Classical Verification of Quantum Computation

Ziyang Jin

Department of Computer Science
University of Toronto
ziyang@cs.toronto.edu

Quantum computing has been a trending research area since the 1990s. One breakthrough was the quantum factoring algorithm introduced by Peter Shor in 1994. Many popular encryption schemes such as RSA, Diffie-Hellman, ElGamal will be broken by running Shor's algorithm on a reliable quantum computer, if one builds it in the future. Therefore, cryptography needs to evolve with the advances of quantum computing.

Professor Yael Tauman Kalai, from MIT, has been working in cryptography for over 25 years. Her seminal work on delegating computation [Goldwasser-Kalai-Rothblum 08] has found many applications in cloud computing. At the Fields Institute, Yael gives a talk on classical verification of quantum computation, when computation is delegated to quantum computers.

Today, we are living in a world where people are still using classical devices. However, given that several big companies such as Google and IBM invest heavily in building powerful quantum computers, we might enter a world where there exists a set of powerful quantum computers, while most people still use classical devices.

1 Delegating Quantum Computations

Yael's talk prepares us for a forseeable quantum future. Suppose some tech giants build a set of quantum computers. How do we leverage these quantum computers to benefit the society? One possibility is that these quantum computers will be deployed in the cloud and people will have remote access to quantum computers using their classical devices. Then, people can connect to the cloud and ask the quantum computer to perform expensive computation tasks and return the result. To put it formally, suppose a client has input x and a function f to compute. The client sends f and x to the quantum server, and the server returns $f(x)$.

In this setup, here are two concerns. First, people may not want the quantum computer to know their private input x, as it could be their personal health data? Second, how do people verify that the returned result is correct, using their classical devices? There is nothing to prevent the cloud service from returning some random value without actually computing $f(x)$ in order to save resources.

To address the first concern, people have developed techniques such as Quantum Fully Homomorphic Encryption (QFHE), which is the quantum extension to the classical Fully Homomorphic Encryption (FHE) introduced by Gentry et al. in 2009.

To address the second concern, since we only have classical devices, it is natural to ask the quantum computer to output a classical proof of correctness. Formally speaking, in a *quantum-classical proof system*, there is a quantum prover P and a polynomial time classical verifier V. The prover needs to prove the quantum computation was performed honestly. Or in complexity theory language, to prove a string x is in some language L. The proof system needs to satisfy completeness and soundness. Completeness means that if $x \in L$, then V accepts the proof from honest prover. Soundness means that if $x \notin L$, then V rejects every cheating quantum prover P^* with high probability. Additionally, we require another property called "Efficiency". It means that the honest prover's runtime is close to the time it takes to decide L, thus it is not too much of a burden for the quantum computer to produce the proof of correctness after performing the quantum computation task. For example, if the language L we consider is in complexity class **BQP** (Bounded-Error Quantum Polynomial Time, a quantum analog of classical complexity class **P**), then the honest prover P should run in quantum polynomial time. If the language L is in complexity class **QMA** (Quantum Merlin-Arthur, which generalizes the classical class **NP** to the quantum setting), then given a witness, the honest prover should run in quantum polynomial time.

Can we construct a quantum-classical proof system that satisfies all of completeness, soundness, and efficiency? Till today, this remains an open question. However, people have been making progress—some positive answers are discovered when the definition of soundness is relaxed. In the proof system defined

above, the cheating prover's runtime is unbounded. In cryptography, people also study *argument* systems, where the setup is the same except that the cheating prover can only run in polynomial time.

2 Mahadev's Measurement Protocol

In her 2018 paper "Classical Verification of Quantum Computations", Urmila Mahadev, showed that for any language $L \in \mathbf{BQP}$, we can construct a quantum-classical argument system, where the soundness holds against a quantum polynomial time cheating prover assuming Learning with Errors (LWE), a "flagship" hardness assumption that no efficient quantum attacks are found yet. This result is known as *Mahadev's measurement protocol*, and it initiated a surge of research on quantum-classical argument systems. Let C be the quantum circuit that decides L, and let $|C|$ be the number of quantum gates in the circuit. One downside of Mahadev's protocol is that the verifier's runtime grows linearly with $|C|$. Thus, it is natural to ask if we can construct succinct argument systems for **BQP**? Here, succinct means that the verifier's runtime and the length of messages communicated during the protocol grows sublinear in $|C|$. Yael and her collaborators answered this question positively. In [Gunn-Kalai-Natarajan-Villanyi 25], they give a succinct argument for **BQP** under LWE, in which the verifier's runtime is $\mathsf{polylog}(|C|)$. Their result is based on the "commit-and-prove" framework in the original paper by Mahadev.

3 Construction from Multiple Provers

An alternative way to Mahadev's approach is Multi-Prover Interactive Proofs with entangled provers ($\mathbf{MIP}^*$) (see Figure 1). For simplicity, we consider the two-prover case. There are two unbounded quantum provers P_1, P_2 and they cannot communicate with each other, but they can share arbitrary entangled quantum states. To prove $x \in L$, the classical verifier V sends queries to P_1, P_2 separately, and P_1, P_2 respond to the verifier separately. In this setup, cheating becomes more difficult—without knowing the queries to the other prover, it is hard for one cheating prover to generate query responses that are consistent with the other cheating prover's responses.

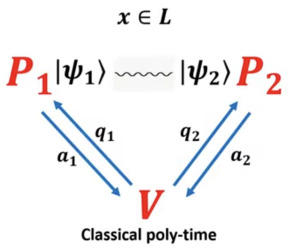

Fig. 1. Multi-Prover Interactive Proofs with entangled provers

In $\mathbf{MIP}^*$ model, Reichardt, Unger, and Vazirani showed that there exists a protocol for every language $L \in \mathbf{QMA}$. How do we leverage the power of $\mathbf{MIP}^*$ in the case of a single prover? In [Kalai-Lombardi-Vaikuntanathan-Yang 23], Yael and her collaborators exhibited a "compiler" that transforms a $\mathbf{MIP}^*$ protocol into an interactive argument system.

Intuitively, the prover P in the argument system plays the role of both P_1 and P_2 in $\mathbf{MIP}^*$. In other words, P emulates both P_1 and P_2 in her head. The challenge here is that when P plays the role of P_2, she should not know about the queries sent to P_1. The key idea is to use quantum fully homomorphic encryption (QFHE). The prover first plays the role of P_1, and the verifier sends encryted queries, denoted by $\mathsf{QEnc}(q_1)$ to P. Then the prover P computes the answer to q_1 under the hood of QFHE, and sends back the answer $\mathsf{QEnc}(a_1)$. Next, the prover plays the role of P_2. The verifier sends q_2 and the prover responses with a_2 in the

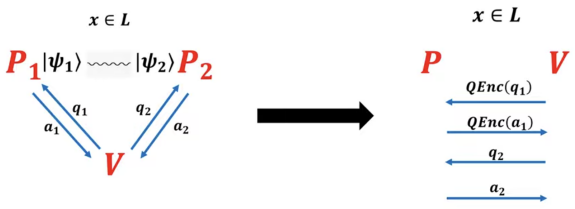

Fig. 2. KLVY Compiler from **MIP*** to Argument

clear. Since q_1 is encrypted by a post-quantum encryption scheme, P does not know about q_1, which makes P responds to q_2 without the knowledge of q_1. We call this construction the *KLVY compiler* (see Figure 2).

The KLVY compiler has some subtleties—it is not clear if the KLVY compiler always produces a sound argument system from every **MIP*** protocol. This is because the soundness proof requires us to reduce a cheating prover P^* in the compiled argument system to two cheating provers P_1^*, P_2^* in **MIP***. Given the quantum states used by the cheating prover P^*, it is hard to figure out how the entangled quantum states will be used by the two cheating provers P_1^*, P_2^*. Therefore, in the original paper for KLVY compiler, the authors only proved the soundness against classical cheating provers. Recently, people have shown that for some specific **MIP*** protocols, the compiler indeed gives a sound argument system.

4 Succinct Classical Commitment to Quantum States

Before going to the next section, let's refresh our mind on basics of quantum computation. A *qubit* is a superposition $\alpha|0\rangle + \beta|1\rangle$ such that $\alpha^2 + \beta^2 = 1$, where α, β are complex numbers. A quantum state $|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x|x\rangle$ is a superposition over many qubits where $\sum \alpha_x^2 = 1$. When you *measure* a qubit, it collapses to 0 with probability α^2 and collapses to 1 with probability β^2. This is calle the standard basis (or Z basis) measurement. Quantum states can be manipulated via unitary transformations. In particular, we can apply a Hadamard transformation and then measure the quantum state. This is called a Hadamard basis (or X basis) measurement. Also, there is a "no cloning principle" for quantum states, which says that a quantum state cannot be cloned.

In [Gunn-Kalai-Natarajan-Villanyi 25], Yael and her collaborators are able to make Mahadev's protocol succinct, using their newly developed tool—succinct classical commitment to quantum states. In [Huang-Kalai 25], Yael uses the same tool to improve the KLVY compiler such that it now produces sound argument systems from any protocol in **MIP***.

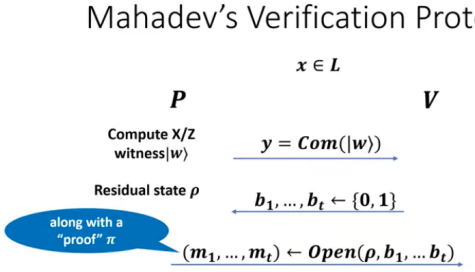

Fig. 3. Mahadev's Protocol, Simplified

To understand succinct classical commitment to quantum states, we need to take a closer look at Mahadev's protocol (see Figure 3), which briefly works as follows. A commitment scheme consists of two algorithms, Com and Open. Initially, the prover P has some quantum state $|w\rangle$. Then it runs the commit

algorithm $\mathsf{Com}(|w\rangle)$ and outputs a classical commitment, represented by string y, to the verifier. Note that after running the commit algorithm on quantum state $|w\rangle$, the prover P is left with a residual quantum state ρ. Let t be a number slightly bigger than the number of gates in the quantum circuit that decides L. In the second round, the verifier samples t random bits $b_1, \ldots, b_t$ and sends to the prover. In the final round, the prover sees the bits t_i for $i \in \{1, \ldots, t\}$. If $t_i = 0$, it opens the commitment in standard basis; else $t_i = 1$, it opens the commitment in Hadamard basis. The prover runs the opening algorithm $\mathsf{Open}(\rho, b_1, \ldots, b_t)$ to obtain the openings $(m_1, \ldots, m_t)$. Then it sends the openings along with a proof π that shows the openings are valid. The commitment scheme should satisfy completeness, which is defined as for every quantum state $|w\rangle$ and every $b_1, \ldots, b_t \in \{0, 1\}$, the openings $\mathsf{Open}(\rho, b_1, \ldots, b_t)$ should be distributed identically as if we directly measure $|w\rangle$ according to the basis specified by $b_1, \ldots, b_t$. The commitment scheme should also satisfy the security requirement: suppose the cheating prover P^* produces some arbitrary openings, then these openings should be consistent with applying the measurements (based on $b_1, \ldots, b_t$) directly on some quantum state $|w^*\rangle$.

Why does Mahadev's protocol work? In [Fitzsimons-Hajdusek-Morimae 18], they have shown that any quantum polynomial time computation can be converted to an "X/Z witness" $|w\rangle$, which is essentially a long quantum state. Then the correctness of the quantum computation can be verified by randomly measuring each qubit in the X basis or Z basis. Suppose the quantum circuit that decides L has $|C|$ gates, then the conversion produces a witness $|w\rangle$ of length t, which is slightly greater than $|C|$. Then we can take this witness $|w\rangle$, and apply cryptography to commit it to a classical string. Note that if $x \notin L$, there exists no valid X/Z witness $|w\rangle$ that proves $x \in L$; however, the security property of the commitment scheme guarantees that the openings need to be consistent with some witness $|w^*\rangle$, so the verifier will reject since $|w^*\rangle$ will not be consistent with x. Note that this protocol is not succinct because the communication complexity grows linearly with the number of gates in the quantum circuit C.

To obtain succinctness from the protocol description above, since the messages communicated is classical, it is natural to think of classical techniques to shrink the length of messages exchanged. For example, the commitment y from the prover can be shrinked using a Merkle tree. The random bits $b_1, \ldots, b_t$ from the verifier to the prover can be shrinked using a pseudorandom generator. However, the above description of Mahadev's protocol is overly simplified. In reality (see Figure 4), the prover commits $|w\rangle$ qubit by qubit, and for each qubit, the verifier first sends to the prover an independent public key (for the commitment scheme). When the verifier obtains the openings, it uses the private key for each qubit to decode the measurements from the openings. Therefore, the communication bottleneck is the total size of the public keys for all qubits in $|w\rangle$, which is challenging to shrink. Note that it is insecure to naively use a single pair of public key and private key for all qubits, as each opening of a measurement to a qubit can leak some information about the private key. [Gunn-Kalai-Natarajan-Villanyi 25] manages to solve this problem. It cleverly use a single public key for the commitment scheme, thus obtaining a succinct classical commitment to quantum states. As a corollary, they also obtain a quantum-classical succinct argument system for **QMA** assuming LWE.

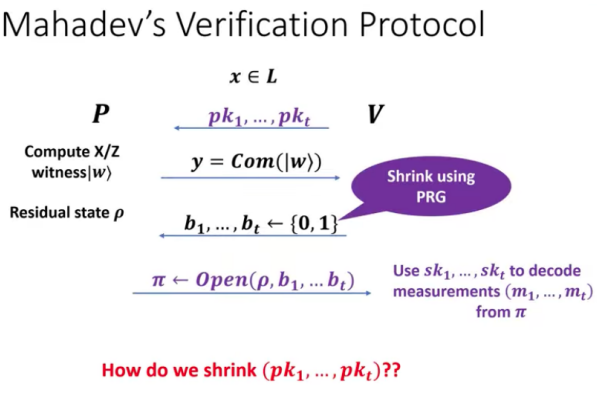

Fig. 4. Mahadev's Protocol, in Reality