Quantum one-time programs

Submitted by:
Ch.K. Siddhartha
Ishan Mankodi

Registration no:
EP20B012
EP20B017

Guide:
Professor Shweta Agrawal
Department of CSE

CS6846

November 20, 2022

Contents

1 Introduction

1.1 One-time program

Computer programs can be arbitrarily copied, analysed, edited, and executed. Due to certain security applications, such a complete transfer of code is unwanted. Goldwasser et al. therefore suggest a

novel sort of program called a "one-time program" Goldwasser (2008). These programs allow for flexible input selection and can only be run once. The one-time zero-knowledge proofs, which can be verified exactly once by any verifier without the prover being present, are a new cryptographic notion that they enable us to conceptualise, define, and realise in this study. The proof is meaningless and cannot be validated again after being verified once by any one verifier. Software protection, electronic tokens, and electronic money can all be directly benefited from these programs.

Informally, A *one-time program (OTP)* for a function f, as introduced by Goldwasser *et al.*Goldwasser (2008), (1) Can be used to compute f on a single input x of one's choice. (2) No efficient adversary, given the one-time program, can learn more about f than can be learned from a single pair $(x, f(x))$, where x is chosen by the adversary.

Broadbent et al. (2013a) defines, An OTP for a function

$$f : \{\text{bit strings}\} \rightarrow \{\text{bit strings}\}$$

is a cryptographic primitive by which a user non-interactively evaluates f on one input x chosen at run-time. No user, after evaluating $f(x)$, should be able to learn anything about $f(x')$ for any $x' \neq x$ beyond that which can be learned from $f(x)$.

Hence, it acts like a black-box function that can only be evaluated once. It's clear that *One-time programs* cannot be achieved by only software alone, as any classical software can be re-run. Therefore, to achieve one-time property must necessarily rely on an additional assumptions such as secure hardware or quantum mechanics.

1.2 One-time memories

It has been demonstrated Goldwasser (2008)how to create a one-time program for any function f using a hypothetical hardware component known as a *one-time memory (OTM)*. A non-interactive version of oblivious transfer, an OTM stores two secret strings (or bits) s_0, s_1 allows a receiver to specify a bit c, retrieve s_c , and then self-destructs, erasing $s_{\bar{c}}$ permanently. OTMs are an appealing basic hardware supposition because they can potentially be mass-produced and their specification is independent of any particular function f .

One advantage of using OTMs as a building block is their simplicity: an OTM is an extremely basic device that does not perform any computation. Using the simplest possible hardware device allows for easier scrutiny against potential hardware flaws such as side-channel attacks. Moreover, the functionality of an OTM is independent of the program itself, and thus OTMs could be mass-produced for a variety of programs. The use of tamper-proof hardware in cryptography is an old and recurring theme Smid (1981), and OTMs in particular have lead to a recent revival in ascertaining what cryptographic primitives can be constructed using minimalistic hardware assumptions that could not otherwise be achieved.

1.3 Non-interactive two party computation

Goyal et al. (2010) In this study, we will look at a generalisation of the previously discussed one-time program known as Non interactive two-party computation (NI2PC). Which contains two parties

(sender and receiver) who want to evaluate a non-interactively public known function

$$g : \{\text{bit strings}\} \times \{\text{bit strings}\} \rightarrow \{\text{bit strings}\}$$

. First, the input string x is passed to the *sender*. The *sender* utilises x to prepare a "program" p(x) and sends this program to the *receiver*. The *receiver* would like to utilise the program $p(x)$ to evaluate $f(x, y)$ for any input string y of her choosing. After evaluating $f(x, y)$, no adversary should be able to discover anything about $f(x, y')$ for any $y' \neq y$ other than what can be inferred from $f(x, y)$.

Goldwasser et al's one-time program is recovered as a particular case of this primitive by seeing the input x as a description of a function $f_x(y)$ and the publicly known function g as a "universal computer" that outputs $g(x, y) = f_x(y)$. Non-interactive secure two-party computation in the simple model is impossible for the same reason that one-time programs are impossible: software can always be copied.

Goldwasser et al. 's one-time programs are secure against a hostile receiver; the question of a malicious sender does not arise in their context. In the more wide context of non-interactive secure two-party computing, malevolent senders could be considered. Goyal et al.'s protocol is secure against both a hostile receiver and a malicious sender.

Goldwasser et al. use OTMs for long strings, whereas Goyal et al. just need OTMs for single bits. Finally, Goldwasser et al. develop computationally constrained adversary security, whereas Goyal et al. provide statistical universally composable security(Discussed in later sections).

1.4 Quantum one-time Programs

We examine *quantum one-time programs* (QOTPs) in this report. In a QOTP, the sender and receiver assess a publicly known channel Φ: (A, B) C specified by a quantum circuit working on registers A (the sender's input), B (the receiver's input), and C (the receiver's output). The security objective is conceptually identical to that of classical functions: for any joint state ρ of the input registers (A,B), a malicious receiver should not be able to learn more about $\Phi(\rho')$ than what can be deduced from $\Phi(\rho)$, for any ρ' .

Note. The *QOTP*s are different from the task of *program obfuscation*, which is known to be impossible classically but remains an open question quantumly.

This section's remaining paragraphs enlarge on the following contributions.

1. A universally composable *QOTP* protocol that uses the same single-bit one-time memory as traditional OTPs and works with any quantum channel. The protocol makes use of a quantum cryptography approach known as quantum computing on authenticated data (*QCAD*).

2. A new quantum authentication scheme called the trap scheme and show that it allows for *QCAD*.

3. Identify problematic classes of trivial OTP-admitting quantum channels and "un-lockable" classical functions without making any hardware assumptions.

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

Quantum information, as contrast to regular classical information, cannot often be replicated. Does quantum information enable for one-time programming without making any hardware assumptions as a result of its no-cloning property? (We call this the *plain quantum model* when there are no hardware assumptions.) A quick thought reveals a negative response to this question for both classical functions and quantum channels: for any function f or channel Φ , a reversible receiver can always reconstruct a quantum "program state" for f or Φ after each use to obtain the evaluation of f or Φ on multiple different inputs. Assumptions based on computation are useless. Given that one-time programming for arbitrary classical functions under secure OTMs exist but not for arbitrary quantum channels in the plain quantum model, the question is: what additional assumptions are needed to obtain one-time programs for quantum channels? This is addressed by the key finding.

Theorem 1 ***(Main result, informal).*** For each channel $\Phi : (A, B) \rightarrow C$ 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.

A malicious sender cannot discover any information about the recipient's piece of the input state ρ because all communication is one-way from sender to receiver. Future research is left to address the issue of security against a dishonest sender who tries to persuade the receiver to accept an output state other than $\Phi(\rho)$. We focus only on the scenario involving non-reactive quantum one-time programs. Standard approaches can be used to create the more general scenario of bounded reactive programs that can be queried a bounded number of times, including the case of an n-use program, just as they are in the classical case.

1.6 A new authentication scheme that admits universal computation

The protocol uses a technique known as quantum computation on authenticated data (QCAD), which involves using quantum gates to process authenticated quantum data without being aware of the authentication key. We provide a brand new authentication method called the trap scheme and demonstrate that it supports QCAD. Additionally, it appears that our trap method offers a practical and effective realisation of the "hidden subspaces" employed in the public-key quantum money scheme.

To apply the gates of to the authenticated input registers, the QOTP protocol instructs the receiver to use QCAD (A, B). Generally speaking, QCAD can only be implemented if the sender and receiver are both permitted to send and receive conventional messages between them (who knows the authentication key). The sender creates a bounded, reactive classical one-time program (BR-OTP), the existence of which follows naturally from the findings , and which is extensively discussed in the complete form of this report, in order to make our protocol non-interactive. This program for the BR-OTP is not dependent on the data in the sender's input register, but rather on the authentication key selected for that register.

Prior to computation, the receiver's input needs to be authorised in order to use QCAD. By instructing the sender to pre-pare a pair of registers in a unique "teleport-through-encode" state, this is performed non-interactively. The (classical) outcome of the Bell measurement utilised for teleportation serves

as the authentication key. At the conclusion of the computation, the receiver non-interactively de-authenticates the output using a unique "teleport-through-decode" state that was also prepared by the sender. The receiver's communications with the BR-OTP must be in accordance with the secret authentication key that the BR-OTP is in possession of in order for de-authentication to be successful. Otherwise, the BR-OTP merely refuses to divulge the output of the receiver's final decryption key.

1.7 Unlockable functions and channels

Strangely, the research has revealed a problematic class of functions and channels that can never be implemented as a single program. For instance, the function $f : (x, y) \rightarrow x + y$cannot have a one-time program as a receiver can infer x from f using his understanding of y. The receiver is allowed to evaluate $f(x, y')$ for any y' of his choosing after having determined x. This particular function exemplifies what we mean by an unlocked function. It is technically erroneous to assert that a function like this can never be turned into a one-time application. As a technicality resulting from the conventional simulation-based definition of security, such functions instead allow trivial one-time programs in the plain model. This phenomenon is akin to trivially obfuscatable functions.

The authors define unlockability and demonstrate that, and only if it is unlockable, a (classical) function f admits a one-time program in the plain quantum model. The situation for quantum channels is pretty intriguing. Weakly and strongly unlockable channels are two kinds of channels defined by P aper. In the simple quantum model, the study shows that every highly unlockable channel admits a trivial one-time program. On the other hand, we demonstrate that any channel that allows a single-time program to run in the plain quantum model must be weakly unlockable. Every channel that can be strongly unlocked can also be weakly unlocked, which makes us believe that the two classes are equivalent. In conclusion, they demonstrate that no "useful" function or channel permits the admission of a one-time application that makes no hardware assumptions.

2 Notation and Background

A quantum register is a specific quantum system (composed, say, out of qubits). A quantum channel $\Phi : \mathrm{A} \rightarrow \mathrm{B}$ refers to any physically-realizable mapping on quantum registers. We use the terms quantum map and quantum channel interchangeably.

2.1 Universal Composability

The universal composability (UC) framework provides an extremely high standard for establishing a strong and rigorous notion of security. The basic idea is to postulate an ideal world, where the protocol parties interact with an ideal functionality, which is secure by definition. Then, we consider the real world, where the protocol parties execute the actual protocol. UC-security holds if, for every real-world adversary, there exists a simulator in the ideal world (taking the role of the real-world adversary) such that no environment can distinguish the real and ideal worlds. The environment is powerful: it provides the parties' inputs, receives outputs and interacts with the adversary at arbitrary points in the protocol. Perfect, statistical, and computational notions of UC security exist depending on the two worlds being equal or statistically or computationally indistinguishable.

The main possibility result heavily relies on classical one-time programs, for which a (classical) UC-secure instantiation exists assuming one-time memories. In particular, we require bounded reactive OTPs, which we construct by extending the results of Goyal et al.[Goyal et al. (2010)]

2.2 Ideal Functionalities

All functionalities involve two parties, the sender and the receiver. An ideal functionality may exist in multiple instances and involves various parties. Formally, instances are denoted by session identifiers and each instance involves labelled parties.

The ideal functionality $\mathcal{F}_{OTM}$ for a one-time memory (OTM) is a two-step process modelled after oblivious transfer.

Functionality 1 Ideal Functionality $\mathcal{F}_{OTM}$.

1. Create: Upon input (s_0, s_1) from the sender with $s_0, s_1 \in 0, 1$, send create to the receiver and store (s_0, s_1).
2. Execute: Upon input c $\in$ 0, 1 from the receiver, send s := s_c to the receiver. Delete any trace of this instance.

Next we describe the ideal functionalities $\mathcal{F}_f^{\text{OTP}}$ and $\mathcal{F}_\Phi^{\text{OTP}}$ of a one-time program for a classical function f and a quantum channel , respectively. Note that the map that is computed (f or Φ) is a public parameter of the functionality and it takes an input from the sender and an input from the receiver. We thus view these ideal functionalities as having the property of hiding the sender's input only.

Functionality 2 Ideal functionality $\mathcal{F}_f^{\text{OTP}}$ for a classical function $f : \{0,1\}^{n+m} \to \{0,1\}^k$.

1. Create: Upon input a $\in \{0,1\}^n$ from the sender, send create to the receiver and store a.
2. Execute: Upon input b $\in \{0,1\}^m$ from the receiver, send f(a, b) to the receiver. Delete any trace of this instance

Functionality 3 Ideal functionality $\mathcal{F}_\Phi^{\text{OTP}}$for a quantum channel $\Phi : (\mathrm{A}, \mathrm{B}) \to \mathrm{C}$

1. Create: Upon input register A from the sender, send "create" to the receiver and store the contents of register A.
2. 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.

It is clear from the description of these ideal functionalities that they may be called only a single time.

Functionalities 1–3 are sender-oblivious since they take an input from the sender and an input from the receiver and deliver the result of the functionality to the receiver (but not the sender). Moreover, they are non-reactive since they interact with the sender and the receiver in a single round.

2.3 Classical OTPs

The construction used in the author's paper heavily relies on Classical OTPs as desribed by Goyal et. al [Goyal et al. (2010)]

Theorem: Let f be a non-reactive, sender-oblivious, polynomial-time computable classical two- party functionality. Then there exists an efficient, non-interactive protocol which statistically classical-UC-emulates $\mathcal{F}_f^{\text{OTP}}$ in the $\mathcal{F}^{\text{OTM}}$-hybrid model.

3 Constructing Quantum OTPs from OTMs

We first begin by stating the important theorem in this section and then explaining the various building blocks for the proof. Let Φ be a non-reactive, sender-oblivious polynomial-time quantum computable two-party functionality. Then there exists an efficient, quantum non-interactive protocol which statistically quantum-UC-emulates $\mathcal{F}_{\Phi}^{OTP}$ in the case of a corrupt reciever in the $\mathcal{F}^{OTM}$- hybrid model. The building blocks for such a functionality are:

1. Quantum Authentication codes
2. Quantum Computation on Authenticated Data
3. Gate teleportation

3.1 Quantum Authentication Codes

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, Crepeau, Gottesman, Smith and Tap[Howard Barnum (2002).

Some of the known quantum authentication schemes have the following general form. Quantum information is encoded according to some quantum error detecting code E chosen uniformly at random from a special family E of codes. The encoded quantum data is then encrypted according to the quantum one-time pad, meaning that a uniformly random Pauli operation P is applied to the data encoded under E. The secret classical key for schemes of this form is the pair (E, P) describing the choice of code E and encryption Pauli P . Authenticated quantum data is later verified by decrypting according to P and then decoding according to E. Verification passes only if the error syndrome for E indicates no errors.

This construction is desirable due to the remarkable property (known as the Pauli twirl) that the Pauli encryption serves to render any attack on the scheme equivalent to a probabilistic Pauli attack on data encoded with a random code E $\in \epsilon$. Thus, to establish a secure authentica- tion scheme one need only construct a family ϵ of quantum error detecting codes that detect Pauli attacks with high probability over the choice of E $\in \epsilon$.

The author's new trap authentication scheme falls in this family of codes. The family of codes ϵ is based on any quantum error detecting code C with distance d that encodes a single qubit into n

qubits. Authentication consists of first encoding a qubit under C and then appending n qubits set to $|0\rangle$ and n qubits set to $|+\rangle$. A random permutation (indexed by a classical key) is then applied.

3.2 Computing on Authenticated Data

At a high level, the main protocol uses quantum authentication codes in order to protect the data from any tampering by the receiver during the computation. An authentication code is insufficient, however, because we want to implement a channel on this authenticated data, as specified by a quantum circuit.

For this, the authors use techniques for quantum computing on authenticated data. Computing on authenticated data allows acting on the encoded registers in order to implement a known gate, but without knowledge of the key. Normally, any non-identity operation would invalidate the authenticated state, but their encoded operations, together with a key update operation, effectively forces the application of the desired gate, as otherwise the state would fail verification under the updated key.

Encoded gates are executed in a manner similar to encoded gates in fault-tolerant quantum computation: some gates (such as Pauli gates or the controlled-not gate) are transversal, while other (such as the /8 gate) require both an auxiliary register and classical interaction with an entity who knows the encoding keys. This classical interaction makes the quantum one-time program "interactive", but only at a classical level. Thus, by extending classical one-time programs to reactive functionalities, the authors manage to encapsulate this interaction into a classical one-time program.

3.3 Gate Teleportation

The main outline of the protocol is now becoming clearer: the receiver executes the encoded circuit, using techniques for computing on authenticated data. But how does the receiver get the authenticated version of her data in the first place? And how does the receiver get the decoded output?

The scheme resolves this by using encoding and decoding gadgets that are inspired by the gate teleportation technique of Gottesman and Chuang. In this technique, a quantum register undergoes a transformation by a quantum circuit via its teleportation through a special entangled state. In our case, encoding is performed by teleporting the input qubit through an EPR pair, half of which has itself undergone the encoding operation. By revealing the classical result of the teleportation, the authentication key for the output of this process is determined. Decoding is similar. The encoding and decoding gadgets are prepared by the sender as part of the quantum one-time program.

4 The trap authentication scheme

In this section, lets introduce a novel quantum authentication method termed the trap scheme and make the case that it allows for quantum computation on verified data (QCAD). A quantum authentication technique entails methods for encoding and decoding quantum information with a secret classical key k in such a way that an adversary who tampers with encoded data without knowing k will most likely be caught.
The trap scheme is based on any fixed quantum error-detecting code C, which converts one logical qubit into n physical qubits with a distance of d. (an [[n,1,d]] code). Such codes produce various trap schemes. Protocol 1 specifies the authentication and de-authentication procedures for the trap scheme

based on a code C.

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 $|0\rangle^{\otimes n}, |1\rangle^{\otimes n}$, 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".

The trap scheme is an example of a family of authentication methods that we refer to as encode-encrypt schemes since it uses encoding and encryption as the first two steps in the authentication process. The fact that an arbitrary attack on such a scheme is equivalent to a probabilistic mixture of Pauli attacks on the underlying family E of codes is one of the many desirable aspects of encode-encrypt schemes. In order to create a safe quantum authentication technique using the encode-encrypt mechanism, it is sufficient to display a family E of codes that is resistant to Pauli assaults.

The family E in the trap scheme is made up of all codes obtained by permuting the data encoded under C along with the registers in the states $|0\rangle^{\otimes n}, |1\rangle^{\otimes n}$. The group of trap codes we refer to as E. The Shor-Preskill security proof for the distribution of quantum keys included an implicit usage of these codes. We create the defences of this family against Pauli assaults, and from there, the defences of the trap design.

Proposition 1 (Security of trap codes against Pauli attacks). *The family $\mathcal{E}$ 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 $E \in \mathcal{E}$- that Q acts non-trivially on logical data and yet has no error syndorme is at most $(2/3)^{\frac{d}{2}}$.

4.1 The trap scheme admits quantum computing on authenticated data

For a wide range of cryptographic applications, authentication systems that also permit QCAD—the application of a universal set of quantum gates on authenticated data without knowing the key—hold enormous potential. In this part, we make the case that the trap system permits QCAD when the underlying code C is properly chosen.

It can be helpful to picture two parties: a reliable validator who creates authenticated data using a secret classical key, k, and an evil attacker who will use the data without knowing about k. The objective is to create a system with the property that an attacker can apply a gadget circuit $\tilde{G}$ on authenticated data in order to implement a logical G for any gate G belonging to some universal set of gates. Additionally, we need that the device $\tilde{G}$ be independent of the choice of the classical key k in order for an attacker to implement it without being aware 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 $k \to k'$. 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.*, gadget design for our trap scheme is inspired by methods for fault-tolerant quantum computation. Authors present gadgets for the universal gate set consisting of Pauli gates, controlled-NOT, Hadamard, i-shift phase $K : |a\rangle \to i^a |a\rangle$, and $\frac{\pi}{8}$-phase $T : |a\rangle \to e^{ai\pi/4} |a\rangle$.

Certain gates, like the controlled-NOT, admit simple bitwise gates. Others, like the $\pi/8$ gate, call for the capacity to measure authenticated data in the computational basis and authenticated "magic states." In order for the attacker to complete these gadgets, the verifier must interpret the results of the conventional measurement for him. Therefore, these devices call for the traditional interaction of the verifier and attacker.

5 Protocol for quantum one-time programs

In this section, we offer their protocol for the quantum BR-OTP hybrid model's one-time quantum programs. We specifically outline how a trustworthy sender should set up her quantum registers and BR-OTP for the receiver as well as how a trustworthy receiver ought to use these objects to deduce the action of. The protocol is fully agnostic of the specific scheme selected, but it does require an encode-encrypt scheme that admits QCAD, such as the trap scheme provided in Section 3.

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

One can quickly locate a circuit for the controlled-U operation, abbreviated c-U, given a circuit U. This circuit operates on registers (A, B, and E) plus one additional control qubit that we conveniently

bundle into the auxiliary register E. According to the protocol, the control qubit should always be initialised to the $|on\rangle$ state when the receiver applies c-U to authenticated data. This technicality is there to make the security proving process easier.

5.1 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 $\tilde{A}, B_{in}, \tilde{B}_{in}, B_{out}, \tilde{B}_{out}, \tilde{E}, \tilde{M} = (\tilde{M}_1, \tilde{M}_2, \ldots \tilde{M}_r)$.
2. An (r+1)-round BR-OTP.

The sender prepares these objects as specified in protocol 2.

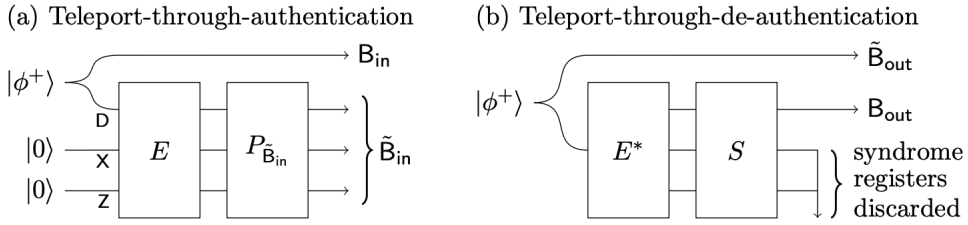

Fig. 1. Circuits for teleporting through authentication and de-authentication. Here the Pauli operations $P_{\tilde{B}_{in}}, P_{\tilde{B}_{out}}$ refer to the protions of P acting on registers $\tilde{B}_{in}, \tilde{B}_{out}$, respectively. source:- Broadbent et al. (2013a)

Protocol 2 Message preparation for an honest sender

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

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

- $(B_{in}, \tilde{B}_{in})$: Teleport-through-authentication state of Figure 1(a).
- $(B_{out}, \tilde{B}_{out})$: Teleport-through-de-authentication state of Figure 1(b).
- $\tilde{A}$: Authenticated input register A.
- $\tilde{E}$: Authenticated ancilla in logical state $|0\rangle\,|on\rangle$.
- $\tilde{M}$: Authenticated ancilla in logical state $ket\mu = |\mu_1\rangle \ldots |\mu_r\rangle$ 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 puported measurement result of the ith authenticated magic state register M_i .

   (b) Decode c_i into a classical bit a_i as dedicated 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.

5.2 Protocol for an honest receiver

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

---

Protocol 3 Protocol for an honest sender

---

1. Perform a Bell measurement on (B, Bin) 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 $\tilde{B}_{in}$.]*

2. Apply a logical c-U to the authenticated registers $(\tilde{A}\tilde{B}_{in}, \tilde{E}, \tilde{M})$. Explicitly:

   (a) Apply the gates of c-U occuring before the firs magic sate measurement.

   (b) For $i = 1, 2, \ldots, r$:

      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 $(\tilde{A}\tilde{B}_{in}, \tilde{E}, \tilde{M})$ holds the authentication version of (A,B,E) with c-U applied.]

3. Perform a Bell measurement on $(\tilde{B}_{in}, \tilde{B}_{out})$ 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 B_{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 Sˆ. Apply this Pauli to B_{out} to recover the output of Φ.

6 Security of quantum one-time programs

A QOTP for a channel Φ is logically considered secure if whatever that any (potentially dishonest) recipient could learn by analysing the program state prepared by the sender could also be learned by interacting with a simulator that uses only one-time access to an idealised black box for Φ. No receiver

can therefore learn anything more than what may be deduced from this optimum functioning for Φ. Formally, Authors use Unruh's definition of security for the quantum UC framework. Authors primary ideal feature, FOTP, involves both the sender and the receiver and is described in Functionality 1. Multiple instances of the functionality may exist, and different parties may be involved.

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

1. **Create:** Upon input register A form 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.

Functionality. A *non-interactive* protocol for evaluation of a channel $\Phi : (A, B) \to C$ consists of (i) an encoding channel $enc : A \to P$ applied by the sender on its input A that prepares a program state P, and (ii) a decoding channel $dec : (P, B) \to C$ applied by the receiver on the program state P and its input B such that $dec \circ enc$ and Φ are indistinguishable.
Security. To demonstrate security, it is sufficient to take into account simply the adversary that transmits messages between the environment and the trustworthy parties due to the completeness of the dummy-adversary. (we can see the environment as performing the attack). As a result, in the BR-OTP-quantum hybrid model, the security of a non-interactive protocol for the evaluation of Φ corresponds to the existence of a simulator that can mimic the sender's message in conjunction with the interactive behavior of the BR-OTP, using only a single, fixed-register A, one-shot access to the Φ black-box.

7 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 (A, B, 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 (A, B, 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 $\tilde{E}$ to "switch off" the application of U 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.

Protocol 4 Simulator

Secret classical key. Authentication key (E, P) for registers $\tilde{A}, \tilde{B}_{in}, \tilde{B}_{out}, \tilde{E}, \tilde{M}$ as in Protocol 2.
Registers prepared by the simulator. Given the input register A, the simulator constructs the following registers:

- (B_{in}, S_{in}) : Simple EPR pairs $|\phi^+\rangle$ for teleportation.
- $(S_{out}, \tilde{B}_{in})$: Teleport-through-authentication state of figure 1(a).
- $(B_{out}, \tilde{B}_{out})$: Teleport-through-de-authentication state of Figure 1(b).
- $\tilde{A}$: Authenticated dummy input register in logical state $|0\rangle$
- $\tilde{E}$: Authenticated dummy ancilla in logical state $|0\rangle\,|off\rangle$.
- $\tilde{M}$: 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 $B_{in}, \tilde{B}_{in}, \tilde{B}_{out}, B_{out}, \tilde{A}, \tilde{E}, \tilde{M}$ 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 (A, S_{in}, E).
3. Perform a Bell measurement on (S_{in}, S_{out}) so as to teleport the contents of S_{in} through authentication and place the result in $\tilde{B}_{in}$. 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.

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

Proof (sketch). As discussed in Section 5, 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 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.)

8 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 two complementary results on channels that admit one-time programs in the plain model.

The possibility result (Theorem 3) is that every strongly unlockable channel admits a trivial one-time program in the plain quantum model, and in fact that this protocol is UC-secure.

The impossibility result (Theorem 4) is that every channel that is not weakly unlockable does not admit a one-time program in the plain quantum model.

8.1 Definitions of Unlockability

Informally, a function or channel is unlockable if there is a key input for the receiver that unlocks enough information to fully simulate the map. For quantum channels the authors present two variants of unlockability, the difference being whether the key that unlocks the channel is a state (strongly unlockable) or a channel that transforms a given input (weakly unlockable).

Definition 1 (Strongly/weakly unlockable channels). A channel $\Phi : (\mathrm{A}, \mathrm{B}) \to \mathrm{C}$ is strongly unlockable if there exists a register K, a key state ξ_0 of (B, K) and a recovery algorithm (i.e., channel) $\mathcal{A} : (\mathrm{C}, \mathrm{K}, \mathrm{B}) \to \mathrm{C}$ with the property that $\mathcal{A} \circ \Phi_0 \approx \Phi$, where the channel Φ_0 is specified by $\Phi_0 : \mathrm{A} \to (\mathrm{C}, \mathrm{K}) : A \mapsto (\Phi \otimes 1_{\mathrm{K}})(A \otimes \xi_0)$

A channel $\Phi : (\mathrm{A}, \mathrm{B}) \to \mathrm{C}$ is weakly unlockable if there exists a register K and a key channel $\Xi_0 : \mathrm{B} \to (\mathrm{B}, \mathrm{K})$ such that the channel $\Phi \circ \Xi_0$ has the following property: for every choice of registers E and channels $\Psi : \mathrm{B} \to (\mathrm{B}, \mathrm{E})$ for the receiver there exists a recovery algorithm (i.e., channel) $\mathcal{A}_\Psi : (\mathrm{C}, \mathrm{K}) \to (\mathrm{C}, \mathrm{E})$ such that $\mathcal{A}_\Psi \circ \Phi \circ \Xi_0 \approx \Phi \circ \Psi$.

See Figure 2 for graphical depictions of these definitions. Here, $\approx$ can denote perfect, statistical, or (for polynomial-time uniform families of channels $\{\Phi_n\}$) computational indistinguishability; in all cases, channels $\Phi_0, \mathcal{A}, \Xi_0$, and $\mathcal{A}_\Psi$ must have circuits of size polynomial in the size of the circuit for Φ.

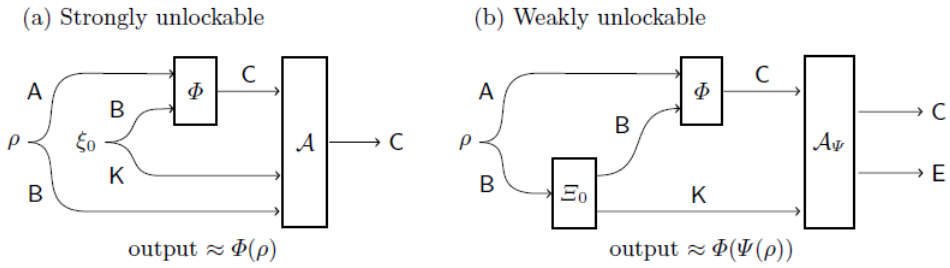

Fig. 2. (a) For a strongly unlockable channel Φ, there exists a key state ξ_0 and a recovery algorithm $\mathcal{A}$ that allows computation of $\Phi(\rho)$ for any ρ. (b) For a weakly unlockable channel Φ, there exists a key channel Ξ_0 such that for any channel Ψ there exists a recovery algorithm $\mathcal{A}_\Psi$ that allows computation of $\Phi(\Psi(\rho))$ for any ρ.

source:- Broadbent et al. (2013b)

It is easy to see that every strongly unlockable channel is also weakly unlockable: if ξ_0 is the key state for Φ, then the key channel Ξ_0 generates ξ_0, sends the B register of ξ_0 to ideal functionality Φ and the K register of ξ_0 and the B register of ρ to $\mathcal{A}_\Psi = \mathcal{A} \circ \Psi$.

Simple examples of strongly 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.

8.2 Trivial one-time programs for unlockable channels

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

Theorem 3 *Let $\Phi : (\mathrm{A}, \mathrm{B}) \to \mathrm{C}$ be a channel specified by a circuit. If Φ is strongly unlockable then there exists an efficient, quantum 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.*

8.3 Impossibility of one-time programs for arbitrary channels

Having seen that, in the plain model, strongly unlockable channels admit onetime programs, we now see that every channel which admits a one-time program must be weakly unlockable.

Theorem 4 *Let $\Phi : (\mathrm{A}, \mathrm{B}) \to \mathrm{C}$ 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 model. Then Φ is weakly unlockable. This holds in the perfect, statistical and computational cases.*

The intuition of the proof is as follows. If a channel has a one-time program, then for any adversary there exists a simulator that can match the behaviour of the adversary. In particular, there must be a simulator that matches the behaviour of the dummy-adversary that just outputs the program state: thus, there must be an algorithm that can reconstruct the program state given the output of the channel, thus allowing computation for any output, meeting the definition of a weakly unlockable channel. An alternate intuition for the impossibility result for classical functions 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.

8.4 A conjecture on unlockable channels

As noted earlier, every strongly unlockable channel is also weakly unlockable. The authors conjecture that the converse also holds. We do not yet have a formal proof of this conjecture for arbitrary Φ

Conjecture 1. Every channel $\Phi : (\mathrm{A}, \mathrm{B}) \to \mathrm{C}$ that is weakly unlockable is also strongly unlockable.

9 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. These works, however, do not consider composability. (Recently, Dunjko, Fitzsimons, Portmann and Renner showed the composability of the blind quantum computing protocol.)

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. Originally studied in the context of quantum interactive proof systems, the protocol of Aharonov et al.[Dorit Aharonov (2008)], 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.

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

---

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 $\{$execute, 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.

The ideal functionality we achieve is described in Functionality 2. Following , we describe the functionality in terms of a prover and 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-UCemulates $\mathcal{F}_{\Phi}^{\text{delegated}}$ 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.

References

Broadbent, A., Gutoski, G., Stebila, D., 2013a. Quantum one-time programs, 344–360.
URL https://arxiv.org/abs/1211.1080

Broadbent, A., Gutoski, G., Stebila, D., 2013b. Quantum one-time programs, 344–360.
URL https://doi.org/10.1007%2F978-3-642-40084-1_20

Dorit Aharonov, Michael Ben-Or, E. E., 2008. Interactive proofs for quantum computations.
URL https://arxiv.org/abs/0810.5375

Goldwasser, S., K. Y.-R. G., 2008. One-time programs. Wagner, D. (ed.) CRYPTO 5157 (52), 39–56.

Goyal, V., Ishai, Y., Sahai, A., Venkatesan, R., Wadia, A., 2010. Founding cryptography on tamper-proof hardware tokens. Cryptology ePrint Archive, Paper 2010/153, https://eprint.iacr.org/2010/153.
URL https://eprint.iacr.org/2010/153

Howard Barnum, Claude Crepeau, D. G. A. S. A. T., 2002. Authentication of quantum messages.
URL https://ieeexplore.ieee.org/document/1181969

Smid, M., 1981. Integrating the data encryption standard into computer networks. IEEE Transactions on Communications 29 (6), 762–772.