Quantum-Secure Coin-Flipping and Applications

Ivan Damgård and Carolin Lunemann

DAIMI, Aarhus University, Denmark
{ivan|carolin}@cs.au.dk

Abstract. In this paper, we prove classical coin-flipping secure in the presence of quantum adversaries. The proof uses a recent result of Watrous [20] that allows quantum rewinding for protocols of a certain form. We then discuss two applications. First, the combination of coin-flipping with any non-interactive zero-knowledge protocol leads to an easy transformation from non-interactive zero-knowledge to interactive quantum zero-knowledge. Second, we discuss how our protocol can be applied to a recently proposed method for improving the security of quantum protocols [4], resulting in an implementation without set-up assumptions. Finally, we sketch how to achieve efficient simulation for an extended construction in the common-reference-string model.

Keywords. quantum cryptography, coin-flipping, common reference string, quantum zero-knowledge.

1 Introduction

In this paper, we are interested in a standard coin-flipping protocol with classical messages exchange but where the adversary is assumed to be capable of quantum computing. Secure coin-flipping allows two parties Alice and Bob to agree on a uniformly random bit in a fair way, i.e., neither party can influence the value of the coin to his advantage. The (well-known) protocol proceeds as follows: Alice commits to a bit a, Bob then sends bit b, Alice opens the commitment and the resulting coin is the exclusive disjunction of both bits, i.e. $coin = a \oplus b$.

For Alice's commitment to her first message, we assume a classical bit commitment scheme. Intuitively, a commitment scheme allows a player to commit to a value, while keeping it hidden (*hiding property*) but preserving the possibility to later reveal the value fixed at commitment time (*binding property*). More formally, a bit commitment scheme takes a bit and some randomness as input. The hiding property is formalized by the non-existence of a distinguisher able to distinguish with non-negligible advantage between a commitment to 0 and a commitment to 1. The binding property is fulfilled, if it is infeasible for a forger to open one commitment to both values 0 and 1. The hiding respectively binding property holds with unconditional (i.e. perfect or statistical) security in the classical and the quantum setting, if the distinguisher respectively the forger is unrestricted with respect to his (quantum-) computational power. In case of a polynomial-time bounded classical distinguisher respectively forger, the commitment is computationally hiding respectively binding. The computationally hiding property translates to the quantum world by simply allowing the distinguisher to be quantum. However, the case of a quantum forger can not be handled in such a straightforward manner, due to the difficulties of rewinding in general quantum systems (see e.g. [12, 5, 20] for discussions).

For our basic coin-flip protocol, we assume the commitment to be *unconditionally binding* and *computationally hiding against a quantum adversary*.[1] Thus, we achieve unconditional security against cheating Alice and quantum-computational security against dishonest Bob. Such a commitment scheme follows, for instance, from any pseudorandom generator [15], secure against a quantum distinguisher. Even though the underlying computational assumption, on which the security of the embedded commitment is based, withstands quantum attacks, the security proof of the entire protocol and its integration into other applications could previously not be naturally translated from the classical to the quantum world. Typically, security against a classical

[1] Recall that unconditionally secure commitments, i.e. unconditionally hiding and binding at the same time, are impossible in both the classical and the quantum world.

adversary is argued using rewinding of the adversary. But in general, rewinding as a proof technique cannot be directly applied, if Bob runs a quantum computer: First, the intermediate state of a quantum system cannot be copied [21], and second, quantum measurements are in general irreversible. Hence, in order to produce a classical output, the simulator had to (partially) measure the quantum system without copying it beforehand, but then it would become generally impossible to reconstruct all information necessary for correct rewinding. For these reasons, no simple and straightforward security proofs for the quantum case were previously known.

In this paper, we show the most natural and direct quantum analogue of the classical security proof for standard coin-flipping, by using a recent result of Watrous [20]. Watrous showed how to construct an efficient quantum simulator for quantum verifiers for several zero-knowledge proof systems such as graph isomorphism, where the simulation relies on the newly introduced *quantum rewinding theorem*. We now show that his quantum rewinding argument can also be applied to classical coin-flipping in a quantum world.

By calling the coin-flip functionality sequentially a sufficient number of times, the communicating parties can interactively generate a common random string from scratch. The generation can then be integrated into other (classical or quantum) cryptographic protocols that work in the common-reference-string model. This way, several interesting applications can be implemented entirely in a simple manner without any set-up assumptions. Two example applications are discussed in the second part of the paper.

The first application relates to zero-knowledge proof systems, an important building block for larger cryptographic protocols. Recently, Hallgren et al. [13] showed that any honest verifier zero-knowledge protocol can be made zero-knowledge against any classical and quantum verifier. Here we show a related result, namely, a simple transformation from non-interactive (quantum) zero-knowledge to interactive quantum zero-knowledge. A non-interactive zero-knowledge proof system can be trivially turned into an interactive *honest verifier* zero-knowledge proof system by just letting the verifier choose the reference string. Therefore, this consequence of our result also follows from [13]. However, our proof is much simpler. In general, the difference between us and [13] is that our focus is on establishing coin-flipping as a stand-alone tool that can be used in several contexts rather than being integrated in a zero-knowledge construction as in [13].

As second application we discuss the interactive generation of a common reference string for the general compiler construction improving the security of a large class of quantum protocols that was recently proposed in [4]. Applying the compiler, it has been shown how to achieve hybrid security in existing protocols for password-based identification [6] and oblivious transfer [1] without significant efficiency loss, such that an adversary must have both large quantum memory *and* large computing power to break the protocol. Here we show how a common reference string for the compiler can be generated from scratch according to the specific protocol requirements in [4].

Finally, we sketch an extended commitment scheme for quantum-secure coin-flipping in the common-reference-string model. This construction can be *efficiently* simulated without the need of rewinding, which is necessary to claim universal composability.

2 Preliminaries

2.1 Notation

We assume the reader's familiarity with basic notation and concepts of quantum information processing as in standard literature, e.g. [16]. Furthermore, we will only give the details of the discussed applications that are most important in the context of this work. A full description of the applications can be found in the referenced papers.

We denote by $negl(n)$ any function of n, if for any polynomial p it holds that $negl(n) \leq 1/p(n)$ for large enough n. As a measure of *closeness* of two quantum states ρ and σ, their trace distance $\delta(\rho, \sigma) = \frac{1}{2}\operatorname{tr}(|\rho - \sigma|)$ or square-fidelity $\langle\rho|\sigma|\rho\rangle$ can be applied. A quantum algorithm consists of

a family $\{C_n\}_{n\in\mathbb{N}}$ of quantum circuits and is said to run in polynomial time, if the number of gates of C_n is polynomial in n. Two families of quantum states $\{\rho_n\}_{n\in\mathbb{N}}$ and $\{\sigma_n\}_{n\in\mathbb{N}}$ are called *quantum-computationally indistinguishable*, denoted $\rho \overset{q}{\approx} \sigma$, if any polynomial-time quantum algorithm has negligible advantage in n of distinguishing ρ_n from σ_n. Analogously, they are *statistically indistinguishable*, denoted $\rho \overset{s}{\approx} \sigma$, if their trace distance is negligible in n. For the reverse circuit of quantum circuit Q, we use the standard notation for the transposed, complex conjugate operation, i.e. $Q^\dagger$. The controlled-NOT operation (CNOT) with a control and a target qubit as input flips the target qubit, if the control qubit is 1. In other words, the value of the second qubit corresponds to the classical exclusive disjunction (XOR). A phase-flip operation can be described by Pauli operator Z. For quantum state ρ stored in register R we write $|\rho\rangle_R$.

2.2 Definition of Security

We follow the framework for defining security which was introduced in [8] and also used in [4]. Our cryptographic two-party protocols run between player Alice, denoted by A, and player Bob (B). Dishonest parties are indicated by A^* and B^*, respectively. The security against a dishonest player is based on the *real/ideal-world paradigm* that assumes two different worlds: The *real-world* that models the actual protocol Π and the *ideal-world* based on the ideal functionality $\mathcal{F}$ that describes the intended behavior of the protocol. If both executions are indistinguishable, security of the protocol in real life follows. In other words, a dishonest real-world player P^* that attacks the protocol cannot achieve (significantly) more than an ideal-world adversary $\hat{\mathsf{P}}^*$ attacking the corresponding ideal functionality.

More formally, the joint input state consists of classical inputs of honest parties and possibly quantum input of dishonest players. A protocol Π consists of an infinite family of interactive (quantum) circuits for parties A and B. A classical (non-reactive) ideal functionality $\mathcal{F}$ is given by a conditional probability distribution $P_{\mathcal{F}(in_A,in_B)|in_A in_B}$, inducing a pair of random variables $(out_A, out_B) = \mathcal{F}(in_A, in_B)$ for every joint distribution of in_A and in_B, where in_P and out_P denote party P's in- and output, respectively. For the definition of (quantum-) computational security against a dishonest Bob, a polynomial-size (quantum) *input sampler* is considered, which produces the input state of the parties.

Definition 2.1 (Correctness). *A protocol Π* correctly implements *an ideal classical functionality $\mathcal{F}$, if for every distribution of the input values of honest Alice and Bob, the resulting common outputs of Π and $\mathcal{F}$ are* statistically indistinguishable.

Definition 2.2 (Unconditional security against dishonest Alice). *A protocol Π implements an ideal classical functionality $\mathcal{F}$* unconditionally securely *against dishonest Alice, if for any real-world adversary A^*, there exists an ideal-world adversary $\hat{\mathsf{A}}^*$, such that for any input state it holds that the output state, generated by A^* through interaction with honest B in the real-world, is* statistically indistinguishable *from the output state, generated by $\hat{\mathsf{A}}^*$ through interaction with $\mathcal{F}$ and A^* in the ideal-world.*

Definition 2.3 ((Quantum-) Computational security against dishonest Bob). *A protocol Π implements an ideal classical functionality $\mathcal{F}$* (quantum-) computationally securely *against dishonest Bob, if for any (quantum-) computationally bounded real-world adversary B^*, there exists a (quantum-) computationally bounded ideal-world adversary $\hat{\mathsf{B}}^*$, such that for any efficient input sampler, it holds that the output state, generated by B^* through interaction with honest A in the real-world, is* (quantum-) computationally indistinguishable *from the output state, generated by $\hat{\mathsf{B}}^*$ through interaction with $\mathcal{F}$ and B^* in the ideal-world.*

For more details and a definition of indistinguishability of quantum states, see [8]. There, it has also been shown that protocols satisfying the above definitions compose sequentially in a classical environment. Furthermore, note that in Definition 2.2, we do not necessarily require the

ideal-world adversary $\hat{\mathsf{A}}^*$ to be efficient. We show in Section 5 how to extend our coin-flipping construction such that we can achieve an efficient simulator.

The coin-flipping scheme in Section 5 as well as the example applications in Sections 4.1 and 4.2 work in the common-reference-string (CRS) model. In this model, all participants in the real-world protocol have access to a classical public CRS, which is chosen before any interaction starts, according to a distribution only depending on the security parameter. However, the participants in the ideal-world interacting with the ideal functionality do not make use of the CRS. Hence, an ideal-world simulator $\hat{\mathsf{P}}^*$ that operates by simulating a real-world adversary P^* is free to choose a string in any way he wishes.

3 Quantum-Secure Coin-Flipping

3.1 The Coin-Flip Protocol

Let n indicate the security parameter of the commitment scheme which underlies the protocol. We use an *unconditionally binding* and *quantum-computationally hiding* commitment scheme that takes a bit and some randomness r of length l as input, i.e. $com : \{0,1\} \times \{0,1\}^l \rightarrow \{0,1\}^{l+1}$. The unconditionally binding property is fulfilled, if it is impossible for any forger to open one commitment to both 0 and 1, i.e. to compute r, r' such that $com(0,r) = com(1,r')$. Quantum-computationally hiding is ensured, if no quantum distinguisher can distinguish between $com(0,r)$ and $com(1,r')$ for random r, r' with non-negligible advantage. As mentioned earlier, for a specific instantiation we can use, for instance, Naor's commitment based on a pseudorandom generator [15]. This scheme does not require any initially shared secret information and is secure against a quantum distinguisher. [2]

We let Alice and Bob run the $\texttt{Coin}-\texttt{Flip Protocol}$ (see Fig. 1), which interactively generates a random and fair *coin* in one execution and does not require any set-up assumptions. Correctness is obvious by inspection of the protocol: If both players are honest, they independently choose random bits. These bits are then combined via exclusive disjunction, resulting in a uniformly random *coin*.

$\texttt{Coin}-\texttt{Flip Protocol}$

1. A chooses $a \in_R \{0,1\}$ and computes $com(a,r)$. She sends $com(a,r)$ to B.
2. B chooses $b \in_R \{0,1\}$ and sends b to A.
3. A sends $open(a,r)$ and B checks if the opening is valid.
4. Both compute $coin = a \oplus b$.

Fig. 1. The Coin-Flip Protocol.

The corresponding ideal coin-flip functionality $\mathcal{F}_{\mathsf{COIN}}$ is described in Figure 2. Note that dishonest A^* may refuse to open $com(a,r)$ in the real-world after learning B's input. For this case, $\mathcal{F}_{\mathsf{COIN}}$ allows her a second input REFUSE, leading to output FAIL and modeling the abort of the protocol.

3.2 Security

Theorem 3.1. *The* $\texttt{Coin}-\texttt{Flip Protocol}$ *is* unconditionally secure *against any unbounded dishonest Alice according to Definition 2.2, provided that the underlying commitment scheme is unconditionally binding.*

[2] We describe the commitment scheme in this simple notation. However, if it is based on a specific scheme, e.g. [15], the precise notation has to be slightly adapted.

`Ideal Functionality` $\mathcal{F}_{\mathsf{COIN}}$:

Upon receiving requests START from Alice and Bob, $\mathcal{F}_{\mathsf{COIN}}$ outputs a uniformly random *coin* to Alice. It then waits to receive Alice's second input OK or REFUSE and outputs *coin* or FAIL to Bob, respectively.

Fig. 2. The Ideal Coin-Flip Functionality.

Proof. We construct an ideal-world adversary $\hat{\mathsf{A}}^*$, such that the real output of the protocol is statistically indistinguishable from the ideal output produced by $\hat{\mathsf{A}}^*$, $\mathcal{F}_{\mathsf{COIN}}$ and A^*.

`Ideal − World Simulation` $\hat{\mathsf{A}}^*$:

1. Upon receiving $com(a, r)$ from A^*, $\hat{\mathsf{A}}^*$ sends START and then OK to $\mathcal{F}_{\mathsf{COIN}}$ as first and second input, respectively, and receives a uniformly random *coin*.
2. $\hat{\mathsf{A}}^*$ computes a and r from $com(a, r)$.
3. $\hat{\mathsf{A}}^*$ computes $b = coin \oplus a$ and sends b to A^*.
4. $\hat{\mathsf{A}}^*$ waits to receive A^*'s last message and outputs whatever A^* outputs.

Fig. 3. The Ideal-World Simulation $\hat{\mathsf{A}}^*$.

First note that a, r and $com(a, r)$ are chosen and computed as in the real protocol. From the statistically binding property of the commitment scheme, it follows that A^*'s choice bit a is uniquely determined from $com(a, r)$, since for any com, there exists at most one pair (a, r) such that $com = com(a, r)$ (except with probability negligible in n). Hence in the real-world, A^* is unconditionally bound to her bit before she learns B's choice bit, which means a is independent of b. Therefore in Step 2, the simulator can correctly (but not necessarily efficiently) compute a (and r). Note that, in the case of unconditional security, we do not have to require the simulation to be efficient. We show in Section 5 how to extend the commitment in order to extract A^*'s inputs efficiently. Finally, due to the properties of XOR, A^* cannot tell the difference between the random b computed (from the ideal, random *coin*) in the simulation in Step 3 and the randomly chosen b of the real-world. It follows that the simulated output is statistically indistinguishable from the output in the real protocol. □

To prove security against any dishonest quantum-computationally bounded B^*, we show that there exists an ideal-world simulation $\hat{\mathsf{B}}^*$ with output quantum-computationally indistinguishable from the output of the protocol in the real-world. In a classical simulation, where we can simply use rewinding, a polynomial-time simulator works as follows. It inquires *coin* from $\mathcal{F}_{\mathsf{COIN}}$, chooses random a and r, and computes $b' = coin \oplus a$ as well as $com(a, r)$. It then sends $com(a, r)$ to B^* and receives B^*'s choice bit b. If $b = b'$, the simulation was successful. Otherwise, the simulator rewinds B^* and repeats the simulation. Note that our security proof should hold also against any quantum adversary. The polynomial-time quantum simulator proceeds similarly to its classical analogue but requires quantum registers as work space and relies on the *quantum rewinding lemma* of Watrous [20] (see Lemma A.1 in Appendix A).

In the paper, Watrous proves how to construct a quantum zero-knowledge proof system for graph isomorphism using his (ideal) quantum rewinding lemma. The protocol proceeds as a Σ-protocol, i.e. a protocol in three-move form, where the verifier flips a single coin in the second step and sends this challenge to the prover. Since these are the essential aspects also in our `Coin − Flip Protocol`, we can apply Watrous' quantum rewinding technique (with slight modifications) as a black-box to our protocol. We also follow his notation and line of argument here. For a more detailed description and proofs, we refer to [20].

Theorem 3.2. *The* `Coin − Flip Protocol` *is* quantum-computationally secure *against any polynomial-time bounded, dishonest Bob according to Definition 2.3, provided that the underlying commitment scheme is quantum-computationally hiding and the success probability of quantum rewinding achieves a non-negligible lower bound* p_0.

Proof. Let W denote B^*'s auxiliary input register, containing an $\tilde{n}$-qubit state $|\psi\rangle$. Furthermore, let V and B denote B^*'s work space, where V is an arbitrary polynomial-size register and B is a single qubit register. A's classical messages are considered in the following as being stored in quantum registers A_1 and A_2. In addition, the quantum simulator uses registers R, containing all possible choices of a classical simulator, and G, representing its guess b' on B^*'s message b in the second step. Finally, let X denote a working register of size $\tilde{k}$, which is initialized to the state $|0^{\tilde{k}}\rangle$ and corresponds to the collection of all registers as described above except W.

The quantum rewinding procedure is implemented by a general quantum circuit R_{coin} with input $(W, X, \mathsf{B}^*, coin)$. As a first step, it applies a unitary $(\tilde{n}, \tilde{k})$-quantum circuit Q to (W, X) to simulate the conversation, obtaining registers (G, Y). Then, a test takes place to observe whether the simulation was successful. In that case, R_{coin} outputs the resulting quantum register. Otherwise, it *quantumly rewinds* by applying the reverse circuit $Q^\dagger$ on (G, Y) to retrieve (W, X) and then a phase-flip transformation on X before another iteration of Q is applied. Note that R_{coin} is essentially the same circuit as R described in [20], but in our application it depends on the value of a given *coin*, i.e., we apply R_0 or R_1 for $coin = 0$ or $coin = 1$, respectively. In more detail, Q transforms (W, X) to (G, Y) by the following unitary operations:

(1) It first constructs the superposition

$$\frac{1}{\sqrt{2^{l+1}}} \sum_{a,r} |a, r\rangle_R |com(a, r)\rangle_{A_1} \left|b' = coin \oplus a\right\rangle_G |open(a, r)\rangle_{A_2} |0\rangle_B \left|0^{\tilde{k}'}\right\rangle_V |\psi\rangle_W ,$$

where $\tilde{k}' < \tilde{k}$. Note that the state of registers (A_1, G, A_2) corresponds to a uniform distribution of possible transcripts of the interaction between the players.

(2) For each possible $com(a, r)$, it then simulates B^*'s possible actions by applying a unitary operator to (W, V, B, A_1) with A_1 as control:

$$\frac{1}{\sqrt{2^{l+1}}} \sum_{a,r} |a, r\rangle_R |com(a, r)\rangle_{A_1} \left|b'\right\rangle_G |open(a, r)\rangle_{A_2} |b\rangle_B \left|\tilde{\phi}\right\rangle_V \left|\tilde{\psi}\right\rangle_W ,$$

where $\tilde{\phi}$ and $\tilde{\psi}$ describe modified quantum states.

(3) Finally, a CNOT-operation is applied to pair (B, G) with B as control to check whether the simulator's guess of B^*'s choice was correct. The result of the CNOT-operation is stored in register G.

$$\frac{1}{\sqrt{2^{l+1}}} \sum_{a,r} |a, r\rangle_R |com(a, r)\rangle_{A_1} \left|b' \oplus b\right\rangle_G |open(a, r)\rangle_{A_2} |b\rangle_B \left|\tilde{\phi}\right\rangle_V \left|\tilde{\psi}\right\rangle_W .$$

If we denote with Y the register that contains the residual $\tilde{n} + \tilde{k} - 1$ -qubit state, the transformation from (W, X) to (G, Y) by applying Q can be written as

$$Q\left(|\psi\rangle_W \left|0^{\tilde{k}}\right\rangle_X\right) = \sqrt{p}|0\rangle_G |\phi_{good}(\psi)\rangle_Y + \sqrt{1-p}|1\rangle_G |\phi_{bad}(\psi)\rangle_Y ,$$

where $0 < p < 1$ and $|\phi_{good}(\psi)\rangle$ denotes the state, we want the system to be in for a successful simulation. R_{coin} then measures the qubit in register G with respect to the standard basis, which indicates success or failure of the simulation. A successful execution (where $b = b'$) results in outcome 0 with probability p. In that case, R_{coin} outputs Y. A measurement outcome 1 indicates

$b \neq b'$, in which case R_{coin} quantumly rewinds the system, applies a phase-flip (on register X) and repeats the simulation, i.e.

$$Q\Big(2\Big(\mathbb{I}\otimes\Big|0^{\tilde{k}}\Big\rangle\Big\langle 0^{\tilde{k}}\Big|\Big)-\mathbb{I}\Big)Q^{\dagger}\,.$$

Watrous' ideal quantum rewinding lemma (without perturbations) then states the following: Under the condition that the probability p of a successful simulation is non-negligible and independent of any auxiliary input, the output $\rho(\psi)$ of R has square-fidelity close to 1 with state $|\phi_{good}(\psi)\rangle$ of a successful simulation, i.e.,

$$\langle\phi_{good}(\psi)|\rho(\psi)|\phi_{good}(\psi)\rangle \geq 1-\varepsilon$$

with error bound $0<\varepsilon<\frac{1}{2}$. Note that for the special case where p equals $1/2$ and is independent of $|\psi\rangle$, the simulation terminates after at most one rewinding.

However, we cannot apply the exact version of Watrous' rewinding lemma in our simulation, since the commitment scheme in the protocol is only (quantum-) computationally hiding. Instead, we must allow for small perturbations in the quantum rewinding procedure as follows. Let adv denote B^*'s advantage over a random guess on the committed value due to his computing power, i.e. $adv = |p-1/2|$. From the hiding property, it follows that adv is negligible in the security parameter n. Thus, we can argue that the success probability p is *close* to independent of the auxiliary input and Watrous' quantum rewinding lemma with small perturbations, as stated in the appendix (Lemma A.1), applies with $q=\frac{1}{2}$ and $\varepsilon = adv$. All operations in Q can be performed by polynomial-size circuits, and thus, the simulator has polynomial size (in the worst case). Furthermore, for negligible ε but non-negligible lower bound p_0 on the success probability p, it follows that the "closeness" of output $\rho(\psi)$ with good state $|\phi_{good}(\psi)\rangle$ is slightly reduced but quantum rewinding remains possible.

Finally, to proof security against quantum B^*, we construct an ideal-world quantum simulator $\hat{\mathsf{B}}^*$ (see Fig. 4), interacting with B^* and the ideal functionality $\mathcal{F}_{\mathsf{COIN}}$ and executing Watrous' quantum rewinding algorithm. We then compare the output states of the real process and the ideal process. In case of indistinguishable outputs, quantum-computational security against B^* follows.

`Ideal − World Simulation` $\hat{\mathsf{B}}^*$:

1. $\hat{\mathsf{B}}^*$ gets B^*'s auxiliary quantum input W and working registers X.
2. $\hat{\mathsf{B}}^*$ sends START and then OK to $\mathcal{F}_{\mathsf{COIN}}$. It receives a uniformly random *coin*.
3. Depending on the value of *coin*, $\hat{\mathsf{B}}^*$ applies the corresponding circuit R_{coin} with input W, X, B^* and *coin*.
4. $\hat{\mathsf{B}}^*$ receives output register Y with $|\phi_{good}(\psi)\rangle$ and "measures the conversation" to retrieve the corresponding $(com(a,r), b, open(a,r))$. It outputs whatever B^* outputs.

Fig. 4. The Ideal-World Simulation $\hat{\mathsf{B}}^*$.

First note that the superposition constructed as described above in circuit Q as Step (1) corresponds to all possible random choices of values in the real protocol. Furthermore, the circuit models any possible strategy of quantum B^* in Step (2), depending on control register $|com(a,r)\rangle_{A_1}$. The CNOT-operation on (B,G) in Step (3), followed by a standard measurement of G, indicate whether the guess b' on B^*'s choice b was correct. If that was not the case (i.e. $b \neq b'$ and measurement result 1), the system gets quantumly rewound by applying reverse transformations (3)-(1), followed by a phase-flip operation. The procedure is repeated until the measurement outcome is 0 and hence $b = b'$. Watrous' technique then guarantees that, assuming negligible ε and non-negligible p_0, then ε' is negligible and thus, the final output $\rho(\psi)$ of the

simulation is close to good state $|\phi_{good}(\psi)\rangle$. It follows that the output of the ideal simulation is indistinguishable from the output in the real-world for any quantum-computationally bounded B^*. □

4 Applications

4.1 Interactive Quantum Zero-Knowledge

Zero-knowledge proofs are an important building block for larger cryptographic protocols. The notion of (interactive) zero-knowledge (ZK) was introduced by Goldwasser et al. [11]. Informally, ZK proofs for any NP language L yield no other knowledge to the verifier than the validity of the assertion proved, i.e. $x \in L$. Thus, only this one bit of knowledge is communicated from prover to verifier and zero additional knowledge. For a survey about zero-knowledge, see for instance [9, 10].

Blum et al. [2] showed that the interaction between prover and verifier in any ZK proof can be replaced by sharing a short, random common reference string according to some distribution and available to all parties from the start of the protocol. Note that a CRS is a weaker requirement than interaction. Since all information is communicated mono-directional from prover to verifier, we do not have to require any restriction on the verifier.

As in the classical case, where ZK protocols exist if one-way functions exist, quantum zero-knowledge (QZK) is possible under the assumption that quantum one-way functions exist. In [14], Kobayashi showed that a common reference string or shared entanglement is necessary for non-interactive quantum zero-knowledge. Interactive quantum zero-knowledge protocols in restricted settings were proposed by Watrous in the honest verifier setting [19] and by Damgård et al. in the CRS model [5], where the latter introduced the first Σ-protocols for QZK withstanding even active quantum attacks. In [20], Watrous then proved that several interactive protocols are zero-knowledge against general quantum attacks.

Recently, Hallgren et al. [13] showed how to transform a Σ-protocol with stage-by-stage honest verifier zero-knowledge into a new Σ-protocol that is zero-knowledge against all classical and quantum verifiers. They propose special bit commitment schemes to limit the number of rounds, and view each round as a stage in which an honest verifier simulator is assumed. Then, by using a technique of [7], each stage can be converted to obtain zero-knowledge against any classical verifier. Finally, Watrous' quantum rewinding lemma is applied in each stage to prove zero-knowledge also against any quantum verifier.

Here, we propose a simpler transformation from non-interactive (quantum) zero-knowledge (NIZK) to interactive quantum zero-knowledge (IQZK) by combining the $\texttt{Coin}-\texttt{Flip Protocol}$ with any $\texttt{NIZK Protocol}$. Our coin-flipping generates a truly random *coin* even in the case of a malicious quantum verifier. A sequence of such coins can then be used in any subsequent $\texttt{NIZK Protocol}$, which is also secure against quantum verifiers, due to its mono-direction. Here, we define a ($\texttt{NIZK}$)-subprotocol as given in [2]: Both parties A and B get common input x. A common reference string ω of size k allows the prover A, who knows a witness w, to give a non-interactive zero-knowledge proof $\pi(\omega, x)$ to a (quantum-) computationally bounded verifier B. By definition, the ($\texttt{NIZK}$)-subprotocol is complete and sound and satisfies zero-knowledge.

The $\texttt{IQZK Protocol}$ is shown in Figure 7. To prove that it is an interactive quantum zero-knowledge protocol, we first construct an intermediate $\texttt{IQZK}^{\mathcal{F}_{\mathsf{COIN}}}\ \texttt{Protocol}$ (see Fig. 5) that runs with the ideal functionality $\mathcal{F}_{\mathsf{COIN}}$. Then we prove that the $\texttt{IQZK}^{\mathcal{F}_{\mathsf{COIN}}}\ \texttt{Protocol}$ satisfies completeness, soundness and zero-knowledge according to standard definitions. Finally, by replacing the calls to $\mathcal{F}_{\mathsf{COIN}}$ with our $\texttt{Coin}-\texttt{Flip Protocol}$, we can complete the transformation to the final $\texttt{IQZK Protocol}$.

Completeness: *If $x \in L$, the probability that (A, B) rejects x is negligible in the length of x.*

$\texttt{IQZK}^{\mathcal{F}_{\mathsf{COIN}}}$ Protocol:

(COIN)
1. A and B invoke $\mathcal{F}_{\mathsf{COIN}}$ k times. If A blocks any output $coin_i$ for $i = 1, \ldots, k$ (by sending REFUSE as second input), B aborts the protocol.

(CRS)
2. A and B compute $\omega = coin_1 \ldots coin_k$.

(NIZK)
3. A sends $\pi(\omega, x)$ to B. B checks the proof and accepts or rejects accordingly.

Fig. 5. Intermediate Protocol for IQZK.

From the ideal functionality $\mathcal{F}_{\mathsf{COIN}}$ it follows that each $coin_i$ in Step 1 is uniformly random for all $i = 1, \ldots, k$. Hence, ω in Step 2 is a uniformly random common reference string of size k. By definition of any (NIZK)-subprotocol, we have acceptance probability

$$Pr[\omega \in_R \{0,1\}^k, \pi(\omega, x) \leftarrow A(\omega, x, w) : B(\omega, x, \pi(\omega, x)) = 1] > 1 - \varepsilon'',$$

where ε'' is negligible in the length of x. Thus, completeness for the $\texttt{IQZK}^{\mathcal{F}_{\mathsf{COIN}}}$ Protocol follows.

Soundness: *If $x \notin L$, then for any unbounded prover A^*, the probability that $(\mathsf{A}^*, \mathsf{B})$ accepts x is negligible in the length of x.*

Any dishonest A^* might stop the $\texttt{IQZK}^{\mathcal{F}_{\mathsf{COIN}}}$ Protocol at any point during execution. For example, she can block the output in Step 1 or she can refuse to send a proof π in the (NIZK)-subprotocol. Furthermore, A^* can use an invalid ω (or x) for π. In all of these cases, B will abort without even checking the proof. Therefore, A^*'s best strategy is to "play the entire game", i.e. to execute the entire $\texttt{IQZK}^{\mathcal{F}_{\mathsf{COIN}}}$ Protocol without making obvious cheats.

A^* can only convince B in the (NIZK)-subprotocol of a π for any given (i.e. normally generated) ω with negligible probability

$$Pr[\omega \in_R \{0,1\}^k, \pi(\omega, x) \leftarrow A^*(\omega, x) : B(\omega, x, \pi(\omega, x)) = 1] \ .$$

Therefore, the probability that A^* can convince B in the entire $\texttt{IQZK}^{\mathcal{F}_{\mathsf{COIN}}}$ Protocol in case of $x \notin L$ is also negligible (in the length of x) and its soundness follows.

Zero-Knowledge: *An interactive proof system $(\mathsf{A}, \mathsf{B}^*)$ for language L is quantum zero-knowledge, if for any quantum verifier B^*, there exists a simulator $\hat{\mathsf{S}}_{\texttt{IQZK}^{\mathcal{F}_{\mathsf{COIN}}}}$, such that $\hat{\mathsf{S}}_{\texttt{IQZK}^{\mathcal{F}_{\mathsf{COIN}}}} \overset{q}{\approx} (\mathsf{A}, \mathsf{B}^*)$ on common input $x \in L$ and arbitrary additional (quantum) input to B^*.*

We construct simulator $\hat{\mathsf{S}}_{\texttt{IQZK}^{\mathcal{F}_{\mathsf{COIN}}}}$, interacting with dishonest B^* and simulator $\hat{\mathsf{S}}_{\texttt{NIZK}}$. Under the assumption on the zero-knowledge property of any NIZK Protocol, there exists a simulator $\hat{\mathsf{S}}_{\texttt{NIZK}}$ that, on input $x \in L$, generates a randomly looking ω together with a valid proof π for x (without knowing witness w). $\hat{\mathsf{S}}_{\texttt{IQZK}^{\mathcal{F}_{\mathsf{COIN}}}}$ is described in Figure 6. It receives a random string ω from $\hat{\mathsf{S}}_{\texttt{NIZK}}$, which now replaces the string of coins produced by the calls to $\mathcal{F}_{\mathsf{COIN}}$ in the $\texttt{IQZK}^{\mathcal{F}_{\mathsf{COIN}}}$ Protocol. The "merging" of coins into ω in Step 2 of the protocol (Fig. 5) is equivalent to the "splitting" of ω into coins in Step 3 of the simulation (Fig. 6). Thus, the simulated proof $\pi(\omega, x)$ is indistinguishable from a real proof, which shows that the $\texttt{IQZK}^{\mathcal{F}_{\mathsf{COIN}}}$ Protocol is zero-knowledge.

It would be natural to think that the IQZK Protocol could be proved secure simply by showing that the $\texttt{IQZK}^{\mathcal{F}_{\mathsf{COIN}}}$ Protocol implements some appropriate functionality and then use the composition theorem from [8]. Unfortunately, a zero-knowledge protocol – which is not necessarily a proof of knowledge – cannot be modeled by a functionality in a natural way. We therefore

$\hat{\mathsf{S}}_{\mathtt{IQZK}^{\mathcal{F}_{\mathsf{COIN}}}}$:

1. $\hat{\mathsf{S}}_{\mathtt{IQZK}^{\mathcal{F}_{\mathsf{COIN}}}}$ gets input x.
2. It invokes $\hat{\mathsf{S}}_{\mathtt{NIZK}}$ with x and receives $\pi(\omega, x)$.
3. Let $\omega = coin_1 \dots coin_k$. $\hat{\mathsf{S}}_{\mathtt{IQZK}^{\mathcal{F}_{\mathsf{COIN}}}}$ sends each $coin_i$ one by one to B^*.
4. $\hat{\mathsf{S}}_{\mathtt{IQZK}^{\mathcal{F}_{\mathsf{COIN}}}}$ sends $\pi(\omega, x)$ to B^* and outputs whatever B^* outputs.

Fig. 6. The Simulation of the Intermediate Protocol for IQZK.

`IQZK Protocol:`

(`CFP`) For all $i = 1, \dots, k$ repeat Steps 1. – 4.
1. A chooses $a_i \in_R \{0,1\}$ and computes $com(a_i, r_i)$. She sends $com(a_i, r_i)$ to B.
2. B chooses $b_i \in_R \{0,1\}$ and sends b_i to A.
3. A sends $open(a_i, r_i)$ and B checks if the opening is valid.
4. Both compute $coin_i = a_i \oplus b_i$.

(`CRS`)
5. A and B compute $\omega = coin_1 \dots coin_k$.

(`NIZK`)
6. A sends $\pi(\omega, x)$ to B. B checks the proof and accepts or rejects accordingly.

Fig. 7. Interactive Quantum Zero-Knowledge.

instead prove explicitly that the `IQZK Protocol` has the standard properties of a zero-knowledge proof as follows.

Completeness: From the analysis of the `Coin − Flip Protocol` and its indistinguishability from the ideal functionality $\mathcal{F}_{\mathsf{COIN}}$, it follows that if both players honestly choose random bits, each $coin_i$ for all $i = 1, \dots, k$ in the (`CFP`)-subprotocol is generated uniformly at random. Thus, ω is a random common reference string of size k and the acceptance probability of the (`NIZK`)-subprotocol as given above holds. Completeness for the `IQZK Protocol` follows.

Soundness: Again, we only consider the case where A^* executes the entire protocol without making obvious cheats, since otherwise, B immediately aborts. Assume that A^* could cheat in the `IQZK Protocol`, i.e., B would accept an invalid proof with non-negligible probability. Then we could combine A^* with simulator $\hat{\mathsf{A}}^*$ of the `Coin − Flip Protocol` (Fig. 3) to show that the $\mathtt{IQZK}^{\mathcal{F}_{\mathsf{COIN}}}$ `Protocol` was not sound. This, however, is inconsistent with the previously given soundness argument and thus proves by contradiction that the `IQZK Protocol` is sound.

Zero-Knowledge: A simulator $\hat{\mathsf{S}}_{\mathtt{IQZK}}$ can be composed of simulator $\hat{\mathsf{S}}_{\mathtt{IQZK}^{\mathcal{F}_{\mathsf{COIN}}}}$ (Fig. 6) and simulator $\hat{\mathsf{B}}^*$ for the `Coin − Flip Protocol` (Fig. 4). $\hat{\mathsf{S}}_{\mathtt{IQZK}}$ gets classical input x as well as quantum input W and X. It then receives a valid proof π and a random string ω from $\hat{\mathsf{S}}_{\mathtt{NIZK}}$. As in $\hat{\mathsf{S}}_{\mathtt{IQZK}^{\mathcal{F}_{\mathsf{COIN}}}}$, ω is split into $coin_1 \dots coin_k$. For each $coin_i$, it will then invoke $\hat{\mathsf{B}}^*$ to simulate one coin-flip execution with $coin_i$ as result. In other words, whenever $\hat{\mathsf{B}}^*$ asks $\mathcal{F}_{\mathsf{COIN}}$ to output a bit (Step 2, Fig. 4), it instead receives this $coin_i$. The transcript of the simulation, i.e. $\pi(\omega, x)$ as well as $(com(a_i, r_i), b_i, open(a_i, r_i))$ $\forall i = 1, \dots, k$ and $\omega = coin_1 \dots coin_k$, is indistinguishable from the transcript of the `IQZK Protocol` for any quantum-computationally bounded B^*, which concludes the zero-knowledge proof.

4.2 Generating Commitment Keys for Improved Quantum Protocols

Recently, Damgård et al. [4] proposed a general compiler for improving the security of a large class of quantum protocols. Alice starts such protocols by transmitting random BB84-qubits to Bob who measures them in random bases. Then some classical messages are exchanged to accomplish different cryptographic tasks. The original protocols are typically unconditionally secure against cheating Alice, and secure against a so-called *benignly* dishonest Bob, i.e., Bob is assumed to handle most of the received qubits as he is supposed to. Later on in the protocol, he can deviate arbitrarily. The improved protocols are then secure against an arbitrary computationally bounded (quantum) adversary. The compilation also preserves security in the bounded-quantum-storage model (BQSM) that assumes the quantum storage of the adversary to be of limited size. If the original protocol was BQSM-secure, the improved protocol achieves hybrid security, i.e., it can only be broken by an adversary who has large quantum memory *and* large computing power.

Briefly, the argument for computational security proceeds along the following lines. After the initial qubit transmission from A to B, B commits to all his measurement bases and outcomes. The (keyed) dual-mode commitment scheme that is used must have the special properties that the key can be generated by one of two possible key-generation algorithms: $\mathcal{G}_{\mathtt{H}}$ or $\mathcal{G}_{\mathtt{B}}$. Depending of the key in use, the scheme provides both flavors of security. Namely, with key $\mathtt{pkH}$ generated by $\mathcal{G}_{\mathtt{H}}$, respectively $\mathtt{pkB}$ produced by $\mathcal{G}_{\mathtt{B}}$, the commitment scheme is unconditionally hiding respectively unconditionally binding. Furthermore, the scheme is secure against a quantum adversary and it holds that $\mathtt{pkH} \overset{q}{\approx} \mathtt{pkB}$. The commitment construction is described in full detail in [4].

In the real-life protocol, B uses the unconditionally hiding key $\mathtt{pkH}$ to maintain unconditional security against any unbounded A^*. To argue security against a computationally bounded B^*, an information-theoretic argument involving simulator $\hat{\mathsf{B}}'$ (see [4]) is given to prove that B^* cannot cheat with the unconditionally binding key $\mathtt{pkB}$. Security in real life then follows from the quantum-computational indistinguishability of $\mathtt{pkH}$ and $\mathtt{pkB}$.

The CRS model is assumed to achieve high efficiency and practicability. Here, we discuss integrating the generation of a common reference string from scratch based on our quantum-secure coin-flipping. Thus, we can implement the *entire process* in the quantum world, starting with the generation of a CRS without any initially shared information and using it during compilation as commitment key.[3]

As mentioned in [4], a dual-mode commitment scheme can be constructed from the lattice-based cryptosystem of Regev [18]. It is based on the learning with error problem, which can be reduced from worst-case (quantum) hardness of the (general) shortest vector problem. Hence, breaking Regev's cryptosystem implies an efficient algorithm for approximating the lattice problem, which is assumed to be hard even quantumly. Briefly, the cryptosystem uses dimension k as security parameter and is parametrized by two integers m and p, where p is a prime, and a probability distribution on $\mathbb{Z}_p$. A regular public key for Regev's scheme is indistinguishable from a case where a public key is chosen independently from the secret key, and in this case, the ciphertext carries essentially no information about the message. Thus, the public key of a regular key pair can be used as the unconditional binding key $\mathtt{pkB}'$ in the commitment scheme for the ideal-world simulation. Then for the real protocol, an unconditionally hiding commitment key $\mathtt{pkH}'$ can simply be constructed by uniformly choosing numbers in $\mathbb{Z}_p^k \times \mathbb{Z}_p$. Both public keys will be of size $O(mk \log p)$, and the encryption process involves only modular additions, which makes its use simple and efficient.

The idea is now the following. We add (at least) k executions of our $\mathtt{Coin-Flip\ Protocol}$ as a first step to the construction of [4] to generate a uniformly random sequence $coin_1 \dots coin_k$. These k random bits produce a $\mathtt{pkH}'$ as sampled by $\mathcal{G}_{\mathtt{H}}$, except with negligible probability. Hence, in the real-world, Bob can use $coin_1 \dots coin_k = \mathtt{pkH}'$ as key for committing to all his basis

[3] Note that implementing the entire process comes at the cost of a non constant-round construction, added to otherwise very efficient protocols under the CRS-assumption.

choices and measurement outcomes. Since an ideal-world adversary $\hat{\mathsf{B}}'$ is free to choose any key, it can generate $(\mathtt{pkB}', \mathtt{sk}')$, i.e. a regular public key together with a secret key according to Regev's cryptosystem. For the security proof, write $\mathtt{pkB}' = coin_1 \dots coin_k$. In the simulation, $\hat{\mathsf{B}}'$ first invokes $\hat{\mathsf{B}}^*$ for each $coin_i$ to simulate one coin-flip execution with $coin_i$ as result. As before, whenever $\hat{\mathsf{B}}^*$ asks $\mathcal{F}_{\mathsf{COIN}}$ to output a bit, it instead receives this $coin_i$. Then $\hat{\mathsf{B}}'$ has the possibility to decrypt dishonest B^*'s commitments during simulation, which binds B^* unconditionally to his committed measurement bases and outcomes. Finally, as we proved in the analysis of the $\mathtt{Coin-Flip\ Protocol}$ that $\mathtt{pkH}'$ is a uniformly random string, Regev's proof of semantic security shows that $\mathtt{pkH}' \overset{q}{\approx} \mathtt{pkB}'$, and (quantum-) computational security of the real protocols in [4] follows.

5 On Efficient Simulation in the CRS Model

For our $\mathtt{Coin-Flip\ Protocol}$ in the plain model, we cannot claim universal composability. As already mentioned, in case of unconditional security against dishonest A^* according to Definition 2.2, we do not require the simulator to be efficient. In order to achieve efficient simulation, $\hat{\mathsf{A}}^*$ must be able to extract the choice bit efficiently out of A^*'s commitment, such that A^*'s input is defined after this step. The standard approach to do this is to give the simulator some trapdoor information related to the common reference string, that A^* does not have in real life. Therefore, we extend the commitment scheme to build in such a trapdoor and ensure efficient extraction. To further guarantee UC-security, we circumvent the necessity of rewinding B^* by extending the construction also with respect to equivocability.

We will adapt an approach to our set-up, which is based on the idea of UC-commitments [3] and already discussed in the full version of [4]. We require a Σ-protocol for a (quantumly) hard relation $R = \{(x, w)\}$, i.e. an honest verifier perfect zero-knowledge interactive proof of knowledge, where the prover shows that he knows a witness w such that the problem instance x is in the language L ($(x, w) \in R$). Conversations are of form $(a_\Sigma, c_\Sigma, z_\Sigma)$, where the prover sends a_Σ, the verifier challenges him with bit c_Σ, and the prover replies with z_Σ. For practical candidates of R, see e.g. [5]. Instead of the simple commitment scheme, we use the keyed dual-mode commitment scheme described in Section 4.2 but now based on a multi-bit version of Regev's scheme [17]. Still we construct it such that depending of the key $\mathtt{pkH}$ or $\mathtt{pkB}$, the scheme provides both flavors of security and it holds that $\mathtt{pkH} \overset{q}{\approx} \mathtt{pkB}$.

In real life, the CRS consists of commitment key $\mathtt{pkB}$ and an instance x' for which it holds that $\nexists\ w'$ such that $(x', w') \in R$, where we assume that $x \overset{q}{\approx} x'$. To commit to bit a, A runs the honest verifier simulator to get a conversation (a_Σ, a, z_Σ). She then sends a_Σ and two commitments c_0, c_1 to B, where $c_a = com_{\mathtt{pkB}}(z_\Sigma, r)$ and $c_{1-a} = com_{\mathtt{pkB}}(0^{z'}, r')$ with randomness r, r' and $z' = |z|$. Then, a, z_Σ, r is send to open the relevant one of c_0 or c_1, and B checks that (a_Σ, a, z_Σ) is an accepting conversation. Assuming that the Σ-protocol is honest verifier zero-knowledge and $\mathtt{pkB}$ leads to unconditionally binding commitments, the new commitment construction is again unconditionally binding.

During simulation, $\hat{\mathsf{A}}^*$ chooses a $\mathtt{pkB}$ in the CRS such that it knows the matching decryption key $\mathtt{sk}$. Then, it can extract A^*'s choice bit a by decrypting both c_0 and c_1 and checking which contains a valid z_Σ such that (a_Σ, a, z_Σ) is accepting. Note that not both c_0 and c_1 can contain a valid reply, since otherwise, A^* would know a w' such that $(x', w') \in R$. In order to simulate in case of B^*, $\hat{\mathsf{B}}^*$ chooses the CRS as $\mathtt{pkH}$ and x. Hence, the commitment is unconditionally hiding. Furthermore, it can be equivocated, since $\exists\ w$ with $(x, w) \in R$ and therefore, c_0, c_1 can both be computed with valid replies, i.e. $c_0 = com_{\mathtt{pkH}}(z_{0\Sigma}, r)$ and $c_1 = com_{\mathtt{pkH}}(z_{1\Sigma}, r')$. Quantum-computational security against B^* follows from the indistinguishability of the keys $\mathtt{pkB}$ and $\mathtt{pkH}$ and the indistinguishablity of the instances x and x', and efficiency of both simulations is ensured due to extraction and equivocability.

Acknowledgments

We thank Christian Schaffner and Serge Fehr for useful comments on an earlier version of the paper and the former also for discussing the issue of efficient simulation in earlier work. CL acknowledges financial support by the MOBISEQ research project funded by NABIIT, Denmark.

References

1. Charles H. Bennett, Gilles Brassard, Claude Crépeau, and Marie-Hélène Skubiszewska. Practical quantum oblivious transfer. In *Advances in Cryptology—CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*, pages 351–366. Springer, 1991.
2. Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *20th Annual ACM Symposium on Theory of Computing (STOC)*, pages 103–112, 1988.
3. Ran Canetti and Marc Fischlin. Universally composable commitments. In *Advances in Cryptology—CRYPTO '01*, volume 2139 of *Lecture Notes in Computer Science*, pages 19–40. Springer, 2001.
4. Ivan B. Damgård, Serge Fehr, Carolin Lunemann, Louis Salvail, and Christian Schaffner. Improving the security of quantum protocols via commit-and-open. In *Advances in Cryptology—CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 408–427. Springer, 2009. Full version available at: `http://arxiv.org/abs/0902.3918`.
5. Ivan B. Damgård, Serge Fehr, and Louis Salvail. Zero-knowledge proofs and string commitments withstanding quantum attacks. In *Advances in Cryptology—CRYPTO '04*, volume 3152 of *Lecture Notes in Computer Science*, pages 254–272. Springer, 2004.
6. Ivan B. Damgård, Serge Fehr, Louis Salvail, and Christian Schaffner. Secure identification and QKD in the bounded-quantum-storage model. In *Advances in Cryptology—CRYPTO '07*, volume 4622 of *Lecture Notes in Computer Science*, pages 342–359. Springer, 2007.
7. Ivan B. Damgård, Oded Goldreich, and Avi Wigderson. Hashing functions can simplify zero-knowledge protocol design (too). Technical Report RS-94-39, BRICS, Department of Computer Science, Aarhus University, Denmark, 1994.
8. Serge Fehr and Christian Schaffner. Composing quantum protocols in a classical environment. In *Theory of Cryptography Conference (TCC)*, volume 5444 of *Lecture Notes in Computer Science*, pages 350–367. Springer, 2009.
9. Oded Goldreich. *Foundations of Cryptography*, volume I: Basic Tools. Cambridge University Press, 2001.
10. Oded Goldreich. Zero-knowledge twenty years after its invention. Available at `http://www.wisdom.weizmann.ac.il/~oded/papers.html`, 2002.
11. Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *17th Annual ACM Symposium on Theory of Computing (STOC)*, pages 291–304, 1985.
12. Jeroen van de Graaf. *Towards a formal definition of security for quantum protocols*. PhD thesis, Université de Montréal, 1997.
13. Sean Hallgren, Alexandra Kolla, Pranab Sen, and Shengyu Zhang. Making classical honest verifier zero knowledge protocols secure against quantum attacks. In *35th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 5126 of *Lecture Notes in Computer Science*, pages 592–603. Springer, 2008.
14. Hirotada Kobayashi. Non-interactive quantum perfect and statistical zero-knowledge. In *ISAAC*, pages 178–188, 2003.
15. Moni Naor. Bit commitment using pseudorandomness. *Journal of Cryptology*, 4(2):151–158, 1991.
16. Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge university press, 2000.
17. Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In *Advances in Cryptology—CRYPTO '08*, volume 5157 of *Lecture Notes in Computer Science*, pages 554–571. Springer, 2008.
18. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 84–93, 2005.
19. John Watrous. Limits on the power of quantum statistical zero-knowledge. In *43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 459–468, 2002.
20. John Watrous. Zero-knowledge against quantum attacks. In *SIAM Journal on Computing*, volume 39.1, pages 25–58, 2009. Preliminary version in *38th Annual ACM Symposium on Theory of Computing (STOC)*, pages 296–305, 2006.
21. William K. Wootters and Wojciech H. Zurek. A single quantum cannot be cloned. *Nature*, 299:802–803, October 1982.

A Watrous' Quantum Rewinding Lemma

Lemma A.1 (Quantum Rewinding Lemma with small perturbations [20]). *Let Q be the unitary $(\tilde{n}, \tilde{k})$-quantum circuit as given in [20]. Furthermore, let $p_0, q \in (0,1)$ and $\varepsilon \in (0, \frac{1}{2})$ be real numbers such that*

1. $|p - q| < \varepsilon$
2. $p_0(1-p_0) \leq q(1-q)$, *and*
3. $p_0 \leq p$

for all $\tilde{n}$-qubit states $|\psi\rangle$. Then there exists a general quantum circuit R of size

$$O\left(\frac{log(1/\varepsilon)size(Q)}{p_0(1-p_0)}\right)$$

such that, for every $\tilde{n}$-qubit state $|\psi\rangle$, the output $\rho(\psi)$ of R satisfies

$$\langle \phi_{good}(\psi)|\rho(\psi)|\phi_{good}(\psi)\rangle \geq 1 - \varepsilon'$$

where $\varepsilon' = 16\varepsilon\frac{log^2(1/\varepsilon)}{p_0^2(1-p_0)^2}$.

Note that p_0 denotes the lower bound on the success probability p, for which the procedure guarantees correctness. Furthermore, for negligible ε but non-negligible p_0, it follows that ε' is negligible. For a more detailed description of the lemma and the corresponding proofs, we refer to [20].