The Quantum Cut-and-Choose Technique and Quantum Two-Party Computation*

Elham Kashefi,[1, 2] Luka Music,[2] and Petros Wallden[1]

[1] *University of Edinburgh, School of Informatics*
[2] *UPMC - LIP6, Departement Informatique et Reseaux*

I. INTRODUCTION

Secure two-party quantum computation, or 2PQC, involves the computation of a function by two distrustful parties using a quantum computer and was first examined in [1] for a quantum honest-but-curious adversaries (called specious) and later made secure against more malicious adversaries in [2]. The latter protocol, did not use any of the standard boosting classical techniques, but instead used a stepwise quantum authentication protocol, where two-ways online quantum communication was required. Both protocols use extra classical cryptographic primitives, which in the case of the malicious [2] is a full actively secure *classical* two-party computation primitive.

The use of classical techniques for boosting security (such as Cut-and-Choose) for quantum protocols is complicated not only because specific care is needed when defining quantum analogues but also for technical reasons since the rewinding method for proving security cannot be directly used in quantum protocols (as demonstrated in [3] and [4]).

Our Contribution

1. We introduce a Quantum Computation Cut-and-Choose (QC-CC) technique. Application of this technique is possible because of the decomposition of quantum computation into a classical control and a quantum resource in the MBQC models. Our precise security analysis shows that, contrary to popular belief, the security of the classical Cut-and-Choose is impacted by quantum adversaries.

2. We give a protocol for 2PQC with classical input and output secure against "quantum covert" adversaries, similar to classical covert adversaries [5]. Our protocol, built by applying QC-CC on [6] (secure against weak specious adversaries), resembles the protocol by Yao [7] (e.g. asymmetry between the two parties) and [8] where Yao's protocol is boosted using the classical Cut-and-Choose.

3. A key obstruction when using classical techniques is that in general rewinding the *quantum* adversary during the simulation is *not* possible. There are two known cases where rewinding can be used for quantum adversaries: Watrous' oblivious rewinding [3] and Unruh's special rewinding [4]. We adapt and use both methods to construct the simulators and prove the security of our protocol. This is one of the few protocols in which quantum rewinding is explicitly used and the only one, to our knowledge, that combines both techniques for a quantum/classical protocol, providing the first example of how to apply them concretely outside of the ZKPoK context.

4. [1, 2] describe 2PQC protocols for symmetric parties. Our protocol crucially differs: (i) There is only one-way offline quantum communication between the parties, (ii) only one party needs involved quantum technological abilities, the other only prepares offline single qubits, (iii) minimal classical cryptographic primitives are required (oblivious transfer and quantum-safe commitments).

II. PRELIMINARIES AND SECURITY DEFINITIONS

A. Standard Definitions of Security for 2PQC

We prove security using the *real/ideal simulation paradigm* (stand-alone security): when considering a party as corrupted, we will construct a simulator interacting with the ideal functionality such that they are not able to detect that they are not in fact interacting directly with a real world honest party instead. The standard definition of security means that the simulated and real states are exponentially close and so indistinguishable for the malicious adversary.

We also introduce a new adversarial model for quantum protocols, based on the covert adversaries in [5]. The quantum covert adversaries are also able to deviate arbitrarily from the protocol. The difference with malicious adversaries is that when they cheat they are caught with high probability but not necessarily exponentially close to 1. This models real world situations where getting caught might have dire consequences for the parties, eg. financial repercussions used as a deterrent against cheating.

Another property which a quantum protocol may satisfy is *verifiability*. This intuitively means that the probability of receiving a corrupted output without aborting is negligible.

During our protocol we will use bit commitment and 1-out-of-2 oblivious transfer (OT).

Bit commitment consists of two phases, Commit and Reveal, such that after the Commit the receiver has no information about the value that has been committed (hiding), while during the Reveal the sender cannot reveal a value different from the one committed previously (binding). Both these properties can be either computationally or unconditionally verified depending on the scheme (but not both unconditionally). We suppose that all the commitments used verify the *strict binding* property of [4].

A 1-out-of-2 oblivious transfer is a two party functionality in which one party (P_1 in our case) has two strings (x_0, x_1) and the other (P_2) has a bit $b \in \{0, 1\}$. At the end of the protocol P_2 recovers x_b. P_1 should not know which of the strings P_2 has chosen while P_2 has no information about the string they did not choose x_{1-b}.

B. Verifiable Blind Quantum Computation

We use the MBQC [9] model, equivalent to the circuit model as it is based on the gate teleportation principle.

* A version of this work has been submitted for publishing at CRYPTO 2017

One starts with a large, generic entangled state (represented by a graph) and, by choosing suitable single qubit measurements, can perform any quantum computation (circuit). The computation is fully characterised by the graph and default measurement angles.

We will consider a client-server setting. The client (P_1) can prepare single qubits while the server (P_2) can perform any general quantum computation. The client sends qubits in the $|+\rangle$ state and the server entangles them according to a computation graph with controlled$-Z$ gates between qubits corresponding to adjacent vertices on the graph, resulting in a *graph state* [10]. The computation is defined by a default measurement angle ϕ_i (depending only on the desired computation). It is carried out by having the server measure single qubits. The actual angle of each measurement depends on ϕ_i and the outcomes of previous measurements. The client performs these classical calculations to adjust the angle and therefore the server returns to the client the result of each measurement [11].

The computation can be totally hidden from the server: if instead of sending $|+\rangle$ states the client chooses at random and sends $|+_\theta\rangle = 1/\sqrt{2}(|0\rangle + e^{i\theta}|1\rangle)$ with $\theta \in \{i \cdot \pi/8\}_{i\in\{0,\ldots,7\}}$ then measuring the qubits in a similarly rotated basis has the same result as the initial non-rotated computation. By keeping the angle hidden, the server is blind as to what computation is being performed. To ensure that no information is leaked from the measurement outcome, we add another parameter r_i for each qubit, which One-Time-Pads the measurement outcome. The client sends rotated qubits (to become the resource state once entangled) and then guides the computation with a set of classical instructions. This in turn leads to the desired blind computation. This idea was formalised in the *universal blind quantum computation* (UBQC) protocol in [11].

In UBQC the server is not forced to follow the instructions and the client cannot verify if the computation is done correctly. This can be achieved by including trap qubits at positions unknown to the server. These are isolated qubits that do not affect the computation and have a deterministic outcome if measured in the correct basis. They can therefore be used as traps: a client can easily detect if one of them has been measured incorrectly but the server is ignorant of their position in the graph. The result is the Verifiable Universal Blind Quantum Computation Protocol (VUBQC) of [12].

Theorem 1 (Verifiability of VUBQC, taken from [12])**.** *The VUBQC Protocol is ϵ_2-verifiable for the Client for any ϵ_2.*

III. QUANTUM REWINDING

Classically the simulator runs the adversary internally and rewinds it by having black box access to the *next message* function. The simulator has to save all messages so that it can send them again later to get a potentially different reply (rewinding), which is impossible in the quantum setting due to no-cloning. We present two techniques given in [3] and [4] which achieve a similar result, with different constraints, show that they are applicable for the simulators of our protocol and calculate the success probability for both cases (see full version of the paper [13]).

Watrous' Oblivious Quantum Rewinding Lemma 8 from [3] states that rewinding is possible if the probability that the simulation is successful is non-negligible and independent of the internal state of this adversary. Rewinding gives a state ϵ-close to that of a successful simulation for any exponentially small ϵ with polynomially many rewinds.

Unruh's Special Quantum Rewinding Watrous' lemma only ensures that the simulation is successful, but *no information* is kept between two rewinds (hence *oblivious rewinding*). In the simulator for covert client we will need two transcripts in order to recover their input (otherwise secret), so another type of rewinding is necessary.

Let Π be a protocol between P_1, with input (x, w), and P_2, with input x and output in $\{0, 1\}$, with three messages: commitment com by P_1, challenge ch sampled uniformly at random by P_2 from a set C_x, and response $resp$ by P_1. P_2 accepts (outputs 0) by a deterministic poly-time computation on $(x, com, ch, resp)$.

Rewinding can be used in such protocol by satisfying two extra conditions : strict soundness and special soundness. Strict soundness means that there is a unique classical response to each challenge. Special soundness means that, given two different accepting transcripts, the simulator can recover the witness w, supposed to be secret.

IV. THE QUANTUM CUT-AND-CHOOSE TECHNIQUE

The Cut-and-Choose method is a standard technique to boost a protocol secure against honest-but-curious to being secure against malicious adversaries.

The client creates s copies of the graph and the server chooses which ones (the *check graphs*) they will check for consistency. If the checks pass and additional precautions are taken, the server is confident that with high probability the remaining graphs (the *evaluation graphs*) were also constructed correctly and can be used for the computation. Here we will have s graphs in total, $s - 1$ *check graphs* and 1 *evaluation graph.*

We extend this technique to quantum computations. We show how to verify quantum states using *Quantum-State-Preparation Cut-and-Choose.* This ensures that the resource state for the quantum computation in VBQC is constructed correctly. Secondly, we define *Classical Instructions Cut-and-Choose*, using the classical Cut-and-Choose to verify that the (classical) instructions for the computation are correct. Finally, we combine the two to get *Quantum Computation Cut-and-Choose.*

Quantum State Preparation Cut-and-Choose (QSP-CC) This functionality allows the receiver to test that a state $|\psi_\alpha\rangle$ was prepared correctly, up to a certain probability, without the sender revealing its classical description. The client sends s states along with commitments to their classical descriptions. The server chooses one of them (α) and the client reveals the commitments for all $i \neq \alpha$. The server measures the states $i \neq \alpha$ according to the decommitted values and verifies that they are correct.

The state at the end is $\frac{1}{\sqrt{s}}$-close to the correctly-prepared one. Note that if we used more than 1 evaluation graphs (as needed for boosting the success probability), the probability of successful cheating does not scale linearly with parallel repetitions of QSP-CC due to coherent (entangled) attacks.

Classical Instructions Cut-and-Choose (CI-CC) To perform the VBQC protocol, even if the resource state is correct, one needs to ensure that the classical instructions are also correct.

To achieve this, the sender commits to the classical instructions for all states after sending the qubits to the receiver. When the commitments are opened, the receiver can deterministically decide if the instructions are correct (w.r.t. the classical description of the state). Intuitively we expect that such a classical Cut-and-Choose has a $\frac{1}{s}$ probability of failure but it is in fact also $\frac{1}{\sqrt{s}}$, due to the special rewinding of [4].

Quantum Computation Cut-and-Choose (QC-CC) The receiver can use the remaining committed instructions to drive the computation by asking the sender to open the instructions corresponding to the measurement outcomes. By combining CI-CC with QSP-CC and using the commitments during the (quantum) computation, the server knows (with high probability) that they have been asked to perform the correct quantum computation.

From Protocol QSP-CC, the state is $\frac{1}{\sqrt{s}}$-close to the ideal state. From Protocol CI-CC, the instructions are constructed correctly up to probability $\frac{1}{\sqrt{s}}$. It follows that the computation is performed correctly up to probability $O(\frac{1}{\sqrt{s}})$.

The proof of security requires the simulator to rewind the simulation using Unruh's special rewinding technique in order to extract the secret parameters of the client (the description and instructions of the evaluation graph), which results in the quadratic cost for the security. This protocol follows the $(com, ch, resp)$ structure of [4] (com corresponds to sending commitments, ch is the server choosing the evaluation graph and $resp$ is the revealing of commitments). It also verifies strict and special soundness (the only acceptable response is to open the correct commitments and given two transcripts we can recover the secret), therefore the use of rewinding is justified.

This provides an example where proving security against a quantum adversary is hard, even for a classical functionality. The part of the protocol that needs rewinding is entirely classical (in happens after sending the qubits) and the same proof (and extra cost) is necessary even for a fully classical CC protocol (with a single evaluation graph). It is not sufficient to use cryptographic primitives resistant against quantum computers (eg. based on LWE), but proof techniques (and security parameters) should also be modified.

V. THE 2PQC PROTOCOL

High-level overview P_1 and P_2 have already chosen a VUBQC graph computing the function fault-tolerantly. P_1 chooses and commits to the randomness for the s versions of the graph (the angles θ and the flips r) and also to all the corresponding corrected measurements angles, the input measurement angles for both parties, the decryption keys for P_2's output and the positions of the traps among P_2's output qubits.

For every input bit of P_2, they perform a 1-out-of-2 OT at the end of which P_2 learns the measurement angles for their input qubits for all graphs (doing the OTs before sending the qubits is essential for the security proof).

They then perform a QC-CC protocol: the qubits of each graph are the states, the commitments correspond to the descriptions of the states and the instructions, they choose the evaluation graph with a coin-tossing protocol, P_1 reveals the commitments of check circuits and P_2 verifies them as well as the states.

Then they perform the evaluation with the VUBQC protocol with P_1 decommitting to the instructions (measurement angles).

At the end they perform a simple key-exchange protocol so that P_2 may decrypt their output.

Theorem 2 (Correctness). *If both parties are honest and follow the steps of the protocol then the protocol is correct.*

Proof Sketch. The parties are honest, all the checks pass and there is no abort. The evaluation is equivalent to the normal VUBQC, with the server keeping part of the output. The last step of the protocol allows P_2 to decrypt it. The correctness directly follows from the correctness of VUBQC. □

Lemma 1 (ϵ-verifiability for the client). *The QYao 2PQC Protocol is ϵ_2-verifiable for P_1, for the same ϵ_2 as the VUBQC Protocol.*

Proof Sketch. If P_1 is honest and P_2 malicious, the client generates correct graphs and commitments. The commitments that are opened do not reveal anything about the evaluation graph and the subsequent evaluation follows exactly as in the VUBQC. This protocol thus inherits the verifiability of VUBQC. □

Theorem 3. *Let s the number of graphs constructed as part of the CC. If the OT is ϵ_2-private against malicious adversaries, the commitments are perfectly hiding and binding and the protocol is ϵ_2-verifiable for P_1, then it is ϵ_2-private against malicious P_2 and $\frac{1}{\sqrt{s}}$-private against covert P_1.*

Proof Sketch.

The simulator for adversarial P_1 is very similar to the one in the proof for the QC-CC Protocol: it obtains one set of values form a first run (runs as usual until P_1 reveals the commitments) then rewinds the adversary to get a second set and recovers the secret parameters of the adversary which it then sends to the ideal functionality, thus getting the ideal output. The simulator runs the evaluation graph with a random input, encrypts the ideal output using the correct keys (which he knows because of the rewind) and returns it to the adversary.

The simulator for adversarial P_2 relies on the construction of a graph which has deterministic output (this graph is indistinguishable from a correct graph because of the UBQC construction). The simulator recovers the adversary's input with the OTs (we work in an OT-hybrid model, during simulation the simulator replaces the OT ideal functionality and receives the inputs in its place) and sends it to the ideal functionality, from which P_2's ideal output is obtained. It then constructs a graph which always produces this output and hides it among the remaining $s-1$ graphs, which are constructed correctly. The simulator biases the choice of evaluation graph by rewinding the coin-toss so that this special graph is chosen (since *no information is kept between rewinds*, which is used only to pick the fake graph, and the probability of success of the rewinding is $\frac{1}{s}$, independent of the internal state of the adversary, we can use the oblivious quantum rewind technique from [3]). The checks pass and P_2 evaluates the fake graph and gets the correct output. □

[1] F. Dupuis, J. B. Nielsen, and L. Salvail, in *Advances in Cryptology–CRYPTO 2010* (Springer, 2010), pp. 685–706.

[2] F. Dupuis, J. B. Nielsen, and L. Salvail, in *Advances in Cryptology–CRYPTO 2012* (Springer, 2012), pp. 794–811.

[3] J. Watrous, SIAM Journal on Computing **39**, 25 (2009), http://dx.doi.org/10.1137/060670997, .

[4] D. Unruh, *Quantum Proofs of Knowledge* (Springer Berlin Heidelberg, Berlin, Heidelberg, 2012), pp. 135–152, ISBN 978-3-642-29011-4, .

[5] Y. Aumann and Y. Lindell, *Security Against Covert Adversaries: Efficient Protocols for Realistic Adversaries* (Springer Berlin Heidelberg, Berlin, Heidelberg, 2007), pp. 137–156, ISBN 978-3-540-70936-7, .

[6] E. Kashefi and P. Wallden, ArXiv e-prints (2016), 1606.06931.

[7] A. Yao, in *Foundations of Computer Science, 1986., 27th Annual Symposium on* (IEEE, 1986), pp. 162–167.

[8] Y. Lindell and B. Pinkas, *An Efficient Protocol for Secure Two-Party Computation in the Presence of Malicious Adversaries* (Springer Berlin Heidelberg, Berlin, Heidelberg, 2007), pp. 52–78, ISBN 978-3-540-72540-4, .

[9] R. Raussendorf and H. J. Briegel, Phys. Rev. Lett. **86**, 5188 (2001), .

[10] M. Hein, J. Eisert, and H. J. Briegel, Physical Review A **69**, 062311 (2004).

[11] A. Broadbent, J. Fitzsimons, and E. Kashefi, in *Proceedings of the 50th Annual Symposium on Foundations of Computer Science* (IEEE Computer Society, 2009), FOCS '09, pp. 517 – 526, ISBN 0272-5428.

[12] E. Kashefi and P. Wallden, Journal of Physics A: Mathematical and Theoretical; preprint arXiv:1510.07408 (2017), .

[13] E. Kashefi, L. Music, and P. Wallden, Submitted to Crypto 2017 (2017), 1703.03754, .