Quantum one-time programs
(short abstract for QCRYPT 2013)[*]

Anne Broadbent Gus Gutoski Douglas Stebila

June 20, 2013

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.

A *one-time program (OTP)* for a function f, as introduced in Ref. [8], is a cryptographic primitive by which a user may evaluate f on only one input chosen by the user at run time. 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 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; in particular, computational assumptions alone will not suffice.

Classically, it has been shown [8, 9] 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 $s_{\overline{c}}$ 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* [9], 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 $f(x, y)$. We use the term "OTP" interchangeably with "non-interactive secure two-party computation".

In this paper 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

[*] An extended abstract will appear in the proceedings of CRYPTO 2013 and is available at Cryptology ePrint Archive: Report 2013/343. A full version dated November 6, 2012 can be found at arXiv:1211.1080.

(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 $\Phi(\rho)$.

Can quantum one-time programs be constructed? If so, how? If not, why not, and under what additional assumptions can they be achieved? If they do exist, QOTPs would be useful for a variety of secure quantum computation tasks, such as providing *copy protection* of software [1] and implementing verification for quantum coin schemes [10]. (Note that QOTPs are different from the task of *program obfuscation*, which is known to be impossible classically [4] 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 short abstract elaborates upon these contributions.

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 he receives his input. As a corollary of our main result we obtain the UC-security of the protocol for *delegated quantum computations* (DQC) from Ref. [3]. Composable security for other variants of DQC was independently established in Ref. [6].

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 for public-key quantum money scheme of Ref. [2].

Prior to our work, the only authentication scheme known to admit QCAD was the *signed polynomial scheme* [5, 3]. Recently, and independently of our work, it was shown in Ref. [7] 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 [9] and is described in detail in the full version. 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.

3 Unlockable functions and channels

Curiously, our study has uncovered a pathological class 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 [4].

We propose a definition of unlockability and prove that a function f admits a one-time program in the plain quantum model if and only if it is unlockable. For quantum channels the situation is quite interesting. We define two classes of channels called *weakly* and *strongly* unlockable. We prove that every strongly unlockable channel admits a trivial one-time program in the plain quantum model. Conversely, we prove that any channel admitting a one-time program in the plain quantum model must be weakly unlockable. It is easy to see that every strongly unlockable channel is also weakly unlockable; we conjecture that the two classes are equal. To summarize, we prove that no "useful" function or channel admits a one-time program without any hardware assumptions.

Acknowledgements

We gratefully acknowledge helpful discussions with Harry Buhrman, Daniel Gottesman, 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. Part of this research conducted while D.S. was a visitor at the University of Waterloo's IQC.

References

[1] Scott Aaronson. Quantum copy-protection and quantum money. In *Proc. 24th IEEE Conference on Computational Complexity (CCC) 2009*, pages 229–242, 2009.

[2] Scott Aaronson and Paul Christiano. Quantum money from hidden subspaces. In *Proc. 44th Symposium on Theory of Computing (STOC) 2012*, pages 41–60, 2012. Full version available as arXiv:1203.4740 [quant-ph].

[3] Dorit Aharonov, Michael Ben-Or, and Elad Eban. Interactive proofs for quantum computations. In *Proc. Innovations in Computer Science (ICS) 2010*, pages 453–469, 2010.

[4] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In *CRYPTO 2001*, volume 2139 of *LNCS*, pages 1–18. Springer, 2001. Full version available at `http://www.wisdom.weizmann.ac.il/~oded/p_obfuscate.html`.

[5] Michael Ben-Or, Claude Crépeau, Daniel Gottesman, Avinatan Hassidim, and Adam Smith. Secure multiparty quantum computation with (only) a strict honest majority. In *FOCS 2006*, pages 249–260, 2006.

[6] Vedran Dunjko, Joseph F. Fitzsimons, Christopher Portmann, and Renato Renner. Composable security of delegated quantum computation. arXiv:1301.3662 [quant-ph], 2013.

[7] Frédéric Dupuis, Jesper Buus Nielsen, and Louis Salvail. Actively secure two-party evaluation of any quantum operation. In *CRYPTO 2012*, volume 7417 of *LNCS*, pages 794–811. Springer, 2012.

[8] Shafi Goldwasser, Yael Kalai, and Guy Rothblum. One-time programs. In *CRYPTO 2008*, volume 5157 of *LNCS*, pages 39–56. Springer, 2008.

[9] Vipul Goyal, Yuval Ishai, Amit Sahai, Ramarathnam Venkatesan, and Akshay Wadia. Founding cryptography on tamper-proof hardware tokens. In *TCC 2010*, volume 5978 of *LNCS*, pages 308–326. Springer, 2010. Full version available at `http://eprint.iacr.org/2010/153`.

[10] Michele Mosca and Douglas Stebila. Quantum coins. In *Error-Correcting Codes, Finite Geometries and Cryptography*, volume 523 of *Contemporary Mathematics*, pages 35–47. American Mathematical Society, 2010.