Classical Verification of Quantum Computations

Urmila Mahadev

UC Berkeley

September 12, 2018
Classical versus Quantum Computers

- Can a classical computer verify a quantum computation?
  - Classical output (decision problem)

- Quantum computers compute in superposition
  - Classical description is exponentially large!

- Classical access is limited to measurement outcomes
  - Only $n$ bits of information
Can a classical computer verify the result of a quantum computation through interaction (Gottesman, 2004)?
- Classical complexity theory: $\text{IP} = \text{PSPACE}$ [Shamir92]

- $\text{BQP} \subseteq \text{PSPACE}$: Quantum computations can be verified, but only through interaction with a much more powerful prover

- Scaled down to an efficient quantum prover?
Relaxations

Error correcting codes
[BFK08][ABE08][FK17][ABEM17]

Bell inequalities
[RUV12]
In this talk: use post quantum classical cryptography to control the BQP prover

To do this, require a specific primitive: trapdoor claw-free functions
Core Primitive

- **Trapdoor claw-free functions** $f$:
  - Two to one
  - Trapdoor allows for efficient inversion: given $y$, can output $x_0, x_1$ such that $f(x_0) = f(x_1) = y$
  - Hard to find a claw $(x_0, x_1)$: $f(x_0) = f(x_1)$
  - Approximate version built from learning with errors in [BCMVV18]

- Quantum advantage: sample $y$ and create a superposition over a random claw

\[
\frac{1}{\sqrt{2}}(|x_0\rangle + |x_1\rangle)
\]

which allows sampling of a string $d \neq 0$ such that

\[
d \cdot (x_0 \oplus x_1) = 0
\]
\[
\frac{1}{\sqrt{2}}(|x_0\rangle + |x_1\rangle) \text{ or } d \cdot (x_0 \oplus x_1) = 0
\]

- Classical verifier can challenge quantum prover
  - Verifier selects \( f \) and asks for \( y \)
  - Verifier has leverage through the trapdoor: can compute \( x_0, x_1 \)

- First challenge: ask for preimage of \( y \)

- Second challenge: ask for \( d \)
\[ \frac{1}{\sqrt{2}}(|x_0\rangle + |x_1\rangle) \quad \text{or} \quad d \cdot (x_0 \oplus x_1) = 0 \]

- In [BCMVV18], used to generate randomness:
  - Hardcore bit: hard to hold both \( d \) and either \( x_0, x_1 \) at the same time
  - Prover must be probabilistic to pass
\[ \frac{1}{\sqrt{2}} (|x_0\rangle + |x_1\rangle) \quad \text{or} \quad d \cdot (x_0 \oplus x_1) = 0 \]

- **Verification:**
  - TCFs are used to constrain prover
  - Use extension of approximate TCF family built in [BCMVV18]
    - Require [BCMVV18] hardcore bit property: hard to hold both \( d \) and either \((x_0, x_1)\)
    - Require one more hardcore bit property: there exists \( d \) such that for all claws \((x_0, x_1)\), \( d \cdot (x_0 \oplus x_1) \) is the same bit and is hard to compute
How to Create a Superposition Over a Claw

\[ \frac{1}{\sqrt{2}} (|x_0\rangle + |x_1\rangle) \]

1. Begin with a uniform superposition over the domain:

\[ \frac{1}{\sqrt{|X|}} \sum_{x \in X} |x\rangle \]

2. Apply the function $f$ in superposition:

\[ \frac{1}{\sqrt{|X'|}} \sum_{x \in X'} |x\rangle |f(x)\rangle \]

3. Measure the last register to obtain $y$
\[ \frac{1}{\sqrt{2}} (|x_0\rangle + |x_1\rangle) \]

- Performing a Hadamard transform on the above state results in:
  \[ \frac{1}{\sqrt{|x'|}} \sum_d ((-1)^{d \cdot x_0} + (-1)^{d \cdot x_1}) |d\rangle \]

- By measuring, obtain a string \( d \) such that
  \[ d \cdot (x_0 \oplus x_1) = 0 \]
Goal: classical verification of quantum computations through interaction

- Define a measurement protocol
  - The prover constructs an $n$ qubit state $\rho$ of his choice
  - The verifier chooses 1 of 2 measurement bases for each qubit
  - The prover reports the measurement result of $\rho$ in the chosen basis

- Link measurement protocol to verifiability

- Construct and describe soundness of the measurement protocol
\[ |\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle \]

- Standard: obtain \( b \) with probability \(|\alpha_b|^2\)
- Hadamard:

\[
H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}
\]

\[
H |\psi\rangle = \frac{1}{\sqrt{2}} (\alpha_0 + \alpha_1) |0\rangle + \frac{1}{\sqrt{2}} (\alpha_0 - \alpha_1) |1\rangle
\]

Obtain \( b \) with probability \( \frac{1}{2} \left| \frac{1}{\sqrt{2}} (\alpha_0 + (-1)^b \alpha_1) \right|^2 \)
**Measurement Protocol Definition**

*Measurement protocol*: interactive protocol which forces the prover to behave as the verifier’s trusted measurement device.

![Diagram](image.png)
Key issue: adaptivity; what if $\rho$ changes based on measurement basis?

- Maybe the prover never constructs a quantum state, and constructs classical distributions instead
• **Soundness:** if the verifier accepts, there exists a quantum state *independent of the verifier’s measurement choice* underlying the measurement results.
Measurement Protocol Soundness

Soundness: if $P$ is accepted with high probability, there exists a state $\rho$ such that for all $h$, $D_{\rho,h}$ and $D_{P,h}$ are computationally indistinguishable.
• The measurement protocol implements the following model:

• Prover sends qubits of state $\rho$ and verifier measures

• Next: show that quantum computations can be verified in the above model
Quantum Analogue of NP

- To verify an efficient classical computation, reduce to a 3-SAT instance, ask for satisfying assignment and verify that it is satisfied

\[
\begin{align*}
3\text{-SAT} & \iff \text{Local Hamiltonian} \\
n \text{bit variable assignment } x & \iff n \text{ qubit quantum state} \\
\text{Number of unsatisfied clauses} & \iff \text{Energy}
\end{align*}
\]

- To verify an efficient quantum computation, reduce to a local Hamiltonian instance \( H \), ask for ground state and verify that it has low energy
  - If the instance is in the language, there exists a state with low energy
Quantum Analogue of NP

3 SAT $\iff$ Local Hamiltonian
Assignment $\iff$ Quantum state
Number of unsatisfied clauses $\iff$ Energy

To verify that a state has low energy with respect to $H = \sum_i H_i$:

- Each $H_i$ acts on at most 2 qubits
- To measure with respect to $H_i$, only Hadamard/standard basis measurements are required [BL08]
Verification with a Quantum Verifier

- Prover sends each qubit of $\rho$ to the quantum verifier

- The quantum verifier chooses $H_i$ at random and measures, using only Hadamard/standard basis measurements [MF2016]

- Measurement protocol can be used in place of the measurement device to achieve verifiability
• Use a TCF with more structure: pair \( f_0, f_1 \) which are injective with the same image

• Given \( f_0, f_1 \), the honest quantum prover entangles a single qubit of his choice with a claw \((x_0, x_1) (y = f_0(x_0) = f_1(x_1))\).

\[
|\psi\rangle \rightarrow \sum_{b \in \{0, 1\}} \alpha_b |b\rangle |x_b\rangle = \text{Enc}(|\psi\rangle)
\]

• Once \( y \) is sent to the verifier, the verifier now has leverage over the prover’s state: he knows \( x_0, x_1 \) but the prover does not
• The verifier generates a TCF $f_0, f_1$ and the trapdoor.

• Given $f_0, f_1$, the honest quantum prover entangles a single qubit of his choice with a claw $(x_0, x_1)$ ($y = f_0(x_0) = f_1(x_1)$).

$$|\psi\rangle = \sum_{b \in \{0,1\}} \alpha_b |b\rangle \rightarrow \sum_{x \in \mathcal{X}} \sum_{b \in \{0,1\}} \alpha_b |b\rangle |x\rangle |f_b(x)\rangle$$

$$f_b(x) = y \quad \sum_{b \in \{0,1\}} \alpha_b |b\rangle |x_b\rangle = \text{Enc}(|\psi\rangle)$$

• Given $y$, the verifier uses the trapdoor to extract $x_0, x_1$. 
• Upon receiving $y$, the verifier chooses either to test or to delegate measurements

• If a test round is chosen, the verifier requests a preimage $(b, x_b)$ of $y$

• The honest prover measures his encrypted state in the standard basis:

$$\text{Enc}(|\psi\rangle) = \sum_{b\in\{0,1\}} \alpha_b |b\rangle |x_b\rangle$$

• Point: the verifier now knows the prover’s state must be in a superposition over preimages
Delegating Hadamard Basis Measurements

- Prover needs to apply a Hadamard transform:

\[ \text{Enc}(|\psi\rangle) = \sum_{b \in \{0,1\}} \alpha_b |b\rangle |x_b\rangle \rightarrow H(\sum_{b \in \{0,1\}} \alpha_b |b\rangle) = H|\psi\rangle \]

- Issue: \( x_0, x_1 \) prevent interference, and prevent the application of a Hadamard transform

- Solution: apply the Hadamard transform to the entire encoded state, and measure the second register to obtain \( d \)
• This results in a different encoding ($X$ is the bit flip operator):

$$\text{Enc}(|\psi\rangle) \xrightarrow{H} X^{d \cdot (x_0 \oplus x_1)} H |\psi\rangle$$

• Verifier decodes measurement result $b$ by XORing $d \cdot (x_0 \oplus x_1)$

• Protocol with honest prover:
Measurement Protocol So Far

- **Soundness**: there exists a quantum state *independent of the verifier’s measurement choice* underlying the measurement results.

- **Necessary condition**: messages required to delegate standard basis must be computationally indistinguishable.

- **To delegate standard basis measurements**: only need to change the first message.
Delegating Standard Basis Measurements

- Let $g_0, g_1$ be trapdoor injective functions: the images of $g_0, g_1$ do not overlap
  - The functions $(f_0, f_1)$ and $(g_0, g_1)$ are computationally indistinguishable

- If prover encodes with $g_0, g_1$ rather than $f_0, f_1$, this acts as a standard basis measurement:
  \[
  \sum_{b \in \{0,1\}} \alpha_b |b\rangle \rightarrow \sum_{b \in \{0,1\}, x} \alpha_b |b\rangle |x\rangle |g_b(x)\rangle
  \]

- With use of trapdoor, standard basis measurement $b$ can be obtained from $y = g_b(x)$
Delegating Standard Basis Measurements

- Protocol is almost the same, except $f_0, f_1$ is replaced with $g_0, g_1$

- Verifier ignores Hadamard measurement results; only uses $y$ to recover standard basis measurement
Measurement Protocol Recap

- Goal: use the prover as a blind, verifiable measurement device
- Verifier selects basis choice; sends claw free function for Hadamard basis and injective functions for standard basis
- Verifier either tests the structure of the state or requests measurement results
Soundness Intuition: Example of Cheating Prover

- Recall adaptive cheating strategy: prover fixes two bits, $b_H$ and $b_S$, which he would like the verifier to stores as his Hadamard/standard basis measurement results.

- Assume there is a claw $(x_0, x_1)$ and a string $d$ for which the prover knows both $x_{b_S}$ and $d \cdot (x_0 \oplus x_1)$.

- How to cheat:
  - To compute $y$: prover evaluates received function on $x_{b_S}$ ($y = g_{b_S}(x_{b_S})$ or $y = f_{b_S}(x_{b_S})$).
  - When asked for a Hadamard measurement: prover reports $d$ and $b_H \oplus d \cdot (x_0 \oplus x_1)$.
Soundness rests on two hardcore bit property of TCFs:

1. For all $d \neq 0$ and all claws $(x_0, x_1)$, it is computationally difficult to compute both $d \cdot (x_0 \oplus x_1)$ and either $x_0$ or $x_1$.

2. There exists a string $d$ such that for all claws $(x_0, x_1)$, the bit $d \cdot (x_0 \oplus x_1)$ is the same and computationally indistinguishable from uniform.
How to Prove Soundness

Key step: enforcing structure in prover’s state

[BFK08][ABE08][FK17][ABEM17]  [RUV12]
Verifier sends qubits encoded with secret error correcting code to the prover.
Verifier plays CHSH with the provers and checks for a Bell inequality violation. If prover passes, he must be holding Bell pairs.
Enforcing structure?

- No way of using previous techniques
- Use test round of measurement protocol as starting point

At some point in time, prover’s state must be of the form:

$$\sum_{b \in \{0,1\}} \alpha_b \left| b \right\rangle \left| x_b \right\rangle \left| \psi_{b,x_b} \right\rangle$$

or

$$\left| b \right\rangle \left| x_b \right\rangle \left| \psi_{b,x_b} \right\rangle$$
Why is this format useful in proving the existence of an underlying quantum state?

\[ \sum_{b \in \{0,1\}} \alpha_b |b\rangle |x_b\rangle |\psi_{b,x}\rangle \quad \text{or} \quad |b\rangle |x_b\rangle |\psi_{b,x}\rangle \]

- Can be used as starting point for prover, followed by deviation from the protocol, measurement and decoding by the verifier
  - Deviation is an arbitrary unitary operator $U$
  - Verifier’s decoding is $d \cdot (x_0 \oplus x_1)$

- The part of the unitary $U$ acting on the first qubit is therefore \textit{computationally randomized}, by both the initial state and the verifier’s decoding
  - Pauli twirl technique?
Why is this format useful in proving the existence of an underlying quantum state?

\[ \sum_{b \in \{0,1\}} \alpha_b |b\rangle |x_b\rangle |\psi_{b,x_b}\rangle \quad \text{or} \quad |b\rangle |x_b\rangle |\psi_{b,x_b}\rangle \]

- Difficulty in using Pauli twirl: converting this computational randomness into a form which can be used to simplify the prover’s deviation
  - Rely on hardcore bit properties regarding \( d \cdot (x_0 \oplus x_1) \)
Conclusion

- Verifiable, secure delegation of quantum computations is possible with a classical machine

- Rely on quantum secure trapdoor claw-free functions (from learning with errors)
  - Use TCF to characterize the initial space of the prover
  - Strengthen the claw-free property to complete the characterization and prove the existence of a quantum state
Thanks!