_t, s)$. This leads us to the definition of the *quantum error probability* of strategy Ψ.

Definition 2.6 — Quantum Error Probability

The quantum error probability of a sampling strategy $\Psi \coloneqq (P_T, P_S, f)$ is defined as the following value, parameterized by $0 < \delta < 1$:

$$\varepsilon^{\delta}_{\mathsf{quantum}}(\Psi) \coloneqq \max_{\mathcal{H}_E} \max_{\left|\psi\right\rangle_{AE}} \min_{\widetilde{\rho}_{TSAE}} \Delta\left(\rho_{TSAE}, \widetilde{\rho}_{TSAE}\right),$$

where the first max is over all finite-dimensional registers E, the second max is over all state $\left|\psi\right\rangle_{AE}$ and the min is over all ideal state $\widetilde{\rho}_{TSAE}$ of the form described above.

Finally, we relate the classical and quantum error probabilities.

Theorem 2.1 ([BF10]). *For any sampling strategy Ψ and $\delta > 0$,*

$$\varepsilon^{\delta}_{\mathsf{quantum}}(\Psi) \leq \sqrt{\varepsilon^{\delta}_{\mathsf{classical}}(\Psi)}.$$

Remark 2.1. The results presented here immediately generalize from the all-zero reference state $\left|0\right\rangle \ldots \left|0\right\rangle$ to an arbitrary reference state $\left|\varphi\right\rangle_A$ of the form $\left|\varphi\right\rangle_A = U_1 \left|0\right\rangle \ldots U_n \left|0\right\rangle$ for unitary operators U_i acting on $\mathbb{C}^d$. Indeed, the generalization follows simply by a suitable change of basis, defined by the U_i's.

In this work, we will only need to analyze one simple sample-and-estimate strategy $\Psi_{\mathsf{uniform}} \coloneqq (P_T, P_S, f)$, where P_T is the uniform distribution over subsets $t \subseteq [\![1, n]\!]$, P_S is empty and $f(t, q|_t) = \omega(q|_t)$. That is, f receives a uniformly random subset $q|_t$ of q, and outputs the relative Hamming weight of $q|_t$ as its guess for the relative Hamming weight of $q|_{\bar{t}}$. The classical error probability of this strategy can be bound using Hoeffding inequalities, which is done in [BF10, Appendix B.3], where it is shown to be bounded by $4\exp(\frac{-n\delta^2}{32})$ for parameter δ. Thus, we have the following corollary of Theorem 2.1.

Corollary 2.1. *The quantum error probability of Ψ_{uniform} with parameter δ is*

$$\varepsilon^{\delta}_{\mathsf{quantum}}(\Psi_{\mathsf{uniform}}) \leq 2\exp(\frac{-n\delta^2}{64}).$$

2.3 Cryptographic Primitives

In this section, we give formal definitions of cryptosystems and their security notions that will be used in subsequent chapters. By default, the security of these notions are defined with respect to QPT adversaries (if we are in the computational setting).

2.3.1 Puncturable Pseudorandom Function

A pseudorandom function (PRF) system [GGM84] consists of a keyed function F and a set of keys $\mathcal{K}$ such that for a randomly chosen key $k \in \mathcal{K}$, the output of the function $F(k,x)$ for any input x in the input space $\mathcal{X}$ "looks" random to a QPT adversary, even when given polynomially many evaluations of $F(k,\cdot)$. Puncturable PRFs have an additional property that some keys can be generated *punctured* at some point, so that they allow to evaluate the PRF at all points except for the punctured points. Furthermore, even with the punctured key, the PRF evaluation at a punctured point still looks random.

Punctured PRFs are originally introduced in [BW13; KPTZ13; BGI14], who observed that it is possible to construct such puncturable PRFs for the construction from [GGM84], which can be based on any one-way function [HILL99].

Definition 2.7 — Puncturable Pseudorandom Function

A pseudorandom function $\mathsf{PRF} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is a *puncturable pseudorandom function* if there is an additional key space $\mathcal{K}_p$ and three PPT algorithms $\mathsf{pPRF} := \langle \mathsf{KeyGen}, \mathsf{Puncture}, \mathsf{Eval} \rangle$ such that:

$\mathsf{k} \leftarrow \mathsf{KeyGen}(1^\lambda)$. The key generation algorithm KeyGen takes the security parameter 1^λ as input and outputs a random key $\mathsf{k} \in \mathcal{K}$.

$\mathsf{k}\{x\} \leftarrow \mathsf{Puncture}(\mathsf{k}, x)$. The puncturing algorithm Puncture takes as input a PRF key $\mathsf{k} \in \mathcal{K}$ and $x \in \mathcal{X}$, and outputs a key $\mathsf{k}\{x\} \in \mathcal{K}_p$.

$y \leftarrow \mathsf{Eval}(\mathsf{k}\{x\}, x')$. The evaluation algorithm takes as input a (possibly punctured) key $\mathsf{k}\{x\} \in \mathcal{K}_p$ and $x' \in \mathcal{X}$, and outputs a classical string $y \in \mathcal{Y}$.

We require the following properties of pPRF.

Functionality preserved under puncturing. For all $\lambda \in \mathbb{N}$, for all $x \in \mathcal{X}$,

$$\Pr\left[\forall x' \in \mathcal{X} \setminus \{x\} : \mathsf{Eval}(\mathsf{k}\{x\}, x') = \mathsf{Eval}(\mathsf{k}, x') \;\middle|\; \begin{array}{r} \mathsf{k} \xleftarrow{\$} \mathsf{KeyGen}(1^\lambda) \\ \mathsf{k}\{x\} \xleftarrow{\$} \mathsf{Puncture}(\mathsf{k}, x) \end{array}\right] = 1.$$

Pseudorandomness at punctured points. For every QPT adversary $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$, and

every $\lambda \in \mathbb{N}$, the following holds:

$$\left| \Pr\left[1 \leftarrow \mathcal{A}_2(\mathsf{k}\{x^*\}, y, \tau) \;\middle|\; \begin{array}{r} (x^*, \tau) \leftarrow \mathcal{A}_1(1^\lambda, \tau) \\ \mathsf{k} \xleftarrow{\$} \mathsf{KeyGen}(1^\lambda) \\ \mathsf{k}\{x^*\} \xleftarrow{\$} \mathsf{Puncture}(\mathsf{k}, x^*) \\ y \leftarrow \mathsf{Eval}(\mathsf{k}, x^*) \end{array} \right] - \Pr\left[1 \leftarrow \mathcal{A}_2(\mathsf{k}\{x^*\}, y, \tau) \;\middle|\; \begin{array}{r} (x^*, \tau) \leftarrow \mathcal{A}_1(1^\lambda, \tau) \\ \mathsf{k} \xleftarrow{\$} \mathsf{KeyGen}(1^\lambda) \\ \mathsf{k}\{x^*\} \xleftarrow{\$} \mathsf{Puncture}(\mathsf{k}, x^*) \\ y \xleftarrow{\$} \mathcal{Y} \end{array} \right] \right| \leq \mathsf{negl}(\lambda),$$

where the probability is taken over the randomness of KeyGen, Puncture, and $\mathcal{A}_1$.

Denote the above probability as $\mathsf{Adv}^{\mathsf{pPRF}}(\lambda, \mathcal{A})$. We further say that pPRF is δ-secure, for some concrete negligible function $\delta(\lambda)$, if for all QPT adversaries $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathsf{pPRF}}(\lambda, \mathcal{A})$ is smaller than $\delta(\lambda)^{\Omega(1)}$.

We also consider several variants as follows.

Definition 2.8 — Invertible Pseudorandom Functions

An invertible pseudorandom function (IPRF) with key-space $\mathcal{K}$, domain $\mathcal{X}$, and range $\mathcal{Y}$ consists of two functions $\mathsf{iPRF} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ and $\mathsf{iPRF}^{-1} : \mathcal{K} \times \mathcal{Y} \to \mathcal{X} \cup \{\perp\}$. An IPRF can also include a setup algorithm $\mathsf{iPRF.Setup}(1^\lambda)$ that on input the security parameter λ, outputs a key $\mathsf{k} \in \mathcal{K}$. The functions iPRF and iPRF^{-1} satisfy the following properties:

- Both iPRF and iPRF^{-1} can be computed by deterministic polynomial-time algorithms.
- For all security parameters λ and all keys k output by $\mathsf{iPRF.Setup}(1^\lambda)$, the function $\mathsf{iPRF}(\mathsf{k}, \cdot)$ is an injective function from $\mathcal{X}$ to $\mathcal{Y}$. Moreover, the function $\mathsf{iPRF}^{-1}(\mathsf{k}, \cdot)$ is the (generalized) inverse of $\mathsf{iPRF}(\mathsf{k}, \cdot)$.
- **(Weak) Quantum Pseudorandomness.** An IPRF $\mathsf{iPRF} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is secure if for all QPT adversaries $\mathcal{A}$,

$$\left| \Pr_{\mathsf{k} \xleftarrow{\$} \mathsf{iPRF.Setup}(1^\lambda)} \left[\mathcal{A}^{|\mathsf{iPRF}(\mathsf{k}, \cdot)\rangle}(1^\lambda) = 1 \right] - \Pr_{R \xleftarrow{\$} \mathsf{InjFuncs}[\mathcal{X}, \mathcal{Y}]} \left[\mathcal{A}^{|R(\cdot)\rangle}(1^\lambda) = 1 \right] \right| \leq \mathsf{negl}(\lambda),$$

where $\mathsf{InjFuncs}[\mathcal{X}, \mathcal{Y}]$ is the set of all *injective* functions from $\mathcal{X}$ to $\mathcal{Y}$.

Definition 2.9 — Statistically Injective Pseudorandom Functions

A statistically injective (puncturable) PRF family with (negligible) failure probability $\varepsilon(\cdot)$ is a (puncturable) PRF family PRF such that with probability $1 - \varepsilon(\lambda)$ over the random choice of key $\mathsf{k} \leftarrow \mathsf{KeyGen}(1^\lambda)$, we have that $\mathsf{PRF}(\mathsf{k}, \cdot)$ is injective.

Definition 2.10 — Extracting Pseudorandom Functions

An extracting (puncturable) PRF with error $\varepsilon(\cdot)$ for min-entropy $k(\cdot)$ is a (puncturable) PRF PRF mapping $n(\lambda)$ bits to $m(\lambda)$ bits such that for all λ, if X is any distribution over $n(\lambda)$ bits with min-entropy greater than $k(\lambda)$, then the statistical distance between $(\mathsf{k}, \mathsf{PRF}(\mathsf{k}, X))$ and $(\mathsf{k}, r \xleftarrow{\$} \{0,1\}^{m(\lambda)})$ is at most $\varepsilon(\cdot)$, where $\mathsf{k} \leftarrow \mathsf{KeyGen}(1^\lambda)$.

2.3.2 Symmetric-key Encryption

Definition 2.11 — Symmetric-key Encryption

A symmetric-key cryptosystem $\mathcal{SE} := \langle \mathcal{K}, \mathsf{SymEnc}, \mathsf{SymDec} \rangle$ consists of three PPT algorithms.

$\mathcal{K}(1^\lambda)$ is a probabilistic key generation algorithm which takes as input a security parameter λ and outputs a secret key k.

$\mathsf{SymEnc}(\mathsf{k}, x; r)$ is a probabilistic encryption algorithm which takes as input a secret key k, a plaintext $x \in \mathcal{X}$ (where $\mathcal{X}$ is some fixed message space), a random coin $r \in \mathcal{R}$ (where $\mathcal{R}$ is the randomness space), and outputs a ciphertext y.

$\mathsf{SymDec}(\mathsf{k}, y)$ is a deterministic decryption algorithm which takes as input a secret key k and a ciphertext y, and outputs a message $x \in \mathcal{X} \cup \{\perp\}$, where $\perp$ is a distinguished symbol indicating decryption failure.

The standard correctness requirement is that for all $\lambda \in \mathbb{N}$, for any key $\mathsf{k} \leftarrow \mathcal{K}(1^\lambda)$, any random coin r of SymEnc and any $x \in \mathcal{X}$, we have $\mathsf{SymDec}(\mathsf{k}, \mathsf{SymEnc}(\mathsf{k}, x; r)) = x$.

Security Definitions. For completeness, we give here a modified version of the Real-or-Random security definition in the classical setting. In this notion, the security game starts with a first learning phase, followed by a challenge phase where $\mathcal{A}$ sends a challenge query (a message x to encrypt) and receives a challenge ciphertext, which is encryption of either x if $b = 1$ or some random message x' if $b = 0$. Note that encrypting a random message x' is equivalent to applying a random function h to x and then encrypting $h(x)$. Afterwards, a second learning phase follows, and finally, $\mathcal{A}$ outputs a solution (its guess for the bit b).

In the standard IND-CCA2 security definition, the decryption oracle in the second learning phase would return $\perp$ if the query is a challenge ciphertext (in both games). However, this is completely equivalent to returning the original plaintext (which was sent to the challenge oracle by the adversary) in both games. We note that the challenger could do that in the classical setting, as it could keep both the challenge plaintext and the challenge ciphertext. We formalize this modified notion below.

We let the string atk be instantiated by any of the formal symbols $cpa, cca1, cca2$, while ATK is the corresponding formal symbol from CPA, CCA1, CCA2. When we say $\mathcal{O}_i := \varnothing$ where $i \in \{1, 2\}$, we mean $\mathcal{O}_i$ is the function which, on any input, returns $\perp$. For a random function h, h^0 is identity, and $h^1 := h$.

Definition 2.12 — Real-or-Random IND-CPA, IND-CCA1, IND-CCA2

Let $\mathcal{SE} := \langle \mathcal{K}, \mathsf{SymEnc}, \mathsf{SymDec} \rangle$ be a symmetric-key encryption scheme and let $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$ be a classical adversary. Let $\mathcal{F}$ be the family of all functions over $\mathcal{X}$. For $atk \in [cpa, cca1, cca2]$, we define the following game, where the oracles $\mathcal{O}_1, \mathcal{O}_2$ are defined according to atk:

Experiment $\mathsf{Expt}^{ind\text{-}atk-b}_{\mathcal{SE}}(\lambda, \mathcal{A})$:

1 : $\mathsf{k} \xleftarrow{\$} \mathcal{K}(1^\lambda)$
2 : $(x, \mathsf{state}) \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{SymEnc}(\mathsf{k},\cdot)}, \mathcal{O}_1}(1^\lambda)$
3 : $h \xleftarrow{\$} \mathcal{F}$
4 : $y^* \leftarrow \mathsf{SymEnc}(\mathsf{k}, h^{1-b}(x))$
5 : $b' \leftarrow \mathcal{A}_2^{\mathcal{O}_{\mathsf{SymEnc}(\mathsf{k},\cdot)}, \mathcal{O}_2}(y^*, \mathsf{state})$
6 : **return** b'

atk	Oracle $\mathcal{O}_1$	Oracle $\mathcal{O}_2$
cpa	$\varnothing$	$\varnothing$
$cca1$	$\mathsf{SymDec}(\mathsf{k}, \cdot)$	$\varnothing$
$cca2$	$\mathsf{SymDec}(\mathsf{k}, \cdot)$	$\mathsf{SymDec}^*(\mathsf{k}, \cdot)$

Here, $\mathsf{SymDec}^*(\mathsf{k}, y)$ returns x if $y = y^*$, otherwise it decrypts normally.
We define $\mathcal{A}$'s advantage by

$$\mathsf{Adv}^{ind\text{-}atk}_{\mathcal{A},\mathcal{SE}}(\lambda) := \left|\Pr\left[\mathsf{Expt}^{ind\text{-}atk-1}_{\mathcal{SE}}(\lambda, \mathcal{A}) = 1\right] - \Pr\left[\mathsf{Expt}^{ind\text{-}atk-0}_{\mathcal{SE}}(\lambda, \mathcal{A}) = 1\right]\right|.$$

We say $\mathcal{SE}$ is secure in the sense of IND-ATK if $\mathcal{A}$ being PPT implies that $\mathsf{Adv}^{ind\text{-}atk}_{\mathcal{A},\mathcal{SE}}(\lambda)$ is negligible.

Next, we give the definition (in the Find-then-Guess style) in the quantum setting, proposed by Boneh and Zhandry [BZ13b]. In the following, we let the string $qatk$ be instantiated by any of the formal symbols $qcpa, qcca1, qcca2$, while qATK is the corresponding formal symbol from qCPA, qCCA1, qCCA2.

Definition 2.13 — IND-qCPA, IND-qCCA1, IND-qCCA2 [BZ13b]

Let $\mathcal{SE} := \langle \mathcal{K}, \mathsf{SymEnc}, \mathsf{SymDec} \rangle$ be a symmetric-key encryption scheme and let $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$ be a quantum adversary. For $qatk \in [qcpa, qcca1, qcca2]$, we define the following game, where the oracles $\mathcal{O}_1, \mathcal{O}_2$ are defined according to $qatk$:

Experiment $\mathrm{Expt}^{ind\text{-}qatk-b}_{\mathcal{SE}}(\lambda, \mathcal{A})$:

1 : $\mathsf{k} \xleftarrow{\$} \mathcal{K}(1^\lambda)$
2 : $|x_0, x_1\rangle\,|\phi\rangle \leftarrow \mathcal{A}_1^{|\mathcal{O}_{\mathsf{SymEnc}(\mathsf{k},\cdot)}\rangle, |\mathcal{O}_1\rangle}(1^\lambda)$
3 : **if** $|x_0| \neq |x_1|$ **then return** 0
4 : $y^* \leftarrow \mathsf{SymEnc}(\mathsf{k}, x_b)$
5 : $b' \leftarrow \mathcal{A}_2^{|\mathcal{O}_{\mathsf{SymEnc}(\mathsf{k},\cdot)}\rangle, |\mathcal{O}_2\rangle}(|y^*\rangle\,|\phi\rangle)$
6 : **return** b'

$qatk$	Oracle $\mathcal{O}_1$	Oracle $\mathcal{O}_2$
$qcpa$	$\varnothing$	$\varnothing$
$qcca1$	$\mathsf{SymDec}(\mathsf{k}, \cdot)$	$\varnothing$
$qcca2$	$\mathsf{SymDec}(\mathsf{k}, \cdot)$	$\mathsf{SymDec}^*(\mathsf{k}, \cdot)$

Here, $\mathsf{SymDec}^*(\mathsf{k}, y)$ returns $\perp$ if $y = y^*$, otherwise it decrypts normally.
We define $\mathcal{A}$'s advantage by

$$\mathsf{Adv}^{ind\text{-}qatk}_{\mathcal{A},\mathcal{SE}}(\lambda) := \left|\Pr\left[\mathsf{Expt}^{ind\text{-}qatk-1}_{\mathcal{SE}}(\lambda, \mathcal{A}) = 1\right] - \Pr\left[\mathsf{Expt}^{ind\text{-}qatk-0}_{\mathcal{SE}}(\lambda, \mathcal{A}) = 1\right]\right|.$$

We say $\mathcal{SE}$ is secure in the sense of IND-qATK if $\mathcal{A}$ being QPT implies that $\mathsf{Adv}^{ind\text{-}qatk}_{\mathcal{A},\mathcal{SE}}(\lambda)$ is negligible.

2.3.3 Public-key Encryption

Definition 2.14 — Public-key Encryption

A public-key cryptosystem $\mathcal{E} := \langle \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec} \rangle$ consists of three PPT algorithms.

$\mathsf{KeyGen}(1^\lambda)$ is a probabilistic key generation algorithm which takes as input the security parameter λ and outputs a pair $(\mathsf{pk}, \mathsf{sk})$ of matching public and secret keys.

$\mathsf{Enc}(\mathsf{pk}, x; r)$ is a probabilistic encryption algorithm which takes as input a public key pk, a plaintext $x \in \mathcal{X}$ (where $\mathcal{X}$ is some fixed message space), a random coin $r \in \mathcal{R}$ (where $\mathcal{R}$ is the randomness space), and outputs a ciphertext y.

$\mathsf{Dec}(\mathsf{sk}, y)$ is a deterministic decryption algorithm which takes as input a secret key sk and a ciphertext y, and outputs a message $x \in \mathcal{X} \cup \{\perp\}$, where $\perp$ is a distinguished symbol indicating decryption failure.

The following correctness definition is taken from [HHK17]. We call a public-key encryption scheme $\mathcal{E}$ δ-*correct* if

$$\mathbb{E}\left[\max_{x \in \mathcal{X}} \Pr_{r \in \mathcal{R}}\left[\,\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, x; r)) \neq x\,\right]\right] \leq \delta,$$

where the expectation is taken over $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$.

Security Definitions. Similar to the symmetric setting, we first give a Real-or-Random security definition for public-key encryption in the classical setting, then Boneh-Zhandry's definitions [BZ13b].

Definition 2.15 — Real-or-Random IND-CPA, IND-CCA1, IND-CCA2

Let $\mathcal{E} := \langle \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec} \rangle$ be a public-key encryption scheme and let $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$ be a classical adversary. Let $\mathcal{F}$ be the family of all functions over $\mathcal{X}$. For $atk \in [cpa, cca1, cca2]$, we define the following game, where the oracles $\mathcal{O}_1, \mathcal{O}_2$ are defined according to atk:

Experiment $\mathsf{Expt}_{\mathcal{E}}^{ind\text{-}atk-b}(\lambda, \mathcal{A})$:

1 : $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$
2 : $(x, \mathsf{state}) \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{Enc}(\mathsf{pk},\cdot)}, \mathcal{O}_1}(\mathsf{pk})$
3 : $h \xleftarrow{\$} \mathcal{F}$
4 : $y^* \leftarrow \mathsf{Enc}(\mathsf{pk}, h^{1-b}(x))$
5 : $b' \leftarrow \mathcal{A}_2^{\mathcal{O}_{\mathsf{Enc}(\mathsf{pk},\cdot)}, \mathcal{O}_2}(y^*, \mathsf{state})$
6 : **return** b'

atk	Oracle $\mathcal{O}_1$	Oracle $\mathcal{O}_2$
cpa	$\varnothing$	$\varnothing$
$cca1$	$\mathsf{Dec}(\mathsf{sk}, \cdot)$	$\varnothing$
$cca2$	$\mathsf{Dec}(\mathsf{sk}, \cdot)$	$\mathsf{Dec}^*(\mathsf{sk}, \cdot)$

Here, $\mathsf{Dec}^*(\mathsf{sk}, y)$ returns x if $y = y^*$, otherwise it decrypts normally.
We define $\mathcal{A}$'s advantage by

$$\mathsf{Adv}_{\mathcal{A},\mathcal{E}}^{ind\text{-}atk}(\lambda) := \left| \Pr\left[\mathsf{Expt}_{\mathcal{E}}^{ind\text{-}atk-1}(\lambda, \mathcal{A}) = 1\right] - \Pr\left[\mathsf{Expt}_{\mathcal{E}}^{ind\text{-}atk-0}(\lambda, \mathcal{A}) = 1\right] \right|.$$

We say $\mathcal{E}$ is secure in the sense of IND-ATK if $\mathcal{A}$ being PPT implies that $\mathsf{Adv}_{\mathcal{A},\mathcal{E}}^{ind\text{-}atk}(\lambda)$ is negligible.

Definition 2.16 — IND-qCPA, IND-qCCA1, IND-qCCA2 [BZ13b]

Let $\mathcal{E} := \langle \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec} \rangle$ be a public-key encryption scheme and let $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$ be a quantum adversary. For $qatk \in [qcpa, qcca1, qcca2]$, we define the following game, where the oracles $\mathcal{O}_1, \mathcal{O}_2$ are defined according to $qatk$:

Experiment $\mathsf{Expt}_{\mathcal{E}}^{ind\text{-}qatk-b}(\lambda, \mathcal{A})$:

1 : $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$
2 : $|x_0, x_1\rangle |\phi\rangle \leftarrow \mathcal{A}_1^{|\mathcal{O}_1\rangle}(\mathsf{pk})$
3 : **if** $|x_0| \neq |x_1|$ **then return** 0
4 : $y^* \leftarrow \mathsf{Enc}(\mathsf{pk}, x_b)$
5 : $b' \leftarrow \mathcal{A}_2^{|\mathcal{O}_2\rangle}(|y^*\rangle |\phi\rangle)$
6 : **return** b'

$qatk$	Oracle $\mathcal{O}_1$	Oracle $\mathcal{O}_2$
$qcpa$	$\varnothing$	$\varnothing$
$qcca1$	$\mathsf{Dec}(\mathsf{sk}, \cdot)$	$\varnothing$
$qcca2$	$\mathsf{Dec}(\mathsf{sk}, \cdot)$	$\mathsf{Dec}^*(\mathsf{sk}, \cdot)$

Here, $\mathsf{Dec}^*(\mathsf{sk}, y)$ returns $\perp$ if $y = y^*$, otherwise it decrypts normally.
We define $\mathcal{A}$'s advantage by

$$\mathsf{Adv}_{\mathcal{A},\mathcal{E}}^{ind\text{-}qatk}(\lambda) := \left| \Pr\left[\mathsf{Expt}_{\mathcal{E}}^{ind\text{-}qatk-1}(\lambda, \mathcal{A}) = 1\right] - \Pr\left[\mathsf{Expt}_{\mathcal{E}}^{ind\text{-}qatk-0}(\lambda, \mathcal{A}) = 1\right] \right|.$$

We say $\mathcal{E}$ is secure in the sense of IND-qATK if $\mathcal{A}$ being QPT implies that $\mathsf{Adv}_{\mathcal{A},\mathcal{E}}^{ind\text{-}qatk}(\lambda)$ is negligible.

2.3.4 One-time Signatures

Definition 2.17 — Digital Signatures

A signature scheme $\mathsf{Sig} := \langle \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verif} \rangle$ consists of three PPT algorithms .

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$ is a randomized procedure that takes as input the security parameter λ and produces a secret key and public key pair $(\mathsf{pk}, \mathsf{sk})$.

$\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, m)$ takes as input the secret key and a message $m \in \mathcal{X}$ (where $\mathcal{X}$ is some fixed message space), and produces a signature σ.

$b \leftarrow \mathsf{Verif}(\mathsf{pk}, m, \sigma)$ is a deterministic decryption algorithm which takes as input a public key pk, a message m, and a supposed signature σ on m, and outputs a bit b.

A signature scheme is correct if Verif accepts signatures outputted by Sign such that

$$\Pr\left[\mathsf{Verif}(\mathsf{pk}, m, \sigma) = 1 \;\middle|\; \begin{array}{r} (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, m) \end{array} \right] \geq 1 - \mathsf{negl}(\lambda).$$

For security, we will for simplicity only consider one-time signature schemes where the adversary only receives a single superposition of messages. Furthermore, for simplicity we assume that the signing function is a deterministic function of the secret key and message; this can be made without loss of generality by using a pseudorandom function to generate the randomness.

Boneh-Zhandry security. Boneh and Zhandry [BZ13b] give the following definition of security for signatures in the presence of quantum adversaries.

Definition 2.18 — (n+1)-Unforgeability [BZ13b]

Let $\mathcal{A}$ be a quantum adversary, and consider the following experiment between $\mathcal{A}$ and a challenger:

- The challenger runs $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$, and gives pk to $\mathcal{A}$.
- $\mathcal{A}$ makes a quantum superposition query to the function $\mathsf{Sign}(\mathsf{sk}, \cdot)$ as $|m, u\rangle \mapsto |m, u \oplus \mathsf{Sign}(\mathsf{sk}, m)\rangle$.
- $\mathcal{A}$ outputs two classical message/signature pairs $((m_0, \sigma_0), (m_1, \sigma_1))$.
- The challenger accepts and outputs 1 if and only if (1) $m_0 \neq m_1$, and (2) $\mathsf{Verif}(\mathsf{pk}, m_b, \sigma_b)$ for both $b \in \{0, 1\}$. Denote this output by $\text{W-BZ-Exp}(\lambda, \mathcal{A})$.

A signature scheme is one-time weakly BZ-secure if, for any quantum polynomial time adversary $\mathcal{A}$, $\text{W-BZ-Exp}(\lambda, \mathcal{A})$ is negligible.

2.3.5 Non-interactive Zero-knowledge Proof Systems

For a NP relation $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{W}$, we let $\mathcal{L}(\mathcal{R}) := \{x : \exists w, (x, w) \in \mathcal{R}\}$.

Definition 2.19 — Non-interactive Zero-knowledge Proof Systems

A non-interactive zero-knowledge (NIZK) proof system for an NP relation $\mathcal{R}$ in the common reference string (CRS) model consists of three PPT algorithms $\mathsf{NIZK} := \langle \mathsf{Setup}, \mathcal{P}, \mathcal{V} \rangle$:

$\mathsf{crs} \leftarrow \mathsf{Setup}(1^n, 1^\lambda)$. On input a statement of length n and the security parameter λ, the setup algorithm Setup outputs a common reference string crs.

$\pi \leftarrow \mathcal{P}(\mathsf{crs}, x, w)$. On input the common reference string crs, an instance x and a witness w such that $(x, w) \in \mathcal{R}$, the proving algorithm $\mathcal{P}$ outputs a proof π.

$b \leftarrow \mathcal{V}(\mathsf{crs}, x, \pi)$. On input the common reference string crs, an instance x and a proof π, the verification algorithm $\mathcal{V}$ outputs a bit $b \in \{0, 1\}$. If $b = 1$, we say that $\mathcal{V}$ accepts, otherwise we say that $\mathcal{V}$ rejects.

The proof system NIZK must satisfy the following requirements for all $\lambda \in \mathbb{N}$.

Completeness. For every $(x, w) \in \mathcal{R}$, we have that

$$\Pr\left[\mathcal{V}(\mathsf{crs}, x, \mathcal{P}(\mathsf{crs}, x, w)) = 1 \mid \mathsf{crs} \leftarrow \mathsf{Setup}(1^{|x|}, 1^\lambda)\right] = 1,$$

where the probability is taken over the randomness of Setup and $\mathcal{P}$.

Statistical soundness. There exists a negligible function $\mathsf{negl}(\lambda)$ such that for any $n \in \mathbb{N}$,

$$\Pr_{\mathsf{crs} \leftarrow \mathsf{Setup}(1^n, 1^\lambda)}\left[\exists (x, \pi^*) \text{ s.t. } \mathcal{V}(\mathsf{crs}, x, \pi^*) = 1 \wedge x \notin \mathcal{L}\right] \leq \mathsf{negl}(\lambda).$$

(Adaptive) post-quantum computational zero-knowledge. There exists a QPT simulator $\mathcal{S} := (\mathcal{S}_1, \mathcal{S}_2)$ such that for any QPT malicious verifier $\mathcal{V}^* := (\mathcal{V}_1^*, \mathcal{V}_2^*)$, for any $n \in \mathbb{N}$,

$$\left| \Pr\left[\mathcal{V}_2^*(\mathsf{crs}, x, \pi, \zeta) = 1 \wedge x \in \mathcal{L} \;\middle|\; \begin{array}{r} \mathsf{crs} \leftarrow \mathsf{Setup}(1^n, 1^\lambda) \\ (x, w, \zeta) \leftarrow \mathcal{V}_1^*(\mathsf{crs}) \\ \pi \leftarrow \mathcal{P}(\mathsf{crs}, x, w) \end{array} \right] - \Pr\left[\mathcal{V}_2^*(\mathsf{crs}, x, \pi, \zeta) = 1 \wedge x \in \mathcal{L} \;\middle|\; \begin{array}{r} (\mathsf{crs}, \mathsf{td}) \leftarrow \mathcal{S}_1(1^\lambda) \\ (x, w, \zeta) \leftarrow \mathcal{V}_1^*(\mathsf{crs}) \\ \pi \leftarrow \mathcal{S}_2(\mathsf{td}, x) \end{array} \right] \right| \leq \mathsf{negl}(\lambda).$$

Definition 2.20 — Unbounded Simulation-Soundness

A zero-knowledge proof system is said to be (unbounded) *simulation-sound* if it has the property that an adversary cannot provide a convincing proof for any false statement, even if it has seen *simulated proofs* of arbitrary statements (including false statements). More precisely, an NIZK proof is simulation sound if for all QPT

adversaries $\mathcal{A}$, we have:

$$\Pr\left[\begin{array}{c|c} (x_i,\pi_i)\notin Q \wedge\ x\notin\mathcal{L} & (\mathsf{crs},\mathsf{td})\leftarrow\mathcal{S}_1(1^\lambda) \\ \wedge\ \mathcal{V}(\mathsf{crs},x,\pi)=1 & (x,\pi)\leftarrow\mathcal{A}^{\mathcal{S}_2(\mathsf{td},\cdot)}(\mathsf{crs}) \end{array}\right]\leq \mathsf{negl}(\lambda),$$

where Q is the list of simulation queries and responses (x_i,π_i).

Definition 2.21 — NIZKs in the (quantum) random oracle model

A definition for NIZK proof systems in the (quantum) random oracle model can be defined similarly as in Definition 2.19, except that the setup algorithm Setup outputs an empty string, and all parties have (quantum) access to a random oracle $\mathcal{O}_H$. Completeness, soundness, zero-knowledge and simulation-soundness can be defined similarly with respect to this change in the model.

2.3.6 Indistinguishability Obfuscation

Definition 2.22 — Indistinguishability Obfuscator [BGIR+01]

A uniform PPT machine $i\mathcal{O}$ is called an indistinguishability obfuscator for a classical circuit class $\{\mathcal{C}_\lambda\}$ if the following conditions are satisfied:

- For all security parameters $\lambda\in\mathbb{N}$, for all $C\in\mathcal{C}_\lambda$, for all input x, we have that
$$\Pr[C'(x)=C(x)\mid C'\leftarrow i\mathcal{O}(\lambda,C)]=1.$$
- For any (not necessarily uniform) QPT distinguisher $\mathcal{D}$, for all security parameters $\lambda\in\mathbb{N}$, for all pairs of circuits $C_0,C_1\in\mathcal{C}_\lambda$, we have that if $C_0(x)=C_1(x)$ for all inputs x, then
$$\mathsf{Adv}^{i\mathcal{O}}(\lambda,\mathcal{A}):=|\Pr[\mathcal{D}(i\mathcal{O}(\lambda,C_0))=1]-\Pr[\mathcal{D}(i\mathcal{O}(\lambda,C_1))=1]|\leq\mathsf{negl}(\lambda).$$

We further say that $i\mathcal{O}$ is δ-secure, for some concrete negligible function $\delta(\lambda)$, if for all QPT adversaries $\mathcal{A}$, the advantage $\mathsf{Adv}^{i\mathcal{O}}(\lambda,\mathcal{A})$ is smaller than $\delta(\lambda)^{\Omega(1)}$.

Quantum-secure instantiations. There has been recent progress in constructing quantum-secure indistinguishability obfuscation schemes [BDGM20; GP20] from cryptographic assumptions that conjecturally hold against quantum adversaries.

In [Zha19b; Shm22a], it is shown that indistinguishability obfuscation schemes have the property of *subspace hiding*.

Lemma 2.8 ([Zha19b; Shm22a]). *Let iO an indistinguishability obfuscation scheme, and assume that injective one-way functions exist. Let $S:=\{S_\lambda\}_{\lambda\in\mathbb{N}}$ a subspace $S\subseteq\mathbb{F}_2^\lambda$. For a subspace S', denote by $C_{S'}$ a classical circuit that checks membership in S'. Then, for every constant $\delta\in(0,1]$ we have the following indistinguishability:*

$$\{i\mathcal{O}(C_{S_\lambda})\}_{\lambda\in\mathbb{N}}\overset{c}{\approx}_0\{i\mathcal{O}(C_T)\mid T\xleftarrow{\$}\mathcal{S}_{S_\lambda}\}_{\lambda\in\mathbb{N}},$$

where $\mathcal{S}_{S_\lambda}$ is the set of all subspaces of dimension $\lambda-\lambda^\delta$ that contain S_λ.

2.3.7 Leveled Hybrid Quantum Fully Homomorphic Encryption

We give a definition of quantum fully homomorphic encryption of a specific structure, which was defined in [Shm22a].

Definition 2.23 — Leveled Hybrid Quantum Fully Homomorphic Encryption

A hybrid leveled quantum fully homomorphic encryption scheme is given by $\mathsf{QFHE} := \langle \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{QOTP}, \mathsf{Eval}, \mathsf{Decrypt} \rangle$ with the following syntax:

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda, 1^\ell)$. A PPT algorithm that given a security parameter $\lambda \in \mathbb{N}$ and target circuit bound $\ell \in \mathbb{N}$, outputs a classical key pair $(\mathsf{pk}, \mathsf{sk})$.

$|\psi\rangle^{(x,z)} \leftarrow \mathsf{QOTP}((x,z), |\psi\rangle)$. A QPT algorithm that takes as input an n-qubit quantum state $|\psi\rangle$ and classical strings as quantum one-time pads (QOTPs) $x, z \in \{0,1\}^n$ and outputs its QOTP transformation $|\psi\rangle^{(x,z)} := (\otimes_{i\in[\![n]\!]} \mathsf{Z}^{z_i}) \cdot (\otimes_{i\in[\![n]\!]} \mathsf{X}^{x_i}) |\psi\rangle$. We often call these one-time pads (x,z) the Pauli keys. Furthermore, if $|\psi\rangle$ is a *classical* string m, we ignore the Pauli key z and write $\mathsf{QOTP}(x,m)$ whose output is $x \oplus m$.

$\mathsf{ct} \leftarrow \mathsf{Encrypt}(\mathsf{pk}, x)$. A PPT algorithm that takes as input a classical string $x \in \{0,1\}^*$ and the public key pk and outputs a classical ciphertext ct.

$x \leftarrow \mathsf{Decrypt}(\mathsf{sk}, \mathsf{ct})$. A PPT algorithm that takes as input a classical ciphertext ct and the secret key sk and outputs a classical string x.

$(|\phi\rangle^{(x',z')}, \mathsf{ct}_{x',z'}) \leftarrow \mathsf{Eval}(\mathsf{pk}, (|\psi\rangle^{(x,z)}, \mathsf{ct}_{x,z}), C)$. A QPT algorithm that takes as input a general quantum circuit C, a quantum one-time pad encrypted state $|\psi\rangle^{(x,z)}$ and a classical ciphertext $\mathsf{ct}_{x,z}$ of the pads. The evaluation outputs a QOTP encryption of some quantum state $|\phi\rangle$ encrypted under new keys (x',z') and a classical ciphertext $\mathsf{ct}_{x',z'}$.

The scheme satisfies the following properties.

Semantic Security. For every polynomials $m(\cdot)$, $\ell(\cdot)$, and QPT algorithm $\mathcal{A} := \{\mathcal{A}_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$ there exists a negligible function $\mathsf{negl}(\cdot)$ such that

$$\left| \Pr\left[1 \leftarrow \mathcal{A}_2(m_0 \oplus x, \mathsf{ct}_x) \;\middle|\; \begin{array}{r} (m_0,m_1) \leftarrow \mathcal{A}_1(1^\lambda) \\ (\mathsf{pk},\mathsf{sk}) \xleftarrow{\$} \mathsf{KeyGen}(1^\lambda, 1^{\ell(\lambda)}) \\ x \xleftarrow{\$} \{0,1\}^{m(\lambda)} \\ \mathsf{ct}_x \leftarrow \mathsf{Encrypt}(\mathsf{pk}, x) \end{array} \right] - \Pr\left[1 \leftarrow \mathcal{A}_2(m_1 \oplus x, \mathsf{ct}_x) \;\middle|\; \begin{array}{r} (m_0,m_1) \leftarrow \mathcal{A}_1(1^\lambda) \\ (\mathsf{pk},\mathsf{sk}) \xleftarrow{\$} \mathsf{KeyGen}(1^\lambda, 1^{\ell(\lambda)}) \\ x \xleftarrow{\$} \{0,1\}^{m(\lambda)} \\ \mathsf{ct}_x \leftarrow \mathsf{Encrypt}(\mathsf{pk}, x) \end{array} \right] \right| \leq \frac{1}{2} + \mathsf{negl}(\lambda),$$

where $\lambda \in \mathbb{N}$ and $m_0, m_1 \in \{0,1\}^{m(\lambda)}$.

Denote the above probability as $\mathsf{Adv}^{\mathsf{QFHE}}(\lambda, \mathcal{A})$. We further say that QFHE is δ-secure, for some concrete negligible function $\delta(\lambda)$, if for all QPT adversaries $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathsf{QFHE}}(\lambda, \mathcal{A})$ is smaller than $\delta(\lambda)^{\Omega(1)}$.

Homomorphism. For every polynomial $\ell := \ell(\lambda)$ there is a negligible function $\mathsf{negl}(\cdot)$ such that the following holds. Let $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda, 1^\ell)$, let x, z equal-length strings, let $\mathsf{ct}_{x,z} \leftarrow \mathsf{Encrypt}(\mathsf{pk}, (x, z))$, let C a quantum circuit of size $\leq \ell$, let $|\psi\rangle$ a $|x|$-qubit state input for C. Then, $\Delta(D_0, D_1) \leq \mathsf{negl}(\lambda)$, where D_0, D_1 are defined as follows.

- D_0: The output state is $|\phi\rangle \leftarrow C(|\psi\rangle)$.
- D_1: The output state generated by first evaluating

$$(|\phi\rangle^{(x',z')}, \mathsf{ct}_{x',z'}) \leftarrow \mathsf{Eval}\left(\mathsf{pk}, (|\psi\rangle^{(x,z)}, \mathsf{ct}_{x,z}), C\right),$$

and then decrypting

$$(x', z') \leftarrow \mathsf{Decrypt}(\mathsf{sk}, \mathsf{ct}_{x',z'}), \text{ and } |\phi\rangle \leftarrow \mathsf{QOTP}\left((x', z'), |\phi\rangle^{(x',z')}\right).$$

Quantum-secure instantiations. Quantum leveled fully-homomorphic encryption with the hybrid structure follows from the work of Mahadev [Mah18a] and Brakerski [Bra18], and can be based on the quantum hardness of Learning with Errors [Reg05]. Consequently, constructing QFHE that has hybrid structure, leveled, and has sub-exponential advantage security can be based on assuming LWE with sub-exponential indistinguishability.

2.3.8 Extended Trapdoor Claw-free Functions

In this section, we recall the definition of extended noisy trapdoor claw-free function family (ENTCF family), which was introduced in [Mah18b]. An ENTCF family consists of two families $\mathcal{F}$ and $\mathcal{G}$ of function pairs. A function pair $(f_{k,0}, f_{k,1}) \in \mathcal{F}$ is called a *claw-free pair* and is indexed by a public key k. Similarly, an *injective pair* is a pair of functions $(f_{k,0}, f_{k,1}) \in \mathcal{G}$, also indexed by a public key k. Informally, the most important properties are the following:

1. For fixed $k \in \mathcal{K}_1$, $f_{k,0}$ and $f_{k,1}$ are bijections with the same image, i.e., for every y in their image there exists a unique pair (x_0, x_1), called a *claw*, such that $f_{k,0}(x_0) = f_{k,1}(x_1) = y$.
2. Given a key $k \in \mathcal{K}_1$ for a claw-free pair, it is quantum-computationally intractable (without access to trapdoor information) to compute both a preimage x_i and a single generalized bit of $x_0 \oplus x_1$ (i.e., $d \cdot (x_0 \oplus x_1)$ for any non-trivial bit string d), where (x_0, x_1) forms a valid claw. This is called the *adaptive hardcore bit property*.
3. For fixed $k \in \mathcal{K}_0$, $f_{k,0}$ and $f_{k,1}$ are injective functions with disjoint images.
4. Given a key $k \in \mathcal{K}_1 \cup \mathcal{K}_0$, it is quantum-computationally hard (without access to trapdoor information) to determine the "function type", i.e., to decide whether k is a key for a claw-free or an injective pair. This is called *injective invariance*.

5. For every key $k \in \mathcal{K}_1 \cup \mathcal{K}_0$, there exists a trapdoor t_k, which can be sampled together with k and with which (ii) and (iv) are computationally easy.

A formal definition follows.

Definition 2.24 — Noisy Trapdoor Claw-Free Functions

Let λ be a security parameter. Let $\mathcal{X}$ and $\mathcal{Y}$ be finite sets. Let $\mathcal{K}_1$ be a finite set of keys. A family of functions

$$\mathcal{F} := \left\{ f_{k,b} : \mathcal{X} \to \mathcal{D}_{\mathcal{Y}} \right\}_{k \in \mathcal{K}_1, b \in \{0,1\}}$$

is called a *noisy trapdoor claw-free (NTCF) family* if the following conditions hold:

Efficient Function Generation. There exists an efficient probabilistic algorithm $\mathsf{Gen}_{\mathcal{F}}$ which generates a key $k \in \mathcal{K}_1$ together with a trapdoor t_k:

$$(k, t_k) \leftarrow \mathsf{Gen}_{\mathcal{F}}(1^\lambda) \ .$$

Trapdoor Injective Pair. For all keys $k \in \mathcal{K}_1$ the following conditions hold.

1. *Trapdoor*: For all $b \in \{0,1\}$ and $x \neq x' \in \mathcal{X}$, $\mathrm{Supp}(f_{k,b}(x)) \cap \mathrm{Supp}(f_{k,b}(x')) = \emptyset$. Moreover, there exists an efficient deterministic algorithm $\mathsf{Inv}_{\mathcal{F}}$ such that for all $b \in \{0,1\}$, $x \in \mathcal{X}$ and $y \in \mathrm{Supp}(f_{k,b}(x))$, $\mathsf{Inv}_{\mathcal{F}}(t_k, b, y) = x$.
2. *Injective pair*: There exists a perfect matching $\mathcal{R}_k \subseteq \mathcal{X} \times \mathcal{X}$ such that $f_{k,0}(x_0) = f_{k,1}(x_1)$ if and only if $(x_0, x_1) \in \mathcal{R}_k$.

Efficient Range Superposition. For all keys $k \in \mathcal{K}_1$ and $b \in \{0,1\}$ there exists a function $f'_{k,b} : \mathcal{X} \mapsto \mathcal{D}_{\mathcal{Y}}$ such that

1. For all $(x_0, x_1) \in \mathcal{R}_k$ and $y \in \mathrm{Supp}\Big(f'_{k,b}(x_b)\Big)$, $\mathsf{Inv}_{\mathcal{F}}(t_k, b, y) = x_b$ and $\mathsf{Inv}_{\mathcal{F}}(t_k, b \oplus 1, y) = x_{b \oplus 1}$.
2. There exists an efficient deterministic procedure $\mathsf{Chk}_{\mathcal{F}}$ that, on input k, $b \in \{0,1\}$, $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, returns 1 if $y \in \mathrm{Supp}\Big(f'_{k,b}(x)\Big)$ and 0 otherwise. Note that $\mathsf{Chk}_{\mathcal{F}}$ is not provided the trapdoor t_k.
3. For every k and $b \in \{0,1\}$,

$$\mathop{\mathbb{E}}_{x \xleftarrow{\$} \mathcal{X}} \left[H^2(f_{k,b}(x),\ f'_{k,b}(x)) \right] \leq \mu(\lambda) \ ,$$

for some negligible function $\mu(\cdot)$. Here H^2 is the Hellinger distance defined between two densities f_1 and f_2 over the same finite domain $\mathcal{X}$ as:

$$H^2(f_1, f_2) := 1 - \sum_{x \in \mathcal{X}} \sqrt{f_1(x) f_2(x)}.$$

Moreover, there exists an efficient procedure $\mathsf{Samp}_{\mathcal{F}}$ that on input k and $b \in \{0,1\}$ prepares the state

$$\frac{1}{\sqrt{|\mathcal{X}|}} \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \sqrt{(f'_{k,b}(x))(y)}\, |x\rangle\, |y\rangle \ . \tag{2.11}$$

Adaptive Hardcore Bit. For all keys $k \in \mathcal{K}_1$ the following conditions hold, for some integer w that is a polynomially bounded function of λ.

1. For all $b \in \{0,1\}$ and $x \in \mathcal{X}$, there exists a set $G_{k,b,x} \subseteq \{0,1\}^w$ such that $\Pr_{d \xleftarrow{\$} \{0,1\}^w}[\, d \notin G_{k,b,x}\,]$ is negligible, and moreover there exists an efficient algorithm that checks for membership in $G_{k,b,x}$ given k, b, x and the trapdoor t_k.
2. There is an efficiently computable injection $J : \mathcal{X} \to \{0,1\}^w$, such that J can be inverted efficiently on its range, and such that the following holds. If

$$\begin{aligned} H_k &:= \Big\{(b, x_b, d, d \cdot (J(x_0) \oplus J(x_1))) \mid b \in \{0,1\},\ (x_0, x_1) \in \mathcal{R}_k, \\ &\qquad\qquad d \in G_{k,0,x_0} \cap G_{k,1,x_1}\Big\}\ , \\ \overline{H}_k &:= \big\{(b, x_b, d, c) \mid (b, x, d, c \oplus 1) \in H_k\big\}\ , \end{aligned}$$

then for any quantum polynomial-time procedure $\mathcal{A}$ there exists a negligible function $\mu(\cdot)$ such that

$$\left| \Pr_{(k,t_k) \leftarrow \mathsf{Gen}_{\mathcal{F}}(1^\lambda)}[\mathcal{A}(k) \in H_k] - \Pr_{(k,t_k) \leftarrow \mathsf{Gen}_{\mathcal{F}}(1^\lambda)}[\mathcal{A}(k) \in \overline{H}_k] \right| \leq \mu(\lambda)\ . \tag{2.12}$$

Definition 2.25 — Trapdoor Injective Function Family

Let λ be a security parameter. Let $\mathcal{X}$ and $\mathcal{Y}$ be finite sets. Let $\mathcal{K}_0$ be a finite set of keys. A family of functions

$$\mathcal{G} := \Big\{g_{k,b} : \mathcal{X} \to \mathcal{D}_{\mathcal{Y}}\Big\}_{b \in \{0,1\}, k \in \mathcal{K}_0}$$

is called a *trapdoor injective family* if the following conditions hold:

Efficient Function Generation. There exists an efficient probabilistic algorithm $\mathsf{Gen}_{\mathcal{G}}$ which generates a key $k \in \mathcal{K}_0$ together with a trapdoor t_k:

$$(k, t_k) \leftarrow \mathsf{Gen}_{\mathcal{G}}(1^\lambda)\ .$$

Disjoint Trapdoor Injective Pair. For all keys $k \in \mathcal{K}_0$, for all $b, b' \in \{0,1\}$ and $x, x' \in \mathcal{X}$, if $(b, x) \neq (b', x')$, $\mathrm{Supp}(g_{k,b}(x)) \cap \mathrm{Supp}(g_{k,b'}(x')) = \emptyset$. Moreover, there exists an efficient deterministic algorithm $\mathsf{Inv}_{\mathcal{F}}$ such that for all $b \in \{0,1\}$, $x \in \mathcal{X}$ and $y \in \mathrm{Supp}(g_{k,b}(x))$, $\mathsf{Inv}_{\mathcal{G}}(t_k, y) = (b, x)$.

Efficient Range Superposition. For all keys $k \in \mathcal{K}_0$ and $b \in \{0,1\}$

1. There exists an efficient deterministic procedure $\mathsf{Chk}_{\mathcal{G}}$ that, on input k, $b \in \{0,1\}$, $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, outputs 1 if $y \in \mathrm{Supp}(g_{k,b}(x))$ and 0 otherwise. Note that $\mathsf{Chk}_{\mathcal{G}}$ is not provided the trapdoor t_k.
2. There exists an efficient procedure $\mathsf{Samp}_{\mathcal{G}}$ that on input k and $b \in \{0,1\}$ returns the state

$$\frac{1}{\sqrt{|\mathcal{X}|}} \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \sqrt{(g_{k,b}(x))(y)}\, |x\rangle\, |y\rangle \ . \tag{2.13}$$

Definition 2.26 — Injective Invariance

A noisy trapdoor claw-free family $\mathcal{F}$ is *injective invariant* if there exists a trapdoor injective family $\mathcal{G}$ such that:

1. The algorithms $\mathsf{Chk}_{\mathcal{F}}$ and $\mathsf{Samp}_{\mathcal{F}}$ are the same as the algorithms $\mathsf{Chk}_{\mathcal{G}}$ and $\mathsf{Samp}_{\mathcal{G}}$.
2. For all quantum polynomial-time procedures $\mathcal{A}$, there exists a negligible function $\mu(\cdot)$ such that

$$\left| \Pr_{(k,t_k) \leftarrow \mathsf{Gen}_{\mathcal{F}}(1^\lambda)} [\mathcal{A}(k) = 0] - \Pr_{(k,t_k) \leftarrow \mathsf{Gen}_{\mathcal{G}}(1^\lambda)} [\mathcal{A}(k) = 0] \right| \leq \mu(\lambda) \tag{2.14}$$

Definition 2.27 — Extended Trapdoor Claw-Free Family

A noisy trapdoor claw-free family $\mathcal{F}$ is an *extended trapdoor claw-free family* if:

1. It is injective invariant.
2. For all $k \in \mathcal{K}_1$ and $d \in \{0,1\}^w$, let:

$$H'_{k,d} \quad = \quad \{d \cdot (J(x_0) \oplus J(x_1)) | (x_0, x_1) \in \mathcal{R}_k\} \tag{2.15}$$

For all quantum polynomial-time procedures $\mathcal{A}$, there exists a negligible function $\mu(\cdot)$ and a string $d \in \{0,1\}^w$ such that

$$\left| \Pr_{(k,t_k) \leftarrow \mathsf{Gen}_{\mathcal{F}}(1^\lambda)} [\mathcal{A}(k) \in H'_{k,d}] - \frac{1}{2} \right| \leq \mu(\lambda) \tag{2.16}$$

In addition, we also define the following functions for convenience:

Definition 2.28 — Decoding Maps

1. For a key $k \in \mathcal{K}_0$ and a $y \in \mathcal{Y}$, we define $\hat{b}(k,y)$ by the condition $y \in \cup_x \mathrm{Supp}\left(f_{k,\hat{b}(k,y)}(x)\right)$. (This is well-defined because $f_{k,0}$ and $f_{k,1}$ form an injective pair.)

2. For a key $k \in \mathcal{K}_0 \cup \mathcal{K}_1$ and a $y \in \mathcal{Y}$, we define $\hat{x}_b(k,y)$ by the condition $y \in \mathrm{Supp}(f_{k,b}(\hat{x}_b(k,y)))$, and $\hat{x}_b(k,y) = \perp$ if $y \notin \cup_x \mathrm{Supp}(f_{k,b}(x))$. For $k \in \mathcal{K}_0$, we also use the shorthand $\hat{x}(k,y) \coloneqq \hat{x}_{\hat{b}(k,y)}(k,y)$.

3. For a key $k \in \mathcal{K}_1$, a $y \in \mathcal{Y}$, and a $d \in \{0,1\}^w$, we define $\hat{u}(k,y,d)$ by the condition $d \cdot (\hat{x}_0(k,y) \oplus \hat{x}_1(k,y)) = \hat{u}(k,y,d)$.

The above decoding maps applied to vector inputs are understood to act in an element-wise fashion. For example, for $\vec{k} \in \mathcal{K}_1^{\times n}, \vec{y} \in \mathcal{Y}^{\times n}$, and $\vec{d} \in \{0,1\}^{w \times n}$, we denote by $\hat{u}(\vec{k},\vec{y},\vec{d}) \in \{0,1\}^n$ the string defined by $\left(\hat{u}(\vec{k},\vec{y},\vec{d})\right)_i \coloneqq \hat{u}(k_i,y_i,d_i)$.

2.3.9 Copy-Protection

In the following, we assume that $\mathcal{F}$ is a family of functions such that each function f in the family has the same domain $\mathcal{X}$ and the same codomain $\mathcal{Y}$ and has a classical description d_f (of size polynomial in λ) that allows for an efficient computation of f.

Definition 2.29 — Copy-Protection Scheme of a Family $\mathcal{F}$

A copy-protection scheme is a tuple of algorithms $\langle \mathsf{Protect}, \mathsf{Eval} \rangle$ with the following properties:

$\rho_f \leftarrow \mathsf{Protect}(1^\lambda, d_f)$. On input the description d_f of a function $f \in \mathcal{F}$, the quantum protection algorithm outputs a quantum state ρ_f.

$y \leftarrow \mathsf{Eval}(1^\lambda, \rho_f, x)$. On input a quantum state ρ_f and an input $x \in \mathcal{X}$, the quantum evaluation algorithm outputs an image $y \in \mathcal{Y}$.

We ask a copy-protection scheme to have *correctness* and *anti-piracy* security.

A copy-protection scheme has *correctness* if the quantum protection of a function f computes f on every x with overwhelming probability, that is for any $\lambda \in \mathbb{N}$, for all $f \in \mathcal{F}$ and for all $x \in \mathcal{X}$, the following holds:

$$\Pr\left[\mathsf{Eval}(1^\lambda, \rho_f, x) = f(x) \mid \rho_f \leftarrow \mathsf{Protect}(1^\lambda, d_f)\right] \geq 1 - \mathsf{negl}(\lambda).$$

Definition 2.30 — Anti-Piracy Game for Copy-Protection

In order to define *anti-piracy security*, we define a piracy game for copy-protection of a family $\mathcal{F}$ with respect to distributions D_f and $\{X_f\}_{f \in \mathcal{F}}$. This game is between a challenger and an adversary represented by three algorithms $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$, where $\mathcal{A}_1$ and $\mathcal{A}_2$ cannot communicate during the game. It is parameterized by a distribution D_f over $\{d_f : f \in \mathcal{F}\}$ and a family of distributions $\{X_f\}_{f \in \mathcal{F}}$ over $\mathcal{X} \times \mathcal{X}$.

The challenger and the adversary proceed in the following way:

(1) Setup phase. The challenger samples $d_f \leftarrow D_f$ and sends $\rho_f \leftarrow \mathsf{Protect}(1^\lambda, d_f)$ to $\mathcal{A}_0$.

(2) Splitting phase. $\mathcal{A}_0$ prepares a bipartite quantum state σ_{12}, and sends σ_1 to $\mathcal{A}_1$ and σ_2 to $\mathcal{A}_2$.

(3) Challenge phase. The challenger samples $(x_1, x_2) \leftarrow X_f$ and sends x_1 to $\mathcal{A}_1$ and x_2 to $\mathcal{A}_2$.

(4) Answer phase. $\mathcal{A}_1$ returns y_1 and $\mathcal{A}_2$ returns y_2.

The adversary wins the game if $y_1 = f(x_1)$ and $y_2 = f(x_2)$.
We denote the random variable that indicates whether an adversary $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ wins the game or not as $\mathsf{APGame}^{\mathsf{Protect},\mathsf{Eval}}_{D_f,\{X_f\}_{f\in\mathcal{F}}}(1^\lambda, (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2))$.

As noted in [CMP20] and [AKLL+22], an adversary can always win the game with a trivial probability (that we define formally next) by applying the following strategy: $\mathcal{A}_0$ forwards the quantum protection state to either $\mathcal{A}_1$ or $\mathcal{A}_2$ and nothing to the other one. The one who receives the state can answer the challenge with probability close to 1 using the Eval algorithm, and the other one returns the optimal answer given their challenge.

Thus, given a family $\mathcal{F}$, and distributions D_f and $\{X_f\}_{f\in\mathcal{F}}$, we define the trivial probability of winning the piracy game as

$$p^{trivial}_{D_f,\{X_f\}_{f\in\mathcal{F}}} := \max_{i\in\{1,2\}} \mathop{\mathbb{E}}_{d_f\in D_f} \max_{y\in\mathcal{Y}} \Pr[y \mid x_i]$$

Definition 2.31 — δ-Anti-Piracy Security of a Copy-Protection Scheme

A copy-protection scheme of a family $\mathcal{F}$ has δ-*anti-piracy security* with respect to the distributions D_f and $\{X_f\}_{f\in\mathcal{F}}$ if no QPT adversary $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ can win the piracy game for $\mathcal{F}$ with respect to the distributions D_f and $\{X_f\}_{f\in\mathcal{F}}$ with a probability significantly greater than $1 - \delta(\lambda)$.
More precisely, for any QPT adversary $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$

$$\Pr\left[\mathsf{APGame}^{\mathsf{Protect},\mathsf{Eval}}_{D_f,\{X_f\}_{f\in\mathcal{F}}}(1^\lambda, (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)) = 1\right] \leq 1 - \delta(\lambda) + \mathsf{negl}(\lambda)$$

When $\delta(\lambda) = 1 - p^{trivial}_{D_f,\{X_f\}_{f\in\mathcal{F}}}$, we simply say that the copy-protection scheme has *anti-piracy security*.

For ease of notations, we will use f and d_f indifferently, and we will not write the dependence on λ in the following when clear from the context.

Part I

Quantum Security

Chapter

3 Quantum Security for Classical Encryption

In this chapter, we present quantum security definitions for classical encryption schemes in both symmetric-key and public-key settings, as well as their characterizations. These notions were introduced in [CEV20], and are the first quantum chosen-ciphertext security notions for classical encryption with quantum challenge queries.

Chapter content

3.1 Defining Security for Encryption Against Quantum Adversaries

Classical Security Notions

Indistinguishability-based security definitions are modeled as a game between a challenger and an adversary $\mathcal{A}$. In the Find-Then-Guess style, the game starts with a first learning phase (with access to some oracles), followed by a challenge phase where $\mathcal{A}$ sends a challenge query (two messages x_0 and x_1 to be encrypted) and receives a challenge ciphertext (encryption of x_b). Afterwards, a second learning phase follows, and finally, $\mathcal{A}$ outputs a solution (its guess for the bit b). The security reduction consists in constructing a new adversary which simulates $\mathcal{A}$ and solves some hard underlying problem. The learning phases define the type of attacks: chosen-plaintext attacks (CPA) if the adversary has access to an encryption oracle in both learning phases, and chosen-ciphertext attacks (CCA) in case it also has access to a decryption oracle in the learning phases (non-adaptive or CCA1 if it is restricted to the first learning phase, and adaptive or CCA2 otherwise).

Indistinguishability against adaptive chosen-ciphertext attack (IND-CCA2) is usually considered the most desirable security notion for encryption. In the CCA2 games, the adversary is restricted not to ask for decryption of the challenge ciphertext, otherwise, this would lead to a trivial guess of the bit b. It is the role of the challenger to ensure that the adversary obeys this rule, which intrinsically requires the ability to copy, store and compare classical strings.

Boneh-Zhandry's Security Notions [BZ13b]

Boneh and Zhandry propose the first definition of IND-CCA for both symmetric and public-key encryption schemes against quantum adversaries allowed to make quantum encryption and decryption queries. But they show that the natural translation of the classical Find-then-Guess paradigm to the quantum setting is unachievable, even for IND-CPA security. To overcome this impossibility, they resort to considering quantum queries during the learning phases only, and classical queries during the challenge phase. In addition to looking artificial, this inconsistency between the learning phases and the challenge phase may lead to a cryptographic construction that fulfills this security notion (IND-qCPA or IND-qCCA) while being subject to an attack.

For instance, in [ATTU16], the authors verify IND-qCPA security of XTS mode of operation (with quantum learning queries and classical challenge queries). They design a block cipher such that an encryption scheme in XTS mode, instantiated with that block cipher, can be attacked during the learning phase using quantum learning queries. However, this attack cannot be used to violate the IND-qCPA security definition. The explanation for this inconsistency is that this attack cannot be implemented in the challenge phase due to the classical restriction imposed on the adversary. This example supports our claim that the inconsistency between the learning phases and the challenge phase can be problematic and should be overcome.

Quantum IND-CCA2 Security Notions

To date, defining the CCA2 security with quantum challenge queries remains unsolved. In [GHS16], the authors address the inconsistency described above for the case of

symmetric encryption, but only for IND-CPA, and leave as an open problem the IND-CCA definitions.

The main obstacle is to define how the challenger should reply to the quantum decryption queries after the adversary has made the quantum challenge queries. When the challenge queries are classical, they can be stored and later the challenger can return $\perp$ if the adversary submits one of them as a decryption query. Although it is trivial and inherent to store the challenge ciphertext in the classical setting, it is highly non-trivial to store ciphertexts in the quantum world, due to a number of technical obstacles, all of which can be traced to quantum no-cloning [WZ82] and the destructiveness of quantum measurements [FP96].

3.1.1 Our Approach

Towards resolution, we start from a recent groundbreaking technique that allows for on-the-fly simulation of random oracles in the quantum setting: Zhandry's compressed oracles [Zha19a]. The goal of his work is to overcome the recording barrier, by allowing the reduction to record information about the adversary's queries, which is a key feature of many classical ROM proofs.

Zhandry's key observations are threefold. First, instead of considering a random function h being chosen beforehand, one can purify the adversary's mixed state by putting h in uniform superposition $\sum_h |h\rangle$. This observation is a technicality that allows us to fulfill the two next points. Then, the next observation is that, by doing the queries in the Fourier basis, the data will be written to the oracle's registers instead of writing to the opposite direction. This enables the simulator to get some information about the adversary's queries. Finally, the last and most important one is that the simulator needs to be ready to forget some point it simulated previously, by performing a particular test on the database after answering the query. In particular, Zhandry defines a test computation that maps $|+\rangle \mapsto |+\rangle |1\rangle$ and $|\phi\rangle \mapsto |\phi\rangle |0\rangle$ for any $|\phi\rangle$ orthogonal to $|+\rangle$, where $|+\rangle = \sum_x |x\rangle$ is the uniform superposition state. The "test-and-forget" procedure can be implemented by first performing the query in the Fourier basis and then doing the test operation on the output registers (of the simulator). This test determines whether the adversary has any information from the oracle at some input. If not, that pair will be removed from the database so that the adversary cannot detect that it is interacting with a simulated oracle.

This technique has been extended from random oracles to lazy-sampling of non-uniform random functions in [CMSZ19]. The intuition is almost the same, except that now one starts from the all-zero state, performs an *efficient* sampling operation that computes the function $f(x)$ according to some non-uniform distribution – it is the quantum Fourier transform (QFT) operation in the uniform setting. One then performs the query in the Fourier basis, transforms back to the computational basis and applies the "test-and-forget" operation (which is defined similarly as in the uniform setting). For this to work, the two important requirements are that: i) the sampling operation must be efficient; ii) the function distribution must be independent for every input.

To define security for encryption, we choose the real-or-random paradigm to work with. This is because partially, the real-or-random paradigm does not suffer from Boneh-Zhandry's impossibility (discussion below). Furthermore, it is actually possible to define quantum chosen-ciphertext security for this paradigm using the quantum lazy-sampling technique we just described. In what follows, let us focus on the random world of the

paradigm. For each challenge query in the random world, the challenger applies a random function to the plaintext registers before encrypting, all aforementioned requirements are met: the encryption of each submitted plaintext is actually an encryption of another uniformly random plaintext, and since the encryption algorithm is efficient, the sampling operation can also be efficiently constructed.

The above idea gives us a reasonable way to define adaptive chosen ciphertext security against quantum challenge queries: by instantiating the encryption oracle with this lazy-sampling technique, we are able to keep track of the information needed to formulate the CCA2 notions, namely the challenge queries the adversary has made, and the challenge ciphertexts it has received. However, applying Zhandry's framework directly to our setting does not work, and more efforts are needed. For example, one main difference is that in our setting, when making queries to the random oracle, there is no response register (from the adversary). In Zhandry's framework, this response register is essential for the technique, as the "test-and-forget" procedure works based on the value of this register. Another problem is how to implement the oracle with an *one-shot* call to the encryption algorithm: this is necessary when defining "one-time" security, or when doing security reductions. We refer the reader to Section 3.2 for technical details.

3.1.2 Discussion

This line of work on defining security for encryption in the quantum world started by Boneh and Zhandry in [BZ13b]. They show that quantizing the notion of classical "left-or-right" indistinguishability is unachievable, even for chosen-plaintext security. In more details, the adversary sends two input-message registers for the challenge phase:

$$\sum_{x_0,x_1,y} \alpha_{x_0,x_1,y} \left|x_0, x_1, y\right\rangle \mapsto \sum_{x_0,x_1,y} \alpha_{x_0,x_1,y} \left|x_0, x_1, y \oplus \mathsf{Encrypt}(x_b)\right\rangle . \tag{3.1}$$

For any classical encryption scheme, the adversary can perform an efficient attack which allows it to get the bit b with overwhelming probability. Follow-up works [GHS16; MS16; GKS21] manage to bypass this impossibility and give security definitions that allow the adversary to send quantum challenge queries. These works use different approaches, we give a discussion on these approaches and relate them to ours below.

On the query models. In [CETU21], all possible qIND-qCPA security notions for symmetric-key encryption have been studied. It was proven that "real-or-random" in the standard oracle model and "left-or-right" in the minimal oracle model are among the strongest ones we could achieve, and at the same time, the two notions are provably incomparable[4][5]. We believe that the standard oracle model is a more realistic query model, and thus security notions defined in this model might be a better one, for several reasons:

- In the symmetric-key setting, [GHS16] shows that with the decryption oracle, minimal oracles can be efficiently simulated by standard oracles. However, we stress that in general, unlike the symmetric setting, in the public-key setting, the

[4]Recall that in the standard oracle model, the query is implemented as $|x, y\rangle \mapsto |x, y \oplus f(x)\rangle$. In the minimal oracle model, the query is implemented as $|x\rangle \mapsto |f(x)\rangle$.

[5]We note that in [CETU21], instead of "real-or-random" (as we are considering here), they consider "real-or-permuted" notion. However, their results translate directly to "real-or-random" notions, and also to the public-key setting (using the hybrid encryption approach).

requirement of having the decryption key simultaneously with the public key is unrealistic in most of the cases. The encryption machine should not hold the secret key for practical use. Thus, defining security for public-key encryption in the minimal oracle model is not possible in general.

- Implementing queries in the minimal oracle model is only applicable to injective functions, which definitely does not include decryption. Thus, one still needs the standard oracle model to define chosen-ciphertext security: the minimal oracle model for encryption queries, and the standard oracle model for the decryption queries. This type of notions is not consistent in our opinion.

- The standard oracle model captures the quantum fault attacks while the minimal oracle model does not (see [GHS16]).

- Oracles implemented in the minimal oracle model require extra quantum computation. That is, the challenger has to use its secret information/randomness twice in the computation (once for encryption and once for recovering the message). In the standard oracle model, the second computation is not needed. (We note that in our notion, in the real game, the challenger implements the encryption oracle straightforwardly in the standard oracle model and does not need any extra computation.) This might limit some quantum attacks. For example, consider the smartcard frozen attacks [GHS16], here if the adversary wants to make an encryption query in superposition, it is arguably better to "use" the standard oracle model, as the minimal oracle model requires a longer coherent time of the device, otherwise, the attacks might not work at all.

One might also ask whether the adversary can only prepare one message register per challenge query in the "real-or-random" notion is somehow limiting. [CETU21] shows that the currently known way to have 2 message registers per challenge query (as in the "left-or-right" paradigm) seems to be to consider the minimal oracle model if one wants to achieve a strong notion. Combining with Boneh-Zhandry's impossibility on defining "left-or-right" security in the standard model [BZ13b], our notions might be the best we can hope for.

Semantic Security in the Quantum World. In the classical setting, semantic security [GM84], the computational complexity analogue to perfect security, is considered as the strongest possible security notion and is shown to be equivalent to all indistinguishability notion. Semantic security formulates that whatever can be efficiently computed (represented by a target function $f_{target}(\cdot)$) from the ciphertext and additional partial information about the plaintext (represented by a function $f_{aux}(\cdot)$) can be efficiently computed given only the length of the plaintext and the same partial information. Quantum semantic security was first studied in [GHS16]. Albeit with some restrictions on the adversary, this notion is equivalent to the "left-or-right" indistinguishability notion in the minimal oracle model. However, we note that this equivalence does not imply that this "left-or-right" indistinguishability notion in the minimal oracle model is the strongest one, as we explained above (essentially, this follows from the results given in [CETU21]).

In the following, we show that a natural translation of semantic security to the quantum setting in the standard oracle model is unachievable. The impossibility follows essentially from the nature of the standard oracle model, in which each query is modeled

as $|x, y\rangle \mapsto |x, y \oplus f(x)\rangle$. The crucial point here is that the adversary receives in the challenge phase not only the ciphertext *but also the plaintext*, that is (a superposition of) $|x, y \oplus \mathsf{Encrypt}(x)\rangle$. This allows the adversary to output any value $f_{target}(x)$ for some target function $f_{target}(\cdot)$. In the simulation, the simulator receives no encryption, but only the auxiliary state α on $|x, y\rangle$, computed by some quantum circuit C_{aux} on the plaintext state. We note that since the quantum circuit C_{aux} is given by the adversary, C_{aux} does not necessarily preserve the input registers $|x\rangle$ (for example, take C_{aux} a quantum circuit tracing out the input registers and outputting a constant). As such, the simulator has no information on the plaintext, while the adversary does. Thus there is no simulator that can simulate the adversary efficiently. This gives us some hints that defining a generic quantum semantic security might not be possible.

Quantum Encryption Approaches [AGM18]. Since we now consider the adversary's challenge queries as quantum states, it may be tempting to think that the approaches from the literature on quantum encryption (that is, the problem of encrypting *quantum* data) would work here. The notorious "recording barrier" that we face in this work has arisen previously in the literature on quantum encryption. In particular, devising the notions of quantum ciphertext indistinguishability under adaptive chosen-ciphertext attack and quantum authenticated encryption [AGM18] requires circumventing similar obstacles. However, [AGM18] defines IND-CCA2 security for *quantum encryption*, which inherently requires the users to have quantum computers, while in this work, we focus on classical encryption that can be implemented on classical computers and only needs to be secure against quantum adversaries. We show below that indeed the approach of [AGM18] would not help.

On a high level, an adversary $\mathcal{A}$ has negligible probability in distinguishing between two experiments: in the real one, it has access to encryption and decryption oracles with no restrictions, whereas in the random one, the challenge encryption oracle replaces $\mathcal{A}$'s queried plaintexts by random ones (half of a maximally-entangled state), and the decryption oracle answers with the originally queried plaintexts if the adversary asked for decryption of a challenge ciphertext (which can be done by first decrypting the ciphertext and applying a measurement on the entangled state), otherwise it answers normally. It is tempting to say that this approach resolves the problem of defining chosen-ciphertext security for the symmetric-key setting in *the minimal oracle model*. However, as explained above, the minimal oracle model does not support decryption queries, and it is not clear if this approach is compatible with the standard oracle model. In the context of standard oracles, this approach does not work unfortunately. The adversary can then use the same strategy to detect the random experiment's simulation: it prepares a maximally-entangled state $|\phi^+\rangle_{XX'}$ and uses half of it (the registers X) as the challenge plaintext, and keeps X'. After receiving the challenge ciphertext, it measures the plaintext registers and X', and trivially distinguishes whether it is in the random experiment. We note that this attack cannot be performed without relaying, that is the plaintext registers X need to be available to $\mathcal{A}$ after the challenge encryption. However, non-relaying is indistinguishable from being traced out the plaintext registers (from $\mathcal{A}$'s perspective). This inherently reduces to a definition with classical challenge queries, which defeats our goals.

3.2 How to Record Encryption Queries in the Random World?

The starting point towards our goal of defining indistinguishability-based security notions for encryption is to explain how the challenger should reply to quantum decryption queries in the second learning phase after the adversary has made the quantum encryption queries in the challenge phase. This implies explaining how it could record these quantum challenge queries. In this section, we show how this can be done in the random world.

3.2.1 Ciphertext Decomposition

For simplicity, let we denote the encryption algorithm as a function f that takes as input a plaintext $x \in \mathcal{X}$, a randomness $r \in \mathcal{R}$ and outputs a ciphertext $y \leftarrow f(x;r) \in \mathcal{Y}$. We also assume that the domain of f is $\mathcal{X} = \{0,1\}^m$, its range is $\mathcal{Y} = \{0,1\}^n$, and the randomness space $\mathcal{R} = \{0,1\}^\ell$. We make a convention that $f(\perp) = 0$, where $\perp$ denotes some symbol outside the domain $\mathcal{X}$ and the range $\mathcal{Y}$. We define ciphertext decomposition as follows.

Definition 3.1 — Ciphertext Decomposition

For a function f, for all messages $x \in \mathcal{X}$, we write $y := (y_1 \| y_2) \leftarrow f(x;r)$ and define:

Message-independent: y_1 is message-independent if for all randomness r, there exists a function g such that $y_1 := g(r)$. In other words, the message-independent component of the ciphertext can be computed solely from the randomness r, independent of the message x. Furthermore, we require that $0 \leq |y_1| \leq |y|$.

Message-dependent: y_2 is message-dependent if for all randomness r, there exists no function g such that $y_2 := g(r)$. In other words, the message-dependent component of the ciphertext can not be computed solely from the randomness r. Furthermore, we require that $1 \leq |y_2| \leq |y|$.

We will also write $f := f_2 \circ f_1$, where f_1 acts only on the randomness, and f_2 acts on both the randomness and the plaintext.

Remark 3.1. Our definition above can be defined for any encryption scheme, without losing of generality. Furthermore, it also does not exclude some artificial encryption scheme such that the encryption is deterministic when the plaintext x is some special value (for example, the secret key), that is, there exists a function g such that $y_2 := g(x)$.

Remark 3.2. The definition of ciphertext decomposition is merely served as a technical step towards constructing the compressed encryption oracle in the random world in subsequent sections. We note that in an actual proof of security of an encryption scheme, one usually needs not to pay attention to this decomposition definition.

3.2.2 Oracle Variations

Here, we describe some oracle variations which will be used later in subsequent sections, the so-called *standard oracle* and *Fourier oracle*. These oracles and their equivalence

are proven in much of literature on quantum-accessible oracles (e.g., see [KKVB02; CMSZ19; Zha19a]).

Standard oracles. For any function f with domain $\mathcal{X} = \{0,1\}^m$ and range $\mathcal{Y} = \{0,1\}^n$, the standard oracle for f is a unitary defined as

$$\mathsf{StdO}_f \sum_{x,y} \alpha_{x,y} \left|x, y\right\rangle_{XY} \mapsto \sum_{x,y} \alpha_{x,y} \left|x, y \oplus f(x)\right\rangle_{XY}. \tag{3.2}$$

The standard oracle can also be implemented in the truth table form: for each query, the oracle's internal state consists of $n2^m$-qubit F registers containing the truth table of the function. For short, we write $|f(0)\| \ldots \|f(2^m-1)\rangle$ as $|D\rangle$. Then, StdO_f performs the following map (on the adversary's basis states):

$$\begin{aligned} \mathsf{StdO}_f \left|x, y\right\rangle_{XY} \otimes \left|D\right\rangle_F &\mapsto \left|x, y \oplus D(x)\right\rangle_{XY} \left|D\right\rangle_F \\ &= \left|x, y \oplus f(x)\right\rangle_{XY} \left|D\right\rangle_F \end{aligned} \tag{3.3}$$

The equivalence of these two oracle variations follows directly from the fact that for each query, if we trace out the oracle's internal registers, the mixed state of the adversary in both cases will be identical.

Fourier oracles. The Fourier oracle model $\mathsf{FourierO}_f$, while technically provides a different interface to the adversary, can be mapped to the standard oracle by QFT operations. The initial state of $\mathsf{FourierO}_f$ is

$$\mathsf{QFT}^F \left|D\right\rangle_F = \frac{1}{\sqrt{2^{n2^m}}} \sum_E (-1)^{E \cdot F} \left|E\right\rangle_F. \tag{3.4}$$

On the basis states, the Fourier oracle $\mathsf{FourierO}_f$ is defined as follows.

$$\begin{aligned} &\mathsf{FourierO}_f \left|x, z\right\rangle_{XY} \otimes \frac{1}{\sqrt{2^{n2^m}}} \sum_E (-1)^{E \cdot D} \left|E\right\rangle_F \\ &\mapsto \frac{1}{\sqrt{2^{n2^m}}} \sum_E (-1)^{E \cdot D} \left|x, z\right\rangle_{XY} \left|E \oplus P_{x,z}\right\rangle_F. \end{aligned} \tag{3.5}$$

where $P_{x,z}$ is the point function that outputs z on x and 0 everywhere else. Intuitively, with the Fourier oracle, instead of adding data from the oracle's registers to the adversary's registers, it adds in the opposite direction.

Lemma 3.1 ([KKVB02; Zha19a]). *For any adversary $\mathcal{A}$ making queries to StdO_f, let $\mathcal{B}$ be the adversary that is identical to $\mathcal{A}$, except it performs the Fourier transformation to the response registers before and after each query. Then* $\Pr\left[\mathcal{A}^{\mathsf{StdO}_f}() = 1\right] = \Pr\left[\mathcal{B}^{\mathsf{FourierO}_f}() = 1\right]$.

Proof. Each oracle can be constructed by an f-independent quantum circuit containing just one copy of the other, that is

$$\mathsf{QFT}^{YF} \circ \mathsf{StdO}_f \circ \mathsf{QFT}^{\dagger YF} = \mathsf{FourierO}_f, \tag{3.6}$$

$$\mathsf{QFT}^{\dagger YF} \circ \mathsf{FourierO}_f \circ \mathsf{QFT}^{YF} = \mathsf{StdO}_f. \tag{3.7}$$

□

3.2.3 Recording Queries in the Random World

As we have explained in Section 3.1, to define chosen-ciphertext security, we follow the real-or-random paradigm. In this section, we show how to process queries and record them in the random world, in which before applying the encryption algorithm f, the challenger chooses a random function h and applies it to the plaintext registers. As such, we also denote the encryption procedure in the random world as $f \circ h$. In what follows, we abuse the notation and write $f \circ h$ in the subscript of the oracle's notation with this meaning: for each query, a random function h is chosen uniformly by the oracle, so that h is not a pre-defined function. We note that the function f is known to the adversary though.

Single-query setting. We first start describing the oracle operations handling a single query and describe the general case later.

Without loss of generality, we assume that the query's response register Y can be decomposed into two parts Y_1, Y_2, in which the first part corresponds to the message-independent component, and the second part corresponds to the message-dependent component. Let $|Y_1| := n_1$ and $|Y_2| := n_2$ where $n_1 + n_2 = n$.

In the standard oracle model, the encryption oracle is implemented by first sampling a randomness r, a function $h : \mathcal{X} \to \mathcal{X}$ uniformly at random, and then applying the encryption algorithm f on the input $(h(x); r)$. From the adversary's point of view, this is equivalent to h being in uniform superposition $\sum_h |h\rangle$ and performing the following map:

$$|x, y\rangle_{XY} \otimes |r\rangle_R \sum_h |h\rangle_H \mapsto \sum_h |x, y \oplus f((h(x)); r)\rangle_{XY} |r\rangle_R |h\rangle_H . \tag{3.8}$$

Augmenting the joint system with a uniform superposition register H is a *purification* of the adversary's mixed state, and tracing out H (i.e., projecting onto the one-dimensional subspace spanned by $|h\rangle$) recovers the original mixed state. Moreover, this projection, which is outside of the adversary's view, is undetectable by any adversary $\mathcal{A}$.

Using ciphertext decomposition definition, we can write Equation (3.8) as follows.

$$\begin{aligned} |x, y_1 \| y_2\rangle_{XY_1Y_2} \otimes |r\rangle_R \sum_h |h\rangle_H &\mapsto \sum_h |x, (y_1 \| y_2) \oplus f(h(x); r)\rangle_{XY_1Y_2} \otimes |r\rangle_R |h\rangle_H \\ &= \sum_h |x, y_1 \oplus f_1(r), y_2 \oplus f_2(h(x); r)\rangle_{XY_1Y_2} \otimes |r\rangle_R |h\rangle_H . \end{aligned} \tag{3.9}$$

We further note that, since the same randomness r is used for all "slots" in superposition, $f_1(r)$ is also the same for all "slots". In other words, $f_1(r)$ is just a classical value, which can be computed independently of the adversary's query. As a result, only the message-dependent registers are needed for recording queries. From now on to the rest of this section, we only consider the message-dependent parts in the adversary's response registers as well as the oracle's registers. These parts are denoted with index 2 in subscript (e.g., $y_2, z_2, f_2, \ldots$).

Now we describe our compressed encryption oracles. We first introduce some local procedures acting on the oracle's side, possibly controlled by the adversary's registers. Let Decomp_x be the identity operator except for

$$\mathsf{Decomp}_x \left(|r\rangle |x\rangle \frac{1}{\sqrt{2^m}} \sum_{u \in \{0,1\}^m} |u\rangle \frac{1}{\sqrt{2^{n_2}}} \sum_v (-1)^{f_2(u;r)\cdot v} |v\rangle \right) = |r\rangle |\perp\rangle |0\rangle |0\rangle , \tag{3.10}$$

and

$$\mathsf{Decomp}_x\left(|r\rangle|\perp\rangle|0\rangle|0\rangle\right) = |r\rangle|x\rangle\frac{1}{\sqrt{2^m}}\sum_{u\in\{0,1\}^m}|u\rangle\frac{1}{\sqrt{2^{n_2}}}\sum_v(-1)^{f_2(u;r)\cdot v}|v\rangle\,. \tag{3.11}$$

It is clear that Decomp_x is a unitary operator. Furthermore, applying it twice results in the identity, thus Decomp_x is an involution.

Using the notion similar to the description of Zhandry's compressed random oracle in [Zha19a], we introduce the notion of a database D that is maintained by the oracle as follows. A database D will be a collection of tuples $(x,(x',y))$, where $(x,(x',y))\in D$ corresponds to $D(x)=(x',y)$. We say $D(x)=\perp$ if there is no such pair for an input x. For a database D with $D(x)\neq\perp$, we also write $D=\{x,u,v\}\cup D'$ where $D'(x)=\perp$. D consists of all the oracle's registers, except the randomness registers R. Decomp is then defined as the related unitary acting on the joint quantum system as follows.

$$\mathsf{Decomp}\,|x,z_2\rangle\otimes|r\rangle|D\rangle = |x,z_2\rangle\otimes\mathsf{Decomp}_x|r\rangle|D\rangle\,. \tag{3.12}$$

Let Init be the procedure that samples a random r uniformly and initializes a new register $|r\rangle|\perp,0,0\rangle$. Let FourierO′ be unitary defined on the adversary's basis states as:

$$\begin{aligned}
&\mathsf{FourierO}'\,|x,z_2\rangle\otimes|r\rangle|D\rangle\\
=\;&\mathsf{FourierO}'\,|x,z_2\rangle\otimes|r\rangle\frac{1}{\sqrt{2^m}}\frac{1}{\sqrt{2^{n_2}}}\sum_{u,v}(-1)^{v\cdot f_2(u;r)}|\{x,u,v\}\cup D'\rangle\\
=\;&|x,z_2\rangle\otimes|r\rangle\frac{1}{\sqrt{2^m}}\frac{1}{\sqrt{2^{n_2}}}\sum_{u,v}(-1)^{v\cdot f_2(u;r)}|\{x,u,v\oplus z_2\}\cup D'\rangle\,.
\end{aligned} \tag{3.13}$$

Finally, we define the $\mathsf{CFourierO}_{f_2\circ h}$ oracle[6]:

$$\mathsf{CFourierO}_{f_2\circ h} := \mathsf{Decomp}\circ\mathsf{FourierO}'\circ\mathsf{Decomp}\circ\mathsf{Init}. \tag{3.14}$$

We state the following lemma:

Lemma 3.2. *In the single-query setting, the compressed Fourier oracle* $\mathsf{CFourierO}_{f_2\circ h}$ *acts on a basis state* $|x,z_2\rangle$ *where* $x\in\mathcal{X}$ *and* $z_2\in\{0,1\}^{n_2}$*, as follows.*

- *If* $z_2=0$*, then* $\mathsf{CFourierO}_{f_2\circ h}|x,z_2\rangle\mapsto|x,z_2\rangle\otimes|r\rangle|\perp,0,0\rangle$*.*
- *If* $z_2\neq0$*, then* $\mathsf{CFourierO}_{f_2\circ h}|x,z_2\rangle\mapsto|x,z_2\rangle\otimes|\phi_{x,z_2}\rangle$*, where*

$$|\phi_{x,z_2}\rangle := |r\rangle\frac{1}{\sqrt{2^{m+n_2}}}\sum_u\sum_v(-1)^{f_2(u;r)\cdot v}|x,u,v\oplus z_2\rangle\,.$$

Furthermore, for any adversary $\mathcal{A}$ *making a single query to* $\mathsf{StdO}_{f_2\circ h}$*, let* $\mathcal{B}$ *be the adversary that is identical to* $\mathcal{A}$*, except it performs the Hadamard transformation* $\mathsf{H}^{\otimes n}$ *to the response registers before and after the query. Then* $\Pr\left[\mathcal{A}^{\mathsf{StdO}_{f_2\circ h}}()=1\right]=\Pr\left[\mathcal{B}^{\mathsf{CFourierO}_{f_2\circ h}}()=1\right]$*.*

[6] For notation consistency, we use the same subscript in compressed oracles as for standard oracles. However, we note that there is no real function h in the implementation of CFourierO and its variants.

Proof. To prove the lemma, it is enough to show that $\mathsf{CFourierO}_{f_2\circ h}$ and $\mathsf{FourierO}_{f_2\circ h}$ are perfectly indistinguishable.

We prove this through a sequence of games. In what follows, we ambiguously denote $\mathsf{QFT}\,|f_2(x;r)\rangle$ by $|\eta_x\rangle$ for each $x \in \{0,1\}^m$. We will also take $y \oplus \perp = y, y \cdot \perp = 0$. When the adversary's response register is $|+\rangle$ (which corresponds to $|0\rangle$ in the Fourier basis), we can write, on the truth table of the oracle (for both $\mathsf{FourierO}_{f_2\circ h}$ and $\mathsf{StdO}_{f_2\circ h}$), the column with index x where x is the query's input as $\perp$.

Game G_0: In this game, the adversary interacts with the Fourier oracle $\mathsf{FourierO}_{f_2\circ h}$, whose initial state is $|r\rangle \frac{1}{\sqrt{2^{m2^m}}} \sum_h |(h(0), \eta_{h(0)})\| \cdots \|(h(2^{m-1}), \eta_{h(2^m-1)})\rangle$.

Game G_1: In this game, we represent the oracle in the form:

$$|r\rangle \frac{1}{\sqrt{2^{m2^m}}} \sum_h |(0, h(0), \eta_{h(0)})\| \cdots \|(2^m-1, h(2^{m-1}), \eta_{h(2^m-1)})\rangle\,.$$

The update procedure for a query is then simply $\mathsf{FourierO}'$. G_1 is identical to G_0, since we have inserted the input points $0, \ldots, 2^m-1$ into the oracle's state, which is independent of the adversary's state.

Game G_2: In this game, the oracle starts out as the "zero" database:

$$|r\rangle\,|(\perp,0,0)\|\cdots\|(\perp,0,0)\rangle\,.$$

Then a query is implemented as $\mathsf{Decomp}'^{\dagger} \circ \mathsf{FourierO}' \circ \mathsf{Decomp}'$, where $\mathsf{Decomp}' \coloneqq \otimes_{i=0}^{2^m-1} \mathsf{Decomp}_i$. At the beginning, Decomp' is applied to the "zero" database, which maps it to the complete database

$$|r\rangle \frac{1}{\sqrt{2^{m2^m}}} \sum_h |(0, h(0), \eta_{h(0)})\| \cdots \|(2^m-1, h(2^{m-1}), \eta_{h(2^m-1)})\rangle\,.$$

Then $\mathsf{FourierO}'$ is applied and the output state of G_2 in this stage will be exactly the output state of G_1. Since $\mathsf{Decomp}'^{\dagger}$ is a unitary that only operates on the oracle's register, its applications is undetectable to the adversary. So G_2 is perfectly indistinguishable from G_1.

Game G_3: In this final game, we use the compressed oracle $\mathsf{CFourierO}_{f_2\circ h}$. Let x be the query's input. We note that $\mathsf{FourierO}'$ and $\mathsf{Decomp}_{x'}$ commute for any $x' \neq x$. Thus, we can move the computation of $\mathsf{Decomp}_{x'}$ to come after $\mathsf{FourierO}'$, consequently, its applications cancel out. We then have:

$$\begin{aligned}
&\mathsf{Decomp}'^{\dagger} \circ \mathsf{FourierO}' \circ \mathsf{Decomp}'(|x,z\rangle \otimes |r\rangle\,|D\rangle)\\
=\ &\mathsf{Decomp}_x^{\dagger} \circ \mathsf{FourierO}' \circ \mathsf{Decomp}_x(|x,z\rangle \otimes |r\rangle\,|D\rangle)\\
=\ &\mathsf{Decomp}^{\dagger} \circ \mathsf{FourierO}' \circ \mathsf{Decomp}(|x,z\rangle \otimes |r\rangle\,|D\rangle).
\end{aligned}$$

We are left with a database D whose support has at most 1 defined point after the query in G_2. The remaining $\geq 2^m - 1$ points are all $(\perp, 0, 0)$. So we may end up with a superposition of databases that have at most one defined point. We then can move this defined point in the database to the first register (this is a unitary operator and is undetectable to the adversary) and obtain a superposition of databases that have a defined point only in the first register. Therefore we can discard all but the first register, without affecting the adversary's state. This shows that G_3 and G_2 are identical. □

The compressed Fourier encryption oracle in the random world $\mathsf{CFourierO}_{f\circ h}$ is straightforwardly obtained by running the message-independent function f_1 on the randomness r, transforming it to the Fourier basis and then composing it with $\mathsf{CFourierO}_{f_2\circ h}$. Formally, $\mathsf{CFourierO}_{f\circ h} \coloneqq (\mathsf{QFT}^{F_1} U^R_{f_1}) \circ \mathsf{CFourierO}_{f_2\circ h}$. We then have

Lemma 3.3. *For any adversary $\mathcal{A}$ making a single query to $\mathsf{StdO}_{f\circ h}$, let $\mathcal{B}$ be the adversary that is identical to $\mathcal{A}$, except it performs the Hadamard transformation $\mathsf{H}^{\otimes n}$ to the response registers before and after the query. Then $\Pr\left[\mathcal{A}^{\mathsf{StdO}_{f\circ h}}() = 1\right] = \Pr\left[\mathcal{B}^{\mathsf{CFourierO}_{f\circ h}}() = 1\right]$.*

Compressed standard encryption oracles. By applying Hadamard to the adversary's response registers before and after the query, and to the oracle's register F after the query, we also obtain the compressed standard encryption oracle $\mathsf{CStO}_{f\circ h}$. The oracle's state after the query is (in superposition of) $|r, x, u, f(u; r)\rangle$. Formally, $\mathsf{CStO}_{f\circ h} \coloneqq \mathsf{QFT}^{YF} \circ \mathsf{CFourierO}_{f\circ h} \circ \mathsf{QFT}^{Y}$. By applying the same argument as in Lemma 3.1 to $\mathsf{CFourierO}_{f\circ h}$ and $\mathsf{CStO}_{f\circ h}$, and combining with Lemma 3.3, the following lemma follows:

Lemma 3.4. *$\mathsf{CStO}_{f\circ h}$ and $\mathsf{StdO}_{f\circ h}$ are perfectly indistinguishable. That is, for any adversary $\mathcal{A}$, we have that $\Pr\left[\mathcal{A}^{\mathsf{StdO}_{f\circ h}}() = 1\right] = \Pr\left[\mathcal{A}^{\mathsf{CStO}_{f\circ h}}() = 1\right]$.*

Many-query setting. We denote $\mathsf{CStO}_{f\circ H}$ as the following oracle: for each query, $\mathsf{CStO}_{f\circ H}$ invokes a new instance of $\mathsf{CStO}_{f\circ h}$ with uniformly and independently randomness r. Similarly, $\mathsf{StdO}_{f\circ H}$ denote the following oracle: for each query, $\mathsf{StdO}_{f\circ H}$ samples uniformly and independently a randomness r and a random function h, and then answers that query using $\mathsf{StdO}_{f\circ h}$. By the standard hybrid argument, it is easy to verify that:

Lemma 3.5. *$\mathsf{CStO}_{f\circ H}$ and $\mathsf{StdO}_{f\circ H}$ are perfectly indistinguishable, in the many-query setting.*

For each i-th query, its oracle's database is $|D_i\rangle \coloneqq |x_i, u_i, f(u_i; r_i)\rangle$. Overall, the oracle's database D will be a collection of many tuples $(x, (x', y))$ where $(x, (x', y)) \in D$ means $f(x'; r) = y$ and $h(x) = x'$ for different random functions h.

3.2.4 A Technical Observation

Notice that from the proof of Lemma 3.2 above, we implement this compressed encryption oracle with at least two computations of f_2 (and so f) via two applications of Decomp. However, as we will see in later sections, it is crucial for our security reductions to simulate $\mathsf{CFourierO}_{f\circ h}$ with only one computation of f, which allows us to "outsource" f computations to other oracles. We now give an intuition why we can reduce many computations of f to one computation. Let's consider the following cases.

- The z_2 registers are all-zero. Note that since the initial state of the oracle database D is also all-zero, applying the first Decomp and then XORing the adversary's registers to the oracle's (i.e., the application of FourierO′) does not change the database's state. Finally, the second application of Decomp brings it back to all-zero state, which can be discarded. At the end of this step, D is empty. In this case, we see that we can skip FourierO′, and two applications of Decomp cancel out, leaving us no applications of f.

- The z_2 registers are not zero. By a similar argument, we have that the second application of Decomp has no effects on the joint system, leaving us only one application of f in the first application of Decomp.

We describe a quantum circuit in Figure 3.1, which applies a single computation of f_2 (denoted as a unitary U_{f_2}), implementing our compressed encryption oracle in the random world. Let Test be the unitary defined as Test $|0\rangle |b\rangle \mapsto |0\rangle |b\rangle$ and Test $|\phi\rangle |b\rangle \mapsto |\phi\rangle |b \oplus 1\rangle$ for any $|\phi\rangle$ orthogonal to $|0^{n_2}\rangle$ and $b \in \{0,1\}$. One can easily verify that this circuit outputs the same quantum state as stated in Lemma 3.2.

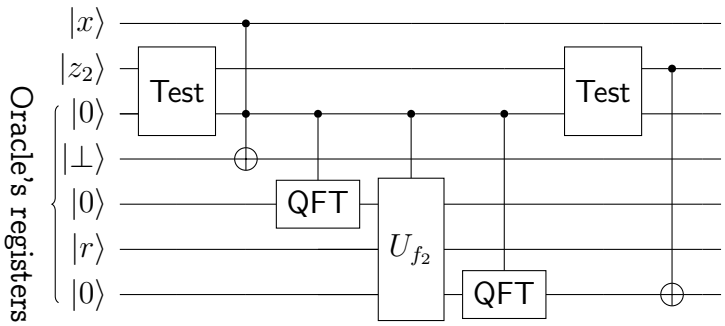

Figure 3.1: A quantum circuit implementing our $\mathsf{CFourierO}_{f_2 \circ h}$ oracle. Depending on the control bit b which is the output of Test, if $b = 1$, we apply U_{f_2}, otherwise, we apply the identity. The bit b will be discarded after the computation.

3.2.5 How to Answer Decryption Queries?

We now describe how to answer decryption queries in the random world using the database constructed above. Generally, we will consider any δ-correct encryption scheme (see Definition in Section 2.3.3).

We will start with a technical lemma, in which the decryption will answer "naively", that is if the ciphertext is $f(x'; r)$ for some x', the decryption oracle is expected to return x', even if x' was the output of a random function. (Roughly speaking, this decryption oracle mimics a standard decryption oracle with no restrictions on the adversary.) We call this decryption oracle the *naive decryption oracle*.

In the following, we abuse the notation and denote f^{-1} as the decryption algorithm. We then give the adversary access to a new oracle denoted $\mathsf{CInvO}_{f^{-1}}$ (this is our naive decryption oracle) which acts on the database, instead of $\mathsf{StdO}_{f^{-1}}$. Given access to $\mathsf{CInvO}_{f^{-1}}$, the bound on the distinguishing probability of the adversary when interacting with the compressed oracle $\mathsf{CStO}_{f \circ H}$ is stated in Lemma 3.6.

We define a classical procedure $\mathsf{FindImage}'$ which takes as input a ciphertext $y \in \mathcal{Y}$, and a database D. Then, it looks for a tuple $(x, (x', y)) \in D$. If found, it outputs $(b = 1, w = x')$, otherwise, it outputs $(b = 0, w = 0)$. Notice that there may be many tuples with the same y in D, but since an encryption scheme must be injective (for decryption to work), these pairs must have the same x'.

We define the unitary operation $\mathsf{CInvO}_{f^{-1}}$ for the inverse queries which maps the

basis state $|y,z\rangle \otimes |D\rangle$ to:

$$\begin{cases} U_{f^{-1}} |y,z\rangle \otimes |D\rangle = |y, z \oplus f^{-1}(y)\rangle \otimes |D\rangle & \text{if } \mathsf{FindImage}'(y,D) = (0,0), \\ |y, z \oplus w\rangle \otimes |D\rangle & \text{if } \mathsf{FindImage}'(y,D) = (1,w). \end{cases}$$

This unitary is implemented by a single call to f^{-1}, controlled by the output bit b of $\mathsf{FindImage}'$ recorded in some ancilla registers.[7]

Lemma 3.6. *For any (unbounded) oracle algorithm $\mathcal{A}$, and any δ-correct encryption scheme:*

$$\left|\Pr\left[\mathcal{A}^{\mathsf{StdO}_{f\circ H}, \mathsf{StdO}_{f^{-1}}}() = 1\right] - \Pr\left[\mathcal{A}^{\mathsf{CStO}_{f\circ H}, \mathsf{CInvO}_{f^{-1}}}() = 1\right]\right| \leq \mathcal{O}(q_i \cdot \delta), \tag{3.15}$$

where q_i is the number of inverse queries.

Proof. We prove this lemma through a sequence of games.

Game G_0: This is the game where $\mathcal{A}$ interacts with the standard oracles $\mathsf{StdO}_{f\circ H}$ and $\mathsf{StdO}_{f^{-1}}$.

Game G_1: This is identical to G_0, except that now the oracle $\mathsf{StdO}_{f\circ H}$ is simulated using the compressed oracle $\mathsf{CStO}_{f\circ H}$. Notice that $\mathsf{StdO}_{f^{-1}}$ operation does not touch the database registers, thus it commutes with any $\mathsf{CStO}_{f\circ h}$ operation. Since $\mathsf{CStO}_{f\circ H}$ is equivalent to the standard oracle $\mathsf{StdO}_{f\circ H}$, $\mathcal{A}$ cannot distinguish G_1 and G_0.

Game G_2: This is identical to G_1, except that now the oracle $\mathsf{StdO}_{f^{-1}}$ is replaced by the oracle $\mathsf{CInvO}_{f^{-1}}$.

Let $|\Psi\rangle$ be the joint system state of the adversary and the oracle before making any inverse query. Denote $\Delta := \mathsf{StdO}_{f^{-1}} - \mathsf{CInvO}_{f^{-1}}$. For each query $|y,z\rangle$ to the inverse oracle, we consider the registers y, z, D. We now examine three cases.

(a) Let D be such that $y \notin D$, that is, $\mathsf{FindImage}(y,D) = (0,0)$. Let P_1 be the projection onto the registers y, D such that $y \notin D$. In this case, the inverse oracle in both games applies the unitary mapping $|y,z\rangle \otimes |D\rangle \mapsto |y, z \oplus f^{-1}(y)\rangle \otimes |D\rangle$. Thus, $\Delta P_1 |\Psi\rangle = 0$.

(b) Let D be such that $y \in D$, that is, $\mathsf{FindImage}(y,D) = (1,w)$. Let P_2 be the projection onto the registers y, D such that $y \in D$ and $f^{-1}(y) = w$. In this case, we also have $\Delta P_2 |\Psi\rangle = 0$.

(c) Let D be such that $y \in D$. Let P_3 be the projection onto the registers y, D such that $y \in D$ but $f^{-1}(y) \neq w$. Thus $\|P_3 |\Psi\rangle\|^2$ is the probability of measuring y, D and get $y \in D$ such that $f^{-1}(y = f(x)) \neq x$ for some pre-image x of y. In this case, we have $\|\Delta P_3 |\Psi\rangle\|^2 \leq \delta$, by the definition that the encryption scheme is δ-correct.

Notice that $P_1 + P_2 + P_3 = \mathcal{I}$. Therefore, we have

$$\|\Delta |\Psi\rangle\|^2 = \left\|\sum_{i=1}^{3} \Delta P_i |\Psi\rangle\right\|^2 \overset{(*)}{\leq} \sum_{i=1}^{3} \|\Delta P_i |\Psi\rangle\|^2 \leq \delta, \tag{3.16}$$

[7] The oracle first computes $\mathsf{FindImage}'$, records the output in some ancilla register, performs the CNOT operation controlled on the output and finally un-compute $\mathsf{FindImage}'$.

where $(*)$ uses triangle inequality. Then the same holds true for any mixed state since any mixed state is in the convex hull of pure states. If $\mathcal{A}$ makes at most q_i inverse queries, the trace distance of the mixed state of the adversary in games G_2 and G_1 is at most $\mathcal{O}(q_i \cdot \delta)$. This completes the proof. □

Now we describe our actual decryption oracle in the random world. Instead of using FindImage′ which returns $(1, x')$, we use an identical FindImage except that it returns $(b = 1, w = x)$ when $(x, (x', y)) \in D$. The oracle $\mathsf{CInvO}_{f^{-1}}$ is redefined using FindImage as follows. It maps the basis state $|y, z\rangle \otimes |D\rangle$ to:

$$\begin{cases} U_{f^{-1}} |y, z\rangle \otimes |D\rangle = |y, z \oplus f^{-1}(y)\rangle \otimes |D\rangle & \text{if } \mathsf{FindImage}(y, D) = (0, 0), \\ |y, z \oplus w\rangle \otimes |D\rangle & \text{if } \mathsf{FindImage}(y, D) = (1, w). \end{cases}$$

3.2.6 Notation

From now on to the rest of this chapter, we will use the following notation:

- $\mathcal{O}$ to denote the standard encryption and decryption oracles StdO (which are distinguished by subscript, e.g., $\mathcal{O}_{\mathsf{SymEnc}}$ for encryption and $\mathcal{O}_{\mathsf{SymDec}}$ for decryption) in the real world.
- $\mathcal{R}$ to denote the compressed encryption and decryption oracles (which are distinguished by subscript, e.g., $\mathcal{R}_{\mathsf{SymEnc}}$ for encryption and $\mathcal{R}_{\mathsf{SymDec}}$ for decryption) in the random world. In particular, the encryption one will be implemented using CStO, and the decryption one using CInvO.

3.3 Quantum-Secure Symmetric Encryption

3.3.1 Definitions of Security

In this section, we use the compressed oracle technique defined above to define quantum real-or-random indistinguishability security notions.

High-level view. During the learning phases, $\mathcal{A}$ has access to the encryption standard oracle $\mathcal{O}_{\mathsf{SymEnc}(\mathsf{k},\cdot)}$. In the CCA case, it also has access to $\mathcal{O}_{\mathsf{SymDec}(\mathsf{k},\cdot)}$ in the first learning phase. We describe informally how we handle the challenge phase and the decryption queries in the second learning phase. The goal is to mimic the (purely) classical CCA security game in which: $\mathcal{A}$ gives a challenge plaintext and receives either encryption of it or encryption of a random message; during the second learning phase, if $\mathcal{A}$ makes a decryption query on the challenge ciphertext, it is given back the challenge plaintext in both games.

In the real-world $(b = 1)$, the adversary has no restrictions on the use of the decryption oracle (in particular, $\mathcal{A}$ can freely decrypt the challenge ciphertext – getting back the challenge plaintext, as in the classical case), so that the encryption oracle is simply implemented as the standard encryption oracle $\mathcal{O}_{\mathsf{SymEnc}(\mathsf{k},\cdot)}$ and the decryption oracle as the standard decryption oracle $\mathcal{O}_{\mathsf{SymDec}(\mathsf{k},\cdot)}$.

In the random-world $(b = 0)$, the challenger implements the challenge encryption oracle using a compressed encryption oracle $\mathcal{R}_{\mathsf{SymEnc}(\mathsf{k},\cdot)}$, and the decryption oracle in

the second phase $\mathcal{R}_{\mathsf{SymDec}(\mathsf{k},\cdot)}$ as described in Section 3.2.5. As in the real-world, this decryption oracle always returns the original plaintext (x) if the query is a challenge one, using the database. Otherwise, it just decrypts normally.

Formal definitions. Formally, denote $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$. In both games, $\mathcal{A}_1$ outputs an internal state $|\Phi\rangle$ after the first phase (i.e., the first learning phase), which will be given to $\mathcal{A}_2$ in the second phase (including the challenge and the second learning phase). We define a "real-or-random" oracle $\mathcal{RR}$ allowing $\mathcal{A}_2$ to make quantum challenge queries. For learning queries, $\mathcal{A}_2$ has access to $\mathcal{O}_{\mathsf{SymEnc}(\mathsf{k},\cdot)}$ and potentially a decryption oracle $\mathcal{DEC}$ defined as follows.

$$\mathcal{RR}(b) = \begin{cases} \mathcal{O}_{\mathsf{SymEnc}(\mathsf{k},\cdot)} & \text{if } b = 1 \\ \mathcal{R}_{\mathsf{SymEnc}(\mathsf{k},\cdot)} & \text{if } b = 0 \end{cases}, \text{ and } \mathcal{DEC}(b) = \begin{cases} \mathcal{O}_{\mathsf{SymDec}(\mathsf{k},\cdot)} & \text{if } b = 1 \\ \mathcal{R}_{\mathsf{SymDec}(\mathsf{k},\cdot)} & \text{if } b = 0 \end{cases}.$$

Definition 3.2 — Quantum Indistinguishability for Symmetric Encryption

Let $\mathcal{SE} := \langle \mathcal{K}, \mathsf{SymEnc}, \mathsf{SymDec} \rangle$ be a symmetric encryption scheme and let $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$ be a quantum adversary. For $qatk \in [qcpa, qcca1, qcca2]$, we define the following game, where the oracles $\mathcal{O}_1, \mathcal{O}_2$ are defined according to $qatk$:

Experiment $\mathsf{Expt}^{qind\text{-}qatk-b}_{\mathcal{SE}}(\lambda, \mathcal{A})$:

1 : $\mathsf{k} \xleftarrow{\$} \mathcal{K}(1^\lambda)$
2 : $|\Phi\rangle \leftarrow \mathcal{A}_1^{|\mathcal{O}_{\mathsf{SymEnc}(\mathsf{k},\cdot)}\rangle, |\mathcal{O}_1\rangle}(1^\lambda)$
3 : $b' \leftarrow \mathcal{A}_2^{|\mathcal{RR}(b)\rangle, |\mathcal{O}_{\mathsf{SymEnc}(\mathsf{k},\cdot)}\rangle, |\mathcal{O}_2\rangle}(|\Phi\rangle)$
4 : **return** b'

$qatk$	Oracle $\mathcal{O}_1$	Oracle $\mathcal{O}_2$
$qcpa$	$\varnothing$	$\varnothing$
$qcca1$	$\mathcal{O}_{\mathsf{SymDec}(\mathsf{k},\cdot)}$	$\varnothing$
$qcca2$	$\mathcal{O}_{\mathsf{SymDec}(\mathsf{k},\cdot)}$	$\mathcal{DEC}(b)$

We define $\mathcal{A}$'s advantage by

$$\mathsf{Adv}^{qind\text{-}qatk}_{\mathcal{A},\mathcal{SE}}(\lambda) := \left|\Pr\left[\mathsf{Expt}^{qind\text{-}qatk-1}_{\mathcal{SE}}(\lambda, \mathcal{A}) = 1\right] - \Pr\left[\mathsf{Expt}^{qind\text{-}qatk-0}_{\mathcal{SE}}(\lambda, \mathcal{A}) = 1\right]\right|.$$

We say $\mathcal{SE}$ is secure in the sense of qIND-qATK if $\mathcal{A}$ being QPT implies that $\mathsf{Adv}^{qind\text{-}qatk}_{\mathcal{A},\mathcal{SE}}(\lambda)$ is negligible.

Equivalence with Boneh-Zhandry's notions. To justify our notions, we show that when restricting our definitions to classical challenge queries, they are equivalent to Boneh-Zhandry's notions (IND-qATK). If we denote our restricted notions by IND-qATK$'$, a scheme $\mathcal{SE}$ is IND-qATK$'$ secure iff it is IND-qATK secure.

Indeed, because the challenge queries are classical, the simulator can store the challenge plaintexts and the challenge ciphertexts. Any simulator that returns $\perp$ if the adversary submits a challenge ciphertext in the sense of IND-qATK can be turned to a simulator that returns the original plaintext x in the sense of IND-qATK$'$, and vice versa. More precisely, we have that:

$$\mathsf{Adv}^{ind\text{-}qatk}_{\mathcal{A},\mathcal{SE}}(\lambda) \leq 2 \cdot \mathsf{Adv}^{ind\text{-}qatk'}_{\mathcal{A},\mathcal{SE}}(\lambda), \text{ and } \mathsf{Adv}^{ind\text{-}qatk'}_{\mathcal{A},\mathcal{SE}}(\lambda) \leq \mathsf{Adv}^{ind\text{-}qatk}_{\mathcal{A},\mathcal{SE}}(\lambda). \tag{3.17}$$

This follows from the standard argument (see [BDJR97]).

Single-message versus many-message security. We have presented definitions which allow the adversary to make $q(\lambda)$-many challenge queries to the real-or-random oracle. A scheme satisfying the definitions in the case when $q(\lambda) = 1$ is said to be *single-message* secure. The question of whether single-message security implies many-message security is the question of composability of the definitions, which is answered affirmatively below.

Theorem 3.1. *A symmetric encryption scheme $\mathcal{SE}$ is many-message* qIND-qATK *secure iff it is single-message* qIND-qATK *secure.*

Proof. The forward implication follows directly.

For the reverse direction, we use the standard hybrid argument that uses an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ with advantage ε to construct a new adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ which breaks the single-message security with advantage ε/q^2.

Define a sequence of games $G_0, \ldots, G_q$ in which $\mathcal{B}$ runs $\mathcal{A}$ and returns $\mathcal{A}$'s output as follows: For any game G_i,

1. $\mathcal{B}_1$ simulates $\mathcal{A}$'s $i-1$ first challenge queries as learning queries, that is, $\mathcal{B}$ just forwards $\mathcal{A}$'s directly to its encryption oracle.
2. $\mathcal{B}$ uses $\mathcal{A}$'s i-th challenge query as its challenge query.
3. For all $\mathcal{A}$'s other challenge queries, $\mathcal{B}_2$ treats them as encryption queries in the random world. In particular, it implements the encryption oracle for $\mathcal{A}$ using the compressed oracle $\mathcal{R}_{\mathsf{SymEnc}}$, except that it queries to its own encryption oracle as a learning query during the oracle implementation. We note that this is possible as explained in Section 3.2.3.

In the case of CCA2 security, $\mathcal{B}_2$ needs to be able to record $\mathcal{A}$'s $(i+1, \ldots, q)$-th challenge queries, since it needs to simulate the decryption correctly. This is done by using our recording technique as described. $\mathcal{B}_2$ also uses a slightly different decryption oracle in the random world in the second phase as follows. Let $\mathcal{R}'_{\mathsf{SymDec}}$ be the decryption oracle of $\mathcal{B}_2$ in the random world in the second phase, D be its database for the challenge query, and $\mathcal{R}_{\mathsf{SymDec}}$ be $\mathcal{B}_2$ simulated decryption oracle for $\mathcal{A}$. Then

$$\mathcal{R}_{\mathsf{SymDec}} |y, z\rangle |D\rangle = \begin{cases} \mathcal{R}'_{\mathsf{SymDec}} |y, z\rangle |D\rangle & \text{if } \mathsf{FindImage}(y, D) = (0, 0^m), \\ |y, z \oplus w\rangle |D\rangle & \text{if } \mathsf{FindImage}(y, D) = (1, w). \end{cases}$$

This oracle can be implemented identically as described in Section 3.2.5, except that instead of applying f^{-1}, it sends a decryption query on the y, z registers to $\mathcal{R}'_{\mathsf{SymDec}}$.

Note that $G_0 = \mathsf{Expt}^{qind\text{-}qatk-1}_{\mathcal{SE}}(\lambda, \mathcal{A})$ and $G_q = \mathsf{Expt}^{qind\text{-}qatk-0}_{\mathcal{SE}}(\lambda, \mathcal{A})$. Because $\mathcal{A}$ is able to distinguish $\mathsf{Expt}^{qind\text{-}qatk-1}_{\mathcal{SE}}$ from $\mathsf{Expt}^{qind\text{-}qatk-0}_{\mathcal{SE}}$, there exists some $g \in [\![1, q]\!]$ such that $\mathcal{A}$ distinguishes G_g from G_{g+1} with advantage at least ε/q. $\mathcal{B}$ can guess g correctly with probability $1/q$, thus $\mathcal{B}$'s overall advantage in breaking the single-message security is ε/q^2. □

3.3.2 A Separation Example

We show that upgrading from classical challenge queries to quantum challenge queries gives the adversary more power. In particular, we show that the IND-qCCA2 secure symmetric encryption scheme given by Boneh and Zhandry [BZ13b] is insecure once

the adversary can make even a single quantum challenge query in the sense of chosen plaintext security (qIND-qCPA). Our attack can be considered as an impossibility to achieve quantum indistinguishability for encryption schemes which follow the stream cipher-like paradigm (such as stream ciphers, block cipher modes of operation including CFB, OFB, CTR, or even some most widely used modes like GCM for authenticated encryptions).

Theorem 3.2. *Under the assumption that quantum-secure pseudorandom functions exist, there is an encryption scheme $\mathcal{SE}$ which is* IND-qCCA2 *secure, but* qIND-qCPA *insecure.*

Proof. We recall Boneh-Zhandry construction as follows.

Construction 3.1 — Boneh-Zhandry's construction [BZ13b]

Let F and G be quantum-secure pseudorandom functions. We construct the following encryption $\mathcal{SE} := \langle \mathcal{K}, \mathsf{SymEnc}, \mathsf{SymDec} \rangle$ where:

$\mathcal{K}(1^\lambda)$:

1 : $\mathsf{k}_1 \xleftarrow{\$} \{0,1\}^\lambda$
2 : $\mathsf{k}_2 \xleftarrow{\$} \{0,1\}^\lambda$
3 : **return** $\mathsf{k}_1 \| \mathsf{k}_2$

$\mathsf{SymEnc}(\mathsf{k}_1 \| \mathsf{k}_2, x)$:

1 : $r \xleftarrow{\$} \{0,1\}^\lambda$
2 : $c_1 \leftarrow F(\mathsf{k}_1, r) \oplus x$
3 : $c_2 \leftarrow G(\mathsf{k}_2, (r, x))$
4 : **return** $r \| c_1 \| c_2$

$\mathsf{SymDec}(\mathsf{k}_1 \| \mathsf{k}_2, r \| c_1 \| c_2)$:

1 : $x \leftarrow c_1 \oplus F(\mathsf{k}_1, r)$
2 : $c_2' \leftarrow G(\mathsf{k}_2, (r, x))$
3 : **if** $c_2 \neq c_2'$ **then**
4 : **return** $\perp$
5 : **return** x

Lemma 3.7 ([BZ13b, Theorem 4.10]). *The encryption scheme $\mathcal{SE}$ in Construction 3.1 is* IND-qCCA2 *secure.*

To show the qIND-qCPA insecurity of this scheme, we establish the following quantum computation. Let U_{OTP} be the unitary implementing the one-time pad encryption, but using the same classical randomness r (which is uniformly chosen beforehand) in superposition. For fixed $x_0, x_1 \in \{0,1\}^m$, we prepare the following state:

$$|\psi_1\rangle := \frac{1}{\sqrt{2}} \left(|x_0\rangle + |x_1\rangle \right) |0^m\rangle . \tag{3.18}$$

Applying U_{OTP} yields:

$$|\psi_2\rangle := \frac{1}{\sqrt{2}} \sum_{b \in \{0,1\}} |x_b, x_b \oplus r\rangle . \tag{3.19}$$

Then we apply a Hadamard transform to $2m$ qubits in all the registers. This yields the state:

$$\begin{aligned} |\psi_3\rangle &:= 2^{-\frac{2m+1}{2}} \sum_b \sum_{u \in \{0,1\}^m} (-1)^{x_b \cdot u} |u\rangle \sum_{v \in \{0,1\}^m} (-1)^{(x_b \oplus r) \cdot v} |v\rangle \\ &= 2^{-\frac{2m-1}{2}} \sum_{u,v} \delta_{u \cdot (x_0 \oplus x_1), v \cdot (x_0 \oplus x_1)} (-1)^{x_0 \cdot u \oplus (x_0 \oplus r) \cdot v} |u, v\rangle . \end{aligned} \tag{3.20}$$

If we measure $|\psi_3\rangle$, with probability 1, we get a random pair (u, v) such that

$$u \cdot (x_0 \oplus x_1) = v \cdot (x_0 \oplus x_1). \tag{3.21}$$

If we apply a random function h to the first registers x_b of $|\psi_1\rangle$ before applying U_{OTP} and then un-compute it, we get the following state:

$$|\psi_2'\rangle \coloneqq \frac{1}{\sqrt{2}} \sum_{b\in\{0,1\}} |x_b, h(x_b) \oplus r\rangle \,. \tag{3.22}$$

Continue with the Hadamard transform as above yields:

$$|\psi_3\rangle \coloneqq \; 2^{-\frac{2m-1}{2}} \sum_{u,v} \delta_{u\cdot(x_0\oplus x_1),v\cdot(h(x_0)\oplus h(x_1))}(-1)^{x_0\cdot u\oplus(h(x_0)\oplus r)\cdot v} \,|u,v\rangle \,. \tag{3.23}$$

Measuring $|\psi_3'\rangle$ yields a random pair (u, v) such that $u \cdot (x_0 \oplus x_1) = v \cdot (h(x_0) \oplus h(x_1))$ where $h(x_b)$ are random m-bit strings. Thus, Equation (3.21) satisfies with probability at most $\frac{1}{2}$. It is now easy to see that:

Lemma 3.8. *$\mathcal{SE}$ is* qIND-qCPA *insecure.*

Proof. In the challenge phase, the adversary $\mathcal{A}$ chooses two fixed messages x_0, x_1, and prepares the following state as its challenge:

$$|\psi\rangle \coloneqq \frac{1}{\sqrt{2}} \sum_b |x_b\rangle\,|0\rangle_R\,|0\rangle_F\,|+\rangle_G \,. \tag{3.24}$$

The challenge ciphertext state will be:

$$|\psi_0\rangle = \frac{1}{\sqrt{2}} \sum_b |x_b\rangle\,|r\rangle_F\,|x_b \oplus F(\mathsf{k}_1, r)\rangle_F\,|+\rangle_G \;\text{ if } b = 0, \tag{3.25}$$

or

$$|\psi_1\rangle = \frac{1}{\sqrt{2}} \sum_b |x_b\rangle\,|r\rangle_R\,|h(x_b) \oplus F(\mathsf{k}_1, r)\rangle_F\,|+\rangle_G \;\text{ if } b = 1. \tag{3.26}$$

Since r is a classical value, $\mathcal{A}$ can discard two registers R and G, which are separate from the others. $\mathcal{A}$ then applies the Fourier sampling (i.e., Hadamard transform followed by a measurement as described above), and outputs 1 if Equation (3.21) is satisfied, otherwise it outputs 0. We have $\Pr\left[\mathsf{Expt}_{\mathcal{SE}}^{\textit{qind-qcpa}-1}(\lambda, \mathcal{A}) = 1\right] = 1$ and $\Pr\left[\mathsf{Expt}_{\mathcal{SE}}^{\textit{qind-qcpa}-0}(\lambda, \mathcal{A}) = 1\right] \leq \frac{1}{2}$, thus $\mathsf{Adv}_{\mathcal{A},\mathcal{SE}}^{\textit{qind-qcpa}}(\lambda) \geq \frac{1}{2}$, which is certainly not negligible. □

3.3.3 Feasibility of Quantum CCA2 Security

The classical Encrypt-then-MAC paradigm [BN08] shows that an IND-CPA secure symmetric encryption scheme can be made IND-CCA2 secure if combined with an EUF-CMA MAC scheme. However, it is not obvious how to prove security in the quantum setting, as the reduction algorithm has no way to tell which ciphertexts the adversary received as the result of an encryption query in the learning phases, and no way to decrypt the ciphertexts if it has received them. To remedy these problems, we choose a specific type of MAC scheme in the construction (that is, any quantum-secure PRF) and leave the general security proof as an open question. The encryption scheme can be instantiated with any qIND-qCPA encryption scheme. In the proof, we simulate the MAC with random oracle and use Zhandry's compressed oracles technique to efficiently check if the adversary has seen a particular ciphertext as a result of an encryption query, and to decrypt in this case.

Construction 3.2 — Encrypt-then-MAC

Let $\mathcal{SE} := \langle \mathcal{K}, \mathsf{SymEnc}, \mathsf{SymDec} \rangle$ be a symmetric encryption scheme and qPRF be a family of quantum-secure pseudorandom functions. A composition of base schemes $\mathcal{SE}$ and qPRF is the symmetric encryption scheme $\mathcal{SE}' := \langle \mathcal{K}', \mathsf{SymEnc}', \mathsf{SymDec}' \rangle$ whose constituent algorithms are defined as follows.

$\mathcal{K}'(1^\lambda)$:

1 : $\mathsf{k}_1 \xleftarrow{\$} \mathcal{K}(1^\lambda)$
2 : $\mathsf{k}_2 \xleftarrow{\$} \{0,1\}^\lambda$
3 : **return** $\mathsf{k}_1 \parallel \mathsf{k}_2$

$\mathsf{SymEnc}'(\mathsf{k}_1 \parallel \mathsf{k}_2, x)$:

1 : $c \leftarrow \mathsf{SymEnc}(\mathsf{k}_1, x)$
2 : $\tau \leftarrow \mathsf{qPRF}(\mathsf{k}_2, c \| x)$
3 : **return** $c \parallel \tau$

$\mathsf{SymDec}'(\mathsf{k}_1 \parallel \mathsf{k}_2, c \parallel \tau)$:

1 : $x \leftarrow \mathsf{SymDec}(\mathsf{k}_1, c)$
2 : **if** $\mathsf{qPRF}(\mathsf{k}_2, c \| x) \neq \tau$ **then**
3 : **return** $\perp$
4 : **return** x

Theorem 3.3. *Let $\mathcal{SE}$ be a* qIND-qCPA *secure symmetric encryption scheme. Let* qPRF *be a family of quantum-secure pseudorandom functions. Then the encryption scheme $\mathcal{SE}'$ defined in Construction 3.2 is* qIND-qCCA2 *secure.*

Remark 3.3. As shown in [Zha12a], quantum-secure PRFs can be constructed from quantum-secure one-way functions. In addition, [GHS16; CETU21] shows how to construct qIND-qCPA secure encryption schemes from quantum-secure pseudorandom permutations.

Proof. We proceed using hybrid games. Let $\mathcal{A}$ be a QPT adversary. For any game G_{index}, we denote by $\Pr[G_{\mathsf{index}}] := |\Pr[G_{\mathsf{index}}(\mathcal{A}) = 1 \mid b = 1] - \Pr[G_{\mathsf{index}}(\mathcal{A}) = 1 \mid b = 0]|$. Also, by event $G_{\mathsf{index}}(\mathcal{A})$, we mean the output of the experiments (defined as in Definition 3.2) in game G_{index} when interacting with $\mathcal{A}$.

Game G_0: This is the standard attack game. In what follows, let $\mathsf{k} := \mathsf{k}_1 \| \mathsf{k}_2 \leftarrow \mathcal{K}'()$.

Game G_1: This is identical to G_0, except that we use, in the role of qPRF, a random function H. (Imagine that the reduction has oracle access to H and uses it in the role of qPRF, notice that the key k_2 is chosen uniformly at random and hidden from $\mathcal{A}$.) By security of qPRF, we have that $|\Pr[G_1] - \Pr[G_0]| \leq \mathsf{negl}(\lambda)$.

Game G_2: This is identical to G_1, except that now we consider H as being implemented in Zhandry's compressed standard random oracle.

We give a description of how this compressed random oracle is implemented. A query to the compressed random oracle is in (superposition of) the form $|c\|x\rangle\,|y_2\rangle$ where the second register is the second part of the adversary's response registers. Furthermore, since the compressed oracle for the encryption makes only *one black-box call* to the unitary implementing the encryption algorithm for each challenge query, there is also only a single call to the compressed random oracle for each challenge query.

Let E denote the database of this compressed random oracle. In more details, E will be a collection of $(c\|x, \tau)$ pairs. For a pair (c, τ) such that there is a pair $(c\|x, \tau) \in E$, we call $w = x$ an *associated input* of (c, τ).

Since the compressed random oracle is equivalent to the standard random oracle, this does not affect the adversary's success probability. We have that $\Pr[G_2] = \Pr[G_1]$.

Game G_3: This is identical to G_2, except now we make the following modification to the decryption oracle in the random world.

In the following, we take the notation as in Lemma 2.6. We define the relation R_c^H to be the set of all (x, τ) such that $x = \mathsf{SymDec}(c) \wedge H(c\|x) = \tau$. Since SymEnc has perfect correctness, for each c there is only one decryption x, and thus $\Gamma_{R_c^H} = 1$. Given the relation R_c^H, the projectors Σ_c^x for $x \in \mathcal{X}$ and $\Sigma_c^\perp$ are defined as in Lemma 2.6. Now the measurement $\mathbb{M} := \{\Sigma_c^x\}_{x\in\mathcal{X}\cup\{\perp\}}$ checks whether there exists a pair in the database E satisfying the relation R_c^H or not. (Note that there is at most one pair satisfying the relation for each c.) Let $\mathbb{M}_{E,P}^c$ be the following purified measurement corresponding to $\mathbb{M}$:

$$\mathbb{M}_{E,P}^c \left|y, z\right\rangle \to \sum_{x\in\mathcal{X}\cup\{\perp\}} \Sigma_c^x \left|y\right\rangle_E \left|z \oplus x\right\rangle_P ,$$

where E is the registers of the database E and P is some ancilla registers. We define the unitary $\mathbb{M}_{E,P}$ that operates on the ciphertext, the registers E and P as:

$$\mathbb{M}_{E,P} \left|c\|\tau\right\rangle \left|y, z\right\rangle_{E,P} \to \left|c\|\tau\right\rangle \otimes \mathbb{M}_{E,P}^c \left|y, z\right\rangle_{E,P} .$$

Note that $\mathbb{M}_{E,P}$ is an evolution.

The modification is as follows. For each decryption query, if $\mathsf{FindImage}_D$ returns $(0, 0)$ the decryption oracle in the random world $\mathcal{R}^3_{\mathsf{SymDec}'(\mathsf{k},\cdot)}$ first applies the unitary $\mathbb{M}_{E,P}$ with the ancilla register P initialized to 0. Then it executes $\mathsf{SymDec}'(\mathsf{k}, \cdot)$. Finally it applies $\mathbb{M}_{E,P}$ again.

$$\mathcal{R}^3_{\mathsf{SymDec}'(\mathsf{k},\cdot)} \left|y, z\right\rangle \left|D\right\rangle \left|E\right\rangle := \begin{cases} \mathbb{M}_{E,P} \circ \mathsf{SymDec}' \circ \mathbb{M}_{E,P} \left|y, z\right\rangle \left|D\right\rangle \left|E\right\rangle & \text{if } \mathsf{FindImage}_D(y, D) = (0, 0), \\ \left|y, z \oplus w\right\rangle \left|D\right\rangle \left|E\right\rangle & \text{if } \mathsf{FindImage}_D(y, D) = (1, w), \end{cases}$$

By Lemma 2.6, we have that $\mathbb{M}_{E,P}$ and SymDec' almost commute: their commutator is bounded by $8\sqrt{2} \cdot 2^{-\frac{\ell}{2}}$, where ℓ is the output's length of qPRF, which is polynomial in the security parameter. Hence, we can swap $\mathbb{M}_{E,P}$ and SymDec' in the implementation of $\mathcal{R}^3_{\mathsf{SymDec}'(\mathsf{k},\cdot)}$ and recover $\mathcal{R}^2_{\mathsf{SymDec}'(\mathsf{k},\cdot)}$ (note that $\mathbb{M}_{E,P}$ is an evolution). The distinguishing probability of this modification is exactly the commutator's bound, and thus the two hybrids are distinguishable to the adversary with probability at most $8\sqrt{2} \cdot 2^{-\frac{\ell}{2}}$. This shows that $|\Pr[G_3] - \Pr[G_2]| \leq \mathsf{negl}(\lambda)$.

Game G_4: This is identical to G_3, except we change how decryption queries are answered in the random world when $\mathsf{FindImage}_D$ returns $(0, 0)$. In this case, the oracle first applies $\mathbb{M}_{E,P}$ with the register P initialized to 0. Then, it XORs the value of P to the response register. It finally applies $\mathbb{M}_{E,P}$ again.

$$\mathcal{R}^4_{\mathsf{SymDec}'(\mathsf{k},\cdot)} \left|y, z\right\rangle \left|D\right\rangle \left|E\right\rangle := \begin{cases} \mathbb{M}_{E,P} U_{\mathsf{k}} \mathbb{M}_{E,P} \left|y, z\right\rangle \left|D\right\rangle \left|E\right\rangle & \text{if } \mathsf{FindImage}_D(y, D) = (0, 0), \\ \left|y, z \oplus w\right\rangle \left|D\right\rangle \left|E\right\rangle & \text{if } \mathsf{FindImage}_D(y, D) = (1, w), \end{cases}$$

where U_{k} is defined as

$$U_{\mathsf{k}} \left|y, z\right\rangle \left|D\right\rangle \left|E\right\rangle_E \left|x\right\rangle_P := \begin{cases} \left|y, z \oplus \perp\right\rangle \left|D\right\rangle \left|E\right\rangle_E \left|x\right\rangle_P & \text{if } x = \perp, \\ \left|y, z \oplus x\right\rangle \left|D\right\rangle \left|E\right\rangle_E \left|x\right\rangle_P & \text{if } x \neq \perp. \end{cases}$$

The only difference between G_4 and G_3 is the definition of $\mathcal{R}_{\mathsf{SymDec}'(\mathsf{k},\cdot)}$, and that the adversary can distinguish between the two hybrids if and only if it sends a query in which $y \in E \setminus D$ or $y \notin D \cup E$ with non-negligible weight. This is because if $y \in D$ then in both hybrids, the original plaintext x was returned.

(i) We first consider the case in which $y \in E \setminus D$. First, we notice that for challenge queries, the challenger only queries the compressed random oracle H if it can record the query (see Figure 3.1). Thus, if $y \in E \setminus D$, in G_3, SymDec′ is used to answer the queries, while in G_4, $\mathbb{M}_{E,P}$ is used to answer the queries. However, SymDec′ and $\mathbb{M}_{E,P}$ return the same output in this case.

(ii) Consider the case in which $y \notin D \cup E$. The query is answered with $\perp$ in G_4 and using SymDec′ in G_3. If SymDec′ returns $x \neq \perp$, then the two hybrids are distinguishable. However, this means that the adversary must be able to procedure a ciphertext $c\|\tau \notin E$ such that $\mathsf{SymDec}'(\mathsf{k}, c\|\tau) \neq \perp$. Since the underlying encryption scheme $\mathcal{SE}$ never outputs $\perp$, it means that the adversary was able to produce a pair $(\widetilde{c}\|\widetilde{x}, \widetilde{\tau}) \notin E$ such that $H(\widetilde{c}\|\widetilde{x}) = \widetilde{\tau}$. Let Forge be the event that at least one of the decryption queries in the random world contains some pairs $(\widetilde{y} = \widetilde{c}\|\widetilde{\tau})$ with overall non-negligible weight such that $H(\widetilde{c}\|\widetilde{x}) = \widetilde{\tau} \notin E$. $\mathcal{A}$ could distinguish the two games if and only if $\Pr[\mathsf{Forge}]$ is non-negligible. We construct a QPT adversary $\mathcal{C}$ from $\mathcal{A}$ that breaks Lemma 2.7 if Forge happens with non-negligible probability. Assume that the adversary $\mathcal{A}$ makes at most q queries to the random oracle (by making queries to the encryption and decryption oracle). $\mathcal{C}$ runs $\mathcal{A}$ as its subroutine, and randomly measures one of $\mathcal{A}$'s decryption queries in the second phase. $\mathcal{C}$ would then obtain a pair that is not in E with probability p in Lemma 2.7 such that $p = \frac{\Pr[\mathsf{Forge}]}{q}$. Since the obtained pair is not in E, the probability p' is 0. If $\Pr[\mathsf{Forge}]$ is non-negligible, p is also non-negligible. This breaks the bound given in Lemma 2.7.

Overall, the two hybrids are identical except for the last case in which $y \notin D \cup E$. However, in this case, the two hybrids are distinguishable with at most negligible probability. This shows that: $|\Pr[G_4] - \Pr[G_3]| \leq \mathsf{negl}(\lambda)$.

Game G_5: We define a procedure $\mathsf{FindImage}_E$ over the database E similar to the one defined over D, except that it takes as input a pair $((c,\tau), E)$, searches over the database E and returns $(1, w)$ where w is the associated input of (c,τ) if $\tau \in E$ and $(0,0)$ otherwise. This hybrid is identical to G_4, except that in the random world, the decryption oracle in the second phase $\mathcal{R}_{\mathsf{SymDec}'(\mathsf{k},\cdot)}$ is implemented as follows.

$$\mathcal{R}^5_{\mathsf{SymDec}'(\mathsf{k},\cdot)}\left|y,z\right\rangle\left|D\right\rangle\left|E\right\rangle \coloneqq \begin{cases} \left|y, z \oplus w\right\rangle\left|D\right\rangle\left|E\right\rangle \text{ if } & \mathsf{FindImage}_D(y,D) = (1,w), \\ \left|y, z \oplus \perp\right\rangle\left|D\right\rangle\left|E\right\rangle \text{ if } & \mathsf{FindImage}_D(y,D) = (0,0) \\ & \wedge\ \mathsf{FindImage}_E(y,E) = (0,0), \\ \left|y, z \oplus w'\right\rangle\left|D\right\rangle\left|E\right\rangle \text{ if } & \mathsf{FindImage}_D(y,D) = (0,0) \\ & \wedge\ \mathsf{FindImage}_E(y,E) = (1,w'). \end{cases}$$

By the correctness of $\mathcal{SE}'$, it must be the case that if $\mathsf{FindImage}_E$ returns $(1,w')$, $\mathbb{M}_{E,P}$ should have the same w' in the register P. Similarly, if $\mathsf{FindImage}_E$ returns $(0,0)$, $\mathbb{M}_{E,P}$ should have $\perp$ in P. Therefore, these two hybrids are identical, that is $\Pr[G_5] = \Pr[G_4]$.

Game G_6: This is identical to G_5, except that in the real world, the decryption oracle in the second phase $\mathcal{O}_{\mathsf{SymDec}'(\mathsf{k},\cdot)}$ is implemented as follows.

$$\mathcal{O}^6_{\mathsf{SymDec}'(\mathsf{k},\cdot)}\left|y,z\right\rangle\left|E\right\rangle := \begin{cases} \left|y,z\oplus\perp\right\rangle\left|E\right\rangle & \text{if } \mathsf{FindImage}_E(y,E)=(0,0),\\ \left|y,z\oplus w\right\rangle\left|E\right\rangle & \text{if } \mathsf{FindImage}_E(y,E)=(1,w).\end{cases}$$

We can consider intermediate hybrids that make similar changes as in G_{3-6} for the real world. By the same argument (except that now we do not have the database D in the real world, hence the arguments in G_4 is simplified), we also have that $|\Pr[G_6]-\Pr[G_5]|\leq \mathsf{negl}(\lambda)$.

Note that from this hybrid, the decryption algorithm SymDec' is no longer needed.

Game G_7: This is identical to G_6, except that in the random world, for each challenge encryption query, instead of applying the random oracle H on $c\|x'$ (where $c=\mathsf{SymEnc}(\mathsf{k}_1,x')$ for some random $x'\in\mathcal{X}$), we apply H on $c\|x$, where x is the original plaintext. Formally, the challenge encryption oracle $\mathcal{R}_{\mathsf{SymEnc}'}$ implements the following mapping:

$$\left|x,y_1\|y_2\right\rangle\mapsto\left|x,(y_1\oplus c)\|(y_2\oplus H(c\|x))\right\rangle\otimes\left|x,x',c\|H(c\|x)\right\rangle_D,$$

where $c=\mathsf{SymEnc}(\mathsf{k}_1,x')$ for some random $x'\in\mathcal{X}$.

We note that the implementation of this oracle is similar to its implementation in G_6 (as described in Figure 3.1), except that the unitary U_{f_2} (corresponding to the unitary of the encryption procedure) acts on four registers (including the original plaintext register): instead of using $c\|x'$ (for some random x' where $c=\mathsf{SymEnc}(\mathsf{k}_1,x')$) as the input to the compressed random oracle H, we use $c\|x$ (where x comes from the adversary's input registers).

Precisely, the function f_2 in Figure 3.1 (which denotes the encryption algorithm) is implemented (privately in the oracle's side) as follows.

$$U_{f_2}\left|x\right\rangle\left|x'\right\rangle\left|r\right\rangle\left|0\right\rangle\mapsto\left|x\right\rangle\left|x'\right\rangle\left|r\right\rangle\left|c\|H(c\|x')\right\rangle \text{ for } c=\mathsf{SymEnc}(\mathsf{k}_1,x';r) \text{ in } G_6,$$

and

$$U_{f_2}\left|x\right\rangle\left|x'\right\rangle\left|r\right\rangle\left|0\right\rangle\mapsto\left|x\right\rangle\left|x'\right\rangle\left|r\right\rangle\left|c\|H(c\|x)\right\rangle \text{ for } c=\mathsf{SymEnc}(\mathsf{k}_1,x';r) \text{ in } G_7.$$

The difference between G_6 and G_7 is that in G_6, the adversary receives $H(c\|x')$ for some random x' while in G_7, the adversary receives $H(c\|x)$ where x is its input and c is an encryption of some random plaintext. Since H is a random oracle, the two distributions are perfectly indistinguishable. Thus $\Pr[G_7]=\Pr[G_6]$.

Game G_8: This is identical to G_7, except that we remove the uses of the database D in the decryption oracle in the random world. In particular, the decryption oracle in the random world $\mathcal{R}_{\mathsf{SymDec}'}$ is implemented as follows:

$$\mathcal{R}^8_{\mathsf{SymDec}'(\mathsf{k},\cdot)}\left|y,z\right\rangle\left|E\right\rangle := \begin{cases} \left|y,z\oplus\perp\right\rangle\left|E\right\rangle & \text{if } \mathsf{FindImage}_E(y,E)=(0,0),\\ \left|y,z\oplus w\right\rangle\left|E\right\rangle & \text{if } \mathsf{FindImage}_E(y,E)=(1,w).\end{cases}$$

The adversary can distinguish between the two hybrids if and only if it sends a query in which $y\in D\setminus E$ with non-negligible weight. This is because if $y\in D\cap E$ then in both

hybrids, the original plaintext x was returned. In the case of $y \in D \setminus E$, the query is answered with $\perp$ in G_8 and $x \neq \perp$ in G_7. However if the adversary can obtain $x \neq \perp$ in G_7 for this case, it means that it has obtained a pair $(\tilde{c}, \tilde{\tau})$ such that $\tilde{\tau} \notin E$ and $H(\tilde{c} \| \tilde{x}) = \tilde{\tau}$ for some $\tilde{x}$.

By a similar argument as in Item (ii), any distinguisher for these two hybrids can be used to construct an adversary breaking the bound given in Lemma 2.7. Thus, we also have that $|\Pr[G_8] - \Pr[G_7]| \leq \mathsf{negl}(\lambda)$.

We note that in this final hybrid, the database D is no longer needed. Furthermore, the advantage of $\mathcal{A}$ in this hybrid can be reduced to its advantage against $\mathcal{SE}$. To see that, we construct a QPT adversary $\mathcal{B}$ from $\mathcal{A}$ as follows: $\mathcal{B}$ runs $\mathcal{A}$ as its subroutine. For each encryption or challenge query, $\mathcal{B}$ implements the compressed random oracle for the MAC. It first sends the plaintext registers to its challenger and receives back a ciphertext, it then tags the received ciphertext and the plaintext registers with the MAC using its compressed random oracle and forwards them to $\mathcal{A}$.

Notice in this hybrid, $\mathcal{B}$ can always answer decryption queries, without needing to query to $\mathcal{SE}$ decryption oracle, by using its own compressed random oracle's database. The advantage of $\mathcal{B}$ against $\mathcal{SE}$ is exactly the advantage of $\mathcal{A}$ in this hybrid, showing that $\Pr[G_8] \leq \mathsf{Adv}^{qind\text{-}qcpa}_{\mathcal{A},\mathcal{SE}}(\lambda)$. Putting everything together, we finish the proof of the theorem. □

3.4 Quantum-Secure Public-key Encryption

3.4.1 Definitions of Security

Indistinguishability Security

The indistinguishability notions can be defined analogously to the ones given in Section 3.3. We define a real-or-random oracle allowing quantum queries and the decryption oracle in the second learning phase as follows.

$$\mathcal{RR}(b) = \begin{cases} \mathcal{O}_{\mathsf{Enc}(\mathsf{pk},\cdot)} & \text{if } b = 1 \\ \mathcal{R}_{\mathsf{Enc}(\mathsf{pk},\cdot)} & \text{if } b = 0 \end{cases}, \text{ and } \mathcal{DEC}(b) = \begin{cases} \mathcal{O}_{\mathsf{Dec}(\mathsf{sk},\cdot)} & \text{if } b = 1 \\ \mathcal{R}_{\mathsf{Dec}(\mathsf{sk},\cdot)} & \text{if } b = 0 \end{cases}.$$

Definition 3.3 — Quantum Indistinguishability for Public-key Encryption

Let $\mathcal{E} := \langle \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec} \rangle$ be a public-key encryption scheme and let $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$ be a quantum adversary. For $qatk \in [qcpa, qcca1, qcca2]$, we define the following game, where the oracles $\mathcal{O}_1, \mathcal{O}_2$ are defined according to $qatk$:

Experiment $\mathsf{Expt}^{qind\text{-}qatk-b}_{\mathcal{E}}(\lambda, \mathcal{A})$:

1 : $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$
2 : $|\Phi\rangle \leftarrow \mathcal{A}_1^{|\mathcal{O}_1\rangle}(\mathsf{pk})$
3 : $b' \leftarrow \mathcal{A}_2^{|\mathcal{RR}(b)\rangle, |\mathcal{O}_2\rangle}(|\Phi\rangle)$
4 : **return** b'

$qatk$	Oracle $\mathcal{O}_1$	Oracle $\mathcal{O}_2$
$qcpa$	$\varnothing$	$\varnothing$
$qcca1$	$\mathcal{O}_{\mathsf{Dec}(\mathsf{sk},\cdot)}$	$\varnothing$
$qcca2$	$\mathcal{O}_{\mathsf{Dec}(\mathsf{sk},\cdot)}$	$\mathcal{DEC}(b)$

We define $\mathcal{A}$'s advantage by

$$\mathsf{Adv}^{qind\text{-}qatk}_{\mathcal{A},\mathcal{E}}(\lambda) := \left|\Pr\left[\mathsf{Expt}^{qind\text{-}qatk-1}_{\mathcal{E}}(\lambda,\mathcal{A})=1\right] - \Pr\left[\mathsf{Expt}^{qind\text{-}qatk-0}_{\mathcal{E}}(\lambda,\mathcal{A})=1\right]\right|.$$

We say $\mathcal{E}$ is secure in the sense of qIND-qATK if $\mathcal{A}$ being QPT implies that $\mathsf{Adv}^{qind\text{-}qatk}_{\mathcal{A},\mathcal{E}}(\lambda)$ is negligible.

Similarly as in Section 3.3, our definitions, restricted to classical challenge queries, are equivalent to Boneh-Zhandry's notions (IND-qATK). Furthermore, the following theorem shows that our notions are closed under composition.

Theorem 3.4. *An encryption scheme $\mathcal{E}$ is many-message* qIND-qATK *secure iff it is single-message* qIND-qATK *secure.*

Proof. The forward implication follows directly. For the reverse direction, we use the standard hybrid argument that uses an adversary $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$ with advantage ε to construct a new adversary $\mathcal{B} := (\mathcal{B}_1, \mathcal{B}_2)$ which breaks the single-message security with advantage ε/q^2.

Define a sequence of games $G_0, \ldots, G_q$ in which $\mathcal{B}$ runs $\mathcal{A}$ and returns $\mathcal{A}$'s output as follows: For any game G_i,

1. $\mathcal{B}_1$ simulates $\mathcal{A}$'s $i-1$ first challenge queries on its own, as in the experiment $\mathsf{Expt}^{qind\text{-}qatk-1}_{\mathcal{E}}$.
2. $\mathcal{B}$ uses $\mathcal{A}$'s i-th challenge query as its challenge query.
3. $\mathcal{B}_2$ simulates all $\mathcal{A}$'s other challenge queries on its own using the compressed encryption oracle, as in the experiment $\mathsf{Expt}^{qind\text{-}qatk-0}_{\mathcal{E}}$, obtaining a database D.

In the case of CCA2 security, $\mathcal{B}_2$ uses a slightly different decryption oracle in the second phase as follows. Let $\mathcal{R}'_{\mathsf{Dec}(\mathsf{sk},\cdot)}$ be the decryption oracle of $\mathcal{B}_2$ and $\mathcal{R}_{\mathsf{Dec}}$ be $\mathcal{B}_2$ simulated decryption oracle for $\mathcal{A}$. Then

$$\mathcal{R}_{\mathsf{Dec}}\left|y,z\right\rangle\left|D\right\rangle = \begin{cases} \left(\mathcal{R}'_{\mathsf{Dec}(\mathsf{sk},\cdot)}\left|y,z\right\rangle\right)\left|D\right\rangle & \text{if } \mathsf{FindImage}(y,D) = (0,0^m), \\ \left|y,z\oplus w\right\rangle\left|D\right\rangle & \text{if } \mathsf{FindImage}(y,D) = (1,w). \end{cases}$$

Note that $G_0 = \mathsf{Expt}^{qind\text{-}qatk-1}_{\mathcal{E}}(\lambda,\mathcal{A})$ and $G_q = \mathsf{Expt}^{qind\text{-}qatk-0}_{\mathcal{E}}(\lambda,\mathcal{A})$. Because $\mathcal{A}$ is able to distinguish $\mathsf{Expt}^{qind\text{-}qatk-1}_{\mathcal{E}}$ from $\mathsf{Expt}^{qind\text{-}qatk-0}_{\mathcal{E}}$, there exists some $g \in [\![1,q]\!]$ such that $\mathcal{A}$ distinguishes G_g from G_{g+1} with advantage at least ε/q. $\mathcal{B}$ can guess g correctly with probability $1/q$, thus $\mathcal{B}$'s overall advantage in breaking the single-message security is ε/q^2. □

We also give a separation construction showing that our notions are strictly stronger than Boneh-Zhandry's notions. The idea is to install a backdoor that only a quantum adversary can use, by doing some quantum computation. We need to ensure that the backdoor is useless even if the adversary has quantum access to the decryption oracle in the learning phases. Our construction follows the hybrid encryption paradigm combining a CCA2-secure public-key encryption and a one-time CCA2-secure symmetric encryption. The attack is similar in spirit to that for symmetric encryption.

Theorem 3.5. *If there exists an encryption scheme $\mathcal{E}$ which is* IND-qCCA2 *secure against* QPT *adversaries, then there exists an encryption scheme $\mathcal{E}'$ which is also* IND-qCCA2 *secure, but* qIND-qCPA *insecure.*

Proof. Assume there exists some IND-qCCA2 secure encryption scheme $\mathcal{E} := \langle \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec} \rangle$. Let $\mathcal{H} := \{h_k\}_k$ be a family of pairwise independent hash functions with the key space $\mathcal{K}$. The new encryption scheme $\mathcal{E}' := \langle \mathsf{KeyGen}', \mathsf{Enc}', \mathsf{Dec}' \rangle$ is defined as follows.

$\mathsf{KeyGen}'(1^\lambda)$:

1 : $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{KeyGen}\left(1^\lambda\right)$
2 : **return** $(\mathsf{pk}, \mathsf{sk})$

$\mathsf{Enc}'(\mathsf{pk}, x)$:

1 : $r \xleftarrow{\$} \mathcal{X}, k \xleftarrow{\$} \mathcal{K}$
2 : $c_1 \leftarrow \mathsf{Enc}(\mathsf{pk}, r\|k)$
3 : $c_2 \leftarrow x \oplus r$
4 : $\sigma \leftarrow h_k(c_2)$
5 : **return** $c_1\|c_2\|\sigma$

$\mathsf{Dec}'(\mathsf{sk}, c_1\|c_2\|\sigma)$:

1 : $r\|k \leftarrow \mathsf{Dec}(\mathsf{sk}, c_1)$
2 : **if** $h_k(c_2) \neq \sigma$ **then**
3 : **return** $\perp$
4 : $x \leftarrow c_2 \oplus r$
5 : **return** x

The proof is completed by establishing that $\mathcal{E}'$ is IND-qCCA2 secure but vulnerable to a qIND-qCPA attack.

Lemma 3.9. *$\mathcal{E}'$ is* IND-qCCA2 *secure.*

Proof. Fix the adversary $\mathcal{A}$ and λ. For the purpose of this separation, it is sufficient to assume that $\mathcal{E}$ is perfectly correct. We prove security through a sequence of games. Let $\Pr[G_i]$ be the probability the adversary wins game G_i.

Game G_0: This is the standard attack game. Let the challenge ciphertext be (c_1^*, c_2^*, σ^*), and $K^* = (r^*, k^*)$ be the randomness used during the encryption process. Then, the decryption oracle in the second phase can be written as $\mathsf{Dec}'^*(\mathsf{sk}, \cdot)$ which rejects decryption when the ciphertext is (c_1^*, c_2^*, σ^*). We denote the set of ciphertexts that will be rejected by Dec'^* is D_0. In this game, $D_0 = \{(c_1^*, c_2^*, \sigma^*)\}$.

Game G_1: This is identical to G_0, except that whenever a ciphertext $(c_1^*, \cdot, \cdot) \in D_0$ is submitted to the decryption oracle in the second phase, the decryption oracle does not apply $\mathsf{Dec}(\mathsf{sk}, c_1^*)$, but instead uses K^* produced in the challenge phase to perform steps $2 - 5$.

This change is just conceptual, since we assume that $\mathcal{E}$ is perfectly correct. Thus, $\Pr[G_1] = \Pr[G_0]$.

Game G_2: This is identical to G_1, but now the challenger computes c_1^* by encrypting a completely random value $K^+ = (r^+, k^+)$ instead of K^*. That is, $c_1^* = \mathsf{Enc}(\mathsf{pk}, r^+\|k^+)$, but $c_2^* = x \oplus r^*$ and $\sigma^* = h_{k^*}(c_2^*)$.

Notice that in games G_1 and G_2, the ciphertext c_1^* need not be submitted for decryption. We show how to turn any distinguisher $\mathcal{A}$ of games G_1 and G_2 into an adversary $\mathcal{A}'$ against the security of the underlying scheme $\mathcal{E}$: $\mathcal{A}'$ runs $\mathcal{A}$ using its oracles to answer $\mathcal{A}$, outputs (K^*, K^+) as its challenge pair. Finally, $\mathcal{A}'$ outputs whatever $\mathcal{A}$ outputs. It is easy to see that we have:

$$|\Pr[G_2] - \Pr[G_1]| \leq \mathsf{Adv}^{ind\text{-}qcca2}_{\mathcal{A}',\mathcal{E}}(\lambda).$$

Game G_3: We further modify G_2 and now change the set D_0 to be $D_0 := \{(c_1^*, \cdot, \cdot)\}$. In other words, it rejects any ciphertext (c_1, c_2, σ) such that $c_1 = c_1^*$.

Let Forge be the event that some ciphertext is rejected in game G_3, but would not have been rejected in the game G_2. Since games G_2 and G_3 are identical until event Forge, we have $|\Pr[G_3] - \Pr[G_2]| \leq \Pr[\text{Forge}]$.

Notice that in the construction of $\mathcal{E}'$, the use of pairwise independent hash functions acts as a one-time secure message authentication code, thus $\Pr[\text{Forge}] = 0$.

In this final game, the component c_2^* is one-time padded of the message x_b^* using a random string r^* chosen uniformly and independently of all other variables, including b. Thus, $\Pr[G_3] = 0$.

By the security of the underlying building blocks, we have the security of $\mathcal{E}'$. □

Lemma 3.10. *$\mathcal{E}'$ is* qIND-qCPA *insecure.*

Proof. In the challenge phase, the adversary $\mathcal{A}$ chooses two fixed messages x_0, x_1, and prepares the following state as its challenge:

$$|\psi\rangle := \frac{1}{\sqrt{2}} \sum_b |x_b\rangle |+\rangle |0\rangle |+\rangle .$$

The challenge ciphertext state will be:

$$|\psi_0\rangle = \frac{1}{\sqrt{2}} \sum_b |x_b\rangle |+\rangle |x_b \oplus r\rangle |+\rangle \text{ if } b = 0,$$

or

$$|\psi_1\rangle = \frac{1}{\sqrt{2}} \sum_b |x_b\rangle |+\rangle |h(x_b) \oplus r\rangle |+\rangle \text{ if } b = 1.$$

$\mathcal{A}$ then applies the Fourier sampling (as described in Section 3.3.2) and breaks the security of $\mathcal{E}'$ with non-negligible probability. □

Non-Malleability Security

Intuitively, the classical definitions [BDPR98; BS06] involve having an adversary play a challenge-response game. In the challenge phase, the adversary is given an encryption y of a message x it produced itself. It must then output a vector of ciphertexts $\vec{y}$ (whose components can be y - in this case, the decryption returns $\perp$) called *adversarial ciphertexts*, together with an arbitrary string. The security definitions require that the distribution of the adversary's output and *the decryptions of the adversarial ciphertexts* is indistinguishable from the distribution when the adversary receives an encryption of some random message $\tilde{x}$ instead of x. The non-malleability property can be established by saying that when an encryption of x given to the adversary is replaced with an encryption of a random $\tilde{x}$, even the *contents* of encryption messages that the adversary sends cannot change in any computationally noticeable way.

A closer look at the adversarial ciphertexts distribution gives us different classical definitions, which leads to different composability properties. As pointed out by Pass, shelat and Vaikuntanathan [PsV07], indistinguishability-based definitions of encryption

may or may not compose in the context of non-malleability, depending on how we treat an "invalid adversary" that outputs *invalid* ciphertexts as part of its adversarial output. In the quantum setting, the adversary can output a superposition of adversarial ciphertexts, which might include invalid ciphertexts, even if it is "hard" to generate invalid ciphertexts. This leaves us no choice but to incorporate invalid adversaries into the definitions. The definitions given here are syntactically close to the classical definitions of [BS06, Definition 4.1].

Definition 3.4 — Quantum Non-Malleability for Public-key Encryption

Let $\mathcal{E} := \langle \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec} \rangle$ be a public-key encryption scheme and let $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ be a quantum adversary. For $qatk \in [qcpa, qcca1, qcca2]$ and $r \in \mathbb{N}$, we define the following game, where the oracles $\mathcal{O}_1, \mathcal{O}_2$ are defined according to $qatk$:

Experiment $\mathsf{Expt}_{\mathcal{E}}^{qnme\text{-}qatk-b}(\lambda, \mathcal{A})$:

1 : $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$

2 : $|\Psi_1\rangle \leftarrow \mathcal{A}_1^{|\mathcal{O}_1\rangle}(\mathsf{pk})$

3 : $|\Psi_2\rangle := \sum_{\vec{y},\vec{z}} \alpha_{\vec{y},\vec{z}} |\vec{y},\vec{z}\rangle |\phi_{\vec{y},\vec{z}}\rangle \leftarrow \mathcal{A}_2^{|\mathcal{RR}(b)\rangle,|\mathcal{O}_2\rangle}(|\Psi_1\rangle)$
where $|\vec{y}| = |\vec{z}| = r$

4 : $|\Psi_3\rangle \leftarrow \mathcal{R}_{\mathsf{Dec}(\mathsf{sk},\cdot)} |\Psi_2\rangle$

5 : $b' \leftarrow \mathcal{A}_3^{|\mathcal{O}_2\rangle}(|\Psi_3\rangle)$

6 : **return** b'

$qatk$	Oracle $\mathcal{O}_1$	Oracle $\mathcal{O}_2$
$qcpa$	$\varnothing$	$\varnothing$
$qcca1$	$\mathcal{O}_{\mathsf{Dec}(\mathsf{sk},\cdot)}$	$\varnothing$
$qcca2$	$\mathcal{O}_{\mathsf{Dec}(\mathsf{sk},\cdot)}$	$\mathcal{DEC}(b)$

We define $\mathcal{A}$'s advantage by

$$\mathsf{Adv}_{\mathcal{A},\mathcal{E}}^{qnme\text{-}qatk}(\lambda) := \left|\Pr\left[\mathsf{Expt}_{\mathcal{E}}^{qnme\text{-}qatk-1}(\lambda, \mathcal{A}) = 1\right] - \Pr\left[\mathsf{Expt}_{\mathcal{E}}^{qnme\text{-}qatk-0}(\lambda, \mathcal{A}) = 1\right]\right|.$$

We say $\mathcal{E}$ is secure in the sense of qNME-qATK if $\mathcal{A}$ being QPT implies that $\mathsf{Adv}_{\mathcal{A},\mathcal{E}}^{qnme\text{-}qatk}(\lambda)$ is negligible.

The following theorem shows that our notions are closed under composition.

Theorem 3.6. *An encryption scheme $\mathcal{E}$ is many-message* qNME-qATK *secure iff it is single-message* qNME-qATK *secure.*

The proof is similar to that of Theorem 3.4; we omit the details.

3.4.2 Relating Indistinguishability and Non-Malleability

A full characterization of fully-quantum indistinguishability and non-malleability notions is summarized in Figure 3.2. These results are identical as in the classical setting [BDPR98]. We use slightly modified constructions of [BDPR98]: the attacks carry in the classical manner, only the security proofs need to be adapted, which are given below.

Theorem 3.7 (qIND-qCCA1 $\not\Rightarrow$ qNME-qCPA). *If there exists an encryption scheme $\mathcal{E}$ that is* qIND-qCCA1 *secure, then there exists an encryption scheme $\mathcal{E}'$ that is* qIND-qCCA1 *secure but* qNME-qCPA *insecure.*

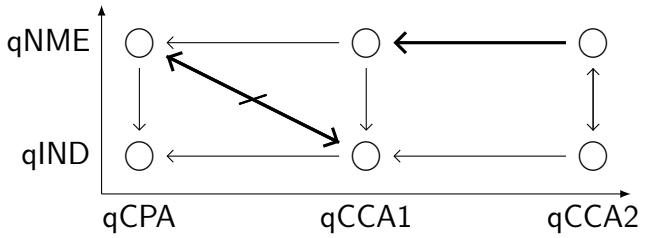

Figure 3.2: An arrow is an implication. There is a path from A to B if and only if A implies B. The bold arrows represent non-trivial separations we actually prove in this section.

Proof. Assume there exists some qIND-qCCA1 secure encryption scheme $\mathcal{E} := \langle \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec} \rangle$. The new encryption scheme $\mathcal{E}' := \langle \mathsf{KeyGen}', \mathsf{Enc}', \mathsf{Dec}' \rangle$ is defined as follows.

$\mathsf{KeyGen}'(1^\lambda)$:
1 : $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{KeyGen}\left(1^\lambda\right)$
2 : **return** $(\mathsf{pk}, \mathsf{sk})$

$\mathsf{Enc}'(\mathsf{pk}, x)$:
1 : $b \xleftarrow{\$} \{0,1\}$
2 : $y \leftarrow \mathsf{Enc}(\mathsf{pk}, x)$
3 : **return** $y \| b$

$\mathsf{Dec}'(\mathsf{sk}, y\|b)$:
1 : $x \leftarrow \mathsf{Dec}(\mathsf{sk}, y)$
2 : **return** x

Claim 3.1. *$\mathcal{E}'$ is* qNME-qCPA *insecure.*

Proof Sketch. The scheme is malleable because given a ciphertext $y\|b$ of a plaintext x, it is trivial to create another ciphertext of x by just outputting $y\|\overline{b}$. □

Claim 3.2. *$\mathcal{E}'$ is* qIND-qCCA1 *secure.*

Proof Sketch. It is easy to see that any adversary $\mathcal{A}$ against $\mathcal{E}'$ can be used to construct an adversary $\mathcal{B}$ that attacks $\mathcal{E}$ as follows. $\mathcal{B}$ runs $\mathcal{A}$ using its own oracle $\mathcal{O}_1$, and uses $\mathcal{A}$'s challenge queries as its own challenge queries. Whenever $\mathcal{B}$ receives a challenge ciphertext, it samples a random bit b and appends it to the challenge ciphertext before forwarding it to $\mathcal{A}$. $\mathcal{B}$ outputs whatever $\mathcal{A}$ outputs. One can verify that $\mathsf{Adv}_{\mathcal{B},\mathcal{E}}(\lambda) = \mathsf{Adv}_{\mathcal{A},\mathcal{E}'}(\lambda)$. Thus, the security of $\mathcal{E}'$ follows from the security of $\mathcal{E}$. □

Theorem 3.8 (qNME-qCPA $\not\Rightarrow$ qIND-qCCA1). *If there exists an encryption scheme $\mathcal{E}$ that is* qNME-qCPA *secure, then there exists an encryption scheme $\mathcal{E}'$ that is* qNME-qCPA *secure but* qIND-qCCA1 *insecure.*

Proof. Assume there exists some qNME-qCPA secure encryption scheme $\mathcal{E} := \langle \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec} \rangle$. Fix a family $\mathsf{qPRF} := \{\mathsf{qPRF} : \{0,1\}^\ell \to \{0,1\}^\ell\}$ of quantum-secure pseudo-random functions. The new encryption scheme $\mathcal{E}' := \langle \mathsf{KeyGen}', \mathsf{Enc}', \mathsf{Dec}' \rangle$ is defined as follows.

$\mathsf{KeyGen}'(1^\lambda)$:
1 : $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{KeyGen}\left(1^\lambda\right)$
2 : $\mathsf{k} \xleftarrow{\$} \{0,1\}^\lambda$
3 : $\mathsf{sk}' \leftarrow \mathsf{sk}\|\mathsf{k}$
4 : **return** $(\mathsf{pk}, \mathsf{sk}')$

$\mathsf{Enc}'(\mathsf{pk}, x)$:
1 : $y \leftarrow \mathsf{Enc}(\mathsf{pk}, x)$
2 : **return** $0\|y$

$\mathsf{Dec}'(\mathsf{sk}\|\mathsf{k}, b\|y)$:
1 : **if** $b = 0$:
2 : **return** $\mathsf{Dec}(\mathsf{sk}, y)$
3 : **else if** $y = \mathsf{qPRF}(\mathsf{k}, 0)$:
4 : **return** sk
5 : **else return** $\mathsf{qPRF}(\mathsf{k}, y)$

Claim 3.3. *$\mathcal{E}'$ is* qIND-qCCA1 *insecure.*

Proof Sketch. The adversary queries $\mathsf{Dec}'(\mathsf{sk}\|\mathsf{k},\cdot)$ at $1\|0$ to get $v = \mathsf{qPRF}(\mathsf{k},0)$, and then queries it at the point $1\|v$ to get sk. At this point, the adversary can obviously break the security of $\mathcal{E}'$. □

Claim 3.4. *$\mathcal{E}'$ is* qNME-qCPA *secure.*

Proof. Fix $\mathcal{A}$ and λ. We prove security through a sequence of games.

Game G_0: This is the standard attack game.

Game G_1: Replace qPRF with a truly random function H.

Since qPRF is a quantum-secure pseudorandom function, $\mathcal{A}$ cannot distinguish G_1 from G_0, except with negligible probability.

Game G_2: This is identical to G_1. The only change is to the decryption algorithm, in which instead of returning sk when $y = H(0)$, it returns $H(H(0))$ which is a random value independent of the secret key sk.

Games G_1 and G_2 proceed identically unless $\mathcal{A}$ successfully outputs $H(H(0))$ with a single query. To bound the distinguishing probability, we invoke the following lemma.

Lemma 3.11 ([Unr14, Theorem 6.6]). *Let $\mathcal{A}$ be any quantum oracle algorithm making a single query to a random function H, with r inputs in the query. Then*

$$\Pr\left[x = H(H(0)) : H \xleftarrow{\$} (\{0,1\}^\ell \to \{0,1\}^\ell), x \leftarrow \mathcal{A}^H()\right] \leq 2^{-\Omega(\ell)}O(r).$$

This probability is negligible for polynomially-bounded r (number of inputs per query, which corresponds to the number of adversarial ciphertexts in a qNME security game).

Finally, we design an adversary $\mathcal{B} := (\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3)$ attacking $\mathcal{E}$ in the qNME-qCPA sense from the adversary $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ in this last game. $\mathcal{B}$ runs $\mathcal{A}$ as its subroutine and simulates a random oracle H itself. $\mathcal{B}_1$ and $\mathcal{B}_3$ output whatever $\mathcal{A}_1$ and $\mathcal{A}_3$ output, respectively. The algorithm $\mathcal{B}_2$ is defined as follows. $\mathcal{B}_2$ receives a vector (in superposition) of adversarial ciphertexts from $\mathcal{A}_2$.

- If the basis state is $|1\|y,z,\phi_{y,z}\rangle$, then it maps this basis state to $|\mathsf{Enc}(H(y)),z,1\|\phi'_{y,z}\rangle$ by allocating new ancilla registers (with proper padding), computing $\mathsf{Enc}(H(y))$ and then swapping these newly created registers with the y registers. The y registers are now included in the auxiliary registers $|\phi'_{y,z}\rangle$.
- Otherwise, it keeps the basis state the same, re-organizes the state to $|y,z\rangle\,|0\|\phi_{y,z}\rangle$.

$\mathcal{B}_2$ then outputs the resulting state as its adversarial ciphertexts.

Let D be the database of $\mathcal{B}$'s challenge queries. Consider $\mathcal{B}_2$'s adversarial ciphertexts state, let Dup be the event that this state has a non-negligible weight on ciphertexts $\mathsf{Enc}(H(y))$ such that $\mathsf{Enc}(H(y)) \in D$. The simulation is indistinguishable if this happens with negligible probability. To see that, imagine that in the real-world experiment, $\mathcal{A}_3$ would receive exactly $H(y)$. The only difference is in the random-world experiment: $\mathsf{Enc}(H(y)) \in D$ means that $H(y)$ is a random message obtained by apply a random function h. $\mathcal{A}_3$ would receive a pre-image of $h(H(y))$ (by the definition of the qNME

decryption oracle), which is different from $H(y)$ with overwhelming probability. This is only detectable if Dup happens with non-negligible probability.

Indeed, we show that $\Pr[\mathsf{Dup}]$ must be negligible, otherwise it would violate the security of $\mathcal{E}$ even in the qIND-qCPA. This is a standard argument, we omit the details. The security of $\mathcal{E}'$ now follows by the security of $\mathcal{E}$. □

Theorem 3.9 (qNME-qCCA1 $\not\Rightarrow$ qNME-qCCA2). *If there exists an encryption scheme $\mathcal{E}$ that is* qNME-qCCA1 *secure, then there exists an encryption scheme $\mathcal{E}'$ that is* qNME-qCCA1 *secure but* qNME-qCCA2 *insecure.*

Proof. Assume there exists some qNME-qCCA1 secure encryption scheme $\mathcal{E} := \langle \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec} \rangle$. Fix a family qPRF of quantum-secure pseudorandom functions. The new encryption scheme $\mathcal{E}' := \langle \mathsf{KeyGen}', \mathsf{Enc}', \mathsf{Dec}' \rangle$ is defined as follows.

$\mathsf{KeyGen}'(\lambda)$:

$(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{KeyGen}(\lambda)$
$\mathsf{k} \xleftarrow{\$} \{0,1\}^\lambda$
$\mathsf{sk}' \leftarrow \mathsf{sk} \| \mathsf{k}$
return $(\mathsf{pk}, \mathsf{sk}')$

$\mathsf{Enc}'(\mathsf{pk}, x)$:

$y \leftarrow \mathsf{Enc}(\mathsf{pk}, x)$
return $0 \| y \| 0$

$\mathsf{Dec}'(\mathsf{sk} \| \mathsf{k}, b \| y \| z)$:

if $b = 0 \wedge z = 0$:
  return $\mathsf{Dec}(\mathsf{sk}, y)$
else if $b = 1$:
  return $\mathsf{qPRF}(\mathsf{k}, y)$
else if $b = 2 \wedge z = \mathsf{qPRF}(\mathsf{k}, y)$:
  return $\mathsf{Dec}(\mathsf{sk}, y)$
else return $\perp$

Claim 3.5. *$\mathcal{E}'$ is* qNME-qCCA2 *insecure.*

Proof Sketch. Let $0 \| y \| 0$ be the classical challenge ciphertext. The adversary first queries $\mathsf{Dec}'(\mathsf{sk} \| \mathsf{k}, \cdot)$ at $1 \| y \| 0$ (which is not the challenge ciphertext) to get $v = \mathsf{qPRF}(\mathsf{k}, y)$, and then queries it at the point $2 \| y \| v$ to get the decryption of y, which is exactly the decryption of the challenge ciphertext. This helps the adversary to break the indistinguishability in the sense of qNME-qCCA2. □

Claim 3.6. *$\mathcal{E}'$ is* qNME-qCCA1 *secure.*

Proof. The proof is similar to that of Claim 3.4: first the pseudorandom function qPRF is replaced by a truly random function H, and for any decryption query of the form $2 \| y \| z$, we return $\perp$ where y is the challenge ciphertext.

The extra step is that we need to consider the case in which the adversary happens to query to the random function involving the challenge ciphertext. However, such event is unlikely since otherwise the scheme $\mathcal{E}$ would not be secure even in the sense of qIND-qCCA1. We formally prove the security through a sequence of games. Fix $\mathcal{A}$ and λ.

Game G_0: This is the standard attack game.

Game G_1: Replace qPRF with a truly random function H.

Since qPRF is a quantum-secure pseudorandom function, $\mathcal{A}$ cannot distinguish G_1 from G_0, except with negligible probability.

Game G_2: This is identical to G_1, except that now we will consider the encryption oracle and the decryption oracle where the random function H is involved as being implemented in the compressed oracle. Since these are equivalent to the standard oracles, these changes do not affect the adversary's success probability. We have $\Pr[G_2] = \Pr[G_1]$.

Game G_3: This is identical to G_2. Let D be the database of the challenge queries. The only change is to the decryption algorithm which is used in the last phase after the adversary has output its adversarial ciphertexts: if the ciphertext is $2\|y\|H(y)$ where $y \in D$ (in the form of $0\|y\|0$), then it returns $\perp$.

The intuition is that the adversary cannot make such a query (i.e., to put a non-negligible weight on inputs $2\|y\|H(y)$ where $y \in D$), except with negligible probability. Thus, the change is undetectable by the adversary. We formally bound the distinguishing probability between G_2 and G_3 by considering the two following events.

- Let ForgeOffline be the event that $\mathcal{A}_1$ (in the first phase) has a non-negligible query weight on inputs containing some $y \in D$ in its queries to H. Lemma 2.7 shows that the success probability of a quantum adversary in a standard oracle game is close to its success probability in the corresponding compressed oracle game. We now show that if ForgeOffline happens with non-negligible probability, we could design an adversary $\mathcal{B}$ that break $\mathcal{E}$ in the sense of qIND-qCCA1.
  - In the first stage, $\mathcal{B}$ implements a compressed random oracle and provides a simulation of the decryption oracle of $\mathcal{A}$ using its decryption oracle. Let D^H is the database kept by $\mathcal{B}$.
  - When $\mathcal{A}$ outputs its challenge, $\mathcal{B}$ measures its database D^H and gets many pairs $D^H = \{(y, H(y))\}$. $\mathcal{B}$ then submits these y values to its decryption oracle, which are legitimately counted as decryption in the first phase, and gets back their plaintexts x. Only at this point, $\mathcal{B}$ outputs $\mathcal{A}$'s challenge as its challenge. After receiving back the challenge ciphertexts, $\mathcal{B}$ measures its challenge query, and checks if there is any value in D^H. If it does then it outputs a bit b depending on whether their plaintexts are the same, otherwise it decides by flipping a coin. Observe that the success of $\mathcal{B}$ is exponentially close to one half the probability of ForgeOffline (by Lemma 2.7 and the standard argument).

  Thus, we have $\Pr[\text{ForgeOffline}]$ must be negligible.
- Let ForgeOnline be the event that the adversary correctly computes $H(H(y))$ for some $y \in D$ using only a single query to H in the last phase. By a similar argument to Lemma 3.11, we have that $\Pr[\text{ForgeOnline}]$ is negligible.

Therefore, we have that

$$|\Pr[G_3] - \Pr[G_2]| \leq \Pr[\text{ForgeOffline}] + \Pr[\text{ForgeOnline}],$$

which is negligible.

Finally, we construct an adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3)$ that attacks $\mathcal{E}$ in the sense of qNME-qCCA1 from any adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ of this last game. This can be argued analogously to the argument in Claim 3.4. We omit the details. □

3.4.3 A Lifting Theorem: From IND-qCCA2 to qIND-qCCA2

We present a compiler transforming IND-qATK security to qIND-qATK security. Our compiler follows the classical hybrid encryption paradigm. The message is encrypted under a random symmetric key each time, and the key is encrypted by the public-key encryption scheme. Since the same randomness is used for each query in superposition, we can use the same random symmetric key in superposition each time. This means that the adversary never has quantum access to the encryption algorithm of the public-key scheme, only the symmetric encryption needs to be secure against quantum queries, which we know how to construct from one-way functions (Theorem 3.3).

Construction 3.3 — qIND-qatk Secure Encryption Scheme

Let $\mathcal{E} := \langle \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec} \rangle$ be a public-key encryption scheme which is IND-qATK secure and δ-correct. Let $\mathcal{SE} := \langle \mathcal{K}, \mathsf{SymEnc}, \mathsf{SymDec} \rangle$ be a one-time qIND-qATK secure symmetric-key encryption scheme. We construct a new public-key encryption scheme $\mathcal{E}' := \langle \mathsf{KeyGen}', \mathsf{Enc}', \mathsf{Dec}' \rangle$ as follows.

$\mathsf{KeyGen}'(1^\lambda)$:

1 : $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{KeyGen}\left(1^\lambda\right)$
2 : **return** $(\mathsf{pk}, \mathsf{sk})$

$\mathsf{Enc}'(\mathsf{pk}, x)$:

1 : $\mathsf{k} \xleftarrow{\$} \mathcal{K}(1^\lambda)$
2 : $c_1 \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{k})$
3 : $c_2 \leftarrow \mathsf{SymEnc}(\mathsf{k}, x)$
4 : **return** $c_1 \| c_2$

$\mathsf{Dec}'(\mathsf{sk}, c_1 \| c_2)$:

1 : $\mathsf{k} \leftarrow \mathsf{Dec}(\mathsf{sk}, c_1)$
2 : $x \leftarrow \mathsf{SymDec}(\mathsf{k}, c_2)$
3 : **return** x

Remark 3.4. In this construction, we make no extra assumptions. We know that the existence of IND-qATK secure encryption implies the existence of quantum-secure one-way functions. IND-qATK secure public-key encryption can be constructed based on quantum-resistant assumptions (e.g., Learning With Errors) [BZ13b].

We give the security proof for adaptive chosen-ciphertext security below, the other cases can be treated similarly.

Theorem 3.10. *The encryption scheme $\mathcal{E}'$ defined in Construction 3.3 is* qIND-qCCA2 *secure, if $\mathcal{E}$ is* IND-qCCA2 *secure, and $\mathcal{SE}$ is one-time* qIND-qCCA2 *secure. In particular, for any* QPT *adversary $\mathcal{A}$, there exist* QPT *adversaries $\mathcal{B}, \mathcal{C}$ such that*

$$\mathsf{Adv}^{qind\text{-}qcca2}_{\mathcal{A},\mathcal{E}'}(\lambda) \leq \mathcal{O}(q_d \cdot \delta) + 2 \cdot \mathsf{Adv}^{ind\text{-}qcca2}_{\mathcal{B},\mathcal{E}}(\lambda) + \mathsf{Adv}^{qind\text{-}qcca2}_{\mathcal{C},\mathcal{SE}}(\lambda),$$

where q_d is the number of decryption queries in the second phase.

Proof. We prove this theorem using hybrid games. Since our definitions are closed under composition, it is sufficient to prove for the single-message security.

Let $\mathcal{A}$ be a QPT adversary. For any game G_{index}, we denote by

$$\Pr[G_{\mathsf{index}}] := |\Pr[G_{\mathsf{index}}(\mathcal{A}) = 1 \mid b = 1] - \Pr[G_{\mathsf{index}}(\mathcal{A}) = 1 \mid b = 0]|.$$

Also, by event $G_{\mathsf{index}}(\mathcal{A})$, we mean the output of the experiments (defined as in Definition 3.3) in game G_{index} when interacting with $\mathcal{A}$.

Game G_0: This is the standard attack game. Let k^* denote the symmetric key used during the encryption process within the oracle.

Game G_1: Notice that the same symmetric key k^* sampled during the encryption process within the challenger's oracle is used for all classical states of the superposition, the ciphertext state of the challenge query would be:

$$\sum_{x,y} \alpha_{x,y} |x, y\rangle \to |\mathsf{Enc}(\mathsf{pk}, \mathsf{k}^*)\rangle \sum_{x,y} \alpha_{x,y} |x, y \oplus \mathsf{SymEnc}(\mathsf{k}^*, x_b)\rangle ,$$

where x_b denotes the actual encrypted plaintext, depending on whether it is the real-world ($b = 1$) or the random-world ($b = 0$) (but the key k^* is independent of b). Notice that the first component c_1 of the ciphertext is a classical value.

This game is identical to G_0, except that now the real-or-random oracle $\mathcal{RR}(b)$ will store the first component c_1 of the challenge ciphertext in its local database D'. Since c_1 is classical, this action is undetectable. Thus, we have:

$$\Pr[G_0] = \Pr[G_1].$$

Game G_2: We define $\mathsf{FindImage}'$ that takes as input a tuple $((c_1, \cdot), D')$ and returns 1 if $c_1 \in D'$ and 0 otherwise. This is identical to G_1, except that in the real world we change the decryption oracle $\mathcal{O}_{\mathsf{Dec}'}$ in the second phase to

$$\mathcal{O}^2_{\mathsf{Dec}'} |y, z\rangle |D'\rangle = \begin{cases} |y, z \oplus \mathsf{Dec}'(\mathsf{sk}, y)\rangle |D'\rangle & \text{if } \mathsf{FindImage}'(y, D') = 0, \\ |y, z \oplus \mathsf{SymDec}(\mathsf{k}^*, c_2)\rangle |D'\rangle & \text{if } \mathsf{FindImage}'(y, D') = 1, \end{cases}$$

where $\mathsf{FindImage}'$ parses its input component y as $y = (c_1, c_2)$.

Essentially, in the second phase of this game, the decryption oracle in the real world does not apply algorithm Dec' to obtain the symmetric key, but instead just uses the key k^* produced by the challenge encryption oracle, if the query contains $c_1 \in D'$. (Notice that the database D' is classical.)

This change is slightly more than just conceptual, since KeyGen' may generate a bad key pair. Let DecFail be the event that $\mathsf{Dec}'(\mathsf{sk}, \mathsf{Enc}'(\mathsf{pk}, x)) = x' \neq x$. This event happens if and only if $\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{k}^*)) \neq \mathsf{k}^*$. Unless this event occurs, G_2 and G_1 proceed identically. Since $\mathcal{SE}$ is perfectly correct (by definition), any decryption failure of $\mathcal{E}'$ is a decryption failure of $\mathcal{E}$. Thus $\mathcal{E}'$ is also δ-correct. We thus have

$$|\Pr[G_2] - \Pr[G_1]| \leq \Pr[\mathsf{DecFail}] \leq \mathcal{O}(q_d \cdot \delta),$$

where the last inequality follows from the definition of correctness.

Game G_3: This is identical to G_2, except that in the real-or-random oracle, we encrypt a complete random value k^+ in place of the symmetric key k^*, that is we compute $c_1 = \mathsf{Enc}(\mathsf{pk}, \mathsf{k}^+)$, but we still use k^* for symmetric encryption and decryption.

It is straightforward to see that any adversary $\mathcal{A}$ that distinguishes games G_3 from G_2 can be turned to an adversary $\mathcal{B}$ attacking the underlying scheme $\mathcal{E}$, whose running time is essentially the same as that of $\mathcal{A}$. To be more precise, we can define two intermediate hybrids $G_{3.0}$ and $G_{3.1} = G_3$, in which this change is applied to the real world in $G_{3.0}$ and then to the random world in $G_{3.1}$. In each hybrid, the adversary $\mathcal{B}$ just runs the adversary $\mathcal{A}$, and uses $(\mathsf{k}^*, \mathsf{k}^+)$ as its challenge pair. Note that in two consecutive hybrids (G_2 and $G_{3.0}$, $G_{3.0}$ and $G_{3.1}$), the challenge ciphertext of $\mathcal{B}$ is c_1, which is classical,

as argued above, and that $\mathcal{B}$ never query to the decryption oracle on the challenge ciphertext, but instead uses k^* to answer the query (if it is simulating the game in the real world), and its compressed encryption oracle's database (if it is simulating the game in the random world). We have

$$|\Pr[G_3] - \Pr[G_2]| \leq 2 \cdot \mathsf{Adv}^{ind\text{-}qcca2}_{\mathcal{B},\mathcal{E}}(\lambda).$$

Furthermore, notice the fact that in this final game, k^+ is independent of the adversary's view and b, we now turn any distinguisher $\mathcal{A}$ of this game to an adversary $\mathcal{C}$ that breaks the one-time security of $\mathcal{SE}$. $\mathcal{C}$ runs $\mathcal{A}$, when it receives the challenge query from $\mathcal{A}$, it first generates a random string k^+ and encrypt it with the public key pk to get c_1, and sends $\mathcal{A}$'s challenge query directly to its challenger. After receiving the answer back, $\mathcal{C}$ appends $|c_1\rangle$ to the result and forwards it to $\mathcal{A}$. In the second phase, for any decryption query, $\mathcal{C}$ removes the first component from the ciphertext and forwards the query to its challenger.

By the security of $\mathcal{SE}$ we have $|\Pr[G_3] - \Pr[G_2]| \leq \mathsf{Adv}^{qind\text{-}qcca2}_{\mathcal{C},\mathcal{SE}}(\lambda)$.

Putting everything together, by the security of the underlying building blocks, we have the security of $\mathcal{E}'$. □

3.5 Bit Encryption Is Complete

In this section, we summarize our result for a fundamental question:

Is bit encryption in the quantum world complete as in the classical world?

We will show that the answer is affirmative, and give a construction for string encryption from bit encryption.

3.5.1 Bit-by-bit Encryption Is Insecure

We note that the question above is not trivial even for simpler cases of CPA and CCA1 security. In the classical setting, under CPA and CCA1 attacks, a secure bit encryption scheme can be applied bit-by-bit to construct a secure many-bit encryption scheme. However, the same construction fails in the quantum setting. This result for the symmetric-key setting was observed in [BBCL+21; CETU21]. Indeed, the same attack is also applicable to the public-key setting.

We will denote the encryption function as f, since the attack applies to both the symmetric-key and the public-key setting. Furthermore, we will show a stronger version of the result, which is bit-by-bit encryption of a qIND-qCCA2-secure scheme is insecure even in the sense of qIND-qCPA.

Theorem 3.11 ([BBCL+21; CETU21]). *Bit-by-bit encryption of a* qIND-qCCA2 *scheme is* qIND-qCPA *insecure. It holds in both symmetric-key and public-key settings.*

Proof. It is sufficient to show the attack for 2-bit encryption. The adversary $\mathcal{A}$ sends the following query as its challenge:

$$|\phi\rangle = |0\|+\rangle_X |0\|+\rangle_Y .$$

Informally, the adversary inserts 0 to the first input register (the first qubit), and $|+\rangle$ to the second input register (the second qubit), and sets the response registers to be $|0\|+\rangle$. The challenge ciphertext state will be:

$$|\phi_0\rangle = |0\|+\rangle_X |f(0)\|+\rangle_Y \text{ if } b = 0,$$

and

$$|\phi_1\rangle = \sum_{x,y\in\{0,1\}} |0\|x\rangle_X |\Big(0\|y\Big) \oplus \Big(f(h(0\|x)|_0)\|f(h(0\|x)|_1)\Big)\rangle \text{ if } b = 1,$$

where $h : \{0,1\}^2 \to \{0,1\}^2$ is a random function, and we write $h(z)|_0$ to denote its first output bit, and $h(z)|_1$ to denote its last output bit.

Measuring the second input register in the Hadamard basis in the case $b = 0$ returns 0 with probability 1, while in the case $b = 1$, this register will be entangled with the response registers with high probability due to the application of h. Thus measuring the second input register in the Hadamard basis in the case $b = 1$ returns 0 with small probability only. □

3.5.2 Completeness of Bit-Encryption

We answer the aforementioned question using our Theorem 3.3 and Theorem 3.10. In particular, we show the following theorem:

Theorem 3.12. *Many-bit* qIND-{qCPA, qCCA1, qCCA2}*-secure encryption schemes exist if and only if* 1*-bit* qIND-{qCPA, qCCA1, qCCA2}*-secure encryption scheme exists.*

Proof (Sketch). We give a sketch of the proof for the claim below.

For the symmetric-key encryption. From our Theorem 3.3, we conclude that if we can construct many-bit qCPA-secure encryption from 1-bit qCPA-secure encryption, then we are done. The steps to achieve this goal are as follows.

- Many-bit qCPA-secure encryption can be constructed from quantum-secure pseudorandom permutations (qPRPs) [CETU21, Theorem 44]. We note that the construction given in [CETU21, Theorem 44] is proven to be secure with respect to real-or-permuted security, but the proof also holds for real-or-random security which is used in our notions.
- qPRPs can be constructed from quantum-secure pseudorandom functions qPRFs [Zha16] (which uses function to permutation converters), or [HI19] (which uses the four-round Luby-Rackoff construction).
- qPRFs with one input bit implies qPRFs with many input bit (for example, via the GGM construction [Zha12a]).
- The existence of one input bit qPRFs is implied by our assumptions that 1-bit encryption scheme exists: the existence of encryption implies the existence of quantum-secure one-way functions, and quantum-secure one-way functions implies quantum-secure pseudorandom number generators [HILL99], which in turn gives us one input bit qPRFs [Zha12a].

For the public-key encryption. From our lifting theorem Theorem 3.10, we know that if we can construct many-bit encryption from 1-bit encryption for public-key schemes which only need to be secure against classical challenge queries, armed with the results in the symmetric-key setting, we are also done. For IND-qCPA, IND-qCCA1 security (with *classical* challenge queries), this follows directly from the bit-by-bit construction. For IND-qCCA2 security, it is not difficult to adapt the classical security proof of [HLW12] to the quantum setting. In particular, the construction and proof in [HLW12] involve defining *detectable*-CCA2 security and some "bad events" when the adversary submits a decryption query. Fortunately, all these notions are defined relatively to the adversary's challenge queries which are classical. Thus they can be defined similarly for the IND-qCCA2 security and the proof carries through. □

Chapter

4 Quantum Simulation-Sound Non-Interactive Zero-Knowledge

In this chapter, continuing the line of work on defining quantum security for classical cryptographic primitives, we describe quantum security notions for non-interactive zero-knowledge systems in which the adversary can make quantum queries to the zero-knowledge simulator. We then use these new notions to prove quantum chosen-ciphertext security (as defined previously in Chapter 3) of (a variant of) the classical Naor-Yung encryption scheme.

Chapter content

4.1 Quantum Zero-Knowledge

4.1.1 Definition

We give below the definition for quantum zero-knowledge in both the common reference string model and the quantum random oracle model. Our definition is a quantum counterpart of the classical definition for post-quantum zero-knowledge: the only difference is that now the adversary can query the simulator in superposition.

Definition 4.1 — Quantum Zero-Knowledge

Let $\mathcal{L}$ be a language in NP. A proof system $\mathsf{NIZK} := \langle \mathsf{Setup}, \mathcal{P}, \mathcal{V} \rangle$ for $\mathcal{L}$ is (adaptive) quantum zero-knowledge if there exists a QPT simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ such that for all QPT distinguisher $\mathcal{D}^* = \{\mathcal{D}^*_\lambda, \rho_\lambda\}_{\lambda \in \mathbb{N}}$, for every $n \in \mathbb{N}$, and for every $\lambda \in \mathbb{N}$:

$$\Big|\Pr\Big[\ \mathcal{D}^*_\lambda(\rho_\lambda, \mathsf{param})^{|\mathcal{P}^{\mathcal{O}}(\mathsf{param},\cdot,\cdot)\rangle, |\mathcal{O}\rangle} = 1 \ \Big|\ \mathsf{param} \leftarrow \mathsf{Setup}(1^n, 1^\lambda)\ \Big] - \Pr\Big[\ \mathcal{D}^*_\lambda(\rho_\lambda, \mathsf{param})^{|\mathcal{S}_2(\mathsf{td},\cdot)\rangle} = 1 \ \Big|\ (\mathsf{param}, \mathsf{td}) \leftarrow \mathcal{S}_1(1^\lambda)\ \Big]\Big| \leq \mathsf{negl}(\lambda),$$

where

- $\mathcal{D}^*$ can make quantum queries to the oracles.
- param is an empty string if we consider NIZK in the quantum random oracle model, or crs in the common reference string model.
- $\mathcal{O}$ is an empty oracle if we consider NIZK in the common reference string model, or a random oracle $\mathcal{O}_H$ in the quantum random oracle model.
- $\mathcal{P}(\mathsf{param}, \cdot, \cdot)$ is the prover algorithm and $\mathcal{S}_2(\mathsf{td}, \cdot)$ only acts on its private trapdoor td, the input statement $x \in \mathcal{L}_{yes} \cap \{0,1\}^n$ and its private random tape.

4.1.2 Construction

We note that any *perfect* adaptive post-quantum zero-knowledge proof system is also (perfect) quantum zero-knowledge proof system. Therefore, in this section, we consider the notion of *quantumly computational zero-knowledge*. We will briefly show that some known non-interactive zero-knowledge proof systems in literature, in particular the one given by Bitansky and Paneth in [BP15], satisfy this notion. At a high-level overview, the Bitansky-Paneth construction is a concrete instantiation of the Goldwasser-Ostrovsky transformation [GO93] which gives NIZKs from *invariant signatures*.

Informally, invariant signatures are digital signatures where all valid signatures of any message are either identical, or share a common property. Concretely, we say that a signature scheme is invariant if there is some efficiently computable property P of signatures such that for any message m^* and any verification key vk there is a unique value $P_{\mathsf{vk}}(m^*)$ such that $P(\sigma) = P_{\mathsf{vk}}(m^*)$ for any valid signature σ with respect to vk. Furthermore, it is required that for every message m^*, for an honestly generated verification key (sampled independently of m^*), the property value $P_{\mathsf{vk}}(m^*)$ is pseudo-random, even given the verification key and a signature oracle on messages $m \neq m^*$. We can also consider a relaxed notion of invariant signatures in the common random string model (CRS).

The Goldwasser-Ostrovsky transformation is based on the construction of Feige, Lapidot and Shamir [FLS90] of NIZKs in the *hidden-bits model*. In this model, a random hidden string is available to the prover but is hidden from the verifier. The prover can reveal to the verifier specific bits of the hidden string in the locations of its choice, but it cannot change the value of these bits. Very briefly, the transformation is as follows: we interpret the CRS (available to both prover and verifier) as containing a CRS

for the invariant signature, as well as a sequence of messages $\{m_i\}$ and one-time pad bits $\{s_i\}$ where every (m_i, s_i) will be used to obtain a single hidden bit b_i. The prover will sample keys $(\mathsf{vk}, \mathsf{sk})$ for the invariant signature and send the verification key vk to the verifier as part of the proof. The hidden bit b_i is then defined as the bit $P_{\mathsf{vk}}(m_i)$, the property value of the message m_i, XORed with the one-time pad bit s_i. To reveal the bit b_i, the prover sends to the verifier a signature σ_i on m_i. The verifier can compute b_i by computing $P(\sigma) = P_{\mathsf{vk}}(m_i)$.

The simulator can be defined based on this strategy, where first we run the simulator of the proof system in the hidden model to obtain a proof π and a set of revealing bits $\{b_i\}$. The CRS will be generated exactly as in the real execution, except that the one-time pad bits $\{s_i\}$ are computed as $P_{\mathsf{vk}}(m_i) \oplus b_i$.

The proof of zero-knowledge is essentially based on pseudo-randomness property of the signature, so that each hidden bit in the simulation is computationally indistinguishable from a uniformly random bit as in the real execution.

There are two important points that make the proof also works in the quantum setting:

- A NIZK proof system in the hidden-bit model can achieve both perfect soundness and perfect zero-knowledge [HU19].
- The computational indistinguishability only appears in the proof of the CRS generation in the real execution and the simulation, which is classical and independent of the adversary's queries.

Since perfect zero-knowledge implies quantum zero-knowledge, the classical proof also carries to the quantum setting, when the building blocks are post-quantumly secure. For more details, we refer the reader to the (classical) proof given in [BP15].

4.2 Quantum Simulation-Soundness

Informally, a zero-knowledge proof system is said to be *simulation-sound* if it has the property that an adversary cannot provide a convincing proof for any false statement, even if it has seen *simulated proofs* of arbitrary statements (including false statements). In the classical setting, simulation-soundness is defined with respect to a zero-knowledge simulator. The simulator would keep a list of statements and simulated proofs, and the adversary is asked to output a new pair of statement and proof that is outside of the list. The classical definition for simulation-soundness thus inherently implies recording and comparison of queries. Translating this definition into the quantum setting can be tricky, mainly because of quantum measurement and the no-cloning theorem, which prevent the simulator to keep such a list when the adversary can make queries quantumly. The same barrier also appears in different context of defining quantum indistinguishability for encryption (see Chapter 3), or defining quantum unforgeability for digital signatures [BZ13b; AMRS20].

To define quantum-simulation-soundness, we take advantage of the fact that the randomness to compute a proof is chosen by the simulator and can be classical, and we assume that the simulator picks a fresh randomness for each query. For each query, we thus ask the simulator to record the randomness used to respond the query in a list R. At the end, the adversary returns two pairs (x_1, π_1), (x_2, π_2) and wins if either of the two following statements is true:

1. None of the randomnesses in R matches with one of the pairs (x_1, π_1), (x_2, π_2) with x_1 or x_2 being a false statement.

2. There exists a randomness $r \in R$ that matches with these two pairs and one of the statements x_1 or x_2 is a false statement.

We note that restricting to classical queries, this definition is stronger than the classical simulation-sound definition. An adversary that breaks the classical definition outputs a new pair (x, π) where x is a false statement. If x has not been queried before, the adversary can break one of the two cases above. If x has been queried before but π is a new proof, the randomness to generate π should not be in R and the adversary breaks case 1.

The formal definition follows.

Definition 4.2 — Quantum-Simulation-Soundness (QSS-NIZK)

Let $\mathcal{L}$ be a language in NP. Consider a proof system $\langle \mathsf{Setup}, \mathcal{P}, \mathcal{V} \rangle$ for $\mathcal{L}$ with zero-knowledge simulator $\mathcal{S} := (\mathcal{S}_1, \mathcal{S}_2)$. In each query, $\mathcal{S}_2$ stores the randomness used to answer the query in a list R. A QPT adversary $\mathcal{A}$ after making polynomial numbers of quantum queries to $\mathcal{S}_2$ outputs two pairs $\{(x_i, \pi_i)\}_{i=1}^2$. The adversary $\mathcal{A}$ wins if either of the following two cases hold:

1. There exists $i \in [\![1, 2]\!]$ such that $x_i \notin \mathcal{L}$, for all $r \in R$, $\mathcal{S}_2(x_i, r) \neq \pi_i$ and $\mathcal{V}(\mathsf{crs}, x_i, \pi_i) = 1$.

2. There exists a randomness $r \in R$ such that $\mathcal{S}_2(x_1, r) = \pi_1$ and $\mathcal{S}_2(x_2, r) = \pi_2$ and at least one of x_1 or x_2 is not in $\mathcal{L}$.

Formally, we say an NIZK proof is *quantum-simulation-sound* if for all $\lambda \in \mathbb{N}$, for all QPT adversaries $\mathcal{A}$, $i, j \in [\![1, 2]\!]$ we have:

$$\Pr\left[\begin{array}{c} \mathcal{V}(\mathsf{crs}, x_i, \pi_i) = 1 \,\forall\, i \,\wedge \\ \Big(\exists\, i \,:\, x_i \notin \mathcal{L}\Big) \,\wedge \\ \Big((\mathcal{S}_2(\mathsf{td}, x_i, r) \neq \pi_i \,\forall\, r \in R) \vee \\ (\exists\, r \in R : \mathcal{S}_2(\mathsf{td}, x_j, r) = \pi_j \,\forall\, j)\Big) \end{array} \;\middle|\; \begin{array}{r} (\mathsf{crs}, \mathsf{td}) \leftarrow \mathcal{S}_1(1^\lambda) \\ \{(x_i, \pi_i)\}_{i=1}^2 \leftarrow \mathcal{A}^{|\mathcal{S}_2(\mathsf{td}, \cdot)\rangle}(\mathsf{crs}) \end{array}\right] \leq \mathsf{negl}(\lambda).$$

Some Technical Discussions

We now remark on a few details on the notion of quantum-simulation-soundness. First, it might seem that our definition does not capture all possible *quantum* attacks. Consider the following adversary $\mathcal{A}$. $\mathcal{A}$ makes a quantum query to the simulator and obtains a superposition of statements and proofs as $\sum_{x \neq x_0, y} \alpha_{x,y} |x, y \oplus \pi\rangle$, where $x_0 \notin \mathcal{L}$. We assume that the simulator answered the query with a classical randomness r, that is hidden from $\mathcal{A}$. (Note that, if r is not hidden, there is a trivial attack.) $\mathcal{A}$ then performs some quantum computation to come up with a proof for x_0, with *the same* randomness r, and during the process, $\mathcal{A}$ also destroys the original state, thus $\mathcal{A}$ cannot procedure two pairs of $\{(x_i, \pi_i)\}_{i=1}^2$ that are computed using the same randomness. Essentially, the adversary makes one or more quantum queries but then must consume

the post-query states completely in order to make a single, but convincing, forgery. Obviously, if such an adversary $\mathcal{A}$ exists, this might consider a quantum attack against quantum-simulation-soundness, but Definition 4.2 does not capture this. However, this so-called attack is inherited from the nature of quantum queries and can be applied in similar scenarios, for instance, the $(n+1)$-definition proposed in [BZ13a; BZ13b] or the blind-unforgeability proposed in [AMRS20] for classical digital signatures. The second condition in Definition 4.2 is thus used to capture classical attacks rather than quantum attacks. We leave this as an open problem, either to find a concrete example for this type of attack, or to show that (in most of the cases) this is not possible.

Secondly, our definition also captures some "malleability" attack that is not captured by the classical definition. In particular, imagine that if the adversary makes a query to the simulator for a statement $x_1 \notin \mathcal{L}$, and outputs a proof for a statement $x_2 \in \mathcal{L}$ with the same randomness used by the simulator. This attack does not violate the classical simulation-soundness, but it is captured by our definition. This is because it is not possible in general to distinguish which statement was queried by the adversary in the quantum setting. We note that the "inverse" case (that is, $x_1 \in \mathcal{L}$ and $x_2 \notin \mathcal{L}$) is obviously an attack and it is captured in both classical and quantum notions.

4.3 Separation Between Post-Quantum and Quantum Security

In this section, we introduce a new notion of *quantum-query advantage functions*, which are functions that can be used to demonstrate advantages of quantum queries over classical queries. Our definition and construction of quantum-query advantage functions follows those of *quantum advantage functions* given in [LMQW22]. The main difference between the two objects is that:

- Quantum advantage functions in [LMQW22] demonstrate a quantum advantage with only *classical queries*, showing separations between *classical* security and *post-quantum* security.
- Our quantum-query advantage functions demonstrate a quantum advantage with *quantum* queries, showing separations between *post-quantum* security and *quantum* security.

We give the definition and construction for quantum-query advantage functions in Section 4.3.2 and use it to show a separation between quantum simulation-sound NIZKs and classical simulation-sound NIZKs in Section 4.3.3.

Remark 4.1. Our quantum-query advantage functions can be constructed trivially from periodic pseudorandom functions (which was used to show separation between quantum-query security and classical-query security for digital signatures and chosen-ciphertext security in [BZ13b]). Informally, for appropriate parameters, the period finding algorithm of Boneh and Lipton [BL95] allows to recover the secret period of the pseudorandom function with a single quantum query, but the function is computationally indistinguishable from a random function if the adversary only has classical access. In this thesis, we present another construction based on recent interactive proofs of quantumness [BCMV+18]. The advantage of our approach is that it allows us to define more properties for the quantum-query advantage function, which we believe to be useful to show separations in other settings.

4.3.1 Preliminaries: Interactive Proof of Quantumness

We first recall the definition of interactive proof of quantumness protocols with 4 messages in total, which corresponds to the best round complexity known for interactive proofs of quantumness in the plain model [BCMV+18].

Definition 4.3 — Interactive Proof of Quantumness

An interactive proof of quantumness is an interactive protocol Π_{ipq} between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ using classical communication, with the following properties:

Quantum completeness: there exists a QPT quantum prover $\mathcal{P}$ such that for all $\lambda \in \mathbb{N}$:

$$\Pr\Big[(\mathcal{P},\mathcal{V})(1^\lambda) = 1\Big] \geq 1 - \mathsf{negl}(\lambda).$$

Classical soundness: for any PPT classical prover $\mathcal{P}^*$, for all $\lambda \in \mathbb{N}$:

$$\Pr[(\mathcal{P}^*,\mathcal{V}) = 1] \leq \mathsf{negl}(\lambda).$$

In a 4-round interactive proof of quantumness protocol, the first message is sent by the verifier to the prover. Let v_1, v_2 (resp. p_1, p_2) denote the messages sent by the verifier (resp. the prover) during the execution of an interactive proof of quantumness Π_{ipq}. An interactive proof of quantumness Π_{ipq} can furthermore satisfy the following optional property:

Public-coin second verifier message: the second verifier message v_2 consists of uniformly and independently sampled random coins.

Semi-quantum soundness: for any QPT *quantum* prover $\mathcal{P}^*$, for all $\lambda \in \mathbb{N}$:

$$\Pr\Big[(\mathcal{P}^*,\mathcal{V}_{\mathsf{semi}})(1^\lambda)\Big] \leq \mathsf{negl}(\lambda),$$

where the verifier $\mathcal{V}_{\mathsf{semi}}$ is defined as follows.

- Let $\mathcal{P}$ denote the efficient quantum prover for Π_{ipq} such that
$$\Pr\Big[(\mathcal{P},\mathcal{V})(1^\lambda)\Big] \geq 1 - \mathsf{negl}(\lambda).$$
- $\mathcal{V}_{\mathsf{semi}}$ runs $\mathcal{V}$ to obtain the first verifier message v_1.
- Whenever $\mathcal{V}_{\mathsf{semi}}$ receives a classical message x from $\mathcal{P}^*$, it runs $\mathcal{P}$ on (v_1, x) and obtains a classical message p_1.
- $\mathcal{P}^*$ is allowed to send a *classical* message x to $\mathcal{V}_{\mathsf{semi}}$ and receive back a tuple of classical message (p_1, v_2) where v_2 is the second verifier message. Then it outputs a classical message p_2.
- $\mathcal{V}_{\mathsf{semi}}$ outputs the output of $\mathcal{V}$ on (v_1, p_1, v_2, p_2).

Intuitively, semi-quantum soundness guarantees that no efficient quantum prover can cheat when the first prover message is generated by a *classical* prover.

The Quantum Certification Protocol from [BCMV+18]

This protocol relies on a post-quantum secure trapdoor claw-free (TCF) family of functions with adaptive hard-core bit property $f_0, f_1 : \{0,1\}^n \to \{0,1\}^m$. A TCF pair is a pair of functions which are injective, with the same image, and satisfy the following property. With knowledge of a secret trapdoor it is possible to efficiently (classically) compute the two pre-images x_0 and x_1 of a given y ($f_0(x_0) = f_1(x_1) = y$), but without the trapdoor, there is no efficient quantum algorithm that can compute such a triple (x_0, x_1, y), referred to as a claw, for any y. The adaptive hardcore bit property states that it is also hard to hold both a single pre-image x_b, as well as a string $d \in \{0,1\}^b \setminus \{0^b\}$ and a bit c such that $c = d \cdot (x_0 \oplus x_1)$.

We note that while the quantum device cannot compute a claw or break the adaptive hard-core bit property, nevertheless it can simultaneously hold an image y as well as a superposition $\frac{1}{\sqrt{2}}(|0, x_0\rangle + |1, x_1\rangle)$ over the two pre-images of y, simply by evaluating f on a uniform superposition over all inputs and measuring the image register y. Then by either measuring the state in the computational basis or the Hadamard basis, the device can obtain either a random pre-image x_b of y, or a pair (c, d) such that $c = d \cdot (x_0 \oplus x_1)$.

A high-level description of the [BCMV+18] protocol is given below.

Protocol 4.1: 4-round Interactive Proof of Quantumness by [BCMV+18]

1. The verifier generates a TCF pair, along with a trapdoor, and sends just the function pair to the prover.
2. The prover returns an image y of the TCF pair.
3. The verifier challenges the prover by randomly asking for either a pre-image of y, or a bit c and an n-bit string d such that $d \cdot (x_0 \oplus x_1) = c$.
4. The prover measures in the computational or Hadamard basis to return the requested output and the verifier checks the validity by using the trapdoor to compute the two pre-images x_0, x_1 of y.

Based on this protocol, we obtain the following lemma.

Lemma 4.1. *Under the LWE assumption, there exists a 4-message interactive proof of quantumness protocol satisfying: (1) public-coin second verifier message and (2) semi-quantum soundness.*

Proof. The protocol we use in the proof is the n-fold parallel repetition of Protocol 4.1. Protocol 4.1 has soundness error $1/2$, and parallel repetition amplifies the soundness of this protocol, which has been shown in [RS20].

By inspecting Protocol 4.1, we note that the verifier's second message is public coin. What remains is to argue that the protocol is also semi-quantum sound: in Protocol 4.1, the crucial point is that the prover can compute a quantum state to obtain its first message p_1 on its own, and later this quantum state will be either measured in the computational basis or the Hadamard basis to answer the challenge from the verifier. Now consider the security game of semi-quantum soundness: the prover can only compute p_1 via sending a *classical* query to $\mathcal{V}_{\mathsf{semi}}$. Furthermore, this computation of y_1 is done by $\mathcal{V}_{\mathsf{semi}}$, which acts exactly as a honest prover in Protocol 4.1 (except that the prover is now classical). Since this is a classical query, no efficient quantum prover can give a

valid answer in the Hadamard basis for a fixed pair (v_1, p_1). Formally, we can construct a simulator $\mathcal{S}$ which simulates the malicious semi-quantum prover $\mathcal{P}^*$ and plays the role of the prover in Protocol 4.1. $\mathcal{S}$ makes a copy of (v_1, x, p_1) (where x is the input of $\mathcal{P}^*$'s query) and later sends (v_1, p_1, v_2) to $\mathcal{P}^*$. Finally $\mathcal{S}$ outputs whatever $\mathcal{P}^*$ outputs. One can see that now if $\mathcal{P}^*$ breaks the semi-quantum soundness, $\mathcal{S}$ breaks the "adaptive hard-core bit" property of Protocol 4.1. □

4.3.2 Quantum Advantage with Quantum Query Algorithms

Definition 4.4 — Quantum-Query Advantage Functions

A *quantum-query* advantage function is a pair of PPT algorithms $\langle \mathsf{Setup}, \mathsf{QAF} \rangle$ with the following properties:

$(\mathsf{pp}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$. On input a security parameter λ, the setup algorithm Setup outputs a public parameter pp and a secret key sk. Without loss of generality, we will consider that the secret key sk includes the public parameter pp.

$\mathsf{QAF}(\mathsf{sk}, x)$. On input a secret key sk and a message x, the (randomized) evaluation algorithm QAF outputs either a message y, or a special "accept" symbol denoted accept, or a special "reject" symbol denoted reject. For our applications later, we require that by default $\mathsf{QAF}(\mathsf{sk}, \cdot)$ is stateless.

We additionally require the following properties:

q-Quantum-query easiness. For any $\lambda \in \mathbb{N}$, there exists a QPT oracle algorithm $\mathcal{A}^{|\mathsf{QAF}(\mathsf{sk},\cdot)\rangle}(\mathsf{pp})$ such that:

$$\Pr\left[\mathsf{QAF}(\mathsf{sk}, x) = \mathsf{accept} \mid x \leftarrow \mathcal{A}^{|\mathsf{QAF}(\mathsf{sk},\cdot)\rangle}(\mathsf{pp})\right] = 1 - \mathsf{negl}(\lambda),$$

where $\mathcal{A}^{|\mathsf{QAF}(\mathsf{sk},\cdot)\rangle}$ makes q *quantum* queries in total to $\mathsf{QAF}(\mathsf{sk}, \cdot)$ before outputting x, and the probability is taken over $(\mathsf{pp}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$.

Classical-query hardness. For any $\lambda \in \mathbb{N}$, for all QPT oracle algorithm $\mathcal{A}^{\mathsf{QAF}(\mathsf{sk},\cdot)}(\mathsf{pp})$ such that:

$$\Pr\left[\mathsf{QAF}(\mathsf{sk}, x) = \mathsf{accept} \mid x \leftarrow \mathcal{A}^{\mathsf{QAF}(\mathsf{sk},\cdot)}(\mathsf{pp})\right] \leq \mathsf{negl}(\lambda),$$

over $(\mathsf{pp}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$.

Construction. Let Π_{ipq} be a 4-message interactive proof of quantumness, in the form of Definition 4.3 in Section 4.3.1 with public-coin second verifier message and semi-quantum soundness properties. We define our quantum-query advantage function as follows.

Construction 4.1 — A quantum-query advantage function from interactive proof of quantumness

$\mathsf{Setup}(1^\lambda)$:

- Run the first verifier message for Π_{ipq} to obtain (v_1, r), where v_1 is the first verifier message and r is the private coin of the verifier of Π_{ipq}.
- Sample a uniformly random string $v_2 \xleftarrow{\$} \{0,1\}^*$ as the second verifier message for Π_{ipq}.
- Set pp as an empty string and $\mathsf{sk} := (\mathsf{pp}, \mathsf{k}, v_1, v_2, r)$ and output $(\mathsf{pp}, \mathsf{sk})$.

$\mathsf{QAF}(\mathsf{sk}, \cdot)$: on input a message x, we consider several distinguished cases (all cases are considered with appropriate input length):

- If x is of the form $(0\|u)$: compute the semi-quantum verifier message for Π_{ipq} on (v_1, u) and obtain (v_1, p_1, v_2). Output (p_1, v_2).
- If x is of the form $(1\|p_1\|p_2)$: if the verifier of Π_{ipq} accepts the transcript (v_1, p_1, v_2, p_2) with the secret state r, output accept, otherwise output reject.
- Otherwise output $\perp$.

Theorem 4.1. *Let Π_{ipq} be a 4-message interactive proof of quantumness satisfying the properties specified in Lemma 4.1: public-coin second verifier message and semi-quantum soundness. Then there exists a quantum-query advantage function satisfying 2-quantum-query easiness (Definition 4.4).*

Combined with Lemma 4.1, we obtain the following:

Corollary 4.1. *Assuming the (classical) hardness of LWE, there exists a quantum advantage function satisfying 2-quantum easiness (Definition 4.4).*

The proof of Theorem 4.1 follows from Lemma 4.2 and Lemma 4.3 stated below.

Lemma 4.2 (Quantum-query easiness). *Suppose Π_{ipq} satisfies quantum completeness (Definition 4.3). Then Construction 4.1 satisfies quantum-query easiness.*

Proof. Let $\mathcal{P}$ denote the efficient quantum prover for Π_{ipq} such that

$$\Pr\left[(\mathcal{P}, \mathcal{V})(1^\lambda) = 1\right] \geq 1 - \mathsf{negl}(\lambda).$$

Define the following QPT algorithm $\overline{\mathcal{P}}$:

- Make a (quantum) query $\sum_x |0\|x, 0\rangle$ to $\mathsf{QAF}(\mathsf{sk}, \cdot)$.
- Measure the response register to get a classical string p_1.
- Run $\mathcal{P}$ on p_1, v_2 and the post-measurement state to obtain p_2.
- Output $p_1\|p_2$.

By the completeness of Π_{ipq}, $\mathsf{QAF}(\mathsf{sk}, \cdot)$ outputs accept with probability $1 - \mathsf{negl}(\lambda)$. □

Lemma 4.3 (Classical-query hardness). *Suppose Π_{ipq} has public-coin second verifier messages and has semi-quantum soundness (Lemma 4.1). Then Construction 4.1 satisfies classical hardness.*

Proof. Let $\mathcal{A}(\mathsf{pp})$ denote a QPT adversary with *classical* oracle access to $\mathsf{QAF}(\mathsf{sk},\cdot)$. Without loss of generality, we assume that $\mathcal{A}$ queries its output x^* to $\mathsf{QAF}(\mathsf{sk},\cdot)$ before halting, and that A outputs the first x^* it queries such that $\mathsf{QAF}(\mathsf{sk},x^*) = \mathsf{accept}$, if such a query exists. Let Q denote the number of oracle queries $\mathcal{A}$ makes. We define the following hybrid experiments, where we change the input-output behavior of $\mathsf{QAF}(\mathsf{sk},\cdot)$:

Game G_0: This is the classical-query hardness experiment (Definition 4.4) where $\mathcal{A}$ has classical oracle access to $\mathcal{O}_0(\mathsf{sk},\cdot) \coloneqq \mathsf{QAF}(\mathsf{sk},\cdot)$, where $(\mathsf{pp},\mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$. We say that the adversary wins the experiment if it outputs x^* such that $\mathcal{O}_0(\mathsf{sk},x^*) = \mathsf{accept}$.

Game G_1: We do not change the behavior of the oracle (i.e., $\mathcal{O}_1 \coloneqq \mathcal{O}_0$), but we change the winning condition of the experiment. We now guess a uniformly random index $j \in [\![Q]\!]$. We now say that $\mathcal{A}$ wins if and only if the following conditions hold:

(a) the j-th oracle query from $\mathcal{A}$, on input x_j is of the form $1\|p_1\|p_2$;

(b) $\mathcal{O}_1(\mathsf{sk},x_j) = \mathsf{accept}$, and for all prior oracle queries x, $\mathcal{O}_1(\mathsf{sk},x) \neq \mathsf{accept}$.

Game G_2: We change how oracle queries are handled and define $\mathcal{O}_2(\mathsf{sk},\cdot)$ as follows. On any query $i \neq j$ of the form $x_i = 1\|p_1\|p_2$, $\mathcal{O}_2$ rejects.

Claim 4.1. *For all classical-query* QPT *adversaries $\mathcal{A}$:*

$$\Pr\left[\mathcal{A}^{\mathcal{O}_1(\mathsf{sk},\cdot)}(\mathsf{pp}) = 1\right] = \frac{1}{Q}\Pr\left[\mathcal{A}^{\mathcal{O}_0(\mathsf{sk},\cdot)}(\mathsf{pp}) = 1\right].$$

Proof. For any execution of the experiment in G_0 such that $\mathcal{A}(\mathsf{pp})$ outputs $x^* = 1\|p_1^*\|p_2^*$ such that $\mathcal{O}_0(\mathsf{sk},x^*) = \mathsf{accept}$, recall that we assume without loss of generality that $\mathcal{A}$ queries the oracle on x^*, and that x^* corresponds to the first query from $\mathcal{A}$ such that $\mathcal{O}_0(\mathsf{sk},x^*) = \mathsf{accept}$. Let j^* denote the index corresponding to the first query $\mathcal{A}$ makes on x^*. Since j is chosen uniformly at random from $[\![Q]\!]$, the probability of $\mathcal{A}$ winning in G_1 is the probability that $j = j^*$, which is $\frac{1}{Q}$. The claim follows. □

Claim 4.2. *For all classical-query* QPT *adversaries $\mathcal{A}$:*

$$\Pr\left[\mathcal{A}^{\mathcal{O}_2(\mathsf{sk},\cdot)}(\mathsf{pp}) = 1\right] = \Pr\left[\mathcal{A}^{\mathcal{O}_1(\mathsf{sk},\cdot)}(\mathsf{pp}) = 1\right].$$

Proof. By definition of the winning condition in G_1, $\mathcal{A}$ loses in the experiment if its i-th oracle query on input x_i satisfies $i < j$ and $\mathcal{O}_1(\mathsf{sk},x_i) = \mathsf{accept}$. Furthermore, the queries made by $\mathcal{A}$ after querying its first accepting input x^*, if such an x^* exists, do not affect its output (as we assume $\mathcal{A}$ would then output x^*). Therefore the winning probability of $\mathcal{A}$ in both G_1 and G_2 are exactly identical. □

Claim 4.3. *For all classical-query* QPT *adversaries $\mathcal{A}$:*

$$\Pr\left[\mathcal{A}^{\mathcal{O}_2(\mathsf{sk},\cdot)}(\mathsf{pp}) \text{ wins in } G_2\right] \leq \mathsf{negl}(\lambda).$$

Proof. Let $\mathcal{A}$ be a classical-query QPT algorithm such that $\mathcal{A}$ wins with probability ε in G_2. We build a prover $\mathcal{P}^*$ that breaks semi-quantum soundness of Π_{ipq} with probability ε as follows:

1. Make a guess $j \xleftarrow{\$} [\![Q]\!]$. Run $\mathcal{A}$.
2. To answer the i-th query, if x_i is of the form $0\|u$, send u to the verifier to obtain $(p_1^{(i)}, v_2^{(i)})$. Store the pair $(p_1^{(i)}, v_2^{(i)})$ in a list R. Otherwise parse $x_i = 1\|p_1\|p_2$ and respond to the query according to the following cases:
   - If $i = j$, check that $p_1 = p_1^j$, set $p_2^j := p_2$ and send p_2^j as the second prover message in Π_{ipq}.
   - Otherwise, response to the query from $\mathcal{A}$ with reject.

Our reduction perfectly simulates the view of $\mathcal{A}$ in G_2, and if $\mathcal{A}$ wins G_2, we have that the pair (p_1^j, p_2^j) is an accepting input for Π_{ipq} in the semi-quantum soundness experiment. □

Overall these claims show that the probability of $\mathcal{A}$ winning in G_0 is negligible, and finishes the proof of Lemma 4.3. □

4.3.3 Separation for QSS-NIZK

In this section, we use our quantum-query advantage functions to give an example of a NIZK proof system that is classically simulation-sound but not quantumly simulation-sound. Our separation is constructed by carefully embedding instances of interactive quantum-query advantage into the simulator of the NIZK system. The key conceptual insight is that although we are considering non-interactive proof systems, the security game for simulation-soundness is interactive, allowing us to use a quantum adversary that makes the quantum-query advantage function accept to also break the simulation-soundness: an efficient quantum adversary given *classical* oracle access to QAF cannot cause it to ever output accept, while it can do so by only making 2 *quantum* queries.

Construction 4.2 — Separation for QSS-NIZK

Let $\mathcal{L}'$ be a language in NP, with the associated relation R'. Let $\mathcal{L}$ denote the NP language defined in Equation (4.1). Let $\Pi = \langle \mathsf{Setup}, \mathcal{P}, \mathcal{V} \rangle$ be a post-quantum simulation-sound non-interactive zero-knowledge proof system for $\mathcal{L}$, and $\langle \mathsf{Setup}, \mathsf{QAF} \rangle$ be a quantum-query advantage function. We define the following NIZK proof system $\overline{\Pi} = \langle \overline{\mathsf{Setup}}, \overline{\mathcal{P}}, \overline{\mathcal{V}} \rangle$ for $\mathcal{L}$ as follows.

$\overline{\mathsf{Setup}(1^\lambda)}$: Output $\mathsf{pp} \leftarrow \Pi.\mathsf{Setup}(1^\lambda)$. (We note that $\mathsf{pp} = \mathsf{crs}$ in the CRS model, and pp is a token that allow the parties to make quantum queries to the random oracle in the QROM.)

$\overline{\mathcal{P}(\mathsf{pp}, x, w)}$: Compute $(\mathsf{pp}', \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$. Compute $y \leftarrow \mathsf{QAF}(\mathsf{sk}, x)$. Generate a proof π using Π for the statement $(x, y) \in \mathcal{L}$. Output $(y\|\pi)$.

$\overline{\mathcal{V}(\mathsf{pp}, x, \overline{\pi})}$: Parse $(y\|\pi) \leftarrow \overline{\pi}$. Output $\mathcal{V}(\mathsf{pp}, (x, y), \pi)$.

We define the following augmented language $\mathcal{L}$ of R':

$$\mathcal{L} := \{(x, y) : \exists\, (\mathsf{sk}, r, w) : R'(x, w) = 1 \wedge (y = \mathsf{reject} \vee y = \mathsf{QAF}(\mathsf{sk}, x; r))\}. \tag{4.1}$$

It is easy to see that completeness and soundness of $\overline{\Pi}$ follow directly from those of Π.

We now construct a simulator $\overline{\mathcal{S}}$ for zero-knowledge property of $\overline{\Pi}$, which is later on also used in the proofs of simulation-soundness. Let $\mathcal{S} := (\mathcal{S}_1, \mathcal{S}_2)$ be a zero-knowledge simulator of Π. The simulator $\overline{\mathcal{S}} := (\overline{\mathcal{S}_1}, \overline{\mathcal{S}_2})$ works as follows.

- $\overline{\mathcal{S}_1}$: Output $\mathcal{S}_1$.
- $\overline{\mathcal{S}_2}$: Initialize an empty list Q. On the input a statement x,
  - For each pair $(\mathsf{pp}'_i, \mathsf{sk}_i) \in Q$, compute $y_i \leftarrow \mathsf{QAF}(\mathsf{sk}_i, x)$.
    - If $y_i = \mathsf{reject}\ \forall\ i$, set $y = \mathsf{reject}$.
    - If there exists an index i such that $y_i = \mathsf{accept}$, set $y = \mathsf{accept}$.
    - Otherwise, compute $(\mathsf{pp}', \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$, store $(\mathsf{pp}', \mathsf{sk})$ in Q and compute $y \leftarrow \mathsf{QAF}(\mathsf{sk}, x)$.
  - Run $\mathcal{S}_2$ on input (x, y) to obtain a simulated proof π.
  - If $y = \mathsf{accept}$, generate a simulated proof π' for a random *false* statement $x' \in \mathcal{L}$ (by sampling $x' \in \{0,1\}^n$ uniformly at random, where n is the length of a statement in $\mathcal{L}$) and output $(\pi, (x', \pi'))$. Otherwise, output $(y \| \pi)$.

Claim 4.4. *Assume that* $\langle \mathsf{Setup}, \mathsf{QAF} \rangle$ *satisfies classical-query hardness (Definition 4.4), then* $\overline{\Pi}$ *is zero-knowledge.*

Proof. We define the following hybrid experiment:

Game G_1: We modify the behavior of the simulator $\mathcal{S}_2$. It computes y and π as normal. However, if $y = \mathsf{accept}$, it aborts. Otherwise, it outputs $y \| \pi$.

For any classical-query QPT adversary $\mathcal{P}^*$, the probability of $\mathcal{P}^*$ making a query with some input x that makes the simulator $\mathcal{S}_2$ abort in G_1 is negligible by classical-query hardness of $\langle \mathsf{Setup}, \mathsf{QAF} \rangle$. Therefore the output of the simulator for $\overline{\Pi}$ is indistinguishable from its output in G_1. Now the zero-knowledge property in G_1 follows directly from the zero-knowledge property of Π, where the reduction samples $(\mathsf{pp}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$, computes $y \leftarrow \mathsf{QAF}(\mathsf{sk}, x)$ on its own and efficiently generates a proof π for the statement (x, y) for each query. □

Using the simulation $\overline{\mathcal{S}}$, we show that our definition is strictly stronger than the classical one below.

Claim 4.5. *Assume that* $\langle \mathsf{Setup}, \mathsf{QAF} \rangle$ *satisfies quantum-query easiness (Definition 4.4), then* $\overline{\Pi}$ *is not quantum-secure simulation-sound.*

Proof. Let $\mathcal{A}$ be the QPT algorithm associated to the quantum-query easiness of $\langle \mathsf{Setup}, \mathsf{QAF} \rangle$. Define $\mathcal{P}^*$ as follows.

1. Run $\mathcal{A}$ by first making a query to $\overline{\mathcal{S}_2}$. Note that the response registers of the query have two component: one to record the output of QAF, the other to record the output of $\mathcal{S}_2$. The first component is initialized as the all-zero string $|0\rangle$, while the second component is initialized as the uniform superposition state $|+\rangle$ to remove the entanglement between the two registers so that after the query, the second response register can be discarded.

2. Continue the execution of $\mathcal{A}$ (with an input x) and obtain a triple $(\pi, (x', \pi'))$. Output two pairs (x, π) and (x', π').

By definition of $\mathcal{S}_2$ and the quantum-query easiness of $\langle \mathsf{Setup}, \mathsf{QAF} \rangle$, both two pairs output by $\mathcal{P}^*$ are valid, and furthermore x' is a false statement, showing that $\overline{\Pi}$ is not quantum-simulation-sound. □

Claim 4.6. *Assume that* $\langle \mathsf{Setup}, \mathsf{QAF} \rangle$ *satisfies classical-query hardness (Definition 4.4), then* $\overline{\Pi}$ *is classically simulation-sound.*

Proof. The proof of this claim follows in an almost identical manner as that of Claim 4.4. □

4.4 Constructions of QSS-NIZK

We devote this section to show that

- In the common reference string model, Sahai's construction of unbounded simulation-sound NIZK [DDOP+01; Sah01], when instantiating with quantum-secure one-time signature scheme (Definition 2.18), is also quantumly simulation-sound (Section 4.4.1).
- In the quantum random oracle model, the Fiat-Shamir transformation [DFMS19; LZ19] is quantumly simulation-sound, if the underlying Sigma protocol satisfies a strong property that we call *randomness collision-resistance* (Section 4.4.2).

4.4.1 Construction in the Common Reference String Model

The Naor commitment scheme. We first recall the bit commitment protocol of Naor [Nao90] based on pseudorandom generators, which will be used later in the construction. Let PRG be a pseudorandom generator stretching λ bits to 3λ bits. The Naor commitment procedure commits to a bit b as follows, using randomness $r \in \{0,1\}^{3\lambda}$ and $s \in \{0,1\}^{\lambda}$.

$$\mathsf{Commit}(b; (r, s)) = \begin{cases} (r, \mathsf{PRG}(s)) & \text{if } b = 0, \\ (r, \mathsf{PRG}(s) \oplus r) & \text{if } b = 1. \end{cases}$$

We note that if PRG is post-quantumly secure (against QPT adversaries with classical access to PRG) then the Naor commitment scheme is also post-quantumly computationally hiding and statistically binding.

Sahai's construction. Let PRF be a family of pseudorandom functions mapping $\{0,1\}^*$ to $\{0,1\}^{\lambda}$. Let $\mathsf{Sig} := \langle \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verif} \rangle$ be a one-time signature scheme. Finally, let Π' be a single-theorem adaptive NIZK systems for a language $\mathcal{L}'$ described below, associated with a QPT simulator $\mathcal{S}' := (\mathcal{S}'_1, \mathcal{S}'_2)$. The construction for a simulation-sound NIZK system Π for some NP language $\mathcal{L}$ is given in Construction 4.3.

Construction 4.3 — QSS-NIZK in the CRS Model [DDOP+01]

Common random string. The random reference string consists of three parts $\mathsf{crs}_1, \mathsf{crs}_2$ and crs_3.

- crs_1 is of length $6\lambda^2$, and breaks up into λ pairs $(r_1, c_1), \ldots, (r_\lambda, c_\lambda)$.
- crs_2 is of length 3λ.
- crs_3 is a common random string of Π'.

Prover. We define the language $\mathcal{L}'$ to be the set of tuples $(x, u, v, \mathsf{crs}_1, \mathsf{crs}_2)$ such that at least one of the following three conditions hold:

- $x \in \mathcal{L}$
- crs_1 consists of commitments to the bits of the λ bit string s: formally, there exists $s = s_1 \cdots s_\lambda$ with $s_i \in \{0,1\}$ for all $i \in [\![1, \lambda]\!]$, and there exists $a_1, \ldots, a_\lambda \in \{0,1\}^\lambda$ such that $(r_i, c_i) = \mathsf{Commit}(s_i; r_i, a_i)$. Furthermore, $u = \mathsf{PRF}(s, v)$.
- There exists $d \in \{0,1\}^\lambda$ such that $\mathsf{crs}_2 = \mathsf{PRG}(d)$.

On input a statement x, a witness ω and the common random string $\mathsf{CRS} := (\mathsf{crs}_1, \mathsf{crs}_2, \mathsf{crs}_3)$, the prover $\mathcal{P}$ does the following:

1. Generate a key pair for the one-time signature scheme: $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Sig.KeyGen}(1^\lambda)$.
2. Sample a uniformly random $u \xleftarrow{\$} \{0,1\}^\lambda$.
3. Using crs_3 as the common random string and ω as the witness, run the prover of Π' to generate a proof that $(x, u, v, \mathsf{crs}_1, \mathsf{crs}_2) \in \mathcal{L}'$. Denote this proof by π'.
4. Output $\pi := (\mathsf{vk}, x, u, \pi', \mathsf{Sig.Sign}(\mathsf{sk}, (x, u, \pi')))$.

Verifier. The verification procedure, on input the instance x, and a proof $\pi := (\mathsf{vk}, x, u, \pi', \sigma)$, with respect to $\mathsf{CRS} := (\mathsf{crs}_1, \mathsf{crs}_2, \mathsf{crs}_3)$ does the following:

1. Verify the validity of the one-time signature: $\mathsf{Sig.Verif}(\mathsf{vk}, (x, u, \pi'), \sigma) = 1$.
2. Verify that π' is a valid proof that $(x, u, \mathsf{vk}, \mathsf{crs}_1, \mathsf{crs}_2) \in \mathcal{L}'$.

Simulator. We now describe the two phases of the simulator $\mathcal{S} := (\mathcal{S}_1, \mathcal{S}_2)$ in Figure 4.1. $\mathcal{S}_1$ outputs a reference string crs along with some trapdoor information td. $\mathcal{S}_2$ takes as input this trapdoor information, the reference string, and an instance x, and outputs a simulated proof for x.

We note that quantum-secure PRFs and the Naor commitment scheme is post-quantumly-secure if quantum-secure one-way functions exist [Zha12a].

$\underline{\mathcal{S}_1(1^\lambda)}$

$s \xleftarrow{\$} \{0,1\}^\lambda$

$r_i \xleftarrow{\$} \{0,1\}^{3\lambda}, a_i \xleftarrow{\$} \{0,1\}^\lambda$ for $i \in [\![1,\lambda]\!]$

$g_i \leftarrow \mathsf{Commit}(s_i; r_i, a_i)$ for $i \in [\![1,\lambda]\!]$

$\mathsf{crs}_1 \coloneqq \{g_1, \cdots, g_\lambda\}$

$\mathsf{crs}_2 \xleftarrow{\$} \{0,1\}^{3\lambda}$

$\mathsf{crs}_3 \leftarrow \Pi'.\mathsf{Setup}(1^\lambda)$

$\mathsf{crs} \coloneqq (\mathsf{crs}_1, \mathsf{crs}_2, \mathsf{crs}_3)$

$\mathsf{td} \coloneqq (s, a_1, \cdots, a_\lambda)$

return $(\mathsf{crs}, \mathsf{td})$

$\underline{\mathcal{S}_2(\mathsf{crs}, \mathsf{td}, x)}$

$(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{Sig.KeyGen}(1^\lambda)$

$u \leftarrow \mathsf{PRF}(s, \mathsf{vk})$

$\pi' \leftarrow \Pi'.\mathcal{S}_2'\Big(\mathsf{crs}_3, (x, u, \mathsf{vk}, \mathsf{crs}_1, \mathsf{crs}_2), (s, a_1, \cdots, a_\lambda)\Big)$

$\sigma \leftarrow \mathsf{Sig.Sign}(\mathsf{sk}, x, u, \pi')$

return $(\mathsf{vk}, x, u, \pi', \sigma)$

Figure 4.1: The simulator of Π.

Theorem 4.2. *If Π' is a single-theorem* quantum *NIZK proof system for $\mathcal{L}'$,* Sig *is a quantum-secure one-time signature scheme and quantum-secure one-way functions exist, the proof system Π described above is an unbounded quantum-simulation-sound NIZK proof system for $\mathcal{L}$.*

Proof. Completeness follows by inspection. Soundness follows by the fact that if crs is chosen uniformly at random, then the probability that crs_1 can be interpreted as a commitment to any string is exponentially small, and likewise the probability that crs_2 is in the image of the pseudorandom generator PRG is exponentially small.

For the proof of adaptive unbounded zero-knowledge, we note that the only difference in the common random string crs between the real protocol and the simulation is crs_1. However, by post-quantum security of the commitment scheme, the two are computationally indistinguishable. (We note that the commitments are classical.) Thus, since the simulator for Π uses only a different witness to prove the same statement, the view of the adversary in the simulator experiment is computationally indistinguishable from the view of the adversary in the modified prover experiment. Thus, adaptive unbounded zero-knowledge follows.

Quantum simulation-soundness proof. The proof of simulation-soundness follows almost identical as the one in the classical setting [Sah01], except for some small modifications on the reductions to quantum security of building blocks. We give the full proof as follows.

Let $\mathcal{A}$ be a QPT adversary. The proof proceeds by a sequence of games where G_0 is defined in which $\mathcal{A}$ can make quantum queries to $\mathcal{S}_2$ (defined in Figure 4.1), and the winning condition is defined as in Definition 4.2. For any game G_i, we denote by $\mathsf{Adv}_i(\mathcal{A})$ the advantage of $\mathcal{A}$ in G_i, that is, $\Pr\big[G_i(1^\lambda, \mathcal{A}) = 1\big]$, where the probability is taken over the random coins of G_i and $\mathcal{A}$.

Game G_0: This is the actual adversary experiment, in which $\mathcal{A}$ can make quantum queries

to the simulator $\mathcal{S}_2$ and outputs two pairs $\{(x_i, \pi_i)\}_{i=1}^2$. Let R be the list of all classical randomness $\mathcal{S}_2$ used to answer each adversarial query during the experiment. We say $\mathcal{A}$ wins if either of the following holds:

(a) There exists $i \in [\![1,2]\!]$ such that $x_i \notin \mathcal{L}$, for all $r \in R$, $\mathcal{S}_2(x_i, r) \neq \pi_i$ and $\mathcal{V}(\mathsf{crs}, x_i, \pi_i) = 1$.

(b) There exists a randomness $r \in R$ such that $\mathcal{S}_2(x_1, r) = \pi_1$ and $\mathcal{S}_2(x_2, r) = \pi_2$ and at least one of x_1 or x_2 is not in $\mathcal{L}$.

Game G_1: In this game, we change the winning condition. The winning condition is now defined as:

(a) There exists $i \in [\![1,2]\!]$ such that $x_i \notin \mathcal{L}$, for all $r \in R$, $\mathcal{S}_2(x_i, r) \neq \pi_i$ and $\mathcal{V}(\mathsf{crs}, x_i, \pi_i) = 1$.

Claim 4.7. *For any* QPT *adversary* $\mathcal{A}$, $|\mathsf{Adv}_0(\mathcal{A}) - \mathsf{Adv}_1(\mathcal{A})| \leq \mathsf{negl}(\lambda)$.

Proof. We show that the probability that the adversary wins by the second condition is negligible, otherwise it must be able to break the unforgeability of the one-time signature. Assume that $\mathcal{A}$ wins by the second condition with non-negligible probability ε.

Let T be the list of verification keys output by the simulator. We note that since verification keys (as well as signing keys) are *classically* and independently of the adversary's queries, T is well-defined as a list of classical strings. Furthermore, with all but exponentially small probability, these verification keys will all be distinct. First, we note that if the output of the adversary can be computed from the same randomness $r \in R$, it means that the verification keys vk_1 and vk_2 (as parts of the proofs) output by the adversary also in T, and furthermore it must be the case that $\mathsf{vk}_1 = \mathsf{vk}_2$, and at least one of the two proofs is a forgery of the signature scheme. Denote this verification key as vk^*, and the forge as (m, t).

We show how to use $\mathcal{A}$ to break the (weak) unforgeability of Sig (the security game W-BZ-Exp as defined in Definition 2.18). Specifically, assume that the adversary $\mathcal{A}$ makes at most q queries to the simulator. The reduction algorithm picks a random index $i \in [\![1, q]\!]$ and uses $\mathcal{A}$'s i-th query in the game W-BZ-Exp. With probability $1/q$, this verification key returned by the challenger in the game W-BZ-Exp is vk^*. In this case, the reduction just returns $\mathcal{A}$'s output pairs $\{(x_i, \pi_i = (\mathsf{vk}^*, x_i, u_i, \pi_i', \sigma_i))\}_{i=1}^2$. If follows that with probability ε/q, $\{((x_i, u_i, \pi_i'), \sigma_i)\}_{i=1}^2$ are valid forges of Sig (with respect to vk^*). We note that $x_1 \neq x_2$ by the assumption. This probability is non-negligible if ε is non-negligible. The proof of the claim follows. □

Game G_2: In this game, we continue changing the winning condition, as follows:

(a) There exists $i \in [\![1,2]\!]$ such that for all $r \in R$, $\mathcal{S}_2(x_i, r) \neq \pi_i$ and $\mathcal{V}(\mathsf{crs}, x_i, \pi_i) = 1$ and $u = \mathsf{PRF}(s, \mathsf{vk})$ where (u, vk) is parts of the output of the proof π_i and s is a part of the trapdoor information td.

We note that now this game can be implemented in quantum-polynomial-time.

Claim 4.8. *For any* QPT *adversary* $\mathcal{A}$, $|\mathsf{Adv}_1(\mathcal{A}) - \mathsf{Adv}_2(\mathcal{A})| \leq \mathsf{negl}(\lambda)$.

Proof. Since crs_2 is a uniformly random string, there is a string d such that $\mathsf{crs}_2 = \mathsf{PRG}(d)$ with only negligible probability. By the definition of the language $\mathcal{L}'$ and the fact that Π' is a proof system for $\mathcal{L}'$, we conclude that if $x \notin \mathcal{L}$, the only way the adversary's proof can be accepted is if $\mathsf{PRF}(s, \mathsf{vk}) = u$ with overwhelming probability. This is because the adversary never sees a valid proof for a false statement of $\mathcal{L}'$ (the simulator is generating the simulated proofs using the commitment witness), thus any adversary that outputs a valid proof for a false statement of $\mathcal{L}'$ (which means $x \notin \mathcal{L} \wedge \mathsf{PRF}(s, \mathsf{vk}) \neq u$) would break the soundness of Π'. Therefore, the winning conditions in G_1 and G_2 are exponentially close. □

Game G_3: In this game, we make crs_2 to be pseudorandom. That is, instead of sampling crs_2 uniformly at random, we compute crs_2 by using a pseudorandom generator PRG. The change is described in Figure 4.2.

$\underline{\mathcal{S}_1(1^\lambda)}$

$d \xleftarrow{\$} \{0,1\}^\lambda$

$s \xleftarrow{\$} \{0,1\}^\lambda$

$r_i \xleftarrow{\$} \{0,1\}^{3\lambda}, a_i \xleftarrow{\$} \{0,1\}^\lambda$ for $i \in [\![1, \lambda]\!]$

$g_i \leftarrow \mathsf{Commit}(s_i; r_i, a_i)$ for $i \in [\![1, \lambda]\!]$

$\mathsf{crs}_1 := \{g_1, \cdots, g_\lambda\}$

$\mathsf{crs}_2 \xleftarrow{\$} \mathsf{PRG}(d)$

$\mathsf{crs}_3 \leftarrow \Pi'.\mathsf{Setup}(1^\lambda)$

$\mathsf{crs} := (\mathsf{crs}_1, \mathsf{crs}_2, \mathsf{crs}_3)$

$\mathsf{td} := (s, a_1, \cdots, a_\lambda)$

return $(\mathsf{crs}, \mathsf{td})$

$\underline{\mathcal{S}_2(\mathsf{crs}, \mathsf{td}, x)}$

$(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{Sig.KeyGen}(1^\lambda)$

$u \leftarrow \mathsf{PRF}(s, \mathsf{vk})$

$\pi' \leftarrow \Pi'.\mathcal{S}_2'\big(\mathsf{crs}_3, (x, u, \mathsf{vk}, \mathsf{crs}_1, \mathsf{crs}_2), (s, a_1, \cdots, a_\lambda)\big)$

$\sigma \leftarrow \mathsf{Sig.Sign}(\mathsf{sk}, x, u, \pi')$

return $(\mathsf{vk}, x, u, \pi', \sigma)$

Figure 4.2: The simulator of game G_3.

Claim 4.9. *For any* QPT *adversary* $\mathcal{A}$, $|\mathsf{Adv}_2(\mathcal{A}) - \mathsf{Adv}_3(\mathcal{A})| \leq \mathsf{negl}(\lambda)$.

Proof. The indistinguishability between G_2 and G_3 follows directly from post-quantum security of PRG. □

Game G_4: In this game, the trapdoor information also includes the seed d of PRG. Furthermore, the simulator $\mathcal{S}_2'$, instead of using the witness of the commitments (that is, $(s, a_1, \cdots, a_\lambda)$), uses the seed d for crs_2 to generate the proof π'. The change is described in Figure 4.3.

Claim 4.10. *For any* QPT *adversary* $\mathcal{A}$, $|\mathsf{Adv}_3(\mathcal{A}) - \mathsf{Adv}_4(\mathcal{A})| \leq \mathsf{negl}(\lambda)$.

$\underline{\mathcal{S}_1(1^\lambda)}$

$d \xleftarrow{\$} \{0,1\}^\lambda$

$s \xleftarrow{\$} \{0,1\}^\lambda$

$r_i \xleftarrow{\$} \{0,1\}^{3\lambda}, a_i \xleftarrow{\$} \{0,1\}^\lambda$ for $i \in [\![1,\lambda]\!]$

$g_i \leftarrow \mathsf{Commit}(s_i; r_i, a_i)$ for $i \in [\![1,\lambda]\!]$

$\mathsf{crs}_1 := \{g_1, \cdots, g_\lambda\}$

$\mathsf{crs}_2 \xleftarrow{\$} \mathsf{PRG}(d)$

$\mathsf{crs}_3 \leftarrow \Pi'.\mathsf{Setup}(1^\lambda)$

$\mathsf{crs} := (\mathsf{crs}_1, \mathsf{crs}_2, \mathsf{crs}_3)$

$\mathsf{td} := (s, a_1, \cdots, a_\lambda, d)$

return $(\mathsf{crs}, \mathsf{td})$

$\underline{\mathcal{S}_2(\mathsf{crs}, \mathsf{td}, x)}$

$(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{Sig.KeyGen}(1^\lambda)$

$u \leftarrow \mathsf{PRF}(s, \mathsf{vk})$

$\pi' \leftarrow \Pi'.\mathcal{S}_2'\big(\mathsf{crs}_3, (x, u, \mathsf{vk}, \mathsf{crs}_1, \mathsf{crs}_2), (d)\big)$

$\sigma \leftarrow \mathsf{Sig.Sign}(\mathsf{sk}, x, u, \pi')$

return $(\mathsf{vk}, x, u, \pi', \sigma)$

Figure 4.3: The simulator of game G_4.

Proof. The indistinguishability between G_3 and G_4 follows the *quantum* zero-knowledge property (Definition 4.1) of Π' (which implies witness-indistinguishability): instead of using witness $(s, a_1, \cdots, a_\lambda)$, we now use witness d to generate the proof. □

Game G_5: In this game, we make crs_1 independent of s: we choose two independent uniformly random strings s, s' and make crs_1 into a commitment to s' rather than s. The change is described in Figure 4.4.

Claim 4.11. *For any* QPT *adversary* $\mathcal{A}$, $|\mathsf{Adv}_4(\mathcal{A}) - \mathsf{Adv}_5(\mathcal{A})| \leq \mathsf{negl}(\lambda)$.

Proof. The indistinguishability between G_4 and G_5 follows the computational hiding property of the Naor's commitment scheme. □

Game G_6: In this game, we replace PRF with a truly random function H (lazy-sampling). The change is described in Figure 4.5.

Claim 4.12. *For any* QPT *adversary* $\mathcal{A}$, $|\mathsf{Adv}_5(\mathcal{A}) - \mathsf{Adv}_6(\mathcal{A})| \leq \mathsf{negl}(\lambda)$.

Proof. The indistinguishability between G_5 and G_6 follows pseudorandomnesss of PRF. Note that here since vk is classical, we only need classical pseudorandomess of PRF against quantum adversaries. □

Claim 4.13. *For any adversary* $\mathcal{A}$, $\mathsf{Adv}_6(\mathcal{A}) \leq 2^{-\lambda}$.

Proof. Since we only consider the case where $\mathsf{vk}^* \notin T$, for any vk^* output by $\mathcal{A}$, $H(\mathsf{vk}^*)$ will be a uniformly selected value that is totally independent of everything the adversary sees. Denote this value as u'. Then the probability that the proof output by $\mathcal{A}$ having $u = u'$ is exactly $2^{-\lambda}$. The claim follows. □

Overall, we conclude the proof of the theorem. □

$\underline{\mathcal{S}_1(1^\lambda)}$

$d \xleftarrow{\$} \{0,1\}^\lambda$

$s, s' \xleftarrow{\$} \{0,1\}^\lambda$

$r_i \xleftarrow{\$} \{0,1\}^{3\lambda}, a_i \xleftarrow{\$} \{0,1\}^\lambda$ for $i \in [\![1, \lambda]\!]$

$g_i \leftarrow \mathsf{Commit}(s'_i; r_i, a_i)$ for $i \in [\![1, \lambda]\!]$

$\mathsf{crs}_1 := \{g_1, \cdots, g_\lambda\}$

$\mathsf{crs}_2 \xleftarrow{\$} \mathsf{PRG}(d)$

$\mathsf{crs}_3 \leftarrow \Pi'.\mathsf{Setup}(1^\lambda)$

$\mathsf{crs} := (\mathsf{crs}_1, \mathsf{crs}_2, \mathsf{crs}_3)$

$\mathsf{td} := (s, a_1, \cdots, a_\lambda, d)$

return $(\mathsf{crs}, \mathsf{td})$

$\underline{\mathcal{S}_2(\mathsf{crs}, \mathsf{td}, x)}$

$(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{Sig.KeyGen}(1^\lambda)$

$u \leftarrow \mathsf{PRF}(s, \mathsf{vk})$

$\pi' \leftarrow \Pi'.\mathcal{S}'_2\big(\mathsf{crs}_3, (x, u, \mathsf{vk}, \mathsf{crs}_1, \mathsf{crs}_2), (d)\big)$

$\sigma \leftarrow \mathsf{Sig.Sign}(\mathsf{sk}, x, u, \pi')$

return $(\mathsf{vk}, x, u, \pi', \sigma)$

Figure 4.4: The simulator of game G_5.

$\underline{\mathcal{S}_1(1^\lambda)}$

$d \xleftarrow{\$} \{0,1\}^\lambda$

$s, s' \xleftarrow{\$} \{0,1\}^\lambda$

$r_i \xleftarrow{\$} \{0,1\}^{3\lambda}, a_i \xleftarrow{\$} \{0,1\}^\lambda$ for $i \in [\![1, \lambda]\!]$

$g_i \leftarrow \mathsf{Commit}(s'_i; r_i, a_i)$ for $i \in [\![1, \lambda]\!]$

$\mathsf{crs}_1 := \{g_1, \cdots, g_\lambda\}$

$\mathsf{crs}_2 \xleftarrow{\$} \mathsf{PRG}(d)$

$\mathsf{crs}_3 \leftarrow \Pi'.\mathsf{Setup}(1^\lambda)$

$\mathsf{crs} := (\mathsf{crs}_1, \mathsf{crs}_2, \mathsf{crs}_3)$

$\mathsf{td} := (s, a_1, \cdots, a_\lambda, d)$

return $(\mathsf{crs}, \mathsf{td})$

$\underline{\mathcal{S}_2(\mathsf{crs}, \mathsf{td}, x)}$

$(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{Sig.KeyGen}(1^\lambda)$

$u \xleftarrow{\$} \{0,1\}^\lambda$

$\pi' \leftarrow \Pi'.\mathcal{S}'_2\big(\mathsf{crs}_3, (x, u, \mathsf{vk}, \mathsf{crs}_1, \mathsf{crs}_2), (d)\big)$

$\sigma \leftarrow \mathsf{Sig.Sign}(\mathsf{sk}, x, u, \pi')$

return $(\mathsf{vk}, x, u, \pi', \sigma)$

Figure 4.5: The simulator of game G_6.

4.4.2 Construction in the Quantum Random Oracle Model

Interactive Proof Systems

In this section, we are mainly interested in a specific class of public-coin interactive proof systems for NP languages, called Σ-protocols. Σ-protocols have a 3-move shape where the first message α, called *commitment*, is sent by the prover and then, alternatively, the parties exchange the other messages β and γ, called (respectively) *challenge* and *response*. Furthermore, the challenge β is public-coin. A formal definition is given below.

Definition 4.5 — Σ-protocols

A Σ-protocol $\Sigma = (\mathcal{P}, \mathcal{V})$ for an NP language $\mathcal{L}$ is a three-round public-coin interactive proof system where $\mathcal{P} = (\mathcal{P}_0, \mathcal{P}_1)$ and $\mathcal{V} = (\mathcal{V}_0, \mathcal{V}_1)$ are PPT algorithms, with the following additional properties:

Completeness. If $x \in \mathcal{L}$, any proper execution of the protocol between $\mathcal{P}$ and $\mathcal{V}$ ends with the verifier accepting $\mathcal{P}$'s proof.

Honest-verifier zero knowledge (HVZK). There exists a QPT algorithm $\mathcal{S}$, called zero-knowledge simulator, such that for any QPT distinguisher $\mathcal{D} = (\mathcal{D}_0, \mathcal{D}_1)$ and for any $(x, w) \in \mathcal{R}_{\mathcal{L}}$, the view of the following two experiments, real and simulated, are computationally indistinguishable:

$\mathsf{Expt}^0_\Sigma(1^\lambda, \mathcal{D})$	$\mathsf{Expt}^1_\Sigma(1^\lambda, \mathcal{D})$
$(x, w, \mathsf{state}) \leftarrow \mathcal{D}_0(1^\lambda)$	$(x, w, \mathsf{state}) \leftarrow \mathcal{D}_0(1^\lambda)$
$\pi \leftarrow \langle \mathcal{P}(1^\lambda, x, w), \mathcal{V}(1^\lambda, x) \rangle$	$\pi \leftarrow \mathcal{S}(1^\lambda, x)$
$b \leftarrow \mathcal{D}_1(\pi, \mathsf{state})$	$b \leftarrow \mathcal{D}_1(\pi, \mathsf{state})$

where $\langle \mathcal{P}(x, w), \mathcal{V}(x) \rangle$ denotes the verdict returned at the end of the interaction between $\mathcal{P}$ and $\mathcal{V}$ on common input x and private input w.

Soundness. If $x \notin \mathcal{L}$ then any malicious (even unbounded) prover $\mathcal{P}^*$ is accepted only with negligible probability.

We say $(\mathcal{P}, \mathcal{V})$ is *non-trivial* if the first message α is computationally indistinguishable from a uniformly random string [FKMV12].

In this section, we also consider a non-standard property of Σ-protocols that we call *randomness collision-resistance* property, which requires that a QPT adversary $\mathcal{A}$ who runs the simulator $\mathcal{S}$ *in superposition*, cannot extract two pairs of $\{(x_i, \pi_i)\}_{i=1}^2$ such that they are both computed using the same randomness *from* the simulator. The definition is formally given as follows.

Definition 4.6 — Randomness collision-resistance

In the following, for each query, the simulator $\mathcal{S}$ stores the (classical) randomness used to answer the query in a list R. For any QPT adversary $\mathcal{A}$, we require that:

$$\Pr\left[\begin{array}{c|c} \exists r \in R : & \\ \mathcal{S}(1^\lambda, x_i; r) = \pi_i \, \forall \, i \in [\![1,2]\!] & \{(x_i,\pi_i)\}_{i=1}^2 \leftarrow \mathcal{A}^{|\mathcal{S}(1^\lambda,\cdot)\rangle}(1^\lambda) \end{array}\right] \leq \mathsf{negl}(\lambda).$$

Remark 4.2. If the zero-knowledge property of the Σ-protocol requires its simulator to use independent randomness for every statement, there is a simple transformation that converts that simulator into another simulator that is secure when a single randomness value is used for an entire query: for each query, choose a fresh random key k for a quantum pseudorandom function (PRF). This will be the single per-query randomness value. To answer a query in superposition, answer each statement x in the superposition using randomness obtained by applying the PRF to x using the key k. From the adversary's point of view, this is indistinguishable from choosing independent randomness for each statement.

Removing Interaction

The Fiat-Shamir paradigm [FS87] applies to any Σ-protocol (and more generally to any three-round public-coin proof system): we start from an interactive protocol $(\mathcal{P},\mathcal{V})$ and remove the interaction between $\mathcal{P}$ and $\mathcal{V}$ by replacing the challenge, chosen at random by the verifier, with a hash value $H(\alpha, x)$ computed by the prover, where H is a hash function modeled as a random oracle, and α is the prover's first message. Thus, the interactive protocol $(\mathcal{P},\mathcal{V})$ is turned into a non-interactive one: The resulting protocol, denoted $(\mathcal{P}^H,\mathcal{V}^H)$, is called Fiat-Shamir proof system.

In this section, we will assume the existence of Σ-protocols with randomness collision-resistance, and prove that the corresponding Fiat-Shamir proof system is quantum simulation-sound. The construction of Σ-protocols with randomness collision-resistance is given in Section 4.4.2.

We note that the security of the Fiat-Shamir transformation in the quantum random oracle model has been proven in [DFMS19; LZ19]. We refer the reader to [DFMS19] for the description of the Fiat-Shamir simulator, for now let us call this simulator the *canonical* simulator.

Notation. Let $H : \mathcal{X}' \to \mathcal{C}$ is a hash function with a domain $\mathcal{X}'$ that contains all pairs (x,α) with $x \in \{0,1\}^n$ and α produced by $\mathcal{P}$, and the range $\mathcal{C}$ is the challenge space of the Σ-protocol. A proof of a Fiat-Shamir system for a statement $x \in \mathcal{L}$ is of the form $\pi := (\alpha, H(x,\alpha), \gamma)$.

Theorem 4.3 (Quantum simulation soundness of the Fiat-Shamir transform). *Consider a* non-trivial *three-round public-coin HVZK interactive proof system $(\mathcal{P},\mathcal{V})$ for a language $\mathcal{L} \in$ NP, with randomness collision-resistance property and super-polynomially sized challenge space. In the quantum random oracle model, the proof system $(\mathcal{P}^H,\mathcal{V}^H)$ derived from $(\mathcal{P},\mathcal{V})$ via the Fiat-Shamir transform is a quantum simulation-sound NIZK (as defined in Definition 4.2) with respect to its canonical simulator $\mathcal{S}$.*

Proof. Completeness, soundness and post-quantum zero-knowledge of the Fiat-Shamir transformation in the quantum random oracle have been proven in [DFMS19; LZ19].

Here, we prove the quantum simulation-soundness. We first recall the measure-and-reprogram in the quantum random oracle lemma introduced in [DFMS19].

Lemma 4.4 ([DFMS19, Theorem 2]). *Let $\mathcal{X}, \mathcal{Y}$ be finite non-empty sets. There exists a black-box quantum polynomial-time two-stage quantum algorithm $\mathcal{S}$ with the following property. Let $\mathcal{A}$ be an arbitrary oracle quantum algorithm that makes q queries to a uniformly random $H : \mathcal{X} \to \mathcal{Y}$ and that outputs some $x \in X$ and a (possibly quantum) output z. Then, the two-stage algorithm $\mathcal{S}^{\mathcal{A}}$ outputs some $x \in \mathcal{X}$ in the first stage and, upon a random $\Theta \in \mathcal{Y}$ as input to the second stage, a (possibly quantum) output z, so that for any $x^\star \in \mathcal{X}$ and any predicate V:*

$$\begin{aligned}&\Pr_{\Theta}\left[x = x^\star \wedge V(x,\Theta,z) = 1 \mid (x,z) \leftarrow \langle \mathcal{S}^{\mathcal{A}}, \Theta\rangle\right] \\ &\quad \geq \frac{1}{\mathcal{O}(q^2)} \Pr_{H}\left[x = x^\star \wedge V(x,H(x),z) = 1 \mid (x,z) \leftarrow \mathcal{A}^{H}\right] - \varepsilon_{x^\star},\end{aligned}$$

where the additive error term $\varepsilon_{x^\star}$ is equal to $\frac{1}{2q|\mathcal{Y}|}$ when summed over all $x^\star$.

Suppose there exists a QPT adversary $\mathcal{A}$ that breaks the quantum simulation-soundness of the non-interactive protocol with non-negligible probability ε. Let $\{(x_i, \pi_i)\}_{i=1}^2$ be the output pairs of $\mathcal{A}$. For simplicity, we denote the *canonical* simulator $\mathcal{S}$ as $(\mathcal{S}_1, \mathcal{S}_2)$, in which $\mathcal{S}_1$ simulates answers to the random oracle H, and $\mathcal{S}_2$ generates simulated proofs. Let q be the number of queries that $\mathcal{A}$ makes to $\mathcal{S}_1$. Without loss of generality, we assume that whenever $\mathcal{A}$ succeeds and outputs two accepting proofs $\{(\alpha_i, \gamma_i)\}_{i=1}^2$, it has previously queried the oracle $\mathcal{S}_1$ on input (x_i, α_i). The argument for this is that it is straightforward to transform any adversary that violates this condition into an adversary that makes two additional classical queries to $\mathcal{S}_1$ and wins with the same probability.

Recall that $\mathcal{A}$ wins if its output satisfies either one of the two conditions defined in Definition 4.2. If $\mathcal{A}$ wins by the first condition, we denote this event as con_1, and similarly the second by the event con_2. Apparently, we have that $\varepsilon := \Pr[\mathcal{A} \text{ wins }] = \Pr[\mathcal{A} \text{ wins } \wedge \mathsf{con}_1] + \Pr[\mathcal{A} \text{ wins } \wedge \mathsf{con}_2]$.

When con_1 happens. This is the case where at least one of the output of $\mathcal{A}$ is computed from a fresh new randomness that were not used by the simulator $\mathcal{S}_2$. Denote this pair as $(x^\star, \pi^\star)$. We will use this adversary $\mathcal{A}$ to build an adversary that breaks the soundness of the underlying Σ protocol.

We fix a family $\mathcal{H}$ of $2(q+2)$-wise independent hash functions and let $\mathcal{S}_1$ simulates the random oracle by choosing a random function $H \in \mathcal{H}$. Furthermore, we observe that, for any fixed x^o, the family $\{H * \Theta x^o \mid H \in \mathcal{H}, \Theta \in \mathcal{C}\}$, is a family of $2(q+2)$-wise independent hash functions as well, where $H * \Theta x^o$ denotes the following function:

$$(H * \Theta x^o)(x) = \begin{cases} H(x) & \text{if } x \neq x^o, \\ \Theta & \text{if } x = x^o. \end{cases}$$

We will use the fact that (even computationally unbounded) q-query adversary $\mathcal{A}$ cannot distinguish a random function $H * \Theta x^o$ in that family from a truly random function H [Zha12a].

Now, we can apply Lemma 4.4 and make a reduction to the standard definition of *soundness* of the underlying Σ protocol. Formally, recall that $(x^\star, \pi^\star = (\alpha^\star, \gamma^\star))$ such that $\mathcal{V}(x^\star, H(x^\star, \alpha^\star), \gamma^\star) = 1$. We will use $(x^\star, \alpha^\star)$ in the execution with the verifier $\mathcal{V}$ of the Σ-protocol. In order to do that, we apply Lemma 4.4, with $(x^\star, \alpha^\star)$ playing the role of what is referred to as x in the lemma statement, the verifier $\mathcal{V}$ is as the predicate V, to obtain the existence of a simulator $\mathcal{S}^{\mathcal{A}}$ that produces $(x^\star, \alpha^\star)$ in a first stage, and upon receiving a random challenge $\beta^\star$ (from the verifier $\mathcal{V}$ of the Σ-protocol) produces $\gamma^\star$, such that:

$$\Pr_{\beta^\star}\left[x = x^\star \wedge \mathcal{V}(x, \alpha^\star, \beta^\star, \gamma^\star) = 1 \mid (x, \alpha^\star, \beta^\star, \gamma^\star) \leftarrow \langle \mathcal{S}^{\mathcal{A}}, \beta^\star \rangle\right]$$
$$\geq \frac{1}{\mathcal{O}(q^2)} \Pr_H\left[x = x^\star \wedge \mathcal{V}(x, \alpha^\star, H(x, \alpha^\star), \gamma^\star) = 1 \mid (x, \alpha^\star, \gamma^\star) \leftarrow \mathcal{A}^H\right] - \varepsilon_{x^\star}.$$

This can be written as

$$\Pr\left[x = x^\star \wedge v = \mathsf{accept} \mid (x, v) \leftarrow \langle \mathcal{S}^{\mathcal{A}}, \mathcal{V} \rangle\right]$$
$$\geq \frac{1}{\mathcal{O}(q^2)} \Pr_H\left[x = x^\star \wedge \mathcal{V}^H(x, \pi^\star) = 1 \mid (x, \pi^\star) \leftarrow \mathcal{A}^H\right] - \varepsilon_{x^\star}, \tag{4.2}$$

where v denote the verifier's output.

We build a reduction $\mathcal{P}^*$ breaking the soundness of the underlying interactive scheme as follows. $\mathcal{P}^*$ uses $\mathcal{S}^{\mathcal{A}}$ (which uses $\mathcal{A}$ as a black-box) to simulate $\mathcal{S}_1$. Whenever $\mathcal{A}$ makes queries to $\mathcal{S}_2$, $\mathcal{P}^*$ runs the HVZK simulator of the interactive protocol (in superposition). It is easy to see that $\mathcal{P}^*$ perfectly simulates the canonical simulator $\mathcal{S}$. Finally, $\mathcal{P}^*$ outputs whatever $\mathcal{S}^{\mathcal{A}}$ outputs.

We note that by the assumption that the Σ-protocol is non-trivial, the probability that in when simulating $\mathcal{S}_2$, the probability that $\mathcal{P}^*$ returns $(x^\star, \alpha^\star, \cdot)$ to $\mathcal{A}$ is negligible. (Note that this is true even when $x^\star$ is the input of the query sent by $\mathcal{A}$.) This means that $(x^\star, \pi^\star)$ must have been computed using a fresh randomness which was not used by $\mathcal{P}^*$. From Equation (4.2), it thus follows that $\Pr[\mathcal{A} \text{ wins } \wedge \mathsf{con}_1]$ is negligible.

When con_2 happens. This is the case where both pairs are computed by the same randomness that were used by the simulator $\mathcal{S}_2$. We will reduce this case to the randomness collision-resistance property of the underlying Σ protocol. Specifically, consider a QPT algorithm $\mathcal{P}^*$ which runs $\mathcal{A}$ internally as a black-box. The description of $\mathcal{P}^*$ is as follows.

- $\mathcal{P}^*$ answers the queries to $\mathcal{S}_1$ using a random $2(q+2)$-wise independent hash function H.
- $\mathcal{P}^*$ keeps a list R of all randomness used to answer queries to $\mathcal{S}_2$. Note that except with exponentially small probability, all the randomess will be distinct.
- Whenever $\mathcal{A}$ outputs fake proofs $\{(x_i, \pi_i = (\alpha_i, \gamma_i))\}_{i=1}^2$, $\mathcal{P}^*$ checks if there exists a randomness $r \in R$ such that $\mathcal{S}_2(x_i, r) = \pi_2$ for $i \in [\![1, 2]\!]$. It then computes $\beta_i = H(x_i, \alpha_i)$.
- $\mathcal{P}^*$ outputs $\{x_i, (\alpha_i, \beta_i, \gamma_i)\}_{i=1}^2$.

Observe that $\mathcal{P}^*$ perfectly simulates $\mathcal{A}$, and thus $\mathcal{P}^*$ breaks the randomness collision-resistance property of the Σ protocol with the same probability that con_2 happens. We obtain $\Pr[\mathcal{A} \text{ wins } \wedge \mathsf{con}_2] \leq \mathsf{negl}(\lambda)$. Overall we have shown that ε is negligible, completing the proof. □

Randomness Collision-resistance Σ-protocols

In this section, we give a compiler that transform any Σ-protocol into a randomness collision-resistance Σ-protocol. Our compiler makes use of quantum-secure one-time signature schemes and is very simple. In the first prover message, the prover also generates a key pair of the signature scheme, and in last message, the prover uses the signing key and signs all the messages that have been exchanged so far (including the one from the original protocol in the last round). The proof of the compiler is almost identical to that of Claim 4.7, we omit the details.

4.5 Application to the Naor-Yung Construction with Quantum CCA Security

In this section, we present and prove quantum security of a simple modification of the classical Naor-Yung scheme [NY90; Sah99]. That is, we show how to construct quantum chosen-ciphertext secure encryption schemes from quantum chosen-plaintext secure schemes and quantum-simulation-sound NIZK proof systems.

4.5.1 Quantum-Secure Invertible Pseudorandom Functions

We first show a construction for invertible pseudorandom functions from standard pseudorandom functions, which will be used later as a building block for our quantum CCA encryption scheme. The construction is the one given in [BKW17].

Construction 4.4 — Invertible Pseudorandom Functions

Let $\mathsf{PRF}_1 : \mathcal{K}_1 \times \mathcal{X} \to \mathcal{Y}$ and $\mathsf{PRF}_2 : \mathcal{K}_2 \times \mathcal{Y} \to \mathcal{X}$ be two pseudorandom functions. Define the following invertible iPRF on domain $\mathcal{X}$ using a key $\mathsf{k} := (\mathsf{k}_1, \mathsf{k}_2) \in \mathcal{K}_1 \times \mathcal{K}_2$:

$\underline{\mathsf{iPRF}((\mathsf{k}_1, \mathsf{k}_2), x)}$

$y_1 \leftarrow \mathsf{PRF}_1(\mathsf{k}_1, x)$

$y_2 \leftarrow \mathsf{PRF}_2(\mathsf{k}_2, y_1) \oplus x$

return (y_1, y_2)

$\underline{\mathsf{iPRF}^{-1}((\mathsf{k}_1, \mathsf{k}_2), (y_1, y_2))}$

$x \leftarrow \mathsf{PRF}_2(\mathsf{k}_2, y_1) \oplus y_2$

if $y_1 \neq \mathsf{PRF}_1(\mathsf{k}_1, x)$

return $\perp$

else return x

Theorem 4.4. *Assume that* $\mathsf{PRF}_1, \mathsf{PRF}_2$ *are quantum-secure (according to Definition 2.7), then* iPRF *in Construction 4.4 is weakly quantum-secure (according to Definition 2.8).*

Proof. We note that in the weak pseudorandom security, the adversary has only quantum access to an evaluation oracle iPRF, and not an inversion oracle iPRF^{-1}. The proof of

the theorem follows from the standard hybrid argument, where we first replace PRF_1 with a truly random function, and then we replace PRF_2 with another truly random function. We omit the details. □

4.5.2 Construction of Our Quantum CCA Encryption Scheme

Our construction uses the following ingredients:

- Let $\mathcal{E} = \langle \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt} \rangle$ be a qIND-qCPA encryption scheme.
- Let $\mathcal{E}' = \langle \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt} \rangle$ be an IND-qCPA encryption scheme.
- Let iPRF be a family of invertible pseudorandom functions.
- Let $\Pi = \langle \mathsf{Setup}, \mathcal{P}, \mathcal{V}, \mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2) \rangle$ be a quantum-simulation-sound NIZK proof system for the language $\mathcal{L}$ of consistent pairs of encryptions, defined formally in Equation (4.3).

$$\begin{aligned} \mathcal{L} := \{ (\mathsf{pk}_0, \mathsf{pk}_1, y_0, y_1, y_2) \ : \ \exists \ (x, \mathsf{k}, r_0, r_1) : \\ y_0 = \mathcal{E}.\mathsf{Encrypt}(\mathsf{pk}_0, x; r_0) \\ \wedge \, y_1 = \mathcal{E}'.\mathsf{Encrypt}(\mathsf{pk}_1, \mathsf{k}; r_1) \wedge y_2 = \mathsf{iPRF}(\mathsf{k}, x) \}. \end{aligned} \tag{4.3}$$

We construct a new encryption scheme $\overline{\mathcal{E}}$ as follows.

Construction 4.5 — Our Quantum CCA Encryption Scheme

$\overline{\mathsf{KeyGen}}(1^\lambda)$:

1 : $\mathsf{crs} \leftarrow \Pi.\mathsf{Setup}(1^\lambda)$
2 : $(\mathsf{pk}_0, \mathsf{sk}_0) \xleftarrow{\$} \mathcal{E}.\mathsf{KeyGen}(1^\lambda)$
3 : $(\mathsf{pk}_1, \mathsf{sk}_1) \xleftarrow{\$} \mathcal{E}'.\mathsf{KeyGen}(1^\lambda)$
4 : $\mathsf{pk} = (\mathsf{crs}, \mathsf{pk}_0, \mathsf{pk}_1)$
5 : $\mathsf{sk} = (\mathsf{crs}, \mathsf{sk}_0, \mathsf{sk}_1)$
6 : **return** $(\mathsf{pk}, \mathsf{sk})$

$\overline{\mathsf{Encrypt}}(\mathsf{pk}, x)$:

1 : $\mathsf{k} \leftarrow \mathsf{iPRF.Setup}(1^\lambda)$
2 : $y_0 \leftarrow \mathcal{E}.\mathsf{Encrypt}(\mathsf{pk}_0, x; r_0)$
3 : $y_1 \leftarrow \mathcal{E}'.\mathsf{Encrypt}(\mathsf{pk}_1, \mathsf{k}; r_1)$
4 : $y_2 \leftarrow \mathsf{iPRF}(k_1, x)$
5 : $\pi \leftarrow \Pi.\mathcal{P}(\mathsf{crs}, (y_0, y_1, y_2), (x, \mathsf{k}, r_0, r_1))$
6 : **return** (y_0, y_1, y_2, π)

$\overline{\mathsf{Decrypt}}(\mathsf{sk}, (y_0, y_1, y_2, \pi))$:

1 : $b \leftarrow \Pi.\mathcal{V}(\mathsf{crs}, (y_0, y_1, y_2), \pi)$
2 : **if** $b = 0$ **then**
3 : **return** $\perp$
4 : **return** $\mathcal{E}.\mathsf{Decrypt}(\mathsf{sk}_0, y_0)$

Theorem 4.5. *The encryption $\overline{\mathcal{E}}$ described in Construction 4.5 above is* qIND-qCCA2 *secure.*

Proof. Let $\mathcal{A}$ be a QPT adversary. For any game G_i, we denote by $\mathsf{Adv}_i(\mathcal{A})$ the advantage of $\mathcal{A}$ in G_i, that is, $\Pr\left[G_i(1^\lambda, \mathcal{A}) = 1\right]$, where the probability is taken over the random coins of G_i and $\mathcal{A}$. The changes in each game are depicted in Figure 4.6.

Game G_0: This is the real-world experiment. In particular, the challenge encryption oracle and the decryption oracle are implemented as follows.

$$\mathcal{RR}_{\mathsf{Encrypt}(\mathsf{pk},\cdot)}\left|x,y\right\rangle \mapsto \left|x,y\oplus\overline{\mathsf{Encrypt}}(\mathsf{pk},x)\right\rangle,$$

and

$$\mathcal{O}_{\mathsf{Decrypt}(\mathsf{sk},\cdot)}\left|y,x\right\rangle \mapsto \left|z,x\oplus\overline{\mathsf{Decrypt}}(\mathsf{sk},y)\right\rangle.$$

Game G_1: This is identical to G_0, except that now in the decryption oracle, instead of using sk_0, we use sk_1, combining with the fact that iPRF is invertible for the decryption.

Claim 4.14. *For any adversary* $\mathcal{A}$, $\mathsf{Adv}_1(\mathcal{A}) = \mathsf{Adv}_0(\mathcal{A})$.

Proof. The proof of the claim follows directly from the correctness of encryption schemes $\mathcal{E}$, $\mathcal{E}'$ and the fact that iPRF is invertible. □

Game G_2: This is identical to G_1, except that now in the challenge encryption oracle, we use the simulator $\mathcal{S}$ of Π to generate the proof instead of using the real prover $\mathcal{P}$.

Claim 4.15. *For any* QPT *adversary* $\mathcal{A}$, $|\mathsf{Adv}_2(\mathcal{A}) - \mathsf{Adv}_1(\mathcal{A})| \leq \mathsf{negl}(\lambda)$.

Proof. The indistinguishability between G_1 and G_2 follows from zero-knowledge property of Π. □

Game G_3: This is identical to G_2, except now in the challenge encryption oracle, instead of encrypting using the actual encryption algorithm $\mathcal{E}.\mathsf{Encrypt}$, we use the encryption oracle in the random world of $\mathcal{E}$. Denote this oracle as $\mathcal{R}_{\mathsf{Encrypt}(\mathsf{pk}_0,\cdot)}$.

Claim 4.16. *For any* QPT *adversary* $\mathcal{A}$, $|\mathsf{Adv}_3(\mathcal{A}) - \mathsf{Adv}_2(\mathcal{A})| \leq \mathsf{negl}(\lambda)$.

Proof. We note that in G_2 and G_3, the secret key sk_0 is not used at all. The indistinguishability between G_2 and G_3 follows immediately from qIND-qCPA security of $\mathcal{E}$. □

We note that starting from G_3, the challenge encryption oracle can be implemented as a compressed encryption oracle (sine we are now in the random world of $\mathcal{E}$). Concretely, the challenge encryption oracle implements the following map:

$$\sum_{x,y}\alpha_{x,y}\left|x,y\right\rangle \mapsto \sum_{x,y}\sum_{u}\alpha_{x,y}\left|x,\mathcal{E}.\mathsf{Encrypt}(\mathsf{pk}_0,u)\|\mathcal{E}'.\mathsf{Encrypt}(\mathsf{pk}_1,\mathsf{k})\|\mathsf{iPRF}(\mathsf{k},x)\|\pi\right\rangle\otimes\left|D\right\rangle$$

where D is the database of the compressed random encryption oracle for $\mathcal{E}$. In particular, D will be in superposition of tuples (x,u,y_0) (if $D(x)\neq\perp$). Furthermore, we note that if $D(x)\neq\perp$, we can re-compute (y_1,y_2,π) and also store these values in the corresponding slot in D. The reason is that from $x\in D$, these values can be computed with the *classical* randomness used in the challenge encryption oracle of $\overline{\mathcal{E}}$.

Game G_4: This is identical to G_3, except that now instead of using sk_1 in the decryption oracle, we use sk_0 and D.

Claim 4.17. *For any* QPT *adversary* $\mathcal{A}$, $|\mathsf{Adv}_4(\mathcal{A}) - \mathsf{Adv}_3(\mathcal{A})| \leq \mathsf{negl}(\lambda)$.

Proof. We show that if $\mathcal{A}$ can distinguish the two games G_3 and G_4 with non-negligible probability ε, then we can construct a QPT adversary $\mathcal{B}$ that runs $\mathcal{A}$ internally as a black-box and breaks the quantum simulation-soundness of Π with non-negligible probability. Notice that the only way $\mathcal{A}$ can distinguish G_3 and G_4 is to submit an "invalid" decryption query in which the proof π if of a false statement but the verification passes.

Formally, $\mathcal{B}$ runs $\mathcal{A}$ and randomly measure one of $\mathcal{A}$'s decryption queries to obtain a tuple $y^* = (y_0^*, y_1^*, y_2^*, \pi^*)$. If $\mathcal{A}$ makes at most q decryption queries, then with probability at least ε/q, y^* will be a pair of statement and proof such that the statement is a false statement but π^* passes the verification of Π. Then $\mathcal{B}$ measure its own database D to obtain another tuple $y = (y_0, y_1, y_2, \pi)$ which is supposed to be generated by the simulator of Π. By the definition of $\overline{\mathsf{Decrypt}}^5$, we have that $y \neq y^*$. Thus by outputting (y, y^*), $\mathcal{B}$ breaks the quantum simulation-soundness of Π with probability ε/q, which completes the proof of the claim. □

We note that from starting from this game, the secret key sk_1 is not used anymore.

Game G_5: This is identical to G_4, except that now in the challenge encryption oracle, we change the encryption $\mathcal{E}'.\mathsf{Encrypt}(\mathsf{pk}_1, \mathsf{k})$ for some random key k by an encryption $\mathcal{E}'.\mathsf{Encrypt}(\mathsf{pk}_1, 0)$.

Claim 4.18. *For any* QPT *adversary* $\mathcal{A}$, $|\mathsf{Adv}_5(\mathcal{A}) - \mathsf{Adv}_4(\mathcal{A})| \leq \mathsf{negl}(\lambda)$.

Proof. The indistinguishability between G_4 and G_5 follows immediately from IND-qCPA security of $\mathcal{E}'$. Note that since here the encryption is a classical encryption of a classical random key k (which is independent of $\mathcal{A}$'s query), we only need qCPA security against classical challenge query of $\mathcal{E}'$. □

Game G_6: This is identical to G_5, except that now in the challenge encryption oracle, instead of computing y_2 as $\mathsf{iPRF}(\mathsf{k}, x)$, we compute $y_2 \leftarrow \mathsf{iPRF}(\mathsf{k}, u)$ where u is extracted from the database D (note that $u \in D(x)$). We abuse the notation and write $D(x) = u$. Furthermore, for consistency, we also allow $\mathcal{E}$'s encryption algorithm to take as input the database D.

Claim 4.19. *For any* QPT *adversary* $\mathcal{A}$, $|\mathsf{Adv}_6(\mathcal{A}) - \mathsf{Adv}_5(\mathcal{A})| \leq \mathsf{negl}(\lambda)$.

Proof. The indistinguishability between G_5 and G_6 follows immediately from (weak) quantum-pseudorandomness of iPRF. We note that here we only need weak security notion, since iPRF^{-1} is never invoked in the decryption oracle. □

Game G_7: This is identical to G_6, except that now in the challenge encryption oracle, instead of computing y_1 as an encryption of 0, we change it back to encryption of a random key k, that is $y_1 \leftarrow \mathcal{E}'.\mathsf{Encrypt}(\mathsf{pk}_1, \mathsf{k})$.

Claim 4.20. *For any* QPT *adversary* $\mathcal{A}$, $|\mathsf{Adv}_7(\mathcal{A}) - \mathsf{Adv}_6(\mathcal{A})| \leq \mathsf{negl}(\lambda)$.

Proof. The indistinguishability between G_6 and G_7 follows immediately from IND-qCPA security of $\mathcal{E}'$. □

Game G_8: This is identical to G_7, except that now in the challenge encryption oracle, we use the real prover $\mathcal{P}$ of Π to generate the proof instead of using the simulator.

Claim 4.21. *For any* QPT *adversary* $\mathcal{A}$, $|\mathsf{Adv}_8(\mathcal{A}) - \mathsf{Adv}_7(\mathcal{A})| \leq \mathsf{negl}(\lambda)$.

Proof. The indistinguishability between G_7 and G_8 follows from zero-knowledge property of Π. □

In this final game G_8, we have the challenge encryption oracle implements exactly as the one in the random-world of $\overline{\mathcal{E}}$. Overall, we complete the proof of the theorem. □

$G_1 : \overline{\mathsf{Decrypt}}(\mathsf{sk}, (y_0, y_1, y_2, \pi))$

$b \leftarrow \Pi.\mathcal{V}(\mathsf{crs}, (y_0, y_1, y_2), \pi)$

if $b = 0$ **then return** $\perp$

$\mathsf{k} \leftarrow \mathcal{E}'.\mathsf{Decrypt}(\mathsf{sk}_1, y_1)$

return $\mathsf{iPRF}^{-1}(\mathsf{k}, y_2)$

$G_2 : \overline{\mathsf{Encrypt}}(\mathsf{pk}, x)$

$\mathsf{k} \xleftarrow{\$} \mathsf{iPRF.Setup}(1^\lambda)$

$y_0 \leftarrow \mathcal{E}.\mathsf{Encrypt}(\mathsf{pk}_0, x; r_0)$

$y_1 \leftarrow \mathcal{E}'.\mathsf{Encrypt}(\mathsf{pk}_1, \mathsf{k}; r_1)$

$y_2 \leftarrow \mathsf{iPRF}(\mathsf{k}, x)$

$\pi \leftarrow \Pi.\mathcal{S}(\mathsf{crs}, (y_0, y_1, y_2))$

return (y_0, y_1, y_2, π)

$G_3 : \overline{\mathsf{Encrypt}(\mathsf{pk}, x)}$

$\mathsf{k} \xleftarrow{\$} \mathsf{iPRF.Setup}(1^\lambda)$

$y_0 \leftarrow \mathcal{R}_{\mathsf{Encrypt}(\mathsf{pk}_0, \cdot)}(x)$

$y_1 \leftarrow \mathcal{E}'.\mathsf{Encrypt}(\mathsf{pk}_1, \mathsf{k}; r_1)$

$y_2 \leftarrow \mathsf{iPRF}(\mathsf{k}, x)$

$\pi \leftarrow \Pi.\mathcal{S}(\mathsf{crs}, (y_0, y_1, y_2))$

return (y_0, y_1, y_2, π)

$G_4 : \overline{\mathsf{Decrypt}}(\mathsf{sk}, (y_0, y_1, y_2, \pi), D)$

$b \leftarrow \Pi.\mathcal{V}(\mathsf{crs}, (y_0, y_1, y_2), \pi)$

if $b = 0$ **then return** $\perp$

if $\exists (x, (y_0, y_1, y_2, \pi)) \in D$ **then**

  return x

return $\mathcal{E}.\mathsf{Decrypt}(\mathsf{sk}_0, y_0)$

$G_5 : \overline{\mathsf{Encrypt}}(\mathsf{pk}, x)$

$\mathsf{k} \xleftarrow{\$} \mathsf{iPRF.Setup}(1^\lambda)$

$y_0 \leftarrow \mathcal{R}_{\mathsf{Encrypt}(\mathsf{pk}_0, \cdot)}(x)$

$y_1 \leftarrow \mathcal{E}'.\mathsf{Encrypt}(\mathsf{pk}_1, 0; r_1)$

$y_2 \leftarrow \mathsf{iPRF}(\mathsf{k}, x)$

$\pi \leftarrow \Pi.\mathcal{S}(\mathsf{crs}, (y_0, y_1, y_2))$

return (y_0, y_1, y_2, π)

$G_6 : \overline{\mathsf{Encrypt}}(\mathsf{pk}, x, D)$

$u \leftarrow D(x)$

$\mathsf{k} \xleftarrow{\$} \mathsf{iPRF.Setup}(1^\lambda)$

$y_0 \leftarrow \mathcal{R}_{\mathsf{Encrypt}(\mathsf{pk}_0, \cdot)}(x)$

$y_1 \leftarrow \mathcal{E}'.\mathsf{Encrypt}(\mathsf{pk}_1, 0; r_1)$

$y_2 \leftarrow \mathsf{iPRF}(\mathsf{k}, u)$

$\pi \leftarrow \Pi.\mathcal{S}(\mathsf{crs}, (y_0, y_1, y_2))$

return (y_0, y_1, y_2, π)

$G_7 : \overline{\mathsf{Encrypt}}(\mathsf{pk}, x, D)$

$u \leftarrow D(x)$

$\mathsf{k} \xleftarrow{\$} \mathsf{iPRF.Setup}(1^\lambda)$

$y_0 \leftarrow \mathcal{R}_{\mathsf{Encrypt}(\mathsf{pk}_0, \cdot)}(x)$

$y_1 \leftarrow \mathcal{E}'.\mathsf{Encrypt}(\mathsf{pk}_1, \mathsf{k}; r_1)$

$y_2 \leftarrow \mathsf{iPRF}(\mathsf{k}, u)$

$\pi \leftarrow \Pi.\mathcal{S}(\mathsf{crs}, (y_0, y_1, y_2))$

return (y_0, y_1, y_2, π)

$G_8 : \overline{\mathsf{Encrypt}}(\mathsf{pk}, x, D)$

$u \leftarrow D(x)$

$\mathsf{k} \xleftarrow{\$} \mathsf{iPRF.Setup}(1^\lambda)$

$y_0 \leftarrow \mathcal{R}_{\mathsf{Encrypt}(\mathsf{pk}_0, \cdot)}(x; r_0)$

$y_1 \leftarrow \mathcal{E}'.\mathsf{Encrypt}(\mathsf{pk}_1, \mathsf{k}; r_1)$

$y_2 \leftarrow \mathsf{iPRF}(\mathsf{k}, u)$

$\pi \leftarrow \Pi.\mathcal{P}(\mathsf{crs}, (y_0, y_1, y_2), (x, r_0, r_1))$

return (y_0, y_1, y_2, π)

Figure 4.6: Description of the changes in games G_i for $i \in [\![1, 8]\!]$. In each program, the changes relative to the previous program are highlighted in light gray.

Part II

Quantum Cryptography

Chapter

5 Semi-Quantum Copy-Protection

> *"When it's classically impossible, that really just means it's quantumly interesting."*

In this chapter, we present a generic compiler that translates many known constructions of quantum copy-protection to semi-quantum copy-protection. This particularly allows us to obtain (the first) semi-quantum copy-protection for digital signatures, public-key decryption, pseudorandom functions and point functions.

Chapter content

5.1 Introduction

5.1.1 Quantum Cryptography From Coset States

Given a subspace $A \subseteq \mathbb{F}_2^n$, the corresponding *subspace state* is defined as a uniform superposition over all vectors in the subspace A, i.e., $|A\rangle := \frac{1}{\sqrt{|A|}} \sum_{v \in A} |v\rangle$. The idea of using hidden subspace state to construct quantum cryptographic primitives was first proposed by Aaronson and Christiano in [AC12] in the oracle model where the parties have access to some membership checking oracles. This idea was realized subsequently in the plain model using indistinguishability obfuscation by Zhandry [Zha19b]. The subspace state idea was later generalized to *coset states* in [CLLZ21; VZ21], which can be seen as quantum one-time pad encrypted subspace states. Formally, for a subspace $A \subseteq \mathbb{F}_2^n$ and two vectors $s, s' \in \mathbb{F}_2^n$, the corresponding coset state is defined as $|A_{s,s'}\rangle := \frac{1}{\sqrt{|A|}} \sum_{x \in A} (-1)^{\langle x, s' \rangle} |x + s\rangle$. The coset state idea has shown a broad range of applications to signature tokens, unclonable decryptors, copy-protection [CLLZ21], classical proof of quantum knowledge [VZ21], public semi-quantum money [Shm22a], semi-quantum signature tokens [Shm22b], and unclonable encryption [AKLL+22]. Indeed, to the best of our knowledge, all known provably secure copy-protection schemes with standard malicious security are based on hidden coset states [CLLZ21; AKLL+22]. In these protocols, the basic ground is to encode the program into random hidden coset states and send these states as copy-protection of the program to the receiver.

5.1.2 (Semi-)Quantum Cryptography From BB84 States

In a breakthrough result [Mah18b], Mahadev introduced a protocol that allows a classical verifier to verifiably delegate a quantum computation to an untrusted quantum prover. The key ingredient of Mahadev's protocol is a *measurement* protocol, which allows the client to delegate single-qubit measurements in the standard or Hadamard basis to a quantum prover, and be able to efficiently verify the measurement outcome, assuming that the prover cannot break certain cryptographic assumptions.[8] This yields the first kind of semi-quantum protocols, so-called *prepare-and-measure protocols*. These protocols involve a quantum prover preparing and sending a quantum state to the verifier and the verifier performing single-qubit measurements on this state. One can use Mahadev's measurement protocol to delegate these quantum measurements to the prover itself. This is in contrast to the other type of quantum protocols: *prepare-and-send* protocols, in which the verifier prepares and sends quantum states to the prover.

It turns out that replacing the quantum communication of prepare-and-send protocols is significantly harder than doing so for prepare-and-measure protocols. At a high level, the main difference is the following: Mahadev's measurement protocol shows that *there exists* a quantum state that is consistent with the distribution of measurement outcomes reported by the prover. (One can think about this similarly as the *proof of membership* in an interactive proof system.) In contrast, if we want to replace the step of the verifier sending a physical quantum state to the prover, we need to show that the prover has *actually constructed* a certain quantum state, not just that such a quantum state exists. This is done by establishing a *rigidity* argument. The idea of rigidity, first formally

[8] These cryptographic assumptions can be based on the quantum hardness of the Learning with Errors problem [Reg05].

introduced by Mayers and Yao [MY04], is that certain games can be used to "self-test" quantum states: if such a game is won with high enough probability, then the self-test property tells us that the players must hold some quantum state, up to local isometry. In our context, a game is a model of the protocol under consideration, and the game is won if the prover passes the client's verification.

Lying strictly between the two notions of "*there exists*" and "*actually constructed*" is the notion of *classical proof of quantum knowledge* formalized in [VZ21]. They also show that indeed Mahadev's measurement protocol achieves this stronger notion of *proof of knowledge*, in the sense that the prover "*knows*" the state it is measuring, not just that it exists mathematically. The same property also holds for a coset states prepare-and-send protocol: if the verifier sends a hidden coset state to the prover, later on the prover can prove that it "knows" the received coset state. Although this setting seems close to our protocol, we note that the proof given in [VZ21] does not directly carry to our setting, where we replace quantum communication by classical communication.[9]

The first semi-quantum protocol that provably forces a quantum prover to prepare a certain quantum state is the single-qubit remote state preparation protocol of [GV19] (see also [CCKW19] for a related result). [MV21] gives a protocol that allows a classical verifier to certify that a quantum prover must have prepared and measured a Bell state, i.e., an entangled 2-qubit quantum state. Finally, [GMP22], by developing new techniques to show a n-fold parallel rigidity proof, gives the first parallel remote BB84 state preparation protocol. Recall that *BB84 states* are the four states: $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$ where $|\pm\rangle := \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$. Their proof technique is the backbone of our soundness proof presented later in Section 5.5. The most interesting point of the [GMP22] protocol is that it allows us to dequantize a number of BB84 states-based quantum cryptographic primitives, yielding a generic and modular way of translating these protocols to a setting where only classical communication is used. The downside of the [GMP22] protocol is that it only achieves *inverse polynomial* soundness, which means that their dequantized protocols can only achieve *inverse polynomial* security at most, even if the original quantum protocols have negligible security.

We remark that the main distinction between random BB84 states and coset states is the (un)learnability with verification oracles: when the verification oracles are accessible, the former is learnable while the latter is unlearnable. This explains why coset states have more applications, mostly in the public-key setting.

5.1.3 Application-specific Approaches for Semi-Quantum Protocols

In addition to this line of work focused on rigidity statements, application-specific semi-quantum protocols were considered for private-key quantum money [RS20], certifiable deletion of quantum encryption [HMNY21], secure software leasing [KNY21], public-key quantum money [Shm22a] and tokenized signature [Shm22b]. The common points of these protocols are that (i) their approaches are less generic and modular than the [GMP22] protocol and the protocol we presented in this thesis; (ii) new analysis are required for each application. However, we note that all these application-specific semi-quantum protocols achieve *negligible* security, as they do not prove that the prover in their protocol behave in a certain way, but only that the output of the prover at the end satisfies certain properties.

[9] Even if we can prove security for the dequantized protocol, it is not clear if it is applicable (with little efforts) to other cryptographic constructions of interest.

5.2 Technical Overview

5.2.1 Our Semi-Quantum Copy Protection Protocol

Security Requirements. We first start with security analysis of known coset states-based quantum protocols. In particular, we focus on the copy-protection point pseudorandom functions scheme in the plain model and the single-decryptor scheme in the plain model presented in [CLLZ21]. The security of these constructions reduce to a *monogamy of entanglement* property of coset states [CLLZ21; CV22]. Informally, this property states that for a triple of quantum algorithms Alice, Bob and Charlie cannot cooperatively win the following monogamy game with a challenger, except with negligible probability. The challenger first prepares a uniformly random coset state $|A_{s,s'}\rangle$ and gives the state to Alice. Alice outputs two (possibly entangled) quantum states and sends them to Bob and Charlie respectively. No communication is allowed between Bob and Charlie. Finally, Bob and Charlie both get the description of the subspace A. The game is won if Bob outputs a vector in $A+s$ and Charlie outputs a vector in $A^{\perp}+s'$, where $A^{\perp}$ denote the dual subspace of A.

If our goal is to design a semi-quantum protocol for preparing coset states such that it can be used in a plug-and-play manner for the aforementioned protocols, our protocol needs to have the following properties:

- *Completeness*. If the prover is honest, at the end of the protocol execution, the prover must have a hidden coset state $|A_{s,s'}\rangle$ in its registers.
- *Soundness*. Any (computationally bounded) prover after interacting with the classical verifier in the protocol, cannot win the monogamy of entanglement game described above (with a single modification in the first step of the game: instead of sending the coset state to the prover, we run the protocol). For a formal definition of the soundness, see Definition 5.12. We note that the soundness property also implies the blindness property: a untrusted prover cannot know the description of A and s, s' through the interaction.

The first attempt. Having described all requirements needed, we now turn into our protocol construction. Our starting point is the recent public semi-quantum money in the plain model introduced by Shmueli in [Shm22a], which uses hybrid quantum homomorphic encryption (QFHE)[10] and indistinguishability obfuscation ($i\mathcal{O}$) as the building blocks. The scheme is as follows.

1. The classical verifier $\mathcal{V}$ samples a random $\frac{\lambda}{2}$-dimensional subspace $A \subseteq \mathbb{F}_2^{\lambda}$ (represented by a matrix $\mathbf{M}_A \in \{0,1\}^{\frac{\lambda}{2}\times\lambda}$), and sends to the prover $\mathcal{P}$ $(\mathbf{M}_A^{p_x}, \mathsf{ct}_{p_x})$, an encryption of the matrix $\mathbf{M}_A$ under QFHE.
2. $\mathcal{P}$ homomorphically evaluates the circuit C, which is a quantum circuit that gets as input the classical description of a subspace $A \subseteq \mathbb{F}_2^{\lambda}$ and generates a uniform superposition over A. $\mathcal{P}$ obtains a homomorphically evaluated ciphertext

$$(|A_{x,z}\rangle, \mathsf{ct}_{x,z}) \leftarrow \mathsf{QFHE.Eval}\Big(\mathsf{pk}, (\mathbf{M}_A^{p_x}, \mathsf{ct}_{p_x}), C\Big),$$

[10] Recall that a hybrid QFHE scheme is one where every encryption of a quantum state $|\psi\rangle$ consists of a quantum one-time pad encryption of $|\psi\rangle$ with Pauli keys $(x,z) \in \{0,1\}^*$, and $\mathsf{ct}_{x,z}$ which is a classical FHE encryption of the Pauli keys.

and sends the classical part $\mathsf{ct}_{x,z}$ to $\mathcal{V}$.

3. $\mathcal{V}$ decrypts $(x, z) \leftarrow \mathsf{QFHE.Decrypt}(\mathsf{sk}, \mathsf{ct}_{x,z})$ and sends obfuscated membership check programs $i\mathcal{O}(A + x)$, $i\mathcal{O}(A^{\perp} + z)$ to $\mathcal{P}$.

Unfortunately, there is an efficient "splitting" attack that breaks the monogamy game described above (even if the adversary does not receive the description of A in the question phase) (see [Shm22b] for the description of the attack). Indeed, in the construction of semi-quantum tokenized signatures [Shm22b], which also based on the [Shm22a] construction above, the author also needs to overcome this problem by carefully changing the security property required for the signature setting. We forego the details of his approach, but we note that his approach is unlikely to be applicable in our setting. The main difference is that in our setting, there are two simultaneous non-communicating adversaries that also receive the description of A in the monogamy game.

The second attempt: running self-testing protocol under QFHE. We make an important observation: in the semi-quantum protocol for preparing BB84 states (also called a self-testing protocol) of [GMP22], instead of asking the prover $\mathcal{P}$ to prepare his own states (which are polynomially many $|+\rangle$ states if $\mathcal{P}$ is honest), the verifier $\mathcal{V}$ can send the input to $\mathcal{P}$ using QFHE. In particular, $\mathcal{V}$ sends encryption of $\mathbf{M}_0$, which is the all-zero matrix. $\mathcal{P}$ homomorphic evaluates a quantum circuit C on the received ciphertext such that if the input matrix is all-zero, C evaluates to a uniform superposition over $\mathbb{F}_2^{\lambda}$, which is exactly product of $|+\rangle$ states. Under QFHE encryption, the quantum part of the evaluated ciphertext is product of random $|\pm\rangle$ states. $\mathcal{P}$ then uses this in the [GMP22] self-testing protocol. We will see that an honest prover $\mathcal{P}$ using product of $|+\rangle$ states as in the [GMP22] protocol or $\mathcal{P}$ using product of $|\pm\rangle$ states does not change the completeness of the protocol, while its soundness is maintained (since the soundness does not depend on which input the prover has been used in the protocol execution).

We now briefly give a description of the [GMP22] protocol, and refer the reader to their paper and Section 5.4.1 for more details. The verifier first runs a number of *test* rounds, where the prover is asked to measure its entire quantum state. These test rounds are used by the verifier to check whether the prover behaves as intended. Once the verifier is convinced of this, the verifier runs a *preparation* round. Test and preparation rounds are indistinguishable from the point of view of the prover, except that unlike in a test round, in a preparation the prover is not asked to measure its final state. Essentially, the [GMP22] protocol can be seen as a 1-over-n cut-and-choose protocol, in which the verifier run n rounds of testing, and 1 round of preparation from $n + 1$ indistinguishable instances. The soundness statement for the test rounds is a self-testing statement, which characterizes which states and measurements the prover used in the protocol. The soundness of the [GMP22] protocol follows from that of the test rounds via a statistical cut-and-choose argument. In the following, we focus on the test sub-protocol.

The main cryptographic primitive underlying the [GMP22] protocol (as well as other self-testing protocols [GV19; MV21]) is the so-called extended noisy trapdoor claw-free function (ENTCF) family, which can be constructed assuming the quantum hardness of LWE [Mah18b]. Recall that an ENTCF family is a family of functions indexed by a set of keys $\mathcal{K}_0 \cup \mathcal{K}_1$. $\mathcal{K}_0$ and $\mathcal{K}_1$ are disjoint sets of keys with the property that the two sets are computationally indistinguishable.

We first describe the single-qubit remote preparation protocol from [GV19], as the [GMP22] test protocol is a n-fold parallel of [GV19].

1. For a given *basis choice* $\theta \in \{0,1\}$ (where "0" corresponds to the computational and "1" to the Hadamard basis), the verifier $\mathcal{V}$ samples a key $k \in \mathcal{K}_\theta$, alongside some trapdoor information t. $\mathcal{V}$ sends k to the prover $\mathcal{P}$ and keeps t private.

2. The verifier and prover then interact classically.

3. For us, the most relevant part is the last round of the protocol, i.e., the last message from the verifier to the prover and back. Before the last round, the remaining quantum state of an *honest* prover is the single-qubit state $|v\rangle_\theta$ for $v \in \{0,1\}$, where $|v\rangle_\theta$ is a conjugate encoding of v in the basis θ: if $\theta = 0$, $|v\rangle_\theta = |v\rangle$, otherwise $|v\rangle_\theta = \frac{1}{\sqrt{2}}(|0\rangle + (-1)^v |1\rangle)$. From the transcript and the trapdoor information, the verifier can compute v; in contrast, the prover, who does not know the trapdoor, cannot efficiently compute θ or v. In the last round, the verifier sends θ to the prover, who returns $v' \in \{0,1\}$; the verifier then checks whether $v' = v$. The honest prover would generate v' by measuring its remaining qubit $|v\rangle_\theta$ in the basis θ and therefore always pass the verifier's check.

In the [GMP22] test protocol, $\mathcal{V}$ runs n independent copies of [GV19] in parallel, except that the basis choice θ_i is the same for each copy. Next, from the [GMP22] protocol, we describe a self-testing protocol for coset states.

Assume that now the verifier has private input which is a description of a coset state (A, x, z). We modify the verification procedure of the [GMP22] test protocol in the last round as follows. Let $\vec{v}$ be the last message sent by $\mathcal{P}$ to $\mathcal{V}$ in the protocol above. If $\theta = 0$ (note that the basis choice is the same for n copies), $\mathcal{V}$ checks if $\vec{v} \in A + x$, otherwise, it decodes[11] $\vec{v}$ to get a vector $\vec{v}'$ and checks if $\vec{v}' \in A^\perp + z$. An honest prover would use the coset state $|A_{x,z}\rangle$, which it obtains after running the [Shm22a] protocol described above, as the input to this self-testing protocol. The honest prover would have measured its state in the computational basis when $\theta = 0$, and in the Hadamard basis when $\theta = 1$. Thus, any honest prover would pass this self-testing protocol for coset states with probability 1.

The crucial point is that, since the prover's input in both the [GMP22] self-testing protocol and the self-testing protocol for coset states described above is encrypted under QFHE, and the fact that the two protocols are identical from the prover point's of view (except the last verification procedure, which is hidden from the prover), the two protocols are computationally indistinguishable. In other words, any computationally bounded prover cannot distinguish if it is playing in the [GMP22] self-testing protocol or the coset-state self-testing protocol. This allows us to carry the rigidity argument of the [GMP22] protocol to our setting. We elaborate more on this later in Section 5.2.2. For time being, let's say we have showed that if the prover $\mathcal{P}$ passes the verification, it must have "used" a coset state in the self-testing protocol (with inverse polynomial soundness).

However, our ultimate goal is to perform a remote state preparation protocol (and not just self-testing). Our final step would be to run this coset-state self-testing protocol in the n-over-$2n$ cut-and-choose fashion: the verifier first sends $2n$ encrypted coset

[11] We omit the details of this decoding procedure, and refer the reader to Section 5.4.1. We note that with the trapdoor t, this procedure can be implemented efficiently by the verifier.

states and $|+\rangle$ to the prover, and it picks n instances uniformly at random for the self-testing protocol. The remaining n instances are used as the output of the final protocol. Building on the simple but powerful "quantum cut-and-choose" formalism of Bouman and Fehr [BF10], we can show that if the prover passes all the test instances, it must have at least 1 coset state in its registers at the end of the protocol (with inverse polynomial soundness). Notably, we will show that even if we only obtain inverse polynomial soundness at this step, our final protocol still achieves negligible security for a monogamy of entanglement game, which is the main property used in many copy-protection schemes.

Our final protocol. Our final protocol (Protocol 5.5) works as follows:

(1) The verifier first sends homomorphic encryption that allows the prover to either construct coset states or BB84 states.

(2) The prover is asked to homomorphically evaluate the instructed circuits and return classical encryption of the one-time pads of the homomorphic encryption, and keep the quantum parts.

(3) Next, the prover and the verifier run a number of self-testing rounds (Protocol 5.3), in which each test round consists of testing either BB84 states (Protocol 5.1) or coset states (Protocol 5.2), forming several test blocks. (In particular, a test block consists of a number of BB84 states testing rounds, and one coset states testing round.) All the BB84 states are consumed after this step, while only half of the coset states are consumed.

(4) Once the verifier is convinced, the verifier runs the coset states generation round on the remaining half of the coset states, in which the verifier sends back to the prover obfuscation of the membership checking programs. The final state of the prover can then be used in coset states based constructions. To be more precise, the output state of a single run of our protocol would satisfies the monogamy of entanglement property that we described above. If a quantum copy-protection scheme requires n random coset states, we can simply run our protocol n times (with independent randomness for each instance).

5.2.2 Soundness Proof

We now give a brief intuition for the soundness of the protocol.

Rigidity argument for the [GMP22] self-testing protocol. Since the soundness proof uses the rigidity argument of the [GMP22] protocol as the backbone, we first give a short sketch of it. Recall that the [GMP22] protocol is a n-fold parallel of [GV19]. The main technical challenge and the bulk of [GMP22] work is to establish that the prover must treat all the parallel copies of the protocol independently, that is, to show that its (a priori uncharacterized) Hilbert space can be partitioned into n identical subspaces, one for each copy of the protocol.

Consider the last round of the [GMP22] self-testing protocol: at the start, the prover has a state $\sigma^{(\theta,\vec{v})}$, which it produced as a result of the previous rounds of the protocol. Upon receiving $\theta \in \{0,1\}$ the prover measures a binary observable Z_i (if $\theta = 0$) or X_i (if $\theta = 1$) and returns the outcome v'_i, one for each copy. Let $Z(\vec{a}) \coloneqq Z_1^{a_1} \cdots Z_n^{a_n}$, similarly for $X(\vec{b})$. The main goal of the [GMP22] soundness proof is to show that when acting

on the prover's (unknown) state $\sigma^{(\theta)}$ (where $\sigma^{(\theta)}$ is like $\sigma^{(\theta,\vec{v})}$, but averaged over all $\vec{v}$), the operators $\{Z(\vec{a})X(\vec{b})\}$ behave essentially like Pauli operators. Formally, this means showing that on average over $\vec{a},\vec{b}\in\{0,1\}^n$,

$$\mathrm{Tr}\left[Z(\vec{a})X(\vec{b})Z(\vec{a})X(\vec{b})\sigma^{(\theta)}\right]\approx(-1)^{\vec{a}\cdot\vec{b}}. \tag{5.1}$$

This is done through the following steps[12]:

(1) Defining inefficient observables $\tilde{X}_i=(-1)^{v_i}X_i$, where v_i is the i-th bit of the verifier's string $\vec{v}$ (Definition 5.8). This observable depends on v_i, which requires the trapdoor information to be computed efficiently. Intuitively, while X_i describes the prover's answer, $\tilde{X}_i$ describes whether that answer is accepted by the verifier. Later in the proof, we will show Equation (5.1) with $\tilde{X}$ instead of X. We note that the Z observable can be efficiently implemented without the trapdoor, and the verifier can also use Z to verify the answer of the prover.

(2) Extending the family of states $\{\sigma^{(\theta)}\}_{\theta\in\{0,1\}}$ to a larger family of "counterfactual states" $\{\sigma^{(\vec{\theta})}\}_{\vec{\theta}\in\{0,1\}^n}$, which are defined as the states the prover would have prepared if the verifier had sent keys $k_i\in\mathcal{K}_{\theta_i}$ for different θ_i. The key point here is that the states $\{\sigma^{(\vec{\theta})}\}_{\vec{\theta}}$ are computationally indistinguishable by the properties of ENTCF families.

(3) Showing various commutation and anti-commutation relations for the observables $Z(\vec{a})$ and $\tilde{X}(\vec{b})$ using the counterfactual states $\sigma^{(\vec{\theta})}$. For example, to show that Z_i and $\tilde{X}_j$ commute, we would choose a $\vec{\theta}$ with $\theta_i=0$ and $\theta_j=1$ since the verifier can check the outcomes of "Z-type observables" for $\theta=0$ and "X-type observables" for $\theta=1$. Then, we can relate these statements back to the prover's actual states $\sigma^{(\theta)}$ using the computational indistinguishability of $\{\sigma^{(\vec{\theta})}\}$.

(4) Combining the commutation and anti-commutation statements from the previous step to show that the observables $\{Z(\vec{a})\tilde{X}(\vec{b})\}$ behave like Pauli observables on $\sigma^{(\theta=1)}$ (Lemma 5.4).

(5) Explicitly defining an isometry $\tilde{V}$ which can be shown to map $\{Z(\vec{a})\tilde{X}(\vec{b})\}$ to the corresponding Pauli observables (Definition 5.10). Furthermore, by using this $\tilde{V}$, we can also define a modified isometry V that maps the efficient observables $\{Z(\vec{a})X(\vec{b})\}$ to the corresponding Pauli observables (Lemma 5.5). This proves Equation (5.1) and finishes the proof.

Rigidity argument for our coset-state self-testing protocol. Using the [GMP22] rigidity argument, we now turn into our coset-state self-testing protocol. Crucially, since the two protocols are identical from the prover's point of view, and the fact that the input of the prover is encrypted, Equation (5.1) also carries to the coset-state self-testing. Specifically, it means that under the isometry V, the prover's observables in the coset-state self-testing protocol also behave like Pauli observables (Lemma 5.8). Roughly speaking, the isometry "teleport" the prover's state into a "concrete" state by means of EPR pairs. In our case, the concrete state would be (close to) a mixed state of a vector $v\in A+x$ if $\theta=0$, or a vector $v'\in A^{\perp}+z$ if $\theta=1$ (up to some classical post-processing), for a coset state instance (A,x,z) (Lemma 5.10).

[12]This sketch is described in [GMP22], we briefly recall it here, and refer the reader to that paper for more details.

This means that we can fix a prover $\mathcal{P}$ and consider a "hypothetical" quantum verifier, which run the protocol in superposition with $\mathcal{P}$, that is, we do not measure to get the prover's classical message as in the original protocol, but only do a projective measurement at the end for the verification. Then under the isometry V, if $\theta = 0$, we should obtain a state that is close to $|A + x\rangle$, and if $\theta = 1$, a state that is close to $|A^{\perp} + z\rangle$. In other words, consider that we run $\mathcal{P}$ with $\theta = 0$ in superposition, check the obtained state is $|A + z\rangle$, then undo the prover computation (described by a unitary), then run $\mathcal{P}$ with $\theta = 1$ in superposition, check the obtained state is $|A^{\perp} + x\rangle$. If both checks passed, it is easy to see that the prover must have a coset state $|A_{z,x}\rangle$ in its registers.

Note that this does not constitute a classical verification of QFHE. What it says is that after the evaluation and if $\mathcal{P}$ passes verification with overwhelming probability it is necessary that it must have a coset state in its register up to an isometry.

We stress that the above rigidity statement has $1/\text{poly}(n)$ closeness, due to the $1/\text{poly}(n)$ closeness in the rigidity argument of the [GMP22] protocol.

Going from self-testing to remote state preparation. We then simply run the self-testing protocol sequentially in the cut-and-choose style. Say we have $2N$ coset state instances, and we run the self-testing protocol over N instances, chosen uniformly at random. The remaining N instances are the output of the protocol. By a particular "quantum sample-and-estimate" strategy defined in [BF10], it means that after running the self-testing rounds, the prover has at least one coset state $|A_{x,z}\rangle$ among N remaining coset state instances in its registers, with inverse polynomial closeness. We can write the prover's state at this step as (inverse polynomially δ-close to) $|A_{x,z}\rangle \otimes \rho$, where ρ can depend on the protocol's transcript and the encryption of (A, x, z) (Proposition 5.2).

Establishing a monogamy of entanglement property. In this final step, we want to show that now if the prover involves in a monogamy of entanglement game, it would have negligible probability of winning. The security game is defined as follows (Definition 5.12).

1. The prover and the verifier jointly execute our semi-quantum protocol to obtain (supposedly) N coset states, which are hidden but kept by the prover.
2. The prover and the verifier play the monogamy game using the output of the semi-quantum protocol:
   (a) The prover splits its state into a bipartite state and sends each part to Bob and Charlie, respectively. No communication is allowed between Bob and Charlie.
   (b) The verifier sends the description of the *subspace* to both Bob and Charlie.
   (c) Bob and Charlie are asked to output N vectors belonging to N cosets (for Bob), and N dual cosets (for Charlie).

However, our current situation is different from the standard monogamy game setting in which the prover only has the coset state, while here the prover also have an auxiliary state that depends on the coset state description. (Even worse, it might be possible that the prover can have two copies of the coset state after the interactive protocol.) The proof of the standard monogamy game does not carry over directly. Hence, for our monogamy of entanglement proof, new ideas are needed.

Injecting quantumness into the reduction. Our idea is to consider an intermediate game as follows.

1. The prover and the verifier jointly execute our semi-quantum protocol.
2. After finishing the protocol execution, the verifier asks the prover to send it a coset state among the remaining coset state instances uniformly at random.
3. Upon receiving a quantum state from the prover, the verifier verifies whether the received state is indeed the expected coset state, then it sends it back unmodified to the prover.
4. The prover and the verifier play the monogamy game.

Here we make few notes. First, with probability $\frac{1}{N}$ the coset state instance that the verifier asked is (A, x, z). It is easy to see that with probability $\frac{(1-\delta)}{N}$, which is non-negligible, any adversary for the original security experiment can be turned into an adversary for this experiment with identical winning probability. Secondly, defining this intermediate game is possible because of our rigidity argument above. Indeed, only in this step we inject quantumness into the reduction and make it a quantum verifier.

The proof continues with the following steps (which are formally described as a series of hybrids in the proof Theorem 5.2).

- We make another important observation is that now when considering only the coset instance (A, x, z), it is exactly the same as the public-key semi-quantum protocol introduced by Shmueli in [Shm22a]. We then follow proof strategies in previous works and carefully modify the experiment to remove the QFHE secret key (corresponding to this coset instance (A, x, z)) from the reduction. This is essentially done by changing the obfuscated membership checking programs sent to the prover in the last step of the protocol, using the following two techniques: subspace-hiding obfuscation [Zha19b], and complexity leveraging to blindly sample the obfuscations [Shm22a]. To use Shmueli's complexity leveraging technique, we will need sub-exponential security of the building blocks (which include the QFHE and the indistinguishability obfuscation).
- Then we make a final change in the reduction: upon receiving the coset state from the prover and if the check is passed, the verifier keeps the received coset state in its internal memory, and send back to the prover another random coset state $|A'_{x',z'}\rangle$. In the monogamy game, instead of sending a description of A (as a basis matrix), the verifier sends a description of A'. Note that now the winning condition is also changed subject to this change in the challenge coset. We can think of $|A'_{x',z'}\rangle$ as the challenge of the original monogamy of entanglement game (with quantum communication).
- In this final experiment, if the prover managed to win the monogamy game, it means that Bob has successfully output a vector $v \in A' + x'$, and Charlie has successfully output a vector $w \in A'^{\perp} + z'$. The verifier then outputs v, w and wins the monogamy game with quantum communication. We conclude that no efficient prover can win this experiment except with negligible probability.

- The last part of the proof is to show that this final experiment is computationally indistinguishable from the previous experiment (in which the QFHE secret key was removed). We do this by invoking the security of the QFHE. However, there is a subtlety that needs to be taken care of. That is, even if we do not use the QFHE secret key in the reduction at this step, the adversary still receives predicate programs on the ciphertext, which are the obfuscated membership checking programs. Thus, we cannot simply send a uniformly random coset state $|A'_{x',z'}\rangle$ to the prover. In the protocol, we change the obfuscation programs so that both $|A_{x,z}\rangle$ and $|A'_{x',z'}\rangle$ make the programs accept. We refer to the formal construction and proof for the description of how these obfuscation programs are generated. Once this is shown, we can complete the proof.

5.3 Coset States

For any subspace $A \subseteq \mathbb{F}_2^n$, its complement is $A^\perp := \{b \in \mathbb{F}_2^n \mid \langle a, b\rangle = 0, \forall a \in A\}$. We have that $\dim(A) + \dim(A^\perp) = n$. We also let $|A| := 2^{\dim(A)}$ denote the size of the subspace A.

Definition 5.1 — Subspace States

For any subspace $A \subseteq \mathbb{F}_2^n$, the subspace state $|A\rangle$ is defined as

$$|A\rangle := \frac{1}{\sqrt{|A|}} \sum_{a \in A} |a\rangle .$$

Note that given A, the subspace state $|A\rangle$ can be constructed efficiently.

Definition 5.2 — Coset States

For any subspace $A \subseteq \mathbb{F}_2^n$, vectors $s, s' \in \mathbb{F}_2^n$, the coset state $|A_{s,s'}\rangle$ is defined as

$$|A_{s,s'}\rangle := \frac{1}{\sqrt{|A|}} \sum_{a \in A} (-1)^{\langle a, s'\rangle} |a + s\rangle .$$

Note that given $|A\rangle$ and s, s', the coset state $|A_{s,s'}\rangle$ can be constructed efficiently.

Furthermore, for a subspace A and vectors s, s', we define $A + s := \{v + s \mid v \in A\}$, and $A^\perp + s' := \{w + s' \mid w \in A^\perp\}$.

When it is clear from the context, for ease of notation, we will write $A + s$ to mean the *program* that checks membership in $A + s$. For example, we will often write $iO(A + s)$ to mean an indistinguishability obfuscation of the program that checks membership in $A + s$.

5.3.1 Strong Monogamy-of-Entanglement Property

Coset states satisfy the following strong monogamy-of-entanglement property, which will be used as the main tool in our construction for copy-protection.

Definition 5.3 — Coset-Monogamy Game [CLLZ21; CV22]

The coset monogamy game between a challenger and a QPT adversary $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ is defined as follows.

(1) Preparation. The challenger picks a uniformly random subspace $A \subseteq \mathbb{F}_2^\lambda$ of dimension $\frac{\lambda}{2}$, and two uniformly random vectors $s, s' \in \mathbb{F}_2^n$. It sends $|A, s, s'\rangle, i\mathcal{O}(A+s), i\mathcal{O}(A^\perp + s')$ to the adversary $\mathcal{A}_0$.

(2) Splitting. The adversary applies a quantum channel: $\Phi : \mathcal{H}_A \to \mathcal{H}_B \otimes \mathcal{H}_C$ where $\mathcal{H}_A = (\mathbb{C}^2)^{\otimes\lambda}$ and $\mathcal{H}_B, \mathcal{H}_C$ are arbitrary. It then computes $\rho_{BC} := \Phi(|A_{s,s'}\rangle\langle A_{s,s'}| \otimes |i\mathcal{O}(A+s), i\mathcal{O}(A^\perp+s')\rangle\langle i\mathcal{O}(A+s), i\mathcal{O}(A^\perp+s')|)$. It sends registers B to $\mathcal{A}_1$ and C to $\mathcal{A}_2$, respectively.

(3) Question. The challenger sends the description of A, in the form of a basis for it, to both $\mathcal{A}_1$ and $\mathcal{A}_2$.

(4) Answer. $\mathcal{A}_1$ returns $s_1 \in \mathbb{F}_2^n$ and $\mathcal{A}_2$ returns $s_2 \in \mathbb{F}_2^n$.

The adversary $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ wins if and only if $s_1 \in A + s$ and $s_2 \in A^\perp + s'$. Let $\mathsf{CosetMonogamy}((\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2), \lambda)$ be a random variable which takes the value 1 if the game above is won by adversary $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$, and takes the value 0 otherwise.

Theorem 5.1 ([CLLZ21, Theorem 4.18]). *Assuming the existence of post-quantum indistinguishability obfuscation and one-way functions, then there exists a negligible function* $\mathsf{negl}(\cdot)$*, for any* QPT *adversary* $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$*,*

$$\Pr\left[\,\mathsf{CosetMonogamy}((\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2), \lambda)\,\right] \leq \mathsf{negl}(\lambda).$$

5.4 Semi-Quantum Copy-Protection

In this section, we introduce our protocol for semi-quantum copy-protection from hidden coset states in Section 5.4.1, followed by proof of correctness in Section 5.4.2 and proof of soundness in Section 5.5.

5.4.1 Construction

Notation. Our Protocol 5.1 and Protocol 5.2 will be (almost) a parallel repetition of a sub-protocol. We make use of vector notation to denote tuples of items corresponding to the different copies of the sub-protocol. For example, if each of the n parallel sub-protocols requires a key k_i, we denote $\vec{k} = (k_1, \ldots, k_n)$. A function that takes as input a single value can be extended to input vectors in the obvious way: for example, if f takes as input a single key k, then we write $f(\vec{k})$ for the vector $(f(k_1), \ldots, f(k_n))$. We will also use $\vec{0}$ and $\vec{1}$ for the bit strings consisting only of 0 and 1, respectively (and whose length will be clear from the context), and $\vec{1}^i \in \{0,1\}^n$ for the bit string whose i-th bit is 1 and whose remaining bits are 0. Let n the length of a vector in a coset state (i.e., if $v \in A$ then $|v| = n$). In our constructions below, we set $n := 2\lambda$.

Ingredients. Our constructions use the following building blocks:

- A quantum hybrid fully homomorphic encryption scheme $\mathsf{QFHE} := \langle \mathsf{KeyGen}, \mathsf{QOTP}, \mathsf{Encrypt}, \mathsf{Eval}, \mathsf{Decrypt} \rangle$, with sub-exponential advantage security.
- A post-quantum secure indistinguishability obfuscation scheme $i\mathcal{O}$.
- A post-quantum secure extended noisy trapdoor claw-free function (ENTCF) family $(\mathcal{F}, \mathcal{G})$.

Our main protocol's construction is given in Protocol 5.5. The protocol involves two parties: a QPT prover (or receiver, denoted as $\mathcal{P}$), and a PPT verifier (or sender, denoted as $\mathcal{V}$).

Protocol 5.1: Semi-Quantum Copy Protection: BB84 Test Round

Input. The verifier initially receives Pauli keys (α, β) with $\alpha, \beta \in \{0,1\}^n$ as private inputs.

1. The verifier selects a uniformly random basis $\theta \xleftarrow{\$} \{0,1\}$, where 0 corresponds to the computational and 1 to the Hadamard basis.
2. The verifier samples keys and trapdoors $\{(k_i, t_i)\}_{i=1}^n$ by computing $(k_i, t_i) \leftarrow \mathsf{Gen}_{\mathcal{K}_\theta}(1^\lambda)$. The verifier then sends $\{k_i\}_{i=1}^n$ to the prover (but keeps the trapdoors $\{t_i\}_{i=1}^n$ private).
3. The verifier receives $\{y_i\}_{i=1}^n$ where $y_i \in \mathcal{Y}$ from the prover.
4. The verifier selects a round type $\in \{\text{pre-image round}, \text{Hadamard round}\}$ uniformly at random and sends the round type to the prover.
   (a) For a *pre-image round*: the verifier receives $\{(b_i, x_i)\}_{i=1}^n$ from the prover, with $b_i \in \{0,1\}$, and $x_i \in \mathcal{X}$. The verifier sets $\mathtt{flag}_{\mathtt{bb84}} \leftarrow \mathtt{flag}_{\mathtt{Pre}}$ and aborts if $\mathsf{Chk}(k_i, t_i, b_i, x_i) = 0$ for any $i \in [\![1, n]\!]$.
   (b) For a *Hadamard round*: the verifier receives $\{d_i\}_{i=1}^n$ from the prover with $d_i \in \{0,1\}^w$ (for some w depends on the security parameter λ). The verifier sends $q = \theta$ to the prover, and receives answers $\{v_i\}_{i=1}^n$ with $v_i \in \{0,1\}$. The verifier performs the following:
      - If $q = \theta = 0$, set $\mathtt{flag}_{\mathtt{bb84}} \leftarrow \mathtt{flag}_{\mathtt{Had}}$ and abort if $\hat{b}(k_i, y_i) \neq v_i$ for some $i \in [\![1, n]\!]$.
      - If $q = \theta = 1$, set $\mathtt{flag}_{\mathtt{bb84}} \leftarrow \mathtt{flag}_{\mathtt{Had}}$ and abort if $\hat{u}(k_i, y_i, d_i) \neq v_i \oplus \beta_i$ for some $i \in [\![1, n]\!]$.

Protocol 5.2: Semi-Quantum Copy Protection: Coset-state Test Round

Input. The verifier initially receives a subspace $A \subseteq \mathbb{F}_2^n$ and Pauli keys (α, β) with $\alpha, \beta \in \{0,1\}^n$ as private inputs.

1. The verifier selects a uniformly random basis $\theta \xleftarrow{\$} \{0,1\}$, where 0 corresponds to the computational and 1 to the Hadamard basis.

2. The verifier samples keys and trapdoors $\{(k_i, t_i)\}_{i=1}^n$ by computing $(k_i, t_i) \leftarrow \mathsf{Gen}_{\mathcal{K}_\theta}(1^\lambda)$. The verifier then sends $\{k_i\}_{i=1}^n$ to the prover (but keeps the trapdoors $\{t_i\}_{i=1}^n$ private).

3. The verifier receives $\{y_i\}_{i=1}^n$ where $y_i \in \mathcal{Y}$ from the prover.

4. The verifier sends "Hadamard round" as the round type to the prover.

5. The verifier receives $\{d_i\}_{i=1}^n$ from the prover with $d_i \in \{0,1\}^w$ (for some w depends on the security parameter λ). The verifier sends $q = \theta$ to the prover, and receives answers $\{v_i\}_{i=1}^n$ with $v_i \in \{0,1\}$.

   The verifier performs the following:

   - If $q = \theta = 0$, let $\vec{v} := v_1 \dots v_n$. Set $\mathtt{flag}_{\mathtt{coset}} \leftarrow \mathtt{flag}_{\mathtt{Had}}$ and abort if $\vec{v} \notin A + \alpha$.
   - If $q = \theta = 1$, let $s_i \leftarrow v_i \oplus \hat{u}(k_i, y_i, d_i)$ and let $s := s_1 \dots s_n$. Set $\mathtt{flag}_{\mathtt{coset}} \leftarrow \mathtt{flag}_{\mathtt{Had}}$ and abort if $\vec{s} \notin A^\perp + \beta$.

Protocol 5.3: Semi-Quantum Copy Protection: Self-Testing

Let M^2 the maximum number of test rounds (for $M \in \mathbb{N}$).

Input. The verifier initially receives a subspace $A \subseteq \mathbb{F}_2^n$ and Pauli keys (α', β') and $\{(\alpha_i, \beta_i)\}_{i=1}^{M^2}$ with $\alpha', \beta', \alpha_i, \beta_i \in \{0,1\}^n$ as private inputs. Note that (A, α', β') corresponds to one coset-state instance, and $\{(\alpha_i, \beta_i)\}_{i=1}^{M^2}$ corresponds to M^2 BB84 instances.

1. The verifier privately samples $B \xleftarrow{\$} [\![1, M-1]\!]$ (this determines the number of BB84 test rounds that will be performed).

2. The verifier performs BM executions of Protocol 5.1 (with corresponding private inputs $\{(\alpha_i, \beta_i)\}$) with the prover. The verifier aborts if Protocol 5.1 aborts for some execution.

3. The verifier privately samples $R \xleftarrow{\$} [\![1, M]\!]$ and executes Protocol 5.1 with the prover $R-1$ times (with corresponding private inputs $\{(\alpha_i, \beta_i)\}$). Then the verifier executes Protocol 5.2 with the prover (with private inputs (A, α', β')) and aborts if Protocol 5.2 aborts.

Protocol 5.4: Semi-Quantum Copy Protection: Self-Testing (with Soundness Amplification)

Let $N := \lambda$ the number of iterations.

Input. The verifier initially receives $\{(A_i, \alpha_i', \beta_i')\}_{i=1}^N$ and $\{(\alpha_i, \beta_i)\}_{i=1}^{NM^2}$ as private inputs. Each tuple in the first set corresponds to a coset-state instance, and each tuple in the second set corresponds to a BB84 instance.

The verifier and the prover sequentially run Protocol 5.3 N times as follows.

1. For each run, the verifier and the prover interactively run Protocol 5.3 with one coset state instance $(A_i, \alpha_i', \beta_i')$ and M^2 BB84 instances $\{(\alpha_i, \beta_i)\}_{i=1}^{M^2}$, each

is picked uniformly at random from the input sets. (If some instance has been picked before, it will be excluded).

2. The verifier aborts unless Protocol 5.3 does not abort in all N iterations.

Protocol 5.5: Semi-Quantum Copy Protection: Main Protocol

Verifier's preparation.

1. **Coset-state instances.** For each $i \in [\![1, 2N]\!]$, the verifier samples a random $\frac{n}{2}$-dimensional subspace $S_i \subseteq \mathbb{F}_2^n$, described by a matrix $\mathbf{M}_{S_i} \in \{0,1\}^{\frac{n}{2}\times n}$. Samples Pauli keys $p_{\alpha_i} \xleftarrow{\$} \{0,1\}^{\frac{n^2}{2}}$ to encrypt $\mathbf{M}_{S_i}^{p_{\alpha_i}} \leftarrow \mathsf{QFHE.QOTP}(p_{\alpha_i}, \mathbf{M}_{S_i})$, and then $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{QFHE.KeyGen}(1^\lambda, 1^{\ell(\lambda)})$ for some polynomial $\ell(\cdot)$, $\mathsf{ct}_i \leftarrow \mathsf{QFHE.Encrypt}(\mathsf{pk}_i, p_{\alpha_i})$.

2. **n-qubit BB84 instances.** For each $i \in [\![1, NM^2]\!]$, the verifier samples Pauli keys $p_{\alpha_i} \xleftarrow{\$} \{0,1\}^{\frac{n^2}{2}}$ to encrypt $\mathbf{M}_0^{p_{\alpha_i}} \leftarrow \mathsf{QFHE.QOTP}(p_{\alpha_i}, \mathbf{M}_0)$ (here, $\mathbf{M}_0$ is the all-zero vector of length $\frac{n^2}{2}$), and then $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{QFHE.KeyGen}(1^\lambda, 1^{\ell(\lambda)})$, $\mathsf{ct}_i \leftarrow \mathsf{QFHE.Encrypt}(\mathsf{pk}_i, p_{\alpha_i})$.

3. For each index $i \in [\![1, 2N + NM^2]\!]$, the verifier picks uniformly at random one instance from either the set of (encrypted) coset states or the set of (encrypted) n-qubit BB84 states prepared above. For each index i, denote the i-th instance as $(\mathsf{pk}_i, \mathbf{M}^{p_{\alpha_i}}, \mathsf{ct}_i)$ with secrets (sk_i, S_i). (If this instance is from the set of n-qubit BB84 states, we understand that $S_i = \mathbf{M}_0$.)

4. The verifier sends $\{\mathsf{pk}_i, \mathbf{M}^{p_{\alpha_i}}, \mathsf{ct}_i\}_{i=1}^{2N+NM^2}$ to the prover.

Prover's homomorphic evaluation.

5. Let C the quantum circuit that for an input matrix $\mathbf{M} \in \{0,1\}^{\frac{n}{2}\times n}$, outputs a uniform superposition of its row span, except that if $\mathbf{M} = \mathbf{M}_0$, it outputs a uniform superposition of all vectors in the space $\mathbb{F}_2^n$. The prover homomorphically evaluates C for each $i \in [\![1, 2N + NM^2]\!]$: $(|S_{i,\alpha_i,\beta_i}\rangle, \mathsf{ct}_{i,\alpha_i,\beta_i}) \leftarrow \mathsf{QFHE.Eval}\big(\mathsf{pk}_i, (\mathbf{M}^{p_{\alpha_i}}, \mathsf{ct}_i), C\big)$, saves the quantum part $|S_{i,\alpha_i,\beta_i}\rangle$ and sends the classical part $\mathsf{ct}_{i,\alpha_i,\beta_i}$ to the verifier.

Self-testing for the prover.

6. For each $i \in [\![1, 2N + NM^2]\!]$, the verifier decrypts $(\alpha_i, \beta_i) \leftarrow \mathsf{QFHE.Decrypt}(\mathsf{sk}_i, \mathsf{ct}_{i,\alpha_i,\beta_i})$. For all coset-state instances, if $\alpha_i \in S_i$, the protocol is terminated.

7. The verifier then runs Protocol 5.4 with these NM^2 prepared BB84 instances and N coset-state instances, where each coset-state instance is picked uniformly at random among $2N$ prepared instances. (If some instance has been picked before, it will be excluded). It aborts if Protocol 5.4 aborts.

Coset-state generation.

8. The verifier samples a random $\frac{n}{2}$-dimensional coset $(\hat{S}, \hat{\alpha}, \hat{\beta}) \subseteq \mathbb{F}_2^n$ independently.[a] Let $\mathbf{M}_{\hat{S}}, \mathbf{M}_{\hat{S}^\perp} \in \{0,1\}^{\frac{n}{2} \times n}$ bases for $\hat{S}$ and $\hat{S}^\perp$, respectively.

9. Let T the set of indexes of the remaining N instances of the coset-states which have not been used in the self-testing protocol above. For each $i \in T$, the verifier does the following:

   (a) Let $\mathbf{M}_{S_i^\perp} \in \{0,1\}^{\frac{n}{2} \times n}$ a basis for $S_i^\perp$ (as a matrix). Compute indistinguishability obfuscations $P_{0,i} \leftarrow i\mathcal{O}\left(i\mathcal{O}(\mathbf{M}_{S_i} + \alpha_i) \vee i\mathcal{O}(\mathbf{M}_{\hat{S}} + \hat{\alpha})\right)$ and $P_{1,i} \leftarrow i\mathcal{O}\left(i\mathcal{O}(\mathbf{M}_{S_i^\perp} + \beta_i) \vee i\mathcal{O}(\mathbf{M}_{\hat{S}^\perp} + \hat{\beta})\right)$, all with appropriate padding.[b]

   (b) Record $\{(\alpha_i, \beta_i, S_i)\}_{i \in T}$.

   (c) Send T and $\{P_{0,i}, P_{1,i}\}_{i \in T}$ to the prover.

The output of the prover is $\{P_{0,i}, P_{1,i}, |S_{i,\alpha_i,\beta_i}\rangle\}_{i \in T}$ where $|T| = N$.

[a] This step is merely an artifact that we will need later for the security proof.

[b] Here, we understand that for any two programs C, C' with binary output, $i\mathcal{O}(C \vee C')(x)$ outputs $C(x) \vee C'(x)$.

Notation. For each execution of Protocol 5.5, we abuse the notation and denote $(|A_{s,s'}\rangle, \mathsf{R}^0, \mathsf{R}^1)$ the state obtained by the receiver, where R^b the obfuscated membership checking programs, computed by concatenating all the obfuscated programs $P_{b,i}$ in Protocol 5.5, and (A, s, s') the "coset" (which in fact consists of polynomial many different real cosets) obtained by the sender. That is, we consider the whole output state of the protocol as a single unclonable state (which we also call "coset state"). This notation will only be used later when we describe the applications of our protocol in the context of semi-quantum copy-protection.[13]

5.4.2 Proof of Completeness

Proposition 5.1. *There exists a* QPT *prover that is accepted in Protocol 5.5 with probability negligibly close to* 1 *in the security parameter* λ*. Furthermore, the final* quantum *state of such a prover at the end of Protocol 5.5 is (negligibly close to) a product of* N *hidden coset states:*

$$\bigotimes_{i \in T} |S_{i,\alpha_i,\beta_i}\rangle\,, \tag{5.2}$$

where $\{(S_i, \alpha_i, \beta_i)\}_{i \in T}$ *are recorded by the verifier at the end of Protocol 5.5.*

Proof. The proof of correctness includes three steps: (1) If the prover ran honestly then its output after the homomorphic evaluation step has negligible trace distance to (QOTP encrypted of) BB84 states and coset states; (2) The self-test protocol passes (that is,

[13] We thank an anonymous reviewer for pointing out this approach, so that our protocol is indeed applicable to copy-protection in a plug-and-play manner.

the protocol does not terminate at this step) with probability negligibly close to 1; (3) In the last step of coset-state generation, after discarding all BB84 states, the output of the prover at the end of Protocol 5.5 has negligible trace distance to the state described in Equation (5.2).

We describe the honest strategy. By the statistical correctness of the homomorphic encryption, at the end of step 5 of Protocol 5.5, the i-th quantum state that an honest prover holds in its quantum-evaluated registers has negligible trace distance to either $\bigotimes_{j=1}^{n} |(-1)^{\beta_{i,j}}\rangle$ (if the corresponding instance is a n-qubit BB84 state) or $|S_{i,\alpha_i,\beta_i}\rangle$ (if the corresponding instance is a coset state). That is, this negligible distance holds with probability 1 over the previous messages of the protocol.

For each coset-state instance i, we claim that the probability for such honest prover to have $\alpha_i \in S_i$ is negligible. It follows from the fact that if $\alpha_i \in S_i$, we have that $|S_{i,\alpha_i,\beta_i}\rangle = |S_{i,0,\beta_i}\rangle$. By just measuring this state in the computational basis, we get a non-zero vector $s \in S_i$ with overwhelming probability, even without knowing S_i or the QFHE secret key. This violates the semantic security of the QFHE, because S_i is a subspace of dimension $\frac{n}{2}$ chosen uniformly at random, for any vector $s \in \mathbb{F}_2^n$, the probability that $s \in S_i$ is negligible. It means that Protocol 5.5 terminates at step 6 with negligible probability.

Next, we show that an honest prover succeeds in the self-test rounds of Protocol 5.5 with probability negligibly close to 1. An honest prover behaves the same way in each execution of Protocol 5.1 and Protocol 5.2. Hence, to show that an honest prover succeeds in Protocol 5.3 with probability negligibly close to 1, it suffices to describe honest strategies for Protocol 5.1 and Protocol 5.2 that succeed with probability negligibly close to 1. We note that Protocol 5.4 is N sequential repetition of Protocol 5.3, and thus the completeness of Protocol 5.4 is also negligibly close to 1.

Claim 5.1. *There exists a* QPT *prover that is accepted in Protocol 5.1 with probability negligibly close to* 1.

Proof. In Protocol 5.1, the prover receives n keys $k_1, \ldots, k_n$ and returns answers for each key k_j individually. Since the verifier's checks are independent for each j, we only need to describe an honest procedure for one key k_j that succeeds in the verifier's checks for that j with probability negligibly close to 1. The honest strategy for a single key k_j is adapted from the one in [GV19; MV21; GMP22]. We spell out the details below.

From now on, for simplicity, we drop the subscript i and understand that we are considering the i-th instance. First, note that at the beginning of Protocol 5.1, for a given key $k_j \in \mathcal{K}_\theta$, the prover is having the state $|(-)^{\beta_j}\rangle = \mathsf{Z}^{\beta_j} |(-)^0\rangle$ in his quantum registers. The prover then adjoins a uniform superposition over all $x \in \mathcal{X}$, evaluate f_{k_j} in superposition to obtain the following state:

$$\frac{1}{\sqrt{2|\mathcal{X}|}} \sum_{b\in\{0,1\}} \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} \sqrt{f_{k_j,b}(x)(y)} \mathsf{Z}^{\beta_j} |b\rangle |x\rangle |y\rangle$$

Preparing this state can be efficiently done (up to negligible error) using the Samp procedure from the definition of ENTCF families ([BCMV+18, Definition 3.1] and [Mah18b, Definition 4.2]). The prover then measures the "image register" (i.e., the register that stores y) to obtain an image $y_j \in \mathcal{Y}$ and sends this back to the verifier. The

post-measurement state for each j is

$$\begin{cases} |\hat{b}(k_j, y_j)\rangle\, |\hat{x}(k_j, y_j)\rangle & \text{if } k_j \in \mathcal{K}_0, \\ \frac{1}{\sqrt{2}} \left(|0\rangle\, |\hat{x}_0(k_j, y_j)\rangle + (-1)^{\beta_j} |1\rangle\, |\hat{x}_1(k_j, y_j)\rangle \right) & \text{if } k_j \in \mathcal{K}_1. \end{cases} \tag{5.3}$$

If the verifier selects a "pre-image round", the prover measures both registers in the computational basis and returns the result. From the states in Equation (5.3) it is clear that the prover succeeds with probability negligibly close to 1 in the pre-image round.

If the verifier selects a "Hadamard round", the prover measures the "x-register" in the Hadamard basis to obtain d_j and returns this to the verifier. We introduce the shorthand $b_j := \hat{b}(k_j, y_j)$ and $u_j := \hat{u}(k_j, y_j, d_j)$. At this point, the prover's state for each j is (up to a global phase):

$$\begin{cases} |b_j\rangle & \text{if } k_j \in \mathcal{K}_0, \\ |(-)^{u_j \oplus \beta_j}\rangle & \text{if } k_j \in \mathcal{K}_1. \end{cases} \tag{5.4}$$

The prover now receives a question $q = \theta$ and measures the remaining qubit in the computational basis if $q = 0$ and in the Hadamard basis if $q = 1$. Then it is clear from the expression for the prover's remaining qubit in Equation (5.4) that the prover will pass the verifier's check. □

Claim 5.2. *There exists a* QPT *prover that is accepted in Protocol 5.2 with probability negligibly close to* 1.

Proof. At the beginning of each instance of Protocol 5.2, the prover is having the state $|A_{\alpha,\beta}\rangle$ with $\alpha, \beta \in \{0,1\}^n$. The honest strategy for the prover in Protocol 5.2 is similar to the honest strategy for Protocol 5.1 described in Claim 5.1: the prover uses the ENTCF family to commits to each qubit of the state $|A_{\alpha,\beta}\rangle$ using the corresponding function key. A formal description of the commitment process is given in [Mah18b, Section 5.1]. In the last round, if $q = \theta = 0$, the prover measures each qubit in the computational basis and in the Hadamard basis if $q = \theta = 1$. It equivalents to either measure the state $|A_{\alpha,\beta}\rangle$ in the computational basis if $q = 0$ and in the Hadamard basis if $q = 1$.

Since the prover applies the same strategy for each qubit in the state, here we describe the state commitment process for the j-th qubit of the state $|A_{\alpha,\beta}\rangle$. For a given key $k_j \in \mathcal{K}_\theta$, we can write the prover's coset state as

$$\sum_{b_j \in \{0,1\}} \gamma_{b_j} |b_j\rangle\, |\psi_{b_j}\rangle$$

The prover then adjoins a uniform superposition over all $x \in \mathcal{X}$, evaluate f_{k_j} in superposition to obtain

$$\frac{1}{\sqrt{|\mathcal{X}|}} \sum_{b_j \in \{0,1\}} \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \gamma_{b_j} \sqrt{f_{k_j, b_j}(x)(y)}\, |b_j\rangle\, |\psi_{b_j}\rangle\, |x\rangle\, |y\rangle \tag{5.5}$$

The prover then measures the "y-register" to obtain an image $y_j \in \mathcal{Y}$ and sends this back to the verifier. The post-measurement state for each j is

$$\begin{cases} |\hat{b}(k_j, y_j)\rangle\, |\psi_{b_j}\rangle\, |\hat{x}(k_j, y_j)\rangle & \text{if } k_j \in \mathcal{K}_0, \\ \sum_{b_j \in \{0,1\}} \gamma_{b_j} |b_j\rangle\, |\psi_{b_j}\rangle\, |\hat{x}_{b_j}(k_j, y_j)\rangle & \text{if } k_j \in \mathcal{K}_1. \end{cases} \tag{5.6}$$

We note that the verifier always sends "Hadamard round" as the round type in Protocol 5.2. The prover measures the "x-register" in the Hadamard basis to obtain d_j and returns this to the verifier. The prover now receives a question $q = \theta$ and measures the j-th qubit in the computational basis if $q = 0$ and in the Hadamard basis if $q = 1$. Recall that we denote $u_j := \hat{u}(k_j, y_j, d_j)$. At this point, the prover's state (before the measurement) is (up to a global phase):

$$\begin{cases} |b_j\rangle\,|\psi_{b_j}\rangle & \text{if } k_j \in \mathcal{K}_0, \\ (\mathsf{X}^{u_j}\mathsf{H} \otimes \mathcal{I})\,|b_j\rangle\,|\psi_{b_j}\rangle & \text{if } k_j \in \mathcal{K}_1. \end{cases} \tag{5.7}$$

The prover measures the j-th qubit and returns a bit v_j to the verifier. It is clear from Equation (5.7) that: (1) if the coset state is measured in the computational basis (corresponding to the case $q = 0$), the verifier obtains a vector $v \in A + \alpha$; or (2) the coset state is measured in the Hadamard basis (corresponding to the case $q = 1$), the verifier obtains a vector $s \in A^{\perp} + \beta$. This concludes the proof of the claim. □

Having described the honest behavior for the self-test step, we finish the proof of completeness. □

5.5 Proof of Soundness

In this section, we prove soundness of Protocol 5.5, following the steps outlined in Section 5.2.2:

1. First, we show a ridigity argument (with inverse polynomial soundness) for our self-testing protocol (Protocol 5.3) in Section 5.5.1.
2. We show that any malicious prover in our remote state preparation protocol must have also constructed a hidden random coset state up to some inverse polynomial error. This final step reduces to a particular "quantum sample-and-estimate strategy", which is a quantum counterpart of the classical "cut-and-choose" as defined by Bouman and Fehr [BF10].
3. We then show the soundness of our final protocol (Protocol 5.5), which is stated as a monogamy of entanglement game, in Section 5.5.2. Notably, even if our rigidity statement achieves only inverse polynomial soundness, we show that our protocol achieves negligible security in this monogamy game.

Informally, a prover that succeeds in Protocol 5.5 has negligible probability of winning a monogamy of entanglement game for coset states, which is formally stated as Theorem 5.2. This means that if we consider the output of our final protocol as a single unclonable state, the situation at the end of Protocol 5.5 is essentially identical to one in which the verifier has sent a hidden coset state to the prover via a quantum channel, whose security is based on the monogamy of entanglement of coset states defined in Definition 5.3.

5.5.1 Self-Testing Protocol Soundness

We denote this section to prove the following proposition, whose proof is given at the end of this section.

Proposition 5.2. *For any $\lambda \in \mathbb{N}$, there exist choices $M = \mathsf{poly}(\lambda)$ and $\delta = 1/\mathsf{poly}(\lambda)$ such that if the verifier executes Protocol 5.5 with an efficient quantum prover whose success probability is lower-bounded by an inverse polynomial, the following holds. We denote by ϕ^T_{SVP} the verifier and prover's joint final state at the end of Protocol 5.5, where T is the set of coset states obtained by the verifier, S is set to $|\bot\rangle\langle\bot|$ by the verifier if the protocol aborts and $|\top\rangle\langle\top|$ otherwise, V is the register in which the verifier records the set T, and P is the prover's registers. Then, denoting the probability of success as $\Pr[\top]$, and writing*

$$\phi^T_{SVP} = \Pr[\top]|\top\rangle\langle\top|_S \otimes \phi^T_{VP|\top} + (1 - \Pr[\top])|\bot\rangle\langle\bot| \otimes \phi^T_{VP|\bot}.$$

Then there exists a coset instance (A, α, β) in T such that the state $\phi^T_{VP|\top}$ conditioned on acceptance satisfies:

$$\phi^T_{VP|\top} \overset{c}{\approx}_{1/\mathsf{poly}(\lambda)} |T\rangle\langle T|_V \otimes |A_{\alpha,\beta}\rangle\langle A_{\alpha,\beta}| \otimes \rho, \tag{5.8}$$

for some auxiliary state ρ.

In order to prove Proposition 5.2, we first show a rigidity argument for our self-testing protocol (Protocol 5.3). The rigidity argument we establish in this section for Protocol 5.3 will be based on the n-fold parallel rigidity proof from [GMP22]. We will make frequent use of some technical lemmas from the proof of that paper.

Devices

We model the actions of a general prover by a "device". This formalizes all possible actions that can be taken by the prover to compute his answers to the verifier in Protocol 5.1 and Protocol 5.2. By Naimark's theorem, up to adding dimensions to the prover's Hilbert space, we can assume without loss of generality that the prover only performs projective measurements (instead of more general POVMs).

Definition 5.4 (Devices [GMP22])**.** *A device $D := (S, \Pi, M, P)$ is specified by the following:*

1. *A set $S = \{\psi^{(\vec{\theta})}\}_{\vec{\theta}\in\{0,1\}^n}$ of states $\psi^{(\vec{\theta})} \in \mathcal{D}(\mathcal{H}_D \otimes \mathcal{H}_Y)$, where $\dim(\mathcal{H}_Y) = |\mathcal{Y}|^n$ and the states are classical on $\mathcal{H}_Y$:*

$$\psi^{(\vec{\theta})} = \sum_{\vec{y}\in\mathcal{Y}^n} \psi^{(\vec{\theta})}_{\vec{y}} \otimes |\vec{y}\rangle\langle\vec{y}|_Y \,. \tag{5.9}$$

   In the context of Protocol 5.1 and Protocol 5.2, $\psi^{(\vec{\theta})}$ is the prover's state after returning $\vec{y}$ for the case where the verifier makes basis choices $\vec{\theta}$.[14] Each $\psi^{(\vec{\theta})}$ also implicitly depends on the specific keys chosen by the verifier (not just the basis choice $\vec{\theta}$); all the statements we make hold on average over key choices (for a fixed basis choice $\vec{\theta}$). Furthermore, since Protocol 5.1 and Protocol 5.2

[14] In Protocol 5.1, the only two basis choices are $\vec{\theta} = \vec{0}$ and $\vec{\theta} = \vec{1}$. However, $\psi^{(\vec{\theta})}$ is still well-defined as the state that the prover (who is defined in terms of the quantum circuits he runs on a given input) would prepare if given keys of basis choice $\vec{\theta}$, even though this never occurs in Protocol 5.1. This is different from Protocol 5.2, as it is crucial for the verifier's procedure in Protocol 5.2 to use only $\vec{0}$ or $\vec{1}$ as the basis choice. Otherwise the protocol would be "undefined".

are actually used as sub-protocols in a bigger protocol (Protocol 5.5), $\psi^{(\vec{\theta})}$ also depends on all messages exchanged (before the executions of these sub-protocols) in Protocol 5.5; for clarity we suppress this dependence from the notation, as we will see later these dependencies do not affect the rigidity proofs of these sub-protocols.

2. *In the case of Protocol 5.1, a projective measurement Π on $\mathcal{H}_D \otimes \mathcal{H}_Y$:*

$$\Pi = \left\{ \Pi^{(\vec{b},\vec{x})} = \sum_{\vec{y}} \Pi_{\vec{y}}^{(\vec{b},\vec{x})} \otimes |\vec{y}\rangle\langle\vec{y}|_Y \right\}_{\vec{b}\in\{0,1\}^n;\, \vec{x}\in\mathcal{X}^n} . \tag{5.10}$$

This is the measurement used by the prover to compute his answer $(\vec{b},\vec{x})$ in the pre-image challenge.

3. *In the case of Protocol 5.2, Π is the identity operator $\mathcal{I}$ on $\mathcal{H}_D \otimes \mathcal{H}_Y$. This is because in Protocol 5.2, there is no pre-image challenge.*

4. *A projective measurement M on $\mathcal{H}_D \otimes \mathcal{H}_Y$:*

$$M = \left\{ M^{(\vec{d})} = \sum_{\vec{y}} M_{\vec{y}}^{(\vec{d})} \otimes |\vec{y}\rangle\langle\vec{y}|_Y \right\}_{\vec{d}\in\{0,1\}^{w\times n}} . \tag{5.11}$$

This is the measurement used by the prover to compute his answer $\vec{d}$ in the Hadamard challenge. We use an additional Hilbert spaces $\mathcal{H}_R$ to record the outcomes of measuring M and write the post-measurement state after applying M to $\psi^{(\vec{\theta})}$ as

$$\sigma^{(\vec{\theta})} := \sum_{\vec{y},\vec{d}} M_{\vec{y}}^{(\vec{d})} \psi_{\vec{y}}^{(\vec{\theta})} M_{\vec{y}}^{(\vec{d})} \otimes |\vec{y},\vec{d}\rangle\langle\vec{y},\vec{d}|_{YR} . \tag{5.12}$$

5. *A set $P = \{P_q\}$, where for each $q \in \{0,1\}$, P_q is a projective measurement on $\mathcal{H}_D \otimes \mathcal{H}_Y \otimes \mathcal{H}_R$:*

$$P_q = \left\{ P_q^{(\vec{v})} = \sum_{\vec{y},\vec{d}} P_{q,\vec{y},\vec{d}}^{(\vec{v})} \otimes |\vec{y},\vec{d}\rangle\langle\vec{y},\vec{d}|_{YR} \right\}_{\vec{v}\in\{0,1\}^n} . \tag{5.13}$$

In the context of Protocol 5.1 and Protocol 5.2, given question q, the prover will measure $\{P_q^{(\vec{v})}\}$ and return the outcome $\vec{v}$ as his answer.

Definition 5.5 (Efficient devices). *A device is called* efficient *if the states $\psi^{(\vec{\theta})}$ can be prepared efficiently and the measurements Π, M, and P_q can be performed efficiently (in the sense of Definition 2.1).*

Success Probabilities of a Device

During the self-testing protocol (Protocol 5.3), the verifier applies certain checks to the answers given by the prover. If the prover fails these checks, the verifier sets a flag to $\texttt{flag}_{\texttt{Pre}}$ or $\texttt{flag}_{\texttt{Had}}$ then aborts. Here, we define the probabilities that the prover passes these checks and relate these probabilities in both protocols Protocol 5.1 and Protocol 5.2.

Definition 5.6 (Success probabilities). *For any device $D \coloneqq (S, \Pi, M, P)$ we define $\gamma_P(D_{\texttt{bb84}})$ as the device's failure probability in a pre-image round, $\gamma_H(D_{\texttt{bb84}})$ as the failure probability in a Hadamard round in Protocol 5.1 and $\gamma_H(D_{\texttt{coset}})$ as the failure probability in a Hadamard round in Protocol 5.2:*

$$\gamma_P(D_{\texttt{bb84}}) \coloneqq \Pr\left[\texttt{flag}_{\texttt{bb84}} = \texttt{flag}_{\texttt{Pre}} \mid \textit{round type} = \textit{pre-image round}\right], \tag{5.14}$$

$$\gamma_H(D_{\texttt{bb84}}) \coloneqq \Pr\left[\texttt{flag}_{\texttt{bb84}} = \texttt{flag}_{\texttt{Had}} \mid \textit{round type} = \textit{Hadamard round}\right], \tag{5.15}$$

$$\gamma_H(D_{\texttt{coset}}) \coloneqq \Pr\left[\texttt{flag}_{\texttt{coset}} = \texttt{flag}_{\texttt{Had}}\right]. \tag{5.16}$$

Next, we give the definition of a perfect prover in Protocol 5.1. Informally, a perfect prover is accepted by the verifier in a pre-image round with probability negligibly close to 1.

Definition 5.7 (Perfect device in Protocol 5.1). *We call a device D* perfect *if $\gamma_P(D_{\texttt{bb84}}) = \mathsf{negl}(\lambda)$.*

The following lemma says that for any device in Protocol 5.1 that has a non-negligible failure probability in the pre-image test, there is another perfect device that is "close" to the original one in the sense that its measurements are the same as for the original device and its states only differ by $O(\gamma_P(D))$. By using this lemma, for the rest of the rigidity proof, it suffices to only consider perfect devices: for any arbitrary device, we can first make a reduction to the corresponding perfect device at the cost of incurring an approximation error of $O(\gamma_P(D))$, and then apply our soundness proof to the perfect device.

Lemma 5.1 ([GMP22, Lemma 4.9]). *Let $D = (S, \Pi, M, P)$ be an efficient device in Protocol 5.1 with $\gamma_P(D_{\texttt{bb84}}) < 1$, where $S = \left\{\psi^{(\vec{\theta})}\right\}$. Then there exists an efficient* perfect *device $D' = (S', \Pi, M, P)$, which uses the same measurements Π, M, P and whose states $S' = \left\{\psi'^{(\vec{\theta})}\right\}$ satisfy for any $\vec{\theta} \in \{0,1\}^n$:*

$$\psi'^{(\vec{\theta})} \approx_{\gamma_P(D_{\texttt{bb84}})} \psi^{(\vec{\theta})}. \tag{5.17}$$

Proof. The proof of this lemma uses essentially the same technique to that of [MV21, Lemma 4.13], which in turn based on [Mah18b, Claim 7.2]. We give a sketch of the proof for completeness. A construction of D' is as follows. D' first prepares the states $\psi^{(\vec{\theta})}$ as D does, then applies the efficient unitary U_Π associated with the measurement Π:

$$|0\rangle\langle 0|_R \otimes \psi^{(\vec{\theta})} \overset{U_\Pi}{\longmapsto} |\vec{b}, \vec{x}\rangle\langle \vec{b}, \vec{x}|_R \otimes \Pi^{(\vec{b},\vec{x})} \psi^{(\vec{\theta})} \Pi^{(\vec{b},\vec{x})}. \tag{5.18}$$

Now D' coherently evaluates the (efficient) Chk-function on the Y-register of $\Pi^{(\vec{b},\vec{x})}\psi^{(\vec{\theta})}\Pi^{(\vec{b},\vec{x})}$ and the new register containing (b_i, x_i) for all $i \in [\![1, n]\!]$. If Chk succeeds, D' applies $U_\Pi^\dagger$ to the state, traces out the ancillary register R, and uses this as $\psi'^{(\vec{\theta})}$. Otherwise, D' repeats the process up to polynomially (in the security parameter) many times, and aborts if the Chk procedure never succeeds. Since $\gamma_P(D_{\texttt{bb84}})$ is defined as the maximum failure probability of the pre-image test, and the Chk procedure fails if the pre-image check fails on any qubit, the probability of the Chk procedure failing is at most $n \cdot \gamma_P(D_{\texttt{bb84}}) = O(\gamma_P(D_{\texttt{bb84}}))$ by a union bound.

If $1 - \gamma_P(D_{\texttt{bb84}})$ is negligible, the trace distance bound between $\psi^{(\vec{\theta})}$ and $\psi'^{(\vec{\theta})}$ is trivially satisfied. If $1 - \gamma_P(D_{\texttt{bb84}})$ is non-negligible, the probability that Chk fails polynomially

many times is negligible. Furthermore, by definition of the ENTCF family, the Chk procedure requires only the function key and not the trapdoor, which implies that it can be computed efficiently by the prover D'. It means that D' is efficient and perfect.

Fix $\vec{\theta}$. By Definition 2.4, we need to show $\left\|\psi'^{(\vec{\theta})} - \psi^{(\vec{\theta})}\right\|_1 \approx_{\gamma_P(D_{\mathsf{bb84}})^{1/2}} 0$. Since the probability of the Chk to succeed is at least $1 - O(\gamma_P(D_{\mathsf{bb84}}))$, by the gentle measurement lemma ([Wil11]), the post-measurement state after Chk has succeeded is $O(\gamma_P(D_{\mathsf{bb84}})^{1/2})$-close in trace distance to $U_\Pi(|0\rangle\langle 0|_R \otimes \psi^{(\vec{\theta})})U_\Pi^\dagger$. Because the trace distance is unitarily invariant, this implies that the state $\psi'^{(\vec{\theta})}$ is also $O(\gamma_P(D_{\mathsf{bb84}})^{1/2})$-close in trace distance to $\psi^{(\vec{\theta})}$. □

Rigidity Proof of Protocol 5.1

The rigidity proof of Protocol 5.1 follows identically from that of [GMP22]. In this section, we recall definitions and related technical lemmas from [GMP22] that are needed for our proof later. The main difference lies in the last verification procedure, in which our verification procedure also involves the Pauli keys from the QFHE. However, one can easily inspect their proof and see that this difference does not change most part of the proof. This essentially follows from the fact that the one-time pads (and generally, the homomorphic enryption) are independent of all the messages and verifier's secrets in the execution of Protocol 5.1, it only is used in the verification of the verifier as its secret input. When the difference appears, we will re-prove the lemma with respect to our protocol.

Definition 5.8 (Observables). *For a device $D \coloneqq (S, \Pi, M, P)$ with projective measurements as in Definition 5.4 and $\vec{\beta} \in \{0,1\}^n$, we define the following binary observables:*

$$Z_i = \sum_{\vec{v}} (-1)^{v_i} P_0^{(\vec{v})}, \tag{5.19}$$

$$X_i = \sum_{\vec{v}} (-1)^{v_i} P_1^{(\vec{v})}, \tag{5.20}$$

$$\tilde{X}_i = \sum_{\vec{v},\vec{y},\vec{d}} (-1)^{\beta_i \oplus v_i \oplus \hat{u}(k_i,y_i,d_i)} P_{1,\vec{y},\vec{d}}^{(\vec{v})} \otimes |\vec{y},\vec{d}\rangle\langle\vec{y},\vec{d}|_{YR}. \tag{5.21}$$

We further use the following notation for products of observables: for $\vec{a} \in \{0,1\}^n$, we define

$$Z(\vec{a}) \coloneqq Z_1^{a_1} \dots Z_n^{a_n} = \sum_{\vec{v}} (-1)^{\vec{a}\cdot\vec{v}} P_0^{(\vec{v})}, \tag{5.22}$$

and likewise for $X(\vec{a})$ and $\tilde{X}(\vec{a})$. It is easy to see that

$$\tilde{X}(\vec{a})_{\vec{y},\vec{d}} = (-1)^{\vec{a}\cdot\left(\vec{\beta}\oplus\hat{u}(\vec{k},\vec{y},\vec{d})\right)} X(\vec{a})_{\vec{y},\vec{d}}. \tag{5.23}$$

Remark 5.1. $\tilde{X}_i$ is not an observable that an efficient prover can implement because it depends on $\hat{u}(k,y,d)$, which requires the trapdoor information to be computed efficiently, and the Pauli key β, which the prover only has an encryption of it. Intuitively, while X_i describes the prover's answer, $\tilde{X}_i$ describes whether that answer is accepted by the verifier.

Definition 5.9 (Partial post-measurement states). *For $k \in \mathcal{K}_0 \cup \mathcal{K}_1$, $v \in \{0,1\}$ and $\beta \in \{0,1\}$ define the set $V_{\beta,k,v} \subseteq \mathcal{Y} \times \{0,1\}^w$ by the following condition:*

$$(y,d) \in V_{\beta,k,v} \text{ iff } \begin{cases} \hat{b}(k,y) = v & \text{if } k \in \mathcal{K}_0\,, \\ \hat{u}(k,y,d) = v \oplus \beta & \text{if } k \in \mathcal{K}_1\,. \end{cases} \tag{5.24}$$

Then for $\vec{\beta}, \vec{k}, \vec{\theta}, \vec{v}$ we define

$$\sigma^{(\vec{\beta},\vec{\theta},\vec{v})} = \sum_{y_1,d_1 \in V_{\beta_1,k_1,v_1}} \cdots \sum_{y_n,d_n \in V_{\beta_n,k_n,v_n}} \sigma^{(\vec{\theta})}_{\vec{y},\vec{d}} \otimes |\vec{y},\vec{d}\rangle\langle\vec{y},\vec{d}|\,. \tag{5.25}$$

Further for $\vec{a} \in \{0,1\}^n$ we define

$$\sigma^{(\vec{\beta},\vec{\theta},v,\vec{a})} := \sum_{\vec{v}\colon \vec{v}\cdot\vec{a}=v} \sigma^{(\vec{\beta},\vec{\theta},\vec{v})}\,. \tag{5.26}$$

Remark 5.2. In the following, once $\vec{\beta}$ is fixed, we can drop $\vec{\beta}$ from these notations and simply write $\sigma^{(\vec{\theta},\vec{v})}$ and $\sigma^{(\vec{\theta},v,\vec{a})}$. The reason is that as we explained above, the involvement of $\vec{\beta}$ is primarily a technicality needed because of our protocol construction, but does not affect the modular proofs we present here. Another way to see it is to consider $\vec{\beta}$ as a part of the trapdoor information $\vec{t}$. Then we can write $\hat{u}'(k,y,d) := \hat{u}(k,y,d) \oplus \beta$ and define $(y,d) \in V_{k,v}$ if $\hat{u}'(k,y,d) = v$ when $k \in \mathcal{K}_1$. For any statement involving these states, we understand that there is some $\vec{\beta}$ known by the verifier and these states are defined with respect to this $\vec{\beta}$.

Intuitively, when $\vec{\theta} = \vec{0}$, then for any $\vec{a} \in \{0,1\}^n$, $\sigma^{(\vec{0},v,\vec{a})}$ is that part of the state $\sigma^{(\vec{0})}$ for which the honest device would receive outcome v when measuring the observable $Z(\vec{a})$. The following lemma shows what outcomes a successful device must produce when measuring the observables from Definition 5.8 on the partial post-measurement states from Definition 5.9.

Lemma 5.2 ([GMP22, Corollary 4.18]). *Consider an efficient device $D = (S, \Pi, M, P)$ and a bit $v \in \{0,1\}$.*

1. *For any $\vec{\theta}, \vec{a} \in \{0,1\}^n$ such that $\theta_i = 0$ if $a_i = 1$, then:*

$$Z(\vec{a}) \approx_{\gamma_H(D_{\mathsf{bb84}}),\sigma^{(\vec{\theta},v,\vec{a})}} (-1)^v \mathcal{I}\,. \tag{5.27}$$

2. *For any $\vec{\theta}, \vec{a} \in \{0,1\}^n$ such that $\theta_i = 1$ if $a_i = 1$, then:*

$$X(\vec{a}) \approx_{\gamma_H(D_{\mathsf{bb84}}),\sigma^{(\vec{\theta},v,\vec{a})}} (-1)^v \mathcal{I}\,. \tag{5.28}$$

Next, we define isometries $\tilde{V}, V$ which can be shown to map the prover's observables to the corresponding Pauli observables.

Definition 5.10 (Rounding isometries [GMP22]). *For a device D with associated Hilbert space $\mathcal{H}_D$ and $\vec{y} \in \mathcal{Y}^{\times n}$, $d \in \{0,1\}^{w \times n}$, we define the isometry $\tilde{V}_{y,d} : \mathcal{H}_D \to \mathcal{H}_D \otimes \mathcal{H}_A \otimes \mathcal{H}_Q$ by the following action on an arbitrary state $|\varphi\rangle_D$:*

$$\tilde{V}_{\vec{y},\vec{d}}|\varphi\rangle_D := \mathop{\mathbb{E}}_{\vec{a},\vec{b}\in\{0,1\}^n} \left(\left(\tilde{X}(\vec{a})_{\vec{y},\vec{d}} Z(\vec{b})_{\vec{y},\vec{d}}\right)_D \otimes \left(\sigma_X(\vec{a})\sigma_Z(\vec{b})\right)_A\right)|\varphi\rangle_D \otimes \left(|\Phi^+\rangle^{\otimes n}\right)_{AQ}\,, \tag{5.29}$$

where $|\Phi^+\rangle = \frac{|00\rangle+|11\rangle}{\sqrt{2}}$ *denotes an EPR pair, and* $\left(|\Phi^+\rangle^{\otimes n}\right)_{AQ}$ *is distributed between* A *and* Q *such that every EPR pair has one qubit in either system. We can combine the different* $V_{y,d}$ *into one isometry*

$$\tilde{V} := \sum_{\vec{y},\vec{d}} \tilde{V}_{\vec{y},\vec{d}} \otimes |\vec{y},\vec{d}\rangle\langle\vec{y},\vec{d}| : \mathcal{H}_D \otimes \mathcal{H}_Y \otimes \mathcal{H}_R \to \mathcal{H}_D \otimes \mathcal{H}_Y \otimes \mathcal{H}_R \otimes \mathcal{H}_A \otimes \mathcal{H}_Q \,. \tag{5.30}$$

We similarly define

$$V_{\vec{y},\vec{d}}|\varphi\rangle_D := \mathop{\mathbb{E}}_{\vec{a},\vec{b}\in\{0,1\}^n} \left(\left(X(\vec{a})_{\vec{y},\vec{d}} Z(\vec{b})_{\vec{y},\vec{d}}\right)_D \otimes \left(\sigma_X(\vec{a})\sigma_Z(\vec{b})\right)_A\right)|\varphi\rangle_D \otimes \left(|\Phi^+\rangle^{\otimes n}\right)_{AQ} \tag{5.31}$$

and

$$V := \sum_{\vec{y},\vec{d}} V_{\vec{y},\vec{d}} \otimes |\vec{y},\vec{d}\rangle\langle\vec{y},\vec{d}| \,. \tag{5.32}$$

The following lemma relates $\tilde{V}$ and V.

Lemma 5.3. *For any keys* $\vec{k} \in \mathcal{K}_1^n$ *and* $\vec{\beta} \in \{0,1\}^n$*:*

$$V_{\vec{y},\vec{d}} = \sigma_Z\left(\hat{u}(\vec{k},\vec{y},\vec{d}) \oplus \vec{\beta}\right)_A \otimes \sigma_Z\left(\hat{u}(\vec{k},\vec{y},\vec{d}) \oplus \vec{\beta}\right)_Q \tilde{V}_{\vec{y},\vec{d}}. \tag{5.33}$$

Proof. For any state $|\varphi\rangle_D$, we have:

$$\begin{aligned}
&\sigma_Z\left(\hat{u}(\vec{k},\vec{y},\vec{d}) \oplus \vec{\beta}\right)_A \otimes \sigma_Z\left(\hat{u}(\vec{k},\vec{y},\vec{d}) \oplus \vec{\beta}\right)_Q \tilde{V}_{\vec{y},\vec{d}}|\varphi\rangle_D \\
&= \mathop{\mathbb{E}}_{a,b\in\{0,1\}^n} \left(\tilde{X}(\vec{a})_{\vec{y},\vec{d}} Z(\vec{b})_{\vec{y},\vec{d}}\right)_D |\varphi\rangle_D \otimes \\
&\qquad \left[\left(\sigma_Z\left(\hat{u}(\vec{k},\vec{y},\vec{d}) \oplus \vec{\beta}\right)\sigma_X(\vec{a})\sigma_Z(\vec{b})\right)_A \otimes \sigma_Z\left(\hat{u}(\vec{k},\vec{y},\vec{d}) \oplus \vec{\beta}\right)_Q \left(|\Phi^+\rangle^{\otimes n}\right)_{AQ}\right]
\end{aligned}$$

Repeatedly using that $(\sigma_Z)_A |\Phi^+\rangle_{AQ} = (\sigma_Z)_Q |\Phi^+\rangle_{AQ}$:

$$\begin{aligned}
&= \mathop{\mathbb{E}}_{a,b\in\{0,1\}^n} \left(\tilde{X}(\vec{a})_{\vec{y},\vec{d}} Z(\vec{b})_{\vec{y},\vec{d}}\right)_D |\varphi\rangle_D \otimes \\
&\qquad \left[\left(\sigma_Z\left(\hat{u}(\vec{k},\vec{y},\vec{d}) \oplus \vec{\beta}\right)\sigma_X(\vec{a})\sigma_Z(\vec{b})\sigma_Z\left(\hat{u}(\vec{k},\vec{y},\vec{d}) \oplus \vec{\beta}\right)\right)_A \left(|\Phi^+\rangle^{\otimes n}\right)_{AQ}\right]
\end{aligned}$$

Since $\sigma_Z\left(\hat{u}(\vec{k},\vec{y},\vec{d}) \oplus \vec{\beta}\right)\sigma_X(\vec{a})\sigma_Z(\vec{b})\sigma_Z\left(\hat{u}(\vec{k},\vec{y},\vec{d}) \oplus \vec{\beta}\right) = (-1)^{\vec{a}\cdot\left(\hat{u}(\vec{k},\vec{y},\vec{d})\oplus\vec{\beta}\right)}\sigma_X(\vec{a})\sigma_Z(\vec{b})$:

$$= \mathop{\mathbb{E}}_{a,b\in\{0,1\}^n} \left((-1)^{\vec{a}\cdot\left(\hat{u}(\vec{k},\vec{y},\vec{d})\oplus\vec{\beta}\right)} \tilde{X}(\vec{a})_{\vec{y},\vec{d}} Z(\vec{b})_{\vec{y},\vec{d}}\right)_D |\varphi\rangle_D \otimes \left[\left(\sigma_X(\vec{a})\sigma_Z(\vec{b})\right)_A \left(|\Phi^+\rangle^{\otimes n}\right)_{AQ}\right]$$

Recalling from Definition 5.8 that $(-1)^{\vec{a}\cdot\left(\hat{u}(\vec{k},\vec{y},\vec{d})\oplus\vec{\beta}\right)} \tilde{X}(\vec{a})_{\vec{y},\vec{d}} = X(\vec{a})_{\vec{y},\vec{d}}$:

$$\begin{aligned}
&= \mathop{\mathbb{E}}_{a,b\in\{0,1\}^n} \left(X(\vec{a})_{\vec{y},\vec{d}} Z(\vec{b})_{\vec{y},\vec{d}}\right)_D |\varphi\rangle_D \otimes \left[\left(\sigma_X(\vec{a})\sigma_Z(\vec{b})\right)_A \left(|\Phi^+\rangle^{\otimes n}\right)_{AQ}\right] \\
&= V|\varphi\rangle_D \,.
\end{aligned}$$

□

We then show that the isometry $\tilde{V}$ maps the observables $\tilde{X}(\vec{a})Z(\vec{b})$ to the corresponding Pauli observables.

Lemma 5.4 ([GMP22, Lemma 4.28]). *For an efficient perfect device $D = (S, \Pi, M, P)$ and any $\vec{a}, \vec{b} \in \{0,1\}^n$ we have*

$$\mathrm{Tr}\left[\tilde{V}^\dagger \left(\sigma_X(\vec{a})\sigma_Z(\vec{b})\right)_Q^\dagger \tilde{V}\tilde{X}(\vec{a})_{DYR}Z(\vec{b})_{DYR}\sigma^{(\vec{1})}_{DYR}\right] \approx_{n^{1/2}\gamma_H(D_{\text{bb84}})^{1/8}} 1\,. \tag{5.34}$$

By combining Lemma 5.3 and Lemma 5.4 we can show that the isometry V maps the observables $X(\vec{a})Z(\vec{b})$ to the corresponding Pauli observables.

Lemma 5.5 ([GMP22, Proposition 4.29]). *For an efficient perfect device $D = (S, \Pi, M, P)$ and any $\vec{a}, \vec{b} \in \{0,1\}^n$ we have*

$$VX(\vec{a})Z(\vec{b})V^\dagger \approx_{n^{1/2}\gamma_H(D_{\text{bb84}})^{1/8}, V\sigma^{(\vec{1})}V^\dagger} \left(\sigma_X(\vec{a})\sigma_Z(\vec{b})\right)_Q \otimes \mathcal{I}_{YRDA}\,. \tag{5.35}$$

Rigidity Proof of Protocol 5.2

Having established a characterization of the prover's observables $X(\vec{a})Z(\vec{b})$ in Protocol 5.1, we now use this to characterize the prover's behavior in Protocol 5.2.

Step 1: Modeling. First, we introduce the corresponding notion of post-measurement states for an efficient device of Protocol 5.2. Note that the two protocols are identical from the prover's point of view when the round type is the Hadamard round, and the marginal observables from Definition 5.8 are defined for Hadamard round. Thus we can use the same notation of marginal observables from Definition 5.8 (in particular, we only need the efficient observables $X(\vec{a})$ and $Z(\vec{b})$) for an efficient device in Protocol 5.2.

Definition 5.11. *For $\vec{k} \in (\mathcal{K}_0 \cup \mathcal{K}_1)^n$, $\vec{v} \in \{0,1\}^n$ and $A \subseteq \mathbb{F}_2^n$, $\vec{\alpha}, \vec{\beta} \in \{0,1\}^n$ define the set $V_{A,\vec{\alpha},\vec{\beta},\vec{k},\vec{v}} \subseteq \mathcal{Y}^n \times \{0,1\}^{w\times n}$ by the following condition:*

$$(\vec{y}, \vec{d}) \in V_{A,\vec{\alpha},\vec{\beta},\vec{k},\vec{v}} \text{ iff } \begin{cases} \hat{b}(\vec{k}, \vec{y}) = \vec{v} \in A + \vec{\alpha} & \text{if } \vec{k} \in \mathcal{K}_0^n\,, \\ \hat{u}(\vec{k}, \vec{y}, \vec{d}) \oplus \vec{v} \in A^\perp + \vec{\beta} & \text{if } \vec{k} \in \mathcal{K}_1^n\,. \end{cases} \tag{5.36}$$

Then for $\vec{\alpha}, \vec{\beta}, \vec{k}, \vec{\theta} \in \{\vec{0}, \vec{1}\}, \vec{v}$ we define

$$\sigma^{(A,\vec{\alpha},\vec{\beta},\vec{\theta},\vec{v})} = \sum_{\vec{y},\vec{d} \in V_{A,\vec{\alpha},\vec{\beta},\vec{k},\vec{v}}} \sigma^{(\vec{\theta})}_{\vec{y},\vec{d}} \otimes |\vec{y}, \vec{d}\rangle\langle \vec{y}, \vec{d}|\,. \tag{5.37}$$

By the same argument as in Remark 5.2, we can write $\sigma^{(\vec{\theta},\vec{v})}$ for simplicity.

We note that different from Definition 5.9, we only consider two basis choices $\vec{\theta} = \vec{0}$ or $\vec{\theta} = \vec{1}$, whereas the post-measurement states in Definition 5.9 can be defined with respect to any basis choice. Similar to Lemma 5.2, we analyze what outcomes a successful device must produce when measuring the observables from Definition 5.8 on the post-measurement states from Definition 5.11.

Lemma 5.6. *For any efficient device $D = (S, \Pi, M, P)$, a coset state description (A, α, β):*

$$\sum_{\vec{v} \in S_0} \mathrm{Tr}\left[Z_i^{(v_i)}\sigma^{(\vec{0},\vec{v})}\right] \approx_{\gamma_H(D_{\text{coset}})} 1\,, \tag{5.38}$$

$$\sum_{\vec{v} \in S_1} \mathrm{Tr}\left[X_i^{(v_i)}\sigma^{(\vec{1},\vec{v})}\right] \approx_{\gamma_H(D_{\text{coset}})} 1\,, \tag{5.39}$$

where $S_0 \coloneqq A + \alpha$ and $S_1 \coloneqq A^\perp + \beta - \hat{u}(\vec{k}, \vec{y}, \vec{d})$.

Proof. We first prove Equation (5.38). Since the case $q = \theta = 0$ occurs with probability $1/2$ in Protocol 5.2, the device's failure probability in this case can be at most $2\gamma_H(D_{\mathsf{coset}})$. Furthermore, since the device only succeeds if $v_i = \hat{b}(k_i, y_i)$ and $\vec{v} \in A+\alpha$ for all $i \in [\![1, n]\!]$ in the protocol, it means that the device succeeds with probability at least $1 - 2\gamma_H(D)$. Now comparing the definition of $\sigma^{(\vec{0},\vec{v})}$ with the verifier's checks in the protocol, this means that for all $i \in [\![1, n]\!]$:

$$\sum_{v \in S_0} \mathrm{Tr}\left[Z_i^{(v_i)} \sigma^{(\vec{0},\vec{v})}\right] \geq 1 - 2\gamma_H(D)\,.$$

For the inequality in the other direction, we note that since $Z_i^{(v_i)}$ is a projector, we immediately have

$$\sum_{\vec{v} \in S_0} \mathrm{Tr}\left[Z_i^{(v_i)} \sigma^{(\vec{0},\vec{v})}\right] \leq \sum_{\vec{v} \in S_0} \mathrm{Tr}\left[\sigma^{(\vec{0},\vec{v})}\right] = \mathrm{Tr}\left[\sigma^{(\vec{0})}\right] = 1,$$

finishing the proof of Equation (5.38).

The proof of Equation (5.39) is completely analogous, combining with the fact that if $\vec{v} + \hat{u}(\vec{k}, \vec{y}, \vec{d}) \in A^\perp + \beta$ iff $\vec{v} \in A^\perp + \beta - \hat{u}(\vec{k}, \vec{y}, \vec{d})$. □

Step 2: Relating Protocol 5.1 and Protocol 5.2. We relate the prover's operators and states in Protocol 5.1 and Protocol 5.2 by the following lemmas.

Lemma 5.7. *For any efficient devices D, D' with the notation given in Definition 5.4. Assume that D is a device of Protocol 5.1 with corresponding states $(\psi^{(\vec{\theta})}, \sigma^{(\vec{\theta})})$ and D' is a device of Protocol 5.2 with corresponding states $(\psi'^{(\vec{\theta}')}, \sigma'^{(\vec{\theta}')})$. Then*

$$\psi^{(\vec{\theta})} \overset{c}{\approx}_0 \psi'^{(\vec{\theta}')}\,, \tag{5.40}$$

and

$$\sigma^{(\vec{\theta})} \overset{c}{\approx}_0 \sigma^{(\vec{\theta}')}\,. \tag{5.41}$$

Proof. At the beginning of each protocol's execution: in Protocol 5.1, the device's state is (encrypted) BB84 states, while in Protocol 5.2, the device's state is (encrypted) coset states. Furthermore, note that executing Protocol 5.1 or Protocol 5.2 does not require the secret key of the QFHE encryption scheme. Equation (5.40) then follows directly from semantic security of the QFHE encryption scheme.

In Protocol 5.2, the verifier never sends a "pre-image round" challenge. In Protocol 5.1, the round type is chosen uniformly at random, so with probability $\frac{1}{2}$, the round type is "Hadamard round". In this case, the execution of two protocols are identical from the prover's point of view. Since the prover is efficient, Equation (5.41) also follows. □

We then obtain the following relation between the success probabilities of devices in Protocol 5.1 and Protocol 5.2.

Corollary 5.1. *For any efficient device $D := (S, \Pi, M, P)$:*

$$\gamma_H(D_{\mathsf{bb84}}) \overset{c}{\approx}_0 2\gamma_H(D_{\mathsf{coset}}). \tag{5.42}$$

Remark 5.3. Due to the relation in Equation (5.42) and the definition of the "$\approx$"-notation (Definition 2.4), from now on, we drop the subscript and simply write $\gamma_H(D)$ when it is clear from the context.

Combining Corollary 5.1 and Lemma 5.7, using the same isometry V defined in Definition 5.10, we can "lift" the approximate-equality relations described in Lemma 5.5 for an efficient device in Protocol 5.1 to an efficient device in Protocol 5.2.

Lemma 5.8. *For an efficient perfect device $D = (S, \Pi, M, P)$ in Protocol 5.2 and any $\vec{a}, \vec{b} \in \{0,1\}^n$ we have*

$$VX(\vec{a})Z(\vec{b})V^\dagger \approx_{n^{1/8}\gamma_H(D)^{1/32}, V\sigma^{(\vec{1})}V^\dagger} \left(\sigma_X(\vec{a})\sigma_Z(\vec{b})\right)_Q \otimes \mathcal{I}_{YRDA}\,. \tag{5.43}$$

Proof. The lemma follows directly from the lifting lemma (Item 6 of Lemma 2.3) and the fact that the isometry V and the operators X, Z are efficient. Using the notation from Lemma 2.3, we have $\delta = 0$, $\varepsilon = n^{1/2}\gamma_H(D)^{1/8}$, the isometry is V, the observable A is $X(\vec{a})Z(\vec{b})$, the observable B is $\sigma_X(\vec{a})\sigma_z(\vec{b}) \otimes \mathcal{I}$. The two states are $V\sigma'^{(\vec{1})}V^\dagger$ of a device in Protocol 5.1 and $V\sigma^{(\vec{1})}V^\dagger$ of a device in Protocol 5.2. □

Step 3: Rigidity. We first prove the following technical lemma.

Lemma 5.9. *For an efficient device $D = (S, \Pi, M, P)$, a coset state description (A, α, β):*

$$\sum_{\vec{v}\in S_0} |\vec{v}\rangle\langle\vec{v}| \otimes (\sigma_{Z,i}^{(v_i)})_Q \approx_{\varepsilon, \sum_{\vec{v}'\in S_0} |\vec{v}'\rangle\langle\vec{v}'| \otimes V\sigma^{\vec{0},\vec{v}'}V^\dagger} \mathcal{I}\,, \tag{5.44}$$

$$\sum_{\vec{v}\in S_1} |\vec{v}\rangle\langle\vec{v}| \otimes (\sigma_{X,i}^{(v_i)})_Q \approx_{\varepsilon, \sum_{\vec{v}'\in S_1} |\vec{v}'\rangle\langle\vec{v}'| \otimes V\sigma^{\vec{1},\vec{v}'}V^\dagger} \mathcal{I}\,, \tag{5.45}$$

where $S_0 = A + \alpha$, $S_1 = A^\perp + \beta - \hat{u}(\vec{k}, \vec{y}, \vec{d})$ and the approximation factor ε will be clarified later in the proof.

Proof. We first prove the first statement. It is easy to check that $\sum_{\vec{v}\in V} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\sigma_{Z,i}^{(v_i)}\right)_Q$ is a projector, so we can expand the definition of the state-dependent distance and compute:

$$\begin{aligned}
&\mathrm{Tr}\left[\left(\sum_{\vec{v}\in S_0} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\sigma_{Z,i}^{(v_i)}\right)_Q - \mathcal{I}\right)^\dagger \left(\sum_{\vec{v}\in S_0} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\sigma_{Z,i}^{(v_i)}\right)_Q - \mathcal{I}\right) \sum_{\vec{v}'\in S_0} |\vec{v}'\rangle\langle\vec{v}'| \otimes V\sigma^{(\vec{0},\vec{v}')}V^\dagger\right] \\
&= \mathrm{Tr}\left[\left(\mathcal{I} - \sum_{\vec{v}\in S_0} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\sigma_{Z,i}^{(v_i)}\right)_Q\right) \sum_{\vec{v}'\in S_0} |\vec{v}'\rangle\langle\vec{v}'| \otimes V\sigma^{(\vec{0},\vec{v}')}V^\dagger\right] \\
&= 1 - \sum_{\vec{v}\in S_0} \mathrm{Tr}\left[\left(|\vec{v}\rangle\langle\vec{v}| \otimes \left(\sigma_{Z,i}^{(v_i)}\right)_Q\right) \sum_{\vec{v}'\in S_0} |\vec{v}'\rangle\langle\vec{v}'| \otimes V\sigma^{(\vec{0},\vec{v}')}V^\dagger\right] \\
&= 1 - \sum_{\vec{v}\in S_0} \mathrm{Tr}\left[\left(\sigma_{Z,i}^{(v_i)}\right)_Q V\sigma^{(\vec{0},\vec{v})}V^\dagger\right]\,,
\end{aligned}$$

To show the first part of the lemma, we need to show that

$$\sum_{\vec{v}\in S_0} \mathrm{Tr}\left[\left(\sigma_{Z,i}^{(v_i)}\right)_Q V\sigma^{(\vec{0},\vec{v})}V^\dagger\right] \approx_\varepsilon 1\,. \tag{5.46}$$

For this, recall from Lemma 5.8 that we have

$$VZ_iV^\dagger \approx_{n^{1/8}\gamma_H(D)^{1/32}, V\sigma^{(\vec{1})}V^\dagger} (\sigma_{Z,i})_Q \otimes \mathcal{I}_{YRDA}\,. \tag{5.47}$$

For shorthand, write $\gamma := n^{1/8}\gamma_H(D)^{1/32}$. Since V and Z_i are efficient, by the lifting lemma (Lemma 2.3) and the fact that $\sigma^{(\vec{0})} \overset{c}{\approx}_0 \sigma^{(\vec{1})}$, this implies that:

$$VZ_iV^\dagger \approx_{\gamma^{1/4}, V\sigma^{(\vec{0})}V^\dagger} (\sigma_{Z,i})_Q \otimes \mathcal{I}_{YRDA}\,. \tag{5.48}$$

Using Lemma 2.4 and Lemma 2.5, we get:

$$\sum_{\vec{v}\in S_0} VZ_i^{(v_i)}V^\dagger \approx_{\gamma^{1/4}, \sum_{\vec{v}\in S_0} V\sigma^{(\vec{0},\vec{v})}V^\dagger} \sum_{\vec{v}\in S_0} \left(\sigma_{Z,i}^{(v_i)}\right)_Q \otimes \mathcal{I}_{YRDA}\,. \tag{5.49}$$

Using the replacement lemma (Lemma 2.1), we obtain

$$\begin{aligned}
\sum_{\vec{v}\in S_0} \mathrm{Tr}\left[\left(\sigma_{Z,i}^{(v_i)}\right)_Q V\sigma^{(\vec{0},v_i,\vec{1}^i)}V^\dagger\right] &\approx_{\gamma^{1/8}} \sum_{\vec{v}\in S_0} \mathrm{Tr}\left[VZ_i^{(v_i)}V^\dagger V\sigma^{(\vec{0},\vec{v})}V^\dagger\right] && (5.50)\\
&= \sum_{\vec{v}\in S_0} \mathrm{Tr}\left[Z_i^{(v_i)}\sigma^{(\vec{0},\vec{v})}\right] && (5.51)\\
&\approx_{\gamma_H(D)} 1\,, && (5.52)
\end{aligned}$$

where the last line follows from Equation (5.38). Set $\varepsilon := \gamma^{1/8}$, this finishes the proof of the first statement.

For the second statement, we can perform the same calculation, but use Equation (5.39). □

Lemma 5.10. *For an efficient perfect device $D = (S, \Pi, M, P)$, a coset state description (A, α, β) and $\vec{\theta} \in \{\vec{0}, \vec{1}\}$, there exists a set of subnormalized states $\{\rho_i^{(\vec{\theta},\vec{v})}\}_{\vec{v}\in S_i}$ where S_i for $i \in \{0,1\}$ are defined as in Lemma 5.9 such that*

$$\sum_{\vec{v}\in S_i} |\vec{v}\rangle\langle\vec{v}| \otimes V\sigma^{(\vec{\theta},\vec{v})}V^\dagger \approx_{2n\varepsilon} \sum_{\vec{v}\in S_i} |\vec{v}\rangle\langle\vec{v}| \otimes \left((\mathsf{H}^{\otimes n})^i|\vec{v}\rangle\langle\vec{v}|(\mathsf{H}^{\otimes n})^i\right)_Q \otimes \rho_i^{(\vec{\theta},\vec{v})}\,, \tag{5.53}$$

where $i = 0$ if $\vec{\theta} = \vec{0}$ and $i = 1$ if $\vec{\theta} = \vec{1}$.

Proof. We define the shorthand

$$M(\theta) = \begin{cases} Z & \text{if } \theta = 0\,, \\ X & \text{if } \theta = 1\,. \end{cases}$$

Applying Lemma 5.9 and Lemma 2.2 to get

$$\begin{aligned}
&\sum_{\vec{v}\in S_i} |\vec{v}\rangle\langle\vec{v}| \otimes V\sigma^{(\vec{\theta},\vec{v})}V^\dagger \\
&\approx_\varepsilon \left(\sum_{\vec{v}\in S_i} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\sigma_{M(\theta_1),1}^{(v_1)}\right)_Q\right) \sum_{\vec{v}\in S_i} |\vec{v}\rangle\langle\vec{v}| \otimes V\sigma^{(\vec{\theta},\vec{v})}V^\dagger \left(\sum_{\vec{v}\in S_i} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\sigma_{M(\theta_1),1}^{(v_1)}\right)_Q\right)
\end{aligned}$$

We repeat this for the remaining indices $j = 2, \ldots, n$. Since there are in total n steps,

the total approximation error will be $n\varepsilon$. We then have

$$\begin{aligned}
&\sum_{\vec{v}\in S_i} |\vec{v}\rangle\langle\vec{v}| \otimes V\sigma^{(\vec{\theta},\vec{v})}V^\dagger \\
&\approx_{n\varepsilon} \left(\sum_{\vec{v}\in S_i} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\sigma^{(v_1)}_{M(\theta_1),1}\right)_Q\right) \cdots \left(\sum_{\vec{v}\in S_i} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\sigma^{(v_n)}_{M(\theta_n),n}\right)_Q\right) \sum_{\vec{v}\in S_i} |\vec{v}\rangle\langle\vec{v}| \otimes V\sigma^{(\vec{\theta},\vec{v})}V^\dagger \\
&\qquad\qquad \left(\sum_{\vec{v}\in S_i} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\sigma^{(v_1)}_{M(\theta_1),1}\right)_Q\right) \cdots \left(\sum_{\vec{v}\in S_i} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\sigma^{(v_n)}_{M(\theta_n),n}\right)_Q\right) \\
&= \sum_{\vec{v}\in S_i} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\prod_j \sigma^{(v_j)}_{M(\theta_j),j}\right)_Q V\sigma^{(\vec{\theta},\vec{v})}V^\dagger \left(\prod_j \sigma^{(v_j)}_{M(\theta_j),j}\right)_Q .
\end{aligned}$$

Now noting that $\prod_j \sigma^{(v_j)}_{M(\theta_j),j} = (\mathsf{H}^{\otimes n})^i|\vec{v}\rangle\langle\vec{v}|(\mathsf{H}^{\otimes n})^i$, we obtain

$$\begin{aligned}
&= \sum_{\vec{v}\in S_i} |\vec{v}\rangle\langle\vec{v}| \otimes \left((\mathsf{H}^{\otimes n})^i|\vec{v}\rangle\langle\vec{v}|(\mathsf{H}^{\otimes n})^i\right)_Q \otimes \left(\langle v|\,(\mathsf{H}^{\otimes n})^i\right)_Q V\sigma^{(\vec{\theta},\vec{v})}V^\dagger \left((\mathsf{H}^{\otimes n})^i\,|v\rangle\right)_Q \\
&= \sum_{\vec{v}\in S_i} |\vec{v}\rangle\langle\vec{v}| \otimes \left((\mathsf{H}^{\otimes n})^i|\vec{v}\rangle\langle\vec{v}|(\mathsf{H}^{\otimes n})^i\right)_Q \\
&\qquad \otimes \mathrm{Tr}_Q\left[\left((\mathsf{H}^{\otimes n})^i|\vec{v}\rangle\langle\vec{v}|(\mathsf{H}^{\otimes n})^i\right)_Q V\sigma^{(\vec{\theta},\vec{v})}V^\dagger \left((\mathsf{H}^{\otimes n})^i|\vec{v}\rangle\langle\vec{v}|(\mathsf{H}^{\otimes n})^i\right)_Q\right]
\end{aligned}$$

Analogously to how we added the factors $\prod_j \sigma^{(v_j)}_{M(\theta_j),j}$ in a previous step, we can now replace the factors $((\mathsf{H}^{\otimes n})^i|\vec{v}\rangle\langle\vec{v}|(\mathsf{H}^{\otimes n})^i)_Q$ inside the partial trace by identity, resulting in

$$\approx_{2n\varepsilon} \sum_{\vec{v}\in S_i} |\vec{v}\rangle\langle\vec{v}| \otimes \left((\mathsf{H}^{\otimes n})^i|\vec{v}\rangle\langle\vec{v}|(\mathsf{H}^{\otimes n})^i\right)_Q \otimes \mathrm{Tr}_Q\left[V\sigma^{(\vec{\theta},\vec{v})}V^\dagger\right] .$$

We then obtain the desired statement by defining

$$\rho_i^{(\vec{\theta},\vec{v})} := \mathrm{Tr}_Q\left[V\sigma^{(\vec{\theta},\vec{v})}V^\dagger\right] , \tag{5.54}$$

with $i = 0$ if $\vec{\theta} = \vec{0}$ and $i = 1$ if $\vec{\theta} = \vec{1}$. □

What Lemma 5.10 says is that up to an isometry, with inverse polynomial error, the device's state must be (information-theoretically) close to a mixed state of vectors in S_i, tensored with an auxiliary state $\rho_i^{(\vec{\theta},\vec{v})}$. We note that it is not hard to show that $\rho_0^{(\vec{0},\vec{v})} \overset{c}{\approx}_0 \rho_1^{(\vec{1},\vec{v})}$. (Though it is not necessary for our soundness proof.)

Furthermore, from the statement of Lemma 5.10, for a fixed efficient device D, if we run Protocol 5.2 "coherently" in superposition, then

(i) when $\vec{\theta} = \vec{0}$, the device's state must be in superposition of all vectors in S_0, that is $|A + \alpha\rangle$,

(ii) when $\vec{\theta} = \vec{0}$, the device's state must be in superposition of all vectors in S_1. By applying a correction (XOR-ing the register Q with $\hat{u}(\vec{k},\vec{y},\vec{d})$), the state would be $|A^\perp + \beta\rangle$.

Thus, with the verifier in Protocol 5.2, we obtain efficient projective measurements to characterize the prover's initial state. Formally, let O_0 be the following process: run Protocol 5.2 in superposition (without measuring any intermediate messages such as y, d, v) with the basis choice $\vec{\theta} = \vec{1}$ and check if the register Q at the end of the protocol is $|A + \alpha\rangle$. O_1 is defined analogously for $\vec{\theta} = \vec{1}$, and it applies a correction by XORing the register Q with $\hat{u}(\vec{k}, \vec{y}, \vec{d})$ and check if the register Q at the end is $|A^{\perp} + \beta\rangle$. We obtain the main technical lemma.

Lemma 5.11. *For any efficient device D, the initial state of the device ψ must be close to (up to some inverse polynomial error) $|A_{\alpha,\beta}\rangle \otimes \rho$:*

$$\psi \approx_{4n\varepsilon} |A_{\alpha,\beta}\rangle \otimes \rho. \tag{5.55}$$

Proof. Let U_0 and U_1 be the efficient unitaries corresponding to operators O_0 and O_1 defined above. Fix a device D. We first apply $U_0\psi$ and record the output to an ancilla register. If the output is 1, apply the inverse $U_0^{\dagger}$ to obtain ψ'. Finally apply $U_1\psi'$. If the output is 1, by the definition of U_i (and O_i), the lemma follows. Note that for each application of U_i, the approximation error is $2n\varepsilon$ which comes from Lemma 5.10. □

Rigidity Proof of Protocol 5.3

We are now ready to prove the rigidity of Protocol 5.3, namely that any efficient quantum prover that does not cause the protocol to abort must have the initial state close to a hidden coset state.

Lemma 5.12. *For any $\lambda \in \mathbb{N}$, there exist choices $M = \mathsf{poly}(\lambda)$ and $\delta = 1/\mathsf{poly}(\lambda)$ such that if the verifier executes Protocol 5.3 with an efficient quantum prover whose success probability is lower-bounded by an inverse polynomial, the following holds. Let (A, α, β) the private input of the verifier for the coset instance. Denoting the probability that the protocol does not abort as $\Pr[\top]$, and let ψ the initial state of the prover. Then, with probability $\Pr[\top]$, we have*

$$\psi \overset{c}{\approx}_{\varepsilon} |A_{\alpha,\beta}\rangle \otimes \rho, \tag{5.56}$$

for some auxiliary state ρ, and the approximation error ε is inverse polynomial on the security parameter λ.

Proof. Essentially, we can see Protocol 5.3 as a cut-and-choose protocol in which the number of evaluation instances is 1 and the number of check instances is $M^2 - 1$. We then can reduce this lemma to Lemma 5.11 using the same argument as in [GMP22, Theorem 4.33]. We omit the details. □

Remark 5.4. We make few comments on the inverse polynomial soundness.[15] First of all, what the soundness lemma (Lemma 5.12) says is effectively the same as a typical self-testing statement, which is that: if the prover succeeds with probability $1 - \varepsilon$ in the protocol, the state it used in the protocol must be, up to an isometry, $\mathsf{poly}(\varepsilon)$-close to ideal (in our setting, the closeness is measured by computational distinguishability rather than trace distance, as in typical self-testing settings). Now, in practice, we would have

[15] We thank Alexandru Gheorghiu for providing us this insightful comments.

to estimate ε by doing many runs of the protocol. In particular, we would need about $1/\varepsilon^2$ repetitions to have high (that is, $1 - \mathsf{negl}(\lambda)$) confidence that the prover's success probability is $1 - \varepsilon$. This implies that if we want ε to be negligible, we would have to do superpolynomial-many repetitions of the protocol and since this is not efficient, we are limited to $\varepsilon = 1/\mathsf{poly}(\lambda)$. It is from doing this $1/\varepsilon^2$ repetitions that we go from the original self-testing statement (Lemma 5.11) to the statement that characterizes the prover's state in the actual protocol.

We now finish this section with the proof of Proposition 5.2.

Proof of Proposition 5.2. Since in the final protocol (Protocol 5.5), we run N instances over $2N$ possible instances of the self-testing protocol (Protocol 5.3) (in the cut-and-choose fashion), we can invoke techniques developed in [BF10] to relate quantum sampling to classical sampling and conclude Proposition 5.2.

In particular, consider the following interaction between a quantum prover $\mathcal{P}$ and a challenger $\mathcal{V}$.

1. $\mathcal{P}$ and $\mathcal{V}$ jointly execute Protocol 5.5. Let $\overline{T}$ be the set of N indices chosen uniformly at random by $\mathcal{V}$ in N runs of the self-testing protocol.

2. Let X_i be the outcome of each of N runs of the self-testing protocol. $\mathcal{V}$ verifies that $X_i = \text{accept}$ for all $i \in \overline{T}$, and aborts otherwise.

This is a natural quantum analogue of the following classical sampling experiment ([BF10, Example 1]) on a length-$2N$ bitstring X to test if X is close to the all-zero string:

1. randomly select a size-N subset $\overline{T} \subset [\![1, 2N]\!]$,

2. compute $\omega(X|_{\overline{T}})$, and accept if the estimate vanishes and else reject.

Noting that this sample-and-estimate strategy is exactly the Ψ_{uniform} strategy described at the end of Section 2.2.5, we have by Corollary 2.1 that the quantum error probability of this strategy is bounded by $2\exp(\frac{-n\delta^2}{64})$, for $\delta = 1/2$. By the definition of quantum error probability (Definition 2.6), this means that, with overwhelming probability over $\overline{T}$, the state of the prover $\mathcal{P}$ in the remaining set T also satisfies Equation (5.55). Indeed, by changing of basis, this reduces to the question of testing if the state of the prover before running the self-testing protocol is close to the all-zero state. Then the quantum sample-and-estimate technique tells us that the state of the prover must be supported on vectors with relative Hamming distance $< 1/2$, and it means there must be at least 1 bit in string which is 0. If this is the case, it corresponds (up to some inverse polynomial error) to the coset state $|A_{\alpha,\beta}\rangle$ in Equation (5.55). This completes the proof of the proposition. □

5.5.2 Soundness of Protocol 5.5

We now formally define the notion of soundness for our protocol, which is described as a coset monogamy game similar to Definition 5.3.

Definition 5.12 — Soundness

For any QPT prover $\mathcal{P} = \{\mathcal{P}_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$ interacting with a PPT verifier $\mathcal{V}$ in Protocol 5.5, after which $\mathcal{V}$ outputs $\{S_i, \alpha_i, \beta_i\}_{i\in T}$ and $\mathcal{P}$ outputs a state ψ, let $(\{S_i, \alpha_i, \beta_i\}_{i\in T}, \psi) \leftarrow \langle \mathcal{P}_\lambda(\rho_\lambda), \mathcal{V}(1^\lambda)\rangle$ denote this interaction. The prover (now modeled as a triple algorithm $(\mathcal{P}, \mathcal{B}, \mathcal{C})$) then interacts with the verifier in the following monogamy game.

(1) Splitting. The prover applies a CPTP map to split ψ into a bipartite state ψ_{BC}; it sends the register B to $\mathcal{B}$ and the register C to $\mathcal{C}$. No communication is allowed between $\mathcal{B}$ and $\mathcal{C}$ after this phase.

(2) Question. The verifier sends the description of $\{S_i\}_{i\in T}$, to both $\mathcal{B}$ and $\mathcal{C}$.

(3) Answer. $\mathcal{B}$ returns $s_1^{(i)} \in \mathbb{F}_2^n$ and $\mathcal{C}$ returns $s_2^{(i)} \in \mathbb{F}_2^n$ for all $i \in T$.

The prover $(\mathcal{P}, \mathcal{B}, \mathcal{C})$ wins if and only if $s_1^{(i)} \in S_i + \alpha_i$ and $s_2^{(i)} \in S_i^\perp + \beta_i$ for all $i \in T$. Let $\mathsf{SMCosetMonogamy}(\mathcal{P}, \lambda)$ be a random variable which takes the value 1 if the game above is won by the prover $(\mathcal{P}, \mathcal{B}, \mathcal{C})$, and takes the value 0 otherwise.
The protocol is secure if the winning probability of any QPT adversary is negligible. Formally, for any QPT malicious prover, the protocol is *computationally sound* if we have

$$\Pr[\mathsf{SMCosetMonogamy}(\mathcal{P}, \lambda) = 1] \leq \mathsf{negl}(\lambda).$$

Theorem 5.2. *Protocol 5.5 is computationally sound, according to Definition 5.12.*

Proof. Let $\mathcal{P} = \{\mathcal{P}_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$ a quantum polynomial time adversary that succeeds in the game SMCosetMonogamy with some non-negligible probability $\varepsilon = \{\varepsilon_\lambda\}_{\lambda\in\mathbb{N}}$. Let $(\{S_i, \alpha_i, \beta_i\}_{i\in T}, \psi) \leftarrow \langle \mathcal{P}_\lambda(\rho_\lambda), \mathcal{V}(1^\lambda)\rangle$. This means that $\mathcal{P} = (\mathcal{P}, \mathcal{B}, \mathcal{C})$ is able to output a pair $(s_1^{(i)}, s_2^{(i)}) \in (S_i + \alpha_i) \times (S_i^\perp + \beta_i)$ for all $i \in T$ in the monogamy game defined in Definition 5.12.

Let $\delta' \in (0, 1]$ the sub-exponential security level of the QFHE (that is, any QPT adversary cannot break the semantic security of the QFHE with advantage bigger than $2^{\lambda^{\delta'}}$), and denote $\delta := \frac{\delta'}{2}$.

We next describe a sequence of hybrid experiments.[16]

Game G_0: This is the original experiment.

We define G_0 as the original attack, where $\mathcal{P}$ interacts with the verifier in Protocol 5.5 and wins the monogamy game SMCosetMonogamy. We say G_0 is successful if $\mathsf{SMCosetMonogamy}(\mathcal{P}, \lambda) = 1$. The experiment G_0 is thus successful with probability ε.

Game G_1: Changing the success definition of the experiment.

Pick a random index $i \in T$, for shorthand, denote this coset instance as (S, α, β), and the adversary's corresponding output in the monogamy game is (s_1, s_2). In the current hybrid, the experiment is defined to be successful if $s_1 \in S + \alpha$ and $s_2 \in S^\perp + \beta$. In particular, in the current hybrid, we only consider the monogamy game for a random instance among $|T|$ coset instances. (The other instances are not considered). Apparently,

[16] Some hybrids follow from the proof given in [Shm22a].

G_1 is successful with probability at least ε. From now on, we only consider this coset instance in later hybrids, and all the changes are only applied to this instance.

Game G_2: Injecting quantum communication into the interaction between the prover and the verifier.

This hybrid is identical to G_1 except that now we consider the verifier as a QPT algorithm instead of a PPT algorithm, and we make an additional round of interaction using quantum communication in the protocol. (Think about the verifier now as a QPT challenger of the experiment.) In particular, right after the last step of Protocol 5.5 (step 9c), we ask the prover to send the coset state $|S_{\alpha,\beta}\rangle$ to the verifier. Denote this state as $|\$\rangle$. The verifier then does the following:

- Verify the received coset state:
  (a) Checks that the output qubit of the computation $i\mathcal{O}(S+\alpha)(|\$\rangle)$[17] is 1.
  (b) Execute Hadamard transform $\mathsf{H}^{\otimes\lambda}$ on $|\$\rangle$ to obtain $|\$'\rangle$ and then check the output qubit of the computation $i\mathcal{O}(S^{\perp}+\beta)(|\$'\rangle)$ is 1.
- If any of these checks returns 0, abort and declare the game as a failure.
- Execute $\mathsf{H}^{\otimes\lambda}$ again on $|\$'\rangle$ to obtain $|\$''\rangle$ and send $|\$''\rangle$ back to the prover.

From Proposition 5.2, it follows that with probability at least $1/|T|$, the adversary's output state ϕ is inverse polynomially ϵ-close to $|S_{\alpha,\beta}\rangle\otimes\rho$ for some auxiliary state ρ. It means that when it is asked, the adversary can always send a state $|\$\rangle$ that is inverse polynomially ϵ-close to $|S_{\alpha,\beta}\rangle$ to the challenger.

Note that the quantum verification described above executes only on the register containing $|\$\rangle$ and thus commutes with any other quantum operation on a register entangled with it at the point where $\mathcal{P}$ finishes executing the real protocol Protocol 5.5. Thus after finishing the above additional interaction, the adversary's state is unchanged, if the verification passed.

The probability that the adversary does not fail in the experiment is $1-\epsilon$. It is then clear that, for any adversary that wins the G_1 with probability ε, it wins G_2 with probability at least $\varepsilon' := \varepsilon(1-\epsilon)/|T|$. Thus, the success probability of G_2 is ε' for some non-negligible ε'.

Game G_3: Removing subspace information from obfuscated circuits.

This hybrid is identical to G_2, with the only difference is that when the verifier returns the obfuscations P_0, P_1 in the last step of Protocol 5.5 (Step 9c), the obfuscations are changed: We sample two random $(\lambda-\lambda^{\delta})$-dimensional subspaces $T_0, T_1 \subseteq \mathbb{F}_2^{\lambda}$ subjected to $T_1^{\perp} \subseteq S \subseteq T_0$. The verifier uses $i\mathcal{O}(T_0+\alpha)$ instead of $i\mathcal{O}(S+\alpha)$, and $i\mathcal{O}(T_1+\beta)$ instead of $i\mathcal{O}(S^{\perp}+\beta)$.

It is easy to see that any QPT distinguisher between G_2 and G_3 can be transformed into a QPT distinguisher between obfuscations of the original functions $S+\alpha, S^{\perp}+\beta$ and obfuscations of $T_0+\alpha, T_1+\beta$. By the subspace hiding property of indistinguishability obfuscators (Lemma 2.8), the success probabilities of G_2 and G_3 are thus negligibly close. Thus the successful probability of G_3 is at least $\varepsilon' - \mathsf{negl}(\lambda)$.

[17] We are running a classical function on a quantum input, which can be interpreted as running a classical function in superposition.

Game G_4: Lowering the need to fully know α, β in order to compute the obfuscations.

This hybrid is identical to G_3, with a modification in the way we check membership in each of the cosets: Let B_0 a basis for T_0, and B_1 a basis for $T_1^\perp$, and let $y_\alpha, y_\beta \in \{0,1\}^{\lambda-\lambda^\delta}$ defined as $y_\alpha := B_0 \cdot \alpha$ and $y_\beta := B_1 \cdot \beta$. $i\mathcal{O}(T_0 + \alpha)$ is changed to be an obfuscation of a circuit that for an input $u \in \{0,1\}^\lambda$ checks whether $B_0 \cdot u = y_\alpha$. $i\mathcal{O}(T_1 + \beta)$ is changed to be an obfuscation of a circuit that for an input $u \in \{0,1\}^\lambda$ checks whether $B_1 \cdot u = y_\beta$.

One can verify that the functionality of the obfuscated circuits $i\mathcal{O}(T_0+\alpha)$, $i\mathcal{O}(T_1+\beta)$ did not change, and thus by the security of the indistinguishability obfuscation schemes, the distributions are indistinguishable and the success probability of G_4 is $\varepsilon' - \mathsf{negl}(\lambda)$.

Game G_5: Changing the order of sampling the subspaces S, T_0, T_1.

This hybrid is identical to G_4, except that we change the order of the subspaces sampling process. In the previous hybrid, we sample a random $\frac{\lambda}{2}$-dimensional subspace $S \subseteq \mathbb{F}_2^\lambda$ then two random $(\lambda-\lambda^\delta)$-dimensional subspaces T_0, T_1 subjected to $T_1^\perp \subseteq S \subseteq T_0$. In the current hybrid, we first sample two random $(\lambda - \lambda^\delta)$-dimensional subspaces $T_0, T_1 \subseteq \mathbb{F}_2^n$ subjected to $T_1^\perp \subseteq T_0$, then sample a random $\frac{\lambda}{2}$-dimensional subspace $S \subseteq \mathbb{F}_2^n$ subjected to $T_1^\perp \subseteq S \subseteq T_0$.

Since the distribution of (S, T_0, T_1) in both hybrids are identical, the success probability of G_5 is $\varepsilon' - \mathsf{negl}(\lambda)$.

Game G_6: Fixing the subspace T_0, T_1.

In the subspace sampling process described in the previous hybrid, T_0 and T_1 are sampled before everything else. Thus we can perform an averaging argument on the sampling of T_0, T_1 to take the samples that maximize the success probability of the previous hybrid. Fix these samples of T_0, T_1 and define G_6 with respect to these samples. It is clear that the success probability of G_6 is $\varepsilon' - \mathsf{negl}(\lambda)$.

Game G_7: Losing the QFHE secret key.

This hybrid is identical to G_6 with one change: In step 6, when the verifier decrypts the QFHE classical part to get the Pauli keys α, β, the current hybrid does not decrypt to get α, β and instead it samples uniformly random $\alpha', \beta' \in \{0,1\}^\lambda$ and computes $y'_\alpha := B_0 \cdot \alpha', y'_\beta := B_1 \cdot \beta'$. The verifier then use these strings as y_α, y_β in the construction of the obfuscations $i\mathcal{O}(T_0 + \alpha)$, $i\mathcal{O}(T_1 + \beta)$, respectively.

We note that this change is only done for the specific coset instance under the consideration, for the other instances, the verifier still decrypts normally using the corresponding QFHE secret key.

Since α', β' are chosen uniformly at random, for fixed bases B_0, B_1, y'_α, y'_β are also uniformly random. Observe that conditioned on the probabilistic event $y'_\alpha = y_\alpha$ and $y'_\beta = y_\beta$ (for which to happen, the probability is exactly $2^{-2\lambda^\delta}$), the current and previous hybrids distribute identically. It follows that the success probability in G_7 is at least $2^{-2\lambda^\delta} \cdot (\varepsilon' - \mathsf{negl}(\lambda)) > 2^{-3\lambda^\delta}$.

Game G_8: Clearing all given knowledge on S and reducing to the original monogamy of entanglement game defined in Definition 5.3.

This hybrid is identical to G_7, except that we make two additional changes as follows.

- In the additional quantum communication round that we added after the end of Protocol 5.5 (see hybrid G_2), instead of sending back the original state $|\$\rangle$, the verifier send $|\hat{S}_{\hat{\alpha},\hat{\beta}}\rangle$. Recall that the coset $(\hat{S},\hat{\alpha},\hat{\beta})$ is the one the verifier sampled independently in step 8.
- In the step 2 in the monogamy game (Definition 5.12), when the challenger (i.e., the verifier) sends the description of the subspace S to both adversaries $\mathcal{B},\mathcal{C}$, it sends $\hat{S}$ instead.
- Consequently, the winning condition is changed to be that $\mathcal{B}$ outputs a vector in $\hat{S}+\hat{\alpha}$ and $\mathcal{C}$ outputs a vector in $\hat{S}^{\perp}+\hat{\beta}$.

We make few observations on the distribution in the current hybrid. First, in order to execute G_8, there is no need to know the secret key (corresponding to the coset instance under the consideration) of the QFHE scheme. However, one needs to care when invoking the semantic security of the QFHE, because even there is no need for the secret key, the adversary is still given a "predicate" check on the ciphertext, that is the obfuscation. Thus, to use the security of the QFHE, it is necessary to use two plaintexts such that the obfuscation evaluation on the ciphertext of these two plaintexts are identical. Our obfuscations $(P_{0,i},P_{1,i})$ were generated so that this condition is satisfied.

Secondly, the obfuscation distribution does not change from the description above, and we can see that in the previous hybrid, the adversary obtains a quantum one-time pad encryption of $|S\rangle$, while in the current hybrid, the adversary obtains a quantum one-time pad of $|\hat{S}\rangle$. More precisely, the adversary in the current hybrid receives an encryption of $|\hat{S}\rangle$ that is $|\hat{S}_{\hat{\alpha},\hat{\beta}}\rangle$ and an encryption of some Pauli keys (α,β) that are different from $(\hat{\alpha},\hat{\beta})$ with overwhelming probability. But because of the semantic security of QFHE.Encrypt (see Definition 2.23), this is indistinguishable from having $|\hat{S}_{\hat{\alpha},\hat{\beta}}\rangle$ and an actual encryption of $(\hat{\alpha},\hat{\beta})$.

From these observations, it follows that we can invoke the security of the QFHE to argue the indistinguishability of the current and previous hybrids, and in particular the indistinguishability between their success probabilities. Using the sub-exponential-advantage security of the QFHE, we have the success probability of G_8 is $> 2^{-3\lambda^{\delta}} - 2^{-2\lambda^{\delta'}} > 2^{-3\lambda^{\delta}-1}$.

At this point of the proof, we can reduce the success probability of an adversary in G_8 to the monogamy of entanglement game defined in Definition 5.3. We note that the coset game in Definition 5.3 can achieve sub-exponentially negligible security, say $2^{-4\lambda^{\delta}}$, if we assume sub-exponential security of the building blocks (i.e., the indistinguishability obfuscation scheme). Now, any QPT adversary of G_8 can be used to construct a QPT adversary for the coset game defined in Definition 5.3 as follows. Specifically, the reduction receives a challenge coset state $|\hat{S}_{\hat{\alpha},\hat{\beta}}\rangle$ and the obfuscated membership checking programs $i\mathcal{O}(\hat{S}+\hat{\alpha}), i\mathcal{O}(\hat{S}^{\perp}+\hat{\beta})$ from its challenger in the coset game in Definition 5.3. The reduction runs Protocol 5.5 with the adversary. Note that the reduction (playing the role of the verifier in Protocol 5.5) only needs $i\mathcal{O}(\hat{S}+\hat{\alpha})$ and $i\mathcal{O}(\hat{S}^{\perp}+\hat{\beta})$ to perfectly simulate the protocol with the adversary. Furthermore, it uses $|\hat{S}_{\hat{\alpha},\hat{\beta}}\rangle$ in the experiment described above instead of generating the state on its own, when it needs to send a coset state back to the adversary. When the reduction receives $\hat{S}$ from its challenger, it sends $\hat{S}$ to $\mathcal{B},\mathcal{C}$, and finally the reduction outputs whatever $\mathcal{B}$ and $\mathcal{C}$ output. (Formally, the reduction now consists of two non-communicating reductions, each interacts with $\mathcal{B}$ and

$\mathcal{C}$ respectively.) This is exactly in contradiction to strong monogamy of entanglement security as we presented above. □

5.6 Copy-Protection of Point Functions

In this section, we present a construction of (semi-quantum) copy-protection for point functions, as an example for demonstrating our semi-quantum copy-protection protocol (Protocol 5.5).

5.6.1 Anti-Piracy Security Definition

Recall that a point functions family $\{\mathsf{PF}_y\}_{y\in\mathcal{X}}$ is indexed by points $y \in \mathcal{X}$ and a point function PF_y returns 1 on input y and 0 on any other input.

Following [CMP20], we give a security definition for copy-protection for point functions by instantiating the general definition of copy-protection (see Section 2.3.9) with the following challenge distribution. We set D_f as the uniform distribution over $\{0,1\}^n$ and, for each function $f = \mathsf{PF}_y$, we set X_f as the distribution that returns:

- (x, y) with probability $\frac{1}{3}$;
- (y, x) with probability $\frac{1}{3}$;
- (x, x') with probability $\frac{1}{3}$;

where x, x' are sampled uniformly at random.

These distributions yield $p^{trivial}_{D_f,\{X_f\}_{f\in\mathcal{F}}} = 2/3$. For a clarity purpose, we will use the notations D_y instead of D_f and X_y instead of X_f.

5.6.2 Construction

Let $\{\mathsf{PF}_y\}_{y\in\{0,1\}^n}$ be the family to be copy-protected, where $n := n(\lambda)$ is a polynomial in λ. We define ℓ_0, ℓ_1, ℓ_2 such that $n = \ell_0 + \ell_1 + \ell_2$ and $\ell_2 - \ell_0 \geq$ is large enough. For this construction, we need three pseudorandom functions (PRFs):

- A puncturable extracting PRF (Definition 2.10) $\mathsf{PRF}_1 : \mathcal{K}_1 \times \{0,1\}^n \to \{0,1\}^m$ with error $2^{-\lambda-1}$, where m is a polynomial in λ and $n \geq m + 2\lambda + 4$;
- A puncturable injective PRF (Definition 2.9) $\mathsf{PRF}_2 : \mathcal{K}_2 \times \{0,1\}^{\ell_2} \to \{0,1\}^{\ell_1}$ with failure probability $2^{-\lambda}$;
- A puncturable PRF $\mathsf{PRF}_3 : \mathcal{K}_3 \times \{0,1\}^{\ell_1} \to \{0,1\}^{\ell_2}$.

Construction 5.1 — Copy-Protection of Point Functions

$\rho_y \leftarrow \mathsf{PF.Protect}(y)$:

- Sample $\left(\{A_i, s_i, s'_i\}_{i\in[\![1,\ell_0]\!]}, \{|A_{i,s_i,s'_i}\rangle\}_{i\in[\![1,\ell_0]\!]}, \{(\mathsf{R}^0_i, \mathsf{R}^1_i)\}_{i\in[\![1,\ell_0]\!]}\right) \leftarrow \mathsf{SampleCoset}(1^\lambda)$, where SampleCoset is described in Figure 5.1.
- Sample PRF keys k_i for PRF_i with $i \in \{1,2,3\}$.
- Prepare the program $\hat{\mathsf{P}} \leftarrow i\mathcal{O}(\mathsf{P})$, where P is described in Figure 5.2.

- Compute $z := \mathsf{PRF}_1(\mathsf{k}_1, y)$.
- Return $\rho_y := \left(\{|A_{i,s_i,s_i'}\rangle\}_{i\in[\![1,\ell_0]\!]}, \hat{\mathsf{P}}, z\right)$.

$m \leftarrow \mathsf{PF.Eval}(\rho_y, x)$:

- Parse $\rho_y = \left(\{|A_{i,s_i,s_i'}\rangle\}_{i\in[\![1,\ell_0]\!]}, \hat{\mathsf{P}}, z\right)$.
- Parse x as $x := x_0\|x_1\|x_2$.
- For each $i \in [\![1,\ell_0]\!]$, if $x_{0,i} = 1$, apply $\mathsf{H}^{\otimes n}$ to $|A_{i,u_i,u_i'}\rangle$; if $x_{0,i} = 0$, leave the state unchanged.
- Let σ be the resulting state (which can be interpreted as a superposition over tuples of ℓ_0 vectors). Run $\hat{\mathsf{P}}$ coherently on input x and σ, and measure the final output register to obtain z'.
- Return 1 if $z' = z$, otherwise return 0.

The semi-quantum copy-protection scheme for point functions is presented in Construction 5.2.

Construction 5.2 — Semi-Quantum Copy-Protection of Point Functions

$\mathsf{PF.Protect}(y)$: $\mathsf{PF.Protect}(y)$ is now an interactive protocol between a sender and a receiver. The sender does the following:

- Run Protocol 5.5 ℓ_0 times with the receiver to obtain $\left(\{A_i, s_i, s_i'\}_{i\in[\![1,\ell_0]\!]}, \{(\mathsf{R}_i^0, \mathsf{R}_i^1)\}_{i\in[\![1,\ell_0]\!]}\right)$. The receiver obtains the corresponding $\{|A_{i,s_i,s_i'}\rangle\}_{i\in[\![1,\ell_0]\!]}$.
- Sample PRF keys k_i for PRF_i with $i \in \{1,2,3\}$.
- Prepare the program $\hat{\mathsf{P}} \leftarrow i\mathcal{O}(\mathsf{P})$, where P is described in Figure 5.2.
- Compute $z := \mathsf{PRF}_1(\mathsf{k}_1, y)$.
- Send $\left(\hat{\mathsf{P}}, z\right)$ to the receiver.

$m \leftarrow \mathsf{PF.Eval}(\rho_y, x)$:

- Parse $\rho_y = \left(\{|A_{i,s_i,s_i'}\rangle\}_{i\in[\![1,\ell_0]\!]}, \hat{\mathsf{P}}, z\right)$.
- Parse x as $x := x_0\|x_1\|x_2$.
- For each $i \in [\![1,\ell_0]\!]$, if $x_{0,i} = 1$, apply $\mathsf{H}^{\otimes n}$ to $|A_{i,u_i,u_i'}\rangle$; if $x_{0,i} = 0$, leave the state unchanged.
- Let σ be the resulting state (which can be interpreted as a superposition over tuples of ℓ_0 vectors). Run $\hat{\mathsf{P}}$ coherently on input x and σ, and measure the final output register to obtain z'.
- Return 1 if $z' = z$, otherwise return 0.

Theorem 5.3. *Assuming the existence of post-quantum indistinguishability obfusca-*

- Sample ℓ_0 subspaces of $\mathbb{F}_2^n$ of dimension $n/2$ $\{A_i\}_{i\in[\![1,\ell_0]\!]}$.
- For each coset A_i, sample two vectors $s_i, s'_i \leftarrow \{0,1\}^n$.
- Prepare the ℓ_0 coset states $\{|A_{i,s_i,s'_i}\rangle\}_{i\in[\![1,\ell_0]\!]}$.
- For each coset state $|A_{i,s_i,s'_i}\rangle$, prepare the obfuscated membership programs $\mathsf{R}_i^0 = i\mathcal{O}(A_i + s_i)$ and $\mathsf{R}_i^1 = i\mathcal{O}(A_i^\perp + s'_i)$.
- Return $\Big(\{A_i, s_i, s'_i\}_{i\in[\![1,\ell_0]\!]}, \{|A_{i,s_i,s'_i}\rangle\}_{i\in[\![1,\ell_0]\!]}, \{(\mathsf{R}_i^0, \mathsf{R}_i^1)\}_{i\in[\![1,\ell_0]\!]}\Big)$.

Figure 5.1: SampleCoset procedure.

Hardcoded: Keys $(\mathsf{k}_1, \mathsf{k}_2, \mathsf{k}_3) \in \mathcal{K}_1 \times \mathcal{K}_2 \times \mathcal{K}_3$, programs $\mathsf{R}_i^0, \mathsf{R}_i^1$ for all $i \in [\![1, \ell_0]\!]$.
On input $x = x_0\|x_1\|x_2$ and vectors $v_0, v_1, \cdots, v_{\ell_0}$ where each $v_i \in \mathbb{F}_2^n$, do the following:

1. (**Hidden Trigger Mode**) If $\mathsf{PRF}_3(\mathsf{k}_3, x_1) \oplus x_2 = x_0\|\mathsf{Q}'$ and $x_1 = \mathsf{PRF}_2(\mathsf{k}_2, x_0\|\mathsf{Q}')$: treat Q' as a classical circuit and output $\mathsf{Q}'(v_1, \cdots, v_{\ell_0})$.
2. (**Normal Mode**) If for all $i \in [\![1, \ell_0]\!]$, $\mathsf{R}_i^{x_i}(v_i) = 1$, then output $\mathsf{PRF}_1(\mathsf{k}_1, x)$. Otherwise, output $\perp$.

Figure 5.2: Program P.

tion, one-way functions, and compute-and-compare obfuscation for the class of unpredictable distributions, the scheme of Construction 5.1 and Construction 5.2 have correctness *and* anti-piracy security*.*

The correctness of our protocols follows directly from the correctness of the copy-protection of PRF's construction of [CLLZ21, Lemma 7.13].

The security of our protocols rely on a new security notion for single-decryptors, which is a cryptographic primitive first defined in [CLLZ21]. We recall the definition of single-decryptors and introduce this new security notion, which we call Real-or-Random CPA anti-piracy in Section 5.6.3. We show that the [CLLZ21]'s single-decryptor's construction also achieves this new security definition. The security proof of our constructions then follow the same strategy as that of copy-protection for PRFs given in [CLLZ21], except that we reduce security to our new single-decryptor definitions. We refer the reader to Section 5.6.4 for a detailed proof. Indeed, the reductions of security from our constructions (Construction 5.1 and Construction 5.2) are identical, except that the latter reduces to a semi-quantum version of the single-decryptor scheme that we present in Section 5.6.3. Thus we only include the proof for Construction 5.1 in Section 5.6.4.

5.6.3 Single-Decryptors

In this section, we present the definition of single-decryptors, as defined in [CLLZ21]. We also introduce a new security property for single-decryptors, namely anti-piracy security of single-decryptors in the real-or-random style. A variant of semi-quantum single-decryptors will be also introduced.

Definition

Definition 5.13 — Single-Decryptor Encryption Scheme [CLLZ21]

A single-decryptor encryption scheme is a tuple of algorithms $\mathcal{E} = \langle \mathsf{Setup}, \mathsf{QKeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt} \rangle$ with the following properties:

$(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Setup}(1^\lambda)$. On input a security parameter λ, the classical setup algorithm Setup outputs a classical secret key sk and a public key pk.

$(\rho_{\mathsf{sk}}) \leftarrow \mathsf{QKeyGen}(\mathsf{sk})$. On input a classical secret key sk, the quantum key generation algorithm QKeyGen outputs a quantum secret key ρ_{sk}.

$y \leftarrow \mathsf{Encrypt}(\mathsf{pk}, x)$. On input a public key pk, a message x in the message space $\mathcal{M}$, the classical encryption algorithm Encrypt outputs a classical ciphertext y.

$x/\perp \leftarrow \mathsf{Decrypt}(\rho_{\mathsf{sk}}, y)$. On input a quantum secret key ρ_{sk}, a classical ciphertext y, the quantum decryption algorithm Decrypt outputs a classical message x or a decryption failure symbol $\perp$.

Correctness. There exists a negligible function $\mathsf{negl}(\cdot)$, such that for all $\lambda \in \mathbb{N}$, for all $x \in \mathcal{M}$, the following holds:

$$\Pr\left[\begin{array}{c|c} \mathsf{Decrypt}(\rho_{\mathsf{sk}}, y) = x & \begin{array}{c}(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Setup}(1^\lambda) \\ \rho_{\mathsf{sk}} \leftarrow \mathsf{QKeyGen}(\mathsf{sk}) \\ y \leftarrow \mathsf{Encrypt}(\mathsf{pk}, x)\end{array}\end{array}\right] \geq 1 - \mathsf{negl}(\lambda).$$

Note that correctness implies that a honestly generated quantum decryption key can be used to decrypt correctly polynomially many times, from the gentle measurement lemma [Wil11].

Anti-Piracy Game of Single-Decryptor (Real-or-Random Style)

We present below an anti-piracy game of single-decryptors in the real-or-random CPA style, parameterized by a single-decryptor scheme $\mathcal{E} = \langle \mathsf{Setup}, \mathsf{QKeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt}\rangle$, a security parameter λ. This game is between a challenger and an adversary represented by three QPT algorithms $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$.

- **Setup phase:**
  - The challenger samples $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Setup}(1^\lambda)$.
  - The challenger samples $\rho_{\mathsf{sk}} \leftarrow \mathsf{QKeyGen}(\mathsf{sk})$.
  - The challenger sends $(\mathsf{pk}, \rho_{\mathsf{sk}})$ to $\mathcal{A}_0$.
- **Splitting phase:**
  - $\mathcal{A}_0$ prepares a bipartite quantum state σ_{12}.
  - $\mathcal{A}_0$ sends σ_1 to $\mathcal{A}_1$ and σ_2 to $\mathcal{A}_2$.
  - $\mathcal{A}_0$ sends a challenge message m to the challenger.
- **Challenge phase:**
  - The challenger samples two uniformly random messages (m', m'').
  - The challenger then generates ciphertexts c_1, c_2 as follows.
    - $c_1 = \mathsf{Encrypt}(\mathsf{pk}, z)$ and $c_2 = \mathsf{Encrypt}(\mathsf{pk}, z')$ with probability $1/3$. Set $b_1 = 0$ and $b_2 = 1$.
    - $c_1 = \mathsf{Encrypt}(\mathsf{pk}, z')$ and $c_2 = \mathsf{Encrypt}(\mathsf{pk}, z)$ with probability $1/3$. Set $b_1 = 1$ and $b_2 = 0$.
    - $c_1 = \mathsf{Encrypt}(\mathsf{pk}, z')$ and $c_2 = \mathsf{Encrypt}(\mathsf{pk}, z'')$ with probability $1/3$. Set $b_1 = 1$ and $b_2 = 1$.
  - The challenger sends c_1 to $\mathcal{A}_1$ and c_2 to $\mathcal{A}_2$.
- **Answer phase:**
  - For $i \in \{1, 2\}$: $\mathcal{A}_i$ outputs a bit b'_i.

The adversary wins the game if $\mathcal{A}_1$ and $\mathcal{A}_2$ both make a correct guess, that is $b'_i = b_i$ for $i \in \{1, 2\}$.

We denote the random variable that indicates whether an adversary $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ wins the game or not as $\mathsf{SD\text{-}AP\text{-}RoR}^{\mathcal{E}}_D(1^\lambda, (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2))$.

Definition 5.14 — Anti-Piracy Security, Real-or-Random style

A single-decryptor scheme has *anti-piracy security (real-or-random style)* if no QPT adversary $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ can win the anti-piracy game (real-or-random style) with a probability significantly greater than $1/2$. More precisely, for any QPT adversary $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$

$$\Pr\left[\mathsf{SD\text{-}AP\text{-}RoR}_D^{\mathcal{E}}(1^\lambda, (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)) = 1\right] \leq 1/2 + \mathsf{negl}(\lambda).$$

We observe that the construction of single-decryptor given in [CLLZ21] also satisfies our definition of anti-piracy in the real-or-random style. For completeness, we recall their construction below.

Construction 5.3 — [CLLZ21]'s Single-Decryptor Scheme

Given a security parameter λ, let $n = \lambda$ and κ be polynomial in λ.

$(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Setup}(1^\lambda)$:

- Sample coset spaces $\{A_i, s_i, s_i'\}_{i\in[\![1,\kappa]\!]}$ where each A_i is of dimension $n/2$;
- Construct the membership programs for each coset $\{\mathsf{R}_i^0, \mathsf{R}_i^1\}_{i\in[\![1,\kappa]\!]}$;
- Return $\left(\mathsf{sk} := \{A_i, s_i, s_i'\}_{i\in[\![1,\kappa]\!]}, \mathsf{pk} := \{\mathsf{R}_i^0, \mathsf{R}_i^1\}_{i\in[\![1,\kappa]\!]}\right)$.

$\rho_{\mathsf{sk}} \leftarrow \mathsf{QKeyGen}(\mathsf{sk})$:

- Parse $\mathsf{sk} \leftarrow \{A_i, s_i, s_i'\}_{i\in[\![1,\kappa]\!]}$;
- Return $\{|A_{i,s_i,s_i'}\rangle\}_{i\in[\![1,\kappa]\!]}$.

$c \leftarrow \mathsf{Encrypt}(\mathsf{pk}, m)$:

- Parse $\mathsf{pk} \leftarrow \{\mathsf{R}_i^0, \mathsf{R}_i^1\}_{i\in[\![1,\kappa]\!]}$;
- Sample $r \xleftarrow{\$} \{0,1\}^\kappa$;
- Generate an obfuscated program $i\mathcal{O}(\mathsf{Q}_{m,r})$ of program $\mathsf{Q}_{m,r}$ described in Section 5.6.3.
- Return $c := (r, i\mathcal{O}(\mathsf{Q}_{m,r}))$.

$m/\perp \leftarrow \mathsf{Decrypt}(\rho_{\mathsf{sk}}, c)$:

- Parse $\rho_{\mathsf{sk}} \leftarrow \{|A_{i,s_i,s_i'}\rangle\}_{i\in[\![1,\kappa]\!]}$ and $c \leftarrow (r, i\mathcal{O}(\mathsf{Q}_{m,r}))$;
- For all $i \in [\![1,\kappa]\!]$, if $r_i = 1$, apply $\mathsf{H}^{\otimes n}$ to $|A_{i,s_i,s_i'}\rangle$;
- Let ρ_{sk}' be the resulting state, run $i\mathcal{O}(\mathsf{Q}_{m,r})$ coherently on ρ_{k}' and measure the final register to get m;
- Return m.

Hardcoded: Keys $\mathsf{k}_1, \mathsf{k}_2, \mathsf{k}_3$, programs $\mathsf{R}_i^0, \mathsf{R}_i^1$ for all $i \in [\![1, \kappa]\!]$.
On input vectors $u_1, u_2, \ldots, u_\kappa$, do the following:

1. If for all $i \in [\![1, \kappa]\!]$, $\mathsf{R}_i^{r_i}(u_i) = 1$, then output m.
2. Otherwise, output $\perp$.

Figure 5.3: Program $\mathsf{Q}_{m,r}$.

Semi-Quantum Single-Decryptor

Alternatively, in the definition of single-decryptors above, we can combine the Setup and QKeyGen algorithms to be a single interactive protocol with classical communication. The security definition is defined analogously, in which the setup phase is now an interactive setup phase where the challenger obtains the the secret key and the adversary obtains the quantum unclonable secret key. This defines a notion of semi-quantum single-decryptors.

Remark 5.5. Of course, now if the sender wants to generate a new quantum secret key, it needs to run the interactive protocol again, which effectively also generates a new classical secret key sk and a new classical public key pk. To recover the original setting where there are only one classical secret/public key pair and possibly many quantum secret keys, the sender can use any post-quantum semantic-secure public-key encryption scheme to encrypt the new classical secret key sk generated by the semi-quantum protocol, and send this encryption of sk to the receiver. This encryption of sk will also be included in the ciphertext, which the sender can decrypt using its "master" secret key and perform the original decryption algorithm. We note that for our construction of semi-quantum copy-protection, this is not necessary though.

A construction of semi-quantum single-decryptors is identical to Construction 5.3, except now we replace the Setup and QKeyGen algorithms by polynomially many runs of Protocol 5.5. Security proof of Construction 5.3 also carries over this semi-quantum setting directly, with only a small change as follows. In the reduction showing that an adversary $\mathcal{A}$ that breaks the anti-piracy game of single-decryptors can be used to construct an adversary $\mathcal{A}'$ breaking the monogamy of entanglement game (defined in Definition 5.12), $\mathcal{A}'$ simulates the security game for $\mathcal{A}$ (in which $\mathcal{A}'$ runs polynomially many executions of Protocol 5.5 with $\mathcal{A}$), $\mathcal{A}'$ then picks one execution uniformly at random and lets $\mathcal{A}$ runs the protocol with $\mathcal{A}'$'s challenger. The rest of the reduction is identical as the one given in [CLLZ21], we omit the full details here.

5.6.4 Proof of Anti-Piracy Security of Construction 5.1

We now prove the anti-piracy security of our copy-protection of points functions scheme. For easy of reading, we present here the proof for the quantum protocol, but we note that the same proof can be used for the semi-quantum protocol.

We first define the GenTrigger procedure (see Figure 5.4) which, given an input's prefix x_0 and a PRF image y returns a so-called *trigger input* x' that: passes the "Hidden Trigger" condition of the program P. The following lemma follows from [CLLZ21].

Given as input $x_0 \in \{0,1\}^{\ell_0}$, $z \in \{0,1\}^m$, $\mathsf{k}_2, \mathsf{k}_3 \in \mathcal{K}_2 \times \mathcal{K}_3$ and cosets $\{A_i, s_i, s'_i\}_{i\in[\![1,\ell_0]\!]}$:

- Let Q be the program which, given $v_0, \ldots, v_{\ell_0}$, returns z if $R_i^{x_{0,i}}(v_i) = 1$ for all i or $\perp$ otherwise.
- $x'_1 \leftarrow \mathsf{PRF}_2(\mathsf{k}_2, x_0 \| \mathsf{Q})$;
- $x'_2 \leftarrow \mathsf{PRF}_3(\mathsf{k}_3, x'_1) \oplus (x_0 \| \mathsf{Q})$;
- Return $x_0 \| x'_1 \| x'_2$.

Figure 5.4: GenTrigger procedure.

Lemma 5.13 ([CLLZ21, Lemma 7.17]). *Assuming post-quantum $i\mathcal{O}$ and one-way functions, any efficient* QPT *algorithm $\mathcal{A}$ cannot win the following game with non-negligible advantage:*

- *A challenger samples $\mathsf{k}_1 \leftarrow \mathsf{Setup}(1^\lambda)$ and prepares a quantum key $\rho_\mathsf{k} := (\{|A_{i,s_i,s'_i}\rangle\}_{i\in[\![1,l_0]\!]}, i\mathcal{O}(\mathsf{P}))$ (recall that P has keys $\mathsf{k}_1, \mathsf{k}_2, \mathsf{k}_3$ hardcoded).*
- *The challenger then samples a random input $x \leftarrow \{0,1\}^n$. Let $z \leftarrow \mathsf{PRF}_1(\mathsf{k}_1, x)$.*
- *The challenger samples challenges x_1, x_2 according to the following distribution:*
  - *$x_1 := x$ and $x_2 \xleftarrow{\$} \{0,1\}^n$ with probability $1/3$;*
  - *$x_1 \xleftarrow{\$} \{0,1\}^n$ and $x_2 := x$ with probability $1/3$;*
  - *$x_1, x_2 \xleftarrow{\$} \{0,1\}^n$ with probability $1/3$;*
- *The challenger parses the inputs x_i as $x_i := x_{i,0}||x_{i,1}||x_{i,2}$ for $i \in \{1,2\}$. Let $x'_i \leftarrow \mathsf{GenTrigger}(x_{i,0}, z_1, \mathsf{k}_2, \mathsf{k}_3, \{A_j, s_j, s'_j\}_{j\in[\![1,\ell_0]\!]})$ for $i \in \{1,2\}$.*
- *The challenger flips a coin b, and sends either x_1, x_2 or x'_1, x'_2 to respectively Bob and Charlie, depending on the value of the coin. $\mathcal{A}$ wins if it guesses b.*

We are now ready to prove the anti-piracy security of our Construction 5.1. We proceed with the proof via a sequence of hybrids. For any pair of hybrids (G_i, G_j), we say that G_i is *negligibly close to* G_j if for every QPT adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$, the probability that $\mathcal{A}$ wins in G_i is negligibly close to the probability that it wins in G_j. For the sake of simplicity, we denote the uniform distribution over $\{(0,1),(1,0),(1,1)\}$ as $D_{1/3}$.

Game G_0: This is the original anti-piracy game of our copy-protection scheme.

(a) **Setup phase**

- The challenger samples $\left(\{A_i, s_i, s'_i\}_{i\in[\![1,\ell_0]\!]}, \{|A_{i,s_i,s'_i}\rangle\}_{i\in[\![1,\ell_0]\!]}, \{(\mathsf{R}_i^0, \mathsf{R}_i^1)\}_{i\in[\![1,\ell_0]\!]}\right) \leftarrow \mathsf{SampleCoset}(1^\lambda)$.
- The challenger samples $\mathsf{k}_i \leftarrow \mathsf{PRF}_i.\mathsf{KeyGen}(1^\lambda)$ for $i \in \{1,2,3\}$.

- The challenger prepares the program $\hat{\mathsf{P}} \leftarrow i\mathcal{O}(\mathsf{P})$.
- The challenger samples a random point $y \in \{0,1\}^n$.
- The challenger computes $z := \mathsf{PRF}_1(\mathsf{k}_1, y)$.
- The challenger sends $\rho_y := \left(\{|A_{i,s_i,s'_i}\rangle\}_{i\in[\![1,\ell_0]\!]}, \hat{\mathsf{P}}, z\right)$ to $\mathcal{A}_0$.

(b) **Splitting phase**

- $\mathcal{A}_0$ prepares a (possibly entangled) bipartite state σ_{12} and sends σ_1 to $\mathcal{A}_1$ and σ_2 to $\mathcal{A}_2$.

(c) **Challenge phase**

- The challenger samples $(b_1, b_2) \xleftarrow{\$} D_{1/3}$.
- For $i \in \{1,2\}$:
  - If $b_i = 0$: the challenger sets $x_i := y$.
  - Otherwise, the challenger samples $x_i \leftarrow \{0,1\}^n$.
- The challenge sends x_1 to $\mathcal{A}_1$ and x_2 to $\mathcal{A}_2$.

(d) **Answer phase**

- $\mathcal{A}_1$ returns b'_1 and $\mathcal{A}_2$ returns b'_2.
- The adversary wins if $b'_1 = b_1$ and $b'_2 = b_2$.

Game G_1: In this game, we replace x_1, x_2 by trigger inputs. The trigger's inputs lemma (Lemma 5.13) implies that G_1 is negligibly close to G_0.

(a) **Setup phase**

- The challenger samples $\left(\{A_i, s_i, s'_i\}_{i\in[\![1,\ell_0]\!]}, \{|A_{i,s_i,s'_i}\rangle\}_{i\in[\![1,\ell_0]\!]}, \{(\mathsf{R}^0_i, \mathsf{R}^1_i)\}_{i\in[\![1,\ell_0]\!]}\right) \leftarrow \mathsf{SampleCoset}(1^\lambda)$.
- The challenger samples $\mathsf{k}_i \leftarrow \mathsf{PRF}_i.\mathsf{KeyGen}(1^\lambda)$ for $i \in \{1,2,3\}$.
- The challenger prepares the program $\hat{\mathsf{P}} \leftarrow i\mathcal{O}(\mathsf{P})$.
- The challenger samples a random point $y \in \{0,1\}^n$.
- The challenger computes $z := \mathsf{PRF}_1(\mathsf{k}_1, y)$.
- The challenger sends $\rho_y := \left(\{|A_{i,s_i,s'_i}\rangle\}_{i\in[\![1,\ell_0]\!]}, \hat{\mathsf{P}}, z\right)$ to $\mathcal{A}_0$.

(b) **Splitting phase**

- $\mathcal{A}_0$ prepares a (possibly entangled) bipartite state σ_{12} and sends σ_1 to $\mathcal{A}_1$ and σ_2 to $\mathcal{A}_2$.

(c) **Challenge phase**

- The challenger samples $(b_1, b_2) \xleftarrow{\$} D_{1/3}$.
- For $i \in \{1,2\}$:
  - If $b_i = 0$: the challenger sets $x_i := y$ and $z_i := z$.
  - Otherwise, the challenger samples $x_i \leftarrow \{0,1\}^n$ and $z_i \leftarrow \{0,1\}^m$.
  - In both case, the challenger computes $x'_i \leftarrow \mathsf{GenTrigger}(x_{i,0}, z_i, \mathsf{k}_2, \mathsf{k}_3, \{A_{i,s_i,s'_i}\}_{i\in[\![1,\ell_0]\!]})$.
- The challenge sends x'_1 to $\mathcal{A}_1$ and x'_2 to $\mathcal{A}_2$.

(d) **Answer phase**

- $\mathcal{A}_1$ returns b'_1 and $\mathcal{A}_2$ returns b'_2.

- The adversary wins if $b'_1 = b_1$ and $b'_2 = b_2$.

Game G_2: In this game, we replace z by a random string and change the challenges accordingly. Because PRF_1 is extracting, G_2 is negligibly close to G_1.

(a) **Setup phase**

- The challenger samples $\left(\{A_i, s_i, s'_i\}_{i\in[\![1,\ell_0]\!]}, \{|A_{i,s_i,s'_i}\rangle\}_{i\in[\![1,\ell_0]\!]}, \{(\mathsf{R}^0_i, \mathsf{R}^1_i)\}_{i\in[\![1,\ell_0]\!]}\right) \leftarrow \mathsf{SampleCoset}(1^\lambda)$.
- The challenger samples $\mathsf{k}_i \leftarrow \mathsf{PRF}_i.\mathsf{KeyGen}(1^\lambda)$ for $i \in \{1,2,3\}$.
- The challenger prepares the program $\hat{\mathsf{P}} \leftarrow i\mathcal{O}(\mathsf{P})$.
- The challenger samples a random point $y \in \{0,1\}^n$.
- The challenger samples $z \leftarrow \{0,1\}^m$.
- The challenger sends $\rho := \left(\{|A_{i,s_i,s'_i}\rangle\}_{i\in[\![1,\ell_0]\!]}, \hat{\mathsf{P}}, z\right)$ to $\mathcal{A}_0$.

(b) **Splitting phase**

- $\mathcal{A}_0$ prepares a (possibly entangled) bipartite state σ_{12} and sends σ_1 to $\mathcal{A}_1$ and σ_2 to $\mathcal{A}_2$.

(c) **Challenge phase**

- The challenger samples $(b_1, b_2) \xleftarrow{\$} D_{1/3}$.
- For $i \in \{1,2\}$:
  - If $b_i = 0$: the challenger sets $x_i := y$ and $z_i := z$.
  - Otherwise, the challenger samples $x_i \leftarrow \{0,1\}^n$ and $z_i \leftarrow \{0,1\}^m$.
  - In both case, the challenger computes $x'_i \leftarrow \mathsf{GenTrigger}(x_{i,0}, z_i, \mathsf{k}_2, \mathsf{k}_3, \{A_{i,s_i,s'_i}\}_{i\in[\![1,\ell_0]\!]})$.
- The challenge sends x'_1 to $\mathcal{A}_1$ and x'_2 to $\mathcal{A}_2$.

(d) **Answer phase**

- $\mathcal{A}_1$ returns b'_1 and $\mathcal{A}_2$ returns b'_2.
- The adversary wins if $b'_1 = b_1$ and $b'_2 = b_2$.

Game G_3: In this game, we recast the experiment using the notation of single-decryptor. The probability of winning this game is the same as for the previous one.

(a) **Setup phase**

- The challenger samples $\left(\{A_{i,s_i,s'_i}\}_{i\in[\![1,\ell_0]\!]}, \rho_{\mathsf{sk}}, \mathsf{pk}\right) \leftarrow \mathsf{SampleCoset}(1^\lambda)$.
- The challenger samples $\mathsf{k}_i \leftarrow \mathsf{PRF}_i.\mathsf{KeyGen}(1^\lambda)$ for $i \in \{1,2,3\}$.
- The challenger prepares the program $\hat{\mathsf{P}} \leftarrow i\mathcal{O}(\mathsf{P})$.
- The challenger samples a random point $y \in \{0,1\}^n$.
- The challenger samples $z \leftarrow \{0,1\}^m$.
- The challenger sends $\rho := \left(\rho_{\mathsf{sk}}, \hat{\mathsf{P}}, z\right)$ to $\mathcal{A}_0$.

(b) **Splitting phase**

- $\mathcal{A}_0$ prepares a (possibly entangled) bipartite state σ_{12} and sends σ_1 to $\mathcal{A}_1$ and σ_2 to $\mathcal{A}_2$.

(c) **Challenge phase**

- The challenger samples $(b_1, b_2) \xleftarrow{\$} D_{1/3}$.
- For $i \in \{1, 2\}$:
  - If $b_i = 0$: the challenger sets $z_i := z$.
  - Otherwise, the challenger samples $z_i \leftarrow \{0,1\}^m$.
  - In both case, the challenger computes $(x_i, Q) \leftarrow \mathsf{SD.Encrypt}(\mathsf{pk}, z_i)$.
  - Finally, the challenger computes x'_i as in GenTrigger using $x_{i,0}$ and Q.
- The challenge sends x'_1 to $\mathcal{A}_1$ and x'_2 to $\mathcal{A}_2$.

(d) **Answer phase**

- $\mathcal{A}_1$ returns b'_1 and $\mathcal{A}_2$ returns b'_2.
- The adversary wins if $b'_1 = b_1$ and $b'_2 = b_2$.

Reduction to Single-Decryptor's Anti-Piracy Game. Assume that there exists a QPT adversary $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ who wins G_3 with advantage δ. We construct an adversary $(\mathcal{B}_0, \mathcal{B}_1, \mathcal{B}_2)$ who wins the anti-piracy game, real-or-random style, of the single-decryptor scheme with the same advantage δ.

1. **Setup phase**
   - The challenger samples $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Setup}(1^\lambda)$.
   - The challenger prepares $\rho_{\mathsf{sk}} \leftarrow \mathsf{SD.QKeyGen}(\mathsf{sk})$.
   - The challenger sends $(\mathsf{pk}, \rho_{\mathsf{sk}})$ to $\mathcal{B}_0$.
2. **Splitting phase**
   - $\mathcal{B}_0$ samples three keys $\mathsf{k}_1, \mathsf{k}_2, \mathsf{k}_3$ and an image $z \leftarrow \{0,1\}^m$.
   - $\mathcal{B}_0$ constructs the program P (based on $\mathsf{k}_1, \mathsf{k}_2, \mathsf{k}_3, \mathsf{pk}$) as in Protect and sets $\hat{\mathsf{P}} \leftarrow i\mathcal{O}(\mathsf{P})$.
   - $\mathcal{B}_0$ sends $(\rho_{\mathsf{sk}}, \hat{\mathsf{P}}, z)$ to $\mathcal{A}_0$ to get σ_{12}.
   - $\mathcal{B}_0$ sends $\sigma'_1 := (\sigma_1, \mathsf{k}_2, \mathsf{k}_3)$ to $\mathcal{B}_1$, $\sigma'_2 := (\sigma_2, \mathsf{k}_2, \mathsf{k}_3)$ to $\mathcal{B}_2$ and z as the challenge message to the challenger.
3. **Challenge phase:**
   - The challenger samples two uniformly random messages (z', z'') and two ciphertexts (c_1, c_2) as follows.
     - $c_1 = \mathsf{Encrypt}(\mathsf{pk}, z)$ and $c_2 = \mathsf{Encrypt}(\mathsf{pk}, z')$ with probability $1/3$.
     - $c_1 = \mathsf{Encrypt}(\mathsf{pk}, z')$ and $c_2 = \mathsf{Encrypt}(\mathsf{pk}, z)$ with probability $1/3$.
     - $c_1 = \mathsf{Encrypt}(\mathsf{pk}, z')$ and $c_2 = \mathsf{Encrypt}(\mathsf{pk}, z'')$ with probability $1/3$.
   - The challenger sends c_1 to $\mathcal{B}_1$ and c_2 to $\mathcal{B}_2$.
4. **Answer phase**
   - For $i \in \{1, 2\}$:
     - $\mathcal{B}_i$, given (σ'_i, c_i), parse c_i as (x_0, Q), then prepares $x' = (x_0 || x'_1 || x'_2)$ as in GenTrigger: $x'_1 := \mathsf{PRF}_2(\mathsf{k}_2, x_0 || Q)$ and $x'_2 := \mathsf{PRF}_3(\mathsf{k}_3, x'_1) \oplus (x_0 || Q)$.
     - $\mathcal{B}_i$ runs $\mathcal{A}_i$ on (σ_i, x'_i) and returns the outcome b'_i.

The adversary $(\mathcal{B})$ perfectly simulates $\mathcal{A}$, and thus $\mathcal{B}$ breaks the anti-piracy security of the single-decryptor scheme with the same probability δ, which completes the proof.

Appendix

A Tokenized Digital Signatures

We report in this chapter a construction of tokenized digital signatures with strong unforgeability security. The notion of quantum tokens for digital signatures was initiated by Ben-David and Sattath [BS17]. In a tokenized digital signature scheme, a signer who gets one copy of the signing token sig can sign a single bit b using a QPT algorithm $\mathsf{Sign}(b, sig)$ whose output is a classical signature. The correctness guarantees that the verification will accept the result as a signature on b. We note that the signing algorithm is a unitary and will produce a superposition of all valid signatures of b; to obtain a classical signature, a destructive measurement to the state is necessary which leads to a collapse of the token state. Thus, a signing token sig can only be used to produce one classical signature of a single bit and any attempt to produce a classical signature of the other bit would fail.

Ben-David and Sattath [BS17] show how to construct a strongly unforgeable tokenized digital signature scheme relative to a classical oracle (a subspace membership oracle). This construction is then successfully instantiated in the plain model due to Coladangelo *et al.* in [CLLZ21], albeit their construction is only weakly unforgeable. In this chapter, we improve [CLLZ21]'s results and show that their construction is indeed strongly unforgeable. [CLLZ21] construction is based on hidden coset states, whose security is reduced to a computational direct product hardness of coset states. We then show another version of the direct product hardness of coset states, which allows us to show strong unforgeability for [CLLZ21]'s construction. Our contribution is thus the proof, not the construction. This is a joint work with Thomas Vidick in 2021, in a failed attempt to construct non-interactive zero-knowledge proof systems for any language in QMA, the quantum analog of the class NP.

Before going into details, we give some intuition behind the construction of weakly unforgeable tokenized digital signatures in [CLLZ21], which our construction is based on. [CLLZ21] establishes the following computational direct product theorem for coset states: a computationally bounded adversary who receives $|A_{s,s'}\rangle$ and programs $i\mathcal{O}(P_{A+s})$, $i\mathcal{O}(P_{A^\perp+s'})$ for uniformly random A, s, s', cannot produce a pair (v, w) such that $v \in A+s$ and $w \in A^\perp + s'$, except with negligible probability. From this computational direct product problem, they give the first instantiation of (weakly unforgeable) tokenized digital signatures based on classical cryptographic assumptions. Very roughly, the signing token is $|A_{s,s'}\rangle$, one can measure the state in the computational basis to obtain a signature for 0, and measure the state in the Hadamard basis to obtain a signature for 1.

To prove this computational direct product theorem, the authors invoke the notion of *subspace-hiding obfuscators*, introduced by Zhandry in [Zha19b]. A subspace hiding obfuscator shO has the property that any computationally bounded adversary who chooses a subspace A cannot distinguish between $\mathsf{shO}(P_A)$ and $\mathsf{shO}(P_B)$ for a uniformly random superspace B of A (of not too large dimension). Zhandry further shows that

indistinguishability obfuscation implies subspace-hiding obfuscation. The proof of the computational direct product problem is done in the following way. We assume that $A \subseteq \mathbb{F}_2^n$ has dimension $\frac{n}{2}$.

- Replace $i\mathcal{O}(P_{A+s})$ by $i\mathcal{O}(P_{B+s})$ for a uniformly random superspace B of A, where $\dim(B) = \frac{3n}{4}$. Similarly, replace $i\mathcal{O}(P_{A^\perp+s'})$ by $i\mathcal{O}(P_{C^\perp+s'})$ for a uniformly random superspace $C^\perp$ of $A^\perp$, where $\dim(C^\perp) = \frac{3n}{4}$.
- Argue that the task of finding a pair of vectors in $(A+s) \times (A^\perp + s')$ given $|A_{s,s'}\rangle, B, C$ for a uniformly random subspace $C \subseteq A \subseteq B$ is as hard as the task of finding a pair of vectors in $(\tilde{A}+z)\times(\tilde{A}+z')$ given $|\tilde{A}_{z,z'}\rangle$ for some uniformly random subspace $\tilde{A}$ of dimension $\frac{n}{4}$. The crucial observation is that, since $B+s = B+s+t$ for any vector $t \in B$, the programs P_{B+s} and P_{B+s+t} are functionally equivalent. So, an adversary who receives $i\mathcal{O}(P_{B+s})$ cannot distinguish this from $i\mathcal{O}(P_{B+s+t})$ for any t. We can think of t as a randomizing masking of s, which removes the adversary's knowledge about the membership programs.
- The latter task of finding such a pair of vectors corresponding to $\tilde{A}, z, z'$ is *information-theoretically* hard (it would even be hard with black-box access to the membership checking oracles for $\tilde{A}+z$ and $\tilde{A}^\perp + z'$).

To construct strongly unforgeable tokenized digital signatures, we would need a stronger version of the computational direct product theorem, where the task is now to find a pair of vectors $(v,w) \in (A+s)\times(A+s)$ or $(v,w) \in (A^\perp+s')\times(A^\perp+s')$ such that $v \neq w$. Applying the same reduction above allows us to reduce this task to the task of finding a pair of different vectors $v, w \in A+s$ given $|A_{s,s'}\rangle, B, C$ such that $C \subseteq A \subseteq B$. Unfortunately, we cannot directly reduce this task to the information-theoretic direct product theorem. The reason is that the adversary can just measure the state in the computational basis (or in the Hadamard basis if the task is to find a pair of different vectors in $A^\perp + s'$) to obtain a random vector $v \in A+s$, sample a non-zero vector $c \in C$ and output $(v, c+v)$. Since $c \in C \subseteq A$, we have that $c+v$ is also in $A+s$. This shows that the adversary can win the game without violating any complexity-theoretic arguments. Overcoming this technical hurdle requires a more involved analysis. We refer to subsequent sections for a formal description and proofs.

The outline of this section is as follows. We first recall some required definitions in Appendix A.1. A variant of the *computational direct product theorem* is given in Appendix A.2. The construction of strongly unforgeable tokenized digital signatures based on this variant and its proof of security are given in Appendix A.3.

A.1 Preliminaries: Tokenized Digital Signature

A formal definition of tokenized digital signature is given below.

Definition A.1 — Tokenized Digital Signature [BS17]

A tokenized digital signature scheme consists of four QPT algorithms $\mathsf{TDS} := \langle \mathsf{KeyGen}, \mathsf{TokenGen}, \mathsf{Sign}, \mathsf{TokenVerif}, \mathsf{Verif} \rangle$ with the following properties:

$(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$. On input the security parameter λ, the key generation algorithm KeyGen outputs a classical verification key vk and a secret key sk.

$\mathit{sig} \leftarrow \mathsf{TokenGen}(\mathsf{sk})$. On input the secret key sk, the token generation algorithm TokenGen outputs a signing token sig. We emphasize that if TokenGen is called ℓ times, it outputs different states $\mathit{sig}_1, \ldots, \mathit{sig}_\ell$.

$\sigma \leftarrow \mathsf{Sign}(m, \mathit{sig})$. On input a message $m \in \{0,1\}^*$ and a signing token sig, the signing algorithm Sign outputs a classical signature $\sigma \in \{0,1\}^{p(\lambda)}$.

$(b, \mathit{sig}') \leftarrow \mathsf{TokenVerif}(\mathsf{vk}, \mathit{sig})$. On input the verification key vk, and a signing token sig, the token verification TokenVerif outputs a single bit $b \in \{0,1\}$, and a post-verified token sig'.

$b \leftarrow \mathsf{Verif}(\mathsf{vk}, m, \sigma)$. On input the verification key vk, a message m and a classical signature σ, the verification algorithm outputs a bit $b \in \{0,1\}$.

A tokenized digital signature scheme TDS must satisfy the following requirements for all $\lambda \in \mathbb{N}$.

Correctness. For every message $m \in \{0,1\}^*$, we have that

$$\Pr\Big[\mathsf{Verif}(\mathsf{vk}, \mathsf{Sign}(m, \mathit{sig})) = 1 \mid (\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda); \mathit{sig} \leftarrow \mathsf{TokenGen}(\mathsf{sk})\Big] = 1,$$

where the probability is taken over randomness of KeyGen and TokenGen.

(Strong) Unforgeability. We introduce the algorithm Verif_k which takes as input the verification key vk and k pairs $(m_1, \sigma_1), \ldots, (m_k, \sigma_k)$ and returns 1 if and only if: (1) all the messages are distinct, that is, $m_i \neq m_j$ for all $1 \leq i \neq j \leq k$; and (2) all the pairs pass the verification, that is, $\mathsf{Verif}(\mathsf{vk}, m_i, \sigma_i) = 1$ for all $i \in [\![1, k]\!]$. For every $\ell \in \mathbb{N}$, no QPT adversary $\mathcal{A}$ can sign $\ell + 1$ different messages by using the verification key and ℓ signing tokens, except with negligible probability:

$$\mathsf{Adv}^{\mathsf{TDS}}(\lambda, \mathcal{A}) := \Pr\Big[\mathsf{Verif}_{\ell+1}(\mathsf{vk}, \mathcal{A}(\mathsf{vk}, \mathit{sig}_1 \otimes \cdots \otimes \mathit{sig}_\ell))\Big] \leq \mathsf{negl}(\lambda).$$

We also say that TDS is *strongly unforgeable* if we only require that k pairs of message/signature are distinct, that is $(m_i, \sigma_i) \neq (m_j, \sigma_j)$ for all $1 \leq i \neq j \leq k$. We further say that TDS is δ-secure, for some concrete negligible function $\delta(\lambda)$, if for all QPT adversaries $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathsf{TDS}}(\lambda, \mathcal{A})$ is smaller than $\delta(\lambda)^{\Omega(1)}$.

Testability. The token testing algorithm TokenVerif, unlike the signing algorithm, does not consume the signing token. If a signing token passes this test, the post-verified token also passes the test, and it can be used to sign a document. That is,

$$\Pr\Big[\mathsf{TokenVerif}(\mathit{sig}) = (1, \mathit{sig}) \mid (\mathsf{vk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{KeyGen}(1^\lambda); \mathit{sig} \xleftarrow{\$} \mathsf{TokenGen}(\mathsf{sk})\Big] = 1.$$

Furthermore, for any QPT adversary $\mathcal{A}$ with access to a verification key vk and polynomially many signing tokens, which generates a message m and a state $\widetilde{sig}$, we have that:

$$\Pr\Big[\mathsf{Verif}(\mathsf{vk}, m, \mathsf{Sign}(m, \widetilde{sig}')) = 1 \mid (1, \widetilde{sig}') \leftarrow \mathsf{TokenVerif}(\widetilde{sig})\Big] \geq 1 - \mathsf{negl}(\lambda),$$

$$\Pr\Big[\mathsf{TokenVerif}(\mathsf{vk}, \widetilde{sig}') = 1 \mid (1, \widetilde{sig}') \leftarrow \mathsf{TokenVerif}(\widetilde{sig})\Big] \geq 1 - \mathsf{negl}(\lambda).$$

A.2 Direct Product Hardness

Informally, the computational direct product hardness [CLLZ21] states that given $|A_{s,s'}\rangle$ and programs $i\mathcal{O}(P_{A+s})$ and $i\mathcal{O}(P_{A^\perp+s'})$ for uniformly random $A \subseteq \mathbb{F}_2^n, s, s' \in \mathbb{F}_2^n$, no QPT adversary can produce a pair $(v, w) \in (A + s) \times (A^\perp + s')$, except with negligible probability in n, where P_{A+s} and $P_{A^\perp+s'}$ are programs that check membership in the cosets $A + s$ and $A^\perp + s'$, respectively. We refer the reader to [CLLZ21] for more details.

In the following, we introduce a variant of the direct product problem, which very roughly states that it is also hard to produce a pair $(v, w) \in (A + s) \times (A + s)$ or $(v, w) \in (A^\perp + s') \times (A^\perp + s')$ such that $v \neq w$.

A.2.1 Information-Theoretic Direct Product Hardness - A Variant

Theorem A.1. *Let $A \subseteq \mathbb{F}_2^n$ be a uniformly random subspace of dimension $n/2$, and s, s' be uniformly random in $\mathbb{F}_2^n$. Let $\varepsilon > 0$ such that $1/\varepsilon = o\big(2^{n/2}\big)$. Let*

$$\Lambda(A, s) := (A + s) \times (A + s),$$

and

$$\Lambda(A^\perp, s') := (A^\perp + s') \times (A^\perp + s').$$

Given one copy of $|A_{s,s'}\rangle$ and quantum membership oracles for $A + s$ and $A^\perp + s'$, an adversary needs $\Omega\big(\sqrt{\varepsilon}2^{n/2}\big)$ queries to output a pair (v, w) such that $v \neq w$ and $(v, w) \in \Lambda(A, s)$ with probability at least ε.

The same number of queries is also required to output a pair $(v, w) \in \Lambda(A^\perp, s')$ satisfying $v \neq w$ with probability at least ε.

The proof of this theorem is similar to the proof of the original information-theoretic direct-product hardness [CLLZ21], which is a random self-reduction to the statement from Ben-David and Sattath [BS17]. We first present the theorem from [BS17].

Theorem A.2 ([BS17, Theorem 28]). *Let $A \subseteq \mathbb{F}_2^n$ be a uniformly random subspace of dimension $n/2$, and let $\varepsilon > 0$ such that $1/\varepsilon = o\big(2^{n/2}\big)$. Given one copy of $|A\rangle$ and quantum membership oracles for A and $A^\perp$ an adversary needs $\Omega\big(\sqrt{\varepsilon}2^{n/2}\big)$ queries to output a pair (v, w) such that $v \neq w$ and $(v, w) \in (A \setminus \{0\}) \times (A \setminus \{0\})$ with probability at least ε.*

The same number of queries is also required to output a pair $(v, w) \in (A^\perp \setminus \{0\}) \times (A^\perp \setminus \{0\})$ satisfying $v \neq w$ with probability at least ε.

Proof of Theorem A.1. We note that finding such a pair of elements in A and finding such a pair in $A^\perp$ are essentially the same task; thus it suffices to prove the result for a pair in $A \setminus \{0\}$, and the result for the other case will follow by symmetry.

Let $\mathcal{A}$ be an adversary for Theorem A.1 who suceeds with probability p, we construct an adversary $\mathcal{B}$ for Theorem A.2 with almost the same success probability making the same number of queries. $\mathcal{B}$ proceeds as follows.

- $\mathcal{B}$ receives $|A\rangle$ for some $A \subseteq \mathbb{F}_2^n$. Sample $s, s' \in \mathbb{F}_2^n$ uniformly at random, and create the state $|A_{s,s'}\rangle$.
- $\mathcal{B}$ gives $|A_{s,s'}\rangle$ as input to $\mathcal{A}$. $\mathcal{B}$ simulates the membership oracles $A+s$ and $A^\perp+s'$ as follows. If it is a query to the oracle $A+s$, $\mathcal{B}$ receives v from $\mathcal{A}$, and sends a query as $v-s$ to its membership oracle for A. It forwards the answer to $\mathcal{A}$. The other case is handled similarly, using its membership oracle for $A^\perp$ and s'.
- Finally, $\mathcal{B}$ receives (v, w) in return from $\mathcal{A}$. $\mathcal{B}$ then outputs $(v-s, w-s)$.

With probability p, $\mathcal{A}$ outputs $(v, w) \in \Lambda(A, s)$ such that $v \neq w$. Thus the output of $\mathcal{B}$ is $(v-s, w-s)$ such that $(v-s, w-s) \in A \times A$ and $v-s \neq w-s$. Next, we argue that with overwhelming probability, we have that $v-s \neq 0$ and $w-s \neq 0$. This is equivalent to show that the probability that $v-s=0$ or $w-s=0$ is negligible. Note that there are $2^{n/2}$ values of $\tilde{s}$ such that $|A_{\tilde{s},s'}\rangle = |A_{s,s'}\rangle$, since translating s by an element $\tilde{s}$ of A does not affect the state. Since s is sampled uniformly at random, the probability that $v-s=0$ or $w-s=0$ is equal to the probability that $v-\tilde{s}=0$ or $w-\tilde{s}=0$. This probability is $2 \cdot \frac{1}{2^{n/2}}$, which is negligible. □

A.2.2 Computational Direct Product Hardness - A Variant

In this section, we prove a computational version of the direct product problem, in which the adversary is given obfuscations of the subspace membership checking programs, but is restricted to be computationally bounded. Our computational version extends the original statement given in [CLLZ21, Theorem 4.6] to include the computational version of Theorem A.1.

Theorem A.3. *Assume the existence of quantum-secure indistinguishability obfuscation and quantum-secure injective one-way functions. Let $A \subseteq \mathbb{F}_2^n$ be a uniformly random subspace of dimension $n/2$, and s, s' be uniformly random in $\mathbb{F}_2^n$. Given one copy of $|A_{s,s'}\rangle$, $i\mathcal{O}(P_{A+s})$ and $i\mathcal{O}(P_{A^\perp+s'})$, any* QPT *adversary outputs a pair (v, w) such that either*

(i) $(v, w) \in \Lambda(A, s)$ *and* $v \neq w$;

(ii) *or* $(v, w) \in \Lambda(A^\perp, s')$ *and* $v \neq w$;

(iii) *or* $(v, w) \in (A+s) \times (A^\perp+s')$;

with negligible probability.

We first state the two lemmas required to prove Theorem A.3, and then assume correctness of these lemmas to prove Theorem A.3. In the subsequent section, we prove the required lemmas.

The first required lemma (proven in Appendix A.2.3) shows that the first and the second requirement (i)–(ii) are provably satisfied, except that we strengthen these requirements to require that the output vectors differ in the last $7n/8$ positions.

Lemma A.1. *Let $T := [\![\frac{n}{8}, n-1]\!]$. Under the same assumptions as Theorem A.3, given one copy of $|A_{s,s'}\rangle$, $i\mathcal{O}(P_{A+s})$ and $i\mathcal{O}(P_{A^\perp+s'})$ any* QPT *adversary outputs a pair (v,w) such that either*

(i) $(v,w) \in \Lambda(A,s)$ and $v|_T \neq w|_T$;

(ii) or $(v,w) \in \Lambda(A^\perp,s')$ and $v|_T \neq w|_T$;

with negligible probability.

The second required lemma is identical to the first lemma, except that now we require the output vectors to be different in the first $7n/8$ positions. The proof of this lemma is trivially adapted from that one of Lemma A.1.

Lemma A.2. *Let $T := [\![0, \frac{7n}{8}-1]\!]$. Under the same assumptions as Theorem A.3, given one copy of $|A_{s,s'}\rangle$, $i\mathcal{O}(P_{A+s})$ and $i\mathcal{O}(P_{A^\perp+s'})$, any* QPT *adversary outputs a pair (v,w) such that either*

(i) $(v,w) \in \Lambda(A,s)$ and $v|_T \neq w|_T$;

(ii) or $(v,w) \in \Lambda(A^\perp,s')$ and $v|_T \neq w|_T$;

with negligible probability.

We also recall the original computational direct product hardness stated in [CLLZ21] for completeness.

Theorem A.4 ([CLLZ21, Theorem 4.6]). *Assume the existence of quantum-secure indistinguishability obfuscation and injective one-way functions. Let $A \subseteq \mathbb{F}_2^n$ be a uniformly random subspace of dimension $n/2$, and s, s' be uniformly random in $\mathbb{F}_2^n$. Given one copy of $|A_{s,s'}\rangle$, $i\mathcal{O}(P_{A+s})$ and $i\mathcal{O}(P_{A^\perp+s'})$, any* QPT *adversary outputs a pair v, w such that $v \in A+s$ and $w \in A^\perp + s'$ with negligible probability.*

Assuming Lemma A.1 and Lemma A.2, we prove Theorem A.3 as follows.

Proof of Theorem A.3. We note that the third item (iii) is proven by Theorem A.4, and that finding such a different pair of vectors in $\Lambda(A,s)$ and finding such a pair in $\Lambda(A^\perp,s')$ are essentially the same task. Thus, it suffices to prove the first item (i), and the second item (ii) will follow by symmetry.

We prove item (i) by contrapositive. Suppose it is false. Then there exists a QPT adversary $\mathcal{A}$ that given $|A_{s,s'}\rangle, i\mathcal{O}(P_{A+s})$ and $i\mathcal{O}(P_{A^\perp+s'})$ for a uniformly random $A \subseteq \mathbb{F}_2^n$ and uniformly random vectors $s, s' \in \mathbb{F}_2^n$, returns $(v,w) \in \Lambda(A,s)$ such that $v \neq w$ with non-negligible probability ϵ.

Let $T_1 := [\![0, \frac{7n}{8}-1]\!]$ and $T_2 := [\![\frac{n}{8}, n-1]\!]$. For any pair $v \neq w$, it must be the case that $v|_{T_1} \neq w|_{T_1}$ or $v|_{T_2} \neq w|_{T_2}$. Let p_1 be the probability that $\mathcal{A}$ returns (v,w) such that $v|_{T_1} \neq w|_{T_1}$, and p_2 be the probability that $\mathcal{A}$ $\mathcal{A}$ returns (v,w) such that $v|_{T_2} \neq w|_{T_2}$. (These probabilities are taken over everything: the randomness of the challenger and of the adversary.) Then, by the union bound, we have that $p_1 + p_2 \geq \epsilon$. Since ϵ is non-negligible, at least one of p_1 or p_2 must be non-negligible. If p_2 is non-negligible, then the adversary $\mathcal{A}$ contradicts Lemma A.1. Similarly, if p_1 is non-negligible, $\mathcal{A}$ contradicts Lemma A.2. □

A.2.3 Proof of Lemma A.1

Before presenting the proof of Lemma A.1, we recall below the notion of *subspace hiding obfuscation* for the reader's convenience.

Subspace Hiding Obfuscation

Subspace-hiding obfuscation was introduced by Zhandry [Zha19b] as a key component in constructing public-key quantum money. This notion requires that the obfuscation of a circuit that computes membership in a subspace A is indistinguishable from the obfuscation of a circuit that computes membership in a uniformly random superspace of A (of dimension sufficiently far from the full dimension). The formal definition is as follows.

Definition A.2 — Subspace Hiding Obfuscation [Zha19b]

A subspace hiding obfuscator for a field $\mathbb{F}$ and dimensions d_0, d_1 is a PPT algorithm shO such that:

Input. shO takes as input the description of a linear subspace $S \subseteq \mathbb{F}^n$ of dimension $d \in \{d_0, d_1\}$. For concreteness, we will assume S is given as a matrix whose rows form a basis for S.

Output. shO outputs a circuit $\hat{S}$ that computes membership in S. Precisely, let $S(x)$ be the function that decides membership in S. Then

$$\Pr\left[\hat{S}(x) = S(x)\ \forall\ x \mid \hat{S} \leftarrow \mathsf{shO}(S)\right] \geq 1 - \mathsf{negl}(\lambda).$$

In the following, we will write shO_A as shorthand for $\mathsf{shO}(A)$.

Security. For security, consider the following game between an adversary and a challenger.

- The adversary submits to the challenger a subspace S_0 of dimension d_0.
- The challenger samples a uniformly random subspace $S_1 \subseteq \mathbb{F}^n$ of dimension d_1 such that $S_0 \subseteq S_1$. It then runs $\hat{S} \leftarrow \mathsf{shO}(S_b)$, and gives $\hat{S}$ to the adversary.
- The adversary makes a guess b' for b.

shO is secure if all QPT adversaries have negligible advantage in this game. We will represent the advantage of the adversary in this game by $\mathsf{Adv}^{\mathsf{shO}}(\lambda, \mathcal{A})$.

Zhandry [Zha19b] gives a construction of a subspace hiding obfuscator based on one-way functions and indistinguishability obfuscation.

Theorem A.5 ([Zha19b, Theorem 6.3]). *If injective one-way functions exist, then any indistinguishability obfuscator, appropriately padded, is also a subspace hiding obfuscator for field $\mathbb{F}$ and dimensions d_0, d_1 as long as $|\mathbb{F}|^{n-d_1}$ is exponential.*

Proof of Lemma A.1

Proof. We note that finding such a pair of elements in $\Lambda(A, s)$ and finding such a pair in $\Lambda(A^{\perp}, s')$ are essentially the same task; thus it suffices to prove the result for a pair in $\Lambda(A, s)$, and the result for the other case will follow by symmetry.

Let $\mathcal{A}$ be a QPT adversary for the direct product game of Lemma A.1. The proof of the lemma proceeds by a sequence of hybrids. For any hybrid G_i, we denote by $\mathsf{Adv}_i(\mathcal{A})$ the advantage of $\mathcal{A}$ in G_i, where the probability is taken over the random coins of G_i and $\mathcal{A}$.

Game G_0: This is the direct product game of Lemma A.1.

Game G_1: This is identical to G_0, except that now the obfuscation $i\mathcal{O}(P_{A+s})$ is replaced by $i\mathcal{O}(i\mathcal{O}(P_A)(\cdot - s))$. For simplicity, in the following, we abuse the notation and write $i\mathcal{O}(i\mathcal{O}(P_A)(\cdot - s))$ as $i\mathcal{O}(P_A(\cdot - s))$.

Claim A.1 (From G_0 to G_1). *For any* QPT *adversary* $\mathcal{A}$*, we have that*

$$|\mathsf{Adv}_0(\mathcal{A}) - \mathsf{Adv}_1(\mathcal{A})| \leq \mathsf{Adv}^{i\mathcal{O}}(\lambda, \mathcal{A}).$$

Proof. We note that both P_{A+s} and $P_A(\cdot - s)$ compute the same functionality, since any vector $v \in A + s$ if and only if $v - s \in A$. By security of $i\mathcal{O}$, the two games are computationally indistinguishable. □

Game G_2: This is identical to G_1, except that now the challenger samples uniformly at random a superspace B of A of dimension $\frac{7n}{8}$, and the obfuscation $i\mathcal{O}(P_A(\cdot - s))$ is replaced by $i\mathcal{O}(P_B(\cdot - s))$.

Claim A.2 (From G_1 to G_2). *For any* QPT *adversary* $\mathcal{A}$*, we have that*

$$|\mathsf{Adv}_1(\mathcal{A}) - \mathsf{Adv}_2(\mathcal{A})| \leq \mathsf{Adv}^{i\mathcal{O}}(\lambda, \mathcal{A}).$$

Proof. Since B is a superspace of A of dimension $\frac{7n}{8}$, by the subspace hiding property of $i\mathcal{O}$ (Lemma 2.8), the two games are computationally indistinguishable. □

Game G_3: This is identical to G_2, except that now the challenger samples uniformly at random an element w_B from B, and the obfuscation $i\mathcal{O}(P_B(\cdot - s)$ is replaced by $i\mathcal{O}(P_B(\cdot - t))$, where $t = s + w_B$.

Claim A.3 (From G_2 to G_3). *For any* QPT *adversary* $\mathcal{A}$*, we have that*

$$|\mathsf{Adv}_2(\mathcal{A}) - \mathsf{Adv}_3(\mathcal{A})| \leq \mathsf{Adv}^{i\mathcal{O}}(\lambda, \mathcal{A}).$$

Proof. We note that both $P_B(\cdot - t)$ and $P_B(\cdot - s)$ compute the same functionality, since for any vector $w_B \in B$, we have $B + w_B$ is the same as B. By security of $i\mathcal{O}$, the two games are computationally indistinguishable. □

Game G_4: This is identical to G_3, except that now the obfuscation $i\mathcal{O}(P_{A^{\perp}+s'})$ is replaced by $i\mathcal{O}(P_{A^{\perp}}(\cdot - s'))$.

Claim A.4 (From G_3 to G_4). *For any* QPT *adversary* $\mathcal{A}$*, we have that*

$$|\mathsf{Adv}_3(\mathcal{A}) - \mathsf{Adv}_4(\mathcal{A})| \leq \mathsf{Adv}^{i\mathcal{O}}(\lambda, \mathcal{A}).$$

Proof. We note that both $P_{A^\perp + s'}$ and $P_{A^\perp}(\cdot - s')$ compute the same functionality, since any vector $v \in A^\perp + s'$ if and only if $v - s' \in A^\perp$. By security of $i\mathcal{O}$, the two games are computationally indistinguishable. □

Game G_5: This is identical to G_4, except that now the challenger samples uniformly at random a superspace $C^\perp$ of $A^\perp$ of dimension $\frac{7n}{8}$, and the obfuscation $i\mathcal{O}(P_{A^\perp}(\cdot - s'))$ is replaced by $i\mathcal{O}(P_{C^\perp}(\cdot - s'))$.

Claim A.5 (From G_4 to G_5). *For any* QPT *adversary* $\mathcal{A}$*, we have that*

$$|\mathsf{Adv}_4(\mathcal{A}) - \mathsf{Adv}_5(\mathcal{A})| \leq \mathsf{Adv}^{i\mathcal{O}}(\lambda, \mathcal{A}).$$

Proof. Since $C^\perp$ is a superspace of $A^\perp$ of dimension $\frac{7n}{8}$, by security of subspace hiding obfuscation, the two games are computationally indistinguishable. □

Game G_6: This is identical to G_5, except that now the challenger samples uniformly at random an element $w_{C^\perp}$ from $C^\perp$, and the obfuscation $i\mathcal{O}(P_{C^\perp}(\cdot - s')$ is replaced by $i\mathcal{O}(P_{C^\perp}(\cdot - t'))$, where $t' = s' + w_{C^\perp}$.

Claim A.6 (From G_5 to G_6). *For any* QPT *adversary* $\mathcal{A}$*, we have that*

$$|\mathsf{Adv}_5(\mathcal{A}) - \mathsf{Adv}_6(\mathcal{A})| \leq \mathsf{Adv}^{i\mathcal{O}}(\lambda, \mathcal{A}).$$

Proof. We note that both $P_{C^\perp}(\cdot - t')$ and $P_{C^\perp}(\cdot - s')$ compute the same functionality, since for any vector $w_{C^\perp} \in C^\perp$, we have $C^\perp + w_{C^\perp}$ is the same as $C^\perp$. By security of $i\mathcal{O}$, the two games are computationally indistinguishable. □

Game G_7: This is identical to G_6, except that now we change the winning condition of the hybrid: instead of asking the adversary to output two vectors $v, w \in \Lambda(A, s)$ such that $v|_T \neq w|_T$, we ask the adversary to output two vectors $v, w \in \Lambda(A, s)$ such that $v|_T \neq w|_T$ and $v - w \in (A \setminus C)$, where C, the dual subspace of $C^\perp$, is of dimension $\frac{n}{8}$.

Claim A.7 (From G_6 to G_7). *For any* QPT *adversary* $\mathcal{A}$*, if* $\mathsf{Adv}_6(\mathcal{A})$ *is non-negligible, then there exists a non-negligible function* $\varepsilon = \varepsilon(\lambda)$ *such that*

$$\mathsf{Adv}_7(\mathcal{A}) \geq \varepsilon.$$

Proof. Due to our choice of dimension of subspaces B, C, we can apply the anti-concentration of the subspace obfuscator to prove the claim. Formally, we invoke the following lemma from [Shm22b], which states that any adversary, given an obfuscation $i\mathcal{O}(P_{C^\perp})$ where $C^\perp$ is a random high-dimensional superspace of $A^\perp$ and outputting a vector in A, has to accidentally hit the subspace $A \setminus C$ with a noticeable probability.

Lemma A.3 ([Shm22b, Lemma 5.1]). *Let $S = \{S_\lambda\}_{\lambda\in\mathbb{N}}$ a subspace $S_\lambda \subseteq \mathbb{F}_2^\lambda$ of dimension $d = \{d_\lambda\}_{\lambda\in\mathbb{N}}$. Let $t = \{t_\lambda\}_{\lambda\in\mathbb{N}}$ such that there is some constant $\delta \in (0,1)$ with $\forall\lambda \in \mathbb{N} : t_\lambda \geq \lambda^\delta$ and $\lambda - d_\lambda - 2\cdot t_\lambda \geq \Omega(\lambda)$. Let $i\mathcal{O}$ a quantum-secure indistinguishability obfuscation scheme for classical circuits and assume that post-quantum injective one-way functions exist. Then, there is no quantum polynomial-time algorithm $\mathcal{A} = \{\mathcal{A}_{\lambda,\rho_\lambda}\}_{\lambda\in\mathbb{N}}$, a negligible function* negl *and a non-negligible function η such that*

$$\Pr\left[\mathcal{A}_\lambda(\rho_\lambda, i\mathcal{O}(P_T)) \in T^\perp \mid T \xleftarrow{\$} \mathcal{S}^{\subseteq}_{\lambda-t}\right] \geq \eta(\lambda),$$

and

$$\Pr\left[\mathcal{A}_\lambda(\rho_\lambda, i\mathcal{O}(P_T)) \in (S^\perp \setminus T^\perp) \mid T \xleftarrow{\$} \mathcal{S}^{\subseteq}_{\lambda-t}\right] \leq \mathsf{negl}(\lambda),$$

where $\{\mathcal{S}^{\subseteq}_{\lambda-t}\}_{\lambda\in\mathbb{N}}$ is the uniform distribution over subspaces of dimension $\lambda - t_\lambda$ that contain S.

By applying the anti-concentration lemma above with $\lambda = n, t = \frac{n}{8}$ and $d = \frac{n}{2}$, we have that if $\mathsf{Adv}_6(\mathcal{A})$ is non-negligible, then the probability that $v - w \in (A \setminus C)$ is at least $\mathsf{Adv}_6(\mathcal{A}) - \mathsf{negl}(\lambda)$, concluding the claim. □

Game G_8: This is identical to G_7, except that now instead of sending obfuscations of membership checking programs, the challenger sends B, C, t, t' in clear to $\mathcal{A}$.

Claim A.8 (From G_7 to G_8). *For any* QPT *adversary $\mathcal{A}$ for G_7, there exists an adversary $\mathcal{B}$ for G_8 such that*

$$\mathsf{Adv}_7(\mathcal{A}) \leq \mathsf{Adv}_8(\mathcal{B}).$$

Proof. This is immediate. □

Claim A.9. *For any (possibly unbounded) adversary $\mathcal{A}$, we have that*

$$\mathsf{Adv}_8(\mathcal{A}) \leq \mathsf{negl}(\lambda).$$

Proof. We will follow the proof of [CLLZ21, Lemma 4.13]. Suppose there exists a QPT adversary $\mathcal{A}$ for G_8 that wins with probability p.

We first show that, without loss of generality, one can take B to be the subspace of vectors such that the last $n/8$ entries are zero (and the rest are free), and one can take C to be such that the last $7n/8$ entries are zero (and the rest are free). We construct the following adversary $\mathcal{B}$ for the game where B and C have the special form above with trailing zeros, call these B_* and C_*, from an adversary $\mathcal{A}$ for the game of G_8.

- $\mathcal{B}$ receives a state $|A_{s,s'}\rangle$ together with t, t', for some $C_* \subseteq A \subseteq B_*$, where $t = s + w_{B_*}$ for $w_{B_*} \xleftarrow{\$} B_*$, and $t' = s' + w_{C_*^\perp}$ for $w_{C_*^\perp} \xleftarrow{\$} C_*^\perp$.
- $\mathcal{B}$ picks uniformly at random subspaces B and C of dimension $7n/8$ and $n/8$ respectively, such that $C \subseteq B$. $\mathcal{B}$ also picks a uniformly random isomorphism $\mathcal{T}$ mapping C_* to C and B_* to B. $\mathcal{B}$ applies to $|A_{s,s'}\rangle$ the unitary $U_\mathcal{T}$ which acts as $\mathcal{T}$ on the standard basis elements. $\mathcal{B}$ gives $U_\mathcal{T}|A_{s,s'}\rangle$ to $\mathcal{A}$ together with $B, C, \mathcal{T}(t), (\mathcal{T}^{-1})^T(t')$.
- $\mathcal{B}$ receives (v, w) from $\mathcal{A}$, and outputs $(\mathcal{T}^{-1}(v), \mathcal{T}^{-1}(w)))$.

First, notice that

$$\begin{aligned} U_{\mathcal{T}} \left|A_{s,s'}\right\rangle &= U_{\mathcal{T}} \sum_{v \in A} (-1)^{\langle v,s' \rangle} \left|v+s\right\rangle \\ &= \sum_{v \in A} (-1)^{\langle v,s' \rangle} \left|\mathcal{T}(v)+\mathcal{T}(s)\right\rangle \\ &= \sum_{w \in \mathcal{T}(A)} (-1)^{\langle \mathcal{T}^{-1}(w),s' \rangle} \left|w+\mathcal{T}(s)\right\rangle \\ &= \sum_{w \in \mathcal{T}(A)} (-1)^{\langle w,(\mathcal{T}^{-1})^T(s') \rangle} \left|w+\mathcal{T}(s)\right\rangle \\ &= \left|\mathcal{T}(A)_{z,z'}\right\rangle, \end{aligned}$$

where $z = \mathcal{T}(s)$ and $z' = (\mathcal{T}^{-1})^T(s')$.

Furthermore, notice that $\mathcal{T}(A)$ is a uniformly random subspace between C and B, and that z and z' are uniformly random vectors in $\mathbb{F}_2^n$. We argue that:

(i) $\mathcal{T}(t)$ is distributed as a uniformly random element of $B + z$.

(ii) $(\mathcal{T}^{-1})^T(t')$ is distributed as a uniformly random element of $C^\perp + z'$.

For (i), notice that

$$\mathcal{T}(t) = \mathcal{T}(s + w_{B_*}) = \mathcal{T}(s) + \mathcal{T}(w_{B_*}) = z + \mathcal{T}(w_{B_*}),$$

where w_{B_*} is uniformly random in B_*. Since $\mathcal{T}$ is an isomorphism with $\mathcal{T}(B_*) = B$, $\mathcal{T}(w_{B_*})$ is uniformly random in B. Thus, $\mathcal{T}(t)$ is distributed as a uniformly random element in $B + z$.

For (ii), notice that

$$\begin{aligned} (\mathcal{T}^{-1})^T(t') = (\mathcal{T}^{-1})^T(s' + w_{C_*^\perp}) &= (\mathcal{T}^{-1})^T(s') + (\mathcal{T}^{-1})^T(w_{C_*^\perp}) \\ &= z' + (\mathcal{T}^{-1})^T(w_{C_*^\perp}), \end{aligned}$$

where $w_{C_*^\perp}$ is uniformly random in $C_*^\perp$. Let $x \in C$, then

$$\langle (\mathcal{T}^{-1})^T(w_{C_*^\perp}), x \rangle = \langle w_{C_*^\perp}, \mathcal{T}^{-1}(x) \rangle = 0,$$

where the last equality follows because $w_{C_*^\perp} \in C_*^\perp$ and $\mathcal{T}^{-1}(C) = C_*$. Thus $(\mathcal{T}^{-1})^T(w_{C_*^\perp})$ belongs to $C^\perp$. Since $(\mathcal{T}^{-1})^T$ is a bijection, $(\mathcal{T}^{-1})^T(w_{C_*^\perp})$ is uniformly random in $C^\perp$. It follows that $(\mathcal{T}^{-1})^T(t')$ is distributed as a uniformly random element in $C^\perp + z'$.

Hence, $\mathcal{A}$ receives the correct distribution, and thus, with probability p, $\mathcal{A}$ returns a pair $(v, w) \in \Lambda(\mathcal{T}(A), z)$ satisfying $v|_T \neq w|_T$ and $v - w \in (\mathcal{T}(A) \setminus C)$.

Notice that:

- If $v \in \mathcal{T}(A) + z$, where $z = \mathcal{T}(s)$, then $\mathcal{T}^{-1}(v) \in A + s$.
- If $v - w \in (\mathcal{T}(A) \setminus C)$, then $v - w \notin C$. Thus, we have $\mathcal{T}^{-1}(v) - \mathcal{T}^{-1}(w) = \mathcal{T}^{-1}(v - w) \notin \mathcal{T}^{-1}(C) = C_*$. Since C_* is the subspace of vectors such that the last $7n/8$ entries are zero, we also have that $\mathcal{T}^{-1}(v)|_T \neq \mathcal{T}^{-1}(w)|_T$.

Thus, with the same probability p, $\mathcal{B}$ returns a pair $(v', w') \in \Lambda(A, s)$ such that $v'|_T \neq w'|_T$ and $v' - w' \in (A \setminus C_*)$, as desired.

So, we can now assume that B is the space of vectors such that the last $\frac{n}{8}$ entries are zero, and C is the space of vectors such that the last $\frac{7n}{8}$ entries are zero. Then, the sampled subspace A is a uniformly random subspace subject to the last $\frac{n}{8}$ entries being zero, and the first $\frac{n}{8}$ entries being free. From an adversary $\mathcal{A}$ for G_8 with such B and C, we will construct an adversary $\mathcal{B}$ for the information-theoretic direct product problem described in Theorem A.1, where the ambient subspace is $\mathbb{F}_2^m$ where $m = \frac{3n}{4}$. $\mathcal{B}$ works as follows.

- $\mathcal{B}$ receives $|A_{s,s'}\rangle$ for uniformly random $A \subseteq \mathbb{F}_2^m$ of dimension $\frac{m}{2}$ and uniformly random $s, s' \in \mathbb{F}_2^m$. $\mathcal{B}$ samples $\tilde{s}, \tilde{s}', \hat{s}, \hat{s}' \xleftarrow{\$} \mathbb{F}_2^{\frac{n}{8}}$.
  Let $|\phi\rangle = \frac{1}{2^{n/16}} \sum_{x \in \{0,1\}^{n/8}} (-1)^{\langle x, \tilde{s}' \rangle} |x + \tilde{s}\rangle$. $\mathcal{B}$ creates the state
  $$|W\rangle = |\phi\rangle \otimes |A_{s,s'}\rangle \otimes |\hat{s}\rangle .$$
  $\mathcal{B}$ gives to $\mathcal{A}$ as input the state $|W\rangle$, together with $t = 0^{7n/8} \| \hat{s} + w_B$ for $w_B \xleftarrow{\$} B$, and $t' = \hat{s}' \| 0^{7n/8} + w_{C^\perp}$, for $w_{C^\perp} \xleftarrow{\$} C^\perp$.
- $\mathcal{A}$ returns a pair $(v, w) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$. Let $v' = v|_{[\![\frac{n}{8}, \frac{7n}{8} - 1]\!]} \in \mathbb{F}_2^m$ be the "middle" $\frac{3n}{4}$ entries of v. Let $w' = w|_{[\![\frac{n}{8}, \frac{7n}{8} - 1]\!]}$. $\mathcal{B}$ outputs (v', w').

Notice that

$$\begin{aligned}
|W\rangle &= |\phi\rangle \otimes |A_{s,s'}\rangle \otimes |\hat{s}\rangle \\
&= \sum_{x \in \{0,1\}^{n/8}, v \in A} (-1)^{\langle x, \tilde{s}' \rangle} (-1)^{\langle v, s' \rangle} |(x + \tilde{s}) \| (v + s) \| \hat{s}\rangle \\
&= \sum_{x \in \{0,1\}^{n/8}, v \in A} (-1)^{\langle (x \| v \| 0^{n/8}), (\tilde{s}' \| s' \| \hat{s}') \rangle} \left| (x \| v \| 0^{n/8} + \tilde{s} \| s \| \hat{s}) \right\rangle \\
&= \sum_{w \in \tilde{A}} (-1)^{\langle w, z' \rangle} |w + z\rangle = |\tilde{A}_{z,z'}\rangle ,
\end{aligned}$$

where $z = \tilde{s} \| s \| \hat{s}$, $z' = \tilde{s}' \| s' \| \hat{s}'$, and $\tilde{A}$ is the subspace in which the first $n/8$ entries are free, the middle $3n/4$ entries belong to the subspace A, and the last $n/8$ entries are zero.

Notice that the subspace $\tilde{A}$, when averaging over the choice of A, is distributed precisely as in the game G_8 (with B and C of special form with trailing zeros); z, z' are uniformly random in $\mathbb{F}_2^n$; t is uniformly random from $B + z$ and t' is uniformly random from $C^\perp + z'$. Thus, with probability p, $\mathcal{A}$ returns to $\mathcal{B}$ a pair $(v, w) \in \Lambda(\tilde{A}, z)$ such that $v|_T \neq w|_T$ and $v - w \in \tilde{A} \setminus C$. Furthermore, we note that if $(v, w) \in \Lambda(\tilde{A}, z)$, the last $n/8$ entries of both v and w must be $\hat{s}$. It follows that, if $v|_T \neq w|_T$, we have that $v' \neq w'$. Overall, we have that with probability p, the answer (v', w') returned by $\mathcal{B}$ is such that $(v', w') \in \Lambda(A, s)$ satisfying $v' \neq w'$.

By Theorem A.1, we deduce that p must be negligible. □

Therefore we show that the advantage of distinguishing G_0 and G_6 is negligible, and the success probability in G_7 is at most the success probability in G_8, which is negligible. We finish the proof by invoking Claim A.7, which concludes that the success probability in G_6 must also be negligible. □

A.3 Strongly Unforgeable Tokenized Digital Signatures

In this section, we show how to construct *strongly unforgeable* tokenized digital signatures from indistinguishability obfuscation.

Following [BS17], we first define a notion of *one-bit one-time* strongly unforgeable tokenized digital signatures. Then, using the construction given in [BS17], one can obtain a full-fledged strongly unforgeable scheme by combining a one-bit one-time strongly unforgeable scheme with any classical strongly unforgeable digital signature scheme against quantum attacks.

Definition A.3 — One-bit One-time Strongly Unforgeable Tokenized Digital Signatures

A tokenized digital signature scheme TDS is *one-bit one-time* strongly unforgeable if for every λ, for every QPT adversary $\mathcal{A}$, we have that

$$\Pr\left[\begin{array}{c|c} m_0, m_1 \in \{0,1\} & (\mathsf{vk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{KeyGen}(1^\lambda) \\ \wedge\ \mathsf{Verif}_2(\mathsf{vk}, m_0, \sigma_0, m_1, \sigma_1) = 1 & \mathit{sig} \xleftarrow{\$} \mathsf{TokenGen}(\mathsf{sk}) \\ \wedge\ (m_0, \sigma_0) \neq (m_1, \sigma_1) & (m_0, \sigma_0, m_1, \sigma_1) \leftarrow \mathcal{A}(\mathsf{vk}, \mathit{sig}) \end{array}\right] \leq \mathsf{negl}(\lambda).$$

Furthermore, let $\mathsf{Adv}^{1-\mathsf{TDS}}(\lambda, \mathcal{A})$ denote the above probability. We say that TDS is δ-strongly unforgeable, for some concrete negligible function $\delta(\lambda)$, if for all QPT adversary $\mathcal{A}$, the advantage $\mathsf{Adv}^{1-\mathsf{TDS}}(\lambda, \mathcal{A})$ is smaller than $\delta(\lambda)^{\Omega(1)}$.

The following theorem, whose proof is given in [BS17], says that one-bit one-time strong unforgeability is sufficient to achieve a full-fledged strong unforgeability.

Theorem A.6 (Adapted from [BS17, Theorem 13][18]). *A one-bit one-time strongly unforgeable tokenized digital signature scheme implies a full-fledged strongly unforgeable tokenized digital signature scheme, assuming the existence of a strongly unforgeable quantum-secure digital signature scheme.*

Construction. Next, we give a construction of one-bit one-time strongly unforgeable digital signatures from hidden coset states in Figure A.1. This construction is identical to the one for weak unforgeability in [CLLZ21].

Theorem A.7. *Assuming the existence of quantum-secure indistinguishability obfuscation and quantum-secure injective one-way functions, the scheme given in Figure A.1 is a one-bit one-time strongly unforgeable tokenized digital signature scheme.*

Proof. The proof of this theorem follows immediately from Theorem A.3. □

Since one-way functions imply digital signatures, we have the following corollary:

Corollary A.1. *Assuming the existence of quantum-secure indistinguishability obfuscation and quantum-secure injective one-way functions, there exists a strongly unforgeable tokenized digital signature scheme.*

Proof. This follows immediately from Theorem A.6 and Theorem A.7. □

[18] While the statement of [BS17, Theorem 13] only applies to weak unforgeability, the same proof extends to strong unforgeability.

KeyGen(1^λ) : Set $n = \mathsf{poly}(\lambda)$.

- Sample uniformly $A \subseteq \mathbb{F}_2^n$ of dimension $\frac{n}{2}$.
- Sample $s, s' \xleftarrow{\$} \mathbb{F}_2^n$.
- Output $\mathsf{sk} := (A, s, s')$ (where by A we mean a description of the subspace A), and $\mathsf{vk} := (i\mathcal{O}(P_{A+s}), i\mathcal{O}(P_{A^\perp+s'}))$.

TokenGen(sk) : Take as input sk of the form (A, s, s').

- Output $\mathit{sig} := |A_{s,s'}\rangle$.

Sign(m, sig) : Take as input $m \in \{0,1\}$, and a state sig on n qubits.

- Compute $\mathsf{H}^{\otimes n}\mathit{sig}$ if $m = 1$, otherwise do nothing to the quantum state.
- Measure the state in the computational basis. Let σ be the outcome.
- Output (m, σ).

Verif$(\mathsf{vk}, (m, \sigma))$: Parse vk as (C_0, C_1) where C_0 and C_1 are circuits.

- Output $C_m(\sigma)$.

TokenVerif$(\mathsf{vk}, \mathit{sig})$: Parse vk as (C_0, C_1) where C_0 and C_1 are circuits.

- Let V_i be the unitary implementing the following operation:

$$V_i |v, z\rangle \mapsto |v, z \oplus C_i(v)\rangle .$$

  Compute $\mathit{sig}' := (\mathsf{H}^{\otimes n} \otimes \mathcal{I})V_1(\mathsf{H}^{\otimes n} \otimes \mathcal{I})V_0\mathit{sig} \otimes |0\rangle$.
- Measure the last register in the computational basis.
- If the outcome is 1, return $(0, \mathit{sig}')$. Otherwise, return $(1, \mathit{sig}')$.

Figure A.1: A one-bit one-time strongly unforgeable scheme from hidden coset states.

Appendix

B Password-Authenticated Quantum Key Exchange

In their 1984 seminal paper [BB84], Bennett and Brassard gave the first proof that the laws of quantum mechanics could lead to an achievement of *unconditional security* for classical cryptographic tasks. Their celebrated Quantum Key Distribution protocol (so-called QKD) allows two parties to agree on a common secret key which is information-theoretic secret, assuming a quantum channel and an authenticated (but not secret) classical channel. Even though this protocol is a conceptual milestone in the quantum cryptography field, the need for an information-theoretically authenticated classical communication channel leads to a bootstrapping problem. In practice, implementations of unconditionally secure QKD leave no choice but requiring Alice and Bob to use a pre-shared short random secret key (to authenticate the messages with authentication codes constructed from universal hashing) in order to obtain a larger random secret key. Another unavoidable problem is that the authentication keys can be run out, because either the adversary makes the execution fail (denial-of-service attack) or due to technical problems (the parties cannot exclude that an eavesdropper was in fact present). Moreover, when considering large scale quantum networks, in which secure communication should be possible between any pair of nodes, the requirement for pre-shared randomness does not scale well: each node would have to store a number of keys, which is linear in the size of the network, let alone the problem of key management.

On the contrary, in so-called *authenticated* key exchange, the two parties are able to generate a shared cryptographic secret key, to be later used with symmetric primitives in order to protect communications, while interacting over an *insecure* network under the control of an adversary. Various authentication means have been introduced for classical networks. The most practical ones are certainly based on either Public Key Infrastructures (PKI) or human-memorable passwords. The latter leads to PAKE, standing for *Password-Authenticated Key Exchange*. PAKE protocols allow users to securely establish a common cryptographic key over an *insecure and unauthenticated* channel only using a low-entropy, human-memorable secret key called a *password*. The advantage of a PAKE, in sharp contrast to all QKD-like schemes, is that no authenticated channel is needed. In the classical setting, PAKE has been extensively studied, resulting in various secure and efficient protocols. However, classical PAKE protocols can only achieve computational security, where the adversary's power is computationally limited. Thus, it is natural to ask if we can achieve a provably stronger security notion for password-based key exchange protocols using quantum communication.

Unfortunately, even if QKD raised a lot of hope on unconditional security using quantum mechanics, a series of no-go theorems showed that the dream of unconditional security brought by quantum communication will never be a reality for many crypto-

graphic tasks. For instance, several attempts have been made to achieve unconditionally secure quantum bit-commitments, until Mayers and Lo and Chau independently showed that statistically hiding and binding quantum commitments are impossible [LC97; May97]. The impossibility of quantum cryptography was further extended to oblivious transfer (OT) by Lo [Lo97], and finally extended to non-trivial two-party computation protocols by Salvail *et al.* and Buhrman *et al.* [SSS09; BCS12]. Intuitively, the insecurity of two-party quantum protocols follows from the fact that the protocol itself allows parties to input a superposed state rather than a classical one, and perform an appropriate measurement on the outcome state. At the end of the protocol, one party can always gain more information on the input of the other than that gained using any honest strategy.

In this appendix, we study the security of password-based key exchange protocols with quantum communication and establish several impossibilities in different security models. This is a joint work with Céline Chevalier and Marc Kaplan in 2018.

B.1 Security Models

We provide a brief overview of security models for *multi-party computation* (MPC), in which n players interact in order to compute securely a given function of their inputs. Formally, consider n players P_i, each owning an input x_i, and a classical n-input function f. The goal is to compute $(y_1, \ldots, y_n) = f(x_1, \ldots, x_n)$ such that each player P_i learns y_i, and cheating players cannot change the outcome of the computation (apart by choosing a different input) and do not learn more about the input (and possibly the output) of honest players than what can be derived from their own input and their output of the function evaluation.

B.1.1 The Simulation-based Paradigm

The first step towards the solution for this security definition is the *simulation-based* paradigm. Instead of introducing different notions for each security property, we consider for each protocol, the "ideal behavior" it should have. Intuitively, we introduce the notion of "ideal world" where there is a trusted party who collects the inputs from all players, computes the output and distributes the output to the players. A real protocol is compared to an ideal protocol, and the real protocol is said to be *at least as secure as* the ideal protocol if the real protocol and the ideal protocol have an indistinguishable input-output behavior. The level of security reached thus also depends on the specification of the ideal protocol.

B.1.2 Universal Composability

However, as being pointed out in the literature, the simulation-based paradigm does not play well with *composition* and in fact, it only achieves *Sequential Composition*, i.e., a protocol that is secure under sequential composition maintains its security when run multiple times, as long as the executions are run sequentially (meaning that each execution concludes before the next execution begins). In the case of *Concurrent Composition* in which many instances of the same protocol with correlated inputs are run concurrently, some problems may occur. For example, the messages from one

protocol could be fed into another, or a message from one sub-protocol of a larger application is fed into another sub-protocol and the overall application becomes insecure. In order to solve this inherent problem, the so-called UC (for *Universal Composability*) framework was introduced. We give a high-level overview of the model below and refer the reader to [Can01] for more details on the classical version and [Unr10] for the quantum version.

Ideal World and Real World. We define in the ideal world an entity that one can never corrupt, called the *ideal functionality* and usually denoted as $\mathcal{F}$. The players privately send their inputs to this entity, and receive their corresponding output the same way. There is no communication between the different players. $\mathcal{F}$ is assumed to behave in a perfectly correct way, without revealing information other than required, and without being possibly corrupted by an adversary. Once $\mathcal{F}$ is defined, the goal of a protocol π, executed in a real world in the presence of an adversary, is then to create a situation equivalent to that obtained with $\mathcal{F}$.

Protocol, Adversary, and Environment. Apart from the protocol participants which are specified by the protocol, there are two more machines taking part in the protocol execution. The *adversary* $\mathcal{A}$ (or $\mathcal{S}$ in the ideal model) is the machine coordinating all corrupted participants analogous to the simulation-based model. The *environment machine* $\mathcal{Z}$, playing the role of the *distinguisher*, models "everything that is outside the protocol being executed". It chooses the inputs, sees the outputs, and may communicate with the adversary at any time. The adversary has access to the communication between players, but not to the inputs and outputs of the honest players (it completely controls the dishonest or corrupted players). On the contrary, the environment has access to the inputs and outputs of all players, but not to their communication, nor to the inputs and outputs of the subroutines they can invoke.

A protocol π *securely realizes* a functionality $\mathcal{F}$ if for every real-world adversary $\mathcal{A}$ there exists an ideal-world adversary $\mathcal{S}$, called the simulator, such that no environment can distinguish whether it is witnessing the real-world execution with adversary $\mathcal{A}$ or the ideal-world execution with simulator $\mathcal{S}$, with a non-negligible advantage. Depending on the assumed computing power of the adversary and the environment we distinguish between *computational* security, where they are all considered to be polynomially bounded machines, and *statistical* security, where they are assumed to be computationally unbounded. Furthermore, in [Unr13], Unruh introduces the notion of *everlasting* security, where the adversary is considered to be a polynomial-time machine but the environment is assumed to have unbounded computational power.

In addition, the notion of "hybrid models" is also introduced to model the concept of set-up assumptions. A protocol π is said to be realized "in the $\mathcal{F}$-hybrid model" if π can invoke the ideal functionality $\mathcal{F}$ as a subroutine multiple times. We note that the environment can never interact directly with $\mathcal{F}$, and thus, $\mathcal{F}$ is usually never invoked at all in the ideal world, and the implementation of $\mathcal{F}$ is simulated solely by the ideal adversary $\mathcal{S}$. The model with no trusted set-up is called *plain* model.

Ideal Functionalities. We denote $\mathcal{F}_{\mathsf{CRS}}$ the common reference string functionality and $\mathcal{F}_{\mathsf{OT}}$ the oblivious transfer functionality. The definitions of these functionalities are given as follows.

The common reference string (CRS) model is modeled by the functionality $\mathcal{F}_{\mathsf{CRS}}^{\mathcal{D}}$, which was presented in [BCNP04]. At each call of $\mathcal{F}_{\mathsf{CRS}}^{\mathcal{D}}$, it sends back the same reference string, chosen by itself, following a known public distribution $\mathcal{D}$. We recall it here in Figure B.1.

Functionality $\mathcal{F}_{\mathsf{CRS}}^{\mathcal{D}}$

The functionality $\mathcal{F}_{\mathsf{CRS}}^{\mathcal{D}}$ is parameterized by a distribution $\mathcal{D}$. It interacts with a set of players and an adversary in the following way:

- Choose a value $r \xleftarrow{\$} \mathcal{D}$.
- Upon receiving a value $(\mathsf{CRS}, \mathsf{sid})$ from a player, send $(\mathsf{CRS}, \mathsf{sid}, r)$ to this player.

Figure B.1: The functionality $\mathcal{F}_{\mathsf{CRS}}^{\mathcal{D}}$.

Oblivious Transfer (OT) is a very powerful tool and is sufficient to realize any secure computation functionality [Kil88]. Informally, OT is a two-party functionality, involving a sender S with input x_0, x_1 and a receiver R with an input $\sigma \in \{0,1\}$. The receiver R learns x_σ (and nothing else), and the sender learns nothing at all. These requirements are captured by the specification of the OT functionality $\mathcal{F}_{\mathsf{OT}}$ from [CLOS02], given in Figure B.2.

Functionality $\mathcal{F}_{\mathsf{OT}}$

The functionality $\mathcal{F}_{\mathsf{OT}}$ interacts with a sender S, a receiver R and an adversary $\mathcal{S}$.

- Upon receiving a message $(\mathsf{sender}, \mathsf{sid}, x_0, x_1)$ from S, where each $x_i \in \{0,1\}^\ell$, store (x_0, x_1). (The length ℓ of the strings is fixed and known to all parties).
- Upon receiving a message $(\mathsf{receiver}, \mathsf{sid}, \sigma)$ from R, check if a sender message was previously sent. If yes, send (sid, x_σ) to R and sid to the adversary $\mathcal{S}$ and halt. If not, send nothing to R (but continue running).

Figure B.2: The oblivious transfer functionality $\mathcal{F}_{\mathsf{OT}}$.

B.2 Reduction from PAKE to EQUALITY

We show our impossibilities by reducing the problem of constructing a scheme for the PAKE functionality to the problem of constructing a scheme for an equality-testing functionality. In particular, we consider an explicit mutual authentication PAKE functionality $\mathcal{F}_{\text{e-pwKE}}$ whose description is given in Figure B.3. The description of the functionality is a modified version of the description in [ACCP09; CDVW12].

We then define an equality-testing functionality $\mathcal{F}_{\mathsf{EQ}}$ (Figure B.4) that, roughly speaking, takes inputs from two parties and does the following:

- if the inputs are equal, outputs the value 1 to both parties; moreover, if either party is corrupted, the adversary is allowed to set the output.

The functionality $\mathcal{F}_{\text{e-pwKE}}$

The functionality $\mathcal{F}_{\text{e-pwKE}}$ is parameterized by a security parameter λ and a "dictionary" $\mathcal{D}$. It interacts with an adversary $\mathcal{A}$ and a set of parties via the following queries:

Upon receiving a query $(\text{NewSession}, sid, P_i, P_j, \pi)$ **from party** P_i:

Send $(\text{NewSession}, sid, P_i, P_j)$ to $\mathcal{A}$. In addition, if this is the first NewSession query, or if this is the second NewSession query and there is a record (sid, P_j, P_i, π'), then record (sid, P_i, P_j, π) and mark this record fresh. In the latter case, also record (sid, ready), and send it to $\mathcal{A}$.

Upon receiving a query $(\text{TestPwd}, sid, P, \pi')$ **from** $\mathcal{A}$:

If $P \in \{P_i, P_j\}$, and there is a record of the form $(sid, P, *, \pi)$ which is fresh, then do: If $\pi = \pi'$, mark the record compromised and reply to $\mathcal{A}$ with "correct guess". If $\pi \neq \pi'$, mark the record interrupted and reply to $\mathcal{A}$ with "wrong guess".

Upon receiving a query $(\text{NewKey}, sid, P_i, P_j, sk)$ **from** $\mathcal{A}$, **where** $|sk| = \lambda$:

If there is a record (sid, ready), and there is a record of the form $(sid, P, *, \pi)$ where $P \in \{P_i, P_j\}$ then:

- If this record is fresh, and there is a record $(sid, *, P, \pi')$ marked fresh with $\pi = \pi'$, pick a new random key sk' of length λ and set out $= sk'$.
- If this record is compromised, or either P_i or P_j is corrupted, then set out $= sk$.
- In any other case, set out $= \perp$.

Either way, mark both record $(sid, P, *, \pi)$ and $(sid, *, P, \pi')$ as completed.

Upon receiving a query $(\text{Deliver}, sid, P)$ **from** $\mathcal{A}$:

If $P \in \{P_i, P_j\}$, and there is a record of the form $(sid, P, *, \pi)$ which is completed, then send $(\text{deliver}, sid, \text{out})$ to P. Ignore all subsequent $(\text{Deliver}, P)$ queries for the same player P.

Figure B.3: The password-based key-exchange functionality $\mathcal{F}_{\text{e-pwKE}}$ with explicit mutual authentication.

- if the inputs are unequal, send both parties the special symbol $\perp$.

The functionality $\mathcal{F}_{\mathsf{EQ}}$

The functionality $\mathcal{F}_{\mathsf{EQ}}$ is parameterized by a security parameter λ and a "dictionary" $\mathcal{D}$. It interacts with two parties P_i, P_j, and an adversary $\mathcal{A}$ via the following queries:

Upon receiving a query $(\mathsf{NewSession}, sid, P_i, P_j, \pi)$ **from party** P_i:

Send $(\mathsf{NewSession}, sid, P_i, P_j)$ to $\mathcal{A}$. In addition, do the following:

- If this is the first NewSession query, then record (sid, P_i, P_j, π) and mark this record fresh.
- If this is the second NewSession query and there is a record (sid, P_j, P_i, π') which is fresh, then do: if $\pi = \pi'$, then set out $= 1$, otherwise, set out $= \perp$. Mark both records completed.

Upon receiving a query $(\mathsf{Test}, sid, P, \pi'), P \in \{P_i, P_j\}$ **from** $\mathcal{A}$:

If there is a record of the form $(sid, P, *, \pi)$ which is fresh, then do: If $\pi = \pi'$, mark the record compromised and reply to $\mathcal{A}$ with "correct guess". If $\pi \neq \pi'$, mark the record interrupted and reply to $\mathcal{A}$ with "wrong guess".

Upon receiving a query $(\mathsf{Output}, sid, \gamma), \gamma \in \{1, \perp\}$ **from** $\mathcal{A}$:

If there is a record of the form $(sid, *, *, \pi)$ which is compromised, or one of the parties is corrupted, then set out $= \gamma$. If this record is interrupted, then set out $= \perp$. Otherwise, do nothing.

Upon receiving a query $(\mathsf{Deliver}, sid, P), P \in \{P_i, P_j\}$ **from** $\mathcal{A}$:

If there is a record of the form $(sid, P, *, \pi)$ which is completed, send $(\mathsf{deliver}, sid, \mathsf{out})$ to the player P. Ignore all subsequent $(\mathsf{Deliver}, P)$ queries for the same player P.

Figure B.4: The equality-testing functionality $\mathcal{F}_{\mathsf{EQ}}$.

More precisely, $\mathcal{F}_{\mathsf{EQ}}$ captures a protocol between two parties P_i, P_j started by having the two parties sending messages to the functionality with their secret strings π_i, π_j. If the inputs match, the functionality assigns the output to be 1, otherwise it sets the output to be $\perp$. Finally, the adversary $\mathcal{A}$ instructs the functionality when to send the output to both parties. Thus, this definition corresponds to achieving *explicit mutual authentication*. We also allow the adversary three special powers. First, we allow him to set the output if one of the parties is corrupted and both the parties have the same input. Furthermore, he controls the delivery of messages to the parties. This is an ability that he inevitably has in the real world. Finally, as in the case of PAKE, the low entropy of the messages in the dictionary $\mathcal{D}$ makes online dictionary attacks unavoidable, which is captured by the Test query given to the adversary. The following lemma shows that the

$\mathcal{F}_{\text{e-pwKE}}$ functionality already implements the $\mathcal{F}_{\text{EQ}}$. Though this seems to be folklore, we also give a proof of this lemma for completeness.

Lemma B.1. *There is a protocol that perfectly implements the $\mathcal{F}_{\text{EQ}}$ functionality in the $\mathcal{F}_{\text{e-pwKE}}$ hybrid model, tolerating adaptive corruptions and without assuming authenticated channels.*

Proof. The protocol that implements $\mathcal{F}_{\text{EQ}}$ simply forwards the parties' messages to the $\mathcal{F}_{\text{e-pwKE}}$ functionality. In particular, on input (sid, π_i) from the environment, the party P_i sends a message $(\text{NewSession}, sid, P_i, P_j, \pi)$ to $\mathcal{F}_{\text{e-pwKE}}$. When P_i receives a message $(\text{deliver}, sid, \text{out})$ back from $\mathcal{F}_{\text{e-pwKE}}$, if out $\neq \perp$, P_i outputs 1, otherwise, it outputs $\perp$ and terminates. Similarly, P_j does the same.

We simply show how to simulate the adversary $\mathcal{A}'s$ messages.

Simulating a $(\text{Test}, sid, P, \pi)$ query from $\mathcal{A}$: If $\mathcal{A}$ already sent a $(\text{Deliver}, P)$ query before, ignore this query. Otherwise, send a query $(\text{TestPwd}, sid, P, \pi)$ to $\mathcal{F}_{\text{e-pwKE}}$, and record the response from $\mathcal{F}_{\text{e-pwKE}}$ (either "correct guess" or "wrong guess").

Simulating a $(\text{Output}, sid, \gamma)$ query from $\mathcal{A}$: If $\mathcal{A}$ already sent a $(\text{Deliver}, P)$ query before, ignore this query. Otherwise, send a query $(\text{NewKey}, sid, P_i, P_j, \gamma)$ to $\mathcal{F}_{\text{e-pwKE}}$.

Simulating a $(\text{Deliver}, sid, P)$ query from $\mathcal{A}$: If $\mathcal{A}$ already sent a $(\text{Deliver}, P)$ query before, ignore this query. Otherwise, send a query $(\text{Deliver}, sid, P)$ to $\mathcal{F}_{\text{e-pwKE}}$.

It is easy to see that the simulation is perfect, and the view of the environment is identical in the real execution of $\mathcal{A}$ in the protocol (in the $\mathcal{F}_{\text{e-pwKE}}$-hybrid model) and the simulated ideal-model execution with $\mathcal{F}_{\text{EQ}}$. □

B.3 On the Impossibility of Securely Realizing PAKE

In this section, we show negative results on the achievable security of Password-based Key Exchange protocols when allowed to use quantum communication. We focus on two composability settings: Either a "minimal" simulation-based security following a real world-ideal world paradigm, as defined in [Can00; FS09], or the full universally composable security [Can01; Unr10].

Following the literature, we call *plain* model the setting in which there are no setup assumptions (such as public-key infrastructure (PKI), common reference string (CRS), random oracles (ROM), etc). Following for instance [KLR06], in which the authors study the connections between information-theoretic security and security under composition, we consider here the information-theoretic setting, in which the adversary is polynomially unbounded. Informally, the output of a real execution of the protocol with a real adversary must be (perfectly or statistically) the same as the output of an ideal execution with a trusted party and an ideal-world adversary/simulator. On the contrary, in the computational setting, we focus on the notion of *everlasting* security [MU07; Unr13], which informally means that the adversary is polynomially bounded during the execution of the protocol, and unbounded afterwards. This models an adversary possibly saving transcripts today, in order to potentially use them at the time a quantum computer is built.

B.3.1 Implicit or Explicit Authentication

We recall an important property of a PAKE protocol: it guarantees that if the same password was entered, the generated session key is the same for both parties, but they might not know at the end of the protocol whether it is so. This property is known as *implicit* authentication, as opposed to *explicit* authentication, in which the parties know whether they share the same session key at the end of the protocol. In both cases, the protocol should guarantee that if the passwords were different, the session keys are independent and random.

The line of work for impossibility results that we continue here focuses on *non-trivial* protocols[19] with explicit authentication. It is known at least since [BPR00, Section 5] that explicit authentication can be added at no security cost to any protocol with implicit authentication, using a *key confirmation* technique. The obtained key K would be used as the key for a PRF secure for 3 queries, one of the players would send $PRF_K(1)$ to the other, the other would send $PRK_K(2)$ to the first one, and both would end up using $PRF_K(0)$ as the final session key[20]. This implies that the following results also hold for protocols with *implicit* authentication.

B.3.2 Impossibility in the Simulation-Based Model

Theorem B.1. *There is no statistically simulation-based secure PAKE protocol with explicit authentication in the plain model.*

Proof. To prove the theorem, we employ a general result which proves that for the class of deterministic, two-sided functionalities including the equality-testing function, the security for one party implies complete insecurity for the other in the simulation-based model.

Lemma B.2 ([BCS12, Theorem 2]). *If a protocol π for the evaluation of a deterministic two-sided function F is ε-correct and ε-secure against Bob, then there is a cheating strategy for Alice (where she uses input u_0 and Bob has input v) which gives her $\tilde{v}$ distributed according to some distribution $Q(\tilde{v}|u_0, v)$ such that for all u:* $\Pr[\tilde{v} \leftarrow Q : F(u,v) = F(u,\tilde{v})] \geq 1 - 28\varepsilon$.

First we note that the reduction from $\mathcal{F}_{\mathsf{EQ}}$ to $\mathcal{F}_{\text{e-pwKE}}$ in Lemma B.1 holds unconditionally in the UC model, which implies perfect security in the simulation-based model. We then prove by contradiction, if there is a statistically secure PAKE protocol in the plain model, then by Lemma B.1, that protocol is also a statistically secure protocol for $\mathcal{F}_{\mathsf{EQ}}$ in the plain model, which violates Lemma B.2. □

[19] As explained for instance in [CHKL+05, Section 7], the results are only interesting for what they call *non-trivial* protocols, in which two parties agree on a shared secret key at the end of the execution of the protocol (except perhaps with negligible probability), if 1) they use the same password and 2) the adversary passes all messages between the parties without modifying them or inserting any messages of its own. This is required since otherwise the *empty* protocol in which parties do nothing would securely realize any PAKE functionality.

[20] A trivial construction of such a (perfect) PRF would be to split the key into three parts, use the two first parts as key confirmations and the last one as the real session key.

B.3.3 Impossibility in the Universally Composability Model

As in the classical case (Canetti *et al.* prove in [CHKL+05] the impossibility of universally composable PAKE in the plain model), the (im)possibility of PAKE depends on the existence of some setup assumption. As shown by Unruh in [Unr13], the classical notion of passive adversaries (which copy all data) does not make sense in the quantum case. He thus considers only *unitary* protocols, which perform no measurements (any protocol can be transformed into such a protocol using additional quantum memory). Unruh then defines a functionality $\mathcal{F}$ to be *quantum-passively-realizable* it there exists a unitary protocol that realizes $\mathcal{F}$ with respect to passive unlimited adversaries (that follow the protocol exactly and do not even copy information). The following lemma gives examples of quantum-passively-realizable functionalities.

Lemma B.3 ([Unr13, Lemma 8]). *The following functionalities are quantum-passively-realizable: $\mathcal{F}_{CT}$ (coin-toss), $\mathcal{F}_{CRS}$ (common reference string), $\mathcal{F}_{EPR}$ (predistributed EPR pair), $\mathcal{F}_{PKI}$ (public key infrastructure; assuming that the secret key is uniquely determined by the public key).*

We state the following impossibility theorem for PAKE in the UC model.

Theorem B.2. *There is no statistically or everlastingly quantum-UC-secure PAKE protocol with explicit authentication which only uses quantum-passively-realizable functionalities as trusted setup assumptions.*

Proof. First note that according to the following lemma, the impossibility of everlasting quantum-UC security implies the impossibility of statistical quantum-UC security.

Lemma B.4 ([Unr13, Lemma 1]). *Let π and ρ be protocols. If π statistically quantum-UC-emulates ρ, then π everlastingly quantum-UC-emulates ρ.*

In the following, we thus focus on the proof for the everlasting security.

Assuming some trusted setup, the following lemma states the impossibility of everlastingly realizing $\mathcal{F}_{\mathsf{EQ}}$ using only quantum-passively-realizable functionalities.

Lemma B.5. *There is no statistically or everlastingly quantum-UC-secure protocol that realizes $\mathcal{F}_{\mathsf{EQ}}$ which only uses quantum-passively-realizable functionalities as trusted setup assumptions.*

Before proving Lemma B.5, we recall the impossibility of achieving everlastingly quantum-UC-secure oblivious transfer.

Lemma B.6 ([Unr13, Theorem 5]). *There is no statistically or everlastingly quantum-UC-secure OT protocol which only uses quantum-passively-realizable functionalities as trusted setup assumptions.*

We use the notion of reductions between MPC functionalities, that allows us to form "classes" of functionalities with similar cryptographic complexity: Following [MPR10], a functionality is said *trivial* or *feasible* if it can be realized in the UC framework in the plain model (with no setup assumptions), and it is said *complete* if it is sufficient for computing arbitrary other functions, under appropriate complexity assumptions, when used as trusted setups. We recall the following results that are proven in [Unr10; FKSZ+13].

Lemma B.7 ([Unr10, Theorem 15] and [FKSZ+13, Theorem 2]). *The following statements hold:*

1. *If a protocol π statistically UC-realizes a functionality $\mathcal{F}$, then π statistically quantum-UC-realizes the functionality $\mathcal{F}$ (Quantum lifting theorem).*
2. *Feasibility in the quantum world is* equivalent *to classical feasibility, in both the computational and statistical setting.*

To show a reduction from $\mathcal{F}_{\mathsf{EQ}}$ to $\mathcal{F}_{\mathsf{OT}}$, we employ the following intermediate results.

Definition B.1 (OT-cores). *Let F be a deterministic two-party function, Γ_A, Γ_B be the input alphabet of two parties, Ω_A, Ω_B be the output distribution of two parties, and f_A, f_B be the output values of the two parties. A quadruple $(x, x', y, y') \in \Gamma_A^2 \times \Gamma_B^2$ is an* OT-core *of F, if the following three conditions are met:*

1. $f_A(x, y) = f_A(x, y')$.
2. $f_B(x, y) = f_B(x', y)$.
3. $f_A(x', y) \neq f_A(x', y')$ *or* $f_B(x, y') \neq f_B(x', y')$ *(or both).*

In [KM11] the so-called *Classification theorem* was proven, which shows a necessary and sufficient condition to have a reduction protocol from an ideal functionality $\mathcal{F}$ to $\mathcal{F}_{\mathsf{OT}}$.

Theorem B.3 (The Classification Theorem [KM11]). *There exists an OT protocol that is statistically secure against passive adversaries in the $\mathcal{F}$-hybrid model, for some $\mathcal{F}$, if and only if $\mathcal{F}$ has an OT-core.*

Proof of Lemma B.5. We first show that the equality-testing function $\mathcal{F}_{\mathsf{EQ}}$ admits an OT-core. Consider $\mathcal{F}_{\mathsf{EQ}} := (\Gamma_A, \Gamma_B, \Omega_A, \Omega_B, f_A, f_B)$, without loss of generality, assume $\Gamma_A = \Gamma_B = \Gamma$. Let $c \in \Gamma$ be a random value drawn from the input distribution, then a quadruple $(c, c+1, c-1, c+1)$ is an OT-core of $\mathcal{F}_{\mathsf{EQ}}$ because:

$$f_A(c, c-1) = f_A(c, c+1) = 0$$
$$f_B(c, c-1) = f_B(c+1, c-1) = 0$$
$$0 = f_A(c+1, c-1) \neq f_A(c+1, c+1) = 1$$

Then the classification theorem (Theorem B.3) tells us that there exists an OT protocol that is statistically secure against passive adversaries in the $\mathcal{F}_{\mathsf{EQ}}$-hybrid model. Using the lifting theorem (Lemma B.7), that protocol is also statistically secure against quantum-passive adversaries in the $\mathcal{F}_{\mathsf{EQ}}$-hybrid model.

We now prove the lemma by contradiction. Assume that there exists an everlasting quantum-UC-secure protocol π realizing $\mathcal{F}_{\mathsf{EQ}}$ which only uses quantum-passively-realizable functionalities. Let ρ be the protocol resulting from π by replacing invocations to $\mathcal{F}_{\mathsf{EQ}}$ by invocations to the subprotocol π. Then ρ is an everlasting quantum-UC-secure protocol realizing $\mathcal{F}_{\mathsf{OT}}$ which only uses quantum-passively-realizable functionalities against quantum-passive adversaries. This contradicts Lemma B.6.

Because of Lemma B.4, the impossibility of statistical security follows immediately from the impossibility of everlasting security. □

The proof of Theorem B.2 then follows directly from Lemma B.1 and Lemma B.5. □

Bibliography

[Aar09] Scott Aaronson: **Quantum copy-protection and quantum money**. In: *2009 24th Annual IEEE Conference on Computational Complexity*. IEEE. 2009, pp. 229–242 (cited on pages 5–6).

[Aar] Scott Aaronson: **Stephen Wiesner (1942-2021)**. https://scottaaronson.blog/?p=5730. Accessed: 2022-10-08 (cited on page 2).

[AC12] Scott Aaronson and Paul Christiano: **Quantum money from hidden subspaces**. In: *44th Annual ACM Symposium on Theory of Computing*. Ed. by Howard J. Karloff and Toniann Pitassi. ACM Press, May 2012, pp. 41–60. doi: 10.1145/2213977.2213983 (cited on pages 4, 103).

[ACCP09] Michel Abdalla, Dario Catalano, Céline Chevalier, and David Pointcheval: **Password-Authenticated Group Key Agreement with Adaptive Security and Contributiveness**. In: *AFRICACRYPT 09: 2nd International Conference on Cryptology in Africa*. Ed. by Bart Preneel. Vol. 5580. Lecture Notes in Computer Science. Springer, Heidelberg, June 2009, pp. 254–271 (cited on page 166).

[ACEM+22] Behzad Abdolmaleki, Céline Chevalier, Ehsan Ebrahimi, Giulio Malavolta, and Quoc-Huy Vu: **Quantum-Secure Simulation-Sound Non-Interactive Zero-Knowledge: Definitions, Constructions and Applications**. 2022 (cited on page 6).

[AGM18] Gorjan Alagic, Tommaso Gagliardoni, and Christian Majenz: **Unforgeable Quantum Encryption**. In: *Advances in Cryptology – EUROCRYPT 2018, Part III*. Ed. by Jesper Buus Nielsen and Vincent Rijmen. Vol. 10822. Lecture Notes in Computer Science. Springer, Heidelberg, Apr. 2018, pp. 489–519. doi: 10.1007/978-3-319-78372-7_16 (cited on page 40).

[AMRS20] Gorjan Alagic, Christian Majenz, Alexander Russell, and Fang Song: **Quantum-Access-Secure Message Authentication via Blind-Unforgeability**. In: *Advances in Cryptology – EUROCRYPT 2020, Part III*. Ed. by Anne Canteaut and Yuval Ishai. Vol. 12107. Lecture Notes in Computer Science. Springer, Heidelberg, May 2020, pp. 788–817. doi: 10.1007/978-3-030-45727-3_27 (cited on pages 3, 5, 74, 76).

[AGKZ20] Ryan Amos, Marios Georgiou, Aggelos Kiayias, and Mark Zhandry: **One-shot signatures and applications to hybrid quantum/classical authentication**. In: *52nd Annual ACM Symposium on Theory of Computing*. Ed. by Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy. ACM Press, June 2020, pp. 255–268. DOI: 10.1145/3357713.3384304 (cited on pages 4–5, 7).

[ATTU16] Mayuresh Vivekanand Anand, Ehsan Ebrahimi Targhi, Gelo Noel Tabia, and Dominique Unruh: **Post-Quantum Security of the CBC, CFB, OFB, CTR, and XTS Modes of Operation**. In: *Post-Quantum Cryptography - 7th International Workshop, PQCrypto 2016*. Ed. by Tsuyoshi Takagi. Springer, Heidelberg, 2016, pp. 44–63. DOI: 10.1007/978-3-319-29360-8_4 (cited on page 36).

[AKLL+22] Prabhanjan Ananth, Fatih Kaleoglu, Xingjian Li, Qipeng Liu, and Mark Zhandry: **On the Feasibility of Unclonable Encryption, and More**. In: *Advances in Cryptology – CRYPTO 2022, Part II*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Vol. 13508. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 2022, pp. 212–241. DOI: 10.1007/978-3-031-15979-4_8 (cited on pages 33, 103).

[AL21] Prabhanjan Ananth and Rolando L. La Placa: **Secure Software Leasing**. In: *Advances in Cryptology – EUROCRYPT 2021, Part II*. Ed. by Anne Canteaut and François-Xavier Standaert. Vol. 12697. Lecture Notes in Computer Science. Springer, Heidelberg, Oct. 2021, pp. 501–530. DOI: 10.1007/978-3-030-77886-6_17 (cited on page 5).

[AQY22] Prabhanjan Ananth, Luowen Qian, and Henry Yuen: **Cryptography from Pseudorandom Quantum States**. In: *Advances in Cryptology – CRYPTO 2022, Part I*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Vol. 13507. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 2022, pp. 208–236. DOI: 10.1007/978-3-031-15802-5_8 (cited on page 3).

[BCNP04] Boaz Barak, Ran Canetti, Jesper Buus Nielsen, and Rafael Pass: **Universally Composable Protocols with Relaxed Set-Up Assumptions**. In: *45th Annual Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Oct. 2004, pp. 186–195. DOI: 10.1109/FOCS.2004.71 (cited on page 166).

[BGIR+01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang: **On the (Im)possibility of Obfuscating Programs**. In: *Advances in Cryptology – CRYPTO 2001*. Ed. by Joe Kilian. Vol. 2139. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 2001, pp. 1–18. DOI: 10.1007/3-540-44647-8_1 (cited on page 26).

[BDJR97] Mihir Bellare, Anand Desai, Eric Jokipii, and Phillip Rogaway: **A Concrete Security Treatment of Symmetric Encryption**. In: *38th Annual Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Oct. 1997, pp. 394–403. DOI: 10.1109/SFCS.1997.646128 (cited on page 50).

[BDPR98] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway: **Relations Among Notions of Security for Public-Key Encryption Schemes**. In: *Advances in Cryptology – CRYPTO'98*. Ed. by Hugo Krawczyk. Vol. 1462. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 1998, pp. 26–45. DOI: 10.1007/BFb0055718 (cited on pages 61–62).

[BN08] Mihir Bellare and Chanathip Namprempre: **Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm**. In: *Journal of Cryptology*, **21**:4 (Oct. 2008), pp. 469–491. DOI: 10.1007/s00145-008-9026-x (cited on page 53).

[BPR00] Mihir Bellare, David Pointcheval, and Phillip Rogaway: **Authenticated Key Exchange Secure against Dictionary Attacks**. In: *Advances in Cryptology – EUROCRYPT 2000*. Ed. by Bart Preneel. Vol. 1807. Lecture Notes in Computer Science. Springer, Heidelberg, May 2000, pp. 139–155. DOI: 10.1007/3-540-45539-6_11 (cited on page 170).

[BS06] Mihir Bellare and Amit Sahai: **Non-Malleable Encryption: Equivalence between Two Notions, and an Indistinguishability-based Characterization**. Cryptology ePrint Archive, Report 2006/228. https://eprint.iacr.org/2006/228. 2006 (cited on pages 61–62).

[BS17] Shalev Ben-David and Or Sattath: **Quantum Tokens for Digital Signatures**. Cryptology ePrint Archive, Report 2017/094. https://eprint.iacr.org/2017/094. 2017 (cited on pages 5, 149–150, 152, 161).

[BBDQ18] Fabrice Benhamouda, Olivier Blazy, Léo Ducas, and Willy Quach: **Hash Proof Systems over Lattices Revisited**. In: *PKC 2018: 21st International Conference on Theory and Practice of Public Key Cryptography, Part II*. Ed. by Michel Abdalla and Ricardo Dahab. Vol. 10770. Lecture Notes in Computer Science. Springer, Heidelberg, Mar. 2018, pp. 644–674. DOI: 10.1007/978-3-319-76581-5_22 (cited on page 8).

[BB84] C. H. Bennett and G. Brassard: **Quantum cryptography: Public key distribution and coin tossing**. In: *Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing*. Bangalore, 1984, p. 175 (cited on pages 2, 163).

[BV93] Ethan Bernstein and Umesh V. Vazirani: **Quantum complexity theory**. In: *25th Annual ACM Symposium on Theory of Computing*. ACM Press, May 1993, pp. 11–20. DOI: 10.1145/167088.167097 (cited on page 1).

[BBCL+21] Ritam Bhaumik, Xavier Bonnetain, André Chailloux, Gaëtan Leurent, María Naya-Plasencia, André Schrottenloher, and Yannick Seurin: **QCB: Efficient Quantum-Secure Authenticated Encryption**. In: *Advances in Cryptology – ASIACRYPT 2021, Part I*. Ed. by Mehdi Tibouchi and Huaxiong Wang. Vol. 13090. Lecture Notes in Computer Science. Springer, Heidelberg, Dec. 2021, pp. 668–698. doi: 10.1007/978-3-030-92062-3_23 (cited on page 69).

[BKS21] Nir Bitansky, Michael Kellner, and Omri Shmueli: **Post-quantum Resettably-Sound Zero Knowledge**. In: *TCC 2021: 19th Theory of Cryptography Conference, Part I*. Ed. by Kobbi Nissim and Brent Waters. Vol. 13042. Lecture Notes in Computer Science. Springer, Heidelberg, Nov. 2021, pp. 62–89. doi: 10.1007/978-3-030-90459-3_3 (cited on page 3).

[BP15] Nir Bitansky and Omer Paneth: **ZAPs and Non-Interactive Witness Indistinguishability from Indistinguishability Obfuscation**. In: *TCC 2015: 12th Theory of Cryptography Conference, Part II*. Ed. by Yevgeniy Dodis and Jesper Buus Nielsen. Vol. 9015. Lecture Notes in Computer Science. Springer, Heidelberg, Mar. 2015, pp. 401–427. doi: 10.1007/978-3-662-46497-7_16 (cited on pages 73–74).

[BC15] Olivier Blazy and Céline Chevalier: **Generic Construction of UC-Secure Oblivious Transfer**. In: *ACNS 15: 13th International Conference on Applied Cryptography and Network Security*. Ed. by Tal Malkin, Vladimir Kolesnikov, Allison Bishop Lewko, and Michalis Polychronakis. Vol. 9092. Lecture Notes in Computer Science. Springer, Heidelberg, June 2015, pp. 65–86. doi: 10.1007/978-3-319-28166-7_4 (cited on page 7).

[BCV19] Olivier Blazy, Céline Chevalier, and Quoc Huy Vu: **Post-quantum UC-secure oblivious transfer in the standard model with adaptive corruptions**. In: *Proceedings of the 14th International Conference on Availability, Reliability and Security*. 2019, pp. 1–6 (cited on page 7).

[BDFL+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry: **Random Oracles in a Quantum World**. In: *Advances in Cryptology – ASIACRYPT 2011*. Ed. by Dong Hoon Lee and Xiaoyun Wang. Vol. 7073. Lecture Notes in Computer Science. Springer, Heidelberg, Dec. 2011, pp. 41–69. doi: 10.1007/978-3-642-25385-0_3 (cited on pages 3, 5).

[BKW17] Dan Boneh, Sam Kim, and David J. Wu: **Constrained Keys for Invertible Pseudorandom Functions**. In: *TCC 2017: 15th Theory of Cryptography Conference, Part I*. Ed. by Yael Kalai and Leonid Reyzin. Vol. 10677. Lecture Notes in Computer Science. Springer, Heidelberg, Nov. 2017, pp. 237–263. doi: 10.1007/978-3-319-70500-2_9 (cited on page 95).

[BL95] Dan Boneh and Richard J. Lipton: **Quantum Cryptanalysis of Hidden Linear Functions (Extended Abstract)**. In: *Advances in Cryptology – CRYPTO'95*. Ed. by Don Coppersmith. Vol. 963. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 1995, pp. 424–437. doi: 10.1007/3-540-44750-4_34 (cited on page 76).

[BW13] Dan Boneh and Brent Waters: **Constrained Pseudorandom Functions and Their Applications**. In: *Advances in Cryptology – ASIACRYPT 2013, Part II*. Ed. by Kazue Sako and Palash Sarkar. Vol. 8270. Lecture Notes in Computer Science. Springer, Heidelberg, Dec. 2013, pp. 280–300. doi: 10.1007/978-3-642-42045-0_15 (cited on page 18).

[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. Vol. 7881. Lecture Notes in Computer Science. Springer, Heidelberg, May 2013, pp. 592–608. doi: 10.1007/978-3-642-38348-9_35 (cited on pages 3, 5, 76).

[BZ13b] Dan Boneh and Mark Zhandry: **Secure Signatures and Chosen Ciphertext Security in a Quantum Computing World**. In: *Advances in Cryptology – CRYPTO 2013, Part II*. Ed. by Ran Canetti and Juan A. Garay. Vol. 8043. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 2013, pp. 361–379. doi: 10.1007/978-3-642-40084-1_21 (cited on pages 3, 5–6, 21–24, 36, 38–39, 51–52, 67, 74, 76).

[BF10] Niek J. Bouman and Serge Fehr: **Sampling in a Quantum Population, and Applications**. In: *Advances in Cryptology – CRYPTO 2010*. Ed. by Tal Rabin. Vol. 6223. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 2010, pp. 724–741. doi: 10.1007/978-3-642-14623-7_39 (cited on pages 16–17, 108, 110, 120, 133).

[BGI14] Elette Boyle, Shafi Goldwasser, and Ioana Ivan: **Functional Signatures and Pseudorandom Functions**. In: *PKC 2014: 17th International Conference on Theory and Practice of Public Key Cryptography*. Ed. by Hugo Krawczyk. Vol. 8383. Lecture Notes in Computer Science. Springer, Heidelberg, Mar. 2014, pp. 501–519. doi: 10.1007/978-3-642-54631-0_29 (cited on page 18).

[Bra18] Zvika Brakerski: **Quantum FHE (Almost) As Secure As Classical**. In: *Advances in Cryptology – CRYPTO 2018, Part III*. Ed. by Hovav Shacham and Alexandra Boldyreva. Vol. 10993. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 2018, pp. 67–95. doi: 10.1007/978-3-319-96878-0_3 (cited on page 28).

[BCMV+18] Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh V. Vazirani, and Thomas Vidick: **A Cryptographic Test of Quantumness and Certifiable Randomness from a Single Quantum Device**. In: *59th Annual Symposium on Foundations of Computer Science*. Ed. by Mikkel

Thorup. IEEE Computer Society Press, Oct. 2018, pp. 320–331. DOI: 10.1109/FOCS.2018.00038 (cited on pages 7, 76–78, 118).

[BDGM20] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta: **Factoring and Pairings are not Necessary for iO: Circular-Secure LWE Suffices**. Cryptology ePrint Archive, Report 2020/1024. https://eprint.iacr.org/2020/1024. 2020 (cited on page 26).

[BGS13] Anne Broadbent, Gus Gutoski, and Douglas Stebila: **Quantum One-Time Programs - (Extended Abstract)**. In: *Advances in Cryptology – CRYPTO 2013, Part II*. Ed. by Ran Canetti and Juan A. Garay. Vol. 8043. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 2013, pp. 344–360. DOI: 10.1007/978-3-642-40084-1_20 (cited on page 5).

[BI20] Anne Broadbent and Rabib Islam: **Quantum Encryption with Certified Deletion**. In: *TCC 2020: 18th Theory of Cryptography Conference, Part III*. Ed. by Rafael Pass and Krzysztof Pietrzak. Vol. 12552. Lecture Notes in Computer Science. Springer, Heidelberg, Nov. 2020, pp. 92–122. DOI: 10.1007/978-3-030-64381-2_4 (cited on page 4).

[BJLP+21] Anne Broadbent, Stacey Jeffery, Sébastien Lord, Supartha Podder, and Aarthi Sundaram: **Secure Software Leasing Without Assumptions**. In: *TCC 2021: 19th Theory of Cryptography Conference, Part I*. Ed. by Kobbi Nissim and Brent Waters. Vol. 13042. Lecture Notes in Computer Science. Springer, Heidelberg, Nov. 2021, pp. 90–120. DOI: 10.1007/978-3-030-90459-3_4 (cited on pages 4–5).

[BL20] Anne Broadbent and Sébastien Lord: **Uncloneable Quantum Encryption via Oracles**. In: Leibniz International Proceedings in Informatics (LIPIcs) **158**: (2020). Ed. by Steven T. Flammia, 4:1–4:22 (cited on page 4).

[BCS12] Harry Buhrman, Matthias Christandl, and Christian Schaffner: **Complete insecurity of quantum protocols for classical two-party computation**. In: *Physical review letters*, **109**:16 (2012), p. 160501 (cited on pages 164, 170).

[Can00] Ran Canetti: **Security and Composition of Multiparty Cryptographic Protocols**. In: *Journal of Cryptology*, **13**:1 (Jan. 2000), pp. 143–202. DOI: 10.1007/s001459910006 (cited on page 169).

[Can01] Ran Canetti: **Universally Composable Security: A New Paradigm for Cryptographic Protocols**. In: *42nd Annual Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Oct. 2001, pp. 136–145. DOI: 10.1109/SFCS.2001.959888 (cited on pages 165, 169).

[CDVW12] Ran Canetti, Dana Dachman-Soled, Vinod Vaikuntanathan, and Hoeteck Wee: **Efficient Password Authenticated Key Exchange via Oblivious Transfer**. In: *PKC 2012: 15th International Conference on Theory and Practice of Public Key Cryptography*. Ed. by Marc Fischlin, Johannes Buchmann, and Mark Manulis. Vol. 7293. Lecture Notes in Computer Science. Springer, Heidelberg, May 2012, pp. 449–466. DOI: 10.1007/978-3-642-30057-8_27 (cited on page 166).

[CHKL+05] Ran Canetti, Shai Halevi, Jonathan Katz, Yehuda Lindell, and Philip D. MacKenzie: **Universally Composable Password-Based Key Exchange**. In: *Advances in Cryptology – EUROCRYPT 2005*. Ed. by Ronald Cramer. Vol. 3494. Lecture Notes in Computer Science. Springer, Heidelberg, May 2005, pp. 404–421. DOI: 10.1007/11426639_24 (cited on pages 170–171).

[CLOS02] Ran Canetti, Yehuda Lindell, Rafail Ostrovsky, and Amit Sahai: **Universally composable two-party and multi-party secure computation**. In: *34th Annual ACM Symposium on Theory of Computing*. ACM Press, May 2002, pp. 494–503. DOI: 10.1145/509907.509980 (cited on page 166).

[CETU21] Tore Vincent Carstens, Ehsan Ebrahimi, Gelo Noel Tabia, and Dominique Unruh: **Relationships Between Quantum IND-CPA Notions**. In: *TCC 2021: 19th Theory of Cryptography Conference, Part I*. Ed. by Kobbi Nissim and Brent Waters. Vol. 13042. Lecture Notes in Computer Science. Springer, Heidelberg, Nov. 2021, pp. 240–272. DOI: 10.1007/978-3-030-90459-3_9 (cited on pages 38–39, 54, 69–70).

[CEV20] Céline Chevalier, Ehsan Ebrahimi, and Quoc-Huy Vu: **On Security Notions for Encryption in a Quantum World**. Cryptology ePrint Archive, Report 2020/237. https://eprint.iacr.org/2020/237. 2020 (cited on pages 5, 35).

[CHV22] Céline Chevalier, Paul Hermouet, and Quoc-Huy Vu: **Semi-Quantum Copy-Protection and More**. 2022 (cited on page 7).

[CCKW19] Alexandru Cojocaru, Léo Colisson, Elham Kashefi, and Petros Wallden: **QFactory: Classically-Instructed Remote Secret Qubits Preparation**. In: *Advances in Cryptology – ASIACRYPT 2019, Part I*. Ed. by Steven D. Galbraith and Shiho Moriai. Vol. 11921. Lecture Notes in Computer Science. Springer, Heidelberg, Dec. 2019, pp. 615–645. DOI: 10.1007/978-3-030-34578-5_22 (cited on page 104).

[CLLZ21] Andrea Coladangelo, Jiahui Liu, Qipeng Liu, and Mark Zhandry: **Hidden Cosets and Applications to Unclonable Cryptography**. In: *Advances in Cryptology – CRYPTO 2021, Part I*. Ed. by Tal Malkin and Chris Peikert. Vol. 12825. Lecture Notes in Computer Science. Virtual Event: Springer, Heidelberg, Aug. 2021, pp. 556–584. DOI: 10.1007/978-3-030-84242-0_20 (cited on pages 5, 7, 103, 105, 113, 141, 143–145, 149, 152–154, 158, 161).

[CMP20] Andrea Coladangelo, Christian Majenz, and Alexander Poremba: **Quantum copy-protection of compute-and-compare programs in the quantum random oracle model**. Cryptology ePrint Archive, Report 2020/1194. https://eprint.iacr.org/2020/1194. 2020 (cited on pages 33, 138).

[CV22] Eric Culf and Thomas Vidick: **A monogamy-of-entanglement game for subspace coset states**. In: *Quantum*, **6**: (2022), p. 791 (cited on pages 105, 113).

[CMSZ19] Jan Czajkowski, Christian Majenz, Christian Schaffner, and Sebastian Zur: **Quantum Lazy Sampling and Game-Playing Proofs for Quantum Indifferentiability**. Cryptology ePrint Archive, Report 2019/428. https://eprint.iacr.org/2019/428. 2019 (cited on pages 37, 42).

[DFNS14] Ivan Damgård, Jakob Funder, Jesper Buus Nielsen, and Louis Salvail: **Superposition Attacks on Cryptographic Protocols**. In: *ICITS 13: 7th International Conference on Information Theoretic Security*. Ed. by Carles Padró. Vol. 8317. Lecture Notes in Computer Science. Springer, Heidelberg, 2014, pp. 142–161. doi: 10.1007/978-3-319-04268-8_9 (cited on page 3).

[DDOP+01] Alfredo De Santis, Giovanni Di Crescenzo, Rafail Ostrovsky, Giuseppe Persiano, and Amit Sahai: **Robust Non-interactive Zero Knowledge**. In: *Advances in Cryptology – CRYPTO 2001*. Ed. by Joe Kilian. Vol. 2139. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 2001, pp. 566–598. doi: 10.1007/3-540-44647-8_33 (cited on pages 6, 84–85).

[Deu85] David Deutsch: **Quantum theory, the Church–Turing principle and the universal quantum computer**. In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, **400**:1818 (1985), pp. 97–117 (cited on page 1).

[DJ92] David Deutsch and Richard Jozsa: **Rapid solution of problems by quantum computation**. In: *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, **439**:1907 (1992), pp. 553–558 (cited on page 1).

[DH76] Whitfield Diffie and Martin E. Hellman: **New Directions in Cryptography**. In: *IEEE Transactions on Information Theory*, **22**:6 (1976), pp. 644–654 (cited on page 1).

[DMV22] Xuan-Thanh Do, Dang-Truong Mac, and Quoc-Huy Vu: **zkSNARKs from Codes with Rank Metrics**. 2022 (cited on page 8).

[DFMS19] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner: **Security of the Fiat-Shamir Transformation in the Quantum Random-Oracle Model**. In: *Advances in Cryptology – CRYPTO 2019, Part II*. Ed. by Alexandra Boldyreva and Daniele Micciancio. Vol. 11693. Lecture Notes

in Computer Science. Springer, Heidelberg, Aug. 2019, pp. 356–383. DOI: 10.1007/978-3-030-26951-7_13 (cited on pages 84, 92–93).

[DFMS22] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner: **Online-Extractability in the Quantum Random-Oracle Model**. In: *Advances in Cryptology – EUROCRYPT 2022, Part III*. Ed. by Orr Dunkelman and Stefan Dziembowski. Vol. 13277. Lecture Notes in Computer Science. Springer, Heidelberg, May 2022, pp. 677–706. DOI: 10.1007/978-3-031-07082-2_24 (cited on page 15).

[FKMV12] Sebastian Faust, Markulf Kohlweiss, Giorgia Azzurra Marson, and Daniele Venturi: **On the Non-malleability of the Fiat-Shamir Transform**. In: *Progress in Cryptology - INDOCRYPT 2012: 13th International Conference in Cryptology in India*. Ed. by Steven D. Galbraith and Mridul Nandi. Vol. 7668. Lecture Notes in Computer Science. Springer, Heidelberg, Dec. 2012, pp. 60–79. DOI: 10.1007/978-3-642-34931-7_5 (cited on page 91).

[FKSZ+13] Serge Fehr, Jonathan Katz, Fang Song, Hong-Sheng Zhou, and Vassilis Zikas: **Feasibility and Completeness of Cryptographic Tasks in the Quantum World**. In: *TCC 2013: 10th Theory of Cryptography Conference*. Ed. by Amit Sahai. Vol. 7785. Lecture Notes in Computer Science. Springer, Heidelberg, Mar. 2013, pp. 281–296. DOI: 10.1007/978-3-642-36594-2_16 (cited on pages 171–172).

[FS09] Serge Fehr and Christian Schaffner: **Composing Quantum Protocols in a Classical Environment**. In: *TCC 2009: 6th Theory of Cryptography Conference*. Ed. by Omer Reingold. Vol. 5444. Lecture Notes in Computer Science. Springer, Heidelberg, Mar. 2009, pp. 350–367. DOI: 10.1007/978-3-642-00457-5_21 (cited on page 169).

[FLS90] Uriel Feige, Dror Lapidot, and Adi Shamir: **Multiple Non-Interactive Zero Knowledge Proofs Based on a Single Random String (Extended Abstract)**. In: *31st Annual Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Oct. 1990, pp. 308–317. DOI: 10.1109/FSCS.1990.89549 (cited on page 73).

[Fey82] Richard P Feynman: **Simulating physics with computers**. In: *International journal of theoretical physics*, **21**:6/7 (1982), pp. 467–488 (cited on page 1).

[FS87] Amos Fiat and Adi Shamir: **How to Prove Yourself: Practical Solutions to Identification and Signature Problems**. In: *Advances in Cryptology – CRYPTO'86*. Ed. by Andrew M. Odlyzko. Vol. 263. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 1987, pp. 186–194. DOI: 10.1007/3-540-47721-7_12 (cited on pages 6, 92).

[FP96] Christopher A Fuchs and Asher Peres: **Quantum-state disturbance versus information gain: Uncertainty relations for quantum information**. In: *Physical Review A*, **53**:4 (1996), p. 2038 (cited on page 37).

[GHS16] Tommaso Gagliardoni, Andreas Hülsing, and Christian Schaffner: **Semantic Security and Indistinguishability in the Quantum World**. In: *Advances in Cryptology – CRYPTO 2016, Part III*. Ed. by Matthew Robshaw and Jonathan Katz. Vol. 9816. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 2016, pp. 60–89. doi: 10.1007/978-3-662-53015-3_3 (cited on pages 3, 5, 36, 38–39, 54).

[GKS21] Tommaso Gagliardoni, Juliane Krämer, and Patrick Struck: **Quantum Indistinguishability for Public Key Encryption**. In: *Post-Quantum Cryptography - 12th International Workshop, PQCrypto 2021*. Ed. by Jung Hee Cheon and Jean-Pierre Tillich. Springer, Heidelberg, 2021, pp. 463–482. doi: 10.1007/978-3-030-81293-5_24 (cited on page 38).

[GP20] Romain Gay and Rafael Pass: **Indistinguishability Obfuscation from Circular Security**. Cryptology ePrint Archive, Report 2020/1010. https://eprint.iacr.org/2020/1010. 2020 (cited on page 26).

[GMNO18] Rosario Gennaro, Michele Minelli, Anca Nitulescu, and Michele Orrù: **Lattice-Based zk-SNARKs from Square Span Programs**. In: *ACM CCS 2018: 25th Conference on Computer and Communications Security*. Ed. by David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang. ACM Press, Oct. 2018, pp. 556–573. doi: 10.1145/3243734.3243845 (cited on page 8).

[GMP22] Alexandru Gheorghiu, Tony Metger, and Alexander Poremba: **Quantum cryptography with classical communication: parallel remote state preparation for copy-protection, verification, and more**. Cryptology ePrint Archive, Report 2022/122. https://eprint.iacr.org/2022/122. 2022 (cited on pages 7, 104, 106–110, 118, 121, 123–125, 127, 132).

[GV19] Alexandru Gheorghiu and Thomas Vidick: **Computationally-Secure and Composable Remote State Preparation**. In: *60th Annual Symposium on Foundations of Computer Science*. Ed. by David Zuckerman. IEEE Computer Society Press, Nov. 2019, pp. 1024–1033. doi: 10.1109/FOCS.2019.00066 (cited on pages 104, 106–108, 118).

[GGM84] Oded Goldreich, Shafi Goldwasser, and Silvio Micali: **How to Construct Random Functions (Extended Abstract)**. In: *25th Annual Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Oct. 1984, pp. 464–479. doi: 10.1109/SFCS.1984.715949 (cited on page 18).

[GM84] Shafi Goldwasser and Silvio Micali: **Probabilistic Encryption**. In: *Journal of Computer and System Sciences*, **28**:2 (1984), pp. 270–299 (cited on pages 2, 39).

[GO93] Shafi Goldwasser and Rafail Ostrovsky: **Invariant Signatures and Non-Interactive Zero-Knowledge Proofs are Equivalent (Extended Abstract)**. In: *Advances in Cryptology – CRYPTO'92*. Ed. by Ernest F. Brickell. Vol. 740. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 1993, pp. 228–245. doi: 10.1007/3-540-48071-4_16 (cited on page 73).

[Got03] Daniel Gottesman: **Uncloneable Encryption**. In: *Quantum Info. Comput.*, **3**:6 (Oct. 2003), pp. 581–602 (cited on page 4).

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby: **A Pseudorandom Generator from any One-way Function**. In: *SIAM Journal on Computing*, **28**:4 (1999), pp. 1364–1396 (cited on pages 18, 70).

[HMNY21] Taiga Hiroka, Tomoyuki Morimae, Ryo Nishimaki, and Takashi Yamakawa: **Quantum Encryption with Certified Deletion, Revisited: Public Key, Attribute-Based, and Classical Communication**. In: *Advances in Cryptology – ASIACRYPT 2021, Part I*. Ed. by Mehdi Tibouchi and Huaxiong Wang. Vol. 13090. Lecture Notes in Computer Science. Springer, Heidelberg, Dec. 2021, pp. 606–636. doi: 10.1007/978-3-030-92062-3_21 (cited on pages 4, 7, 104).

[HMNY22] Taiga Hiroka, Tomoyuki Morimae, Ryo Nishimaki, and Takashi Yamakawa: **Certified Everlasting Zero-Knowledge Proof for QMA**. In: *Advances in Cryptology – CRYPTO 2022, Part I*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Vol. 13507. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 2022, pp. 239–268. doi: 10.1007/978-3-031-15802-5_9 (cited on page 4).

[HHK17] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz: **A Modular Analysis of the Fujisaki-Okamoto Transformation**. In: *TCC 2017: 15th Theory of Cryptography Conference, Part I*. Ed. by Yael Kalai and Leonid Reyzin. Vol. 10677. Lecture Notes in Computer Science. Springer, Heidelberg, Nov. 2017, pp. 341–371. doi: 10.1007/978-3-319-70500-2_12 (cited on page 22).

[HU19] Dennis Hofheinz and Bogdan Ursu: **Dual-Mode NIZKs from Obfuscation**. In: *Advances in Cryptology – ASIACRYPT 2019, Part I*. Ed. by Steven D. Galbraith and Shiho Moriai. Vol. 11921. Lecture Notes in Computer Science. Springer, Heidelberg, Dec. 2019, pp. 311–341. doi: 10.1007/978-3-030-34578-5_12 (cited on page 74).

[HLW12] Susan Hohenberger, Allison B. Lewko, and Brent Waters: **Detecting Dangerous Queries: A New Approach for Chosen Ciphertext Security**. In: *Advances in Cryptology – EUROCRYPT 2012*. Ed. by David Pointcheval and Thomas Johansson. Vol. 7237. Lecture Notes in Computer Science. Springer, Heidelberg, Apr. 2012, pp. 663–681. doi: 10.1007/978-3-642-29011-4_39 (cited on page 71).

[HI19] Akinori Hosoyamada and Tetsu Iwata: **4-Round Luby-Rackoff Construction is a qPRP**. In: *Advances in Cryptology – ASIACRYPT 2019, Part I*. Ed. by Steven D. Galbraith and Shiho Moriai. Vol. 11921. Lecture Notes in Computer Science. Springer, Heidelberg, Dec. 2019, pp. 145–174. doi: 10.1007/978-3-030-34578-5_6 (cited on page 70).

[HS18a] Akinori Hosoyamada and Yu Sasaki: **Cryptanalysis Against Symmetric-Key Schemes with Online Classical Queries and Offline Quantum Computations**. In: *Topics in Cryptology – CT-RSA 2018*. Ed. by Nigel P. Smart. Vol. 10808. Lecture Notes in Computer Science. Springer, Heidelberg, Apr. 2018, pp. 198–218. doi: 10.1007/978-3-319-76953-0_11 (cited on page 4).

[HS18b] Akinori Hosoyamada and Yu Sasaki: **Quantum Demiric-Selçuk Meet-in-the-Middle Attacks: Applications to 6-Round Generic Feistel Constructions**. In: *SCN 18: 11th International Conference on Security in Communication Networks*. Ed. by Dario Catalano and Roberto De Prisco. Vol. 11035. Lecture Notes in Computer Science. Springer, Heidelberg, Sept. 2018, pp. 386–403. doi: 10.1007/978-3-319-98113-0_21 (cited on page 4).

[JLS18] Zhengfeng Ji, Yi-Kai Liu, and Fang Song: **Pseudorandom Quantum States**. In: *Advances in Cryptology – CRYPTO 2018, Part III*. Ed. by Hovav Shacham and Alexandra Boldyreva. Vol. 10993. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 2018, pp. 126–152. doi: 10.1007/978-3-319-96878-0_5 (cited on page 3).

[KLLN16a] Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia: **Breaking Symmetric Cryptosystems Using Quantum Period Finding**. In: *Advances in Cryptology – CRYPTO 2016, Part II*. Ed. by Matthew Robshaw and Jonathan Katz. Vol. 9815. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 2016, pp. 207–237. doi: 10.1007/978-3-662-53008-5_8 (cited on pages 3–4).

[KLLN16b] Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia: **Quantum Differential and Linear Cryptanalysis**. In: *IACR Transactions on Symmetric Cryptology*, **2016**:1 (2016). https://tosc.iacr.org/index.php/ToSC/article/view/536, pp. 71–94. doi: 10.13154/tosc.v2016.i1.71-94 (cited on page 4).

[KKVB02] Elham Kashefi, Adrian Kent, Vlatko Vedral, and Konrad Banaszek: **Comparison of quantum oracles**. In: *Physical Review A*, **65**:5 (2002), p. 050304 (cited on page 42).

[KV09] Jonathan Katz and Vinod Vaikuntanathan: **Smooth Projective Hashing and Password-Based Authenticated Key Exchange from Lattices**. In: *Advances in Cryptology – ASIACRYPT 2009*. Ed. by Mitsuru Matsui. Vol. 5912. Lecture Notes in Computer Science. Springer, Heidelberg, Dec. 2009, pp. 636–652. doi: 10.1007/978-3-642-10366-7_37 (cited on page 8).

[KPTZ13] Aggelos Kiayias, Stavros Papadopoulos, Nikos Triandopoulos, and Thomas Zacharias: **Delegatable pseudorandom functions and applications**. In: *ACM CCS 2013: 20th Conference on Computer and Communications Security*. Ed. by Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung. ACM Press, Nov. 2013, pp. 669–684. doi: 10.1145/2508859.2516668 (cited on page 18).

[Kil88] Joe Kilian: **Founding Cryptography on Oblivious Transfer**. In: *20th Annual ACM Symposium on Theory of Computing*. ACM Press, May 1988, pp. 20–31. doi: 10.1145/62212.62215 (cited on pages 7, 166).

[KNY21] Fuyuki Kitagawa, Ryo Nishimaki, and Takashi Yamakawa: **Secure Software Leasing from Standard Assumptions**. In: *TCC 2021: 19th Theory of Cryptography Conference, Part I*. Ed. by Kobbi Nissim and Brent Waters. Vol. 13042. Lecture Notes in Computer Science. Springer, Heidelberg, Nov. 2021, pp. 31–61. doi: 10.1007/978-3-030-90459-3_2 (cited on pages 5, 7, 104).

[KM11] Daniel Kraschewski and Jörn Müller-Quade: **Completeness Theorems with Constructive Proofs for Finite Deterministic 2-Party Functions**. In: *TCC 2011: 8th Theory of Cryptography Conference*. Ed. by Yuval Ishai. Vol. 6597. Lecture Notes in Computer Science. Springer, Heidelberg, Mar. 2011, pp. 364–381. doi: 10.1007/978-3-642-19571-6_22 (cited on page 172).

[KLR06] Eyal Kushilevitz, Yehuda Lindell, and Tal Rabin: **Information-theoretically secure protocols and security under composition**. In: *38th Annual ACM Symposium on Theory of Computing*. Ed. by Jon M. Kleinberg. ACM Press, May 2006, pp. 109–118. doi: 10.1145/1132516.1132532 (cited on page 169).

[LZ19] Qipeng Liu and Mark Zhandry: **Revisiting Post-quantum Fiat-Shamir**. In: *Advances in Cryptology – CRYPTO 2019, Part II*. Ed. by Alexandra Boldyreva and Daniele Micciancio. Vol. 11693. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 2019, pp. 326–355. doi: 10.1007/978-3-030-26951-7_12 (cited on pages 84, 92–93).

[Lo97] Hoi-Kwong Lo: **Insecurity of quantum secure computations**. In: *Physical Review A*, **56**:2 (1997), p. 1154 (cited on page 164).

[LC97] Hoi-Kwong Lo and Hoi Fung Chau: **Is quantum bit commitment really possible?** In: *Physical Review Letters*, **78**:17 (1997), p. 3410 (cited on page 164).

[LMQW22] Alex Lombardi, Ethan Mook, Willy Quach, and Daniel Wichs: **Post-Quantum Insecurity from LWE**. Cryptology ePrint Archive, Report 2022/869. https://eprint.iacr.org/2022/869. 2022 (cited on page 76).

[Mah18a] Urmila Mahadev: **Classical Homomorphic Encryption for Quantum Circuits**. In: *59th Annual Symposium on Foundations of Computer Science*. Ed. by Mikkel Thorup. IEEE Computer Society Press, Oct. 2018, pp. 332–338. doi: 10.1109/FOCS.2018.00039 (cited on page 28).

[Mah18b] Urmila Mahadev: **Classical Verification of Quantum Computations**. In: *59th Annual Symposium on Foundations of Computer Science*. Ed. by Mikkel Thorup. IEEE Computer Society Press, Oct. 2018, pp. 259–267. doi: 10.1109/FOCS.2018.00033 (cited on pages 7, 28, 103, 106, 118–119, 123).

[MPR10] Hemanta K. Maji, Manoj Prabhakaran, and Mike Rosulek: **A Zero-One Law for Cryptographic Complexity with Respect to Computational UC Security**. In: *Advances in Cryptology – CRYPTO 2010*. Ed. by Tal Rabin. Vol. 6223. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 2010, pp. 595–612. doi: 10.1007/978-3-642-14623-7_32 (cited on page 171).

[Man80] Yuri Manin: **Computable and Uncomputable**. In: *Sovetskoye Radio, Moscow*, **128**: (1980) (cited on page 1).

[May97] Dominic Mayers: **Unconditionally secure quantum bit commitment is impossible**. In: *Physical review letters*, **78**:17 (1997), p. 3414 (cited on page 164).

[MY04] Dominic Mayers and Andrew Yao: **Self Testing Quantum Apparatus**. In: *Quantum Info. Comput.*, **4**:4 (July 2004), pp. 273–286 (cited on page 104).

[MV21] Tony Metger and Thomas Vidick: **Self-Testing of a Single Quantum Device Under Computational Assumptions**. In: *ITCS 2021: 12th Innovations in Theoretical Computer Science Conference*. Ed. by James R. Lee. Vol. 185. LIPIcs, Jan. 2021, 19:1–19:12. doi: 10.4230/LIPIcs.ITCS.2021.19 (cited on pages 13–14, 104, 106, 118, 123).

[MY22] Tomoyuki Morimae and Takashi Yamakawa: **Quantum Commitments and Signatures Without One-Way Functions**. In: *Advances in Cryptology – CRYPTO 2022, Part I*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Vol. 13507. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 2022, pp. 269–295. doi: 10.1007/978-3-031-15802-5_10 (cited on page 3).

[MS16] Shahram Mossayebi and Rüdiger Schack: **Concrete Security Against Adversaries with Quantum Superposition Access to Encryption and Decryption Oracles**. In: *arXiv preprint arXiv:1609.03780*, (2016) (cited on page 38).

[MU07] Jörn Müller-Quade and Dominique Unruh: **Long-Term Security and Universal Composability**. In: *TCC 2007: 4th Theory of Cryptography Conference*. Ed. by Salil P. Vadhan. Vol. 4392. Lecture Notes in Computer Science. Springer, Heidelberg, Feb. 2007, pp. 41–60. DOI: 10.1007/978-3-540-70936-7_3 (cited on page 169).

[Ms09] Steven Myers and abhi shelat: **Bit Encryption Is Complete**. In: *50th Annual Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Oct. 2009, pp. 607–616. DOI: 10.1109/FOCS.2009.65 (cited on page 6).

[Nao90] Moni Naor: **Bit Commitment Using Pseudo-Randomness**. In: *Advances in Cryptology – CRYPTO'89*. Ed. by Gilles Brassard. Vol. 435. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 1990, pp. 128–136. DOI: 10.1007/0-387-34805-0_13 (cited on page 84).

[NY90] Moni Naor and Moti Yung: **Public-key Cryptosystems Provably Secure against Chosen Ciphertext Attacks**. In: *22nd Annual ACM Symposium on Theory of Computing*. ACM Press, May 1990, pp. 427–437. DOI: 10.1145/100216.100273 (cited on page 95).

[PsV07] Rafael Pass, abhi shelat, and Vinod Vaikuntanathan: **Relations Among Notions of Non-malleability for Encryption**. In: *Advances in Cryptology – ASIACRYPT 2007*. Ed. by Kaoru Kurosawa. Vol. 4833. Lecture Notes in Computer Science. Springer, Heidelberg, Dec. 2007, pp. 519–535. DOI: 10.1007/978-3-540-76900-2_32 (cited on page 61).

[RS20] Roy Radian and Or Sattath: **Semi-Quantum Money**. Cryptology ePrint Archive, Report 2020/414. https://eprint.iacr.org/2020/414. 2020 (cited on pages 7, 78, 104).

[Reg05] Oded Regev: **On lattices, learning with errors, random linear codes, and cryptography**. In: *37th Annual ACM Symposium on Theory of Computing*. Ed. by Harold N. Gabow and Ronald Fagin. ACM Press, May 2005, pp. 84–93. DOI: 10.1145/1060590.1060603 (cited on pages 7–8, 28, 103).

[Sah99] Amit Sahai: **Non-Malleable Non-Interactive Zero Knowledge and Adaptive Chosen-Ciphertext Security**. In: *40th Annual Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Oct. 1999, pp. 543–553. DOI: 10.1109/SFFCS.1999.814628 (cited on page 95).

[Sah01] Amit Sahai: **Simulation-Sound Non-Interactive Zero Knowledge**. In: (2001) (cited on pages 6, 84, 86).

[SW14] Amit Sahai and Brent Waters: **How to use indistinguishability obfuscation: deniable encryption, and more**. In: *46th Annual ACM Symposium on Theory of Computing*. Ed. by David B. Shmoys. ACM Press, May 2014, pp. 475–484. DOI: 10.1145/2591796.2591825 (cited on page 3).

[SSS09] Louis Salvail, Christian Schaffner, and Miroslava Sotáková: **On the Power of Two-Party Quantum Cryptography**. In: *Advances in Cryptology – ASIACRYPT 2009*. Ed. by Mitsuru Matsui. Vol. 5912. Lecture Notes in Computer Science. Springer, Heidelberg, Dec. 2009, pp. 70–87. DOI: 10.1007/978-3-642-10366-7_5 (cited on page 164).

[Shm22a] Omri Shmueli: **Public-key quantum money with a classical bank**. In: *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*. 2022, pp. 790–803 (cited on pages 7, 26–27, 103–107, 111, 134).

[Shm22b] Omri Shmueli: **Semi-quantum Tokenized Signatures**. In: *Advances in Cryptology – CRYPTO 2022, Part I*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Vol. 13507. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 2022, pp. 296–319. DOI: 10.1007/978-3-031-15802-5_11 (cited on pages 7, 103–104, 106, 157–158).

[Sho99] Peter W Shor: **Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer**. In: *SIAM review*, **41**:2 (1999), pp. 303–332 (cited on page 1).

[Sim94] Daniel R. Simon: **On the Power of Quantum Computation**. In: *35th Annual Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Nov. 1994, pp. 116–123. DOI: 10.1109/SFCS.1994.365701 (cited on page 1).

[Unr10] Dominique Unruh: **Universally Composable Quantum Multi-party Computation**. In: *Advances in Cryptology – EUROCRYPT 2010*. Ed. by Henri Gilbert. Vol. 6110. Lecture Notes in Computer Science. Springer, Heidelberg, May 2010, pp. 486–505. DOI: 10.1007/978-3-642-13190-5_25 (cited on pages 165, 169, 171–172).

[Unr13] Dominique Unruh: **Everlasting Multi-party Computation**. In: *Advances in Cryptology – CRYPTO 2013, Part II*. Ed. by Ran Canetti and Juan A. Garay. Vol. 8043. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 2013, pp. 380–397. DOI: 10.1007/978-3-642-40084-1_22 (cited on pages 165, 169, 171).

[Unr14] Dominique Unruh: **Revocable Quantum Timed-Release Encryption**. In: *Advances in Cryptology – EUROCRYPT 2014*. Ed. by Phong Q. Nguyen and Elisabeth Oswald. Vol. 8441. Lecture Notes in Computer Science. Springer, Heidelberg, May 2014, pp. 129–146. DOI: 10.1007/978-3-642-55220-5_8 (cited on page 64).

[VZ21] Thomas Vidick and Tina Zhang: **Classical Proofs of Quantum Knowledge**. In: *Advances in Cryptology – EUROCRYPT 2021, Part II*. Ed. by Anne Canteaut and François-Xavier Standaert. Vol. 12697. Lecture Notes in Computer Science. Springer, Heidelberg, Oct. 2021, pp. 630–660. doi: 10.1007/978-3-030-77886-6_22 (cited on pages 103–104).

[Wie83] Stephen Wiesner: **Conjugate coding**. In: *ACM Sigact News*, **15**:1 (1983), pp. 78–88 (cited on pages 2, 4).

[Wil11] Mark M Wilde: **From classical to quantum Shannon theory**. In: *arXiv preprint arXiv:1106.1445*, (2011) (cited on pages 124, 142).

[WZ82] William K Wootters and Wojciech H Zurek: **A single quantum cannot be cloned**. In: *Nature*, **299**:5886 (1982), pp. 802–803 (cited on pages 2, 37).

[Zha12a] Mark Zhandry: **How to Construct Quantum Random Functions**. In: *53rd Annual Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Oct. 2012, pp. 679–687. doi: 10.1109/FOCS.2012.37 (cited on pages 3, 5, 54, 70, 85, 93).

[Zha12b] Mark Zhandry: **Secure Identity-Based Encryption in the Quantum Random Oracle Model**. In: *Advances in Cryptology – CRYPTO 2012*. Ed. by Reihaneh Safavi-Naini and Ran Canetti. Vol. 7417. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 2012, pp. 758–775. doi: 10.1007/978-3-642-32009-5_44 (cited on page 12).

[Zha15] Mark Zhandry: **Cryptography in the Age of Quantum Computers**. PhD thesis. Stanford University, 2015 (cited on page 4).

[Zha16] Mark Zhandry: **A Note on Quantum-Secure PRPs**. Cryptology ePrint Archive, Report 2016/1076. https://eprint.iacr.org/2016/1076. 2016 (cited on page 70).

[Zha19a] Mark Zhandry: **How to Record Quantum Queries, and Applications to Quantum Indifferentiability**. In: *Advances in Cryptology – CRYPTO 2019, Part II*. Ed. by Alexandra Boldyreva and Daniele Micciancio. Vol. 11693. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 2019, pp. 239–268. doi: 10.1007/978-3-030-26951-7_9 (cited on pages 15–16, 37, 42, 44).

[Zha19b] Mark Zhandry: **Quantum Lightning Never Strikes the Same State Twice**. In: *Advances in Cryptology – EUROCRYPT 2019, Part III*. Ed. by Yuval Ishai and Vincent Rijmen. Vol. 11478. Lecture Notes in Computer Science. Springer, Heidelberg, May 2019, pp. 408–438. doi: 10.1007/978-3-030-17659-4_14 (cited on pages 4, 26, 103, 111, 149, 155).

Résumé : La cryptographie moderne a un ennemi de taille à l'horizon : la montée inévitable des ordinateurs quantiques. Cependant, cette même puissance de calcul permettrait également de trouver des solutions sur des tâches cryptographiques qui sont tout simplement impossibles à réaliser avec la technologie actuelle. Dans cette thèse, nous mettons les pieds dans un univers où le quantique est omniprésent en y présentant notamment deux principales contributions.

Nous mettons en avant à la fois des nouveaux modèles et de nouvelles analyses de sécurité pour deux primitives cryptographiques : les chiffrements et les preuves à divulgation nulle de connaissance non interactives. Les définitions usuelles de sécurité de ces primitives requièrent intrinsèquement la capacité d'enregistrer et de comparer des chaînes classiques. Cependant, les tâches d'enregistrement et de comparaison sont extrêmement difficiles dans le monde quantique en raison du principe d'incertitude. Nous proposons deux alternatives afin de surmonter cette barrière. De plus, nos notions de sécurité sont les premières à prendre pleinement en compte les attaques quantiques dans lesquelles les attaquants peuvent interagir avec les utilisateurs finaux sur des canaux quantiques.

D'autre part, nous montrons que la disponibilité des ordinateurs quantiques se révèle être également à l'avantage des cryptographes, même lorsque les utilisateurs finaux n'utilisent que des communications classiques. En particulier, nous présentons un protocole interactif entre une Alice classique et un Bob quantique. Ce dispositif permet à Alice d'envoyer un état quantique caché non clonable à Bob par des canaux classiques. En outre, cet état quantique non clonable établit une forte propriété dite de monogamie de l'intrication, qui décrit les limites de la force des corrélations multipartites quantiques. Enfin, nous appliquons notre protocole et nous donnons les premiers schémas semi-quantiques de protection contre la copie.

Descripteurs : Cryptographie quantique, Modèles de sécurité, Chiffrement, Preuves à divulgation nulle de connaissance, Cryptographie non clonable.

Abstract: Modern cryptography has a major foe on the horizon: the inevitable rise of quantum computers. However, the same computing power will also unlock solutions to cryptographic tasks that are simply impossible to achieve with the current technology. This thesis sets foot in a ubiquitous quantum world, where everyone will be running quantum computers, with two main contributions.

Firstly, we put forth new security models and security analyses for two cryptographic primitives: encryption and non-interactive zero- knowledge proofs. Classical security definitions of these primitives inherently require the ability to record and compare classical strings. However, the tasks of recording and comparing are highly non-trivial in the quantum setting, due to the quantum uncertainty principle. We propose two different ways to overcome this recording barrier. Our security notions are the first to fully capture quantum attacks in which the codebreakers can interact with the end-users over quantum channels.

Secondly, we show that the availability of quantum computers turns out to be also the advantage of codemakers, even when the end- users only use classical communication. In particular, we exhibit an interactive protocol between a classical Alice and a quantum Bob which allows Alice to send a hidden unclonable quantum state to Bob through classical channels. Furthermore, the constructed unclonable quantum state establishes a strong monogamy-of-entanglement property, which describes the limitations on the strength of quantum multipartite correlations. We further apply our protocol to quantum copy-protection and give the first semi-quantum copy-protection schemes.

Keywords: Quantum Cryptography, Security Models, Encryption, Zero-Knowledge Proofs, Unclonable Cryptography.

Nota : cette page, dernière de couverture, sera retournée avant reliure.