Zero-Knowledge Against Quantum Attacks

John Watrous

Institute for Quantum Information Science
University of Calgary

December 8, 2005

The Graph Isomorphism problem

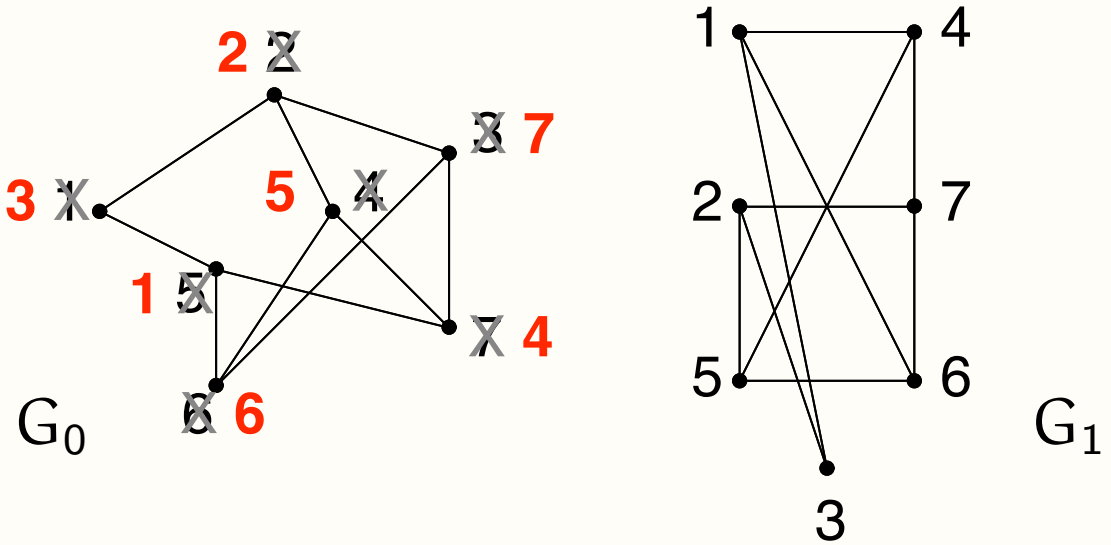
Consider the following problem (not known to be in BQP).

The Graph Isomorphism Problem

Input: Two (simple and undirected) graphs G_0 and G_1.

Yes: G_0 and G_1 are isomorphic ($G_0 \cong G_1$).

No: G_0 and G_1 are not isomorphic ($G_0 \ncong G_1$).



Isomorphism:

$1 \to 3, \quad 2 \to 2, \quad 3 \to 7, \quad 4 \to 5, \quad 5 \to 1, \quad 6 \to 6, \quad 7 \to 4.$

Zero-knowledge proof for graph isomorphism?

Consider the following scenario:

- There are two parties: a **prover** P and a (polynomial-time) **verifier** V.
- Both parties receive a pair of graphs (G_0, G_1).
- Under the assumption that $G_0 \cong G_1$, the prover P knows a permutation $\sigma \in S_n$ satisfying $\sigma(G_1) = G_0$.
- The prover P wants to convince the verifier V that $G_0 \cong G_1$ (even if this is not the case).
- The prover P does not want to reveal the permutation σ (or any other "knowledge" about G_0 and G_1).

A protocol (or *interactive proof system*) that achieves the final condition is said to be **zero-knowledge**.

Three required properties

A prover/verifier pair (P, V) constitutes a valid zero-knowledge interactive proof system for the graph isomorphism problem if the following three properties are satisfied:

1. Completeness. If $\mathrm{G}_0 \cong \mathrm{G}_1$, the prover P successfully convinces the verifier V that $\mathrm{G}_0 \cong \mathrm{G}_1$ (with high probability).

2. Soundness. If $\mathrm{G}_0 \ncong \mathrm{G}_1$, then **no prover** P' can successfully convince V that $\mathrm{G}_0 \cong \mathrm{G}_1$ (except with small probability).

3. Zero-knowledge property. If $\mathrm{G}_0 \cong \mathrm{G}_1$, then **no verifier** V' can *extract any knowledge** through an interaction with P.

---

* Definition required!

A zero-knowledge proof system for Graph Isomorphism

The following protocol (described for honest parties) is a zero-knowledge protocol for Graph Isomorphism [GOLDREICH, MICALI & WIDGERSON, 1991].

The GMW Graph Isomorphism Protocol

Assume the input is a pair (G_0, G_1) of simple, undirected graphs each having vertex set $\{1, \ldots, n\}$. Let $\sigma \in S_n$ be a permutation satisfying $\sigma(G_1) = G_0$ if $G_0 \cong G_1$, and let σ be arbitrary otherwise.

Prover's step 1: Choose $\pi \in S_n$ uniformly at random and send $H = \pi(G_0)$ to the verifier.

Verifier's step 1: Choose $a \in \{0, 1\}$ randomly and send a to the prover. (Implicit: challenge prover to show $H \cong G_a$.)

Prover's step 2: Let $\tau = \pi\sigma^a$ and send τ to the verifier.

Verifier's step 2: Accept if $\tau(G_a) = H$, reject otherwise.

Sequential repetition reduces soundness error...

A zero-knowledge proof system for Graph Isomorphism

The **completeness and soundness** properties are straightforward:

- If $G_0 \cong G_1$, then the verifier will accept every time.
- If $G_0 \ncong G_1$, then H cannot be isomorphic to both G_0 and G_1; so the verifier must reject with probability at least $1/2$ (regardless of any cheating prover's strategy).

If the protocol is repeated sequentially m times (with independent random choices for each repetition) then the prover can succeed every time if $G_0 \cong G_1$. If $G_0 \ncong G_1$, however, the maximum probability with which any prover could succeed in each repetition drops to 2^{-m}.

It remains to consider the zero-knowledge property...

Definition of zero-knowledge

Let $(V'(z), P)(x)$ denote the random variable describing the output of verifier V' given *auxiliary input* z after interacting with P on input x.

Definition of Zero-knowledge (classical)

An interactive proof system (P, V) for a given problem $A = (A_{yes}, A_{no})$ is **zero-knowledge** if, for every polynomial-time verifier V' there exists a **polynomial-time simulator** S such that, for every $x \in A_{yes}$,

$$(V'(z), P)(x) \quad \text{and} \quad S(x, z)$$

are indistinguishable.* [GOLDWASSER, MICALI & RACKOFF, 1989].

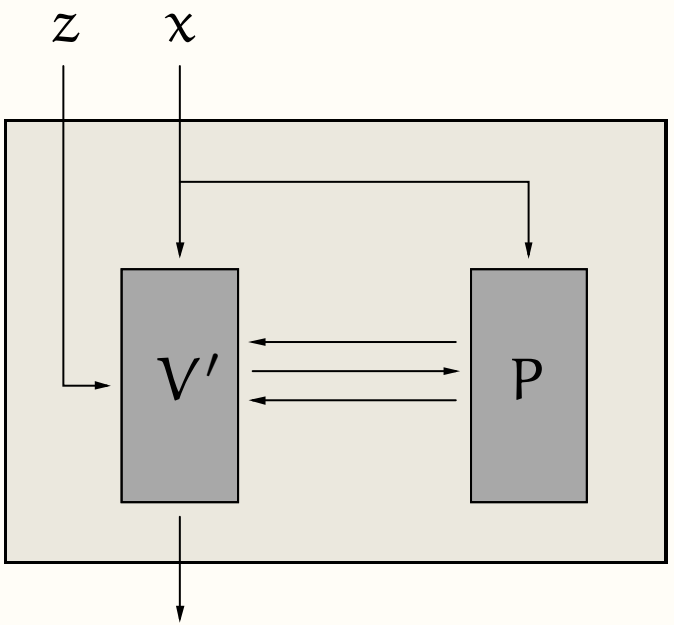
This **auxiliary input** definition captures the idea that zero-knowledge proofs should not **increase** knowledge.

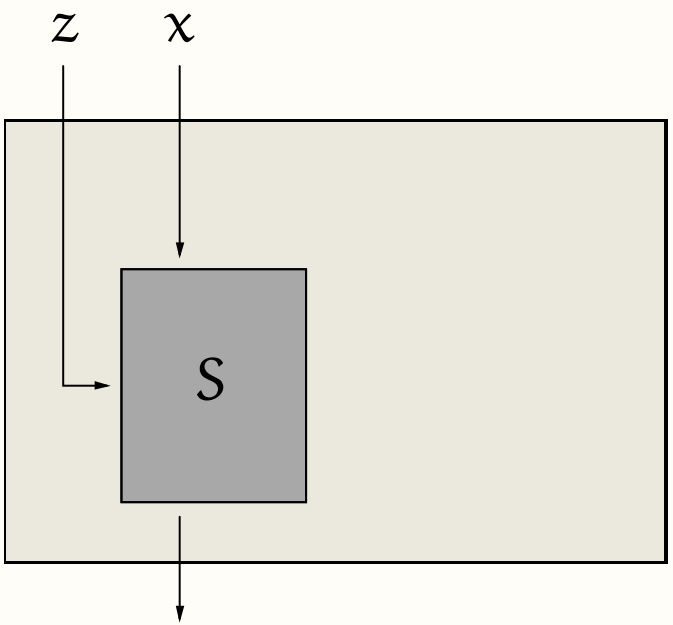
---

* Different notions of indistinguishability give rise to different variants of zero-knowledge, such as **statistical** and **computational** zero-knowledge.

Definition of zero-knowledge

In other words, these two processes should be indistinguishable provided x is a "yes" input to the problem being considered:





The zero-knowledge property requires nothing in case x is a "no" input...

Zero-knowledge property for the GMW protocol

How might a (classical) cheating verifier V' act?

Interaction between P and a cheating verifier V'

Prover's step 1: Choose $\pi \in S_n$ uniformly at random and send $H = \pi(G_0)$ to the verifier.

Verifier's step 1: Perform some **arbitrary** polynomial-time computation on (G_0, G_1), auxiliary input z, and H to obtain $a \in \{0, 1\}$. Send a to P.

Prover's step 2: Let $\tau = \pi\sigma^a$ and send τ to the verifier.

Verifier's step 2: Perform some **arbitrary** polynomial-time computation on (G_0, G_1), z, H, and τ to produce output.

Zero-knowledge property for the GMW protocol

We can simulate any such classical verifier as follows:

Simulator for V'

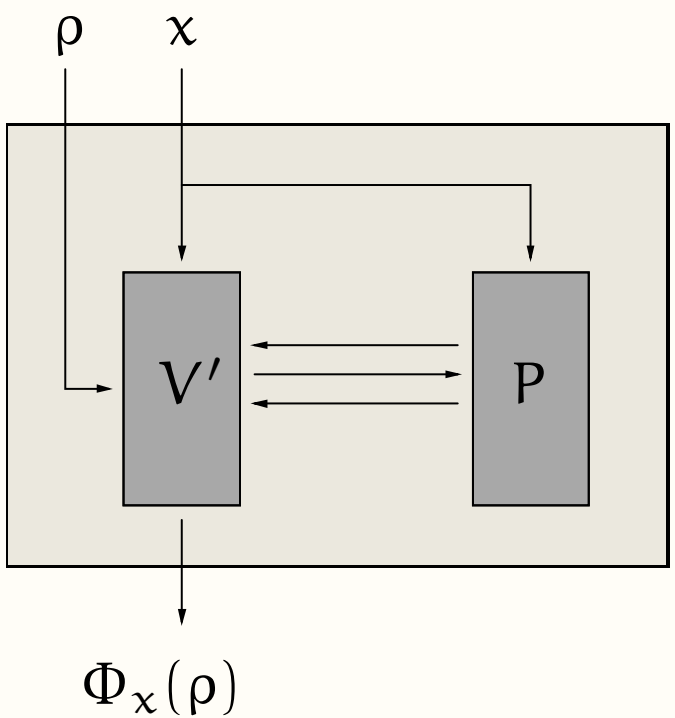
1. Choose $b \in \{0, 1\}$ and $\tau \in S_n$ uniformly, and let $H = \tau(G_b)$.
2. Simulate whatever V' does given prover message H. Let a denote the resulting message back to the prover.
3. If $a \neq b$ then rewind: go back to step 1 and try again.
4. Output whatever V' would after receiving τ.

Note: this gives an **expected** polynomial-time simulator: output agrees **exactly** with the distribution representing the view of V' . . . can be converted to a **worst case** polynomial-time simulator whose output agrees **almost exactly** with the view of V'.

Quantum version of the definition

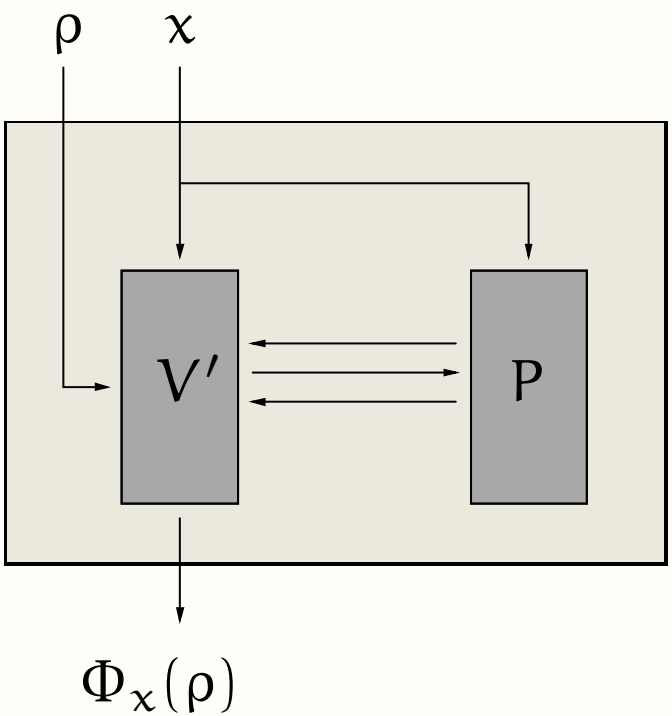
Suppose that some verifier V' tries to use **quantum information** to extract knowledge from P. (Note that the prover P is still classical, so the input x and any information exchanged between V' and P must be classical.)

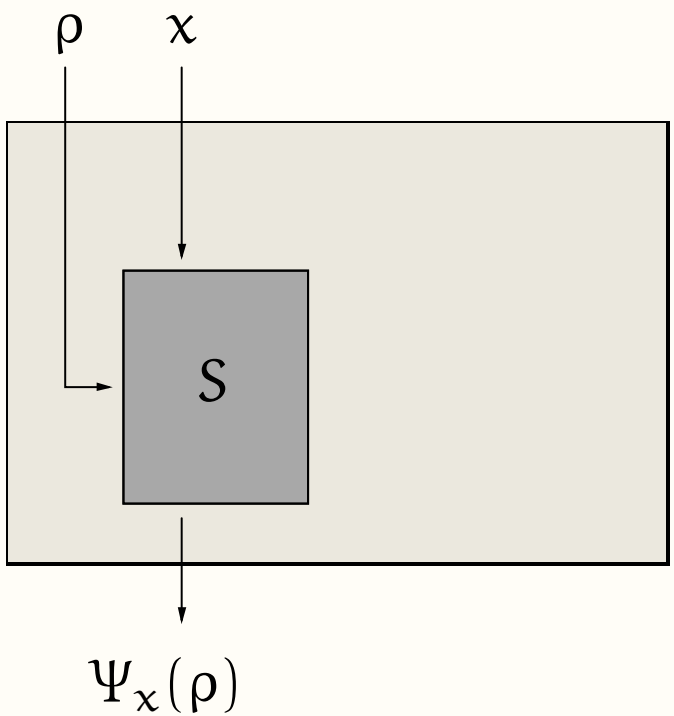
The interaction between V' and P on input x induces some **admissible mapping** on the auxiliary input:



Quantum version of the definition

If P is zero-knowledge even against a verifier V' that uses **quantum information**, then there should exist a simulator S that, given any "yes" input x, performs an **admissible mapping** Ψ_x on the auxiliary input that is **indistinguishable** from Φ_x:





Problem with the quantum definition?

These definitions are fairly straightforward. . . but have been considered problematic for several years.

The problem: No nontrivial protocols were previously shown to be zero-knowledge with respect to these definitions, even protocols already proved zero-knowledge in the classical setting.

The problem was first identified by Jeroen van de Graaf in his 1997 PhD thesis:

> *"Rewinding by reversing the unitary transformation induced by [the verifier], or taking snapshots is impossible.*
>
> *But. . . showing that rewinding by reversing or by taking snapshots is impossible does not show that no other ways to rewind in polynomial time exist."*
>
> [VAN DE GRAAF, 1997]

Problem with the quantum definition?

Consider a deviant quantum verifier V' for the Graph Isomorphism protocol that acts as follows:

Verifier V'

1. Begin the protocol with auxiliary quantum register **W** (possibly provided by a third party, and possibly entangled with some other register).
2. Receive graph H from P.
3. Measure **W** with respect to some binary-valued projective measurement $\{\Pi_0^H, \Pi_1^H\}$ **that depends on** H. Let a be the outcome, and send a to P.
4. After receiving τ from P, output (H, a, τ) along with the register **W**.

How can we simulate such a verifier?

Problem with the quantum definition?

Two principles are working against us:

- The **no cloning theorem** prevents making a copy of the auxiliary input register's state.
- Measurements are **irreversible**.

Suppose that we randomly choose b and τ, and let $\mathrm{H} = \tau(\mathrm{G}_{\mathrm{b}})$ as for our simulator before...

...if the measurement $\{\Pi_0^{\mathrm{H}}, \Pi_1^{\mathrm{H}}\}$ gives outcome b, the simulation works. But if the measurement outcome is **not** b, then the state of **W** is irreparably harmed, and we cannot recover the original state.

Note: one can imagine **potential attacks** based on this issue. It seems plausible that a verifier V' could **transfer knowledge** to a third party that an interaction with P really took place.

New results

In the remainder of this talk I will argue that the GMW Graph Isomorphism protocol is indeed zero-knowledge against quantum verifiers:

- For any quantum verifier V', there exists a simulator S that induces precisely the same admissible mapping as the interaction between V' and P (on a "yes" input to the problem).
- The method gives a way to "rewind" the simulator, but it requires more than just reversing the verifier's actions. (The entire simulation will be quantum, even though the prover is classical.)
- The method generalizes to several other protocols (but I will only discuss the Graph Isomorphism example in this talk for simplicity).

Assumptions on $\mathcal{V}'$

Assume $\mathcal{V}'$ uses three registers:

W: stores the auxiliary input.
V: represents workspace of arbitrary size.
A: single qubit representing the message sent by $\mathcal{V}'$.

Register **W** starts in the auxiliary state, and registers **V** and **A** are initialized to all zeroes.

Assume $\mathcal{V}'$ operates as follows:

- For each graph H on $\mathfrak{n}$ vertices, $\mathcal{V}'$ has a corresponding unitary transformation $\mathcal{V}_{\mathrm{H}}$ that acts on $(\mathbf{W}, \mathbf{V}, \mathbf{A})$.
- Upon receiving H from P, the $\mathcal{V}'$ applies $\mathcal{V}_{\mathrm{H}}$ to $(\mathbf{W}, \mathbf{V}, \mathbf{A})$, measures **A** in the standard basis, and sends the result $\mathfrak{a}$ to P.
- After P responds with some permutation τ, $\mathcal{V}'$ simply outputs $(\mathbf{W}, \mathbf{V}, \mathbf{A})$ along with the prover messages H and τ.

Simulator construction

The simulator will use registers **W**, **V**, and **A** along with:

Y: stores the provers first message.
B: stores the simulator's guess for a.
Z: stores the prover's second message.
R: stores "randomness" used to generate transcripts.

Define a unitary operator V on $(\mathbf{W}, \mathbf{V}, \mathbf{A}, \mathbf{Y})$ that represents a unitary realization of V':

$$V = \sum_{H} V_H \otimes |H\rangle \langle H| .$$

Define T to be a unitary operation on registers $(\mathbf{Y}, \mathbf{B}, \mathbf{Z}, \mathbf{R})$ for which

$$T : |00\cdots 0\rangle \mapsto \frac{1}{\sqrt{2n!}} \sum_{b,\tau} |\tau(G_b)\rangle |b\rangle |\tau\rangle |b,\tau\rangle .$$

The operation T produces a superposition over *transcripts*.

Simulator construction

Now define the simulator as follows:

Simulator

1. Perform T, followed by V.
2. Perform a measurement $\{\Pi_0, \Pi_1\}$ whose outcome corresponds to the XOR of **A** and **B** (in the computational basis).
3. If the measurement outcome is 1, we need to **rewind and try again:**

   - Perform V^* followed by T^*.
   - Perform a **phase flip** in case any of the qubits in any of the registers $(\mathbf{V}, \mathbf{A}, \mathbf{Y}, \mathbf{B}, \mathbf{Z}, \mathbf{R})$ is set to 1 (i.e., perform $2\Delta - \mathrm{I}$, where $\Delta = \mathrm{I}_{\mathbf{W}} \otimes |00\cdots0\rangle\langle 00\cdots0|$.)
   - Perform T followed by V.

4. Output registers $(\mathbf{W}, \mathbf{V}, \mathbf{A}, \mathbf{Y}, \mathbf{Z})$. (Registers **B** and **R** are traced out.)

Analysis of simulator

Assume that the auxiliary input is $|\psi\rangle$, and $x = (G_0, G_1)$ for $G_0 \cong G_1$. Let

$$|\varphi\rangle = |\psi\rangle\,|00\cdots0\rangle$$

be the state of all registers given this input.

The simulator performs T, then V, then measures w.r.t. $\{\Pi_0, \Pi_1\}$. Assuming $G_0 \cong G_1$, the outcome will **always be uniformly distributed.**

First, suppose that the measurement $\{\Pi_0, \Pi_1\}$ gives **outcome 0**. The resulting state of all registers is

$$|\sigma_0\rangle = \sqrt{2}\Pi_0 V T\,|\varphi\rangle\,.$$

This is the **target state**: it represents a successful simulation because

$$\mathrm{tr}_{\mathbf{B},\mathbf{R}}\,|\sigma_0\rangle\,\langle\sigma_0| = \Phi(|\psi\rangle\,\langle\psi|).$$

(Nothing is surprising here. . . the simulator has been lucky and didn't need to rewind.)

Analysis of simulator

Suppose on the other hand that the **measurement outcome was 1**. The resulting state is

$$|\sigma_1\rangle = \sqrt{2}\Pi_1 V T |\varphi\rangle .$$

Time to rewind and try again...

Performing the "rewind and try again" procedure results in the state

$$V T (2\Delta - I) T^* V^* |\sigma_1\rangle .$$

Claim

$$V T (2\Delta - I) T^* V^* |\sigma_1\rangle = |\sigma_0\rangle \quad \text{(the target state)}.$$

Note: this would not happen for **arbitrary** choices of $|\varphi\rangle$, V, T, Π_0, Π_1, etc...the claim relies on the fact that the measurement $\{\Pi_0, \Pi_1\}$ gives outcome 0 and 1 with equal probability for **all** choices of $|\psi\rangle$.

Proof of claim

The fact that the measurement $\{\Pi_0, \Pi_1\}$ gives outcomes 0 and 1 with equal probability for **all** choice of $|\psi\rangle$ implies

$$\Delta T^* V^* \Pi_0 V T \Delta = \Delta T^* V^* \Pi_1 V T \Delta = \frac{1}{2}\Delta.$$

Therefore

$$\begin{aligned}
&\langle \sigma_0 | V T (2\Delta - I) T^* V^* | \sigma_1 \rangle \\
&\qquad = 2 \langle \varphi | T^* V^* \Pi_0 V T (2\Delta - I) T^* V^* \Pi_1 V T | \varphi \rangle \\
&\qquad = 4 \langle \varphi | T^* V^* \Pi_0 V T \Delta T^* V^* \Pi_1 V T | \varphi \rangle \\
&\qquad\quad - 2 \langle \varphi | T^* V^* \Pi_0 V T T^* V^* \Pi_1 V T | \varphi \rangle \\
&\qquad = 4 \langle \varphi | \Delta T^* V^* \Pi_0 V T \Delta T^* V^* \Pi_1 V T \Delta | \varphi \rangle \\
&\qquad = \langle \varphi | \Delta | \varphi \rangle \\
&\qquad = 1,
\end{aligned}$$

so $V T (2\Delta - I) T^* V^* |\sigma_1\rangle = |\sigma_0\rangle$. □

Analysis of simulator

This establishes that the admissible map Ψ agrees with the map Φ corresponding to the actual interaction on all pure state auxiliary inputs:

$$\Psi(|\psi\rangle\langle\psi|) = \Phi(|\psi\rangle\langle\psi|)$$

for all $|\psi\rangle$.

Admissible maps are **completely determined** by their actions on pure state inputs, however, so

$$\Psi = \Phi;$$

the simulator **agrees precisely** with the actual interaction on **every possible state** of the auxiliary input register (including the possibility it is entangled with another register).

Other protocols

The simulation method just described can be adapted to prove several other protocols are zero-knowledge against quantum attacks, including:

- Quantum protocols for any problem having an **honest verifier** quantum statistical zero-knowledge proof system:

$$\mathrm{QSZK} = \mathrm{QSZK}_{\mathrm{HV}}.$$

- The Goldreich-Micali-Wigderson **Graph 3-Coloring** protocol assuming unconditionally binding and quantum computationally concealing bit commitments. (See [Adcock & Cleve, 2002].)
- Presumably several other proof systems...

Adapting the simulator to other protocols may require iterating the "rewind and try again" process.

Future work/open questions

1. Find further applications and generalizations of the method.

2. Identify limitations of the method.

3. Identify good candidates for quantum one-way functions.