Classical Cryptographic Protocols in a Quantum World[*]

Sean Hallgren [†‡]   Adam Smith [†§]   Fang Song [¶]

Abstract

Cryptographic protocols, such as protocols for secure function evaluation (SFE), have played a crucial role in the development of modern cryptography. The extensive theory of these protocols, however, deals almost exclusively with classical attackers. If we accept that quantum information processing is the most realistic model of physically feasible computation, then we must ask: what classical protocols remain secure against quantum attackers?

Our main contribution is showing the existence of classical two-party protocols for the secure evaluation of any polynomial-time function under reasonable computational assumptions (for example, it suffices that the learning with errors problem be hard for quantum polynomial time). Our result shows that the basic two-party feasibility picture from classical cryptography remains unchanged in a quantum world.

Contents

1 Introduction

Cryptographic protocols, such as protocols for secure function evaluation (SFE), have played a crucial role in the development of modern cryptography. Goldreich, Micali and Wigderson [38], building on the development of zero-knowledge (ZK) proof systems [40, 39], showed that SFE protocols exist for any polynomial-time function under mild assumptions (roughly, the existence of secure public-key cryptosystems). Research into the design and analysis of such protocols is now a large subfield of cryptography; it has also driven important advances in more traditional areas of cryptography such as the design of encryption, authentication and signature schemes.

The extensive theory of these protocols, however, deals almost exclusively with classical attackers. However, given our current understanding of physics, quantum information processing is the most realistic model of physically feasible computation. It is natural to ask: *what classical protocols remain secure against quantum attackers?* In many cases, even adversaries with modest quantum computing capabilities, such as the ability to share and store entangled photon pairs, are not covered by existing proofs of security.

Clearly not all protocols are secure: we can rule out anything based on the computational hardness of factoring, the discrete log [65], or the principal ideal problem [45]. More subtly, the basic techniques used to reason about security may not apply in a quantum setting. For example, some information-theoretically secure two-prover ZK and commitment protocols are analyzed by viewing the provers as long tables that are fixed before queries are chosen by the verifier; quantum entanglement breaks that analysis and some protocols are insecure against colluding quantum provers (Crépeau *et al.*, [23]).

In the computational realm, *rewinding* is a key technique for basing the security of a protocol on the hardness of some underlying problem. Rewinding proofs consist of a mental experiment in which the adversary is run multiple times using careful variations of a given input. At first glance, rewinding seems impossible with a quantum adversary since running it multiple times might modify the entanglement between its internal storage and an outside reference system, thus changing the overall system's behavior.

In a breakthrough paper, Watrous [72] showed that a specific type of zero-knowledge proof (3-round, GMW-style protocols) can be proven secure using a rewinding argument tailored to quantum adversaries. Damgård and Lunemann [24] showed that a similar analysis can be applied to a variant of Blum's coin flipping protocol. Hallgren *et al.* [46] showed certain classical transformations from honest-verifier to malicious-verifier ZK can be modified to provide security against malicious quantum verifiers. Some information-theoretically secure classical protocols are also known to resist quantum attacks [20, 7, 32, 68]. Finally, there is a longer line of work on protocols that involve quantum communication, dating back to Bennett and Brassard [8]. Overall, however, little is known about how much of the classical theory can be carried over to quantum settings. See "Related Work", below, for more detail.

1.1 Our Contributions

Our main contribution is showing the existence of classical two-party protocols for the secure evaluation of any polynomial-time function under reasonable computational assumptions (for example, it suffices that the learning with errors problem [64] be hard for quantum polynomial time). Our result shows that *the basic two-party feasibility picture from classical cryptography remains unchanged in a quantum world*. The only two-party general SFE protocols which had previously been analyzed in the presence of quantum attackers required quantum computation and communication on the part of the honest participants (e.g. [21, 27]).

In what follows, we distinguish two basic settings: in the *stand-alone* setting, protocols are designed to be run in isolation, without other protocols running simultaneously; in *network* settings, the protocols must remain secure even when the honest participants are running other protocols (or copies of the same protocol) concurrently. Protocols proven secure in the *universal composability* (UC) model [13] are secure in arbitrary network settings, but UC-security is impossible to achieve in many scenarios.

Our contributions can be broken down as follows:

General Modeling of Stand-Alone Security with Quantum Adversaries. We describe a security model for two-party protocols in the presence of a quantum attackers. Proving security in this model amounts to showing that a protocol for computing a function f behaves indistinguishably from an "ideal" protocol in which f is computed by a trusted third party, which we call the ideal functionality $\mathcal{F}$. Our model is a quantum analogue of the model of stand-alone security developed by Canetti [12] in the classical setting. It slightly generalizes the existing model of Damgård et al. [27] in two ways. First, our model allows for protocols in which the ideal functionalities process quantum information (rather than only classical functionalities). Second, it allows for adversaries that take arbitrary quantum advice, and for arbitrary entanglement between honest and malicious players' inputs. Our model may be viewed as a restriction of the quantum UC model of Unruh [68] to noninteractive distinguishers, and we use that connection in our protocol design (see below). We also discuss possible variants of quantum stand-alone models and initiate a study on their relationships, which connects to interesting questions in a broad scope.

We show a sequential modular composition theorem for protocols analyzed in our model. Roughly, it states that one can design protocols modularly, treating sub-protocols as equivalent to their ideal versions when analyzing security of a high-level protocol. While the composition result of Damgård et al. [27] allows only for classical high-level protocols, our result holds for arbitrary quantum protocols.

Classical Zero-knowledge Arguments of Knowledge Secure Against Quantum Adversaries. We construct a classical zero-knowledge argument of knowledge (ZKAoK) protocol that can be proven secure in our stand-alone model. Our construction is "witness-extendable" (Lindell [50]), meaning that one can simulate an interaction with a malicious prover and simultaneously extract a witness of the statement whenever the prover succeeds. Our security proof overcomes a limitation of the previous construction of (two-party) quantum proofs of knowledge (Unruh [70]), which did not have a simulator for malicious provers. Such a simulator is important since it allows one to analyze security when using a proof of knowledge as a subprotocol. As in the classical case, our ZKAoK protocol is an important building block in designing general SFE protocols.

The main idea behind our construction is to have the prover and verifier first execute a weak coin-flipping protocol to generate a public key for a special type of encryption scheme. The prover encrypts his witness with respect to this public key and proves consistency of his ciphertext with the statement x using the ZK protocols analyzed by Watrous [72]. A simulator playing the role of the verifier can manipulate the coin-flipping phase to generate a public key for which she knows the secret key, thus allowing her to extract the witness without needing to rewind the prover. A simulator playing the role of the prover, on the other hand, cannot control the coin flip (to our knowledge) but can ensure that the public key is nearly random. If the encryption scheme satisfies additional properties (that can be realized under widely used lattice-type assumptions), we show that the verifier's view can nonetheless be faithfully simulated.

Classical UC Protocols in a Quantum Context: Towards Unruh's Conjecture. We show that a large class of protocols which are UC-secure against computationally bounded classical adversaries are also UC-secure against quantum adversaries. Unruh [68] showed that any classical protocol which is proven UC-secure against unbounded classical adversaries is also UC-secure against unbounded quantum adversaries. He conjectured (roughly, see [68] for the exact statement) that classical arguments of *computational* UC security should also go through as long as the underlying computational primitives are not easily breakable by quantum computers.

We provide support for this conjecture by describing a family of classical security arguments that go through verbatim with quantum adversaries. We call these arguments "simple hybrid arguments". They use rewinding neither in the simulation nor in any of the steps that show the correctness of simulation.[1]

---

[1] In general, it is hard to clearly define what it means for a security proof to "not use rewinding". It is not enough for the protocol to have a straight-line simulator, since the proof of the simulator's correctness might still employ rewinding. Simple

Our observation allows us to port a general result of Canetti et al. [15] to the quantum setting. We obtain the following: in the $\mathcal{F}_{ZK}$-hybrid model, where a trusted party implementing ZKAoK is available, there exist classical protocols for the evaluation of any polynomial-time function f that are UC-secure against quantum adversaries under reasonable computational assumptions. As an immediate corollary, we get a classical protocol that quantum-UC emulates the ideal functionality $\mathcal{F}_{CF}$ for coin-flipping, assuming UC-secure ZK.

New Classical UC Protocols Secure Against Quantum Attacks. We construct new two-party protocols that are UC-secure against quantum adversaries. Adapting ideas from Lindell [50], we show a *constant-round* classical coin-flipping protocol from ZK (i.e. in $\mathcal{F}_{ZK}$-hybrid model). Note that the general feasibility result from above already implies the existence of a quantum-UC secure coin-flipping protocol, but it needs polynomially many rounds. Conversely, we can also construct a *constant-round* classical protocol for ZKAoK that is UC-secure against quantum adversaries, assuming a trusted party implementing coin-flipping, i.e. in the $\mathcal{F}_{CF}$-hybrid model (essentially equivalent to the *common reference string* model, where all participants have access to a common, uniformly distributed bit string). This establishes the equivalence between $\mathcal{F}_{ZK}$ and $\mathcal{F}_{CF}$ in the quantum UC model, which may be of independent interest, e.g., in simplifying protocol designs. It has also motivated a subsequent work by Fehr et el. [34] where they showed interesting connections between ideal functionalities in the quantum-UC model in a systematic way.

Implications. The modular composition theorem in our stand-alone model allows us to get the general feasibility result below by combining our stand-alone ZKAoK protocol and the UC-secure protocols in $\mathcal{F}_{ZK}$-hybrid model:

Under standard assumptions, *there exist* classical *SFE protocols in the plain model (without a shared random string) which are stand-alone-secure against static* quantum *adversaries*. This parallels the classic result of Goldreich, Micali and Wigderson [38].

The equivalence of zero-knowledge and coin-flipping functionalities in the UC model also has interesting implications. First, the availability of a common reference string (CRS) suffices for implementing quantum-UC secure protocols. Secondly, given our stand-alone ZKAoK protocol, we get a quantum stand-alone coin-flipping protocol.

Independently of our work, Lunemann and Nielsen [54], via a different route, also showed the existence of classical two-party SFE protocols secure against quantum attacks. See the discussion at the end of "Related Work".

1.2 Related work

In addition to the previous work mentioned above, we expand here on three categories of related efforts.

Composition Frameworks for Quantum Protocols. Systematic investigations of the composition properties of quantum protocols are relatively recent. Canetti's UC framework and Pfitzmann and Waidner's closely related *reactive functionality* framework were extended to the world of quantum protocols and adversaries by Ben-Or and Mayers [5] and Unruh [67, 68]. These frameworks (which share similar semantics) provide extremely strong guarantees—security in arbitrary network environments. They were used to analyze a number of unconditionally secure quantum protocols (key exchange [6] and multi-party computation with honest majorities [7]). However, many protocols are not universally composable, and Canetti [13] showed that classical protocols cannot UC-securely realize even basic tasks such as commitment and zero-knowledge proofs without some additional setup assumptions such as a CRS or public-key infrastructure.

Damgård *et al.*[27], building on work by Fehr and Schaffner [32], proposed a general composition framework which applies only to secure quantum protocols of a particular form (where quantum communication occurs only at the lowest levels of the modular composition). As noted earlier, our model is more general and captures both classical and quantum protocols. Recently, Maurer and Renner proposed a new

---

hybrid arguments provide a clean, safe subclass of arguments that go through with quantum adversaries.

composable framework called Abstract Cryptography [55], and it has been adapted to analyzing quantum protocols as well [29].

Analyses of Quantum Protocols. The first careful proofs of security of quantum protocols were for key exchange (Mayers [57], Lo and Chau [53], Shor and Preskill [66], Beaver [2]). Research on quantum protocols for two-party tasks such as coin-flipping, bit commitment and oblivious transfer dates back farther [11, 9], though some initially proposed protocols were insecure [57]. The first proofs of security of such protocols were based on computational assumptions [28, 21]. They were highly protocol-specific and it was not known how well the protocols composed. The first proofs of security using the simulation paradigm were for information-theoretically-secure protocols for multi-party computations assuming a strict majority of honest participants [20, 22, 7]. More recently, Dupuis et al. [30, 31] constructed two-party quantum protocols for evaluating arbitrary unitary operations, which they proved secure under reasonable computational assumptions in a simulation-based definition similar to what we propose in this work. There was also a line of work on the *bounded quantum storage* model [26, 25, 32, 69] developed tools for reasoning about specific types of composition of two-party protocols, under assumptions on the size of the adversary's quantum storage. Many tools have been developed in recent years on modeling and analyzing composable security for protocols of device-independent quantum key-exchange and randomness expansion [33, 71, 58, 18].

Straight-Line Simulators and Code-Based Games. As mentioned above, we introduce "simple hybrid arguments" to capture a class of straightforward security analyses that go through against quantum adversaries. Several formalisms have been introduced in the past to capture classes of "simple" security arguments. To our knowledge, none of them is automatically compatible with quantum adversaries. For example, *straight-line black-box simulators* [49] do not rewind the adversary nor use an explicit description of its random coins; however, it may be the case that rewinding is necessary to prove that the straight-line simulator is actually correct. In a different vein, the *code-based games* of Bellare and Rogaway [4] capture a class of hybrid arguments that can be encoded in a clean formal language; again, however, the arguments concerning each step of the hybrid may still require rewinding.

Independent Work. Lunemann and Nielsen [54] independently obtained similar results to the ones described here, via a slightly different route. Specifically, they start by constructing a stand-alone coin-flipping protocol that is fully simulatable against quantum poly-time adversaries. Then they use the coin-flipping protocol to construct a stand-alone ZKAoK protocol, and finally by plugging into the GMW construction, they get quantum stand-alone-secure two-party SFE protocols as well. The computational assumptions in the two works are similar and the round complexities of the stand-alone SFE protocols are both polynomial in the security parameter. Our approach to composition is more general, however, leading to results that also apply (in part) to the UC model.

1.3 Future Directions

Our work suggests a number of straightforward conjectures. For example, it is likely that our techniques in fact apply to all the results in CLOS (multi-party, adaptive adversaries) and to corresponding results in the "generalized" UC model [16]. Essentially all protocols in the semi-honest model seem to fit the simple hybrids framework, in particular protocols based on Yao's garbled-circuits framework (e.g. [3]). It is also likely that existing proofs in security models which allow super-polynomial simulation (e.g., [62, 63, 1]) will carry through using a similar line of argument to the one here.

However, our work leaves open some basic questions: for example, can we construct constant-round ZK with negligible completeness and soundness errors against quantum verifiers? Watrous's technique does not immediately answer it since sequential repetition seems necessary in his construction to reduce the soundness error. A quick look at classical constant-round ZK (e.g., [35]) suggests that witness-indistinguishable proofs of knowledge are helpful. Is it possible to construct constant-round witness-extendable WI proofs of

knowledge? Do our analyses apply to extensions of the UC framework, such the *generalized UC* framework of Canetti et al. [16]? Finally, more generally, which other uses of rewinding can be adapted to quantum adversaries? Aside from the original work by Watrous [72], Damgård and Lunemann [24] and Unruh [70] have shown examples of such adaption.

Organization. The rest of the paper is organized as follows: Section 2 reviews basic notations and definitions. In Section 3, we propose our quantum stand-alone security model. We show our main result in Section 4. Specifically, Section 4.1 establishes quantum-UC secure protocols in $\mathcal{F}_{\mathsf{ZK}}$-hybrid model. A quantum stand-alone-secure ZKAoK protocol is developed in Section 4.2. Finally in Section 5, we discuss equivalence of $\mathcal{F}_{\mathsf{ZK}}$ and $\mathcal{F}_{\mathsf{CF}}$.

2 Preliminaries

For $m \in \mathbb{N}$, $[m]$ denotes the set $\{1, \ldots, m\}$. We use $n \in \mathbb{N}$ to denote a *security parameter*. The security parameter, represented in unary, is an implicit input to all cryptographic algorithms; we omit it when it is clear from the context. Quantities derived from protocols or algorithms (probabilities, running times, etc.) should be thought of as functions of n, unless otherwise specified. A function $f(n)$ is said to be negligible if $f = o(n^{-c})$ for any constant c, and $\mathrm{negl}(n)$ is used to denote an unspecified function that is negligible in n. We also use $poly(n)$ to denote an unspecified function $f(n) = O(n^c)$ for some constant c. When D is a probability distribution, the notation $x \leftarrow D$ indicates that x is a sample drawn according to D. When D is a finite set, we implicitly associate with it the uniform distribution over the set. If $D(\cdot)$ is a probabilistic algorithm, $D(y)$ denotes the distribution over the output of D corresponding to input y. We will sometimes use the same symbol for a random variable and for its probability distribution when the meaning is clear from the context. Let $\mathbf{X} = \{X_n\}_{n \in \mathbb{N}}$ and $\mathbf{Y} = \{Y_n\}_{n \in \mathbb{N}}$ be two ensembles of binary random variables. We call $\mathbf{X}, \mathbf{Y}$ *indistinguishable*, denoted $\mathbf{X} \approx \mathbf{Y}$, if $|\Pr(X_n = 1) - \Pr(Y_n = 1)| \leq \mathrm{negl}(n)$.

We assume the reader is familiar with the basic concepts of quantum information theory (see, e.g., [61]). We use a sans serif letter (e.g. X) to denote both a quantum register and the corresponding Hilbert space. We use X_n if we want to be specific about the security parameter. Let $\mathcal{H}_n$ denote the space for n qubits. Let $\mathsf{D}(\mathsf{X})$ be the set of density operators acting on space X and $\mathsf{L}(\mathsf{X}, \mathsf{Y})$ be the set of linear operators from space X to Y.

Quantum Machine Model. We adapt Unruh's machine model in [68] with minor changes. A *quantum interactive machine* (QIM) M is an ensemble of interactive circuits $\{M_x\}_{x \in I}$. The index set I is typically the natural numbers $\mathbb{N}$ or a set of strings $I \subseteq \{0, 1\}^*$ (or both). We give our description here with respect to $\{M_n\}_{n \in \mathbb{N}}$. For each value n of the security parameter, M_n consists of a sequence of circuits $\{M_n^{(i)}\}_{i=1,\ldots,\ell(n)}$, where $M_n^{(i)}$ defines the operation of M in one round i and $\ell(n)$ is the number of rounds for which M_n operates (we assume for simplicity that $\ell(n)$ depends only on n). We omit the scripts when they are clear from the context or are not essential for the discussion. Machine M (or rather each of the circuits that it comprises) operates on three registers: a state register S used for input and workspace; an output register O; and a network register N for communicating with other machines. The size (or running time) $t(n)$ of M_n is the sum of the sizes of the circuits $M_n^{(i)}$. We say a machine is polynomial time if $t(n) = poly(n)$ and there is a deterministic classical Turing machine that computes the description of $M_n^{(i)}$ in polynomial time on input $(1^n, 1^i)$.

When two QIMs M and M' interact, they share network register N. The circuits $M_n^{(i)}$ and $M'^{(i)}_n$ are executed alternately for $i = 1, 2, \ldots, \ell(n)$. When three or more machines interact, the machines may share different parts of their network registers (for example, a private channel consists of a register shared between only two machines; a broadcast channel is a register shared by all machines). The order in which machines are activated may be either specified in advance (as in a synchronous network) or adversarially controlled.

A non-interactive quantum machine (referred to as QTM hereafter) is a QIM M with network register empty and it runs for only one round (for all n). This is equivalent to the *quantum Turing machine* model (see [73]).

A classical interactive Turing machine is a special case of a QIM, where the registers only store classical strings and all circuits are classical. This is also called an interactive Turing machine (ITM) with advice (Canetti [12, 13]).

Indistinguishability of Quantum States. Let $\rho = \{\rho_n\}_{n\in\mathbb{N}}$ and $\eta = \{\eta_n\}_{n\in\mathbb{N}}$ be ensembles of mixed states indexed by $n \in \mathbb{N}$, where ρ_n and η_n are both $r(n)$-qubit states for some polynomially bounded function r. We first define a somewhat weak notion of indistinguishability of quantum state ensembles.

Definition 2.1 ((t, ε)-weakly indistinguishable states)**.** *We say two quantum state ensembles $\rho = \{\rho_n\}_{n\in\mathbb{N}}$ and $\eta = \{\eta_n\}_{n\in\mathbb{N}}$ are (t, ε)-weakly indistinguishable, denoted $\rho \approx_{wqc}^{t,\varepsilon} \eta$, if for every $t(n)$-time QTM $\mathcal{Z}$,*

$$|\Pr[\mathcal{Z}(\rho_n) = 1] - \Pr[\mathcal{Z}(\eta_n) = 1]| \leq \varepsilon(n).$$

The states ρ and η are called *weakly computationally indistinguishable*, denoted $\rho \approx_{wqc} \eta$, if for every polynomial $t(n)$, there exists a negligible $\varepsilon(n)$ such that ρ_n and η_n are (t, ε)-weakly computationally indistinguishable.

A stronger notion of indistinguishability of quantum states was proposed by Watrous [72, Definition 2]. The crucial distinction is that a distinguisher is allowed to take quantum advice.

Definition 2.2 ((t, ε)-indistinguishable states)**.** *We say two quantum state ensembles $\rho = \{\rho_n\}_{n\in\mathbb{N}}$ and $\eta = \{\eta_n\}_{n\in\mathbb{N}}$ are (t, ε)-indistinguishable, denoted $\rho \approx_{qc}^{t,\varepsilon} \eta$, if for every $t(n)$-time QTM $\mathcal{Z}$ and any mixed state σ_n,*

$$|\Pr[\mathcal{Z}(\rho_n \otimes \sigma_n) = 1] - \Pr[\mathcal{Z}(\eta_n \otimes \sigma_n) = 1]| \leq \varepsilon(n).$$

The states ρ and η are called *quantum computationally indistinguishable*, denoted $\rho \approx_{qc} \eta$, if for every polynomial $t(n)$, there exists a negligible $\varepsilon(n)$ such that ρ_n and η_n are (t, ε)-indistinguishable.

The two definitions above subsume classical distributions as a special case, since classical distributions can be represented by density matrices that are diagonal with respect to the standard basis.

Indistinguishability of Quantum Machines. Now we introduce the notion of distinguishing two QTMs.

Definition 2.3 ((t, ε)-indistinguishable QTMs)**.** *We say two QTMs M_1 and M_2 are (t, ε)-indistinguishable, denoted $M_1 \approx_{qc}^{t,\varepsilon} M_2$, if for any $t(n)$-time QTM $\mathcal{Z}$ and any mixed state $\sigma_n \in \mathsf{D}(\mathsf{S}_n \otimes \mathsf{R}_n)$, where R_n is an arbitrary reference system,*

$$\left|\Pr[\mathcal{Z}((M_1 \otimes \mathbb{1}_{\mathsf{L}(\mathsf{R})})\sigma_n) = 1] - \Pr[\mathcal{Z}((M_2 \otimes \mathbb{1}_{\mathsf{L}(\mathsf{R})})\sigma_n) = 1]\right| \leq \varepsilon(n).$$

Machines M_1 and M_2 are called quantum computationally indistinguishable, *denoted $M_1 \approx_{qc} M_2$, if for every polynomial $t(n)$, there exists a negligible $\varepsilon(n)$ such that M_1 and M_2 are (t, ε)-computationally indistinguishable.*

This definition is equivalent to quantum computationally indistinguishable super-operators proposed by Watrous [72, Definition 6]. If we do not restrict the running time of the distinguisher, we obtain a statistical notion of indistinguishability. Let $\mathrm{TD}(\cdot, \cdot)$ be the trace distance between density operators.

Definition 2.4 (ε-indistinguishable QTMs in diamond norm)**.** *We say two QTMs M_1 and M_2 are ε-indistinguishable in diamond norm, denoted $M_1 \approx_\diamond^\varepsilon M_2$, if for any $\sigma_n \in \mathsf{D}(\mathsf{S}_n \otimes \mathsf{R}_n)$, R being an arbitrary reference system,*

$$\mathrm{TD}[(M_1 \otimes \mathbb{1}_{\mathsf{L}(\mathsf{R})})\sigma_n, (M_2 \otimes \mathbb{1}_{\mathsf{L}(\mathsf{R})})\sigma_n] \leq \varepsilon(n).$$

QIMs M_1 and M_2 are said to be indistinguishable in diamond norm, *denoted $M_1 \approx_\diamond M_2$, if there exists a negligible $\varepsilon(n)$ such that M_1 and M_2 are ε-indistinguishable in diamond norm.*

Indistinguishability of QIMs. Next, we generalize the definitions of indistinguishability above to interactive quantum machines. Let $\mathcal{Z}$ and M be two QIMs, we denote $\langle \mathcal{Z}(\sigma), M \rangle$ as the following process: machine $\mathcal{Z}$ is initialized with σ, it then provides input to M and interacts with M. In the end, the output register of M is given to $\mathcal{Z}$ and $\mathcal{Z}$ generates one classical bit on its own output register.

Definition 2.5 ((t, ε)-indistinguishable QIMs)**.** *We say two QIMs M_1 and M_2 are (t, ε)-interactively indistinguishable, denoted $M_1 \approx_i^{t,\varepsilon} M_2$, if for any quantum $t(n)$-time interactive machine $\mathcal{Z}$ and any mixed state σ_n on $t(n)$ qubits,*

$$|\Pr[\langle \mathcal{Z}(\sigma_n), M_1 \rangle = 1] - \Pr[\langle \mathcal{Z}(\sigma_n), M_2 \rangle = 1]| \leq \varepsilon(n).$$

QIMs M_1 and M_2 are called quantum computationally interactively indistinguishable, *denoted $M_1 \approx_{qci} M_2$, if for every $t(n) \leq poly(n)$, there exists a negligible $\varepsilon(n)$ such that M_1 and M_2 are (t, ε)-interactively indistinguishable.*

We may call such $\mathcal{Z}$ an interactive distinguisher. We can likewise define *statistically interactively indistinguishable* QIMs, denoted $M_1 \approx_{qsi} M_2$, if we allow unbounded interactive distinguisher $\mathcal{Z}$.

Remark 1. Quantum interactive machines, as we defined earlier, actually can be seen as a subset of quantum strategies, formulated in [44]. Namely, a QIM is a strategy in which each channel can be implemented by a uniformly generated circuit. Therefore we can as well define statistically interactively indistinguishability using the $\| \cdot \|_{\diamond r}$ norm for quantum strategies. See Gutoski [42] and Chiribella et al. [17] for details about characterizing distinguishability of quantum strategies using the $\| \cdot \|_{\diamond r}$ norm.

Ideal functionalities. We sketch ideal functionalities, i.e., the programs of a trusted party in an ideal protocol, for a few basic cryptographic tasks.
• Commitment $\mathcal{F}_{\text{COM}}$: At "Commit" stage, Alice (the committer) inputs a bit b and Bob (the receiver) receives from $\mathcal{F}_{\text{COM}}$ a notification that a bit was committed. At "Open" stage, Alice can input the command open to $\mathcal{F}_{\text{COM}}$ who then sends Bob b.
• Oblivious Transfer $\mathcal{F}_{\text{OT}}$: Alice (the sender) inputs 2 bits (s_0, s_1) and Bob (the receiver) inputs a selection bit $c \in \{0, 1\}$. Bob receives s_c from $\mathcal{F}_{\text{OT}}$.
• Zero-knowledge $\mathcal{F}_{\text{ZK}}$: Let R_L be an NP relation. Upon receiving (x, w) from Alice, $\mathcal{F}_{\text{ZK}}$ verifies $(x, w) \overset{?}{\in} R_L$. If yes, it sends x to Bob; otherwise it instructs Bob to reject.
• Coin Flipping $\mathcal{F}_{\text{CF}}$: Alice and Bob input the request 1^n to $\mathcal{F}_{\text{CF}}$, and $\mathcal{F}_{\text{CF}}$ randomly chooses $r \leftarrow \{0, 1\}^n$ and sends it to Alice. Alice responds $\mathcal{F}_{\text{CF}}$ with "acc" or "rej" indicating continuing or aborting respectively. In the case of "acc", $\mathcal{F}_{\text{CF}}$ sends r to Bob and otherwise sends Bob $\perp$. Note the functionality is asymmetric in the sense that Alice gets the coins first. This avoids the complicated issue about fairness, which has been an active line of research in classical cryptography (see for example [19, 48, 59, 41]) and is beyond the scope of this paper.

3 Modeling Security in the Presence of Quantum Attacks

In this section, we propose a stand-alone security model for two-party protocols in the presence of quantum attacks and show a modular composition theorem in this model, which allows us to use secure protocols as ideal building blocks to construct larger protocols. We also discuss variants of our stand-alone model in a unified framework. To be self-contained, we review in Section 3.2 the quantum universal-composable (UC) security model, which is a generalization of classical UC model to the quantum setting.

3.1 A General Quantum Stand-Alone Security Model

Our model follows the *real-world/ideal-world* simulation paradigm. It proceeds in three high-level steps: (i) Formalizing the process of executing a protocol in the presence of adversarial activities. (ii) Formalizing an *ideal-world* protocol for realizing the desired task. This is an (imaginary) idealized protocol which captures the security guarantees we want to achieve. (iii) Finally we say a (real-world) protocol realizes a task securely if it "behaves similarly" to the ideal-world protocol for that task (Definition 3.3). "Behaving similarly" is formalized by the notion of stand-alone *emulation* between protocols (Definition 3.1, 3.2).

Our definition can be viewed in two ways: either as a quantum analogue of Canetti's classical stand-alone model [12] or as a relaxed notion of (a variant of) Unruh's quantum UC security [68]. Prior to our work, stand-alone security definitions for quantum attacks were largely developed *ad hoc*[2]; the first systematical treatments appear in [32, 27]. Our model generalizes the existing model of Damgård *et al.* [27] in two ways. First, our model allows protocols in which the functionalities can process quantum information (rather than only classical functionalities). Second, it allows adversaries that take arbitrary quantum advice, and for arbitrary entanglement between honest and malicious players' inputs. This distinction is reflected in the composability that the model provides (see details in Section 3.1.2). While the composition results of Damgård *et al.* allow only for classical high-level protocols, our result holds for arbitrary quantum protocols.

3.1.1 The Model

We describe our model for the two-party setting; it is straightforward to extend to multi-party setting. We first introduce a few important objects in our model. We formalize a cryptographic task by an interactive machine called a *functionality*. It contains the instructions to realize the task, and we usually denote it by $\mathcal{F}$ or $\mathcal{G}$. While our model applies to both classical and quantum functionalities, our focus in this work will be efficient classical functionalities. Namely $\mathcal{F}$ is a classical probabilistic polynomial-time machine. A two-party protocol for a task $\mathcal{F}$ consists of a pair of interactive machines (A, B). We call a protocol poly-time if (A, B) are both poly-time machines. We typically use Greek letters (e.g., Π) to denote protocols. If we want to emphasize that a protocol is classical, i.e., computation and all messages exchanged are classical, we then use lower-case letters (e.g., π). Finally, an adversary, usually denoted $\mathcal{A}$ or $\mathcal{S}$, is another interactive machine that intends to attack a protocol. Very often we abuse notation and do not distinguish a machine and the player that runs the machine. This should not cause any confusion.

Protocol Execution. We consider executing a protocol $\Pi = (A, B)$ in the presence of an adversary $\mathcal{A}$. Their state registers are initialized by a secure parameter 1^n and a joint quantum state σ_n. Adversary $\mathcal{A}$ gets activated first and coordinates the execution. Specifically, the operations of each party are:

- Adversary $\mathcal{A}$: it may either **deliver** a message to some party or **corrupt** a party. Delivering a message is simply instructing the designated party (i.e., the receiver) to read the proper segment of his network register. We assume all registers are authenticated so that $\mathcal{A}$ cannot modify them and in particular if the register is private to the party, $\mathcal{A}$ may not read the content. Other than that, $\mathcal{A}$ can for example schedule the messages to be delivered in any arbitrary way. If $\mathcal{A}$ corrupts a party, the party passes all of its internal state to $\mathcal{A}$ and follows the instructions of $\mathcal{A}$. In the two-party setting, corrupting a party can be simply thought of as substituting the machine of $\mathcal{A}$ for the machine of the corrupted party.

- Parties in Π: once a party receives a message from $\mathcal{A}$, it gets activated and runs its machine. At the end of one round, some message is generated on the network register. Adversary $\mathcal{A}$ is activated again and controls message delivery. At some round, the party generates some output and terminates.

---

[2]E.g., Fehr and Schaffner [32] write: "It is still common practice in quantum cryptography that every paper proposes its own security definition of a certain task and proves security with respect to the proposed definition. However, it usually remains unclear whether these definitions are strong enough to guarantee any kind of composability, and thus whether protocols that meet the definition really behave as expected."

Clearly, we can view Π and $\mathcal{A}$ as a whole and model the composed system as another QIM, call it $M_{\Pi,\mathcal{A}}$. Then executing Π in the presence of $\mathcal{A}$ is just running $M_{\Pi,\mathcal{A}}$ on some input state, which may be entangled with a reference system that will be handed to the distighuisher.

Protocol Emulation. As indicated earlier, a secure protocol is supposed to "emulate" an idealized protocol. Here we formally define emulation between protocols. Let Π and Γ be two protocols. Let $M_{\Pi,\mathcal{A}}$ be the composed machine of Π and an adversary $\mathcal{A}$, and $M_{\Gamma,\mathcal{S}}$ be that of Γ and another adversary $\mathcal{S}$. Informally, Π emulates Γ if the two machines $M_{\Pi,\mathcal{A}}$ and $M_{\Gamma,\mathcal{S}}$ are indistinguishable.

Definition 3.1 (Computationally Quantum-Stand-Alone Emulation). *Let Π and Γ be two poly-time protocols. We say Π computationally quantum-stand-alone (*C-QSA*) emulates Γ, if for any poly-time QIM $\mathcal{A}$ there exists a poly-time QIM $\mathcal{S}$ such that $M_{\Pi,\mathcal{A}} \approx_{qc} M_{\Gamma,\mathcal{S}}$.*

Definition 3.2 (Statistically Quantum-Stand-Alone Emulation). *Let Π and Γ be two poly-time protocols. We say Π statistically quantum-stand-alone (*S-QSA*) emulates Γ, if for any QIM $\mathcal{A}$ there exists an QIM $\mathcal{S}$ that runs in poly-time of that of $\mathcal{A}$, such that $M_{\Pi,\mathcal{A}} \approx_{\diamond} M_{\Gamma,\mathcal{S}}$.*
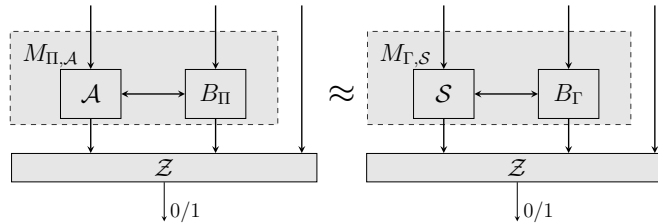

Figure 1: Quantum stand-alone emulation between protocols.

Remark 2. (i) The adversary $\mathcal{S}$ is usually called a simulator because typical constructions of $\mathcal{S}$ simulate the given $\mathcal{A}$ internally. (ii) In the statistical setting, we require the complexity of $\mathcal{S}$ and $\mathcal{A}$ to be polynomially related. This ensures that the statistical notion actually implies the computational one. See Canetti [12] for discussion of this issue in the classical context.

Ideal-world Protocol and Secure Realization of a Functionality. We formalized protocol emulation in a general form which applies to any two protocols. But it is of particular interest to emulate a special type of protocol which captures the security guarantees we want to achieve. We formalize the so-called *ideal-world* protocol $\tilde{\Pi}_{\mathcal{F}}$ for a functionality $\mathcal{F}$. In this protocol, two (dummy) parties $\tilde{A}$ and $\tilde{B}$ have access to an additional "trusted" party that implements $\mathcal{F}$. We may abuse notation and call the trusted party $\mathcal{F}$ too. Basically $\tilde{A}$ and $\tilde{B}$ invoke $\mathcal{F}$ with their inputs, and then $\mathcal{F}$ runs on the inputs and sends the respective outputs back to $\tilde{A}$ and $\tilde{B}$. An execution of $\tilde{\Pi}$ with an adversary $\mathcal{S}$ is similar to our prior description for executing a (real-world) protocol, except that $\mathcal{F}$ cannot be corrupted. Likewise, we denote the composed machine of $\mathcal{F}$ and $\tilde{\Pi}_{\mathcal{F}}$ as $M_{\mathcal{F},\mathcal{S}}$. We state the definition in the computational setting; statistical emulation is defined analogously.

Definition 3.3 (C-QSA Realization of a Functionality). *Let $\mathcal{F}$ be a poly-time two-party functionality and Π be a poly-time two-party protocol. We say Π computationally quantum-stand-alone realizes $\mathcal{F}$, if Π *C-QSA* emulates $\tilde{\Pi}_{\mathcal{F}}$. Namely, for any poly-time $\mathcal{A}$, there is a poly-time $\mathcal{S}$ such that $M_{\Pi,\mathcal{A}} \approx_{qc} M_{\mathcal{F},\mathcal{S}}$.*

It is conventional to use $\mathbf{EXEC}_{\Pi,\mathcal{A},\mathcal{Z}} := \{\mathcal{Z}((M_{\Pi,\mathcal{A}} \otimes \mathbb{1}_{L(R)})(\sigma_n))\}_{n \in \mathbb{N}}$ to denote the binary output distribution ensemble of $\mathcal{Z}$ that runs on the output state of an execution of Π and $\mathcal{A}$ with input $(1^n, \sigma_n)$. Likewise, $\mathbf{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}} := \{\mathcal{Z}((M_{\mathcal{F},\mathcal{S}} \otimes \mathbb{1}_{L(R)})(\sigma_n))\}_{n \in \mathbb{N}}$ denotes the binary output distribution ensemble of $\mathcal{Z}$ in an execution of the ideal-world protocol $\tilde{\Pi}_{\mathcal{F}}$. Definition 3.3 can be restated as requiring
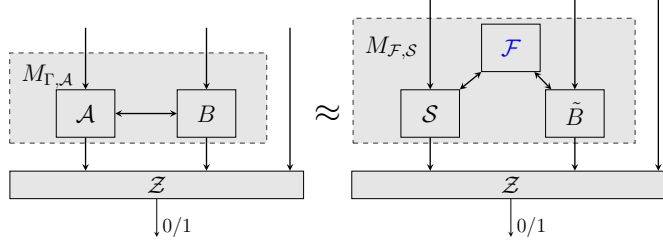
Figure 2: Quantum Stand-alone Realization of a functionality.

that for any poly-time $\mathcal{A}$ there exists a poly-time $\mathcal{S}$ such that, for any poly-time $\mathcal{Z}$ and state σ_n, we have $\mathbf{EXEC}_{\Pi,\mathcal{A},\mathcal{Z}} \approx \mathbf{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}}$.

Types of Attack. Typically, we need to speak of security against a specific class of adversaries. We have distinguished two classes of adversaries according to their computational complexity, i.e., poly-time vs. unbounded time. We also categorize adversaries according to how they corrupt the parties and how they deviate from the honest behavior defined by the protocol. The standard two types of corruptions considered in the literature are *static* vs. *adaptive* corruptions. Under static corruption, the identities of corrupted parties are determined before protocol starts. In contrast, adaptive corruption allows an adversary to change the party to corrupt adaptively during the execution. This work only concerns static corruption.

In terms of what dishonest behaviors are permitted for an adversary, again two classes are considered standard in the literature: *semi-honest* (a.k.a. *passive or honest-but-curious*) and *malicious* (a.k.a. *active*). A semi-honest adversary, after corrupting a party, still follows the party's circuit, except that in the end it processes the output and the state of the party. A malicious adversary, however, can substitute any circuit for the corrupted party. In the definitions of the protocol emulation, unless otherwise specified, the two adversaries in the real-world and ideal-world must belong to the same class. For example, if $\mathcal{A}$ is semi-honest, $\mathcal{S}$ must also be semi-honest.

These notions of different classes of adversaries naturally extend to quantum adversaries, except for one subtlety in defining semi-honest quantum adversaries. There are two possible definitions. One definition, which may be referred to as the Lo-Chau-Mayers semi-honest model [52, 56], allows $\mathcal{A}$ to run the circuit of the corrupted party, which is specified by the protocol, *coherently*. Namely $\mathcal{A}$ purifies the circuit of corrupted party so that all operations are unitary. For example, instead of measuring a quantum state, the register is "copied" by a CNOT operation to an ancillary register. Another definition forces the adversary to exactly faithfully follow the corrupted party's circuit during the protocol execution, so that any quantum measurement occurs instantaneously and possibly destructively. In other words, in the second model, a semi-honest quantum adversary $\mathcal{A}$ only corrupts a party at the end of the protocol execution, and then processes the internal state and transcript that the corrupted party holds. This second model is generally weaker than the first, in the sense that the adversary is more restricted. In this paper, we focus on the second of these two notions.

3.1.2 Modular Composition Theorem

It is common practice in the design of large protocols to divide a task into several subtasks. We first realize each subtask, and then use these modules as building blocks (subroutines) to realize the initial task. In this section, we show that our definition allows such modular design.

Composition Operation. Let Π be a protocol that uses another protocol Γ as a subroutine, and let Γ' be a protocol that QSA emulates Γ. We define the *composed* protocol, denoted $\Pi^{\Gamma/\Gamma'}$, to be the protocol in which each invocation of Γ is replaced by an invocation of Γ'. We allow multiple calls to a subroutine and

also using multiple subroutines in a protocol Π. However, we require that at any point, only **one** subroutine call be in progress; that is, we handle *sequential* composition. This is weaker than the "network" setting, where many instances and subroutines may be executed *concurrently*.

We can show that our quantum stand-alone model admits a modular composition theorem.

Theorem 3.4 (Modular Composition: General Statement). *Let Π, Γ and Γ' be two-party protocols such that Γ'* C-QSA *(resp.* S-QSA*) emulates Γ, then $\Pi^{\Gamma/\Gamma'}$* C-QSA *(resp.* S-QSA*) emulates Π.*

The proof can be found in Appendix A. Here we discuss an important type of protocol where the composition theorem is especially useful.

Protocols in a Hybrid Model. We next define a *hybrid* model, in which the parties can make calls to an ideal-world protocol $\tilde{\Pi}_{\mathcal{G}}$ of some functionality $\mathcal{G}$[3]. We call such a protocol a $\mathcal{G}$-*hybrid* protocol, and denote it $\Pi^{\mathcal{G}}$. The execution of a hybrid-protocol in the presence of an adversary $\mathcal{A}$ proceeds in the usual way.

Now assume that we have a protocol Γ that realizes $\mathcal{G}$ and we have designed a $\mathcal{G}$-hybrid protocol $\Pi^{\mathcal{G}}$ realizing another functionality $\mathcal{F}$. Then the composition theorem allows us to treat sub-protocols as equivalent to their ideal versions when analyzing security of a high-level protocol.

Corollary 3.5 (Modular Composition: Realizing Functionalities). Let $\mathcal{F}$ and $\mathcal{G}$ be poly-time functionalities. Let $\Pi^{\mathcal{G}}$ be a $\mathcal{G}$-hybrid protocol that C-QSA (resp. S-QSA) realizes $\mathcal{F}$, and Γ be a protocol that C-QSA (resp. S-QSA) realizes $\mathcal{G}$, then $\Pi^{\mathcal{G}/\Gamma}$ C-QSA (resp. S-QSA) realizes $\mathcal{F}$.
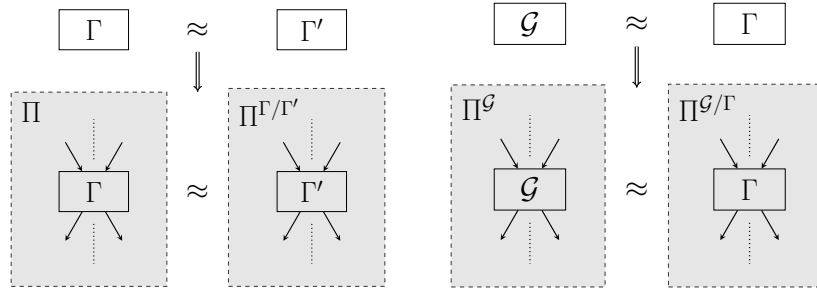

Figure 3: Illustration of modular composition theorem: the general case (left) and in hybrid model (right).

3.1.3 Variants of Quantum Stand-Alone Models: A Unified Framework

When defining a security model, there are lots of choices qualifying and quantifying the power of the adversaries to account for various security requirements. Here we provide an abstract stand-alone model for both classical and quantum cryptographic protocols, illustrated in Figure 4, which contains three natural choices for the adversaries which we think are essential. This abstract model captures all existing stand-alone security models (including ours) and this allows for a unified study of, and comparison among, these models. The relationship between these models may be interesting beyond the study of SFE.

The model contains an environment $\mathcal{Z}$ and a protocol. Depending on whether the protocol is in real or ideal world, we have the honest party, the (real-or ideal-world) adversary and possibly the trusted party. Here we think of the environment as two separate machines: $\mathcal{Z}_1$, which we may call an input sampler, prepares inputs to the players; and $\mathcal{Z}_2$ that receives outputs and makes the decision. Now we consider the following choices:

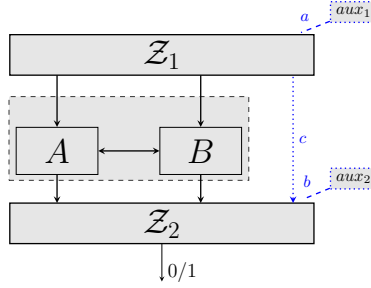[3]In contrast, we call it the *plain model* if no such trusted set-ups are available.

Figure 4: Possible choices in defining a security model

(a) Does $\mathcal{Z}_1$, the input sampler, have a quantum advice aux_1? In other words, do we allow arbitrary input states or only states that can be generated efficiently?

(b) Does $\mathcal{Z}_2$, which is essentially a distinguisher, take quantum advice aux_2?

(c) Does $\mathcal{Z}_1$ pass a state to $\mathcal{Z}_2$? Namely, does the environment keep state during the execution?

Notice that positive answers potentially give more power to the adversaries and thus provides stronger security guarantee. Also note that all machines are always allowed to take classical advice. We may denote a security model as $\mathcal{M}_{\cdot,\cdot,\cdot}$ where the subscripts are from $\{aux_1, \overline{aux_1}, aux_2, \overline{aux_2}, state, \overline{state}\}$ indicating each of the choices made for the model. For example $\mathcal{M}_{aux_1, \overline{aux_2}, state}$ corresponds to the model that $\mathcal{Z}_1$ gets quantum advice; $\mathcal{Z}_2$ takes no quantum advice and $\mathcal{Z}_1$ passes state to $\mathcal{Z}_2$ –this exactly leads to our model in Def. 3.1. Similarly, $\mathcal{M}_{aux_1, aux_2, state}$ is the model where $\mathcal{Z}_1$ and $\mathcal{Z}_2$ both take quantum advice, and there is state passing from $\mathcal{Z}_1$ to $\mathcal{Z}_2$.

We say two models $\mathcal{M}$ and $\mathcal{M}'$ are equivalent if for any two protocols Π and Γ, it holds that Π emulates Γ in $\mathcal{M}$ *if and only if* Π emulates Γ in $\mathcal{M}'$. It is conceivable that some of the $2^3 = 8$ combinations collapse to the same model. For example, if all players are classical circuits, then all eight models $\mathcal{M}_{\cdot,\cdot,\cdot}$ collapse. This is because classical (non-uniform) machines can only measure a quantum state in computational basis to obtain a classical string from a certain distribution. But a classical circuit can be hardwired with any classical string, and so (quantum) advice gives no extra power to a classical circuit. Passing state likewise becomes vacuous.

When we consider an adversary and environment consisting of quantum circuits, the situation becomes generally more complicated. We can observe that choice (b) becomes irrelevant once we permit arbitrary input state and state passing (i.e., $\mathcal{M}_{aux_1, aux_2, state} \equiv \mathcal{M}_{aux_1, \overline{aux_2}, state}$). We conjecture that state passing makes no difference either. If this is indeed true, then all the variants collapse when $\mathcal{Z}_1$ takes quantum advice. On the other hand, if $\mathcal{Z}_1$ takes no advice (i.e. only efficiently generated input states are allowed), we are left with two variants $\mathcal{M}_{\overline{aux_1}, aux_2, \cdot}$ and $\mathcal{M}_{\overline{aux_1}, \overline{aux_2}, \cdot}$. The relationship between these two models is closely related to the fundamental question in quantum complexity theory regarding $\mathrm{BQP}/\mathrm{poly} \overset{?}{=} \mathrm{BQP}/\mathbf{q}\mathrm{poly}$. We leave further investigations as future work. In Appendix B, we discuss another variant that appears in the literature [27, 32], in which $\mathcal{Z}_1$ may only generate input states of a special form. We show that this does not change the model in the case that $\mathcal{Z}_1$ takes quantum advice.

3.2 Quantum UC Model: An Overview

So far, our security model falls into the *stand-alone* setting, where protocols are assumed to be executed in isolation. However, in practice we often encounter a *network* setting, where many protocols are running concurrently. A protocol proven secure according to a stand-alone security definition ensures nothing if we

run it in a network environment. In view of this issue, Canetti [13] proposed the (classical) Universally Composable (UC) security model. It differs from the stand-alone definition of security in that the environment is allowed to be *interactive*: during the execution of the protocol, the environment may provide inputs and receive the outputs of the honest players, and exchange arbitrary messages with the adversary. In contrast, the environment in the stand-alone model runs only at the end of the protocol execution (and, implicitly, before the protocol starts, to prepare the inputs to all parties). UC-secure protocols enjoy a property called *general* (or *universal*) *composition*[4]: loosely speaking, the protocol remains secure even if it is run concurrently with an unbounded number of other arbitrary protocols (whereas proofs of security in the stand-alone model only guarantee security when only a single protocol at a time is running).

Earlier work on defining UC security and proving universal composition in the quantum setting appears in [5, 67]. We will adapt the somewhat simpler formalism of Unruh [68].

Modulo a small change in Unruh's model (quantum advice, discussed below), our stand-alone model is exactly the restriction of Unruh's model to a *non-interactive* environment, that is one which is idle from the start to the finish of the protocol. The only apparent difference is that in the UC model, the environment runs for some time before the protocol starts to prepare inputs, while in Section 3.1 we simply quantify over all joint states σ of the honest players' and adversary's inputs. This difference is only cosmetic, though: the state σ can be taken to be the joint state of the outputs and internal memory of the environment at the time the protocol begins.

We make one change to Unruh's model in order to be consistent with our earlier definitions and the work of Watrous on zero-knowledge [72]: we allow the environment to take quantum advice, rather than only classical advice. In the language of [68, p. 11], we change the initialization phase of a network execution to create a state $\rho \in \mathcal{P}(\mathcal{H}_\mathbf{N})$ which equals the classical string $|(\varepsilon, \texttt{environment}, \varepsilon)\rangle$ in $\mathcal{H}^{class}$ (instead of $|(\varepsilon, \texttt{environment}, z)\rangle)$, and an arbitrary state σ in $\mathcal{H}^{quant}$ (instead of $|\varepsilon\rangle$). Here ε denotes the empty string. Moreover, we change the definition of *indistinguishable networks* [68, p. 12] to quantify over all states σ instead of all classical strings z. This change is not significant for statistical security, since an unbounded adversary may reconstruct a quantum advice state from a (exponentially long) classical description. However, it may be significant for polynomial-time adversaries: it is not known how much quantum advice affects the power of, say BQP, relative to classical advice. For completeness, we state this modified definition of quantum UC security below.

Definition 3.6 (Computationally Quantum-UC Emulation). *Let Π and Γ be two-party protocols. We say Π computationally quantum-UC (`C-QUC`) emulates Γ, if for any poly-time QIM $\mathcal{A}$, there is a poly-time QIM $\mathcal{S}$ such that $M_{\Pi,\mathcal{A}} \approx_{qci} M_{\Gamma,\mathcal{S}}$ (as per Def. 2.5).*

Here $M_{\Pi,\mathcal{A}}$ (and $M_{\Gamma,\mathcal{A}}$ likewise) denotes the composed system of Π and $\mathcal{A}$, which can be viewed as a QIM. Its network register consists of part of the adversary's network register, and is used for external communication with another party (e.g., an environment). Alternatively, define $\mathbf{EXEC}_{\Pi,\mathcal{A},\mathcal{Z}} := \{\langle \mathcal{Z}(\sigma_n), M_{\Pi,\mathcal{A}}\rangle\}_{n\in\mathbb{N}, \sigma_n\in\mathrm{D}(\mathcal{H}_n)}$ and $\mathbf{EXEC}_{\mathcal{F},\mathcal{S},\mathcal{Z}} := \{\langle \mathcal{Z}(\sigma_n), M_{\Gamma,\mathcal{S}}\rangle\}_{n\in\mathbb{N}, \sigma_n\in\mathrm{D}(\mathcal{H}_n)}$. We can rephrase the condition as "*for any poly-time QIM $\mathcal{A}$, there is a poly-time QIM $\mathcal{S}$, such that for any poly-time QIM $\mathcal{Z}$,* $\mathbf{EXEC}_{\Pi,\mathcal{A},\mathcal{Z}} \approx \mathbf{EXEC}_{\Gamma,\mathcal{S},\mathcal{Z}}$."

If we allow $\mathcal{A}$ and $\mathcal{Z}$ to be unbounded machines, i.e., we require that $M_{\Pi,\mathcal{A}} \approx_{qsi} M_{\Gamma,\mathcal{S}}$, then we get the notion of statistically quantum-UC (`S-QUC`) emulation. As suggested in [43], we can also use the $\|\cdot\|_{\diamond r}$ norm on strategies to define it. Namely, we require that for any $\mathcal{A}$ there exists $\mathcal{S}$ such that $\|M_{\Pi,\mathcal{A}} - M_{\Gamma,\mathcal{S}}\|_{\diamond r} \leq \mathrm{negl}(n)$.

General (Concurrent) Composition. The most striking feature of UC model is that it admits a very general

---

[4]There is a distinction between UC security (a definition that may be satisfied by a specific protocol and ideal functionality) and universal composition (a property of the class of protocols that satisfy a security definition). Not all definitions that admit universal composition theorems are equivalent to UC security. See [47, 51] for discussion.

form of composition, concurrent composition[5]. Specifically, consider a protocol Π that makes subroutine calls to a protocol Γ. In contrast to the stand-alone setting, we now allow multiple instances of Γ running concurrently. (For a formal description of general composition operation, see Canetti [13].) As before, we write $\Pi^{\Gamma/\Gamma'}$ to denote the protocol obtained by Π by substituting Γ' for subroutine calls to Γ.

Our modifications of Unruh's definition do not affect the validity of the *universal composition* theorem:

Theorem 3.7 (Quantum UC Composition Theorem (Unruh [68])). *Let Π, Γ and Γ' be poly-time protocols. Assume that Γ quantum-UC emulates Γ'. Then $\Pi^{\Gamma/\Gamma'}$ quantum-UC emulates Π.*

There is another useful property that simplifies the proof of UC emulation. In both classical and quantum UC models, it suffices to consider a special adversary, which is called the *dummy* adversary $\mathcal{A}_{dummy}$. The dummy adversary $\mathcal{A}_{dummy}$ just forwards messages between a protocol and an environment and leaves any further processing to the environment. Here we only restate the completeness of dummy adversary in the quantum setting:

Theorem 3.8 (Completeness of the dummy adversary (Unruh [68, Lemma 12])). *Assume that Π quantum-UC emulates Γ with respect to the dummy adversary (i.e., instead of quantifying over all adversaries, we fix $\mathcal{A} := \mathcal{A}_{dummy}$). Then Π quantum-UC emulates Γ. This holds both for computational and statistical settings.*

4 Classical Protocols with Quantum Security

This section studies what classical protocols remain secure against *quantum* attacks in the computational setting. Let $\mathcal{F}$ be a classical two-party poly-time functionality. For technical reasons, $\mathcal{F}$ needs to be *well-formed*. See [13, 15] for a formal definition and discussions. Throughout this paper, we only consider well-formed functionalities as well. Classically, there are two important families of secure protocols:

- Stand-alone secure computation [38]: Assuming the existence of enhanced trapdoor permutations, there exists poly-time protocols that computationally stand-alone emulates $\mathcal{F}$.

- Universal-composable secure computation [15]: Assuming the existence of enhanced trapdoor permutations, there exists protocols in the $\mathcal{F}_{\mathsf{ZK}}$-hybrid model that computationally UC emulates $\mathcal{F}$.

Our main result shows that these general feasibility results largely remain unchanged against quantum attacks:

Theorem (Informal). *For any classical two-party functionality $\mathcal{F}$, there exists a classical protocol π that quantum computationally stand-alone emulates $\mathcal{F}$, under suitable quantum-resistant computational assumptions.*

The proof of the theorem can be broken into two parts. First we show a quantum analogue of [15] in Section 4.1. Namely, there exist functionalities, such as $\mathcal{F}_{\mathsf{ZK}}$, that are as powerful as to realizing any other functionalities based on them, even with respect to computationally quantum-UC security. To achieve this, we develop a framework called *simple hybrid arguments* in Sect 4.1.1 to capture a large family of classical security analyses that go through against quantum adversaries. As a result, it amounts to design a (stand-alone) secure protocol for $\mathcal{F}_{\mathsf{ZK}}$, which is the content of Section 4.2. We stress that security of existing protocols for $\mathcal{F}_{\mathsf{ZK}}$ relies on a sophisticated rewinding argument, and it is not clear if the arguments are still valid against quantum adversaries. Hence we need new ideas to get around this difficulty.

---

[5]People often refer to this type of composition as UC composition, presumably because security in the UC model implies generally concurrent composition. This should not cause any further confusion.

4.1 Basing Quantum UC Secure Computation on $\mathcal{F}_{ZK}$

We show here that $\mathcal{F}_{ZK}$ is sufficient for UC secure computation of any two-party functionality against any computational bounded quantum adversaries. That is, for any well-formed functionalities $\mathcal{F}$, there exists an $\mathcal{F}_{ZK}$-hybrid protocol that C-QUC emulates $\mathcal{F}$. We stress that these protocols are all *classical*, which can be implemented efficiently with classical communication and computation devices.

Theorem 4.1. *Let $\mathcal{F}$ be a two-party functionality. Under Assumptions 1 and 2, there exists a classical $\mathcal{F}_{ZK}$-hybrid protocol that C-QUC emulates $\mathcal{F}$ in the presence of polynomial-time malicious quantum adversaries with static corruption.*

Assumption 1. There exists a classical pseudorandom generator secure against any poly-time quantum distinguisher.

Based on this assumption and the construction of [60], we can obtain a statistically binding and quantum computationally hiding commitment scheme π_{com}. All commitment schemes we use afterwards refer to this one. This assumption also suffices for Watrous's ZK proof system for any NP-language against quantum attacks.

Assumption 2. There exists a *dense* classical public-key crypto-system that is IND-CPA (chosen-plaintext attack) secure against quantum distinguishers.

A public-key crypto-system is dense if a valid public key is indistinguishable in quantum poly-time from a uniformly random string of the same length. Although it is likely that standard reductions would show that Assumption 2 implies Assumption 1, we chose to keep the assumptions separate because the instantiation one would normally use of the pseudorandom generator would not be related to the public-key system (instead, it would typically be based on a symmetric-key block or stream cipher). Both assumptions hold, for instance, assuming the hardness of *leaning with errors* (LWE) problem [64].

4.1.1 Simple Hybrid Argument.

Our analysis is based on a new abstraction called a *simple hybrid argument* (SHA). It captures a family of classical security arguments in the UC model which remains valid in the quantum setting (as long as the underlying primitives are secure against quantum adversaries).

Definition 4.2 (Simply related machines)**.** *We say two QIMs M_a and M_b are (t, ε)-simply related if there is a time-t QTM M and a pair of classical distributions (D_a, D_b) such that*

(a) $M(D_a) \equiv M_a$ *(for two QIMs N_1 and N_2, we say $N_1 \equiv N_2$ if the two machines behave identically on all inputs, that is, if they can be described by the same circuits),*

(b) $M(D_b) \equiv M_b$, *and*

(c) $D_a \approx_{qc}^{2t, \varepsilon} D_b$.

Example 1. Figure 5 illustrates a pair of simply related machines.

Lemma 4.3. *If two machines M_a and M_b are (t, ε)-simply related, then $M_a \approx_{qci}^{t, \varepsilon} M_b$, i.e., they are (t, ε)-interactively indistinguishable (as per Definition 2.5).*

Proof. By definition, $M_a \equiv M(D_a)$ and $M_b \equiv M(D_b)$. If there is a $\mathcal{Z}$ with quantum advice σ that distinguishes M_a and M_b with advantage $\varepsilon' > \varepsilon$ in time t, we can construct a time-$2t$ distinguisher $\mathcal{D}$ for D_a and D_b with advantage ε' as well. This contradicts $D_a \approx_{qc}^{2t, \varepsilon} D_b$. Distinguisher $\mathcal{D}$ works by taking an input
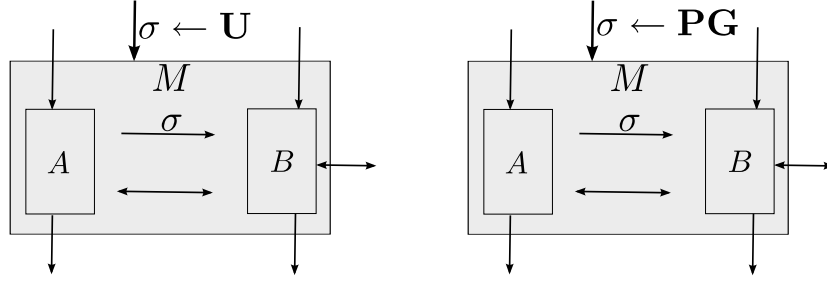
Figure 5: Two simply related machines: M_a is machine M on input σ chosen uniformly at random; M_b is machine M on input a pseudorandom string $\mathbf{PG}(r)$.

sample d from either D_a or D_b, simulates $\langle \mathcal{Z}(\sigma), M(d) \rangle$, and outputs whatever $\mathcal{Z}$ outputs. Obviously, $\mathcal{D}$ runs in time at most $2t$ and distinguishes D_a and D_b with the same advantage that $\mathcal{Z}$ distinguishes M_a and M_b. Thus we conclude $|\Pr(\langle \mathcal{Z}(\sigma), M_a \rangle = 1) - \Pr(\langle \mathcal{Z}(\sigma), M_b \rangle = 1)| \leq \varepsilon$ for any time-t environment $\mathcal{Z}$. $\qquad\square$

Definition 4.4 (Simple hybrid argument). *Two machines M_0 and M_ℓ are related by a (t,ε)-simple hybrid argument of length ℓ if there is a sequence of intermediate machines $M_1, M_2, ..., M_{\ell-1}$ such that each adjacent pair M_{i-1}, M_i of machines, $i = 1, \ldots, \ell$, is $(t, \frac{\varepsilon}{\ell})$-simply related.*

Lemma 4.5. *For any t, ε and ℓ, if two machines are related by a (t,ε)-simple hybrid argument of length ℓ, then the machines are (t,ε)-interactively indistinguishable.*

Proof. This is by a standard hybrid argument. Suppose, for contradiction, there exists a time-t machine $\mathcal{Z}$ with advice σ such that

$$|\Pr(\langle \mathcal{Z}(\sigma), M_0 \rangle = 1) - \Pr(\langle \mathcal{Z}(\sigma), M_\ell \rangle = 1)| \geq \varepsilon .$$

Then by triangle inequality we can infer that there must exist some i such that

$$|\Pr(\langle \mathcal{Z}(\sigma), M_i \rangle = 1) - \Pr(\langle \mathcal{Z}(\sigma), M_{i+1} \rangle = 1)| > \varepsilon/\ell .$$

However, by assumption M_i and M_{i+1} are $(t, \frac{\varepsilon}{\ell})$-simply related and in particular no time-t machines can distinguish them with advantage greater than ε/ℓ. $\qquad\square$

4.1.2 Lifting CLOS to Quantum UC Security.

Now we apply our *simple hybrid argument* framework to analyze the protocol in CLOS. We first review the structure of the construction of CLOS in the static setting:

(a) Let $\mathcal{F}$ be a two-party functionality. Design a protocol π that computationally (classical) UC (`C-CUC`) emulates $\mathcal{F}$ against semi-honest adversaries. The protocol uses a semi-honest oblivious transfer (ShOT) protocol, which can be constructed assuming existence of enhanced trapdoor permutations.

(b) Let $\mathcal{F}_{\text{CP}}$ be the "commit-and-prove" functionality of [15, Figure 8]. A protocol is constructed in $\mathcal{F}_{\text{ZK}}$-hybrid model that `C-CUC` emulates $\mathcal{F}_{\text{CP}}$, assuming existence of a statistically binding and computationally hiding commitment scheme. Such a commitment scheme in turn can be constructed from a pseudorandom generator [60].

18

(c) In $\mathcal{F}_{\text{CP}}$-hybrid model, a generic complier COMP is designed. Let $\pi' = \text{COMP}(\pi)$ be the $\mathcal{F}_{\text{CP}}$-hybrid protocol after compilation. It is shown in [15, Proposition 8.1] that: *for every classical adversary $\mathcal{A}'$, there exists a classical adversary $\mathcal{A}$ with running time polynomial in that of $\mathcal{A}'$ such that* $\mathbf{EXEC}_{\pi',\mathcal{A}',\mathcal{Z}} \equiv \mathbf{EXEC}_{\pi,\mathcal{A},\mathcal{Z}}$. That is, the interaction of $\mathcal{A}'$ with honest players running π' is *identical* to the interaction of $\mathcal{A}$ with π in the semi-honest model, i.e., $M_{\pi',\mathcal{A}'} \equiv M_{\pi,\mathcal{A}}$.

It then follows that, by the UC composition theorem, π' C-CUC emulates $\mathcal{F}$ in the $\mathcal{F}_{\text{ZK}}$-hybrid model.

We then show how to make the construction secure against quantum adversaries using proper quantum-resistant assumptions. The key observation is that the security proofs of the semi-honest protocol and of the $\mathcal{F}_{\text{CP}}$ protocol in the $\mathcal{F}_{\text{ZK}}$-hybrid model fall into our simple hybrid argument framework. Thus once we augment the computational assumptions to be quantum-resistant, they immediately become secure against quantum adversaries. This is stated more precisely below.

Observation 4.6 (CLOS proof structure). In CLOS, the security proofs for the semi-honest protocol and the protocol for $\mathcal{F}_{\text{CP}}$ in $\mathcal{F}_{\text{ZK}}$-hybrid model against static adversaries consist of simple hybrid arguments with $t = poly(n)$ and $\varepsilon = \text{negl}(n)$.

Moreover, the underlying indistinguishable distributions in the CLOS arguments consist of either (*i*) switching between a real public key and a uniformly random string, (*ii*) changing the plaintext of an encryption, or (*iii*) changing the message in the commit phase of a commitment protocol.

From this observation, we get the corollary below.

Corollary 4.7 (CLOS—simple hybrids). (a) In the $\mathcal{F}_{\text{ZK}}$-hybrid model and under Assumption 1, there is a non-trivial protocol that UC-emulates $\mathcal{F}_{\text{CP}}$ in the presence of polynomial-time *malicious* static *quantum* adversaries.

(b) Let $\mathcal{F}$ be a well-formed two-party functionality. In the plain model, there is a protocol that UC-emulates $\mathcal{F}$ in the presence of polynomial-time *semi-honest* static *quantum* adversaries under Assumption 2.

Proof. Observation 4.6 tells us there are two types of proofs in CLOS, so we only have to show both can be augmented to hold against quantum adversaries. On the one hand, simple hybrid arguments in CLOS still hold if we make assumptions 1 and 2, because the underlying distributions in these hybrid experiments will remain indistinguishable against quantum distinguishers. On the other hand, we know quantum UC composition also holds by Theorem 3.7.

More specifically, for the $\mathcal{F}_{\text{CP}}$ protocol in $\mathcal{F}_{\text{ZK}}$-hybrid model, the simply hybrid machines in its proof are related by switching the messages being committed. Hence $\mathcal{F}_{\text{CP}}$ protocol remains secure against malicious static quantum adversaries under Assumption 1. In the semi-honest setting, an OT protocol can be constructed from a dense crypto-system (Assumption 2), see Goldreich [37]. Its proof consists of simply related machines that are related by either switching between a valid public key and a random string (when sender is corrupted) or switching the plaintext of an encryption (when receiver is corrupted). Therefore, this protocol C-QUC emulates $\mathcal{F}_{\text{OT}}$ against semi-honest quantum adversaries. Next in $\mathcal{F}_{\text{OT}}$-hybrid model, the construction for an arbitrary $\mathcal{F}$ is unconditionally secure, which, by Unruh's lifting theorem, remains quantum-UC secure. Hence quantum UC composition theorem gives that there is a classical protocol that C-QUC emulates $\mathcal{F}$ in the presence of semi-honest static quantum adversaries. $\square$

Combining the previous arguments we can prove Theorem 4.1.

Proof of Theorem 4.1. Fix a well-formed functionality $\mathcal{F}$ and let π be the protocol for $\mathcal{F}$ in the semi-honest model guaranteed by the second part of Corollary 4.7. Now consider $\pi' = \text{COMP}(\pi)$. We want to show that it

`C-QUC` emulates $\mathcal{F}$. Theorem 3.8 tells us that it suffices to consider the classical dummy adversary $\mathcal{A}_{dummy}$. By [15, Proposition 8.1], the interaction of the dummy adversary $\mathcal{A}_{dummy}$ with π' (in the $\mathcal{F}_{CP}$ hybrid model) is identical to the interaction of the adversary $\mathcal{A}$ with π (in the semi-honest model). By the security of π in the semi-honest model, there exists an ideal-world adversary $\mathcal{S}$ such that $M_{\mathcal{F},\mathcal{S}} \approx_{qci} M_{\pi,\mathcal{A}} \equiv M_{\pi',\mathcal{A}_{dummy}}$. Thus, π' securely emulates $\mathcal{F}$ in the $\mathcal{F}_{CP}$-hybrid model against malicious adversaries. By the quantum UC composition theorem, we can compose π' with the protocol for $\mathcal{F}_{CP}$ to get a protocol secure against malicious quantum adversaries in the $\mathcal{F}_{ZK}$-hybrid model. $\qquad\square$

4.2 Realizing $\mathcal{F}_{ZK}$ with Stand-alone Security

In this section, we construct a protocol Π_{ZK} that quantum stand-alone emulates $\mathcal{F}_{ZK}$. In the stand-alone model, $\mathcal{F}_{ZK}$ is more commonly referred to as *zero-knowledge argument of knowledge*.

We will use a dense encryption scheme $\mathcal{E} = (\textbf{Gen}, \textbf{Enc}, \textbf{Dec})$ as in Assumption 2. Note that **Enc** is a randomized algorithm and we denote by $\textbf{Enc}_{pk}(m, r)$ the encryption of a message m under a public key pk using randomness r, But unless when needed, we usually omit the randomness in the notation and write $\textbf{Enc}_{pk}(m)$. We will also need a result of Watrous's [72], where he showed that there exist classical zero-knowledge proofs for any NP language that are secure against any poly-time quantum verifiers. For completeness we give his definition (adapted to our terminology) of quantum computational zero-knowledge proof [72, Definition 7].

Definition 4.8. *An interactive proof system (P, V) for a language L is* quantum computational zero-knowledge *if, for every poly-time QIM V', there exists a poly-time QIM $\mathbf{S}_{V'}$ that satisfies the following requirements.*

(a) *The verifier V' and simulator $\mathbf{S}_{V'}$ agree on the polynomially bounded functions q and r that specify the number of auxiliary input qubits and output qubits, respectively.*

(b) *Let $M_{\langle P,V'\rangle(x)}$ be the machine describing that interaction between V' and P on input x, and let $M_{\mathbf{S}_{V'}(x)}$ be the simulator's machine on input x. Then the ensembles $\{M_{\langle P,V'\rangle(x)} : x \in L\}$ and $\{M_{\mathbf{S}_{V'}(x)} : x \in L\}$ are quantum computationally indistinguishable as per Definition 2.3.*

Now that we have all building blocks ready, our construction of a classical ZKAoK protocol is as follows.

Theorem 4.9. *Protocol Π_{ZK} quantum stand-alone-emulates $\mathcal{F}_{ZK}$.*

The full proof appears in Sect. 4.2.1. We provide a brief and intuitive justification here. Roughly speaking, Phase 1 constitutes what may be called a "semi-simulatable" coin-flipping protocol. Specifically we can simulate a corrupted Prover. This implies that a simulator $\mathcal{S}$ can "cheat" in Phase 1 and force the outcome to be a public key pk of which he knows a corresponding secret key sk, so that $\mathcal{S}$ can decrypt e to recover w in the end. This allows us to show argument of knowledge (in our stand-alone model). On the other side, although generally we cannot simulate a corrupted verifier in Phase 1, we can guarantee that the outcome pk is uniformly random if the verifier behaves honestly. This is good enough to show zero-knowledge, because we can later encrypt an all-zero string and use the simulator for the ZK protocol in Phase 2 to produce a fake proof. In reality, a corrupted verifier may bias the coin-flipping outcome by aborting dependent on Prover's message b for example. This technical subtlety, nonetheless, is not hard to deal with. Intuitively the verifier only sees "less" information about the witness if he/she decides to abort in Phase 1.

ZKAoK Protocol Π_{ZK}

Phase 1

   (a) **V** chooses $a \leftarrow \{0,1\}^n$ at random, and sends **P** a commitment of a: $c = \mathbf{comm}(a)$.

   (b) **P** sends $b \leftarrow \{0,1\}^n$ to **V**.

   (c) **V** sends **P** string a.

   (d) **V** proves to **P** that c is indeed a commitment of a using Watrous's ZK protocol.

   (e) **P** and **V** set $pk = a \oplus b$ and interpret it as a public key.

Phase 2

   (a) **P**, holding an instance x and a witness w, encrypts w under pk. Let $e = \mathbf{Enc}_{pk}(w)$. **P** sends (x, e) to **V**.

   (b) **P** proves to **V** that e encodes a witness of x using Watrous's ZK protocol. **V** accepts if it accepts in this ZK protocol. Otherwise it rejects and halts.

4.2.1   Proof of Theorem 4.9: Quantum Stand-alone Secure ZKAoK

For the sake of clarity, we propose a non-interactive notion of *simple hybrid argument*, analogous to Def. 4.4, which formalizes a common structure in stand-alone security proofs.

Definition 4.10 (Simply related non-interactive machines)**.** *We say two QTMs M_a and M_b are (t, ε)-simply related if there is a time-t QTM M and a pair of QTMs (N_a, N_b) such that*

   (a) $M^{N_a} \equiv M_a$ (for two QTMs N_1 and N_2, we say $N_1 \equiv N_2$ if they can be described by the same circuits),

   (b) $M^{N_b} \equiv M_b$, and

   (c) $N_a \approx_{qc}^{2t, \varepsilon} N_b$.

Remark 3. (i) M^N is the machine that gives M oracle access to N. (ii) As a typical example of a pair of indistinguishable QTMs, consider N_a being a QTM describing a ZK protocol with a (dishonest) verifier, and N_b being a simulator's machine. Then by definition of a valid simulator, we have $N_a \approx_{qc} N_b$. (iii) Machines (N_a, N_b) in the definition also capture pair of indistinguishable classical distributions that are efficiently samplable. Namely, we can let N_a and N_b be algorithms that sample from distributions D_a and D_b respectively.

Definition 4.11 (Simple hybrid argument (non-interactive version))**.** *Two machines M_0 and M_ℓ are related by a (t, ε)-simple hybrid argument of length ℓ if there is a sequence of intermediate machines $M_1, M_2, ..., M_{\ell-1}$ such that each adjacent pair M_{i-1}, M_i of machines, $i = 1, \ldots, \ell$, is $(t, \frac{\varepsilon}{\ell})$-simply related.*

Lemma 4.12. *For any t, ε and ℓ, if two machines are related by a (t, ε)-simple hybrid argument of length ℓ, then the machines are (t, ε)-indistinguishable.*

Proof. Suppose for contradiction, there exists a time-t QTM $\mathcal{Z}$ with advice σ such that $|\Pr[\mathcal{Z}((M_0 \otimes \mathbb{1}_{L(R)})\sigma_n) = 1] - \Pr[\mathcal{Z}((M_\ell \otimes \mathbb{1}_{L(R)})\sigma_n) = 1]| > \varepsilon$. Then by triangle inequality we can infer that there must exist some i s.t. $|\Pr[\mathcal{Z}((M_i \otimes \mathbb{1}_{L(R)})\sigma_n) = 1] - \Pr[\mathcal{Z}((M_{i+1} \otimes \mathbb{1}_{L(R)})\sigma_n) = 1]| > \varepsilon/\ell$. However,

by assumption M_i and M_{i+1} are $(t, \varepsilon/\ell)$-simply related and in particular no time-t QTMs can distinguish them with advantage greater than ε/ℓ. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\square$

Remark 4. Actually, the proof of our modular composition can be seen as a simple hybrid argument. Specifically in step 3, $M_{\Pi',\mathcal{A}}$ and $M_{\Pi,\mathcal{S}}$ are simply related by $M_{\Gamma',\mathcal{A}_{\Gamma'}}$ and $M_{\Gamma,\mathcal{A}_\Gamma}$.

We now prove Theorem 4.9 following the (non-interactive) simply hybrid argument framework.

Proof of Theorem 4.9. We denote the two ZK proof systems in Phase 1 & 2 by ZK_1 and ZK_2 respectively. The two NP languages, formalized below, are denoted by L_1 and L_2 respectively.

$$
\begin{aligned}
L_1 &= \{(c,a) : \exists r \in \{0,1\}^* \text{ s.t. } \textbf{comm}(a,r) = c\} \\
L_2 &= \{(pk,x,e) : \exists w, r \in \{0,1\}^*, \text{ s.t. } \textbf{Enc}_{pk}(w,r) = e \wedge (x,w) \in R_L\}
\end{aligned}
$$

The simulators of ZK_1 and ZK_2 are denoted by $\mathbf{S}_1$ and $\mathbf{S}_2$ respectively. We stress that Watrous's ZK protocol has negligible completeness and soundness errors, and in addition the simulator succeeds for arbitrary quantum poly-time verifiers on true instances, except with negligible probability.

Prover is Corrupted. For any real-world adversary $\mathcal{A}$, we construct an ideal-world adversary $\mathcal{S}$.

Simulator $\mathcal{S}$: Prover is corrupted

Input: $\mathcal{A}$ as a black box; security parameter 1^n.

1. $\mathcal{S}$ initializes $\mathcal{A}$ with whatever input state it receives.

2. In **Phase 1**, $\mathcal{S}$ does the following:

    (a) Compute $c = \textbf{comm}(0^n)$ and send it to $\mathcal{A}$.
    (b) Obtain $b \in \{0,1\}^n$ from $\mathcal{A}$.
    (c) Run $\textbf{Gen}(1^n)$ to obtain (pk, sk). Send $a = pk \oplus b$ to $\mathcal{A}$.
    (d) Run the simulator $\mathbf{S}_1$ for ZK_1 with input (c,a).

3. In **Phase 2**, $\mathcal{S}$ obtains (x,e) and executes ZK_2 with $\mathcal{A}$. If ZK_2 succeeds, $\mathcal{S}$ decrypts e to get $w = \textbf{Dec}_{sk}(e)$ and sends (x,w) to $\mathcal{F}_{ZK}$.

4. $\mathcal{S}$ outputs whatever $\mathcal{A}$ outputs.

Let $M_{\mathcal{F}_{ZK},\mathcal{S}}$ be the QTM of ideal-world interaction between $\mathcal{S}$, $\mathcal{F}_{ZK}$ and $\mathbf{V}$; and let $M_{\Pi_{ZK},\mathcal{A}}(P)$ describing real-world interaction between $\mathcal{A}$ and $\mathbf{V}$.

Lemma 4.13. $M_{\Pi_{ZK},\mathcal{A}(P)} \approx_{qc} M_{\mathcal{F}_{ZK},\mathcal{S}}$.

Proof. We define a sequence of machines to form a hybrid argument:
Now it is easy to see:

- $M_0 \approx_{qc} M_1$. These two QTMs would behave differently only if ZK_2 succeeds but w is not a valid witness. Namely $\mathcal{A}$ (corrupted prover) has managed to prove a false statement that e encodes a true witness. By soundness property of ZK_2, however, this only occurs with negligible probability.

- Machines $M_1, \ldots, M_4$ form a simple hybrid argument. More specifically, each adjacent pair of machines constitutes simply related machines:

Hybrid Machines: relating $M_{\Pi_{\text{ZK}},\mathcal{A}(P)}$ and $M_{\mathcal{F}_{\text{ZK}},\mathcal{S}}$

- $M_0 := M_{\mathcal{F}_{\text{ZK}},\mathcal{S}}$. Specifically, on any input state, the output has two parts: one part corresponds to the adversary $\mathcal{A}$'s output state, and the other corresponds to the dummy verifier's output, which is accepting if w obtained by $\mathcal{S}$ in step 3 is a valid witness, i.e., $(x, w) \in R_L$.

- M_1: differ from M_0 only in that M_1 always let the dummy verifier accept as long as ZK_2 succeeds.

- M_2: differs from M_1 in the message a in Phase 1: instead of sending $pk \oplus b$, in M_2, $a \leftarrow \{0,1\}^n$ is set to be a uniformly random string.

- M_3: in the first step of Phase 1, $\mathcal{M}_3$ commits to a instead of committing to 0^n.

- M_4: instead of running simulator $\mathbf{S}_1$, M_4 executes the actual ZK_1 protocol. Observe that $M_4 \equiv M_{\Pi_{\text{ZK}},\mathcal{A}(P)}$.

- M_1 and M_2 are simply related by switching valid public keys to uniformly random strings.
- M_2 and M_3 are simply related by changing the messages being committed to.
- M_3 and M_4 are simply related via a pair of indistinguishable QTMs N_a and N_b, where N_a is the simulator $\mathbf{S}_1$, and N_b is the machine describing ZK_1.

Thus $M_{\Pi_{\text{ZK}},\mathcal{A}(P)} \approx_{qc} M_{\mathcal{F}_{\text{ZK}},\mathcal{S}}$. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\square$

Verifier is Corrupted. We construct ideal world $\mathcal{S}$ for any adversary $\mathcal{A}$ that corrupts the verifier as follows:

Simulator $\mathcal{S}$: Verifier is corrupted

Input: $\mathcal{A}$ as a black box; security parameter 1^n.

1. $\mathcal{S}$ initializes $\mathcal{A}$ with whatever input state it receives.

2. Wait till get x from $\mathcal{F}_{\text{ZK}}$. Then do the following.

3. In **Phase 1**, $\mathcal{S}$ behave honestly and aborts if $\mathcal{A}$ aborts. Let the outcome be pk.

4. In **Phase 2**:

    (a) $\mathcal{S}$ picks an arbitrary string, say $0^{w(n)}$, and send $e = \mathbf{Enc}_{pk}(0^{w(n)})$ to $\mathcal{A}$.

    (b) $\mathcal{S}$ runs the simulator $\mathbf{S}_2$ for ZK_2 with input (pk, e, x).

5. $\mathcal{S}$ outputs whatever $\mathcal{A}$ outputs.

Let $M_{\mathcal{F}_{\text{ZK}},\mathcal{S}}$ be the QTM of ideal-world interaction between $\mathbf{P}, \mathcal{F}_{\text{ZK}}$ and $\mathcal{S}$; and let $M_{\Pi_{\text{ZK}},\mathcal{A}(V)}$ describing the real-world interaction between $\mathbf{P}$ and $\mathcal{A}$.

Lemma 4.14. $M_{\Pi_{\text{ZK}},\mathcal{A}(V)} \approx_{qc} M_{\mathcal{F}_{\text{ZK}},\mathcal{S}}$.

Proof. The proof again follows a hybrid argument. We define the following hybrids.

Hybrid Machines: relating $M_{\Pi_{\text{ZK}}, \mathcal{A}(V)}$ and $M_{\mathcal{F}_{\text{ZK}}, \mathcal{S}}$

- $M_0 := M_{\Pi_{\text{ZK}}, \mathcal{A}(V)}$.

- M_1: M_1 runs the simulator $\mathbf{S}_2$ instead of invoking the actual ZK_2 protocol.

- M_2: M_2 encrypts $0^{w(n)}$ instead of a valid witness w. Observe that $M_2 \equiv M_{\mathcal{F}_{\text{ZK}}, \mathcal{S}}$.

Clearly machines M_0 and M_1 are simply related via a pair of QTMs N_a and N_b, where N_a is the simulator $\mathbf{S}_2$, and N_b is the machine describing ZK_2. Hence they are quantum computationally indistinguishable. Showing indistinguishability of M_1 and M_2 slightly deviates from our simple hybrid argument framework. We will modify M_1 and M_2 to get two machines $\hat{M}_1$ and $\hat{M}_2$ which may run in super-polynomial time. We can then show that $\hat{M}_1 \approx_{qc} \hat{M}_2$[6], and that $\hat{M}_1 \approx_{qc} \hat{M}_2$ implies $M_1 \approx_{qc} M_2$.

Specifically $\hat{M}_1$ makes one change from M_1: if corrupted verifier aborts during Phase 1, $\mathbf{P}$ extracts $\hat{a}$ from c using possibly super-polynomial-time brute-force search. Because the commitment scheme is statistically binding, there is a well-defined $\hat{a}$ with overwhelming probability. In addition, soundness of ZK_1 ensures that $\hat{a} = a$ except for negligible soundness error. In this way, $\mathbf{P}$ still gets $pk := a \oplus b$ and we let $\mathbf{P}$ send $\mathbf{Enc}_{pk}(w)$ to the verifier even in case of abort. $\hat{M}_2$ is modified similarly. Namely a (super-polynomial-time) simulator extracts pk and sends $\mathbf{Enc}_{pk}(0^{w(n)})$ in case of abort.

Note that $\hat{M}_1$ and $\hat{M}_2$ are simply related by switching the plaintexts, and therefore $\hat{M}_1 \approx_{qc} \hat{M}_2$ follows by our simple hybrid argument framework. Next we claim that if $\hat{M}_1 \approx_{qc} \hat{M}_2$, then $M_1 \approx_{qc} M_2$. This is because that if there exists a distinguisher D that tells apart M_1 and M_2, then one can as well distinguish $\hat{M}_1$ from $\hat{M}_2$ by ignoring the ciphertext in case of aborting and then invoking D.

Therefore we have that $M_{\Pi_{\text{ZK}}, \mathcal{A}(V)} \approx_{qc} M_{\mathcal{F}_{\text{ZK}}, \mathcal{S}}$. □

Finally, we conclude that Theorem 4.9 holds. □

4.3 Putting It Together

Recall the results that we have obtained so far:

(a) Under Assumptions 1 and 2, for any well-formed two-party functionality $\mathcal{F}$, there is a classical protocol $\pi^{\mathcal{F}_{\text{ZK}}}$ that quantum-UC emulates $\mathcal{F}$ in the $\mathcal{F}_{\text{ZK}}$-hybrid model. (Theorem 4.1)

(b) Under Assumption 1 and 2, There exists classical protocol π_{ZK} that C-QSA emulates $\mathcal{F}_{\text{ZK}}$. (Theorem 4.9)

Applying our modular composition theorem (Theorem 3.4) to the above, we obtain the main theorem:

Theorem 4.15. *For any well-formed classical two-party functionality $\mathcal{F}$, there exists a classical protocol Π that* C-QSA *realizes $\mathcal{F}$ against malicious static quantum adversaries in the plain model, under Assumptions 1 and 2.*

---

[6]Although the machines $\hat{M}$ are not necessarily poly-time, we can still talk about distinguishing them by poly-time distinguishers according to Definition 2.3. If the output register of $\hat{M}$ exceeds the dimension of the input of the distinguisher, we assume that the distinguisher just takes an arbitrary portion that fits.

5 Equivalence Between $\mathcal{F}_{\text{ZK}}$ and $\mathcal{F}_{\text{CF}}$

We have seen that $\mathcal{F}_{\text{ZK}}$ functionality is sufficient to realize any other functionality. It is interesting to find out if this holds as well for other functionalities. More generally, we may ask what the relations of different functionalities are. In this section, we show that $\mathcal{F}_{\text{ZK}}$ and $\mathcal{F}_{\text{CF}}$ are equivalent in the sense that one can be UC-realized from the other.

Theorem 5.1 (Equivalence between $\mathcal{F}_{\text{ZK}}$ and $\mathcal{F}_{\text{CF}}$). *(a) Under Assumption 1, there is a constant-round protocol $\Pi_{\text{CF}}^{\mathcal{F}_{\text{ZK}}}$ that* C-QUC *emulates $\mathcal{F}_{\text{CF}}$ in the $\mathcal{F}_{\text{ZK}}$-hybrid model.*

(b) Under Assumptions 1 and 2, there is a constant-round protocol $\Pi_{\text{ZK}}^{\mathcal{F}_{\text{CF}}}$ that C-QUC *emulates $\mathcal{F}_{\text{ZK}}$ in the $\mathcal{F}_{\text{CF}}$-hybrid model.*

It is possible to obtain more connections between different functionalities. For example, [14] gives a ZK protocol that statistically UC and hence C-QUC emulates $\mathcal{F}_{\text{ZK}}$ in the $\mathcal{F}_{\text{COM}}$-hybrid model. On the other hand, our Theorem 4.1 implies that $\mathcal{F}_{\text{COM}}$ can be C-QUC realized in $\mathcal{F}_{\text{ZK}}$-hybrid model. Thus $\mathcal{F}_{\text{ZK}}$ and $\mathcal{F}_{\text{COM}}$ are equivalent in the computationally quantum-UC model. See [34] for a systematic study of the reducibility and characterizing functionalities in the quantum UC model.

5.1 From $\mathcal{F}_{\text{ZK}}$ to $\mathcal{F}_{\text{CF}}$

Theorem 4.1 already implies that $\mathcal{F}_{\text{CF}}$ can be C-QUC realized from $\mathcal{F}_{\text{ZK}}$-hybrid model. However, that relies on the generic construction of CLOS, which is typically not optimal in terms of the number of rounds (i.e., *round complexity*) and the amount of messages exchanged (i.e., *communication complexity*). Here we give a direct reduction which is simple and more efficient. Specifically, we show that the parallel coin-flipping protocol of Lindell [50], once executed in $\mathcal{F}_{\text{ZK}}$-hybrid model, i.e., the (stand-alone) ZKAoK protocol is replaced by the ideal protocol for $\mathcal{F}_{\text{ZK}}$, is C-QUC secure. This yields a *constant-round* protocol for $\mathcal{F}_{\text{CF}}$, and we need only Assumption 1: existence of a quantum-secure PRG. The protocol is shown below.

Coin-Flipping Protocol $\Pi_{\text{CF}}^{\mathcal{F}_{\text{ZK}}}$

1. A chooses $a \leftarrow \{0,1\}^n$ at random, and sends B a commitment of a: $c = \mathbf{comm}(a,r)$.

2. A proves knowledge of (a,r) using $\mathcal{F}_{\text{ZK}}$.

3. B sends $b \leftarrow \{0,1\}^n$ to A.

4. A sends B string a.

5. A proves to B that c is indeed a commitment of a using $\mathcal{F}_{\text{ZK}}$.

6. A and B set $s = a \oplus b$ as the outcome.

We give proofs for corrupted A and corrupted B separately.

Player A is Corrupted. We construct an ideal world $\mathcal{S}$ for any adversary $\mathcal{A}$ corrupting A.

Claim 5.2. For any $\mathcal{A}$ corrupting A, $M_{\Pi_{\text{CF}}^{\mathcal{F}_{\text{ZK}}}, \mathcal{A}} \approx_{qci} M_{\mathcal{F}_{\text{CF}}, \mathcal{S}}$.

Simulator $\mathcal{S}$: A is corrupted

Input: $\mathcal{A}$ as a black box; security parameter 1^n

1. $\mathcal{S}$ initializes $\mathcal{A}$ with whatever input state it receives from the environment.

2. $\mathcal{S}$ obtains s from $\mathcal{F}_{\mathrm{CF}}$ which is chosen uniformly at random $s \leftarrow \{0,1\}^n$.

3. $\mathcal{S}$ receives a commitment c from $\mathcal{A}$.

4. $\mathcal{A}$ shows knowledge of (a, r) to $\mathcal{F}_{\mathrm{ZK}}$, which is emulated by $\mathcal{S}$ here. $\mathcal{S}$ verifies if $c = \mathbf{comm}(a, r)$ and aborts if not. This allows $\mathcal{S}$ to learn a.

5. $\mathcal{S}$ sends $b = a \oplus s$ to $\mathcal{A}$.

6. $\mathcal{A}$ sends a to $\mathcal{S}$. $\mathcal{S}$ aborts if $\mathcal{A}$ sends some other string not equal to a.

7. $\mathcal{A}$ needs to prove that c is a valid commitment of a. It sends $((c, a), r)$ to $\mathcal{F}_{\mathrm{ZK}}$. $\mathcal{S}$ verifies them. Abort if verification fails.

8. If $\mathcal{A}$ aborts at any point, $\mathcal{S}$ aborts $\mathcal{F}_{\mathrm{CF}}$. Otherwise, instruct $\mathcal{F}_{\mathrm{CF}}$ to send s to the other (dummy) party $\tilde{B}$.

9. $\mathcal{S}$ outputs whatever $\mathcal{A}$ outputs.

Proof. Because s is chosen uniformly, $b = a \oplus s$ is also uniformly random. The adversary must behave identical in the real world and the ideal world, and the two machines will look identical from the perspective of the environment.

$\square$

Player B is Corrupted. For any real-world adversary $\mathcal{A}$ that corrupts B, we construct an ideal-world adversary $\mathcal{S}$.

Claim 5.3. For any $\mathcal{A}$ corrupting B, $M_{\Pi_{\mathrm{CF}}^{\mathcal{F}_{\mathrm{ZK}}}, \mathcal{A}} \approx_{qci} M_{\mathcal{F}_{\mathrm{CF}}, \mathcal{S}}$.

Proof. We define an intermediate machine M which behaves differently from $M_{\mathcal{F}_{\mathrm{CF}}, \mathcal{S}}$ merely in that a uniformly random string $a \leftarrow \{0,1\}^n$ is chosen and sent to $\mathcal{A}$ in M, instead of sending $a = s \oplus b$. Then observe that the only difference between M and $M_{\Pi_{\mathrm{CF}}^{\mathcal{F}_{\mathrm{ZK}}}, \mathcal{A}}$ appears in the first commitment message: M commits to 0^n while $M_{\Pi_{\mathrm{CF}}^{\mathcal{F}_{\mathrm{ZK}}}, \mathcal{A}}$ commits to a. Hence we can claim that:

- $M_{\mathcal{F}_{\mathrm{CF}}, \mathcal{S}} \equiv M$ since s is chosen uniformly at random by $\mathcal{F}_{\mathrm{CF}}$ and hence $s \oplus b$ is still uniformly random just as a in M. Thus the two machines are identical.

- $M \approx_{qci} M_{\Pi_{\mathrm{CF}}^{\mathcal{F}_{\mathrm{ZK}}}, \mathcal{A}}$ because they are simply related by changing the underlying message of a commitment.

$\square$

5.2 From $\mathcal{F}_{\mathrm{CF}}$ to $\mathcal{F}_{\mathrm{ZK}}$

We construct a classical *constant-round* protocol for $\mathcal{F}_{\mathrm{ZK}}$ in the $\mathcal{F}_{\mathrm{CF}}$-hybrid model. Our $\Pi_{\mathrm{ZK}}^{\mathcal{F}_{\mathrm{CF}}}$ protocol uses a standard transformation from a witness-indistinguishable (WI) proof system in the $\mathcal{F}_{\mathrm{CF}}$-hybrid model.

<div style="border: 1px solid black; padding: 10px;">

Simulator $\mathcal{S}$: B is corrupted

Input: $\mathcal{A}$ as a black box; security parameter 1^n

1. $\mathcal{S}$ initializes $\mathcal{A}$ with whatever input state it receives from the environment.

2. $\mathcal{S}$ obtains s from $\mathcal{F}_{\mathrm{CF}}$ which is chosen uniformly at random $s \leftarrow \{0,1\}^n$.

3. $\mathcal{S}$ computes $c = \mathbf{comm}(0^n)$ and sends it to $\mathcal{A}$.

4. $\mathcal{S}$ plays the role of $\mathcal{F}_{\mathrm{ZK}}$ and sends c to $\mathcal{A}$.

5. Obtain $b \in \{0,1\}^n$ from $\mathcal{A}$.

6. $\mathcal{S}$ sends $a = s \oplus b$ to $\mathcal{A}$.

7. $\mathcal{S}$ mimics $\mathcal{F}_{\mathrm{ZK}}$ and sends (c, a) to $\mathcal{A}$.

8. If $\mathcal{A}$ aborts at any point, $\mathcal{S}$ aborts.

9. $\mathcal{S}$ outputs whatever $\mathcal{A}$ outputs.

</div>

We give a definition for WI against quantum adversaries and show a repetition theorem analogous to the classical setting to amplify the soundness of WI protocols. We also show that Blum's 3-round ZK protocol for Hamiltonian Cycle is in fact quantum-secure WI under suitable assumptions.

Definition 5.4 (Quantum computationally witness-indistinguishable $\mathtt{QC\text{-}WI}$). *Let $\Pi = \langle P, V \rangle$ be an inter-active proof (or argument) system for a language $L \in$ NP. We say Π is* quantum computational witness-indistinguishable *for R_L, if for any polynomial-time QIM $\mathbf{V}^*$, any two collections $\{w_x^1\}_{x \in L}$ and $\{w_x^2\}_{x \in L}$ with $w_x^i \in R_L(x), i = 1, 2$, the two machines $M_1 := \{M_{w_x^1, \mathbf{V}^*}\}_{x \in L}$ and $M_2 := \{M_{w_x^2, \mathbf{V}^*}\}_{x \in L}$ are quantum computationally indistinguishable (i.e., $M_1 \approx_{qc} M_2$). Here $M_{w_x^i, \mathbf{V}^*}$ denotes the composed machine of P and $\mathbf{V}^*$ on instance x, and P uses witness w_x^i.*

We know that classically WI is preserved under parallel repetition when the prover is efficient [36]. By a similar argument, one can also show that $\mathtt{QC\text{-}WI}$ protocols remain $\mathtt{QC\text{-}WI}$ under parallel reception. This is useful for reducing the soundness error of a $\mathtt{QC\text{-}WI}$ protocol. Here we only state this property and skip the proof.

Lemma 5.5 (Parallel composition of $\mathtt{QC\text{-}WI}$ protocols). *Let $L \in$ NP and suppose that $\langle P, V \rangle$ is a $\mathtt{QC\text{-}WI}$ for R_L and P is polynomial time given a witness. Let $q(\cdot)$ be a polynomial and let $\langle P^q, V^q \rangle$ be machines so that they invoke $\langle P, V \rangle$ q-times in parallel. Specifically, on common input $\{x_i : i = 1, \dots, q\}$ and (private) input $\{w_i : i = 1, \dots, q\}$ to P^q, the ith invocation is $\langle P(w_i), V \rangle(x_i)$. Then $\langle P^q, V^q \rangle$ is $\mathtt{QC\text{-}WI}$ for the relation*

$$R_L^q := \{(\{x_i : i = 1, \dots, q\}, \{w_i : i = 1, \dots, q\}) : \forall i, (x_i, w_i) \in R_L\}.$$

It is easy to see that quantum ZK implies $\mathtt{QC\text{-}WI}$. Meanwhile if we use a statistically binding and *quantum computationally hiding* commitment (as the one following Assumption 1) in Blum's basic (3-round) ZK protocol [10], we can show that the resulting protocol, call it HC_Q, is quantum ZK using the techniques from [72]. Therefore, we claim that HC_Q is $\mathtt{QC\text{-}WI}$. Using a polynomial number of parallel

repetitions of HC_Q, we have a QC-WI protocol for NP with negligible soundness error which we call Π_{WI} and will use in later constructions.

We now construct $\pi_{ZK}^{\mathcal{F}_{CF}}$ that quantum-UC emulates $\mathcal{F}_{ZK}$ in the $\mathcal{F}_{CF}$-hybrid model.

Let L be an NP language and R_L be the corresponding NP-relation. Let **PG** be a quantum secure pseudorandom generator as in Assumption 1, and let $\mathcal{E} = (\mathbf{Gen}, \mathbf{Enc}, \mathbf{Dec})$ be an encryption scheme as in Assumption 2. We define another relation

$$R = \{((x_1, x_2, pk, e), \hat{w}) | (\hat{w} = (w, r) \wedge \mathbf{Enc}_{pk}(w, r) = e \wedge (x_1, w) \in R_L) \text{ or } (\mathbf{PG}(\hat{w}) = x_2)\}.$$

It is clear that R is an NP-relation, and thus there is a WI proof for R. The key idea of constructing $\Pi_{ZK}^{\mathcal{F}_{CF}}$ is to exploit the outcome of the coin-flipping in some clever way. We will interpret the coins s as two parts (s_1, s_2), where $s_1 = pk$ will be used as a public key pk for $\mathcal{E}$, and s_2 will sometimes be an output string of **PG**. Our $\Pi_{ZK}^{\mathcal{F}_{CF}}$ has a simple form then: **P** and **V** get $s = (s_1, s_2)$, **P** sends x and $e = \mathbf{Enc}_{s_1}(w)$ to **V**, and next they run a WI protocol on $(x_1 = x, x_2 = s_2, pk = s_1, e)$ using witness w. Intuitively, if the adversary $\mathcal{A}$ corrupts the verifier **V**, then $\mathcal{S}$ can choose a fake $s' = (s_1', s_2')$ where s_2' is generated by **PG** with random seed r, i.e., $s_2' = \mathbf{PG}(r)$. Then it generates an arbitrary ciphertext as e and uses r as a witness in the WI proof, and witness-indistinguishability ensures the $\mathcal{A}$ can not distinguish from the case where **P** uses a real witness w of x. If the prover is corrupted, $\mathcal{S}$ can simply generate $(pk, sk) \leftarrow \mathbf{Gen}(1^n)$ and assign pk as s_1', while s_2' is still uniformly chosen. Therefore, whenever $\mathcal{A}$ convinces $\mathcal{S}$ in the WI protocol, $\mathcal{S}$ then decrypts (it knows sk) $w = \mathbf{Dec}_{sk}(e)$. However, there is one subtlety. Namely, R has two witnesses, either a real w (which is what we really ask for) s.t. $(x, w) \in R_L$ or a random seed r s.t. $\mathbf{PG}(r) = s_2'$. We do not want $\mathcal{A}$ to be capable of achieving the latter case. This is easy to guarantee though, because we can choose a generator **PG** with sufficient expansion factor, e.g., if $\mathbf{PG} : \{0,1\}^n \rightarrow \{0,1\}^{3n}$. Then given a uniformly random $3n$-bit string s_2', the probability that there is a seed $r \in \{0,1\}^n$ getting mapped to s_2' is negligible. Thus whenever a prover succeeds in WI, it must have proved the statement with respect to R_L rather than with respect to **PG**. The formal description of protocol $\Pi_{ZK}^{\mathcal{F}_{CF}}$ follows.

UC-secure ZKAoK Protocol $\Pi_{ZK}^{\mathcal{F}_{CF}}$

(a) **P** and **V** get $s = (s_1, s_2) \in \{0,1\}^n \times \{0,1\}^{3n}$ from $\mathcal{F}_{CF}$.

(b) **P** sends x and $e = \mathbf{Enc}_{s_1}(w, r)$ to **V**.

(c) **P** and **V** invoke a WI protocol Π_{WI} for relation R with input instance $(x_1 = x, x_2 = s_2, pk = s_1, e)$. **P** uses (w, r) as a witness for (x_1, x_2, pk, e).

(d) **V** accepts if it accepts in Π_{WI}.

Lemma 5.6. *The classical protocol $\Pi_{ZK}^{\mathcal{F}_{CF}}$ C-QUC emulates $\mathcal{F}_{ZK}$.*

Proof. We first deal with the case in which the prover is corrupted.

Claim 5.7. *For any $\mathcal{A}$ corrupting the prover, $M_{\Pi_{ZK}^{\mathcal{F}_{CF}}, \mathcal{A}(P)} \approx_{qci} M_{\mathcal{F}_{ZK}, \mathcal{S}}$.*

Proof. Note that in the ideal world, if $(x, w) \notin R_L$, the dummy verifier will reject. Define an intermediate machine M in which $\mathcal{F}_{ZK}$ always sends x to the dummy verifier (i.e. it accepts), and M is identical to $M_{\mathcal{F}_{ZK}, \mathcal{S}}$ otherwise. M and $M_{\mathcal{F}_{ZK}, \mathcal{S}}$ behave differently only when Π_{WI} succeeds but somehow $(x, w) \notin R_L$. This however violates the soundness property of Π_{WI}. Hence $M_{\mathcal{F}_{ZK}, \mathcal{S}} \approx_{qci} M$. Then M and $M_{\Pi_{ZK}^{\mathcal{F}_{CF}}, \mathcal{A}(P)}$ are

<div style="border:1px solid black; padding:1em;">

Simulator $\mathcal{S}$: prover is corrupted

Input: adversary $\mathcal{A}$; security parameter 1^n.

(a) $\mathcal{S}$ initializes $\mathcal{A}$ with whatever input state it receives from the environment.

(b) $\mathcal{S}$ internally generates $(pk, sk) \leftarrow \mathbf{Gen}(1^n)$ and set $s_1' = pk$. Choose $s_2' \leftarrow \{0,1\}^{3n}$ uniformly at random. Let $s' = (s_1', s_2')$ be the fake coins and it is given to $\mathcal{A}$.

(c) When $\mathcal{S}$ receives (x, e) from $\mathcal{A}$, it decrypts e to get $w = \mathbf{Dec}_{sk}(e)$.

(d) $\mathcal{S}$ runs Π_{WI} with $\mathcal{A}$ on input instance (x, s_2', s_1', e) where $\mathcal{S}$ plays the role of a verifier. If $\mathcal{S}$ accepts in Π_{WI}, it sends (x, w) to $\mathcal{F}_{ZK}$.

(e) $\mathcal{S}$ outputs whatever $\mathcal{A}$ outputs.

</div>

simply related by switching between a valid public key and a truly random string. The lemma then follows from Assumption 2. $\qquad\square$

Now we consider the case where $\mathcal{A}$ corrupts the verifier.

<div style="border:1px solid black; padding:1em;">

Simulator $\mathcal{S}$: verifier is corrupted

Input: given adversary $\mathcal{A}$; security parameter 1^n;

(a) $\mathcal{S}$ initializes $\mathcal{A}$ with whatever input state it receives from the environment.

(b) Wait till it receives x from $\mathcal{F}_{ZK}$. Then $\mathcal{S}$ internally generates $s_1' \leftarrow \{0,1\}^n$. It also generates $r \leftarrow \{0,1\}^n$ and sets $s_2' = \mathbf{PG}(r)$. Let $s' = (s_1', s_2')$ be the fake coins and it is given to $\mathcal{A}$.

(c) $\mathcal{S}$ sends x and $e = \mathbf{Enc}_{s_1'}(0^n)$ to $\mathcal{A}$ and then invokes Π_{WI} with $\mathcal{A}$ on input instance (x, s_2', s_1', e). $\mathcal{S}$ uses r as a witness.

(d) $\mathcal{S}$ outputs whatever $\mathcal{A}$ outputs.

</div>

Claim 5.8. For any $\mathcal{A}$ corrupting the verifier, $M_{\Pi_{ZK}^{\mathcal{F}_{CF}}, \mathcal{A}(V)} \approx_{qci} M_{\mathcal{F}_{ZK}, \mathcal{S}}$.

Proof. We define a sequence of indistinguishable machines as follows.

Now we can see that:

- $M_0 \approx_{qci} M_1$ because they simply related by changing the plaintext of encryption e.

- $M_1 \approx_{qci} M_2$ because Π_{WI} is QC-WI. Otherwise we can construct a malicious $\mathbf{V}^*$ such that $M_{w_x^1, \mathbf{V}^*}$ and $M_{w_x^2, \mathbf{V}^*}$ become distinguishable.

- $M_2 \approx_{qci} M_3$ because they are simply related by switching a pseudorandom string to a uniformly random string.

Thus Claim 5.8 holds.

$\qquad\square$

- $M_0 := M_{\mathcal{F}_{\mathrm{ZK}}, \mathcal{S}}$. The ideal-world machine describing $\mathbf{P}, \mathcal{S}$ and $\mathcal{F}_{\mathrm{ZK}}$ as a single interactive machine.

- M_1: same as M_0 except that the ciphertext is changed from $\mathbf{Enc}_{s_1'}(0^n)$ to $e = \mathbf{Enc}_{s_1'}(w)$. Here w is a witness for x, i.e., $R_L(x, w) = 1$.

- M_2: identical to M_1 except that M_2 uses w as a witness in the Π_{WI}.

- M_3: s_2' is also chosen uniformly random, rather than pseudorandom. Note M_3 is exactly the real-world machine $M_{\Pi_{\mathrm{ZK}}^{\mathcal{F}_{\mathrm{CF}}}, \mathcal{A}(V)}$.

We finally get $\Pi_{\mathrm{ZK}}^{\mathcal{F}_{\mathrm{CF}}}$ quantum-UC emulates $\mathcal{F}_{\mathrm{ZK}}$.

$\square$

Acknowledgments

References

[1] Boaz Barak and Amit Sahai. How to play almost any mental game over the net - concurrent composition via super-polynomial simulation. In *FOCS*, pages 543–552. IEEE Computer Society, 2005. ISBN 0-7695-2468-0.

[2] Donald Beaver. On deniability in quantum key exchange. In Lars R. Knudsen, editor, *EUROCRYPT*, volume 2332 of *Lecture Notes in Computer Science*, pages 352–367. Springer, 2002. ISBN 3-540-43553-0.

[3] Donald Beaver, Silvio Micali, and Phillip Rogaway. The round complexity of secure protocols (extended abstract). In Harriet Ortiz, editor, *STOC*, pages 503–513. ACM, 1990. ISBN 0-89791-361-2.

[4] Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In Serge Vaudenay, editor, *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 409–426. Springer, 2006. ISBN 3-540-34546-9.

[5] Michael Ben-Or and Dominic Mayers. General security definition and composability for quantum and classical protocols, September 2004. arxiv:quant-ph/0409062v2.

[6] Michael Ben-Or, Michal Horodecki, Debbie W. Leung, Dominic Mayers, and Jonathan Oppenheim. The universal composable security of quantum key distribution. In Joe Kilian, editor, *TCC*, volume 3378 of *Lecture Notes in Computer Science*, pages 386–406. Springer, 2005. ISBN 3-540-24573-1.

[7] Michael Ben-Or, Claude Crépeau, Daniel Gottesman, Avinatan Hassidim, and Adam Smith. Secure multiparty quantum computation with (only) a strict honest majority. In *FOCS*, pages 249–260. IEEE Computer Society, 2006. ISBN 0-7695-2720-5.

[8] Charles H Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*, pages 175–179. New York, 1984.

[9] Charles H. Bennett, Gilles Brassard, Claude Crépeau, and Marie-Hélène Skubiszewska. Practical quantum oblivious transfer. In Joan Feigenbaum, editor, *CRYPTO*, volume 576 of *Lecture Notes in Computer Science*, pages 351–366. Springer, 1991. ISBN 3-540-55188-3.

[10] Manuel Blum. How to prove a theorem so no one else can claim it. In *Proceedings of the International Congress of Mathematicians*, volume 1, page 2. American Mathematical Society, 1986.

[11] Gilles Brassard and Claude Crépeau. Quantum bit commitment and coin tossing protocols. In Alfred Menezes and Scott A. Vanstone, editors, *CRYPTO*, volume 537 of *Lecture Notes in Computer Science*, pages 49–61. Springer, 1990. ISBN 3-540-54508-5.

[12] Ran Canetti. Security and composition of multiparty cryptographic protocols. *J. Cryptology*, 13(1): 143–202, 2000.

[13] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *FOCS*, pages 136–145. IEEE Computer Society, 2001. ISBN 0-7695-1390-5.

[14] Ran Canetti and Marc Fischlin. Universally composable commitments. In *Advances in Cryptology—Crypto 2001*, pages 19–40. Springer, 2001.

[15] Ran Canetti, Yehuda Lindell, Rafail Ostrovsky, and Amit Sahai. Universally composable two-party and multi-party secure computation. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 494–503. ACM, 2002.

[16] Ran Canetti, Yevgeniy Dodis, Rafael Pass, and Shabsi Walfish. Universally composable security with global setup. In Salil P. Vadhan, editor, *TCC*, volume 4392 of *Lecture Notes in Computer Science*, pages 61–85. Springer, 2007. ISBN 3-540-70935-5.

[17] Giulio Chiribella, Giacomo M. D'Ariano, and Paolo Perinotti. Memory effects in quantum channel discrimination. *Phys. Rev. Lett.*, 101:180501, Oct 2008.

[18] Kai-Min Chung, Yaoyun Shi, and Xiaodi Wu. Physical randomness extractors: Generating random numbers with minimal assumptions, 2014. arXiv:1402.4797 [quant-ph]. Presented at QIP'14.

[19] Richard Cleve. Limits on the security of coin flips when half the processors are faulty. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pages 364–369. ACM, 1986.

[20] Claude Crépeau, Daniel Gottesman, and Adam Smith. Secure multi-party quantum computation. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 643–652. ACM, 2002.

[21] Claude Crépeau, Paul Dumais, Dominic Mayers, and Louis Salvail. Computational collapse of quantum state with application to oblivious transfer. In Moni Naor, editor, *TCC*, volume 2951 of *Lecture Notes in Computer Science*, pages 374–393. Springer, 2004. ISBN 3-540-21000-8.

[22] Claude Crépeau, Daniel Gottesman, and Adam Smith. Approximate quantum error-correcting codes and secret sharing schemes. In Ronald Cramer, editor, *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 285–301. Springer, 2005. ISBN 3-540-25910-4.

[23] Claude Crépeau, Louis Salvail, Jean-Raymond Simard, and Alain Tapp. Two provers in isolation. In *Advances in Cryptology–ASIACRYPT 2011*, pages 407–430. Springer, 2011.

[24] Ivan Damgård and Carolin Lunemann. Quantum-secure coin-flipping and applications. In Mitsuru Matsui, editor, *ASIACRYPT*, volume 5912 of *Lecture Notes in Computer Science*, pages 52–69. Springer, 2009. ISBN 978-3-642-10365-0.

[25] Ivan Damgård, Serge Fehr, Louis Salvail, and Christian Schaffner. Secure identification and qkd in the bounded-quantum-storage model. In Alfred Menezes, editor, *CRYPTO*, volume 4622 of *Lecture Notes in Computer Science*, pages 342–359. Springer, 2007. ISBN 978-3-540-74142-8.

[26] Ivan Damgård, Serge Fehr, Louis Salvail, and Christian Schaffner. Cryptography in the bounded-quantum-storage model. *SIAM J. Comput.*, 37(6):1865–1890, 2008.

[27] Ivan Damgård, Serge Fehr, Carolin Lunemann, Louis Salvail, and Christian Schaffner. Improving the security of quantum protocols via commit-and-open. In Shai Halevi, editor, *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 408–427. Springer, 2009. ISBN 978-3-642-03355-1.

[28] Paul Dumais, Dominic Mayers, and Louis Salvail. Perfectly concealing quantum bit commitment from any quantum one-way permutation. In Bart Preneel, editor, *EUROCRYPT*, volume 1807 of *Lecture Notes in Computer Science*, pages 300–315. Springer, 2000. ISBN 3-540-67517-5.

[29] Vedran Dunjko, Joseph F Fitzsimons, Christopher Portmann, and Renato Renner. Composable security of delegated quantum computation. In *Advances in Cryptology–ASIACRYPT 2014*, pages 406–425. Springer, 2014.

[30] Frédéric Dupuis, Jesper Buus Nielsen, and Louis Salvail. Secure two-party quantum evaluation of unitaries against specious adversaries. In *Advances in Cryptology–CRYPTO 2010*, pages 685–706. Springer, 2010.

[31] Frédéric Dupuis, Jesper Buus Nielsen, and Louis Salvail. Actively secure two-party evaluation of any quantum operation. In *Advances in Cryptology–CRYPTO 2012*, pages 794–811. Springer, 2012.

[32] Serge Fehr and Christian Schaffner. Composing quantum protocols in a classical environment. In Omer Reingold, editor, *TCC*, volume 5444 of *Lecture Notes in Computer Science*, pages 350–367. Springer, 2009. ISBN 978-3-642-00456-8.

[33] Serge Fehr, Ran Gelles, and Christian Schaffner. Security and composability of randomness expansion from bell inequalities. *Physical Review A*, 87(1):012335, 2013.

[34] Serge Fehr, Jonathan Katz, Fang Song, Hong-Sheng Zhou, and Vassilis Zikas. Feasibility and completeness of cryptographic tasks in the quantum world. In *Theory of Cryptography*, pages 281–296. Springer, 2013.

[35] Uriel Feige and Adi Shamir. Zero knowledge proofs of knowledge in two rounds. In Gilles Brassard, editor, *CRYPTO*, volume 435 of *Lecture Notes in Computer Science*, pages 526–544. Springer, 1989. ISBN 3-540-97317-6.

[36] Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 416–426. ACM, 1990.

[37] Oded Goldreich. *Foundations of Cryptography*, volume 2. Cambridge university press, 2009.

[38] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In Alfred V. Aho, editor, *STOC*, pages 218–229. ACM, 1987. ISBN 0-89791-221-7.

[39] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity for all languages in np have zero-knowledge proof systems. *J. ACM*, 38(3):691–729, 1991.

[40] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989.

[41] S Dov Gordon, Carmit Hazay, Jonathan Katz, and Yehuda Lindell. Complete fairness in secure two-party computation. *Journal of the ACM (JACM)*, 58(6):24, 2011.

[42] Gus Gutoski. On a measure of distance for quantum strategies. *Journal of Mathematical Physics*, 53 (3):032202, 2012.

[43] Gus Gutoski. Personal communication, 2013. See also `http://pirsa.org/12120028/`.

[44] Gus Gutoski and John Watrous. Toward a general theory of quantum games. In *Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing*, STOC '07, pages 565–574, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-631-8.

[45] Sean Hallgren. Polynomial-time quantum algorithms for pell's equation and the principal ideal problem. *J. ACM*, 54(1):4:1–4:19, March 2007. ISSN 0004-5411. doi: 10.1145/1206035.1206039. URL `http://doi.acm.org/10.1145/1206035.1206039`.

[46] Sean Hallgren, Alexandra Kolla, Pranab Sen, and Shengyu Zhang. Making classical honest verifier zero knowledge protocols secure against quantum attacks. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *ICALP (2)*, volume 5126 of *Lecture Notes in Computer Science*, pages 592–603. Springer, 2008. ISBN 978-3-540-70582-6.

[47] Dennis Hofheinz and Dominique Unruh. Simulatable security and polynomially bounded concurrent composability. In *IEEE Symposium on Security and Privacy*, pages 169–183. IEEE Computer Society, 2006. ISBN 0-7695-2574-1.

[48] Jonathan Katz. On achieving the best of both worlds in secure multiparty computation. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 11–20. ACM, 2007.

[49] Eyal Kushilevitz, Yehuda Lindell, and Tal Rabin. Information-theoretically secure protocols and security under composition. *SIAM J. Comput.*, 39(5):2090–2112, 2010.

[50] Yehuda Lindell. Parallel coin-tossing and constant-round secure two-party computation. *J. Cryptology*, 16(3):143–184, 2003.

[51] Yehuda Lindell. General composition and universal composability in secure multiparty computation. *J. Cryptology*, 22(3):395–428, 2009.

[52] H. K. Lo and H. F. Chau. Is quantum bit commitment really possible? *Physical Review Letters*, 78: 3410–3413, 1997.

[53] Hoi-Kwong Lo and H. F. Chau. Unconditional security of quantum key distribution over arbitrarily long distances. *Science*, 283(5410):2050–2056, 1999.

[54] Carolin Lunemann and Jesper Buus Nielsen. Fully simulatable quantum-secure coin-flipping and applications. In Abderrahmane Nitaj and David Pointcheval, editors, *AFRICACRYPT*, volume 6737 of *Lecture Notes in Computer Science*, pages 21–40. Springer, 2011. ISBN 978-3-642-21968-9.

[55] Ueli Maurer and Renato Renner. Abstract cryptography. In *In Innovations in Computer Science*. Tsinghua University Press, 2011.

[56] Dominic Mayers. Unconditionally secure quantum bit commitment is impossible. *Physical Review Letters*, 78:3414–3417, 1997.

[57] Dominic Mayers. Unconditional security in quantum cryptography. *J. ACM*, 48(3):351–406, 2001.

[58] Carl A Miller and Yaoyun Shi. Robust protocols for securely expanding randomness and distributing keys using untrusted quantum devices. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 417–426. ACM, 2014.

[59] Tal Moran, Moni Naor, and Gil Segev. An optimally fair coin toss. In *Theory of Cryptography*, pages 1–18. Springer, 2009.

[60] Moni Naor. Bit commitment using pseudorandomness. *J. Cryptology*, 4(2):151–158, 1991.

[61] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[62] Rafael Pass. Simulation in quasi-polynomial time, and its application to protocol composition. In Eli Biham, editor, *EUROCRYPT*, volume 2656 of *Lecture Notes in Computer Science*, pages 160–176. Springer, 2003. ISBN 3-540-14039-5.

[63] Manoj Prabhakaran and Amit Sahai. New notions of security: achieving universal composability without trusted setup. In László Babai, editor, *STOC*, pages 242–251. ACM, 2004. ISBN 1-58113-852-0.

[64] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6): 34, 2009. Preliminary version in *STOC* 2005.

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

[66] Peter W. Shor and John Preskill. Simple proof of security of the BB84 quantum key distribution protocol. *Phys. Rev. Lett.*, 85(2):441–444, Jul 2000.

[67] Dominique Unruh. Simulatable security for quantum protocols, 2004. arXiv:quant-ph/0409125v2.

[68] Dominique Unruh. Universally composable quantum multi-party computation. In Henri Gilbert, editor, *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 486–505. Springer, 2010. ISBN 978-3-642-13189-9.

[69] Dominique Unruh. Concurrent composition in the bounded quantum storage model. In Kenneth G. Paterson, editor, *EUROCRYPT*, volume 6632 of *Lecture Notes in Computer Science*, pages 467–486. Springer, 2011. ISBN 978-3-642-20464-7.

[70] Dominique Unruh. Quantum proofs of knowledge. In *Eurocrypt 2012*, volume 7237 of *LNCS*, pages 135–152. Springer, April 2012. Preprint on IACR ePrint 2010/212.

[71] Umesh Vazirani and Thomas Vidick. Fully device-independent quantum key distribution. *Physical review letters*, 113(14):140501, 2014.

[72] John Watrous. Zero-knowledge against quantum attacks. *SIAM J. Comput.*, 39(1):25–58, 2009. Preliminary version in *STOC* 2006.

[73] Andrew Chi-Chih Yao. Quantum circuit complexity. In *FOCS*, pages 352–361. IEEE Computer Society, 1993.

A   Proof of Modular Composition Theorem

Proof of Theorem 3.4. Let $\Pi' := \Pi^{\Gamma/\Gamma'}$ be the composed protocol. We show the theorem in the computational setting, and proofs for the statistical and perfect settings are analogous. Specifically, we need to show that

$$\forall \mathcal{A} \, \exists \mathcal{S} : M_{\Pi',\mathcal{S}} \approx_{qc} M_{\Pi,\mathcal{A}} \, .$$

Without loss of generality, we assume that Π only calls Γ once. The proof will proceed in three steps:

(a) From any adversary $\mathcal{A}$ attacking Π', we construct another adversary $\mathcal{A}_{\Gamma'}$ attacking Γ'. Notice that Γ' is a subroutine in Π'. Basically $\mathcal{A}_{\Gamma'}$ consists of the segment of the circuits of $\mathcal{A}$ during the subroutine call of Γ'.

(b) By the assumption that Γ' C-QSA emulates Γ, we know that $\forall \mathcal{A}_{\Gamma'} \exists \mathcal{A}_{\Gamma} : M_{\Gamma',\mathcal{A}_{\Gamma'}} \approx_{qc} M_{\Gamma,\mathcal{A}_{\Gamma}}$. This gives us an adversary $\mathcal{A}_{\Gamma}$.

(c) Finally the adversary $\mathcal{S}$ will be constructed by "composing" the machines $\mathcal{A}$ and $\mathcal{A}_{\Gamma}$: when Π makes the subroutine call to Γ, $\mathcal{S}$ runs $\mathcal{A}_{\Gamma}$, otherwise it follows the operations of $\mathcal{A}$. Then $M_{\Pi',\mathcal{A}} \approx_{qc} M_{\Pi,\mathcal{S}}$ basically follows from $M_{\Gamma',\mathcal{A}_{\Gamma'}} \approx_{qc} M_{\Gamma,\mathcal{A}_{\Gamma}}$.

Next we give the details.

Step 1 (Constructing $\mathcal{A}_{\Gamma'}$ from $\mathcal{A}$). Adversary $\mathcal{A}_{\Gamma'}$ represents the segment of $\mathcal{A}$ during the subroutine Γ'. It starts with some state that supposedly represents the joint state in an execution of Π' with $\mathcal{A}$ right before the invocation of Γ'. It then runs $\mathcal{A}$ till completion of Γ'.

Adversary $\mathcal{A}_{\Gamma'}$

Input: adversary $\mathcal{A}$; security parameter 1^n;

(a) $\mathcal{A}_{\Gamma'}$ initiates $\mathcal{A}$ with whatever input it receives from the environment. It then runs $\mathcal{A}$ in the execution of Γ'.

(b) When Γ' terminates, $\mathcal{A}_{\Gamma'}$ outputs the state on all of $\mathcal{A}$'s registers.

Step 2 (Simulating $\mathcal{A}_{\Gamma'}$ by $\mathcal{A}_\Gamma$). This step is straightforward from the hypothesis that Γ' C-QSA emulates Γ, which means that $\forall \mathcal{A}_{\Gamma'} \exists \mathcal{A}_\Gamma : M_{\Gamma',\mathcal{A}_{\Gamma'}} \approx_{wqc} M_{\Gamma,\mathcal{A}_\Gamma}$.

Step 3 (Constructing $\mathcal{S}$ from $\mathcal{A}_\Gamma$ and $\mathcal{A}$). The construction is as described above. Here we show that $M_{\Pi',\mathcal{A}} \approx_{qc} M_{\Pi,\mathcal{S}}$. Suppose for contradiction that there exists a distinguisher $\mathcal{Z}$ and state[7] σ_n such that:

$$\left| \Pr[\mathcal{Z}((\mathbb{1}_{L(R)} \otimes M_{\Pi,\mathcal{A}})(\sigma_n)) = 1] - \Pr[\mathcal{Z}((\mathbb{1}_{L(R)} \otimes M_{\Pi',\mathcal{S}})(\sigma_n)) = 1] \right| \geq \varepsilon(n) \,,$$

with $\varepsilon(n) \geq 1/poly(n)$. We show a distinguisher $\tilde{\mathcal{Z}}$ and state $\tilde{\sigma}_n$ such that on input $\tilde{\sigma}_n$, $M_{\Gamma',\mathcal{A}_{\Gamma'}}$ and $M_{\Gamma,\mathcal{A}_\Gamma}$ becomes distinguishable under $\tilde{\mathcal{Z}}$.

- Let $\tilde{\sigma}_n$ be the joint state of executing Π' in the presence of $\mathcal{A}$ on input σ_n right before the invocation of Γ'. Clearly it is identical to the joint state of executing Π in the presence of $\mathcal{S}$ on input σ_n right before the invocation of Γ.

- Distinguisher $\tilde{\mathcal{Z}}$ runs the circuits of $\mathcal{A}$ after execution of the subroutine Γ' (equivalently the circuits of $\mathcal{S}$ after execution of the subroutine Γ) and then runs $\mathcal{Z}$.

It is easy to see that

$$\tilde{\mathcal{Z}}((\mathbb{1}_{L(R)} \otimes M_{\Gamma',\mathcal{A}_{\Gamma'}})(\tilde{\sigma}_n)) \equiv \mathcal{Z}((\mathbb{1}_{L(R)} \otimes M_{\Pi',\mathcal{A}})(\sigma_n))$$
$$\text{and } \tilde{\mathcal{Z}}((\mathbb{1}_{L(R)} \otimes M_{\Gamma,\mathcal{A}_\Gamma})(\tilde{\sigma}_n)) \equiv \mathcal{Z}((\mathbb{1}_{L(R)} \otimes M_{\Pi,\mathcal{S}})(\sigma_n)) \,,$$

where "$\equiv$" means identical distributions. This implies that

$$\left| \Pr[\tilde{\mathcal{Z}}((\mathbb{1}_{L(R)} \otimes M_{\Gamma',\mathcal{A}_{\Gamma'}})(\tilde{\sigma}_n)) = 1] - \Pr[\tilde{\mathcal{Z}}((\mathbb{1}_{L(R)} \otimes M_{\Gamma,\mathcal{A}_\Gamma})(\tilde{\sigma}_n)) = 1] \right| \geq \varepsilon(n) \,.$$

This contradicts the assumption that $M_{\Gamma',\mathcal{A}_{\Gamma'}} \approx_{qc} M_{\Gamma,\mathcal{A}_\Gamma}$.

This concludes our proof for the modular composition theorem. □

B A Special Constraint in Quantum Stand-Alone Model: Markov Condition

Another choice exists in the literature [27, 32], where a stand-alone model was proposed to capture secure emulation of *classical* functionalities. Only a special form of inputs is allowed there, which satisfy what we call the *Markov condition*. As opposed to a general bipartite state with one part being classical (a.k.a *cq-states*): $\rho_{AB} = \sum_a \lambda_a |a\rangle\langle a| \otimes \rho_B^a$, the Markov condition requires that the input to dishonest Bob contains a classical subsystem Y such that conditioned on Y Bob's quantum input is independent of Alice's classical input. Such states are denoted as

$$\rho_{A \leftrightarrow Y \leftrightarrow B} = \sum_{a,b} \lambda_{a,b} |a\rangle\langle a|_A \otimes |b\rangle\langle b|_Y \otimes \rho_B^b \,.$$

Now let us analyze how Markov condition affects our abstract model discussed above. It turns out that the effect of Markov condition, again, depends on whether $\mathcal{Z}_1$ takes quantum advice.

$\mathcal{Z}_1$ **takes quantum advice: Markov condition becomes redundant.** We denote models with Markov condition $\mathcal{M}^*$.

---

[7]More precisely there exists a family of states $\{\sigma_n\}_{n \in \mathbb{N}}$.

Lemma B.1. $\mathcal{M}^*_{aux_1,\cdot,\cdot} \equiv \mathcal{M}_{aux_1,\cdot,\cdot}$ *regardless of the choices for aux_2 and state passing. Namely, the model where inputs must satisfy Markov condition is equivalent to the model where inputs could be any bipartite cq-states.*

Proof. To be concrete, we consider two models $\mathcal{M} := \mathcal{M}_{aux_1,\overline{aux_2},\overline{state}}$ and $\mathcal{M}' := \mathcal{M}^*_{aux_1,\overline{aux_2},\overline{state}}$. The same argument applies to other cases.

One direction is obvious, namely, if a protocol Π emulates Γ in $\mathcal{M}$ then it automatically holds that Π emulates Γ in $\mathcal{M}'$. This is because we can think of the Markov condition as specifying a subclass of possible $\mathcal{Z}_1$ allowed in $\mathcal{M}$. Now we show the converse by contradiction. Specifically, we prove that if there is an adversary $\mathcal{A}$ in $\mathcal{M}$, and $\forall \mathcal{S}$, there exist $(\mathcal{Z}_1, \mathcal{Z}_2)$ such that $\mathcal{Z}_2$ can distinguish $M_{\Pi,\mathcal{A}}$ and $M_{\Gamma,\mathcal{S}}$, then in model $\mathcal{M}'$ we construct $\mathcal{A}'$, $\mathcal{Z}_1'$ and $\mathcal{Z}_2'$ such that no $\mathcal{S}'$ that is able to simulate $\mathcal{A}'$. By our hypothesis, there is an input state σ_n, which can always be written as $\sum_a \lambda_a |a\rangle\langle a|_A \otimes \sigma_B^a$ with $\sum_i \lambda_i = 1$ such that

$$|\Pr[\mathcal{Z}_2(M_{\Pi,\mathcal{A}}(\sigma_n)) = 1] - \Pr[\mathcal{Z}_2(M_{\Gamma,\mathcal{S}}(\sigma_n)) = 1]| \geq 1/poly(n)$$

holds for any poly-time $\mathcal{S}$. Observe that each summand $|a\rangle\langle a|_A \otimes \sigma_B^a$ of σ_n trivially satisfies Markov condition. Since σ_n is a convex combination of $|a\rangle\langle a|_A \otimes \sigma_B^a$, there must be a $\tilde{\sigma}_n = |\tilde{a}\rangle\langle\tilde{a}|_A \otimes \sigma_B^{\tilde{a}}$ such that

$$|\Pr[\mathcal{Z}_2(M_{\Pi,\mathcal{A}}(\tilde{\sigma}_n)) = 1] - \Pr[\mathcal{Z}_2(M_{\Gamma,\mathcal{S}}(\tilde{\sigma}_n)) = 1]| \geq 1/poly(n)\,,$$

for any poly-time $\mathcal{S}$. This observation tells us that we can simply let $\mathcal{A}' := \mathcal{A}$, $\mathcal{Z}_2' := \mathcal{Z}_2$, and let $\mathcal{Z}_1'$ be the machine that takes quantum advice $\{\tilde{\sigma}_n\}$ and hands $\tilde{\sigma}_n$ to players as input. Then for any poly-time $\mathcal{S}'$,

$$\left|\Pr[\mathcal{Z}_2'(M_{\Pi,\mathcal{A}'}(\tilde{\sigma}_n)) = 1] - \Pr[\mathcal{Z}_2'(M_{\Gamma,\mathcal{S}'}(\tilde{\sigma}_n)) = 1]\right|$$
$$= |\Pr[\mathcal{Z}_2(M_{\Pi,\mathcal{A}}(\tilde{\sigma}_n)) = 1] - \Pr[\mathcal{Z}_2(M_{\Gamma,\mathcal{S}'}(\tilde{\sigma}_n)) = 1]| \geq 1/poly(n)\,.$$

This shows that emulation in $\mathcal{M}'$ implies emulation in $\mathcal{M}$. $\qquad\square$

$\mathcal{Z}_1$ Takes No Quantum Advice: Markov Condition may Matter. The argument in Lemma B.1 does not necessarily apply here because previously we could simply give σ_n^* to $\mathcal{Z}_1$ directly as an advice. However, σ_n^* might be impossible to generate on a poly-time QTM. It is interesting to either construct a concrete example to show a separation or otherwise showing a proof of equivalence. We do not have clear insight into the Markov condition in this case, and leave the possibility of a separation as an open question.