Making Classical Honest Verifier Zero Knowledge Protocols Secure Against Quantum Attacks

Sean Hallgren
Pennsylvania State University
University Park, PA, U.S.A.
hallgren@cse.psu.edu

Alexandra Kolla
U C Berkeley
Berkeley, CA, U.S.A.
akolla@cs.berkeley.edu

Pranab Sen
Tata Institute of Fundamental Research
Mumbai, India.
pgdsen@tcs.tifr.res.in

Shengyu Zhang
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, possibly cheating classical and quantum polynomial time verifiers. 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

One of the main impacts of quantum computation thus far has been its potential implications for cryptography. Public key cryptography, a central concept in cryptography, is used to protect web transactions, and its security relies on the hardness of certain number theory problems. Exponential speedups by quantum computers have been found for problems such as factoring and discrete log [Sho97], Pell's equation [Hal02], and for finding the unit group and class group of a number field [Hal05, SV05]. This implies that a quantum computer could break RSA and Diffie-Hellman, which are currently used, as well as potentially more secure systems such as the Buchmann-Williams key-exchange protocol [SBW94]. Understanding which cryptosystems are secure against quantum computers is one of the fundamental questions in the field.

Another central concept in cryptography is that of a zero-knowledge protocol. 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 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.

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 both of 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 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 there are 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 shows that every problem in $\mathbf{SZK}$ has a *quantum* protocol that is statistical zero-knowledge against any cheating non-uniform polynomial time quantum verifier. However, it is preferable that the 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 results can be stated as:

Theorem 1 (Main). 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 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].

By slightly abusing terminology, here and in what follows we interchangeably use the terms 'efficient machine', 'verifier', to refer to non-uniform polynomial time machines, either classical or quantum.

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 $\mathbf{HVZK}$ protocol into another public-coin protocol $\mathbf{ZK}$ 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 main crucial observation is that if the honest-verifier simulator for the original classical public coin $\mathbf{ZK}$ protocol uses its internal randomness in a 'stage-by-stage' fashion, 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. The main reason is that the classical or quantum simulator for the new protocol can rewind the verifier polynomial number of times within each round, 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 round is inverse polynomial, polynomially many rewinding steps will result in a successful simulation of the current round 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 final 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 round-by-round 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 Haitner and Reingold [HR]and Ong and Vadhan [OV] 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. Haitner et al. [OV] 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 proof system by $(P, V) : (\alpha_1, \beta_1, ..., \alpha_N, \beta_N)$, 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 $y \in \{0,1\}^t$, $1 \leq |f^{-1}(y)| \leq (s-1)2^{s-t}+1$. 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 the DGW transformation 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 property which we will crucially use [DGW94].

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

For the completeness and zero-knowledge property of DGW, the reader is referred to [DGW94].

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 $r = r_1 \circ \cdots \circ r_N$ such that in each stage i, the simulated messages $(\hat{\alpha}_{i1}, \hat{\beta}_{i1}, \ldots, \hat{\alpha}_{ic}, \hat{\beta}_{ic})$ are functions of $r_1, \ldots, r_i$ and the input alone, and for every fixed $r_1, \ldots, r_{i-1}$, for every $j \in [c]$, for every fixed prefix $(\bar{\alpha}_{i1}, \bar{\beta}_{i1}, \ldots, \bar{\alpha}_{i,j-1}, \bar{\beta}_{i,j-1})$ of the simulated transcript, the distribution of $\hat{\alpha}_{ij}$ as r_i varies is uniform.*

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 [OV] which will be used in our construction of interactive protocols with honest-verifier stage-by-stage simulators.

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, possibly malicious, 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 (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 $\epsilon(|x|)$. In addition, the scheme Com_x is said to be* exponentially hiding statistically *for all $x \in \Pi_Y$ if the views of 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.*

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.

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, this allows us to rewind a stage polynomially many times without disturbing previous stages and ensure a successful simulation with very high probability. □

A formal proof of the classical and quantum parts of the above lemma theorem 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 $I \subseteq \{0,1\}^*$. 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 a promise problem* $\Pi = (\Pi_Y, \Pi_N) \in \mathbf{HVCZK_Q}$. *Then for every* $x \in \Pi_Y \cup \Pi_N$, *there is a* $\mathbf{P}$*-sampleable probability distribution* D_x *on* $\{0,1\}^{m(|x|)}$, *and a subset* $I \subseteq \Pi_Y$ *such that* $\{D_x\}_{x \in I}$ *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. □

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 Haitner and Reingold [HR] and Ong and Vadhan [OV].

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 comment 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 [OV].

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 [OV].

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. Moreover, if the problem is in* $\mathbf{HVSZK}$ *then the resulting protocol is constant round and the simulated transcripts are exponential close in total variation distance from the actual transcripts.*

Proof. We observe that the standard procedure for converting a classical interactive protocol $\mathcal{P}$ into a zero-knowledge protocol $\mathcal{P}'$ using bit commitments [IY88, BGG$^+$90] gives rise to a protocol with a stage-by-stage honest verifier simulator. Here, we use the instance-dependent bit commitments guaranteed by Lemma 4.

By the results of Furer et al. [FGM$^+$89], and Goldwasser and Sipser [GS89], $\mathcal{P}$ can be assumed to be public coin with perfect completeness and exponentially small soundness error.

We now briefly sketch the proof for the $\mathbf{HVSZK}$ case. For languages in $\mathbf{SZK}$ the guaranteed instance dependent commitments are constant-round, statistically hiding on the yes instances and statistically binding on no instances. Thus we can obtain a constant-round, public coin, honest verifier SZK protocol for any $\mathbf{SZK}$ language with soundness inverse polynomial. The idea is roughly the following: $\mathbf{SZK}$ is contained in $\mathbf{AM}$, so the prover commits to the message he would have sent in the AM protocol. Then he proves in zero-knowledge that the verifier would have accepted the prover's message. Since we only require zero-knowledge against the honest verifier, we can now repeat the protocol in parallel to get exponentially small soundness error and at the same time preserve the number of rounds.

We next present a more formal proof for the $\mathbf{HVCZK}$ case. Let x be the input to the problem Π. In $\mathcal{P}'$, the prover commits to his messages that he would have sent in $\mathcal{P}$. For this, he runs copies of the bit commitment protocol for x in parallel for each bit of a message of the prover of $\mathcal{P}$. After the commitment of each message of the prover of $\mathcal{P}$, the verifier in $\mathcal{P}'$ just sends a uniformly random message independent of the message transcript so far, as in $\mathcal{P}$. After the commitment for the last round of $\mathcal{P}$ is over, the decision of whether the verifier of $\mathcal{P}'$ accepts or rejects is an $\mathbf{NP}$-predicate determined by the input, prover's commitments and the verifier's public coins. The prover of $\mathcal{P}'$ then attempts to convince the verifier of $\mathcal{P}'$ via the graph 3-coloring zero-knowledge protocol of Goldreich, Micali and Wigderson [GMW91] to accept the input x. Again, several copies of the bit commitment protocol for x are run in parallel in order to keep the number of rounds of the 3-coloring protocol constant. As originally described in [GMW91], the graph 3-coloring protocol is a 3-message protocol with one message from the verifier in which he sends along a random edge of the graph. Since the graph in our application is implicitly defined by the earlier messages of $\mathcal{P}'$ and the input, it is a variable quantity and this prevents the 3-coloring protocol from being public coin. However, this shortcoming can be remedied easily in a manner akin to making the DGW transformation public coin. If m is the number of vertices in the implicit graph, which can be assumed to be the same irrespective of the input or the prior messages exchanged, the verifier just sends a random number between 1 and $\binom{m}{2}!$ which the prover then interprets appropriately depending on the actual graph defined by the input and prior messages of $\mathcal{P}'$. Since for any x, the bit commitment protocols are classical and public coin, $\mathcal{P}'$ is a classical public coin protocol too.

Since for $x \in \Pi_N$, the bit commitment protocols for x have exponentially small binding error for any prover, the soundness error of $\mathcal{P}'$ is bounded away from one by an inverse polynomial. Also, $\mathcal{P}'$ has perfect completeness since the bit commitment protocols have perfect completeness for any x. We now discuss the honest-verifier zero-knowledge property of $\mathcal{P}'$. The honest-verifier simulator for $\mathcal{P}'$ just commits to the all-zeroes string from the prover's side in every round, and at the end, simulates the graph 3-coloring protocol of [GMW91] in a standard fashion. For $x \in \Pi_Y$, the bit commitment protocols for x are quantum computationally hiding, which proves that the simulated transcript for $\mathcal{P}'$ produced by the honest verifier simulator is quantum computationally indistinguishable from the actual transcript of $\mathcal{P}'$ obtained via the prover honest-verifier interaction. Since for any x, the bit commitment protocols are constant round, $\mathcal{P}'$ has a stage-by-stage honest verifier simulator where each stage is either a transformation of a round of $\mathcal{P}$ or the final zero knowledge protocol for 3-coloring transformed by bit commitments for x. In either case, each stage of $\mathcal{P}'$ consists of a constant number of rounds and the number of stages is one more than the number of rounds of $\mathcal{P}$. In fact in the simulated transcript of $\mathcal{P}'$, the probability distributions of the stages corresponding to transformed rounds of $\mathcal{P}$ are independent of each other. The distribution of the last stage of $\mathcal{P}'$ does depend on the previous stages, but nevertheless satisfies the conditions of the definition of stage-by-stage simulation (Definition 1).

Parallely repeating $\mathcal{P}'$ polynomially many times gives a new protocol $\mathcal{P}''$ with exponentially small soundness error and other properties as claimed in the statement of the proposition.

□

Combining Lemma 1 together with Proposition 1, we prove the main theorem of the paper.
Theorem 1:$\mathbf{HVCZK_Q} = \mathbf{CZKQ}$ and $\mathbf{HVSZK} = \mathbf{SZKQ}$.

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 Haitner, Ong, Reingold and Vadhan on zero-knowledge. We thank Shien Jin Ong and Salil Vadhan for clarifying many doubts about instance-dependent bit commitments and zero-knowledge. 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] M. Ben-Or, O. Goldreich, S. Goldwasser, J. Håstad, J. Kilian, S. Micali, and P. Rogaway. Every provable is provable in zero-knowledge. In *Proceedings of Crypto88*, Lecture Notes in Computer Science, vol. 403, pages 37–56. Springer-Verlag, 1990.

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

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

[FGM$^+$89] M. Furer, O. Goldreich, Y. Mansour, M. Sipser, and S. Zachos. *On completeness and soundness in interactive proof systems*, volume 5, pages 429–442. JAC Press, Inc., 1989.

[GMW91] O. Goldreich, S. Micali, and A. Widgerson. 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] O. Goldreich. *Foundations of cryptography*, volume 1. Cambridge University Press, 2001.

[GS89] S. Goldwasser and M. Sipser. *Private coins versus public coins in interactive proof systems*, volume 5 of *Advances in Computing Research*, pages 73–90. JAC Press, Inc., 1989.

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

[GV97] O. Goldreich and S. Vadhan. 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*, pages 448–457, 1997.

[Hal02] S. Hallgren. Polynomial-time quantum algorithms for Pell's equation and the principal ideal problem. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 653–658, 2002.

[Hal05] S. Hallgren. Fast quantum algorithms for computing the unit group and class group of a number field. In *Proceedings of the 37th Anuual ACM Symposium on Theory of Computing*, pages 468–474, 2005.

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

[HR] I. Haitner and O Reingold. A new interactive hashing theorem. In *Technical Report TR06-096, Electronic colloquioum on Computational Complexity.*

[IY88] R. Impagliazzo and M. Yung. Direct zero-knowledge computations. In *Proceedings of Crypto87*, Lecture Notes in Computer Science, vol. 293, pages 40–51. Springer-Verlag, 1988.

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

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

[OV] S. Ong and S Vadhan. An equivalence between zero knowledge and commitments. In *TCC 2008, to appear*.

[SBW94] R. Scheidler, J. Buchmann, and H. Williams. A key-exchange protocol using real quadratic fields. *Journal of Cryptology*, 7(3):171–199, 1994.

[Sho97] P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

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

[SV05] A. Schmidt and U. Vollmer. Polynomial time quantum algorithm for the computation of the unit group of a number field. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 475–480, 2005.

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

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

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

Appendix

A Proof for the quantum case of Lemma 1

In this section we will prove Lemma 1 for the quantum verifier. The case of classical verifier can be proved in a similar fashion, as shown in the next section. Some notations are as follows: Unless otherwise stated, the variable ij ranges over $i \in [N], j \in [c]$ (refer to Definition 1). Without loss of generality, by padding, we assume that each r_i is of the same length m and each α_{ij} is of the same length s. Denote the original protocol by (P_1, V_1), and the new protocol obtained by applying DGW to (P_1, V_1) by (P_2, V_2). The honest-verifier simulator for (P_1, V_1), M_1, is stage-by-stage.

A.1 Quantum simulator: construction

Let $x \in \Pi_Y$ be the input to the problem. All operations taking place in the protocols as well as simulators below depend on x, but for simplicity, we shall omit this dependence on x in the notation. Suppose $|\psi\rangle$ is the auxiliary input to the cheating polynomial time quantum verifier V_2' in the new protocol. We can assume that the auxiliary input is a pure state since otherwise we can take a purification of the auxiliary input as we are consider a cheating verifier V_2', and this cannot decrease the distinguishability between the simulated and actual views of the verifier. Let V_2' use quantum register W as its work space and classical registers $\{F_{ij}, Y_{ij}, A_{ij}, B_{ij}\}_{i\in[N],j\in[c]}$ as its message registers. Initially $|\psi\rangle$ is held in register Aux, which is a part of W. At round ij, that is, the j-th round in stage i, suppose the message transcript so far is hist_{ij}; then V_2' performs on W a unitary transformation $U_{ij,\mathrm{hist}_{ij}}$ followed by a measurement whose outcome f_{ij} is saved in register F_{ij}. Then V_2' sends this f_{ij} to P_2, who chooses a random $y_{ij} \in \{0,1\}^t$, which is saved by V_2' in the register Y_{ij}. Then V_2' performs another unitary transformation $V_{ij,\mathrm{hist}_{ij},f_{ij},y_{ij}}$ followed by another measurement whose outcome α_{ij} is saved in register A_{ij} and sent to P_2, who replies with β_{ij} and finishes the DGW transformation of the round ij. We require that $\alpha_{ij} \in f_{ij}^{-1}(y_{ij})$ without loss of generality.

Let $M_{ij,\mathrm{hist}_{ij},f_{ij}}$ be the operation $U_{ij,\mathrm{hist}_{ij}}$ followed by the projection on the subspace with the outcome f_{ij}, so $\sum_{f_{ij}} M_{ij,\mathrm{hist}_{ij},f_{ij}} = U_{ij,\mathrm{hist}_{ij}}$. Similarly, let $N_{ij,\mathrm{hist}_{ij},f_{ij},y_{ij},\alpha_{ij}}$ be the operation $V_{ij,\mathrm{hist}_{ij},f_{ij},y_{ij}}$ followed by the projection on the subspace with the outcome α_{ij}.

We will define a simulator M_2', and then show the zero-knowledge property of (P_2, V_2') by showing that $(P_2, V_2')(|\psi\rangle\langle\psi|)$ is negligibly distinguishable from $M_2'(|\psi\rangle\langle\psi|)$, where the above notation denotes the view of the verifier V_2' in the actual interaction (P_2, V_2') and the simulated interaction M_2' respectively, with the pure state $|\psi\rangle$ as auxiliary input.

We will first write down $(P_2, V_2')(|\psi\rangle\langle\psi|)$. The probability distribution of β_{ij} in (P_2, V_2') depends only on the previous $\alpha_{i'j'}$s and is the same as in (P_1, V_1). We shall use $\mathbf{Pr}_{P_1}[(\beta_{ij})_{ij} \mid (\alpha_{ij})_{ij}]$ to denote the probability of the sequence of β_{ij}s conditioned on a particular sequence of α_{ij}s. It is easy to see that

$$(P_2, V_2')(|\psi\rangle\langle\psi|) = 2^{-Nct} \sum_{(f_{ij},y_{ij},\alpha_{ij},\beta_{ij})_{ij}} \mathbf{Pr}_{P_1}[(\beta_{ij})_{ij} \mid (\alpha_{ij})_{ij}] \Big(K_{(f_{ij},y_{ij},\alpha_{ij},\beta_{ij})_{ij}} |\psi\rangle\langle\psi| K^{\dagger}_{(f_{ij},y_{ij},\alpha_{ij},\beta_{ij})_{ij}} \big(|f_{ij}\rangle\langle f_{ij}| \otimes |y_{ij}\rangle\langle y_{ij}| \otimes |\alpha_{ij}\rangle\langle\alpha_{ij}| \otimes |\beta_{ij}\rangle\langle\beta_{ij}|\big)_{ij} \Big)$$

where the α_{ij} ranges over $f_{ij}^{-1}(y_{ij})$, and

$$K_{(f_{ij},y_{ij},\alpha_{ij},\beta_{ij})_{ij}} := N_{Nc,\mathrm{hist}_{Nc},f_{Nc},y_{Nc},\alpha_{Nc}} M_{Nc,\mathrm{hist}_{Nc},f_{Nc}} \cdots N_{11,f_{11},y_{11},\alpha_{11}} M_{11,f_{11}}$$

is the operation $M_{11,f_{11}}$ followed by $N_{11,f_{11},y_{11},\alpha_{11}}$, then followed by $M_{12,(f_{11},y_{11},\alpha_{11},\beta_{11}),f_{12}}$ and followed by $N_{12,(f_{11},y_{11},\alpha_{11},\beta_{11}),f_{12},y_{12},\alpha_{12}}$ and so on, until finally finishing the whole transcript $(f_{ij}, y_{ij}, \alpha_{ij}, \beta_{ij})_{ij}$.

The quantum simulator M_2' is defined in the following way. The work space of M_2' consists of register W, which contains Aux, and registers $\{F_{ij}, Y_{ij}, A_{ij}, B_{ij}\}_{ij}$, as well as new registers $\{R_i\}_i$, $\{A'_{ij}, F'_{ij}\}_{ij}$ for other purposes like randomness generation and backups. All the registers of M_2' described above are quantum. The precise algorithm specifying M_2' is as follows. Since V_2' is a polynomial time machine, it is easy to see that M_2' runs in polynomial time.

Quantum simulator M_2': Input x and auxiliary input $|\psi\rangle$

```
1. Initialize all registers with the all zero state $|0\rangle$ except Aux which contains $|\psi\rangle$;
2. for $i = 1$ to $N$
   (a) Repeat the following loop $N \cdot 2^{c(s-t)}$ times:
       i. Generate a uniform superposition $2^{-m/2} \sum_{r_i} |r_i\rangle$ in register $R_i$;
       ii. Conditioned on $R_1, \ldots, R_i$'s content being $r_1, \ldots, r_i$, run $M_1$ to get $(\hat{\alpha}_{i1}, \hat{\beta}_{i1}, \ldots, \hat{\alpha}_{ic}, \hat{\beta}_{ic})$ and save them in the registers $(A'_{i1}, B_{i1}, \ldots, A'_{ic}, B_{ic})$ respectively;
       iii. for $j = 1$ to $c$
               // Simulate the DGW transformation
           A. Conditioned on the message registers' content so far being $\mathrm{hist}_{ij}$, apply $U_{ij,\mathrm{hist}_{ij}}$ to $W$. Then conditioned on the current state in $W$ being in the subspace corresponding to $f_{ij}$, write $f_{ij}$ in the register $F_{ij}$ and $F'_{ij}$;
           B. Conditioned on $F_{ij}$ containing $f_{ij}$ and $A'_{ij}$ containing $\hat{\alpha}_{ij}$, compute $y_{ij} = f_{ij}(\hat{\alpha}_{ij})$ and write the result in $Y_{ij}$;
           C. Conditioned on the content of the message registers' so far being $(\mathrm{hist}_{ij}, f_{ij}, y_{ij})$, apply $V_{ij,\mathrm{hist}_{ij},f_{ij},y_{ij}}$ to $W$. Then conditioned on the current state in $W$ being in the subspace corresponding to $\alpha'_{ij}$, write $\alpha'_{ij}$ in the register $A_{ij}$;
           // Check whether simulation for stage i succeeded and rewind if not
       iv. If $\hat{\alpha}_{ij} \neq \alpha'_{ij}$ for some $j \in [c]$, do the following:
           A. Apply $U^\dagger$, where $U$ is the unitary transformation corresponding to the execution from Step 2(a)(i) till just before Step 2(a)(iv);
           B. Reflect about the subspace
                  $$W \otimes (\otimes_{i'<i,j\in[c]}(F_{i'j}Y_{i'j}A_{i'j}B_{i'j}A'_{i'j}F'_{i'j}R_{i'})) \otimes |0\rangle$$
              where $|0\rangle$ is the all zero vector in $\otimes_{i'\geq i,j\in[c]}(F_{i'j}Y_{i'j}A_{i'j}B_{i'j}A'_{i'j}F'_{i'j}R_{i'})$, that is, the remaining registers of $V_2'$;
           C. Go to Step 2(a)(i);
       v. Else, proceed to stage $i+1$ in Step 2 since $\hat{\alpha}_{ij} = \alpha'_{ij}$ for all $j \in [c]$ and the simulation of stage $i$ succeeded;
   (b) Output Fail and terminate the whole simulation, as we have failed to simulate stage $i$ successfully;
3. Trace out $A'_{ij}, F'_{ij}, R_i$ for all $ij$s and output the remaining registers;
```

A.2 Quantum simulator: analysis

We now formally analyze the behavior of M_2'. Suppose in Step 2(a)(iv) of M_2' instead of conditional rewinding, we do a real measurement to see whether $\hat{\alpha}_{ij} = \alpha'_{ij}$ for all $j \in [c]$. Call this new machine conditioned on all "equality" tests succeeding as M_2''. We use $M_2''(|\psi\rangle\langle\psi|)$ to denote the verifier's view outputted by M_2'' on auxiliary input $|\psi\rangle$. We will show that $M_2'(|\psi\rangle\langle\psi|)$ is exponentially close to $M_2''(|\psi\rangle\langle\psi|)$ in trace distance, and $(P_2, V_2')(|\psi\rangle\langle\psi|)$ is information theoretically or quantum computationally close to $M_2''(|\psi\rangle\langle\psi|)$ as appropriate.

We start by proving the following lemma.

Lemma 5. *Fix coins $r_1, \ldots, r_{i-1}$ of M_2''. Let $j \in [c]$. Fix a prefix $(\bar{f}_{i'j'}, \bar{y}_{i'j'}, \bar{\alpha}_{i'j'}, \bar{\beta}_{i'j'})_{i'j'<ij} \circ \bar{f}_{ij}$ of the simulated transcript created by M_2''; the coin r_i of M_2'' is allowed to vary conditioned on this prefix. Then,*

1. $\mathbf{Pr}_{M_2''}[\hat{\alpha}_{ij} = \alpha'_{ij}] = 2^{t-s}$;
2. *For any $\bar{y}_{ij} \in \{0,1\}^t$,* $\mathbf{Pr}_{M_2''}[\bar{f}_{ij}(\hat{\alpha}_{ij}) = \bar{y}_{ij} \mid \hat{\alpha}_{ij} = \alpha'_{ij}] = 2^{-t}$;
3. *For any $\bar{y}_{ij} \in \{0,1\}^t$, $\bar{\alpha}_{ij} \in \bar{f}_{ij}^{-1}(\bar{y}_{ij})$,*

$$\mathbf{Pr}_{M_2''}[\hat{\alpha}_{ij} = \bar{\alpha}_{ij} \mid \bar{f}_{ij}(\hat{\alpha}_{ij}) = \bar{y}_{ij}, \hat{\alpha}_{ij} = \alpha'_{ij}] = \mathbf{Pr}_{V_2'}[\alpha'_{ij} = \bar{\alpha}_{ij} \mid y_{ij}].$$

Proof. We use f, $\hat{\alpha}$, α' and y as shorthands for $\bar{f}_{ij}$, $\hat{\alpha}_{ij}$, α'_{ij} and $\bar{y}_{ij}$ respectively. Then,

$$\begin{aligned}\mathbf{Pr}_{M_2''}[\alpha' = \hat{\alpha}] &= \sum_y \mathbf{Pr}_{M_2''}[f(\hat{\alpha}) = y]\mathbf{Pr}_{M_2''}[\alpha' = \hat{\alpha} | f(\hat{\alpha}) = y] \\ &= \sum_y \mathbf{Pr}_{M_2''}[f(\hat{\alpha}) = y] \sum_{\bar{\alpha} \in f^{(-1)}(y)} \mathbf{Pr}_{M_2''}[\alpha' = \hat{\alpha} = \bar{\alpha} | f(\hat{\alpha}) = y],\end{aligned}$$

where above and below, summation over y means summation over all $y \in \{0,1\}^t$. Note that for any fixed transcript of the simulated messages so far and f, the simulation M_2'' has the property that α' only depends on y and not on which $\hat{\alpha} \in f^{(-1)}(y)$. Therefore, conditioned on $f(\hat{\alpha}) = y$, the events $\alpha' = \bar{\alpha}$ and $\hat{\alpha} = \bar{\alpha}$ are independent. So

$$\begin{aligned}&\mathbf{Pr}_{M_2''}[\alpha' = \hat{\alpha}] \\ &= \sum_y \mathbf{Pr}_{M_2''}[f(\hat{\alpha}) = y] \sum_{\bar{\alpha} \in f^{(-1)}(y)} \mathbf{Pr}_{M_2''}[\alpha' = \bar{\alpha} | f(\hat{\alpha}) = y] \cdot \mathbf{Pr}_{M_2''}[\hat{\alpha} = \bar{\alpha} | f(\hat{\alpha}) = y] \\ &= \sum_y \frac{\mathbf{Pr}_{M_2''}[f(\hat{\alpha}) = y]}{|f^{-1}(y)|} \sum_{\bar{\alpha} \in f^{(-1)}(y)} \mathbf{Pr}_{M_2''}[\alpha' = \bar{\alpha} | f(\hat{\alpha}) = y] \\ &= \sum_y \frac{2^{-s}|f^{-1}(y)|}{|f^{-1}(y)|} = 2^{t-s}.\end{aligned}$$

Fix $y = \bar{y}_{ij}$. Then,

$$\begin{aligned}\mathbf{Pr}_{M_2''}[f(\hat{\alpha}) = y \mid \hat{\alpha} = \alpha'] &= \frac{\mathbf{Pr}_{M_2''}[f(\hat{\alpha}) = y] \cdot \mathbf{Pr}_{M_2''}[\hat{\alpha} = \alpha' \mid f(\hat{\alpha}) = y]}{\mathbf{Pr}_{M_2''}[\hat{\alpha} = \alpha']} \\ &= \frac{2^{-s}|f^{-1}(y)|}{2^{t-s}|f^{-1}(y)|} = 2^{-t},\end{aligned}$$

where we use the earlier calculation in the second inequality.

Let $\bar{\alpha} := \bar{\alpha}_{ij}$. Similarly,

$$\begin{aligned}
\mathbf{Pr}_{M_2''}[\hat{\alpha} = \bar{\alpha} \mid f(\hat{\alpha}) = y, \hat{\alpha} = \alpha'] &= \frac{\mathbf{Pr}_{M_2''}[\alpha' = \hat{\alpha} = \bar{\alpha}]}{\mathbf{Pr}_{M_2''}[f(\hat{\alpha}) = y] \cdot \mathbf{Pr}_{M_2''}[\hat{\alpha} = \alpha' \mid f(\hat{\alpha}) = y]} \\
&= \frac{\mathbf{Pr}_{M_2''}[\hat{\alpha} = \bar{\alpha}] \cdot \mathbf{Pr}_{M_2''}[\alpha' = \bar{\alpha} \mid \hat{\alpha} = \bar{\alpha}]}{2^{-s}|f^{-1}(y)||f^{-1}(y)|^{-1}} \\
&= \frac{2^{-s} \cdot \mathbf{Pr}_{V_2'}[\alpha' = \bar{\alpha} \mid y]}{2^{-s}} \\
&= \mathbf{Pr}_{V_2'}[\alpha' = \bar{\alpha} \mid y].
\end{aligned}$$

This completes the proof of the lemma. □

Now we are ready to show the closeness of $(P_2, V_2')(|\psi\rangle\langle\psi|)$ and $M_2''(|\psi\rangle\langle\psi|)$.

Lemma 6. *$(P_2, V_2')(|\psi\rangle\langle\psi|)$ is quantum computationally indistinguishable from $M_2''(|\psi\rangle\langle\psi|)$ if M_1 outputs simulated transcripts that are quantum computationally indistinguishable from the actual interaction in (P_1, V_1). If M_1 outputs simulated transcripts that are exponentially close in total variation distance to the actual interaction transcripts in (P_1, V_1), then $(P_2, V_2')(|\psi\rangle\langle\psi|)$ is exponentially close in trace distance to $M_2''(|\psi\rangle\langle\psi|)$.*

Proof. From Lemma 5, it is easy to see that

$$\begin{aligned}
M_2''(|\psi\rangle\langle\psi|) = 2^{-Nct} \sum_{(f_{ij}, y_{ij}, \alpha_{ij}, \beta_{ij})_{ij}} & \mathbf{Pr}_{M_1}[(\beta_{ij})_{ij} \mid (\alpha_{ij})_{ij}] \\
& \Big(K_{(f_{ij}, y_{ij}, \alpha_{ij}, \beta_{ij})_{ij}} |\psi\rangle\langle\psi| K^\dagger_{(f_{ij}, y_{ij}, \alpha_{ij}, \beta_{ij})_{ij}} \\
& \quad (|f_{ij}\rangle\langle f_{ij}| \otimes |y_{ij}\rangle\langle y_{ij}| \otimes |\alpha_{ij}\rangle\langle\alpha_{ij}| \otimes |\beta_{ij}\rangle\langle\beta_{ij}|)_{ij} \Big),
\end{aligned}$$

where the α_{ij} ranges over $f_{ij}^{-1}(y_{ij})$. Let the total variation distance between the simulated transcripts of M_1 and the actual interaction transcripts of (P_1, V_1) be at most δ. Then,

$$\begin{aligned}
& \|(P_2, V_2')(|\psi\rangle\langle\psi|) - M_2''(|\psi\rangle\langle\psi|)\|_{\mathrm{tr}} \\
&= 2^{-Nct} \sum_{(f_{ij}, y_{ij}, \alpha_{ij}, \beta_{ij})_{ij}} |\mathbf{Pr}_{M_1}[(\beta_{ij})_{ij} \mid (\alpha_{ij})_{ij}] - \mathbf{Pr}_{P_1}[(\beta_{ij})_{ij} \mid (\alpha_{ij})_{ij}]| \|K_{(f_{ij}, y_{ij}, \alpha_{ij}, \beta_{ij})_{ij}} |\psi\rangle\|^2 \\
&\leq \sum_{(\beta_{ij})_{ij}} |\mathbf{Pr}_{M_1}[(\beta_{ij})_{ij} \mid (\alpha_{ij})_{ij}] - \mathbf{Pr}_{P_1}[(\beta_{ij})_{ij} \mid (\alpha_{ij})_{ij}]| \leq \delta,
\end{aligned}$$

where the α_{ij} ranges over $f_{ij}^{-1}(y_{ij})$. Thus, if δ is exponentially small, we see that $M_2''(|\psi\rangle\langle\psi|)$ is statistically close to $(P_2, V_2')(|\psi\rangle\langle\psi|)$. Also, observe that $M_2''(|\psi\rangle\langle\psi|)$ can be obtained by an efficient quantum circuit from the simulated transcript outputted by M_1, and $(P_2, V_2')(|\psi\rangle\langle\psi|)$ can be obtained by the same efficient quantum circuit from the actual (P_1, V_1) interaction transcript. Thus quantum computationally indistinguishability of the simulated transcripts of M_1 versus the actual interaction transcripts of (P_1, V_1) translates into the quantum computational indistinguishability of $M_2''(|\psi\rangle\langle\psi|)$ and $(P_2, V_2')(|\psi\rangle\langle\psi|)$.

This completes the proof of the lemma. □

In order to show that $M_2'(|\psi\rangle\langle\psi|)$ is exponentially close to $M_2''(|\psi\rangle\langle\psi|)$ in trace distance, we need the following geometric fact implicitly used by Watrous [Wat06].

Fact 3. *Suppose V_1 and V_2 are two subspaces in a Hilbert space H. Let P_{V_1} and P_{V_2} be the projectors onto V_1 and V_2 respectively. Let $0 < p < 1$. Suppose for any unit vector $|\psi_1\rangle$ in V_1, it holds that $\|P_{V_2}|\psi_1\rangle\|^2 = p$ independent of $|\psi_1\rangle$. Let R_{V_1}, $R_{|\psi_1\rangle}$ denote reflection about the subspaces V_1 and $|\psi_1\rangle$ respectively. Then, $R_{|\psi_1\rangle}P_{V_2}|\psi_1\rangle = R_{V_1}P_{V_2}|\psi_1\rangle$. That is, the reflection of $P_{V_2}|\psi_1\rangle$ about the vector $|\psi_1\rangle$ is the same as the reflection of $P_{V_2}|\psi_1\rangle$ about the subspace V_1.*

Proof. Since $P_{|\psi_1\rangle} = |\psi_1\rangle\langle\psi_1|$, $R_{|\psi_1\rangle} = 2P_{|\psi_1\rangle} - \mathbb{1}$ and $R_{V_1} = 2P_{V_1} - \mathbb{1}$, where $\mathbb{1}$ is the identity operator on H, we know that

$$R_{|\psi_1\rangle}P_{V_2}|\psi_1\rangle - R_{V_1}P_{V_2}|\psi_1\rangle = 2(|\psi_1\rangle\langle\psi_1|P_{V_2}|\psi_1\rangle - P_{V_1}P_{V_2}|\psi_1\rangle).$$

Denote by $|\psi_1'\rangle$ the normalized vector of $P_{V_1}P_{V_2}|\psi_1\rangle$, i.e. $|\psi_1'\rangle = P_{V_1}P_{V_2}|\psi_1\rangle/\|P_{V_1}P_{V_2}|\psi_1\rangle\|$. Note that $\|P_{V_1}P_{V_2}|\psi_1\rangle\| > 0$. Decompose $|\psi_1'\rangle = \sqrt{q}|\psi_1\rangle + \sqrt{1-q}|\phi_1\rangle$ where $0 < q < 1$, $|\phi_1\rangle$ is also in V_1 and $\langle\phi_1|\psi_1\rangle = 0$. By the standard trick of considering $\frac{\langle\psi_1|+\langle\phi_1|}{\sqrt{2}}P_{V_2}\frac{|\psi_1\rangle+|\phi_1\rangle}{\sqrt{2}} = p$ and $\frac{\langle\psi_1|+i\langle\phi_1|}{\sqrt{2}}P_{V_2}\frac{|\psi_1\rangle-i|\phi_1\rangle}{\sqrt{2}} = p$, we get

$$\langle\psi_1|P_{V_2}|\phi_1\rangle + \langle\phi_1|P_{V_2}|\psi_1\rangle = 0, \quad \langle\psi_1|P_{V_2}|\phi_1\rangle - \langle\phi_1|P_{V_2}|\psi_1\rangle = 0,$$

which implies $\langle\psi_1|P_{V_2}|\phi_1\rangle = 0$. So finally by noting that

$$P_{V_1}P_{V_2}|\psi_1\rangle = |\psi_1\rangle\langle\psi_1|P_{V_2}|\psi_1\rangle + |\phi_1\rangle\langle\phi_1|P_{V_2}|\psi_1\rangle = |\psi_1\rangle\langle\psi_1|P_{V_2}|\psi_1\rangle,$$

we complete the proof of the fact. □

Lemma 7. *$M_2'(|\psi\rangle\langle\psi|)$ is exponentially close to $M_2''(|\psi\rangle\langle\psi|)$ in trace distance.*

Proof. Observe that by Part 1 of Lemma 5, the probability of succeeding in Step 2(a)(iv) of M_2' for any stage i is $2^{c(t-s)}$ independent of the auxiliary state $|\psi\rangle$. By Fact 3, if M_2' does not output Fail, then $M_2'(|\psi\rangle\langle\psi|)$ is equal to $M_2''(|\psi\rangle\langle\psi|)$. Finally, the probability of M_2' outputting Fail in any stage i is exponentially small; hence the probability of M_2' outputting Fail overall is also exponentially small. The completes the proof of the lemma. □

This completes the proof of Lemma 1 for non-uniform polynomial time quantum verifiers.

B Proof for the classical case of Lemma 1

The proof for the classical case is similar to that of the quantum case, only simpler as we do not need Watrous' rewinding lemma (Fact 3). The construction of a simulator M_2' for V_2' is similar to that in the quantum case. The basic idea is to rewind within each stage.

B.1 Classical simulator: construction

Classical simulator M_2':

1. Fix V_2''s internal random coin $r_{V_2'}$ making it a non-uniform deterministic polynomial time machine for Step 2 below;

```
2. for $i = 1$ to $N$
   (a) Repeat the following loop $N \cdot 2^{c(s-t)}$ times:
       i. Pick a random $r_i \in \{0,1\}^m$;
       ii. Conditioned on $R_1, \ldots, R_i$'s content being $r_1, \ldots, r_i$, run $M_1$ to get $(\hat{\alpha}_{i1}, \hat{\beta}_{i1}, \ldots, \hat{\alpha}_{ic}, \hat{\beta}_{ic})$;
       iii. for $j = 1$ to $c$
               // Simulate the DGW transformation
            A. Feed the message registers' content so far, $\mathrm{hist}_{ij}$, to $V_2'$ and obtain its output $f_{ij}$;
            B. Compute $y_{ij} = f_{ij}(\hat{\alpha}_{ij})$ and feed $y_{ij}$ to $V_2'$;
            C. Obtain $V_2'$'s output $\alpha_{ij}'$;
            // Check whether simulation for stage $i$ succeeded and rewind if not
       iv. If $\hat{\alpha}_{ij} \neq \alpha_{ij}'$ for some $j \in [c]$, Go to Step 2(a)(i);
       v. Else, proceed to stage $i+1$ in Step 2 since $\hat{\alpha}_{ij} = \alpha_{ij}'$ for all $j \in [c]$ and the simulation of stage $i$ succeeded;
   (b) Output Fail and terminate the whole simulation, as we have failed to simulate stage $i$ successfully;
3. Output $(r_{V_2'}; (f_{ij}, y_{ij}, \hat{\alpha}_{ij}, \hat{\beta}_{ij}))$ as the simulated view of $V_2'$.
```

The analysis of the classical simulator M_2' is similar to that of its quantum counterpart and is omitted from this extended abstract.