Making Classical Honest Verifier Zero Knowledge Protocols Secure against Quantum Attacks

Sean Hallgren[1], Alexandra Kolla[2], Pranab Sen[3], and Shengyu Zhang[4]

[1] Pennsylvania State University, University Park, PA, U.S.A.
[2] U C Berkeley, Berkeley, CA, U.S.A.
akolla@cs.berkeley.edu
[3] Tata Institute of Fundamental Research, Mumbai, India
pgdsen@tcs.tifr.res.in
[4] California Institute of Technology, Pasadena, CA, U.S.A.
shengyu@caltech.edu

Abstract. We show that any problem that has a classical zero-knowledge protocol against the honest verifier also has, under a reasonable condition, a classical zero-knowledge protocol which is secure against all classical and quantum polynomial time verifiers, even cheating ones. Here we refer to the generalized notion of zero-knowledge with classical and quantum auxiliary inputs respectively.

Our condition on the original protocol is that, for positive instances of the problem, the simulated message transcript should be quantum computationally indistinguishable from the actual message transcript. This is a natural strengthening of the notion of honest verifier computational zero-knowledge, and includes in particular, the complexity class of honest verifier statistical zero-knowledge. Our result answers an open question of Watrous [Wat06], and generalizes classical results by Goldreich, Sahai and Vadhan [GSV98], and Vadhan [Vad06] who showed that honest verifier statistical, respectively computational, zero knowledge is equal to general statistical, respectively computational, zero knowledge.

1 Introduction

Zero knowledge protocols are a central concept in cryptography. These protocols allow a prover to convince a verifier about the truth of a statement without revealing any additional information about the statement, even if the verifier *cheats* by deviating from the prescribed protocol. For a nice overview of definitions and facts about zero-knowledge we refer the reader to [Gol01]. In practice, zero-knowledge protocols are used as primitives in larger cryptographic protocols in order to limit the power of malicious parties to disrupt the security of the larger protocol. For example, at the start of a secure online transaction Alice may be required to prove her identity to Bob. She does this by demonstrating that she knows a particular secret which only she is supposed to know. However, Alice wants to prevent the possibility of Bob committing identity theft, that is, Bob should not be able to masquerade as Alice later on. Thus, Bob should gain no information about Alice's secret even if he acts maliciously during the identity verification protocol.

L. Aceto et al. (Eds.): ICALP 2008, Part II, LNCS 5126, pp. 592–603, 2008.

With the advent of quantum computation an important question rears its head: what happens to classical zero-knowledge protocols when the cheating verifier has access to a quantum computer? Note that even if the verifier cheats quantumly, the messages exchanged with the prover and the prover itself continue to be classical. Thus, the prover does not know if it is interacting with a classical or quantum verifier. One may expect that quantum computers can break some classical zero-knowledge protocols, i.e. a quantum verifier interacting with the prover may be able to extract information about from the message transcript (sequence of all messages exchanged) that a classical verifier cannot. As one example, the Feige-Fiat-Shamir [FFS88] zero-knowledge protocol for identity verification can be broken by a quantum computer simply because it relies on the hardness of factoring for security.

Watrous [Wat06] recently showed that two well-known classical protocols continue to be zero-knowledge against cheating quantum verifiers. In particular, he showed that the graph isomorphism protocol of Goldreich, Micali and Wigderson [GMW91] is secure, and also that the graph 3-coloring protocol in [GMW91] is secure if one can find classical commitment schemes that are concealing against quantum computers. However, the general question of which classical zero-knowledge protocols continue to be secure against cheating quantum verifiers was left open by Watrous.

In this paper, we answer this question for a large family of classical protocols. We show that all protocols that are honest verifier zero-knowledge ($\mathbf{HVZK}$) and satisfy some reasonable assumption on their simulated transcripts can be made secure against all efficient classical and quantum machines. More specifically, any protocol which is honest verifier statistical zero-knowledge ($\mathbf{HVSZK}$) can be transformed to be statistical zero-knowledge against all classical and quantum verifiers ($\mathbf{SZKQ}$). Also, any protocol which is honest verifier computational zero-knowledge and has classical message transcripts of the interaction between the prover and the honest verifier that yield no information to an efficient quantum machine ($\mathbf{HVCZK_Q}$), can be transformed to be computational zero knowledge against all classical and quantum verifiers ($\mathbf{CZKQ}$). Note that classically it was shown that any language in $\mathbf{HVCZK}$ also has a protocol which is zero-knowledge against any cheating verifier (the class $\mathbf{CZK}$).

As in the classical case, by starting with fairly weak assumption on protocols, we show that a much stronger protocol exists. Note that being zero-knowledge against quantum verifiers does not imply being zero-knowledge against classical verifiers owing to a technical requirement in the definition of zero-knowledge to be elucidated later. The significance of our result is that we give a single classical protocol zero-knowledge against both types of verifiers. Our work substantially generalizes Watrous' results [Wat06].

Formally, a protocol is said to be zero-knowledge if for every non-uniform polynomial time verifier there is a non-uniform polynomial time simulator that can produce, for inputs in the language, a simulated *view* of the verifier that is indistinguishable to the verifier's view in an actual interaction with the prover. The view of the verifier consists of the message transcript together with the internal state of the verifier, and represents what the verifier can 'learn' from interacting with the prover. The existence of a polynomial time simulator for every polynomial time verifier captures the intuition that the verifier learns nothing that it could not have learned on its own from the input, even

by being malicious. For a classical verifier the simulator is required to be classical. For a quantum verifier the simulator is quantum. Thus, zero-knowledge against quantum verifiers does not immediately imply zero-knowledge against classical verifiers.

Constructing a simulator appears to be counterintuitive since it seems to replace the role of the prover who is usually assumed to be computationally unbounded whereas the simulator is polynomial time. The difference between the prover and the simulator is that the prover has to respond to verifiers queries in an 'online' fashion, that is immediately, whereas the simulator can work 'offline' and generates the messages 'out of turn', as well as 'rewind'. By rewinding, we mean a simulator runs parts of the verifier during the simulation and produces a fragment of the conversation that has some desired property with a certain probability. If the simulator fails then it rewinds, that is it just runs the part of the verifier again from scratch. In the quantum case one would have a quantum simulator using the quantum verifier to produce such a fragment of the conversation and attempting to rewind if it fails.

Protocols that are classically zero-knowledge are not necessarily zero-knowledge against quantum verifiers. In the case of the two problems graph isomorphism and graph 3-coloring that Watrous [Wat06] studied, the essential difference between classical and quantum simulators comes from one additional requirement of zero-knowledge protocols. In order for zero-knowledge protocols to sequentially compose, which is essential to achieve reasonable error parameters as well as ensure the security of the protocol when used as part of a larger cryptographic system, the simulator must still work when the simulators and verifiers are given an arbitrary *auxiliary* state. This is a natural requirement if one considers that, for example, perhaps the verifier has interacted with the prover already to compute some intermediate information modeled by the auxiliary state, and now during the next interaction it gains even more information. In the quantum case the auxiliary state is an unknown *quantum* state. But unknown quantum states cannot be copied, and measurements of unknown quantum states are irreversible operations in general, and as pointed out by Watrous [Wat06], even determining if the simulator was successful in producing a fragment of the conversation with the desired property may destroy the state. Therefore the simulator cannot trivially rewind since it cannot feed the auxiliary state into the verifier a second time if the state was destroyed during the first attempt at simulation. Nevertheless, Watrous [Wat06] showed that it is possible to quantumly rewind in a clever way in the case of Goldreich, Micali and Wigderson's [GMW91] classical zero-knowledge protocols for graph isomorphism and graph 3-coloring.

When searching for more classical zero-knowledge protocols that are secure against quantum cheating verifiers we come across new difficulties not encountered by Watrous [Wat06]. One restriction of the protocols he analyzes is that they are three-round public coin protocols where the second message is $O(\log n)$ uniformly random bits from the verifier. This leaves out many languages in $\mathbf{SZK}$ and $\mathbf{CZK}$ including the complete problems *statistical difference* [SV03] and *entropy difference* [GV97] for $\mathbf{SZK}$. In a different vein [Wat02, Wat06], Watrous showed that every problem in $\mathbf{SZK}$ has a *quantum* protocol that is statistical zero-knowledge against any cheating non-uniform polynomial time quantum verifier. Very recently, Kobayashi [Kob08] extended Watrous' result to the case of quantum protocols that are quantum computationally zero

knowledge. However, it is preferable that the prescribed protocols themselves are classical since they can be implemented using current technology yet remain secure against all potential quantum attacks in the future. In this paper, we show that a large class of polynomial round, polynomial verifier message length classical zero-knowledge protocols can be made secure against cheating quantum verifiers.

Classically, the construction of zero-knowledge protocols has been greatly simplified by showing that $\mathbf{HVSZK}$ or $\mathbf{HVCZK}$ is equal to $\mathbf{SZK}$ or $\mathbf{CZK}$ [GSV98, Vad06]. Concretely, if one can design a protocol for a given language that is zero-knowledge against (only) the honest verifier, which is typically much easier, then there is also a protocol for the language that is zero-knowledge against an arbitrary cheating verifier. We follow this approach: we show that if one can find a classical protocol zero-knowledge for just the honest (classical!) verifier such that the actual and simulated message transcripts with respect to the honest verifier are indistinguishable by polynomial sized quantum circuits, then there is also a classical protocol that is zero-knowledge against all classical and quantum cheating verifiers. More precisely, our result can be stated as:

Result 1

1. $\mathbf{SZK} = \mathbf{HVSZK} = \mathbf{SZKQ}$*, where* $\mathbf{SZKQ}$ *is the class of languages with a classical protocol that is statistical zero knowledge against all classical and quantum verifiers.*
2. $\mathbf{HVCZK_Q} = \mathbf{CZKQ} = \mathbf{CZK_Q}$*, Where* $\mathbf{HVCZK_Q}$ *(resp.* $\mathbf{CZK_Q}$*) is the class of languages with a classical protocol that is honest verifier computational zero-knowledge (resp. computational zero-knowledge) and for YES instances,the classical message transcripts of the interaction between the prover and the honest verifier are quantum computationally indistinguishable from the simulated message transcripts. Similarly,* $\mathbf{CZKQ}$ *is the class of languages with a classical protocol that is computational zero knowledge against all classical and quantum verifiers.*

We note that the classical results $\mathbf{HVSZK} = \mathbf{SZK}$ and $\mathbf{HVCZK} = \mathbf{CZK}$ are known and can be found in Goldreich, Sahai and Vadhan [GSV98] and Vadhan [Vad06]. Also, observe that $\mathbf{HVSZK} \subseteq \mathbf{HVCZK_Q} \subseteq \mathbf{HVCZK}$.

Finally, we would like to remark that the definition of zero knowledge in quantum computation in the literature assumes that we can do error-free computation. Constructing a simulator for a cheating verifier typically involves a polynomial multiplicative factor overhead. Thus in reality, it may happen that a simulator fails to successfully simulate the cheating verifier's view because of additional noise incurred by the overheads. However, if we take the view that noise rates in hardware can be decreased by polynomial factors with polynomial effort, the current definition of zero knowledge in quantum computation is justified.

1.1 Overview of Our Proof: Ideas and Difficulties

Damgård, Goldreich and Wigderson [DGW94] gave a method, hereafter called DGW, for transforming any classical constant round public coin honest verifier zero knowledge protocol into another classical constant round public coin protocol that is zero knowledge against all classical verifiers. We first observe that Watrous' quantum rewinding

trick [Wat06] can be used to show that the new protocol resulting from DGW is secure against all quantum verifiers also. This allows us to handle protocols with verifier messages of polynomial length. The shortcoming is that, as in the classical case, the quantum simulator succeeds in almost correctly simulating the prover-verifier interaction with non-negligible probability only if the original protocol has a constant number of rounds. This arises from the fact that the classical and quantum simulators from DGW 'rewind from scratch', that is, they attempt to simulate all the rounds of the protocol in one shot, and if they fail, they rewind the verifier to the beginning of the protocol. The success probability of one attempt at simulation drops exponentially in the number of rounds, and hence, we can only handle a constant number of rounds using the DGW transformation.

Building on Damgård et al.'s work, Goldreich, Sahai and Vadhan [GSV98] gave a method, hereafter called GSV, for transforming any classical public-coin **HVZK** protocol into another public-coin protocol **ZK** against all classical verifiers. Their transformation handles protocols with a polynomial number of rounds. However, one cannot apply Watrous' quantum rewinding technique [Wat06] to the new protocol resulting from GSV for the following technical reason: the simulator for the new protocol rewinds the new verifier polynomial number of times for each round. In order to do the same thing quantumly using Watrous' rewinding lemma, one needs that for most messages of the verifier in the original protocol, the success probability of the simulation attempt conditioned on the old verifier's message be independent of the quantum auxiliary state. Unfortunately this cannot be ensured for any message of the verifier in the original protocol, and hence, we are unable to show that GSV makes the protocol secure against cheating quantum verifiers.

Our crucial observation is that if the honest-verifier simulator for the original classical public coin **ZK** protocol uses its internal randomness in a *stage-by-stage* fashion, where each stage consists of a constant number of rounds, then applying DGW gives a new protocol which is zero-knowledge against all classical and quantum verifiers. This is still the case even the original protocol has a polynomial number of rounds. This is because now the classical or quantum simulator for the new protocol can rewind the verifier polynomial number of times within each stage, where each iteration preserves the simulated message transcript of the earlier rounds and uses fresh random coins to attempt to simulate the current round. Since the success probability of one simulation attempt for a stage is inverse polynomial as it has a constant number of rounds, polynomially many rewinding steps will result in a successful simulation of the current stage with very high probability. This leads us to the question of which problems possess zero-knowledge protocols with stage-by-stage honest-verifier simulators.

Our next observation is that the standard technique of converting any public coin interactive protocol into a zero-knowledge protocol [IY88, BGG$^+$90] based on bit commitments actually gives rise to a new protocol with a stage-by-stage honest verifier simulator. Note that any interactive protocol can be converted into a public coin protocol [GS89] where the messages of the verifier are uniformly distributed random strings independent of the previous messages of the protocol, and the final decision of the verifier to accept or reject is a deterministic function of the message transcript and the input. The only caveat is that the existence of bit commitment schemes seems to be conditional on the existence of one-way functions. However, the recent work of Vadhan [Vad06], Nguyen and Vadhan [NV06] and Ong and Vadhan [OV08] gives a

way of replacing standard bit commitments by instance-dependent bit commitments, which exist unconditionally as shown by them. An instance-dependent bit commitment scheme is a protocol which depends on the input instance to the problem such that the protocol is hiding on the bit to be committed for positive instances of the problem and binding on the bit for negative instances of the problem. Since the hiding and binding properties are not required to hold simultaneously, the need for unproven assumptions like the existence of one-way functions is avoided. Ong and Vadhan [OV08] show that every problem with an honest verifier zero-knowledge protocol gives rise to a public coin constant round instance dependent bit commitment scheme which is statistically binding on the negative instances. For positive instances, the hiding property of the commitment scheme is statistical if the original protocol is $\mathbf{HVSZK}$, and computational against polynomial sized classical circuits if the original protocol is $\mathbf{HVCZK}$. We can show that their proofs can be modified to ensure that the hiding property is computational against polynomial sized quantum circuits if the original classical protocol is in $\mathbf{HVCZK_Q}$. Replacing the bit commitments in the standard compilation of interactive proofs to zero-knowledge by instance dependent commitments gives us a zero-knowledge protocol with an honest-verifier simulator that uses its internal randomness in a stage-by-stage fashion, where each stage consists of a constant number of rounds. Applying the DGW transformation to such a protocol gives rise to a new public coin classical protocol zero-knowledge against all non-uniform polynomial time classical and quantum verifiers. That fact follows since the success probability of correctly simulating a stage in the new protocol continues to be inverse polynomial and also the simulator for the new protocol can rewind in a stage-by-stage fashion.

2 Preliminaries

2.1 The DGW Transformation

We denote a classical N-round public coin interactive protocol by the notation $(P, V) : (\alpha_1, \beta_1, ..., \alpha_N, \beta_N)$, which means that in the round i, the (honest) classical verifier V sends a uniformly random string α_i and the (honest) classical prover P responds with a string β_i, which in general is a function of the previous transcript and the prover's randomness. Without loss of generality, each α_i has the same length s. Let $t < s$ be a positive integer. Damgård, Goldreich and Wigderson [DGW94] describe a family $\mathcal{F}_{s,t}$ of nearly s-wise independent hash functions from $\{0,1\}^s$ to $\{0,1\}^t$. Every function $f \in \mathcal{F}_{s,t}$ has a description of length s^2 bits and for all $y \in \{0,1\}^t$, $1 \leq |f^{-1}(y)| \leq (s-1)2^{s-t} + 1$, where $f^{-1}(y) := \{x \in \{0,1\}^s : f(x) = y\}$. Computing $f^{-1}(y)$ can be done in randomized time polynomial in s and 2^{s-t}. In DGW, $s - t$ is taken to be logarithmic in the input length, so 2^{s-t} will be a polynomial in the input length. Using this family $\mathcal{F}_{s,t}$, Damgård et al. describe a process to transform a random message $\alpha \in_R \{0,1\}^s$ from the verifier in the original protocol, giving rise to a new protocol with twice as many messages.

1. The verifier chooses f uniformly in $\mathcal{F}_{s,t}$ and sends it to the prover.
2. The prover chooses y uniformly in $\{0,1\}^t$ and sends it to the verifier.
3. The verifier chooses α uniformly in $f^{-1}(y)$ and sends it to the prover.

As described, the second message of the verifier in the DGW transformation is not public coin. However, it can be made public coin by letting the verifier send a random $r \in ((s-1)2^{s-t}+1)!$, which the prover interprets as the $(r \bmod |f^{-1}(y)|)$th element of $f^{-1}(y)$. Note that since $(s-1)2^{s-t}+1$ is polynomial in the input size, r can be described using polynomially many bits. Henceforth, we shall assume that the new protocol arising from the application of DGW is public coin but we shall continue to use the description of DGW given above for simplicity.

Applying DGW to an N-round public coin protocol $(\alpha_1, \beta_1, ..., \alpha_N, \beta_N)$ gives a new public coin protocol $(f_1, y_1, \alpha_1, \beta_1, ..., f_N, y_N, \alpha_N, \beta_N)$ where each β_i is obtained in the same way as the original prover does on seeing the previous $(\alpha_1, ..., \alpha_i)$. The DGW transformation satisfies the following soundness and completeness property which we will crucially use [DGW94].

Fact 1. *Suppose the original N-round public coin protocol has perfect completeness and soundness error ϵ_0, then the DGW transformation gives a new public coin protocol with perfect completeness and soundness error $\epsilon_1 = \epsilon_0 + N(2s2^{(t-s)/4} + 2^{-s})$.*

The zero knowledge properties of DGW will be the main topic of discussion in the later sections of this paper.

2.2 Stage-by-Stage Simulator

We now give the formal definition of the important notion of an interactive protocol possessing a 'stage-by-stage' honest-verifier simulator, which is central to our work.

Definition 1. *Suppose (P, V) is a classical public coin protocol with N stages, each stage i containing constant number c of rounds $(\alpha_{i1}, \beta_{i1}, ..., \alpha_{ic}, \beta_{ic})$, where α_{ij}, β_{ij} are verifier's, respectively prover's messages and all α_{ij}s are of the same length. We say that an honest-verifier simulator M is* stage-by-stage *if its internal random string r can be decomposed as $r = r_1 \circ \cdots \circ r_N$, $r_1, \ldots, r_N$ uniform and independent random variables, such that in each stage i, the simulated messages $(\hat{\beta}_{i1}, \ldots, \hat{\beta}_{ic})$ are functions of $r_1, \ldots, r_i$ and the input alone, and $(\hat{\alpha}_{i1}, \ldots, \hat{\alpha}_{ic})$ is a function of r_i alone.*

A public coin constant round protocol can be trivially considered to be a stage-by-stage with only one stage. Note that we do not assume anything about how the simulator uses its randomness in each stage; it can be used arbitrarily. But since each stage only contains a constant number of rounds, rewinding to the beginning of the stage is affordable while simulating the new protocol arising from the application of DGW.

2.3 Instance-Dependent Bit Commitments

We recall the definition of *instance-dependent bit commitment* protocols [OV08] which will be used in our construction of interactive protocols with honest-verifier stage-by-stage simulators. Below, by an *exponentially small* function $\epsilon(n)$ we mean a function of a positive parameter n that grows smaller than 2^{-n^c} for some fixed $c > 0$. By the *total variation distance*, also known as *statistical distance*, between two probability distributions P, Q on the same sample space, we mean the ℓ_1-distance $\|P - Q\| = \sum_i |P(i) - Q(i)|$.

Definition 2. *For a promise problem $\Pi = (\Pi_Y, \Pi_N)$, a classical public coin constant round instance-dependent bit commitment scheme consists of a classical public coin interactive protocol Com_x for every $x \in \Pi_Y \cup \Pi_N$ between two parties called* sender S_x *and* receiver R_x*, with the following properties:*

1. *Protocol Com_x has two stages, a* commit *stage and a* reveal *stage;*
2. *At the beginning of the commit stage, S_x gets a private input $b \in \{0,1\}$ which represents the bit he has to commit to. The commit stage proceeds for a constant number of rounds, and its transcript $c_{x;b}$ is defined to be the* commitment *to the bit b;*
3. *Later on, in the reveal stage, S_x reveals the bit b and sends another string $d_{x;b}$ called the* decommitment *string for b. The receiver R_x accepts or rejects deterministically based on $c_{x;b}$, b and $d_{x;b}$.*
4. *Sender S_x and receiver R_x can be implemented in randomized time polynomial in $|x|$;*
5. *For all $x \in \Pi_Y \cup \Pi_N$, for all $b \in \{0,1\}$, R_x accepts with probability 1 if both S_x and R_x follow the prescribed protocol;*

The scheme Com_x is said to be exponentially binding statistically *for all $x \in \Pi_N$, if for any sender S_x^*, there exists an exponentially small function $\epsilon(\cdot)$ such that if c_x^* denotes the commitment obtained by the interaction of S_x^* and the honest R_x, the probability that there exist decommitment strings $d_{x;0}^*$, $d_{x;1}^*$ in the reveal stage so that R_x accepts on c_x^*, 0, $d_{x;0}^*$ as well as c_x^*, 1, $d_{x;1}^*$ is less than $\epsilon(|x|)$. The binding property is required to hold for malicious senders too who do not follow the prescribed protocol. In addition, the scheme Com_x is said to be* exponentially hiding statistically *for all $x \in \Pi_Y$ if the views of the honest receiver R_x when $b = 0$ and $b = 1$ have exponentially small total variation distance. Similarly, if the two views are negligibly distinguishable by polynomial sized classical or quantum circuits, the scheme Com_x is said to be* computationally*, respectively* quantum computationally*, hiding.*

Remark: Observe that we only require the hiding property to hold for the honest receiver R_x in the above definition. The reason for this is as follows. As mentioned earlier in the introduction, our initial aim is only to get a protocol with a stage-by-stage honest verifier simulator. We will then make that protocol resilient against all malicious verifiers by applying the DGW transformation. The hiding property of the commitment scheme against the honest receiver translates to zero knowledge against the honest verifier in Proposition 1, where we show how to achieve our initial aim.

3 Applying DGW to Protocols with Stage-by-Stage Simulators

In this section, we will show that applying the DGW transformation to a classical public coin interactive protocol with a stage-by-stage honest verifier simulator results in a classical public coin protocol zero-knowledge against all non-uniform polynomial time classical and quantum verifiers.

Lemma 1. *If a classical public-coin protocol $\mathcal{P}$ has a stage-by-stage honest-verifier simulator M such that the simulated transcript is quantum computationally indistinguishable from the actual prover honest-verifier interaction, then applying DGW to it gives a new classical public coin protocol $\mathcal{P}'$ with inverse polynomially larger soundess error that is computationally zero-knowledge against all non-uniform polynomial time classical and quantum verifiers. If in addition $\mathcal{P}$ is statistical zero knowledge against the honest verifier, $\mathcal{P}'$ is statistically zero knowledge against all non-uniform polynomial time classical and quantum verifiers.*

Proof. **(Sketch)** The claim about soundness error follows from Fact 1 with an appropriate setting of the parameters of the DGW transformation. The zero-knowledge property crucially relies on the stage-by-stage assumption and the zero-knowledge property of DGW. Below we sketch the main points of difference from the standard classical setting.

First, the classical proof attempts to simulate all the rounds of the protocol failing which it rewinds from scratch. Here, we do a stage-by-stage simulation, that is, we try to simulate all the rounds of one stage failing which we rewind to the beginning of the stage only. The stage-by-stage property of the honest-verifier simulator M allows us to do this, since rewinding to the beginning of stage i just means tossing a fresh coin r_i without disturbing the earlier coin tosses $r_1, \ldots, r_{i-1}$. Since each stage consists of only a constant number of rounds, the success probability of one attempt at simulating DGW on a stage is inverse polynomial. Thus polynomially many rewinding steps for a stage suffices to simulate the stage successfully with very high probability. After successfully simulating a stage, we can proceed to simulating the next stage, and so on for polynomially many stages.

The second point of difference is that in the proof of security against quantum verifiers, we use Watrous' rewinding technique [Wat06] at the end of a stage. The reason this is possible is because the DGW transformation ensures that the success probability of one attempt at simulation of a stage is independent of the quantum auxiliary input. Combined with the observation above that the probability of successfully simulating a stage is inverse polynomial, this allows us to rewind a stage polynomially many times quantumly without disturbing previous stages and ensure a successful simulation with very high probability. □

A more formal proof of the classical and quantum parts of the above lemma is given in the appendix.

4 Designing Protocols with Stage-by-Stage Simulators

In this section, we indicate how to design a classical public coin interactive protocol for any promise problem in $\mathbf{HVSZK}$ and $\mathbf{HVCZK_Q}$ with perfect completeness, exponentially small soundness and possessing a stage-by-stage honest-verifier simulator. For problems in $\mathbf{HVSZK}$ the simulated transcript will be exponentially close in total variation distance to the actual transcript, and for problems in $\mathbf{HVCZK_Q}$ the two transcripts will be negligibly distinguishable against polynomial sized quantum circuits.

The following statement follows by modifying the arguments of Vadhan [Vad06]. But first, we have to define the notion of a *quantumly secure false entropy generator* which is the natural quantum generalization of a so-called false entropy generator [HILL99].

Definition 3. *Let $I \subseteq \{0,1\}^*$, and $m(\cdot)$ be a polynomial function. For $x \in I$, a family D_x of probability distributions on $\{0,1\}^{m(|x|)}$ is said to be* P-sampleable *if there exists a probabilistic polynomial time algorithm whose output is distributed according to D_x on input x. A P-*sampleable *family D_x is said to be a* quantumly secure false entropy generator *if there exists a family F_x of probability distributions on $\{0,1\}^{m(|x|)}$ that is negligibly distinguishable from D_x by polynomial sized quantum circuits such that $H(F_x) \geq H(D_x) + 1$, where $H(\cdot)$ is the Shannon entropy of a probability distribution.*

Lemma 2. *Suppose $\Pi = (\Pi_Y, \Pi_N)$ is a promise problem in $\mathbf{HVCZK_Q}$. Then there is a family $\{D_x\}_{x \in \Pi_Y \cup \Pi_N}$ of $\mathbf{P}$-sampleable probability distributions on $\{0,1\}^{m(|x|)}$, and a subset $I \subseteq \Pi_Y$ such that $\{D_y\}_{y \in I}$ is a quantumly secure false entropy generator. Also, $(\Pi_Y \setminus I, \Pi_N) \in \mathbf{HVSZK}$.*

Proof. **(Sketch)** The proof follows by observing that the arguments of [Vad06] go through equally well for quantum indistinguishability as for classical indistinguishability. Essentially, this is because the proof of [Vad06] uses reducibility arguments where the computational hardness of a primitive is used as a black box. A more detailed proof is left for the full version of the paper. □

We need the following result about the existence of classical public coin constant round instance dependent bit commitment protocols for problems in $\mathbf{HVSZK}$ by Ong and Vadhan [OV08].

Fact 2. *Every promise problem in $\mathbf{HVSZK}$ gives rise to a classical constant round public coin instance dependent bit commitment scheme that is exponentially hiding on the positive instances and exponentially binding on the negative instances statistically.*

Remark: In fact for our purposes, we do not really require the full strength of the above fact. A weaker primitive of classical constant round public coin instance-dependent *two-phase* bit commitment scheme that is statistically hiding on the positive instances and statistically 1-*out-of*-2 binding on the negative instances suffices for us. Such schemes were first constructed by Nguyen and Vadhan [NV06]. However, our construction of an interactive protocol with a stage-by-stage honest-verifier simulator is more complicated if we use 1-out-of-2 binding schemes. Hence, we use the stronger scheme of the above fact in our proof.

Finally, we need the following statement which follows by modifying the arguments of Håstad, Impagliazzo, Levin and Luby [HILL99], and Naor [Nao91].

Lemma 3. *Let $I \subseteq J \subseteq \{0,1\}^*$. Suppose D_x, $x \in J$ is a $\mathbf{P}$-sampleable family of probability distributions on $\{0,1\}^{m(x)}$. Also, suppose D_x, $x \in I$ is a quantumly secure false entropy generator. Then there is a classical constant round public coin instance-dependent bit commitment scheme for all $x \in J$ which is exponentially binding statistically for all $x \in J$ and quantum computationally hiding for all $x \in I$.*

Proof. **(Sketch)** Same reasoning as in the proof of Lemma 2. □

By combining Lemmas 2 and 3, and Fact 2, and using the techniques of Vadhan [Vad06], we can conclude the following quantum analogue of results of Ong and Vadhan [OV08].

Lemma 4. *Every promise problem in* $\mathbf{HVCZK_Q}$ *gives rise to a classical constant round public coin instance dependent bit commitment scheme that is quantum computationally hiding on the positive instances and exponentially binding statistically on the negative instances.*

We are now finally in a position to show that every problem in $\mathbf{HVCZK_Q}$ has a classical public coin interactive protocol with a stage-by-stage honest verifier simulator. For the classical counterparts of the proposition below, we refer the reader to Ong and Vadhan [OV08].

Proposition 1. *Every promise problem* $\Pi = (\Pi_Y, \Pi_N)$ *in* $\mathbf{HVCZK_Q}$ *has a classical public coin interactive protocol with perfect completeness, exponentially small soundness and a stage-by-stage honest-verifier simulator that produces simulated transcripts that are negligibly quantum computationally distinguishable from the actual prover honest-verifier interaction transcripts. Furthermore if* $\Pi \in \mathbf{HVSZK}$*, then the resulting protocol is constant round and the simulated transcripts are exponentially close in total variation distance from the actual transcripts.*

A proof sketch can be found in the appendix.

Combining Lemma 1 together with Proposition 1, we prove the main theorem of the paper.

Theorem 1. $\mathbf{HVCZK_Q} \subseteq \mathbf{CZKQ}$ *and* $\mathbf{HVSZK} \subseteq \mathbf{SZKQ}$.

Acknowledgments

We are grateful to an anonymous referee of an earlier version of this paper for detecting a subtle bug in that version, and also for informing us about the recent work of Ong and Vadhan [OV08] on zero-knowledge. We thank Shien Jin Ong and Salil Vadhan for clarifying many doubts about instance-dependent bit commitments and zero-knowledge. We also thank the anonymous referees of this version for helpful comments. S.Z. thanks Iordanis Kerenidis, Manoj Prabhakaran, Ben Reichardt, Amit Sahai, Yaoyun Shi, Robert Spalek, Shanghua Teng, Umesh Vazirani and Andy Yao for listening to the progress of the work, clarifying things and giving interesting comments. P.S. thanks Jaikumar Radhakrishnan and Thomas Vidick for helpful feedback. All authors thank Wei Huang and Martin Rötteler for discussions at an early stage of the work.

References

[BGG+90] Ben-Or, M., Goldreich, O., Goldwasser, S., Håstad, J., Kilian, J., Micali, S., Rogaway, P.: Every provable is provable in zero-knowledge. In: Goldwasser, S. (ed.) CRYPTO 1988. LNCS, vol. 403, pp. 37–56. Springer, Heidelberg (1990)

[DGW94] Damgård, I., Goldreich, O., Wigderson, A.: Hashing functions can simplify zero-knowledge protocol design (too). Technical Report RS-94-39, BRICS (1994)

[FFS88] Feige, U., Fiat, A., Shamir, A.: Zero-knowledge proofs of identity. Journal of Cryptology 1(2), 77–94 (1988)

[GMW91] Goldreich, O., Micali, S., Widgerson, A.: Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. Journal of the ACM 38(1), 691–729 (1991)

[Gol01] Goldreich, O.: Foundations of cryptography, vol. 1. Cambridge University Press, Cambridge (2001)

[GS89] Goldwasser, S., Sipser, M.: Private coins versus public coins in interactive proof systems. Advances in Computing Research, vol. 5, pp. 73–90. JAC Press, Inc. (1989)

[GSV98] Goldreich, O., Sahai, A., Vadhan, S.: Honest-verifier statistical zero-knowledge equals general statistical zero-knowledge. In: Proceedings of the 30th Annual ACM Symposium on Theory of Computing, pp. 399–408 (1998)

[GV97] Goldreich, O., Vadhan, S.: Comparing entropies in statistical zero knowledge with applications to the structure of SZK. In: Proceedings of the 14th Annual IEEE Symposium on Foundations of Computer Science, pp. 448–457 (1997)

[HILL99] Håstad, J., Impagliazzo, R., Levin, L., Luby, M.: A pseudorandom generator from any one-way function. SIAM Journal on Computing 28(4), 1364–1396 (1999)

[IY88] Impagliazzo, R., Yung, M.: Direct zero-knowledge computations. In: Pomerance, C. (ed.) CRYPTO 1987. LNCS, vol. 293, pp. 40–51. Springer, Heidelberg (1988)

[Kob08] Kobayashi, H.: General properties of quantum zero-knowledge proofs. In: Proceedings of the 5th Theory of Cryptography Conference, pp. 107–124 (2008), Also quant-ph/0705.1129

[Nao91] Naor, M.: Bit commitment using pseudorandom generator. Journal of Cryptology 4, 151–158 (1991)

[NV06] Nguyen, M.-H., Vadhan, S.: Zero knowledge with efficient provers. In: Proceedings of the 38th Annual ACM Symposium on Theory of Computing, pp. 287–295 (2006)

[OV08] Ong, S., Vadhan, S.: An equivalence between zero knowledge and commitments. In: Proceedings of the 5th Theory of Cryptography Conference (to appear, 2008)

[SV03] Sahai, A., Vadhan, S.: A complete promise problem for statistical zero-knowledge. Journal of the ACM 50(2), 196–249 (2003)

[Vad06] Vadhan, S.: An unconditional study of computational zero knowledge. SIAM Journal on Computing 36(4), 1160–1214 (2006)

[Wat02] Watrous, J.: Limits on the power of quantum statistical zero-knowledge. In: Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science, pp. 459–468 (2002)

[Wat06] Watrous, J.: Zero-knowledge against quantum attacks. In: Proceedings of the 38th Annual ACM Symposium on Theory of Computing, pp. 296–305 (2006)