Quantum one-time programs

(extended abstract)*

Anne Broadbent[1], Gus Gutoski[2], and Douglas Stebila[3]

[1] Institute for Quantum Computing and
Department of Combinatorics and Optimization
University of Waterloo, Waterloo, Ontario, Canada
albroadb@iqc.ca

[2] Institute for Quantum Computing and School of Computer Science
University of Waterloo, Waterloo, Ontario, Canada
gus.gutoski@uwaterloo.ca

[3] School of Electrical Engineering and Computer Science and
School of Mathematical Sciences, Science and Engineering Faculty
Queensland University of Technology, Brisbane, Queensland, Australia
stebila@qut.edu.au

Abstract. A *one-time program* is a hypothetical device by which a user may evaluate a circuit on exactly one input of his choice, before the device self-destructs. One-time programs cannot be achieved by software alone, as any software can be copied and re-run. However, it is known that every circuit can be compiled into a one-time program using a very basic hypothetical hardware device called a *one-time memory*. At first glance it may seem that quantum information, which cannot be copied, might also allow for one-time programs. But it is not hard to see that this intuition is false: one-time programs for classical or quantum circuits based solely on quantum information do not exist, even with computational assumptions.

This observation raises the question, "what assumptions are required to achieve one-time programs for *quantum* circuits?" Our main result is that any quantum circuit can be compiled into a one-time program assuming only the *same basic one-time memory devices* used for classical circuits. Moreover, these quantum one-time programs achieve statistical universal composability (UC-security) against any malicious user. Our construction employs methods for computation on authenticated quantum data, and we present a new quantum authentication scheme called the *trap scheme* for this purpose. As a corollary, we establish UC-security of a recent protocol for delegated quantum computation.

1 Introduction

A *one-time program (OTP)* for a function f, as introduced by Goldwasser, Kalai, and Rothblum [1], is a cryptographic primitive by which a user may evaluate f on only one input chosen by the user at run time. (See also Refs. [2,3] for subsequent improvements.) No adversary, after evaluating the one-time program on x, should be able to learn anything about $f(x')$ for any $x' \neq x$ beyond what can be inferred from $f(x)$. One-time programs cannot be achieved by software alone, as any classical software can be re-run. Thus, any hope of achieving any one-time property must necessarily rely on an additional assumptions such as secure hardware or quantum mechanics: computational assumptions alone do not suffice.

Classically, it has been shown [1,2,3] how to construct a one-time program for any function f using a hypothetical hardware device called a *one-time memory (OTM)*. An OTM is non-interactive idealization of oblivious transfer: it stores two secret strings (or bits) s_0, s_1; a receiver can specify a bit c, obtain s_c, and then the OTM self-destructs so that $s_{\bar{c}}$ is lost forever. OTMs are an attractive minimal hardware assumption: their specification is independent of any specific function f, so they could theoretically be mass-produced.

OTPs are a special form of *non-interactive secure two-party computation* [3], in which two parties evaluate a publicly known function $f(x, y)$ as follows: the *sender* uses her input string x to prepare a *program* $p(x)$ for the *receiver*, who uses this program and his input y to compute $f(x, y)$. A malicious receiver should not be able to learn anything about $f(x, y')$ beyond what can be inferred from $f(x, y)$, for any y'. We use the term "OTP" interchangeably with "non-interactive secure two-party computation".

In this extended abstract we study *quantum one-time programs (QOTPs)*, in which the sender and receiver evaluate a publicly known channel $\Phi : (\mathsf{A}, \mathsf{B}) \to \mathsf{C}$ specified by a quantum circuit acting on registers A (the sender's input), B (the receiver's input), and C (the receiver's output). The security goal is similar in spirit to that for classical functions: for each joint state ρ of the input registers (A, B), a malicious receiver should not be able to learn anything about $\Phi(\rho')$ beyond what can be inferred from $\Phi(\rho)$, for any ρ'.

Can quantum one-time programs be constructed? If so, how? If not, why not, and under what additional assumptions can they be achieved? QOTPs, if they do exist, would be useful for a variety of secure quantum computation tasks, such as providing *copy protection* of software [4] and implementing verification for quantum coin schemes [5]. (Note that QOTPs are different from the task of *program obfuscation*, which is known to be impossible classically [6] but remains an open question quantumly.)

Our main contributions are as follows: (i) We present a universally composable QOTP protocol for *any quantum channel*, assuming only the *same single-bit one-time memories* used in classical OTPs. Our protocol employs *quantum computation on authenticated data (QCAD)*, a technique of independent interest in quantum cryptography. (ii) We present a new quantum authentication scheme called the *trap scheme* and show that it allows for QCAD. (iii) We identify

pathological classes of "unlockable" classical functions and quantum channels that admit trivial OTPs without any hardware assumptions. The remainder of this section elaborates upon these contributions.

1.1 Quantum one-time programs from classical one-time memories

Unlike ordinary classical information, quantum information cannot in general be copied. This no-cloning property prompts one to ask: does quantum information allow for one-time programs without hardware assumptions? (When there are no hardware assumptions, we refer to this as the *plain quantum model*.)

For both classical functions and quantum channels, a moment's thought reveals a negative answer to this question: for any function f or channel Φ, a quantum "program state" for f or Φ can always be re-constructed by a reversible receiver after each use to obtain the evaluation of f or Φ on multiple distinct inputs. Computational assumptions do not help.

Given that one-time programs do *not* exist for arbitrary quantum channels in the plain quantum model, and that one-time programs *do* exist for arbitrary classical functions assuming secure OTMs, we ask: what additional assumptions are required to achieve one-time programs for quantum channels? Our main result answers this question.

Theorem 1 (Main result, informal). *For each channel $\Phi : (\mathsf{A}, \mathsf{B}) \to \mathsf{C}$ specified by a quantum circuit there is a non-interactive two-party protocol for the evaluation of Φ, assuming classical one-time memory devices. The run time of this protocol is polynomial in the size of the circuit specifying Φ and the protocol achieves statistical quantum universal composability (UC-security) against a malicious receiver.*

Since all communication is one-way from sender to receiver, a malicious sender cannot learn anything about the receiver's portion of the input state ρ. The question of security against a malicious sender who tries to convince the receiver to accept an output state other than $\Phi(\rho)$ is left for future work. We restrict our attention to the case of *non-reactive* quantum one-time programs. The more general scenario of *bounded reactive* programs which can be queried a bounded number of times (including the case of an n-use program) may be implemented using standard techniques as is done in the classical case. Most of the components of our QOTP for Φ are independent of the sender's input register A and so can be compiled by the sender before she receives her input. As a corollary of our main result we obtain the UC-security of the protocol for *delegated quantum computations* (DQC) from Ref. [7]. Composable security for other variants of DQC was independently studied in Ref. [8].

1.2 A new authentication scheme that admits universal computation

Our protocol employs a method for quantum computation on authenticated data (QCAD), which refers to the application of quantum gates to authenticated

quantum data without knowing the authentication key. We propose a new authentication scheme, called the *trap scheme*, and show that it allows for QCAD. Our trap scheme also seems to provide a concrete and efficient realization of the "hidden subspaces" used in the public-key quantum money scheme of Ref. [9].

Prior to our work, the only authentication scheme known to admit QCAD was the *signed polynomial scheme* [10,7]. Recently, and independently of our work, it was shown in Ref. [11] that the *Clifford authentication scheme* can be used to authenticate two-party quantum computations. However, that protocol requires two parties to process quantum information and so cannot be used for QCAD or QOTPs.

Our QOTP protocol calls for the receiver to use QCAD to apply the gates of Φ to the authenticated input registers (A, B). In general, QCAD can only be performed if the receiver (who holds the authenticated data) is allowed to exchange classical messages with the sender (who knows the authentication key). To keep our protocol non-interactive, all the classical interaction is encapsulated by a *bounded, reactive classical one-time program (BR-OTP)* prepared by the sender, the existence of which follows straightforwardly from the work of Ref. [3] and is described in detail in the full version of this extended abstract [12]. This program for the BR-OTP depends upon the authentication key chosen for the sender's input register, but *not* on the contents of that register. By selecting this key in advance, the BR-OTP can be prepared before the sender gets his input register.

To implement QCAD, the receiver's input must be authenticated prior to computation. This is accomplished non-interactively by having the sender prepare a pair of registers in a special "teleport-through-encode" state. The authentication key is determined by the (classical) result of the Bell measurement used for teleportation. The receiver non-interactively de-authenticates the output at the end of the computation by means of a special "teleport-through-decode" state, also prepared by the sender. In order to successfully de-authenticate, the receiver's messages to the BR-OTP must be consistent with the secret authentication key held by the BR-OTP. Otherwise, the BR-OTP simply declines to reveal the final decryption key for the receiver's output.

1.3 Unlockable functions and channels

Curiously, our study has uncovered pathological classes of functions and channels that can *never* be made into a one-time program. For example, the function $f : (x, y) \mapsto x + y$ cannot have a one-time program because a receiver can use his knowledge of y to deduce x from $f(x, y)$. Once he has deduced x, the receiver is free to evaluate $f(x, y')$ for any y' of his choosing. This function is an example of what we call an *unlockable* function. Technically, it is incorrect to say that such a function can never be made into a one-time program. Rather, such functions admit *trivial* one-time programs in the *plain model*—a technicality arising from the standard simulation-based definition of security. This phenomenon is somewhat akin to trivially obfuscatable functions [6].

We propose a definition of unlockability and prove that a classical function or quantum channel admits a one-time program in the plain quantum model if

and only if it is unlockable, implying that no "useful" function or channel admits a one-time program without any hardware assumptions.

2 Security of quantum one-time programs

Intuitively, a QOTP for a channel Φ is secure if anything that any (possibly cheating) receiver could learn by processing the program state prepared by the sender could also be learned by interacting with a simulator that uses only one-time access to an idealized black box for Φ. Thus, no receiver can learn anything beyond what can be inferred from this ideal functionality for Φ.

Formally, we define security in the quantum UC framework as defined by Unruh [13]. Our main ideal functionality, $\mathcal{F}_{\Phi}^{\mathrm{OTP}}$, is specified in Functionality 1 and involves two parties, the *sender* and the *receiver*. The functionality may exist in multiple instances and involve various parties.[4]

Functionality 1 Ideal functionality $\mathcal{F}_{\Phi}^{\mathrm{OTP}}$ for a quantum channel $\Phi : (\mathsf{A}, \mathsf{B}) \to \mathsf{C}$

1. **Create:** Upon input register A from the sender, send `create` to the receiver and store the contents of register A.
2. **Execute:** Upon input register B from the receiver, evaluate Φ on registers A, B and send the contents of the output register C to the receiver. Delete any trace of this instance.

The map Φ that is computed is a public parameter of the functionality and it takes an input from the sender and an input from the receiver, so $\mathcal{F}_{\Phi}^{\mathrm{OTP}}$ hides the sender's *input* only. If the intention is to hide the map Φ itself—as in the intuitive notion of one-time programs—then we can consider a universal map U that takes as part of the sender's input a representation of Φ (see [14,15,16]). Sometimes we emphasize the fact that the ideal functionality may be called only a single time by saying "one-shot access to an ideal functionality for Φ". The functionality $\mathcal{F}_{\Phi}^{\mathrm{OTP}}$ is *sender-oblivious* since it delivers the result of the functionality to the receiver but not the sender.

We now give some intuition on how the notions of UC translate to the context of QOTPs.

Functionality. A *non-interactive protocol for evaluation of a channel* $\Phi : (\mathsf{A}, \mathsf{B}) \to \mathsf{C}$ consists of (i) an *encoding channel* $\mathrm{enc} : \mathsf{A} \to \mathsf{P}$ applied by the sender on its input A that prepares a program state P, and (ii) a *decoding channel* $\mathrm{dec} : (\mathsf{P}, \mathsf{B}) \to \mathsf{C}$ applied by the receiver on the program state P and its input B such that $\mathrm{dec} \circ \mathrm{enc}$ and Φ are indistinguishable.

[4] Formally, instances are denoted by *session identifiers* and each instance involves labelled parties. For simplicity, we have omitted these identifiers as they are implicit from the context.

When P consists solely of a quantum register, we call this the *plain quantum model*. In the *bounded reactive OTP-quantum-hybrid model*, the program state is a quantum register P augmented with one or more BR-OTPs. (For our construction, it suffices to consider a single BR-OTP.) In this setting, the actions of any receiver (honest or otherwise) can be viewed as the serialization of a multi-round "interaction" in which the first message consists of the quantum registers from the sender and subsequent messages consist of purely classical data exchanged with the BR-OTP.

Security. By the completeness of the dummy-adversary [13], in order to show security, it suffices to consider only the adversary that relays messages between the environment and the honest parties (we can see the environment as performing the attack). Thus, security of a non-interactive protocol for the evaluation of Φ in the BR-OTP-quantum-hybrid model corresponds to the existence of a *simulator* that can mimic the sender's message, combined with the interactive behaviour of the BR-OTP, using only one-shot, black-box access to Φ with register A fixed.

A key result of Unruh [13] is the *quantum lifting theorem* which establishes that, in the statistical case, classical-UC-secure protocols are quantum-UC-secure. We apply this result to the protocol of Goyal, Ishai, Sahai, Venkatesan, and Wadia [3], which establishes statistically classical-UC-secure one-time programs in the OTM-hybrid model (i.e., assuming one-time memories); by quantum lifting, this protocol is also statistically quantum-UC-secure and hence we can use it our construction. Ideal functionalities for OTMs, OTPs, and BR-OTPs, as well as a proof extending Goyal et al.'s result for OTPs to BR-OTPs, appear in the full version [12].

3 The trap authentication scheme

In this section we present a new quantum authentication scheme called the *trap scheme* and argue that it admits quantum computation on authenticated data (QCAD). A *quantum authentication scheme* consists of procedures for encoding and decoding quantum information with a secret classical key k such that an adversary with no knowledge of k who tampers with encoded data will be detected with high probability. Quantum authentication codes were first introduced by Barnum, Crépeau, Gottesman, Smith and Tapp [17].

3.1 Trap codes yield a secure authentication scheme

Our trap scheme is based on any fixed quantum error-detecting code C that encodes one logical qubit into n physical qubits with distance d (an $[[n, 1, d]]$-*code*). Each such code induces a different trap scheme. Authentication and deauthentication operations for the trap scheme based on a code C are specified in Protocol 1.

The trap scheme is an example of a class of authentication schemes that we call *encode-encrypt schemes*, owing to a two-step authentication process of

Protocol 1 Authentication and de-authentication for the trap scheme based on an $[[n, 1, d]]$-code C

Classical key. A pair (π, P) consisting of a permutation π on $3n$ elements and a (description of a) $3n$-qubit Pauli operator P.

Authentication. *Input:* one qubit. *Output:* $3n$ qubits.

1. Encode the data qubit under C, producing an n-qubit register.
2. Introduce two new n-qubit *trap registers* in states $|0\rangle^{\otimes n}$, $|+\rangle^{\otimes n}$, respectively.
3. Permute all $3n$ qubits according to π.
4. Encrypt all $3n$ qubits by applying P.

De-authentication. *Input:* $3n$-qubits. *Output:* one qubit and "accept"; or "reject".

1. Decrypt all $3n$ qubits by applying P.
2. Permute all $3n$ qubits according to π^{-1}.
3. Decode the data qubit under C.
4. Measure the trap registers to ensure they are in their proper states. If these measurements succeed and if C indicated no error syndrome then "accept" and output the data qubit, otherwise "reject".

encoding followed by encryption. Encode-encrypt schemes have many desirable properties, chief among them the fact that an arbitrary attack on such a scheme is equivalent to a probabilistic mixture of Pauli attacks on the underlying family $\mathscr{E}$ of codes. Thus, by the encode-encrypt mechanism, in order to construct a secure quantum authentication scheme it suffices to exhibit a family $\mathscr{E}$ of codes that is secure against Pauli attacks.

In the trap scheme, the family $\mathscr{E}$ consists of all codes obtained by permuting data encoded under C together with registers in states $|0\rangle^{\otimes n}$, $|+\rangle^{\otimes n}$. We call $\mathscr{E}$ a *family of trap codes*. The first use of these codes was implicit in the Shor–Preskill security proof for quantum key distribution [18]. (See also Ref. [19].) We establish security of this family against Pauli attacks, from which the security of the trap scheme follows.

Proposition 1 (Security of trap codes against Pauli attacks). *The family $\mathscr{E}$ of trap codes based on a code of distance d is $(2/3)^{d/2}$-secure against Pauli attacks.*

That is, for each fixed choice of $3n$-qubit Pauli operation Q it holds that the probability—taken over a uniformly random choice of code $E \in \mathscr{E}$—that Q acts nontrivially on logical data and yet has no error syndrome is at most $(2/3)^{d/2}$.

See the full version [12] for proofs of Proposition 1 and several other properties of encode-encrypt schemes.

3.2 The trap scheme admits quantum computing on authenticated data

Authentication schemes that also allow for QCAD—the implementation of a universal set of quantum gates on authenticated data without knowing the key—hold great promise for a host of cryptographic applications. In this section we

argue that the trap scheme allows for QCAD for appropriate choices of the underlying code C.

It helps to think of two parties: a trusted *verifier* who prepares authenticated data with secret classical key k and a malicious *attacker* who is to act upon the authenticated data without knowledge of k. The goal is to construct a scheme with the property that for each gate G belonging to some universal set of gates there exists a *gadget* circuit $\tilde{G}$ that the attacker can apply to authenticated data so as to implement a logical G. Furthermore, we require that the gadget $\tilde{G}$ be *independent of the choice of classical key* k so that it may be implemented by an attacker without knowledge of k.

Normally, any non-identity gadget $\tilde{G}$ would invalidate the authenticated state. We therefore require a scheme which allows the verifier to validate the state again simply by updating the classical key $k \mapsto k'$. Moreover, by updating the key in this way the verifier effectively *forces* the attacker to apply the desired gadget $\tilde{G}$ as otherwise the state would fail verification under the updated key k'.

Following the example of the polynomial scheme of Ben-Or et al. [10], gadget design for our trap scheme is inspired by methods for fault-tolerant quantum computation. In the full version [12] we present gadgets for the universal gate set consisting of Pauli gates, controlled-NOT, Hadamard, i-shift phase $K : |a\rangle \mapsto i^a|a\rangle$, and $\pi/8$-phase $T : |a\rangle \mapsto e^{ai\pi/4}|a\rangle$.

Some gates, such as the controlled-NOT, admit straightforward bitwise gadgets. Others, such as the $\pi/8$ gate, require authenticated "magic states" and the ability to measure authenticated data in the computational basis. For these gadgets the verifier must interpret the classical measurement result for the attacker so that he may complete the gadget. Thus, these gadgets require classical interaction between verifier and attacker.

Our gadgets require that the underlying code C allow bitwise implementation of logical controlled-NOT and Hadamard gates—that is, that C be a self-dual CSS code. For a concrete example, it suffices that C be the seven-qubit Steane code nested a sufficient number of levels so as to achieve distance d.

4 Protocol for quantum one-time programs

In this section we present our protocol for quantum one-time programs in the quantum BR-OTP hybrid model. In particular, we specify how an honest sender prepares her quantum registers and BR-OTP for the receiver and how an honest receiver should use these objects to recover the action of Φ. The protocol requires an encode-encrypt scheme that admits QCAD such as the trap scheme presented in Section 3, but is completely independent of the specific choice of scheme.

We assume without loss of generality that the channel Φ has the form $\Phi : (\mathsf{A}, \mathsf{B}) \to \mathsf{B}$ so that the receiver's output register $\mathsf{C} \cong \mathsf{B}$ has the same size as the input register and that Φ is specified by a unitary circuit U acting on registers $(\mathsf{A}, \mathsf{B}, \mathsf{E})$. The extra register E is an auxiliary register initialized to the $|0_\mathsf{E}\rangle$ state. The action of Φ is recovered from U by discarding registers (A, E) so that $\Phi : \rho \mapsto \mathrm{Tr}_{\mathsf{AE}}(U\,(\rho \otimes |0_\mathsf{E}\rangle\langle 0_\mathsf{E}|)\,U^*)$.

Given a circuit U one can efficiently find a circuit for the controlled-U operation, which we denote c-U. This circuit acts on registers $(\mathsf{A}, \mathsf{B}, \mathsf{E})$ plus an extra control qubit, which we bundle into the auxiliary register E for convenience. Our protocol calls for the receiver to apply c-U to authenticated data with the control qubit always initialized to the $|\text{on}\rangle$ state. The purpose of this technicality is to better facilitate the proof of security. We also have an alternate protocol in which logical U is implemented directly with no need for c-U. However, the security proof for this alternate protocol is more technically cumbersome than our protocol for c-U, so we have elected to present only the protocol for c-U in this extended abstract.

4.1 Protocol for an honest sender

Let r be the number of gates in c-U that require magic states. After the parties have received their input registers A, B, a non-interactive protocol for c-U consists of a single message from the sender to the receiver containing the following objects:

1. Quantum registers $\tilde{\mathsf{A}}, \mathsf{B}_{\mathsf{in}}, \tilde{\mathsf{B}}_{\mathsf{in}}, \mathsf{B}_{\mathsf{out}}, \tilde{\mathsf{B}}_{\mathsf{out}}, \tilde{\mathsf{E}}, \tilde{\mathsf{M}} = (\tilde{\mathsf{M}}_1, \ldots, \tilde{\mathsf{M}}_r)$.
2. An $(r+1)$-round BR-OTP.

The sender prepares these objects as specified in Protocol 2 and Figure 1.

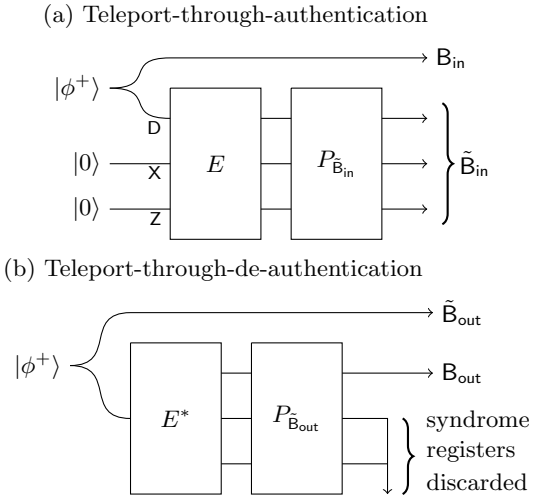

Fig. 1. Circuits for teleporting through authentication and de-authentication. Here the Pauli operations $P_{\tilde{\mathsf{B}}_{\mathsf{in}}}, P_{\tilde{\mathsf{B}}_{\mathsf{out}}}$ refer to the portions of P acting on registers $\tilde{\mathsf{B}}_{\mathsf{in}}, \tilde{\mathsf{B}}_{\mathsf{out}}$, respectively.

Protocol 2 Message preparation for an honest sender

Secret classical key. Authentication key for registers $(\tilde{\mathsf{A}}, \tilde{\mathsf{B}}_{\mathsf{in}}, \tilde{\mathsf{B}}_{\mathsf{out}}, \tilde{\mathsf{E}}, \tilde{\mathsf{M}})$. In particular, a random pair (E, P) consisting of a code $E \in \mathscr{E}$ and Pauli P acting on these registers.

Registers prepared by the sender. Given the input register A the sender prepares the following registers:

$(\mathsf{B}_{\mathsf{in}}, \tilde{\mathsf{B}}_{\mathsf{in}})$:	Teleport-through-authentication state of Figure 1(a).
$(\tilde{\mathsf{B}}_{\mathsf{out}}, \mathsf{B}_{\mathsf{out}})$:	Teleport-through-de-authentication state of Figure 1(b).
$\tilde{\mathsf{A}}$:	Authenticated input register A.
$\tilde{\mathsf{E}}$:	Authenticated ancilla in logical state $\|0\rangle\|\text{on}\rangle$.
$\tilde{\mathsf{M}}$:	Authenticated ancilla in logical state $\|\mu\rangle = \|\mu_1\rangle \cdots \|\mu_r\rangle$ where $\|\mu_1\rangle, \ldots, \|\mu_r\rangle$ are the r magic states required for c-U.

BR-OTP prepared by the sender.

1. Receive (a classical description of) a purported teleport-through-authentication correction Pauli T^{in}.
2. For $i = 1, \ldots, r$:
   (a) Receive a classical bit string c_i—a purported measurement result of the ith authenticated magic state register $\tilde{\mathsf{M}}_i$.
   (b) Decode c_i into a classical bit a_i as dictated by T^{in} and the authentication key (E, P). If the decoding process indicates a non-zero error syndrome then cheating has been detected. Return the decoded bit a_i to the user.
3. Receive a purported teleport-through-de-authentication correction Pauli T^{out}. If cheating was never detected in step 2b then return a decryption Pauli $\hat{S}$. Otherwise return random bits.

4.2 Protocol for an honest receiver

An honest receiver can recover $\Phi(\rho)$ from an honest sender's message as specified in Protocol 3.

5 Simulator and proof of UC security

The simulator must not pre-process the sender's input register A. Instead, the simulator is permitted only one-shot, black-box access to the "ideal functionality" for Φ. We represent this ideal functionality by a single call to an oracle for U acting on registers $(\mathsf{A}, \mathsf{B}, \mathsf{E})$ prepared by the simulator. The rules for permissible preparation and disposal of these registers are as follows:

1. The simulator must pass the input register A directly to U without any pre-processing.
2. The simulator must prepare the ancillary register E in pure state $|0\rangle$.
3. Upon receiving the output registers $(\mathsf{A}, \mathsf{B}, \mathsf{E})$ from the oracle for U, the simulator must discard registers A, E without any post-processing.

The simulator is specified in Protocol 4. The main idea is that our simulator will use the control qubit contained in register $\tilde{\mathsf{E}}$ to "switch off" the application of U

Protocol 3 Protocol for an honest receiver

1. Perform a Bell measurement on $(\mathsf{B}, \mathsf{B}_{\mathsf{in}})$ so as to teleport-through-authentication. Let T^{in} be the correction Pauli indicated by this measurement. Send T^{in} as the first message to the BR-OTP. *[At this time the contents of* B *have been authenticated and placed in register* $\tilde{\mathsf{B}}_{\mathsf{in}}$*.]*
2. Apply a logical c-U to the authenticated registers $(\tilde{\mathsf{A}}, \tilde{\mathsf{B}}_{\mathsf{in}}, \tilde{\mathsf{E}}, \tilde{\mathsf{M}})$. Explicitly:
   (a) Apply the gates of c-U occurring before the first magic state measurement.
   (b) For $i = 1, \ldots, r$:
      i. Measure the ith magic state register in the computational basis and send the result to the BR-OTP.
      ii. The BR-OTP provides a single bit indicating the proper correction.
      iii. Apply the gates of c-U occurring after the ith magic state measurement but before the $(i+1)$th magic state measurement.

   [The implementation of c-U *is now complete. At this time the register* $(\tilde{\mathsf{A}}, \tilde{\mathsf{B}}_{\mathsf{in}}, \tilde{\mathsf{E}})$ *holds the authenticated version of* $(\mathsf{A}, \mathsf{B}, \mathsf{E})$ *with* c-U *applied.]*
3. Perform a Bell measurement on $(\tilde{\mathsf{B}}_{\mathsf{in}}, \tilde{\mathsf{B}}_{\mathsf{out}})$ so as to teleport-through-de-authentication. Let T^{out} be the correction Pauli indicated by this measurement. Send T^{out} as the final message to the BR-OTP. *[At this time the register* $\mathsf{B}_{\mathsf{out}}$ *holds the receiver's output. This register is encrypted but not authenticated.]*
4. For its final output, the BR-OTP provides the Pauli decryption key $\tilde{S}$. Apply this Pauli to $\mathsf{B}_{\mathsf{out}}$ to recover the output of Φ.

that would have been implemented by an honest receiver. Instead, the black-box call to the ideal functionality will be embedded at the proper time so as to recover the required action of U. An additional teleportation step is required so that our simulator can embed U at the proper time.

We now sketch a proof that the simulator of Protocol 4 certifies the security of the QOTP protocol presented in Section 4. We begin with a formal re-statement of Theorem 1 in the language of UC-security. Details appear in the full version [12].

Theorem 2 (Main theorem, formal). *For each channel* $\Phi : (\mathsf{A}, \mathsf{B}) \to \mathsf{C}$ *specified by a quantum circuit, there is an efficient, non-interactive, quantum protocol in the OTM-hybrid model that statistically quantum-UC-emulates* $\mathcal{F}_{\Phi}^{\mathrm{OTP}}$ *against a malicious receiver.*

Proof (sketch). As discussed in Section 2, UC-security of our protocol is established by proving that that no entity (or *environment*) could possibly distinguish an interaction with our simulator from an interaction with a real sender. We employ a highly technical, "brute-force" approach to this end. In particular, we begin by writing down a general form that every environment must have. From such a description we derive an expression for the final state of all the registers in the environment's possession at the end of an interaction with a real sender. We perform a similar analysis for the environment's final state at the end of an interaction with our simulator. Finally, we argue that these two final states are statistically indistinguishable—that is, the trace distance between them is proportional to the security parameter of the underlying encode-encrypt scheme

Protocol 4 Simulator

Secret classical key. Authentication key (E, P) for registers $(\tilde{\mathsf{A}}, \tilde{\mathsf{B}}_{\mathsf{in}}, \tilde{\mathsf{B}}_{\mathsf{out}}, \tilde{\mathsf{E}}, \tilde{\mathsf{M}})$ as in Protocol 2.

Registers prepared by the simulator. Given the input register A, the simulator constructs the following registers:

$(\mathsf{B}_{\mathsf{in}}, \mathsf{S}_{\mathsf{in}})$:	Simple EPR pairs $\|\phi^+\rangle$ for teleportation.
$(\mathsf{S}_{\mathsf{out}}, \tilde{\mathsf{B}}_{\mathsf{in}})$:	Teleport-through-authentication state of Figure 1(a).
$(\tilde{\mathsf{B}}_{\mathsf{out}}, \mathsf{B}_{\mathsf{out}})$:	Teleport-through-de-authentication state of Figure 1(b).
$\tilde{\mathsf{A}}$:	Authenticated dummy input register in logical state $\|0\rangle$.
$\tilde{\mathsf{E}}$:	Authenticated dummy ancillary register in logical state $\|0\rangle\|\text{off}\rangle$.
$\tilde{\mathsf{M}}$:	Authenticated magic states as in Protocol 2.
E :	To be used in the call to the ideal functionality. Ancillary register in state $\|0\rangle$.

Execution of the simulator.

1. Send the registers $\mathsf{B}_{\mathsf{in}}, \tilde{\mathsf{B}}_{\mathsf{in}}, \tilde{\mathsf{B}}_{\mathsf{out}}, \mathsf{B}_{\mathsf{out}}, \tilde{\mathsf{A}}, \tilde{\mathsf{E}}, \tilde{\mathsf{M}}$ to the environment.
2. The environment responds with a Pauli T^{in}. Apply T^{in} to register S_{in}. Then use the ideal black-box to apply U to $(\mathsf{A}, \mathsf{S}_{\mathsf{in}}, \mathsf{E})$.
3. Perform a Bell measurement on $(\mathsf{S}_{\mathsf{in}}, \mathsf{S}_{\mathsf{out}})$ so as to teleport the contents of S_{in} through authentication and place the result in $\tilde{\mathsf{B}}_{\mathsf{in}}$. Let T^{sim} denote the teleportation Pauli indicated by this measurement.
4. Execute the BR-OTP of Protocol 2 under the assumption that T^{sim} was received in the first round.

upon which our protocol is based. (For example, if the trap scheme built on a code of distance d is used with our protocol then this trace distance is exponentially small in d.) □

6 Impossibility of non-trivial OTPs in the plain model

In this section we propose a definition of unlockability for quantum channels, from which a definition for classical functions arises as a special case. We then prove that a channel Φ admits a one-time program in the plain quantum model if and only if Φ is unlockable, from which it follows that a classical function f admits a one-time program in the plain classical model if and only if f is unlockable.

Specifically, our *possibility* result (Theorem 3) is that every unlockable channel admits a trivial one-time program in the plain quantum model, and in fact that this protocol is UC-secure. Conversely, our *impossibility* result (Theorem 4) is that every channel that is *not* unlockable does *not* admit a one-time program in the plain quantum model. This impossibility result holds even if we relax to an approximate case or allow computational assumptions.[5]

[5] Although our impossibility result is stated in the UC framework, the impossibility is not an artifact of the high security required by UC, but seems inherent in the notion of OTPs, and the impossibility argument applied for any relaxation we attempted.

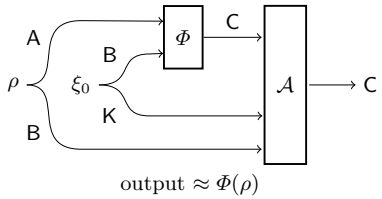

Fig. 2. A channel Φ is *unlockable* if there exists a key state ξ_0 and a recovery algorithm $\mathcal{A}$ that allows computation of $\Phi(\rho)$ for any ρ.

6.1 Definition of unlockability

Informally, a function or channel is unlockable if there is a *key*[6] input for the receiver that unlocks enough information to fully simulate the map.

Definition 1 (Unlockable channel). *A channel* $\Phi : (\mathsf{A}, \mathsf{B}) \to \mathsf{C}$ *is* unlockable *if there exists a register* K*, a* key state ξ_0 *of* (B, K) *and a* recovery algorithm *(i.e., channel)* $\mathcal{A} : (\mathsf{C}, \mathsf{K}, \mathsf{B}) \to \mathsf{C}$ *with the property that* $\mathcal{A} \circ (\Phi_0 \otimes \mathbb{1}_{\mathsf{B}}) \approx \Phi$*, where the channel* Φ_0 *is specified by* $\Phi_0 : \mathsf{A} \to (\mathsf{C}, \mathsf{K}) : \rho_A \mapsto (\Phi \otimes \mathbb{1}_{\mathsf{K}})(\rho_A \otimes \xi_0)$*. Here,* $\approx$ *can denote perfect, statistical, or (for polynomial-time uniform families of channels* $\{\Phi_n\}$*) computational indistinguishability; in all cases, the channels* Φ_0 *and* $\mathcal{A}$ *must have circuits of size polynomial in the size of the circuit for* Φ*. See Figure 2 for a graphical depiction of unlockability.*

For completeness let us note that, in the classical case, a function $f : A \times B \to C$ is *unlockable* if there exists a *key input* $b_0 \in B$ and a *recovery algorithm* $\mathcal{A} : C \times B \to C$ such that, for all $a \in A$ and $b \in B$, it holds that $f(a, b) = \mathcal{A}(f(a, b_0), b)$. Intuitively, an unlockable classical function admits an algorithm that can compute all values of $f(a, \cdot)$ given a one-time program for $f(a, \cdot)$. But this is okay because a simulator given one-shot oracle access to $f(a, \cdot)$ can also compute $f(a, b)$ for all b: this function is "learnable" in one shot and so a simulator can do everything any algorithm can.

Simple examples of unlockable channels include all unitary channels of the form $\Phi : X \mapsto UXU^*$ for some unitary U and all constant channels of the form $\Phi : X \mapsto \mathrm{Tr}(X)\sigma$ for some fixed state σ. Simple examples of unlockable functions include permutations.

6.2 Trivial one-time programs for unlockable channels

We can now see that unlockable channels have OTPs; but trivially so.

Theorem 3 (OTPs for unlockable channels). *Let* $\Phi : (\mathsf{A}, \mathsf{B}) \to \mathsf{C}$ *be a channel specified by a circuit. If* Φ *is unlockable then there exists an efficient, non-interactive protocol which quantum-UC-emulates* $\mathcal{F}_\Phi^{\mathrm{OTP}}$ *in the plain quantum model. This holds in the perfect, statistical and computational cases.*

[6] Note we use "key" not in the cryptographic sense of a secret key, but in the metaphorical sense of something that unlocks a lock.

Proof (sketch). The protocol is simple: to prepare the program register, the sender applies Φ to his input register A and the B-portion of the key state ξ_0. The receiver can recover the action of Φ simply by applying the recovery algorithm $\mathcal{A}$ to his input register B and the program register received from the sender.

In order to show that this protocol is secure it suffices to exhibit a simulator that can re-create the sender's program state using only the ideal functionality for Φ. But this is easy: the simulator can produce this state exactly as the real sender would—by using the ideal functionality to apply Φ to the environment's input register A and the simulator's B-portion of the key state ξ_0.

Theorem 3 is in the quantum setting; it follows from the proof that if Φ is a classical channel then the resulting protocol is a purely classical protocol.

6.3 Impossibility of one-time programs for arbitrary channels

Having seen that unlockable channels admit one-time programs in the plain quantum model, we now show the converse—that every channel which admits a one-time program must be unlockable.

Theorem 4 (Channels that admit OTPs are unlockable). *Let $\Phi : (\mathsf{A}, \mathsf{B}) \to \mathsf{C}$ be a channel specified by a circuit and suppose that Φ admits an efficient, non-interactive quantum protocol which quantum-UC-emulates $\mathcal{F}_\Phi^{\mathrm{OTP}}$ in the plain quantum model. Then Φ is unlockable. This holds in the perfect, statistical and computational cases.*

Proof (sketch). Because Φ has a one-time program there must be a simulator that can reproduce the program state using only the ideal functionality for Φ applied to the sender's input register A. Any such simulator has the following form: (i) prepare a state ξ, (ii) apply Φ to A and the B-portion of ξ, (iii) post-process the result by applying some channel $\mathcal{A}$. It follows that Φ is unlockable with key state ξ and recovery algorithm $\mathcal{A}$.

For classical functions, an alternate intuition for this impossibility result can be found by considering rewinding. Any correct one-time program state ρ_x for a classical function $f(x, \cdot)$ must result in the receiver obtaining an output state $\rho_{x,y}$ that is (almost) diagonal in the basis in which the receiver measures it, because the measurement of $\rho_{x,y}$ results in $f(x, y)$ with (almost) certainty. As a result, measurement does not disturb the state (much), so the receiver can reverse the computation to obtain (almost) the program state again, and then rerun the computation to obtain (close to) $f(x, y')$ for a different y'. It is possible to give a proof for impossibility of OTPs for classical functions in the plain quantum model using this rewinding argument. Impossibility for classical functions also follows as a special case of the impossibility shown in Ref. [20].

7 UC-security of delegated quantum computations

Several protocols have been designed to allow a computationally weak client to interface with a quantum computer in order to remotely accomplish a quantum computation while maintaining privacy of the user's input [21,19,7]. These works, however, do not consider composability. (Recently, Dunjko, Fitzsimons, Portmann and Renner [8] showed the composability of the blind quantum computing protocol of Ref. [19].)

In this section we show that our main proof technique can be used to establish the statistical quantum-UC security of a family of protocols for delegated quantum computations, closely related to the protocol of Aharonov et al. [7]. Originally studied in the context of *quantum interactive proof systems*, the protocol of Aharonov et al., which provides a mechanism to ensure both privacy of the user's input *and* verifiability of the computation, was not originally shown to be secure according to any rigorous cryptographic security definition.

We generalize the protocol of Aharonov et al. to support delegated quantum computation (in contrast to only deciding membership in a language) by making two minor modifications. First we instantiate the protocol using any encode-encrypt quantum authentication scheme that admits computing on authenticated data (such as the trap scheme or the signed polynomial scheme as used by Aharonov et al.). Analogously to our main protocol, we also introduce as an aid in the proof a control bit so that the circuit being implemented is a controlled-unitary.

The ideal functionality we achieve is described in Functionality 2. Following [7], we describe the functionality in terms of a *prover* and *verifier*.

Functionality 2 Ideal functionality $\mathcal{F}_{\Phi}^{\text{delegated}}$ for a quantum channel $\Phi : \mathsf{A} \to \mathsf{C}$

1. **Create:** Upon input register A from the verifier, send `create` to the prover and store the contents of register A.
2. **Execute:** The prover provides an input in $\{\texttt{execute}, \texttt{abort}\}$. Upon input `execute`, evaluate Φ on register A, and send the contents of the output register C to the verifier; upon input `abort`, output $\perp$ to the verifier.

Theorem 5. *Let Φ be a channel specified by a circuit. There exists an efficient quantum interactive protocol in the plain model that statistically quantum-UC-emulates $\mathcal{F}_{\Phi}^{\text{delegated}}$ against a malicious prover. Furthermore, the only quantum power required of the verifier is to encode the input and auxiliary quantum registers and to decode the output. In particular, all the interaction is classical except for the first and last messages.*

The proof of Theorem 5 follows as a special case of our main result about QOTPs (Theorem 2). In the case of a general Φ, the registers that the verifier prepares in Theorem 5 are polynomial-size in the security parameter. In the

interactive proof scenario of Aharonov et al., the input to Φ is the all-$|0\rangle$ product state, the output is a single classical bit, and it suffices to implement $\mathcal{F}_{\Phi}^{\text{delegated}}$ with only constant security. Given these assumptions, the only quantum power required of the verifier is the ability to prepare constant-sized quantum registers in the first round.

Acknowledgements

We gratefully acknowledge helpful discussions with Harry Buhrman, Daniel Gottesman, Christopher Portmann, Bruce Richmond, Christian Schaffner and Dominique Unruh. A.B. acknowledges support from the Canadian Institute for Advanced Research (CIFAR), Canada's NSERC and Industry Canada. G.G. acknowledges support from Industry Canada, Ontario's Ministry of Research and Innovation, NSERC, DTO-ARO, CIFAR, and QuantumWorks. D.S. acknowledges support from the Australian Research Council. Part of this research conducted while D.S. was a visitor at the University of Waterloo's IQC.

References

1. Goldwasser, S., Kalai, Y., Rothblum, G.: One-time programs. In: CRYPTO 2008. Volume 5157 of LNCS., Springer (2008) 39–56
2. Bellare, M., Hoang, V.T., Rogaway, P.: Adaptively secure garbling with applications to one-time programs and secure outsourcing. In: ASIACRYPT 2012. Volume 7658 of LNCS., Springer (2012) 134–153
3. Goyal, V., Ishai, Y., Sahai, A., Venkatesan, R., Wadia, A.: Founding cryptography on tamper-proof hardware tokens. In: TCC 2010. Volume 5978 of LNCS. Springer (2010) 308–326 Full version available at http://eprint.iacr.org/2010/153.
4. Aaronson, S.: Quantum copy-protection and quantum money. In: Proc. 24th IEEE Conference on Computational Complexity (CCC) 2009. (2009) 229–242
5. Mosca, M., Stebila, D.: Quantum coins. In: Error-Correcting Codes, Finite Geometries and Cryptography. Volume 523 of Contemporary Mathematics., American Mathematical Society (2010) 35–47
6. Barak, B., Goldreich, O., Impagliazzo, R., Rudich, S., Sahai, A., Vadhan, S., Yang, K.: On the (im)possibility of obfuscating programs. In: CRYPTO 2001. Volume 2139 of LNCS., Springer (2001) 1–18 Full version available at http://www.wisdom.weizmann.ac.il/~oded/p_obfuscate.html.
7. Aharonov, D., Ben-Or, M., Eban, E.: Interactive proofs for quantum computations. In: Proc. Innovations in Computer Science (ICS) 2010. (2010) 453–469
8. Dunjko, V., Fitzsimons, J.F., Portmann, C., Renner, R.: Composable security of delegated quantum computation. arXiv:1301.3662 [quant-ph] (2013)
9. Aaronson, S., Christiano, P.: Quantum money from hidden subspaces. In: Proc. 44th Symposium on Theory of Computing (STOC) 2012. (2012) 41–60 Full version availables as arXiv:1203.4740 [quant-ph].
10. Ben-Or, M., Crépeau, C., Gottesman, D., Hassidim, A., Smith, A.: Secure multiparty quantum computation with (only) a strict honest majority. In: FOCS 2006. (2006) 249–260

11. Dupuis, F., Nielsen, J.B., Salvail, L.: Actively secure two-party evaluation of any quantum operation. In: CRYPTO 2012. Volume 7417 of LNCS., Springer (2012) 794–811
12. Broadbent, A., Gutoski, G., Stebila, D.: Quantum one-time programs (full version) (2013) arXiv:1211.1080 [quant-ph].
13. Unruh, D.: Universally composable quantum multi-party computation. In: EUROCRYPT 2010. Volume 6110 of LNCS. Springer (2010) 486–505 Full version available as arXiv:0910.2912 [quant-ph].
14. Bera, D., Fenner, S., Green, F., Homer, S.: Efficient universal quantum circuits. Quantum Information and Computation **10**(1) (January 2010) 16–28
15. Nielsen, M.A., Chuang., I.L.: Programmable quantum gate arrays. Physical Review Letters **79** (1997) 321–324
16. de Sousa, P.B., Ramos, R.V.: Universal quantum circuit for N-qubit quantum gate: a programmable quantum gate. Quantum Information and Computation **7**(3) (March 2007) 228–242
17. Barnum, H., Crépeau, C., Gottesman, D., Smith, A., Tapp, A.: Authentication of quantum messages. In: FOCS 2002. (2002) 449–458 Full version available as arXiv:quant-ph/0205128.
18. Shor, P., Preskill, J.: Simple proof of security of the BB84 quantum key distribution protocol. Physical Review Letters **85** (2000) 441–444
19. Broadbent, A., Fitzsimons, J., Kashefi, E.: Universal blind quantum computation. In: FOCS 2009, IEEE (2009) 517–526
20. Buhrman, H., Christandl, M., Schaffner, C.: Complete insecurity of quantum protocols for classical two-party computation. Physical Review Letters **109** (Oct 2012) 160501
21. Childs, A.: Secure assisted quantum computation. Quantum Information and Computation **5** (2005) 456–466