UCSD Summer school notes

Interactive proofs for quantum computation

1 Introduction

As you all well known by now, it is generally believed that BQP does *not* contain NP-complete languages. But it is also believed that BQP is not *included* in NP either. While some of the most famous problems in BQP, such as factoring, do lie in NP, Bennett et al. [BBBV97] showed that BQP is *not* in NP, *relative to a random oracle* (we won't explain what this means here — the strength of such "evidence" is a debated topic). In fact there is even fairly strong evidence, based on some form of generalization of the Bernstein-Vazirani algorithm, that BQP is not included in the polynomial hierarchy [Aar10].

The fact that BQP is not (believed to be) in NP implies that in general we do not expect there to exist classically verifiable proofs for the correctness of an arbitrary quantum computation. This poses a challenge: as we see quantum computers emerging, how will we test their predictions? This is a practical problem — will anyone trust the "quantum cloud" — but also a philosophical one — is quantum mechanics a testable theory? For more on this, see [AV13].

Not all is lost. What we *do* know, is that BQP is included in PSPACE. And PSPACE $=$ IP. So all languages in BQP have *classical* interactive proofs, with an efficient classical client! Unfortunately, there is a major caveat to this observation. The proof that PSPACE is in IP is based on the classical SUM-CHECK protocol, which in general requires the server to execute PSPACE-complete computations (basically, the server has to compute exponentially large sums in order to determine answers that will satisfy the client). So, even though a protocol exists, it is unknown if there is such a protocol in which a honest server is only required to have the power of BQP. Today this is a major open question:

Open Question 1. Do all languages in BQP have single-server interactive proofs, in which the client has the power of BPP, and for which completeness holds with a BQP server and soundness holds against any server?

There is active research around this question. There are some partial impossibility results [ACGK17], as well as indication that the problem may have a solution if one is willing to make computational assumptions [Mah17] (e.g. that the BQP server cannot break the learning with errors (LWE) problem from crypography). In this lecture we will discuss the following two scenario in which the question is known to have a positive answer:

1. The client is allowed limited quantum server, such as the ability to prepare single qubits in arbitrary states and send them to the server;

2. The client is allowed to interact with multiple quantum servers sharing entanglement.

The question as formulated above asks for *verifiable* delegation: given a quantum circuit (deciding some BQP language L), is there a protocol that allows a classical client to extract the outcome of the circuit from a BQP server, in a way that any cheating server, attempting to convince the client of the wrong outcome, will be detected? A second desirable property of a delegation protocol is *blindness*: while the user would like to learn the valid outcome of her circuit, she might not want to disclose the particular circuit or input she is interested in to the server. This is a distinct poperty than verifiability; in particular, one may ask for blindness in the "honest-but-curious" model, where verifiability is trivial. The following definition introduces these properties slightly more formally.

Definition 1 (Delegated computation)**.** In the task of delegated computation, a client (sometimes called the *verifier*) has an input $(x, \mathcal{C})$, where x is a classical string and $\mathcal{C}$ the classical description of a quantum circuit. The client has a multiple-round interaction with a quantum server (sometimes also called *server*). At the end of the interaction, Alice either returns a classical output y, or she aborts. A protocol for delegated computation is called:

- *Correct* if whenever both the client and the server follow the protocol, with high probability Alice accepts (she does not abort) and $y = \mathcal{C}(x)$. (This property is sometimes called *completeness*.)
- *Verifiable* if for any server deviating from the protocol, the client either aborts or returns $y = \mathcal{C}(x)$. (This property is sometimes called *soudness*.)
- *Blind* if for any server deviating from the protocol, at the end of the protocol the server has no information at all about the client's input $(x, \mathcal{C})$.

The definition remains rather informal. For example, how should we formalize the "information" that the server has at the end of the computation? This can be rather delicate, especially once one starts taking into account a small chance ε of deviation from the perfect properties. A precise definition satisfying all the desired properties (universal composability in particular) would take us many pages. Such a definition was given using the framework of *abstract cryptography* in [DFPR14].

The informal definition will be sufficient for our purposes. Note that in spite of being rather similar neither of the properties of verifiability or blindness is known to directly implies the other. In practice it will often be the case that verifiability follows from blindness by arguing, using "traps", that if a protocol is already blind, the server's trustworthiness can be tested by making it run "dummy" computations for which Alice already knows the output, without the server being able to distinguish whether it is asked to do a real or dummy computation. We will see an example of this technique later on.

Open Question 2. Is there a general transformation from any protocol satisfying blindness, to a protocol satisfying both verifiability and blindness? See [KMW17] for a possible approach.

Remark 2. The problem of delegating computation is interesting even for classical computation. In this case the client herself could directly execute the classical circuit $\mathcal{C}$. But it makes sense to be even more demanding, and seek protocols where the client is super-efficient: the best we could hope for is a client that runs in time *linear* in the input length, and independent of the size of the circuit. In addition, we would like the overhead for the server to be as small as possible, so that the honest behavior requires a server effort of the same order as the size of the circuit, $|\mathcal{C}|$. This kind of interactive proofs are called *doubly efficient* interactive proofs [GKR08]. The paper [RRR16] shows how to achieve such proofs with client runtime that is linear in the input length, polynomial in the space required by $\mathcal{C}$, and polylogarithmic in $|\mathcal{C}|$. If one is willing to make computational assumptions (essentially, subexponential LWE) then even more efficient delegation is possible [KRR14], with client runtime that is linear in the input size and poly-logarithmic in $|\mathcal{C}|$.

These results usually do not put emphasis on the requirement of blindness: they focus on verifiability alone. One reason for this is that blindness is "trivially solved" by employing homomorphic encryption [Gen09]. This, however, requires computational assumptions, and induces significant computational overhead.

Resources. A great recent survey on quantum delegated computation is [GKK17]. See also the accompanying "Week 10" notes on delegated computation.

2 Computing on encrypted quantum data

The question of delegating quantum computation to a quantum server, by a client with limited quantum abilities, is first considered by Childs [Chi01]. The basic idea is already present in the proof of the Gottesman-Knill theorem [Got98]. Recall that this theorem shows that circuits made only of Clifford gates can be simulated in classical polynomial time. The idea for the proof of the theorem is to keep track of a quantum state through its *stabilizers*, i.e. the subgroup of Pauli operators such that the state is the unique common eigenvalue-1 eigenvector. The reason this is possible is the definition of Clifford gates, which by definition conjugate Pauli operators to Pauli operators.

Childs' protocol for delegation achieves blindness by hiding the input using the quantum one-time pad, a quantum analogue of the classical one-time pad, that we review next.

2.1 The quantum one-time pad

You are all familiar with the classical one-time pad (OTP): given a secret key $k \in \{0,1\}^n$, any message $m \in \{0,1\}^n$ can be encrypted with perfect (information-theoretic) security as $E_k(m) = m \oplus k$, where the XOR is taken bitwise. Decryption is the same operation, $D_k(c) = m \oplus k$.

Here is another way to think about the one-time pad. Write the classical message as a quantum state in the computational basis, $|m\rangle = |m_1\rangle \cdots |m_n\rangle$. The encryption is $E_k(|m\rangle) = |k_1 \oplus m_1\rangle \cdots |k_n \oplus m_n\rangle$. So E_k is just the unitary operation $E_k = \sigma_X^{k_1} \otimes \cdots \otimes \sigma_X^{k_n}$, an operation we will denote as $\sigma_X(k)$.

Now let's find out what is the quantum mechanical representation of the whole system after encryption, including not only the encrypted message but also the user's choice of a random key. We can think of the user as first creating a uniform superposition over keys, $2^{-n/2} \sum_{k\in\{0,1\}^n} |k\rangle$, in a private register. Then the user can create a copy of the key in a register used to implement the encryption algorithm: $2^{-n/2} \sum_{k\in\{0,1\}^n} |k\rangle|k\rangle$. Note that if the first register is traced out, the second one is in a uniform mixture, corresponding to a uniformly random choice of key.

Next the user writes down the message $|m\rangle$, and applies the encryption operation, yielding the joint state

$$\frac{1}{\sqrt{2^n}} \sum_k |k\rangle|k\rangle|E_k(m)\rangle = \frac{1}{\sqrt{2^n}} \sum_k |k\rangle|k\rangle|m \oplus k\rangle \ .$$

The key(!) observation here is that, from the point of view of someone not having access to any of the key registers, the state of the system is represented by its reduced density matrix, which is

$$\frac{1}{2^n} \sum_k |k \oplus m\rangle\langle k \oplus m| \ = \ \frac{1}{2^n} \mathrm{Id} \ ,$$

the totally mixed state. This state is completely independent of the message m, which justifies the claim that the classical one-time pad provides information-theoretic security.

Now, the quantum one-time pad achieves exactly the same effect, except that it works for quantum states. Overve that if $|m\rangle$ is a quantum state such as $|+\rangle$, then $\sigma_X|+\rangle = |+\rangle$, so the encryption leaves the message invariant: the classical one-time pad is completely insecure when used with quantum messages. To avoid this issue the idea is to apply the one-time pad not only in the computational basis, but also in the Hadamard basis: the quantum one-time pad E_k uses a key $k = (a, b)$ and it maps

$$|\psi\rangle \mapsto E_k|\psi\rangle = \sigma_X(a)\sigma_Z(b)|\psi\rangle = \big(\sigma_X^{a_1} \otimes \cdots \otimes \sigma_X^{a_n}\big)\big(\sigma_Z^{b_1} \otimes \cdots \otimes \sigma_Z^{b_n}\big)|\psi\rangle \ .$$

To show that this works we must extend the calculation above to arbitrary quantum states. This is done in the following exercise.

Exercise 3. Show that the 2^{2n} tensor products of the Pauli σ_X and σ_Z matrices form an orthonormal basis for the complex vector space of $2^n \times 2^n$ matrices, equipped with the trace inner product. By expanding an arbitrary n-qubit density matrix ρ in this basis, deduce that

$$\frac{1}{2^{2n}} \sum_{k=(a,b)} E_k(\rho) = \frac{1}{2^n} \mathrm{Id} \ .$$

2.2 The basic protocol

We start with a basic protocol in which the server essentially acts as a giant memory for the client. Given an input $x \in \{0,1\}^n$, the client encrypts $|x\rangle$ using a quantum one-time pad as $|\tilde{x}\rangle = \sigma_X(a)\sigma_Z(b)|x\rangle$ where $a, b \in \{0,1\}^n$ and sends $|\tilde{x}\rangle$ to the server. Note that in case x is a string of bits then $|\tilde{x}\rangle$ has a classical description; in fact it is just a classical one-time padded version of the classical string x, since the σ_Z operators act as identity on computational basis states.

Next the client proceeds through the gates $C_1, \ldots, C_m$ of her circuit, one at a time. For each gate C_i, the client requests the (at most two) qubits that the gate acts on from the server. The client undoes the one-time pad on those qubits, applies the gate, and inserts a new one-time pad. She sends the qubits back to the server and proceeds with the next gate.

This simple scheme is blind, because to any party not in possession of the OTP keys, any qubit sent from the client to the server is uniformly mixed and uncorrelated with any other qubit present in the protocol. (This is why we insert new keys at each step; to make this argument obvious.)

In the next two sections we propose two improvements to the basic scheme. The first is due to Childs. It gives a protocol that remains blind, but in which the client only has to hold at most two qubits at a time, and moreover only has to perform single-qubit Pauli operations. The second is due to Aharonov et al., and it builds on the basic scheme by incorporating authentication, to achieve a protocol that is verifiable.

2.3 Simplifying the client: the Childs protocol

The basic idea behind the Childs protocol is the following observation. Suppose first that the circuit $\mathcal{C}$ chosen by the client consists only of Clifford gates. Suppose that, instead of returning the qubits to the client as in the basic scheme, the server executes the circuit $\mathcal{C}$ directly on the encrypted input $|\tilde{x}\rangle$, measures the output qubit in the computational basis, and returns the outcome to the client. Is this of any use?

Here is how the client can recover her plaintext output. Given a classical description of $\mathcal{C}$ as a sequence of Clifford gates, the client classically computes n-bit strings a' and b' such that $\mathcal{C}\sigma_X(a)\sigma_Z(b) =$

$\sigma_X(a')\sigma_Z(b')C$. Such strings exist by virtue of our assumption that the circuit is a Clifford circuit, and they can be computed efficiently by going through the circuit one gate at a time.

Exercise 4. Compute a' and b' for the case of σ_X, H, and $CNOT$.

At the end of its computation the server will return a bit that is 1 with probability exactly

$$|\langle 1|\mathcal{C}\sigma_X(a)\sigma_Z(b)|x\rangle|^2 = |\langle 1|\sigma_X(a')\sigma_Z(b')\mathcal{C}|x\rangle|^2 = |\langle 1\oplus a'_1|\mathcal{C}|x\rangle|^2 ,$$

where for the second equality we used that σ_Z acts as identity on computational basis states, and that only the first qubit is measured. In other words, if the client determines that $a'_1 = 0$ then she can directly use the value returned by the server as the correct outcome of the computation, and if $a'_1 = 1$ then she can easily determine the outcome by flipping the value reported by the server. From the point of view of the server a'_1 is a uniformly random bit, and so no information is leaked.

This gives us a scheme for blind delegation of Clifford circuits. To delegate arbitrary circuits we need to allow circuits that include an additional one-qubit gate, such as the T gate, which together with Clifford gates makes for a universal gate set. Unfortunately the T gate does not conjugate Paulis to Paulis (of course, since it is not a Clifford gate!). In particular, while $T\sigma_Z = \sigma_Z T$, with respect to σ_X we have the following relation, for $a \in \{0,1\}$:

$$T\sigma_X(a) = \sigma_X(a)S^aT = iS^a\sigma_Z(a)\sigma_X(a)T ,$$

where $S = T^2$. This means that, depending on the current key a, after application of the T gate by the server, it may or may not be necessary to apply an additional phase gate S. To avoid revealing the key a to the server, in Childs' protocol, for every T gate in the circuit the client has a short two-round quantum interaction with the server, as follows:

1. The client requests the qubit on which the T gate has been applied.
2. If $a = 0$, the client keeps the qubit and sends a dummy qubit to the server; if $a = 1$ it sends the same qubit back to the server. To ensure the server cannot tell the difference between the two cases, the client inserts an additional one-time pad, with a fresh key, on the qubit.
3. In all cases, the server applies an S gate and sends the qubit back to the client.
4. Finally, the client returns the original qubit (case $a = 0$) or the same qubit (case $a = 1$) to the server, after having inserted yet another one-time pad.

This gives a blind protocol with a client with limited quantum abilities (it needs to store and operate on a single qubit at a time). Even though the protocol may seem rather simple, and the effort of the client is non-negligible, it was not a priori obvious that a complete quantum computation could be implemented while performing only single-qubit operations, and in a blind way. (Note that our description gives a protocol that is input-blind, but the server needs to be told what the circuit is. This can easily be avoided by requiring the circuit to implement a fixed, universal circuit $\mathcal{U}$, and encoding the actual circuit $\mathcal{C}$ in the input to $\mathcal{U}$. In this case only the size of $\mathcal{C}$, or an upper bound on it, needs to be revealed to the server.)

2.4 Authentication: the ABOE protocol

The main drawback of the Childs protocol is that it is blind, but not verifiable: there is no mechanism for the client to verify that the server does not simply send back completely random qubits. To obtain a protocol that is verifiable, Aharonov et al. [ABOE08] use authentication. The idea is to map each qubit $|\psi\rangle$ to an authenticated state $|\tilde{\psi}\rangle$ on m qubits, where m can be chosen as a function of the security parameter. Aside from this slight twist, the protocol has the same structure as the Childs protocol:

1. The server is sent the (individually authenticated) qubits of the input state, one by one.
2. When a gate is to be applied, the server is asked to return the authenticated state of the qubits on which the gate acts to the client.
3. The client de-authenticates, applies the gate, re-authenticates, and sends the qubits back to the server.

Note that here, contrary to the Childs protocol, the client is asked to apply the gate herself. This may seem like it is a lot of work, but it only requires a constant-size quantum computer, where the constant should be chosen as a function of the security one aims to achieve.

There are two main authentication schemes that have been considered. The first, and simplest, is called the Clifford scheme. This encodes a qubit $|\psi\rangle$ by appending m ancilla qubits in the state $|0\rangle$, then applying a uniformly random Clifford gate C_k on the $(m+1)$ qubits:

$$|\psi\rangle\langle\psi| \;\mapsto\; \mathrm{E}_{C\sim\mathfrak{C}_{m+1}}\big[\, C|\psi\rangle\langle\psi|\otimes|0\rangle\langle 0|^{\otimes m}C^\dagger\,\big]\;,$$

where here the expectation is over a uniformly random C in the $(m+1)$-qubit Clifford group $\mathfrak{C}_{m+1}$. This group has size $O(m^2)$, and we will take it for granted that selecting an element at random can be done efficiently.

The basis for the authentication property of this scheme is provided by the *Clifford twirl*, which is the property expressed in the following exercise.

Exercise 5. Let $P \neq P'$ be Pauli operators. Show that for any density matrix ρ,

$$\sum_{C\in\mathfrak{C}_m} \big(C^\dagger P C\otimes \mathrm{Id}\big)\,\rho\,\big(C^\dagger P' C\otimes\mathrm{Id}\big) \;=\; 0\,,$$

where $\mathfrak{C}_m$ is the group of m-qubit Clifford unitaries.

To formally define authentication requires a little work, so we won't give the full definition here. Informally, the requirement is that for any operation T on the encoded state $E_k(|\psi\rangle\langle\psi|)$, it holds that, either the decoding procedure aborts with non-negligible probability (the last m qubits are not found in the state $|0\rangle\langle 0|^{\otimes m}$ after having inverted the Clifford unitary) or the state recovered is close to the initial state $|\psi\rangle$. The Clifford twirl is used in the proof of the authentication property to show that, if the the map that results from the "attack" T, conjugated by a random Clifford unitary (corresponding to the encoding and decoding maps) has an action on the plaintext space $|\psi\rangle\langle\psi|\otimes|0\rangle\langle 0|^{\otimes m}$ that decomposes as a uniform mixture of "Pauli attacks", which correspond to the application of a Pauli operator on the $(m+1)$ qubits. But a uniform mixture of Pauli attacks is easy to detect: with probability $1-2^{-m}$, this will send the ancilla $|0\rangle$ to an orthogonal state, which is detected by the decoding procedure.

In [ABOE08] the authors introduce a second authentication scheme, called the *polynomial code*. Using this authentication code, they give a delegation protocol where the interaction required to apply a gate is only classical. (The client still needs to be quantum, because she needs to prepare the authenticated qubits in the first place.) The main property used for the analysis is that the polynomial code allows transversal application of Clifford gates (just like for the non-authenticated quantum one-time pad). For the Toffoli gate, it is a bit more complicated, but it can be done by giving the server some additional ancilla states, called magic states, and initiating a classical interaction between the client and the server.

With the idea of authentication in place, there is still a lot of work required to complete the analysis. The goal is to argue that an arbitrary deviating action by the server, either has no noticeable effect on the outcome of the circuit (as decoded by the client), or is detected by the client. The interactive nature of the protocol makes the analysis a little delicate, and we refer to the paper [ABOEM17] for details.

Open Question 3. Can the protocol be made fault-tolerant? We can always ask the server to implement a fault-tolerant circuit. But what about the client? Protecting the qubits she prepares using a sufficiently good code would seem to require a message size that scales with the number of qubits in the circuit.

3 Fitzsimons-Morimae protocol

So far we've seen one blind protocol and one blind and verifiable protocol. If we only require verifiability, but not blindness, there is a very simple protocol based on the circuit-to-Hamiltonian construction that we saw earlier.

Note in the case of a classical circuit the solution is completely trivial: the server can perform the whole computation, and send a tableau representing the values of all wires in the circuit, at all time steps. The client can check this efficiently by verifying that each gate has been propagated correctly. Of course, she might as well have performed the computation herself.

In the quantum case, as you know, the situation is more subtle. But we've seen a solution: the natural idea, proposed by Fitzsimons and Morimae [MF16], is to ask the server to prepare a history state for the computation. The server can do this efficiently if it has a universal quantum computer.

So suppose the server has prepared the history state for the 5-local construction. The corresponding Hamiltonian $H = \sum_j H_j$ can be efficiently computed, in classical polynomial time, from a description of the circuit. The protocol then proceeds as follows:

1. The client secretly selects one 5-local term H_j uniformly at random.
2. The server sends the qubits of the history state to the client, one at a time.
3. The client immediately discards any qubit it receives, *except* if it is a qubit on which the local term H_j acts.
4. Once it has received all qubits on which H_j acts, the client performs the same measurement $\{H_j, \mathrm{Id} - H_j\}$ as the client in the QMA protocol for the local Hamiltonian problem, and accepts if and only if the correct outcome is obtained.

Remark 6. It is possible to make the client even more efficient by further tuning the Kitaev Hamiltonian to be in so-called XZ form. In this case, each of the terms is either of the form $\sigma_X(i)\sigma_X(k)$, or $\sigma_Z(i)\sigma_Z(k)$, for $i, j \in \{1, \dots, n\}$. Therefore the client only needs to store two qubits, and measure them in either the computational or Hadamard bases.

A major drawback of this protocol is that the soundness error is very high: one can easily construct a cheating server that has only an inverse polynomial chance of being detected by the client. In order to achieve a constant completeness-soundness gap the whole protocol needs to be repeated a polynomial number of times, so in the end the client has to make a polynomial number of measurements (though each of them still is only a single-qubit σ_X or σ_Z measurement, and the client only ever needs to hold a constant number of qubits at a time).

In the classical case this kind of protocol can be made very efficient by choosing an encoding of the witness (recall this is just a tableau for the computation) using the PCP theorem [BFLS91]. In this case the proof remains of polynomial size in the computation, but verification can be accomplished in polylogarithmic time (given access to a suitably encoded form of the input), by making only a constant number of queries to entries of the proof.

In subsequent lectures we will develop techniques that bring us closer to a similarly efficient verifiation of quantum proofs.

Exercise 7. Show, under appropriate complexity-theoretic assumption, that there is no constant-round protocol for BQP of the form above, where communication is purely classical. *[Hint:* $\mathrm{IP}[k] \subseteq \Sigma_r^p$].

References

[Aar10] Scott Aaronson. Bqp and the polynomial hierarchy. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 141–150. ACM, 2010.

[ABOE08] Dorit Aharonov, Michael Ben-Or, and Elad Eban. Interactive proofs for quantum computations. *arXiv preprint arXiv:0810.5375*, 2008.

[ABOEM17] Dorit Aharonov, Michael Ben-Or, Elad Eban, and Urmila Mahadev. Interactive proofs for quantum computations. *arXiv preprint arXiv:1704.04487*, 2017.

[ACGK17] Scott Aaronson, Alexandru Cojocaru, Alexandru Gheorghiu, and Elham Kashefi. On the implausibility of classical client blind quantum computing. *arXiv preprint arXiv:1704.08482*, 2017.

[AV13] Dorit Aharonov and Umesh Vazirani. *Is quantum mechanics falsifiable? A computational perspective on the foundations of quantum mechanics*. Computability: Turing, Gödel, Church, and Beyond. MIT Press, 2013.

[BBBV97] Charles H Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM journal on Computing*, 26(5):1510–1523, 1997.

[BFLS91] László Babai, Lance Fortnow, Leonid A Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 21–32. ACM, 1991.

[Chi01] Andrew M Childs. Secure assisted quantum computation. *arXiv preprint quant-ph/0111046*, 2001.

[DFPR14] Vedran Dunjko, Joseph F Fitzsimons, Christopher Portmann, and Renato Renner. Composable security of delegated quantum computation. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 406–425. Springer, 2014.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, volume 9, pages 169–178, 2009.

[GKK17] Alexandru Gheorghiu, Theodoros Kapourniotis, and Elham Kashefi. Verification of quantum computation: An overview of existing approaches. *arXiv preprint arXiv:1709.06984*, 2017.

[GKR08] Shafi Goldwasser, Yael Tauman Kalai, and Guy N Rothblum. Delegating computation: interactive proofs for muggles. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 113–122. ACM, 2008.

[Got98] Daniel Gottesman. The heisenberg representation of quantum computers. *arXiv preprint quant-ph/9807006*, 1998.

[KMW17] Elham Kashefi, Luka Music, and Petros Wallden. The quantum cut-and-choose technique and quantum two-party computation. *arXiv preprint arXiv:1703.03754*, 2017.

[KRR14] Yael Tauman Kalai, Ran Raz, and Ron D Rothblum. How to delegate computations: the power of no-signaling proofs. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 485–494. ACM, 2014.

[Mah17] Urmila Mahadev. Classical homomorphic encryption for quantum circuits. *arXiv preprint arXiv:1708.02130*, 2017.

[MF16] Tomoyuki Morimae and Joseph F Fitzsimons. Post hoc verification with a single prover. *arXiv preprint arXiv:1603.06046*, 2016.

[RRR16] Omer Reingold, Guy N Rothblum, and Ron D Rothblum. Constant-round interactive proofs for delegating computation. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 49–62. ACM, 2016.