Towards Quantum One-Time Memories from Stateless Hardware

Anne Broadbent* Sevag Gharibian† Hong-Sheng Zhou‡

Abstract

A central tenet of theoretical cryptography is the study of the minimal assumptions required to implement a given cryptographic primitive. One such primitive is the one-time memory (OTM), introduced by Goldwasser, Kalai, and Rothblum [CRYPTO 2008], which is a classical functionality modeled after a non-interactive 1-out-of-2 oblivious transfer, and which is complete for one-time classical and quantum programs. It is known that secure OTMs do not exist in the standard model in both the classical and quantum settings.

Here, we propose a scheme for using quantum information, together with the assumption of stateless (*i.e.*, reusable) hardware tokens, to build statistically secure OTMs. Via the semidefinite programming-based quantum games framework of Gutoski and Watrous [STOC 2007], we prove security for a malicious receiver, against a linear number of adaptive queries to the token, in the quantum universal composability framework. We prove stand-alone security against a malicious sender, but leave open the question of composable security against a malicious sender, as well as security against a malicious receiver making a polynomial number of adaptive queries. Compared to alternative schemes derived from the literature on quantum money, our scheme is technologically simple since it is of the "prepare-and-measure" type. We also show our scheme is "tight" according to two scenarios.

1 Introduction

The study of theoretical cryptography is centered around the question of building cryptographic primitives secure against adversarial attacks. In order to allow a broader set of such primitives to be implemented, one often considers restricting the power of the adversary. For example, one can limit the *computing* power of adversaries to be polynomial bounded [Yao82, BM82], restrict the *storage* of adversaries to be bounded or noisy [Mau92, CM97, DFSS05], or make *trusted setups* available to honest players [Kil88, BFM88, Can01, CLOS02, IPS08, PR08, LPV09, MPR09, MPR10, MR11, KMQ11, KMPS14], to name a few. One well-known trusted setup is *tamper-proof hardware* [Kat07, GKR08], which is assumed to provide a specific input-output functionality, and which can only be accessed in a "black box" fashion. The hardware can maintain a state (*i.e.*, is *stateful*) and possibly carry out complex functionality, but presumably may be difficult or expensive to implement or manufacture. This leads to an interesting research direction: Building cryptography primitives using the *simplest* (and hence easiest and cheapest to manufacture) hardware.

*Department of Mathematics and Statistics, University of Ottawa, Ontario, Canada. Email: abroadbe@uottawa.ca.

†Department of Computer Science, University of Paderborn, Germany, and Virginia Commonwealth University, USA. Email: sevag.gharibian@gmail.com.

‡Department of Computer Science, Virginia Commonwealth University, Virginia, USA. Email: hszhou@vcu.edu.

In this respect, two distinct simplified notions of hardware have captured considerable interest. The first is the notion of a *one-time memory (OTM)* [GKR08], which is arguably the simplest possible notion of *stateful* hardware. An OTM, modeled after a non-interactive 1-out-of-2 oblivious transfer, behaves as follows: first, a player (called the *sender*) embeds two values s_0 and s_1 into the OTM, and then gives the OTM to another player (called the *receiver*). The receiver can now read his choice of precisely one of s_0 or s_1; after this "use" of the OTM, however, the unread bit is lost forever. Interestingly, OTMs are complete for implementing *one-time* use programs (OTPs): given access to OTMs, one can implement statistically secure OTPs for any efficiently computable program in the universal composability (UC) framework [GIS+10]. (OTPs, in turn, have applications in software protection and one-time proofs [GKR08].) In the quantum UC model, OTMs enable *quantum* one-time programs [BGS13]. (This situation is analogous to the case of *oblivious transfer* being complete for two-party secure function evaluation [Kil88, IPS08].) Unfortunately, OTMs are inherently *stateful*, and thus represent a very strong cryptographic assumption — any physical implementation of such a device must somehow maintain internal knowledge between activations, *i.e.*, it must completely "self-destruct" after a single use.

This brings us to a second important simplified notion of hardware known as a *stateless* token [CGS08], which keeps no record of previous interactions. On the positive side, such hardware is presumably easier to implement. On the negative side, an adversary can run an experiment with stateless hardware as many times as desired, and each time the hardware is essentially "reset". (Despite this, stateless hardware has been useful in achieving *computationally secure* multi-party computation [CGS08, GIS+10, CKS+14], and *statistically secure* commitments [DS13].) It thus seems impossible for stateless tokens to be helpful in implementing any sort of "self-destruct" mechanism. Indeed, classically stateful tokens are trivially more powerful than stateless ones, as observed in, *e.g.*, [GIS+10]. This raises the question:

> *Can* quantum *information, together with a classical stateless token, be used to simulate "self destruction" of a hardware token?*

In particular, a natural question along these lines is whether quantum information can help implement an OTM. Unfortunately, it is known that quantum information *alone* cannot implement an OTM (or, more generally, any one-time program) [BGS13]; see also Section 4 below. We thus ask the question: What are the minimal cryptographic assumptions required in a quantum world to implement an OTM?

Contributions and summary of techniques. Our main contribution is to propose a prepare-and-measure quantum protocol that constructs OTMs from stateless hardware tokens. We provide a proof which establishes information theoretic security against an adversary making a linear (in n, the security parameter) number of adaptive queries to the token. Even in this setting of linear queries, this result is in sharp contrast to the classical case, in which such a construction is trivially impossible. We also show stand-alone security against a malicious sender.

HISTORICAL NOTE. A preliminary version of this work [BGZ15] claimed security against a *polynomial* number of token queries. This was claimed via a reduction from the interactive to the non-interactive setting. We thank an anonymous referee for catching a subtle, but important bug in that proof attempt. For clarity, our current proof against a linear number of queries uses a different approach. Since our original paper was posted, recent related work [CGLZ18] has shown a quantum transformation from stateful to stateless tokens, directly using constructions and proofs

from the literature on quantum money [BDS18]. As far as we are aware, this family of schemes requires highly entangled states that do not satisfy the requirements of a prepare-and-measure scheme. This recent work also raises a concern about our security model involving an honest sender; this concern is addressed in Section 3.2 where we show stand-alone security against a malicious sender.

CONSTRUCTION. Our construction is inspired by Wiesner's idea for *conjugate coding* [Wie83]: the quantum portion of the protocols consists in n quantum states chosen uniformly at random from $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$ (note this encoding is independent of the classical bits of the OTM functionality). We then couple this n-qubit quantum state, $|\psi\rangle$ (sometimes called the *quantum key*) with a *classical* stateless hardware token, which takes as inputs a choice bit b, together with an n-bit string y. If $b = 0$, the hardware token verifies that the bits of y that correspond to *rectilinear* ($|0\rangle$ or $|1\rangle$, *i.e.*, Z basis) encoded qubits of $|\psi\rangle$ are consistent with the measurement of $|\psi\rangle$ in the computational basis, in which case the bit s_0 is returned. If $b = 1$, the hardware token verifies that the bits of y that correspond to *diagonal* ($|+\rangle$ or $|-\rangle$, *i.e.*, X basis) encoded qubits of $|\psi\rangle$ are consistent with the measurement of $|\psi\rangle$ in the diagonal basis, in which case the bit s_1 is returned.

ASSUMPTION. Crucially, the hardware token is specified to accept *classical* input only[1] (*i.e.*, it cannot be queried in superposition). Although this may seem a strong assumption, in Section 4.1 we show that any token which can be queried in superposition in a reversible way, cannot be used to construct a secure OTM (with respect to our setting in which the adversary is allowed to apply arbitrary quantum operations). Similar classical-input hardware has previously been considered in, e.g., [Unr13, BGS13].

SECURITY AND INTUITION. Stand-alone security against a malicious sender is relatively straightforward to establish, since the protocol consists in a single message from the sender to the receiver, and since stand-alone security only requires simulation of the *local* view of the adversary.

The intuition underlying security against a malicious receiver is clear: in order for a receiver to extract a bit s_b as encoded in the OTM, she must perform a complete measurement of the qubits of $|\psi\rangle$ in order to obtain a classical key for s_b (since, otherwise, she would likely fail the test as imposed by the hardware token). But such a measurement would invalidate the receiver's chance of extracting the bit s_{1-b}! This is exactly the "self-destruct"-like property we require in order to implement an OTM. This intuitive notion of security was already present in Wiesner's proposal for quantum money[2] [Wie83], and is often given a physical explanation in terms of the no-cloning theorem [WZ82], or the Heisenberg uncertainty relation [Hei27].

Formally, we work in the statistical (*i.e.*, information-theoretic) setting of the quantum *Universal Composability* (UC) framework [Unr10], which allows us to make strong security statements that address the *composability* of our protocol within others. As a proof technique, we describe a simulator, such that for any "quantum environment" wishing to interact with the OTM, the environment statistically cannot tell whether it is interacting with the *ideal* OTM functionality or the *real* OTM instance provided by our scheme. The security of this simulator requires a statement of the following form: Given access to a (randomly chosen) "quantum key" $|\psi_k\rangle$ and corresponding stateless token V_k, it is highly unlikely for an adversary to successfully extract keys for *both* the secret bits s_0 and s_1 held by V_k. We are able to show this statement for any adversary which makes a linear number of queries, by which we mean an adversary making m queries succeeds with probability at

[1] This can be simulated on quantum inputs by having the token immediately measure its input in the standard basis.

[2] Intuitively, quantum money aims to construct a physical currency which is impossible to counterfeit by the laws of quantum mechanics.

most $O(2^{2m-0.228n})$ (for n the number of quantum key bits in $|\psi_k\rangle$). In other words, if the adversary makes at most $m = cn$ queries with $c < 0.114$, then its probability of cheating successfully is exponentially small in n. To show this statement, we exploit the semidefinite programming-based quantum games framework of Gutoski and Watrous [GW07] to model interaction with the token. We describe this technique in Section 3.4 and provide the full details in the Appendix C.

Summarizing, we show the following.

Main Theorem (informal). *There exists a protocol Π, which together with a classical stateless token and the ability to randomly prepare single qubits in one of four pure states, implements the OTM functionality with statistical security in the UC framework against a corrupted receiver making a linear number of adaptive queries.*

We leave open the question of security against a polynomial number of adaptive queries.

Further Related work. Our work contributes to the growing list of functionalities achievable with quantum information, yet unachievable classically. This includes: unconditionally secure key expansion [BB84], physically uncloneable money [Wie83, MVW13, PYJ+12], a reduction from oblivious transfer to bit commitment [BBCS92, DFL+09] and to other primitives such as "cut-and choose" functionality [FKS+13], and revocable time-release quantum encryption [Unr14]. Importantly, these protocols all make use of the technique of conjugate coding [Wie83], which is also an important technique used in protocols for OT in the bounded quantum storage and noisy quantum storage models [DFSS05, WST08] (see [BS16] for a survey).

A number of proof techniques have been developed in the context of conjugate coding, including entropic uncertainty relations [WW10]. In the context of QKD, another successful technique is the use of de Finetti reductions [Ren08] (which exploit the symmetry of the scheme in order to simplify the analysis). Recently, semidefinite programming approaches have been applied to analyze security of conjugate coding [MVW13]. This is the type of approach we adopt for our proof (Section 3.4 and Appendix C), though here we work with the more general quantum games framework of Gutoski and Watrous [GW07]. Reference [PYJ+12] has also made use of Gavinsky's [Gav12] quantum retrieval games framework.

Continuing with proof techniques, somewhat similar to [PYJ+12], Aaronson and Christiano [AC12] have studied quantum money schemes in which one interacts with a verifier. They introduce an "inner product adversary method" to lower bound the number of queries required to break their scheme.

We remark that [PYJ+12] and [MVW13] have studied schemes based on conjugate coding similar to ours, but in the context of quantum money. In contrast to our setting, the schemes of [PYJ+12] and [MVW13] (for example) involve dynamically chosen random challenges from a verifier to the holder of a "quantum banknote", whereas in our work here the "challenges" are fixed (*i.e.*, measure all qubits in the Z or X basis to obtain secret bit s_0 or s_1, respectively), and the verifier is replaced by a stateless token.

Also, we note that prior work has achieved oblivious transfer using quantum information, together with some assumption (*e.g.*, bit commitment [BBCS92] or bounded quantum storage [DFSS05]). These protocols typically use an interaction phase similar to the "commit-and-open" protocol of [BBCS92]; because we are working in the non-interactive setting, these techniques appear to be inapplicable.

Finally, Liu [Liu14a, Liu14b, Liu15] has given stand-alone secure OTMs using quantum information. In contrast to our setting, in which we allow unbounded and unrestricted quantum

adversaries, Liu's results are set in the *isolated-qubit model*, which assumes that an adversary can perform only single-qubit operations (no entangling gates are permitted). However, in exchange for restricting the adversary, Liu is able to avoid the use of trusted setups. The security notion of OTMs by Liu is weaker than the simulation-based notion that is studied in this paper, and it is unclear whether this type of OTM is composable.

Significance. Our results show a strong separation between the classical and quantum settings, since classically, stateless tokens cannot be used to securely implement OTMs. To the best of our knowledge, our work is the first to combine conjugate coding with *stateless* hardware tokens. Moreover, while our protocol shares similarities with prior work in the setting of quantum money, building OTMs appears to be a new focus here [3].

Our protocol has a simple implementation, fitting into the single-qubit prepare-and-measure paradigm (one needs only the ability to prepares states $|0\rangle, |1\rangle, |+\rangle, |-\rangle$). In addition, from a theoretical cryptographic perspective, our protocol is attractive in that its implementation requires an assumption of a stateless hardware token, which is conceivably easier and cheaper to mass produce than a stateful token.

In terms of security guarantees, we allow *arbitrary* operations on behalf of a malicious quantum receiver in our protocol (*i.e.*, all operations allowed by quantum mechanics), with the adversary restricted in that the stateless token is assumed only usable as a black box. The security we obtain is statistical, with the only computational assumption being on the number of *queries* made to the token.Finally, our proofs are rigorous statements in the quantum UC framework, meaning our protocol can be easily composed with others proved secure in this framework (*e.g.*, combining our results with [BGS13]'s protocol immediately yields UC-secure quantum OTPs against a dishonest receiver).

We close by remarking that our scheme is "tight" with respect to two impossibility results, both of which assume the adversary has black-box access to both the token and its inverse operation [4]. First, the assumption that the token be queried only in the computational basis cannot be relaxed: Section 4.1 shows that if the token can be queried in superposition, then an adversary in our setting can easily break any OTM scheme. Second, our scheme has the property that corresponding to each secret bit s_i held by the token, there are exponentially many valid keys one can input to the token to extract s_i. In Section 4.2, we show that for any "measure-and-access" OTM (*i.e.*, an OTM in which one measures a given quantum key and uses the classical measurement result to access a token to extract data, of which our protocol is an example), a polynomial number of keys implies the ability to break the scheme with inverse polynomial probability (more generally, Δ keys allows probability at least $1/\Delta^2$ of breaking the scheme).

Open Questions. While our work shows the fundamental advantage quantum information yields in a stateful to stateless reduction, it does leave a number of open questions:

1. **Security against polynomially many queries.** Can our security proof be strengthened to show information theoretic security against a polynomial number of queries to the token? We conjecture this to be the case, but finding a formal proof has been elusive.

[3] We remark, however, that a reminiscent concept of single usage of quantum "tickets" in the context of quantum money is very briefly mentioned in Appendix S.4.1 of [PYJ+12].

[4] This is common in quantum computation, where a function $f : \{0,1\}^n \mapsto \{0,1\}$ is typically implemented via a unitary U_f acting on $n+1$ qubits, and satisfying $U_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle$. By definition, U_f is self-inverse; thus, a user with access to U_f trivially also has access to its inverse $U_f^\dagger = U_f$.

2. **Composable security against a malicious sender.** While we show composable security against a malicious receiver, our protocol can achieve standalone security against a malicious sender. Could an adaptation of our protocol ensure composable security against a malicious sender as well?

3. **Non-reversible token.** Our impossibility result for quantum one-time memories with *quantum* queries (Section 4) assumes the adversary has access to reversible tokens; can a similar impossibility result be shown for non-reversible tokens?

4. **Imperfect devices.** While our prepare-and-measure scheme is technologically simple, it is still virtually unrealizable with current technology, due to the requirement of perfect quantum measurements. We leave open the question of tolerance to a small amount of noise.

Acknowledgements. We thank anonymous referees for pointing out that the impossibility result against quantum queries applies only if we model the token as a *reversible* process, as well as for finding an error in a prior version of this work, which erroneously claimed a reduction from the adaptive to non-adaptive case. We thank Jamie Sikora for related discussions. This material is based upon work supported by the U.S. Air Force Office of Scientific Research under award number FA9550-17-1-0083, Canada's NSERC, an Ontario ERA, and the University of Ottawa's Research Chairs program.

Organization of the paper. The remainder of the paper is organized as follows. We begin in Section 2 with preliminaries, including the ideal functionalities for an OTM and stateless token. In Section 3, we give our construction for an OTM based on a stateless hardware token; the proof ideas for security are also provided. In Section 4, we discuss "tightness" of our construction by showing two impossibility results for "relaxations" of our scheme. In the Appendix, we include the description of classical UC and quantum UC (Appendix A); Appendix B establishes notation required in the definition of stand-alone security against a malicious sender. Appendix C gives our formal security proof against a linear number of queries to the token; these results are used to finish the security proof in Section 3.) In addition, the security proof for a lemma in Section 4 can be found in Appendix D.

2 Preliminaries

Notation. We say two binary distributions $\mathbf{X}$ and $\mathbf{Y}$ are *indistinguishable*, denoted $\mathbf{X} \approx \mathbf{Y}$, if it holds that $|\Pr(X_n = 1) - \Pr(Y_n = 1)| \leq \mathrm{negl}(n)$. We define single-qubit $|0\rangle_+ = |0\rangle$ and $|1\rangle_+ = |1\rangle$, so that $\{|0\rangle_+, |1\rangle_+\}$ form the *rectilinear basis*. We also define $|0\rangle_\times = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|1\rangle_\times = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, so that $\{|0\rangle_\times, |1\rangle_\times\}$ form the *diagonal basis*. For strings $x = x_1, x_2, \ldots x_n \in \{0,1\}^n$ and $\theta = \theta_1, \theta_2, \ldots, \theta_n \in \{+, \times\}^n$, we define $|x\rangle_\theta = \bigotimes_{i=1}^n |x_i\rangle_{\theta_i}$. Finally, H denotes the standard 2×2 Hadamard gate $H = (1\ 1;\ 1\ -1)/\sqrt{2}$ in quantum information.

Quantum universal composition (UC) framework. We consider simulation-based security in this paper. In particular, we prove the security of our construction against a malicious receiver in the quantum universal composition (UC) framework [Unr10]. Please see Appendix A for a brief description of the classical UC [Can01] and the quantum UC [Unr10]. In the next two paragraphs, we introduce two relevant ideal functionalities of one-time memory and of stateless hardware token.

One-time memory (OTM). The one-time memory (OTM) functionality $\mathcal{F}_{\texttt{OTM}}$ involves two parties, the sender and the receiver, and consists of two phases, "Create" and "Execute". Please see Functionality 1 below for details; for the sake of simplicity, we have omitted the session/party identifiers as they should be implicitly clear from the context. We sometimes refer to this functionality $\mathcal{F}_{\texttt{OTM}}$ as an *OTM token*.

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

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 $b \in \{0,1\}$ from the receiver, send s_b to receiver. Delete any trace of this instance.

Stateless hardware. The original work of Katz [Kat07] introduces the ideal functionality $\mathcal{F}_{\texttt{wrap}}$ to model stateful tokens in the UC-framework. In the ideal model, a party that wants to create a token, sends the Turing machine to $\mathcal{F}_{\texttt{wrap}}$. $\mathcal{F}_{\texttt{wrap}}$ will then run the machine (keeping the state), when the designated party will ask for it. The same functionality can be adapted to model stateless tokens. It is sufficient that the functionality does not keep the state between two executions. A simplified version of the $\mathcal{F}_{\texttt{wrap}}$ functionality as shown in [CGS08] (that is very similar to the $\mathcal{F}_{\texttt{wrap}}$ of [Kat07]) is described below. Note that, again for the sake of simplicity, we have omitted the session/party identifiers as they should be implicitly clear from the context.

Although the environment and adversary are unbounded, we specify that stateless hardware can be queried only a polynomial number of times. This is necessary, since otherwise the hardware token model is vacuous (with unbounded queries, the entire input-output behavior of stateless hardware can be deduced, hence there is nothing left to hide).

Functionality 2 Ideal functionality $\mathcal{F}_{\texttt{wrap}}$.

The functionality is parameterized by a polynomial $p(\cdot)$, and an implicit security parameter n.

1. **Create:** Upon input (create, M) from the sender, where M is a Turing machine, send create to the receiver and store M.
2. **Execute:** Upon input (run, msg) from the receiver, execute $M(msg)$ for at most $p(n)$ steps, and let *out* be the response. Let $out := \perp$ if M does not halt in $p(n)$ steps. Send *out* to the receiver.

3 Feasibility of Quantum OTMs using Stateless Hardware

In this section, we present a *quantum* construction for one-time memories by using stateless hardware (Section 3.1). We also state our main theorem (Theorem 3.3). In Section 3.3, we describe the Simulator and prove Theorem 3.3 using the technical results of Appendix C. The intuition and techniques behind the proofs in Appendix C are sketched in Section 3.4.

3.1 Construction

We now present the OTM protocol Π in the $\mathcal{F}_{\texttt{wrap}}$ hybrid model, between a sender P_{s} and a receiver P_{r}. Here the security parameter is n.

- Upon receiving input (s_0, s_1) from the environment where $s_0, s_1 \in \{0,1\}$, sender P_{s} operates as follows:
  - The sender chooses uniformly random strings $x \in_R \{0,1\}^n$ and $\theta \in_R \{+,\times\}^n$, and prepares $|x\rangle_\theta$. Then the sender, based on tuple (s_0, s_1, x, θ), prepares the program M as in **Program 1**.

Program 1 Program for hardware token

Hardcoded values: $s_0, s_1 \in \{0,1\}$, $x \in \{0,1\}^n$, and $\theta \in \{+,\times\}^n$
Inputs: $y \in \{0,1\}^n$ and $b \in \{0,1\}$, where y is a claimed measured value for the quantum register, and b the evaluator's choice bit

1. If $b = 0$, check that the $\theta = +$ positions return the correct bits in y according to x. If Accept, output s_0. Otherwise output $\perp$.
2. If $b = 1$, check that the $\theta = \times$ positions return the correct bits in y according to x. If Accept, output s_1. Otherwise output $\perp$.

  - The sender sends $|x\rangle_\theta$ to the receiver.
  - The sender sends (create, M) to functionality $\mathcal{F}_{\texttt{wrap}}$, and the functionality sends create to notify the receiver.

- The receiver P_{r} operates as follows:
  Upon input b from the environment, and $|x\rangle_\theta$ from the receiver, and create notification from $\mathcal{F}_{\texttt{wrap}}$,
  - If $b = 0$, measure $|x\rangle_\theta$ in the computational basis to get string y and input $(\mathsf{run}, (y, b))$ into $\mathcal{F}_{\texttt{wrap}}$.
  - If $b = 1$, apply $\mathrm{H}^{\otimes n}$ to $|x\rangle_\theta$, then measure in the computational basis to get string y and input $(\mathsf{run}, (y, b))$ into $\mathcal{F}_{\texttt{wrap}}$.

  Return the output of $\mathcal{F}_{\texttt{wrap}}$ to the environment.
  It is easy to see that the output of $\mathcal{F}_{\texttt{wrap}}$ is s_b for both $b = 0$ and $b = 1$.

Note again that the hardware token, as defined in **Program 1**, accepts only classical input (*i.e.*, it cannot be queried in superposition). As mentioned earlier, relaxing this assumption yields impossibility of a secure OTM implementation (assuming the receiver also has access to the token's inverse operation), as shown in Section 4.

3.2 Stand-Alone Security Against a Malicious Sender

We note that in protocol Π of Section 3.1, once the sender prepares and sends the token, she is no longer involved (and in particular, the sender does not receive any further communication from the receiver). We call such a protocol a *one-way* protocol. Because of this simple structure, and because the ideal functionality $\mathcal{F}_{\texttt{wrap}}$ also does not return any message to the sender, we can easily establish

stand-alone security against a malicious sender (see Theorem 3.2). Note that this rules out a *trivial* construction that is pointed out in [CGLZ18].

In order to define stand-alone security against a malicious sender (Definition 3.1), in our context, we closely follow definitions given in prior work [DNS10]. Please see Appendix B for details. Note that, instead of considering the *approximate* case for security, we are able to use the *exact* one.

Definition 3.1. *An n-step quantum two-party protocol with oracle calls,* $\Pi^{\mathcal{O}} = (\mathscr{A}, \mathscr{B}, \mathcal{O}, n)$ *is statistically* stand-alone *secure against a corrupt* $\mathscr{A}$ *if for every adversary* $\tilde{\mathscr{A}}$ *there exists a simulator* $\mathcal{S}$ *such that for every input* ρ_{in}*,*

$$\mathrm{Tr}_{\mathcal{B}_n \otimes \mathcal{R}}(\tilde{\mathscr{A}} \circledast \mathscr{B}) = \mathrm{Tr}_{\mathcal{B}_n \otimes \mathcal{R}}(\mathcal{S} \circledast \mathscr{B})\,. \tag{1}$$

We note that Definition 3.1 is weaker than some other definitions for active security used in the literature, *e.g.*, [DNS12], because we ask only that the *local* view of the adversary be simulated.

Given the simple structure of our protocol and ideal functionality, the construction and proof of the simulator is straightforward as shown below.

Theorem 3.2. *Protocol* Π *is statistically stand-alone secure against a corrupt sender.*

Proof. Since Π consists in a single message from the sender to the receiver (together with a call to the ideal functionality for the token), we have that $\mathscr{A} = (\mathscr{A}_1)$. Furthermore, since the ideal functionality $\mathcal{F}_{\texttt{wrap}}$ does not return anything to the sender, there is no need for our simulator $\mathcal{S}$ to call an ideal functionality.

We thus build $\mathcal{S}$ that runs $\mathscr{A}$ on the input in register $\mathcal{A}_0$. When $\mathscr{A}$ calls the $\mathcal{F}_{\texttt{wrap}}$ ideal functionality, the simulator does nothing. Since Π is a one-way protocol, and since the ideal functionality also does not allow communication from the receiver to the sender,

$$\mathrm{Tr}_{\mathcal{B}_n \otimes \mathcal{R}}(\tilde{\mathscr{A}} \circledast \mathscr{B}) = \mathscr{A}(Tr_{\mathcal{B}_0 \otimes \mathcal{R}}(\rho_{\mathrm{in}})) = \mathcal{S}(Tr_{\mathcal{B}_0 \otimes \mathcal{R}}(\rho_{\mathrm{in}}))\,. \tag{2}$$

This concludes the proof. □

3.3 UC-Security against a corrupt receiver

Our main theorem, which establishes security against a corrupt receiver is now stated as follows.

Theorem 3.3. *Construction* Π *above quantum-UC-realizes* $\mathcal{F}_{\texttt{OTM}}$ *in the* $\mathcal{F}_{\texttt{wrap}}$ *hybrid model with statistical security against an actively-corrupted receiver making at most* cn *number of adaptive queries to the token, for any fixed constant* $c < 0.114$*.*

To prove Theorem 3.3, we must construct and analyze an appropriate simulator, which we now proceed to do.

3.3.1 The simulator

In order to prove Theorem 3.3, for an adversary $\mathcal{A}$ that corrupts the receiver, we need to build a simulator $\mathcal{S}$ (having access to the OTM functionality $\mathcal{F}_{\texttt{OTM}}$), such that for any unbounded environment $\mathcal{Z}$, the executions in the real model and that in simulation are statistically indistinguishable. Our simulator $\mathcal{S}$ is given below:

- The simulator emulates an internal copy of the adversary $\mathcal{A}$ who corrupts the receiver. The simulator emulates the communication between $\mathcal{A}$ and the external environment $\mathcal{Z}$ by forwarding the communication messages between $\mathcal{A}$ and $\mathcal{Z}$.
- The simulator $\mathcal{S}$ needs to emulate the whole view for the adversary $\mathcal{A}$. First, the simulator picks dummy inputs $\tilde{s}_0 = 0$ and $\tilde{s}_1 = 0$, and randomly chooses $x \in \{0,1\}^n$, and $\theta \in \{+,\times\}^n$, and generates program $\tilde{M}$. Then the simulator plays the role of the sender to send $|x\rangle_\theta$ to the adversary $\mathcal{A}$ (who controls the corrupted receiver). The simulator also emulates $\mathcal{F}_{\texttt{wrap}}$ to notify $\mathcal{A}$ by sending create to indicate that the hardware is ready for queries.
- For each query $(\mathsf{run}, (b, y))$ to $\mathcal{F}_{\texttt{wrap}}$ from the adversary $\mathcal{A}$, the simulator evaluates program $\tilde{M}$ (that is created based on $\tilde{s}_0, \tilde{s}_1, x, \theta$) as in the construction, and then acts as follows:
  1. If this is a rejecting input, output $\perp$.
  2. If this is the first accepting input, call the external $\mathcal{F}_{\texttt{OTM}}$ with input b, and learn the output s_b from $\mathcal{F}_{\texttt{OTM}}$. Output s_b.
  3. If this is a subsequent accepting input, output s_b (as above).

3.3.2 Analysis

We now show that the simulation and the real model execution are statistically indistinguishable. There are two cases in an execution of the simulation which we must consider:

- *Case 1: In all its queries to $\mathcal{F}_{\texttt{wrap}}$, the accepting inputs of $\mathcal{A}$ have the same choice bit b.* In this case, the simulation is perfectly indistinguishable.
- *Case 2: In its queries to $\mathcal{F}_{\texttt{wrap}}$, $\mathcal{A}$ produces accepting inputs for both $b = 0$ and $b = 1$.* In this case, it is possible that the simulation fails (the environment can distinguish the real model from the ideal model), since the simulator is only able to retrieve a single bit from the external OTM functionality $\mathcal{F}_{\texttt{OTM}}$ (either corresponding to $b = 0$ or $b = 1$).

Thus, whereas in Case 1 the simulator behaves perfectly, in Case 2 it is in trouble. Fortunately, in Theorem 3.4 we show that the probability that Case 2 occurs is exponentially small in n, the number of qubits comprising $|x\rangle_\theta$, provided the number of queries to the token scales as cn for a sufficiently small fixed constant $c > 0$. Specifically, we show that for an arbitrary m-query strategy (*i.e.*, any quantum strategy allowed by quantum mechanics, whether efficiently implementable or not, which queries the token at most m times), the probability of Case 2 occurring is at most $O(2^{2m-0.228n})$. (The constant c above thus needs to be chosen as $c < 0.114$.) This concludes the proof.

3.4 Security analysis for the token: Intuition

Our simulation proof showing statistical security of our Quantum OTM construction of Section 3.1 relies crucially on Theorem 3.4, stated below. As the proof of this theorem uses quantum information theoretic and semidefinite programming techniques (as opposed to cryptographic techniques), let us introduce notation in line with the formal analysis of Appendix C.

With respect to the construction of Section 3.1, let us replace each two-tuple $(x, \theta) \in \{0,1\}^n \times \{+,\times\}^n$ by a single string $z \in \{0,1\}^{2n}$, which we denote the *secret key*. Bits $2i$ and $2i+1$ of z specify the basis and value of conjugate coding qubit i for $i \in \{1,\ldots,n\}$ (*i.e.*, $z_{2i} = \theta_i$ and $z_{2i+1} = x_i$). Also, rename the "quantum key" (or conjugate coding key) $|\psi_z\rangle := |x\rangle_\theta \in (\mathbb{C}^2)^{\otimes n}$. Thus, the protocol

begins by having the sender pick a *secret key* $z \in \{0,1\}^{2n}$ uniformly at random, and preparing a joint state

$$|\psi\rangle = \frac{1}{2^n} \sum_{z \in |0,1\rangle^{2n}} |\psi_z\rangle_R |z\rangle_T.$$

The first register, R, is sent to the receiver, while the second register, T, is kept by the token. (Thus, the token knows the secret key z, and hence also which $|\psi_z\rangle$ the receiver possesses.) The mixed state describing the receiver's state of knowledge at this point is given by

$$\rho_R := \frac{1}{2^{2n}} \sum_{z \in \{0,1\}^{2n}} |\psi_z\rangle\langle\psi_z|.$$

We can now state Theorem 3.4.

Theorem 3.4. *Given a single copy of ρ_R, and the ability to make m (adaptive) queries to the hardware token, the probability that an unbounded quantum adversary can force the token to output both bits s_0 and s_1 scales as $O(2^{2m-0.228n})$.*

Thus, the probability of an unbounded adversary (*i.e.*, with the ability to apply the most general maps allowed in quantum mechanics, trace-preserving completely positive (TPCP) maps, which are not necessarily efficiently implementable) to successfully cheat using $m = cn$ for $c < 0.114$ queries is exponentially small in the quantum key size, n.

The full proof of Theorem 3.4 is given in Appendix C. Let us now give the intuition behind the proof approach.

Proof intuition. The challenge in analyzing security is the additional resource the receiver (henceforth called the user) is given, the state ρ_R, which the user may arbitrarily tamper with (in any manner allowed by quantum mechanics) while making queries to the token.

To prove Theorem 3.4, we model an adversary's actions as a two-party interaction between the user and token via the Gutoski-Watrous (GW) theory of quantum games [GW07]. At a high level, the GW framework can be used to model our setting via Figure 3.4 (reproduced from Appendix C for completeness), which we now discuss. The bottom "row" of Figure 3.4 depicts the token's actions, and the top row the user's actions. The protocol begins by imagining the token sends initial state $\rho_0 = \rho_R$ to the user via register $\mathcal{X}_1$. The user then applies an arbitrary sequence of TPCP maps Φ_i to its private memory (modeled by register $\mathcal{Z}_i$ in round i), each time sending a query to the token via register $\mathcal{Y}_i$. Given any such query in round i, the token applies its own TPCP map Ψ_i to determine how to respond to the query. The action of Ψ_i is fully determined by Program 1, and in principle all Ψ_i are identical since the token is stateless (*i.e.*, the action of the token in round i is unaffected by previous rounds $\{1, \ldots, i-1\}$). (We use the term "in principle" for the following reason. In practice, the token indeed has all Ψ_i being identical. To model our security analysis in the GW framework, which allows *quantum* interaction, it is convenient to imagine the token keeps a history of all queries it has seen thus far, which will technically make the Ψ_i distinct. What the history allows the token to do is simulate a *measurement* in the standard basis of any query the user sends; thus, the user can even send a quantum (*i.e.*, superposition) query, which the token immediately measures in the standard basis to recover a classical query string. In other words, we can simulate forcing the user to make classical queries in the GW framework by exploiting the well-known principle of deferred measurement. Crucially, in our security analysis, the token does

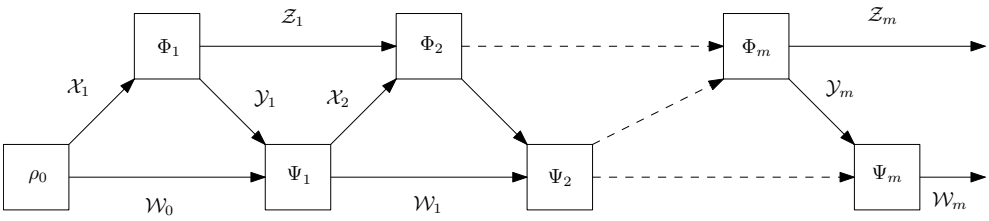

Figure 1: A general interaction between two quantum parties.

not condition any of its future actions on this query history it maintains; this ensures the token is stateless in its behavior. See Appendix C for details.) Finally, after receiving the mth query in register $\mathcal{Y}_m$, we imagine the token makes a measurement (not depicted in Figure 3.4) based on the query responses it returned; if the user managed to extract both bits s_0 and s_1 via queries, then the token "accepts", and otherwise it "rejects". (Again, here we are assuming the token keeps a history of all its responses in our security analysis; this is used only for the final measurement to judge whether the user cheated successfully, and not in the determination of any other action the token takes.)

With this high-level setup in place, the output of the GW framework is a semidefinite program, which we denote Γ:

$$\begin{aligned}
\text{min:} \quad & p && && (3)\\
\text{subject to:} \quad & Q_1 \preceq R_{m+1} && && (4)\\
& R_k = P_k \otimes I_{\mathcal{Y}_k} && \text{for } 1 \leq k \leq m+1 && (5)\\
& \mathrm{Tr}_{\mathcal{X}_k}(P_k) = R_{k-1} && \text{for } 1 \leq k \leq m+1 && (6)\\
& R_0 = p && && (7)\\
& R_k \in \mathrm{Pos}(\mathcal{Y}_{1,\ldots,k} \otimes \mathcal{X}_{1,\ldots,k}) && \text{for } 1 \leq k \leq m+1 && (8)\\
& P_k \in \mathrm{Pos}(\mathcal{Y}_{1,\ldots,k-1} \otimes \mathcal{X}_{1,\ldots,k}) && \text{for } 1 \leq k \leq m+1 && (9)
\end{aligned}$$

Above, Q_1 encodes the actions of the token. The variable p denotes the "cheating probability" (*i.e.*, the probability with which both s_0 and s_1 are extracted), subject to the constraint that operator R_{m+1} encodes a valid "strategy" for the user of the token. The constraints which enforce a valid "strategy" are given by Equations (5)-(9). These force the tuple $(R_1, \ldots, R_{m+1}, P_1, \ldots, P_{m+1})$ to encode a *Choi-Jamiołkowski representation of a quantum strategy* [GW07]. Let us briefly discuss the Choi-Jamiołkowski representation (details in Appendix C) and outline why such a framework might output an SDP. Recall that in round i of the protocol, the user applies TPCP map Φ_i. Any TPCP map has a number of known representations, one of which is the *Choi-Jamiołkowski representation [Cho75, Jam72]*. An advantage of the latter is it allows a simple characterization of the trace-preserving and completely positive properties of Φ_i via linear and semidefinite constraints, respectively. Reference [GW07] extends this definition of the Choi-Jamiołkowski representation from a single TPCP map Φ_i to an interactive protocol with multiple such maps $(\Phi_1, \ldots, \Phi_m)$; thus, intuitively one might expect such interactive strategies to also be characterized by linear and positive semidefinite constraints (*i.e.* by an SDP such as Γ).

Intuition for Q_1 and an upper bound on p. It remains to give intuition as to how one derives Q_1 in Γ, and how an upper bound on the optimal p is obtained. Without loss of generality, one may

assume that each of the token's TPCP maps Ψ_i are given by *isometries* $A_i : \mathcal{Y}_i \otimes \mathcal{W}_{i-1} \mapsto \mathcal{X}_{i+1} \otimes \mathcal{W}_i$, meaning $A_i^\dagger A_i = I_{\mathcal{Y}_i \otimes \mathcal{W}_{i-1}}$ (due to the Stinespring dilation theorem). (We omit the first isometry which prepares state ρ_0 in our discussion here for simplicity.) Let us denote their sequential application by a single operator $A := A_m \cdots A_1$ (note: to make the product well-defined, in Equation (16) of Appendix C, one uses tensor products with identity matrices appropriately). Then, the Choi-Jamiołkowski representation of A is given by [GW07]

$$\mathrm{Tr}_{\mathcal{Z}_m}(\mathrm{vec}(A)\,\mathrm{vec}(A)^\dagger),$$

where we trace out the token's private memory register $\mathcal{Z}_m$. (The operator $\mathrm{vec}(\cdot)$ reshapes matrix A into a vector; its precise definition is given in Section C, and is not required for our discussion here.) However, since in our security analysis, we imagine the token also makes a final measurement via some POVM $P = \{P_0, P_1\}$, whereupon obtaining outcome P_1 the token "accepts", and upon outcome P_0 the token rejects, we require a slightly more complicated setup — Q_1 will actually be defined as [GW07]

$$Q_1 = \mathrm{Tr}_{\mathcal{Z}_m}(\mathrm{vec}(B_1)\,\mathrm{vec}(B_1)^\dagger),$$

for $B_1 := P_1 A$.

The full derivation of Q_1 in our setting takes a few steps, and is given in Appendix C. Here, let us simply state Q_1 and give intuition:

$$\begin{aligned} Q_1 \quad = \quad & \frac{1}{4^n} \sum_{s \in S} |s_m b_{s_m}\rangle\langle s_m b_{s_m}|_{\mathcal{X}_{m+1}} \otimes \cdots \otimes |s_1 b_{s_1}\rangle\langle s_1 b_{s_1}|_{\mathcal{X}_2} \otimes \\ & \left(\sum_{(x,z) \in X_s} |x_m\rangle\langle x_m|_{\mathcal{Y}_m} \otimes \cdots \otimes |x_1\rangle\langle x_1|_{\mathcal{Y}_1} \otimes |\psi_z\rangle\langle\psi_z|_{\mathcal{X}_1} \right). \end{aligned}$$

Intuitively, each string $s_i b_{s_i} \in \{0,1\}^3$ encodes the response of the token given the ith query from the user; hence, the corresponding projectors in Q_1 act on spaces $\mathcal{X}_2$ through $\mathcal{X}_{m+1}$. Each string $x_i \in \{0,1\}^{n+1}$ denotes the ith query sent from the user to the token, where each $x_i = b_i y_i$ in the notation of Program 1, *i.e.* $b_i \in \{0,1\}$ is the choice bit for each query. Each such message is passed via register $\mathcal{Y}_i$. The states $|\psi_z\rangle$ and strings z are defined as in the beginning of Section 3.4; recall $z \in \{0,1\}^{2n}$ and $|\psi_z\rangle \in (\mathbb{C}^2)^{\otimes n}$ denote the secret key and corresponding quantum key, respectively. Finally, the relation X_s encodes the constraint that for all $i \in \{1, \ldots, m\}$, the tuple (x_i, z) (*i.e.* each message to the token x_i and secret key z) is consistent with the response returned by the token, s_i.

Upper bounding p. To now upper bound p, our approach is to give a feasible solution R_{m+1} satisfying the constraints of Γ. Note that giving even a solution which attains $p = 1$ for all n and m is *non*-trivial — such a solution is given in Lemma C.2 of Appendix C.3. Here, we shall give a solution which attains $p \in O(2^{2m - 0.228n})$, as claimed in Theorem 3.4 (and formally proven in Theorem C.3 of Appendix C.3). Namely, we set

$$R_{m+1} \quad = \quad \frac{1}{|S|} \sum_{s \in S} |s_m b_{s_m}\rangle\langle s_m b_{s_m}|_{\mathcal{X}_{m+1}} \otimes \cdots \otimes |s_1 b_{s_1}\rangle\langle s_1 b_{s_1}|_{\mathcal{X}_2} \otimes I_{Y_1 \otimes \cdots \otimes Y_m} \otimes \frac{I}{2^n}_{\mathcal{X}_1}.$$

This satisfies constraint (5) of Γ due to the identity term $I_{Y_1 \otimes \cdots \otimes Y_m}$. The renormalization factor of $(|S|\,2^n)^{-1}$ above ensures that tracing out all $\mathcal{X}_i$ registers yields $R_0 = 1$ in constraint (7) of Γ. We are thus reduced to choosing the minimum p such that constraint (4) is satisfied. Note that

setting $p = 1$ will *not* work for large enough m for this choice of R_{m+1}. To see why, observe we have chosen R_{m+1} to align with the block-diagonal structure of Q_1 on registers $\mathcal{X}_2, \ldots, \mathcal{X}_m$. Since registers $\mathcal{Y}_1 \otimes \cdots \otimes \mathcal{Y}_m$ and $\mathcal{X}_1$ of R_{m+1} are proportional to the identity matrix, it thus suffices to characterize the largest eigenvalue of Q_1, $\lambda_{\max}(Q_1)$. This is done by Lemma C.4 of Appendix C.3, which says

$$\lambda_{\max}(Q_1) = \frac{2}{4^n}\left(1 + \frac{1}{\sqrt{2}}\right)^n.$$

Combining this bound on $\lambda_{\max}(Q_1)$ with the parameters of R_{m+1} above now yields the desired claim that $p \in O(2^{2m-0.228n})$. We conclude that for $m < 0.114n$, the probability that a user of the token successfully cheats and thus that the simulation fails is exponentially small in the key size, n.

4 Impossibility Results

We now discuss "tightness" of our protocol with respect to impossibility results. To begin, it is easy to argue that OTMs cannot exist in the plain model (*i.e.*, without additional assumptions) in both the classical and quantum settings: in the classical setting, impossibility holds, since software can always be copied. Quantumly, this follows by a simple rewinding argument [BGS13]. Here, we give two simple no-go results for the quantum setting which support the idea that our scheme is "tight" in terms of the minimality of the assumptions it uses. Both results assume the token is reversible, meaning the receiver can run both the token and its inverse operation. The results can be stated as:

1. A stateless token which can be queried in *superposition* cannot be used to securely construct an OTM (Section 4.1).
2. For *measure and access* schemes such as ours, in order for a stateless token to allow statistical security, it must have an *exponential* number of keys per secret bit (Section 4.2).

4.1 Impossibility: Tokens which can be queried in superposition

In our construction, we require that all queries to the token be classical strings, *i.e.*, no querying in superposition is allowed. It is easy to argue via a standard rewinding argument that relaxing this requirement yields impossibility of a secure OTM, as long as access to the token's adjoint (inverse) operation is given, as we now show. Specifically, let M be a quantum OTM implemented using a hardware token. Since the token access is assumed to be reversible, we may model it as an oracle O_f realizing a function $f : \{0,1\}^n \mapsto \{0,1\}^m$ in the standard way, *i.e.*, for all $y \in \{0,1\}^n$ and $b \in \{0,1\}^m$,

$$O_f|y\rangle|b\rangle = |y\rangle|b \oplus f(y)\rangle. \tag{10}$$

Now, suppose our OTM stores two secret bits s_0 and s_1, and provides the receiver with an initial state $|\psi\rangle \in A \otimes B \otimes C$, where A, B, and C are the algorithm's workspace, *query* (*i.e.*, input to O_f), and *answer* (*i.e.*, O_f's answers) registers, respectively. By definition, an honest receiver must be able to access precisely one of s_0 or s_1 with certainty, given $|\psi\rangle$. Thus, for any $i \in \{0,1\}$, there exists a quantum query algorithm $A_i = U_m O_f \cdots O_f U_2 O_f U_1$ for unitaries $U_i \in \mathcal{U}(A \otimes B \otimes C)$ such that $A_i|\psi\rangle = |\psi'\rangle_{AB}|s_i\rangle_C$. For any choice of i, however, this implies a malicious receiver can now classically copy s_i to an external register, and then "rewind" by applying $A_i^\dagger$ to $|\psi'\rangle_{AB}|s_i\rangle_C$ to

recover $|\psi\rangle$. Applying $A_{i'}$ for $i' \neq i$ to $|\psi\rangle$ now yields the second bit i' with certainty as well. We conclude that a quantum OTM which allows superposition queries to a reversible stateless token is insecure.

Remark 4.1. *Above, we assumed the OTM outputs s_i with certainty. The argument can be generalized to the setting in which the OTM outputs s_i with probability at least $1 - \epsilon$ for small $\epsilon > 0$; in this case, Winter's Gentle Measurement Lemma [Win99] can be used to show that both bits can again be recovered with non-negligible probability.*

Remark 4.2. *Our argument crucially relies on the fact that the receiver has superposition access to the $A_i^\dagger$ operation. In certain models (e.g., software), such access is unavoidable. However, our result does not rule out the possibility that* non-reversible *superposition access to a token would allow for quantum OTMs.*

4.2 Impossibility: Tokens with a bounded number of keys

We have observed that allowing superposition queries to the token prevents an OTM from being secure. One might next ask how simple a hardware token with classical queries can be, while still allowing a secure OTM. We now explore one such strengthening of our construction in which the token is forced to have a bounded number of keys.

To formalize this, let us define the notion of a "measure-and-access (MA)" OTM, *i.e.*, an OTM in which given an initial state $|\psi\rangle$, an honest receiver applies a prescribed measurement to $|\psi\rangle$, and feeds the resulting classical string (*i.e.*, key) y into the token O_f to obtain s_i. Our construction is an example of a MA memory in which each bit s_i has an *exponential* number of valid keys y such that $f(y) = s_i$. One might ask whether the construction can be strengthened such that each s_i has a bounded number (*e.g.*, a polynomial number) of keys. We now show that such a strengthening would preclude security, assuming the token is reversible.

For clarity, implicitly in our proof below, we model the oracle O_f as having three possible outputs: 0, 1, or 2, where 2 is output whenever O_f is fed an invalid key y. This is required for the notion of having "few" keys to make sense (*i.e.*, there are 2^n candidate keys, and only two secret bits, each of which is supposed to have a bounded number of keys). Note that our construction indeed fits into this framework.

Lemma 4.3. *Let M be an MA memory with oracle O_f, such that O_f cannot be queried in superposition. If a secret bit s_i has at most Δ keys y_i such that $f(y_i) = s_i$, then given a single copy of $|\psi\rangle$, one can extract both s_0 and s_1 from M with probability at least $1/\Delta^2$.*

Remark 4.4. *The proof is given in Appendix D. Lemma 4.3 shows that in the paradigm of measure-and-access memories, our construction is essentially tight — in order to bound the adversary's success probability of obtaining both secret bits by an inverse exponential, we require each secret bit to have exponentially many valid keys. Second, as in the setting of superposition queries, the above proof can be generalized to the setting in which the OTM returns the correct bit s_i with probability at least $1 - \epsilon$ for small $\epsilon > 0$. Finally, the question of whether a similar statement to Lemma 4.3 holds for a* non-reversible *token remains open.*

A Universal Composition (UC) Framework

We consider simulation-based security. The Universal Composability (UC) framework was proposed by Canetti [Can01, Can00b], culminating a long sequence of simulation-based security

definitions (*c.f.* [GMW87, GL91, MR92, Bea91, Can00a]); please see also [PW01, PS04, CDPW07, LPV09, MR11] for alternative/extended frameworks. Recently Unruh [Unr10] extend the UC framework to the quantum setting. Next, we provide a high-level description of the original classical UC model by Canetti [Can01, Can00b], and then the quantum UC model by Unruh [Unr10].

A.1 Classical UC Model ([Can01, Can00b])

Machines. The basic entities involved in the UC model are players $P_1, \ldots, P_k$ where k is polynomial of security parameter n, an adversary $\mathcal{A}$, and an environment $\mathcal{Z}$. Each entity is modeled as a interactive Turing machine (ITM), where $\mathcal{Z}$ could have an additional non-uniform string as advice. Each P_i has identity i assigned to it, while $\mathcal{A}$ and $\mathcal{Z}$ have special identities $id_{\mathcal{A}} := \texttt{adv}$ and $id_{\mathcal{Z}} := \texttt{env}$.

Protocol Execution. A protocol specifies the programs for each P_i, which we denote as $\pi = (\pi_1, \ldots, \pi_k)$. The execution of a protocol is coordinated by the environment $\mathcal{Z}$. It starts by preparing inputs to all players, who then run their respective programs on the inputs and exchange messages of the form $(id_{\texttt{sender}}, id_{\texttt{receiver}}, \texttt{msg})$. $\mathcal{A}$ can corrupt an arbitrary set of players and control them later on. In particular, $\mathcal{A}$ can instruct a corrupted player sending messages to another player and also read messages that are sent to the corrupted players. During the course of execution, the environment $\mathcal{Z}$ also interacts with $\mathcal{A}$ in an arbitrary way. In the end, $\mathcal{Z}$ receives outputs from all the other players and generates one bit output. We use $\mathsf{EXEC}[\mathcal{Z}, \mathcal{A}, \pi]$ denote the distribution of the environment $\mathcal{Z}$'s (single-bit) output when executing protocol π with $\mathcal{A}$ and the P_i's.

Ideal Functionality and Dummy Protocol. Ideal functionality $\mathcal{F}$ is a trusted party, modeled by an ITM again, that perfectly implements the desired multi-party computational task. We consider an "dummy protocol", denoted $P^{\mathcal{F}}$, where each party has direct communication with $\mathcal{F}$, who accomplishes the desired task according to the messages received from the players. The execution of $P^{\mathcal{F}}$ with environment $\mathcal{Z}$ and an adversary, usually called the simulator $\mathcal{S}$, is defined analogous as above, in particular, $\mathcal{S}$ monitors the communication between corrupted parties and the ideal functionality $\mathcal{F}$. Similarly, we denote $\mathcal{Z}$'s output distribution as $\mathsf{EXEC}[\mathcal{Z}, \mathcal{S}, P^{\mathcal{F}}]$.

Definition A.1 (Classical UC-secure Emulation)**.** *We say π (classically) UC-emulates π' if for any adversary $\mathcal{A}$, there exists a simulator $\mathcal{S}$ such that for all environments $\mathcal{Z}$,*

$$\mathsf{EXEC}[\mathcal{Z}, \mathcal{A}, \pi] \approx \mathsf{EXEC}[\mathcal{Z}, \mathcal{S}, \pi'] \tag{11}$$

We here consider that $\mathcal{A}$ and $\mathcal{Z}$ are computationally unbounded, and we call it statistical UC-security. We require the running time $\mathcal{S}$ is polynomial in that of $\mathcal{A}$. We call this property Polynomial Simulation*.*

Let $\mathcal{F}$ be a well-formed two party functionality. We say π (classically) UC-realizes $\mathcal{F}$ if for all adversary $\mathcal{A}$, there exists a simulator $\mathcal{S}$ such that for all environments $\mathcal{Z}$, $\mathsf{EXEC}[\mathcal{Z}, \mathcal{A}, \pi] \approx \mathsf{EXEC}[\mathcal{Z}, \mathcal{S}, P^{\mathcal{F}}]$. We also write $\mathsf{EXEC}[\mathcal{Z}, \mathcal{A}, \pi] \approx \mathsf{EXEC}[\mathcal{Z}, \mathcal{S}, \mathcal{F}]$ if the context is clear.

UC-secure protocols admit a general composition property, demonstrated in the following universal composition theorem.

Theorem A.2 (UC Composition Theorem [Can00b])**.** *Let π, π' and σ be n-party protocols. Assume that π UC-emulates π'. Then σ^{π} UC-emulates $\sigma^{\pi'}$.*

A.2 Quantum UC Model ([Unr10])

Now, we give a high-level description of quantum UC model by Unruh [Unr10].

Quantum Machine. In the quantum UC model, all players are modeled as quantum machines. A quantum machine is a sequence of quantum circuits $\{M^n\}_{n\in\mathbb{N}}$, for each security parameter n. M^n is a completely positive trace preserving operator on space $\mathcal{H}^{\mathtt{state}} \otimes \mathcal{H}^{\mathtt{class}} \otimes \mathcal{H}^{\mathtt{quant}}$, where $\mathcal{H}^{\mathtt{state}}$ represents the internal workspace of M^n and $\mathcal{H}^{\mathtt{class}}$ and $\mathcal{H}^{\mathtt{quant}}$ represent the spaces for communication, where for convenience we divide the messages into classical and quantum parts. We allow a non-uniform quantum advice[5] to the machine of the environment $\mathcal{Z}$, while all other machines are uniformly generated.

Protocol Execution. In contrast to the communication policy in classical UC model, we consider a network $\mathbf{N}$ which contains the space $\mathcal{H}_{\mathbf{N}} := \mathcal{H}^{\mathtt{class}} \otimes \mathcal{H}^{\mathtt{quant}} \otimes_i \mathcal{H}_i^{\mathtt{state}}$. Namely, each machine maintains individual internal state space, but the communication space is shared among all . We assume $\mathcal{H}^{\mathtt{class}}$ contains the message $(id_{\mathtt{sender}}, id_{\mathtt{receiver}}, \mathtt{msg})$ which specifies the sender and receiver of the current message, and the receiver then processes the quantum state on $\mathcal{H}^{\mathtt{quant}}$. Note that this communication model implicitly ensures authentication. In a protocol execution, $\mathcal{Z}$ is activated first, and at each round, one player applies the operation defined by its machine M^n on $\mathcal{H}^{\mathtt{class}} \otimes \mathcal{H}^{\mathtt{quant}} \otimes \mathcal{H}^{\mathtt{state}}$. In the end $\mathcal{Z}$ generates a one-bit output. Denote $\mathsf{EXEC}[\mathcal{Z}, \mathcal{A}, \Pi]$ the output distribution of $\mathcal{Z}$.

Ideal Functionality. All functionalities we consider in this work are classical, *i.e.*, the inputs and outputs are classical, and its program can be implemented by an efficient classical Turing machine. Here in the quantum UC model, the ideal functionality $\mathcal{F}$ is still modeled as a quantum machine for consistency, but it only applies classical operations. Namely, it measures any input message in the computational basis to get an classical bit-string, and implements the operations specified by the classical computational task.

We consider an "dummy protocol", denoted $P^{\mathcal{F}}$, where each party has direct communication with $\mathcal{F}$, who accomplishes the desired task according to the messages received from the players. The execution of $P^{\mathcal{F}}$ with environment $\mathcal{Z}$ and an adversary, usually called the simulator $\mathcal{S}$, is defined analogous as above, in particular, $\mathcal{S}$ monitors the communication between corrupted parties and the ideal functionality $\mathcal{F}$. Similarly, we denote $\mathcal{Z}$'s output distribution as $\mathsf{EXEC}[\mathcal{Z}, \mathcal{S}, P^{\mathcal{F}}]$. For simplicity, we also write it as $\mathsf{EXEC}[\mathcal{Z}, \mathcal{S}, \mathcal{F}]$.

Definition A.3 (Quantum UC-secure Emulation)**.** *We say* Π *quantum-UC-emulates* Π' *if for any quantum adversary* $\mathcal{A}$*, there exists a (quantum) simulator* $\mathcal{S}$ *such that for all quantum environments* $\mathcal{Z}$*,*

$$\mathsf{EXEC}[\mathcal{Z}, \mathcal{A}, \Pi] \approx \mathsf{EXEC}[\mathcal{Z}, \mathcal{S}, \Pi'] \tag{12}$$

We consider here that $\mathcal{A}$ *and* $\mathcal{Z}$ *are computationally unbounded, we call it (quantum) statistical UC-security. We require the running time* $\mathcal{S}$ *is polynomial in that of* $\mathcal{A}$*. We call this property* Polynomial Simulation*.*

[5] Unruh's model only allows classical advice, but we tend to take the most general model. It is easy to justify that almost all results remain unchanged, including the composition theorem. See [HSS11, Section 5] for more discussion.

Similarly, (quantum) computational UC-security can be defined. Let $\mathcal{F}$ be a well-formed two party functionality. We say Π **quantum-UC-realizes** $\mathcal{F}$ if for all quantum adversary $\mathcal{A}$, there exists a (quantum) simulator $\mathcal{S}$ such that for all quantum environments $\mathcal{Z}$, $\mathsf{EXEC}[\mathcal{Z}, \mathcal{A}, \Pi] \approx \mathsf{EXEC}[\mathcal{Z}, \mathcal{S}, P^{\mathcal{F}}]$.

Quantum UC-secure protocols also admit general composition:

Theorem A.4 (Quantum UC Composition Theorem [Unr10, Theorem 11])**.** *Let Π, Π' and Σ be quantum-polynomial-time protocols. Assume that Π quantum UC-emulates Π'. Then Σ^{Π} quantum UC-emulates $\Sigma^{\Pi'}$.*

Remark A.5. *Out of the two protocol parties (the sender and the receiver), we consider security only in the case of the receiver being a corrupted party. Note that we are only interested in cases where the same party is corrupted with respect to all composed protocol. Furthermore, we only consider static corruption.*

B Stand-Alone Security in the case of a Malicious Sender

Here, we recall notation that is used in the analysis of two-party quantum protocol [DNS10].

Definition B.1. *An n-step quantum two-party protocol with oracle calls, denoted $\Pi^{\mathcal{O}} = (\mathscr{A}, \mathscr{B}, \mathcal{O}, n)$ consists of:*

1. *input space $\mathcal{A}_0$ and $\mathcal{B}_0$ for parties $\mathscr{A}$ and $\mathscr{B}$ respectively.*
2. *memory spaces $\mathcal{A}_1, \dots \mathcal{A}_n$ and $\mathcal{B}_1, \dots \mathcal{B}_n$ for $\mathscr{A}$ and $\mathscr{B}$, respectively.*
3. *An n-tuple of quantum operations $(\mathscr{A}_1, \dots \mathscr{A}_n)$ for $\mathscr{A}$, $\mathscr{A}_i : \mathrm{L}(\mathcal{A}_{i-1}) \mapsto \mathrm{L}(\mathcal{A}_i), (1 \leq i \leq n)$.*
4. *An n-tuple of quantum operations $(\mathscr{B}_1, \dots \mathscr{B}_n)$ for $\mathscr{B}$, $\mathscr{B}_i : \mathrm{L}(\mathcal{B}_{i-1}) \mapsto \mathrm{L}(\mathcal{B}_i), (1 \leq i \leq n)$.*
5. *Memory spaces $\mathcal{A}_1, \dots, \mathcal{A}_n$ and $\mathcal{B}_1, \dots, \mathcal{B}_n$ can be written as $\mathcal{A}_i = {\mathcal{A}_i}^{\mathcal{O}} \otimes {\mathcal{A}_i}'$ and $\mathcal{B}_i = {\mathcal{B}_i}^{\mathcal{O}} \otimes {\mathcal{B}_i}'$, $(1 \leq i \leq n)$ and $\mathcal{O} = (\mathcal{O}_1, \dots, \mathcal{O}_n)$ is an n-tuple of quantum operations: $\mathcal{O}_i : \mathrm{L}(\mathcal{A}_i^{\mathcal{O}} \otimes \mathcal{B}_i^{\mathcal{O}}) \mapsto \mathrm{L}(\mathcal{A}_i^{\mathcal{O}} \otimes \mathcal{B}_i^{\mathcal{O}})$, $(1 \leq i \leq n)$.*

If $\Pi^{\mathcal{O}} = (\mathscr{A}, \mathscr{B}, \mathcal{O}, n)$ is an n-turn two-party protocol, then the final state of the interaction upon input $\rho_{\mathrm{in}} \in \mathrm{D}(\mathcal{A}_0 \otimes \mathcal{B}_0 \otimes \mathcal{R})$ where $\mathcal{R}$ is a system of dimension $\dim \mathcal{A}_0 \dim \mathcal{B}_0$, is:

$$[\mathscr{A} \circledast \mathscr{B}](\rho_{\mathrm{in}}) = (\mathbb{1}_{\mathrm{L}(\mathcal{A}'_n \otimes \mathcal{B}'_n \otimes \mathcal{R})} \otimes \mathcal{O}_n)(\mathscr{A}_n \otimes \mathscr{B}_n \otimes \mathbb{1}_{\mathcal{R}}) \dots (\mathbb{1}_{\mathrm{L}(\mathcal{A}'_1 \otimes \mathcal{B}'_1 \otimes \mathcal{R})} \otimes \mathcal{O}_1)(\mathscr{A}_1 \otimes \mathscr{B}_1 \otimes \mathbb{1}_{\mathcal{R}})(\rho_{\mathrm{in}}) \,. \tag{13}$$

As in [DNS10], we specify that an oracle $\mathcal{O}$ can be a communication oracle or an ideal functionality oracle.

An *adversary* $\tilde{\mathscr{A}}$ for an honest party $\mathscr{A}$ in $\Pi^{\mathcal{O}} = (\mathscr{A}, \mathscr{B}, \mathcal{O}, n)$ is an n-tuple of quantum operations matching the input and outputs spaces of $\mathscr{A}$. A *simulator* for $\tilde{\mathscr{A}}$ is a sequence of quantum operations $(\mathcal{S}_i)_{i=1}^n$ where $\mathcal{S}_i$ has the same input-output spaces as the maps of $\tilde{\mathscr{A}}$ at step i. In addition, $\mathcal{S}$ has access to the ideal functionality for the protocol Π.

C Security Analysis for the Token

We now provide the technical result (Theorem 3.4) that is used to prove security of our Quantum OTM construction of Section 3.1 against a linear number of queries. The statement below is informal; to make it formal, in Section C.3 we model the user's interaction with the token via the Gutoski-Watrous framework for quantum games [GW07]. The resulting formal statement we desire, which immediately yields the informal claim below, is given in Theorem C.3.

Theorem C.1 (Informal). *For any stateless hardware token implemented as in Program 1,* i.e., *using an n-qubit conjugate coding state $|x\rangle_\theta$, and for any user of the token (restricted only by the laws of quantum mechanics, meaning using any trace-preserving completely positive maps desired, regardless of efficiency of their implementation) making m queries to the token, the probability the user successfully queries the token to extract both secret bits s_0 and s_1 is at most $O(2^{2m-0.228n})$.*

Thus, we are able to prove that if the user makes at most $m = cn$ queries with $c < 0.114$, then the user's probability of cheating successfully is exponentially small in n.

The next sections show this claim, and are organized as follows. Sections C.1 and C.2 introduce notation, terminology, semidefinite programming, and the Gutoski-Watrous quantum games framework. Section C.3 shows the formal version of the claim above, namely Theorem C.3.

C.1 Notation, quantum channels, and semidefinite programming

Notation. Let $\mathcal{X}$ be a finite dimensional complex Hilbert space. Then, $\mathcal{L}(\mathcal{X})$, $\mathrm{Herm}(\mathcal{X})$, $\mathrm{Pos}(\mathcal{X})$, and $\mathcal{D}(\mathcal{X})$ denote the sets of linear, Hermitian, positive semidefinite, and density operators acting on $\mathcal{X}$, respectively. The notation $A \succeq B$ means $A - B$ is positive semidefinite.

Quantum channels. A linear map $\Phi : \mathcal{L}(\mathcal{X}) \mapsto \mathcal{L}(\mathcal{Y})$ is a *quantum channel* if Φ is trace-preserving and completely positive (TPCP). These are the channels which map density operators to density operators. Although we will not directly make use of it here (the Gutoski-Watrous framework in Section C.2 will use the concept indirectly in our presentation), a useful representation of linear maps (or "superoperators") $\Phi : \mathcal{L}(\mathcal{X}) \mapsto \mathcal{L}(\mathcal{Y})$ is the Choi-Jamiołkowski representation, $J(\Phi) \in \mathcal{L}(\mathcal{Y} \otimes \mathcal{X})$. The latter is defined (with respect to some choice of orthonormal basis $\{|i\rangle\}$ for $\mathcal{X}$) as

$$J(\Phi) = \sum_{i,j} \Phi(|i\rangle\langle j|) \otimes |i\rangle\langle j|. \tag{14}$$

The following properties of $J(\Phi)$ hold [Cho75, Jam72]: (1) Φ is completely positive if and only if $J(\Phi) \succeq 0$, and (2) Φ is trace-preserving if and only if $\mathrm{Tr}_{\mathcal{Y}}(J(\Phi)) = I_{\mathcal{X}}$. In a nutshell, the Gutoski-Watrous framework generalizes this definition to *interacting* strategies [GW07].

Semidefinite programs. We give a brief overview of semidefinite programs (SDPs) from the perspective of quantum information, as done *e.g.*, in the notes of Watrous [Wat11] or [MVW13]. For further details, a standard text on convex optimization is Boyd and Vandenberghe [BV04].

Given any 3-tuple (A, B, Φ) for operators $A \in \mathrm{Herm}(\mathcal{X})$ and $B \in \mathrm{Herm}(\mathcal{Y})$, and linear map $\Phi : \mathcal{L}(\mathcal{X}) \mapsto \mathcal{L}(\mathcal{Y})$ mapping Hermitian operators to Hermitian operators, one can state a *primal* and *dual* semidefinite program:

Primal problem (P)		Dual problem (D)	
sup	$\mathrm{Tr}(AX)$	inf	$\mathrm{Tr}(BY)$
s.t.	$\Phi(X) = B,$	s.t.	$\Phi^*(Y) \succeq A$
	$X \in \mathrm{Pos}(\mathcal{X}),$		$Y \in \mathrm{Herm}(\mathcal{Y}),$

where Φ^* denotes the *adjoint* of Φ, which is the unique map satisfying $\mathrm{Tr}(A^\dagger \Phi(B)) = \mathrm{Tr}((\Phi^*(A))^\dagger B)$ for all $A \in \mathcal{L}(\mathcal{Y})$ and $B \in \mathcal{L}(\mathcal{X})$. Not all SDPs have feasible solutions (*i.e.* a solution satisfying all constraints); in this case, we label the optimal values as $-\infty$ for P and ∞ for D, respectively. Note also that the SDP we derive in Equation (21) will for simplicity not be written in precisely the form above, but can without loss of generality be made so.

C.2 The Gutoski-Watrous framework for quantum games

We now recall the Gutoski-Watrous (GW) framework for quantum games [GW07], which can be used to model quantum interactions between spatially separated parties. The setup most relevant to our protocol here is depicted in Figure C.2. Here, we imagine one party, A, prepares an initial

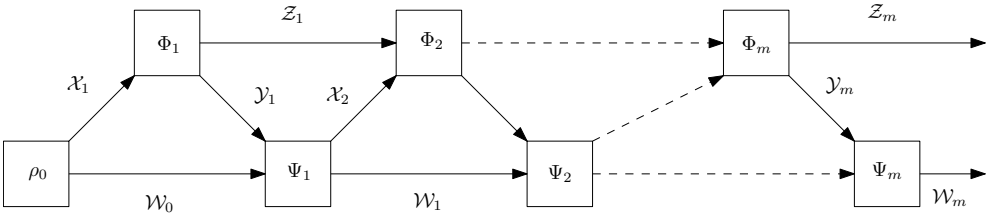

Figure 2: A general interaction between two quantum parties.

state $\rho_0 \in \mathcal{D}(\mathcal{X}_1 \otimes \mathcal{W}_0)$. Register $\mathcal{X}_1$ is then sent to the second party ($\mathcal{W}_0$ is kept as private memory), B, who applies some quantum channel $\Phi_i : \mathcal{L}(\mathcal{X}_1) \mapsto \mathcal{L}(\mathcal{Y}_1 \otimes \mathcal{Z}_1)$. B keeps register $\mathcal{Z}_1$ as private memory, and sends $\mathcal{Y}_1$ back to A, who applies channel $\Psi_1 : \mathcal{L}(\mathcal{W}_0 \otimes \mathcal{Y}_1) \mapsto \mathcal{L}(\mathcal{X}_2 \otimes \mathcal{W}_1)$, and sends $\mathcal{X}_2$ to B. The protocol continues for m messages back and forth, until the final operation $\Psi_m : \mathcal{L}(\mathcal{W}_m \otimes \mathcal{Y}_m) \mapsto \mathbb{C}$, in which A performs a two-outcome measurement (specifically, a POVM $P = \{P_0, P_1\}$, meaning $P_0, P_1 \succeq 0$, $P_0 + P_1 = I$) in order to decide whether to reject (P_0) or accept (P_1). As done in [GW07], we may assume without loss of generality[6] that all channels are given by linear isometries A_k, i.e. $\Phi_k(X) = A_k X A_k^\dagger$. (A linear isometry $A \in \mathcal{L}(\mathcal{S}, \mathcal{T})$ satisfies $A^\dagger A = I_{\mathcal{S}}$. Such maps are roughly generalizations of unitary maps to non-square matrices.) Reference [GW07] refers to $(\Phi_1, \ldots, \Phi_m)$ as a *strategy* and $(\rho_0, \Psi_1, \ldots, \Psi_m)$ as a *co-strategy*. In our setting, the former is "non-measuring", meaning it makes no final measurement after Φ_m is applied, whereas the latter is "measuring", since we will apply a final measurement on space $\mathcal{W}_m$ (not depicted in Figure C.2).

The GW framework then gives the *Choi-Jamiołkowski* (CJ) representation of a strategy and (measuring) co-strategy as follows. (Recall the definition of the *Choi-Jamiołkowski* representation for superoperators is given in Section C.1, but the relationship between that definition and the more generalized development below for strategies/co-strategies is not *a priori* obvious.)

[6] This is due to the Stinespring dilation theorem.

CJ representation of (non-measuring) strategy. The CJ representation is given by

$$\mathrm{Tr}_{\mathcal{Z}_m}(\mathrm{vec}(A)\,\mathrm{vec}(A)^\dagger), \tag{15}$$

where $A \in \mathcal{L}(\mathcal{X}_1 \otimes \cdots \otimes \mathcal{X}_m, \mathcal{Y}_1 \otimes \cdots \otimes \mathcal{Y}_m \otimes \mathcal{Z}_m)$ is defined as the product of the isometries A_i,

$$A := (I_{\mathcal{Y}_1 \otimes \cdots \otimes \mathcal{Y}_{m-1}} \otimes A_m) \cdots (A_1 \otimes I_{\mathcal{X}_2 \otimes \cdots \otimes \mathcal{X}_m}), \tag{16}$$

and the vec : $\mathcal{L}(\mathcal{S}, \mathcal{T}) \mapsto \mathcal{T} \otimes \mathcal{S}$ mapping is the linear extension of the map $|i\rangle\langle j| \mapsto |i\rangle|j\rangle$ defined on all standard basis states $|i\rangle, |j\rangle$.

CJ representation of (measuring) co-strategy. Let $P := \{P_0, P_1\}$ denote a POVM with reject and accept measurement operators P_0 and P_1, respectively. A measuring strategy which ends with a measurement with respect to POVM P replaces, for $P_a \in P$, Equation (15) with

$$\begin{aligned} Q_a &:= \mathrm{Tr}_{\mathcal{Z}_m}((P_a \otimes I_{\mathcal{Y}_1 \otimes \cdots \otimes \mathcal{Y}_m})\,\mathrm{vec}(A)\,\mathrm{vec}(A)^\dagger) \\ &= \mathrm{Tr}_{\mathcal{Z}_m}(\mathrm{vec}((\sqrt{P_a} \otimes I_{\mathcal{Y}_1 \otimes \cdots \otimes \mathcal{Y}_m})A)\,\mathrm{vec}((\sqrt{P_a} \otimes I_{\mathcal{Y}_1 \otimes \cdots \otimes \mathcal{Y}_m})A)^\dagger) \\ &=: \mathrm{Tr}_{\mathcal{Z}_m}(\mathrm{vec}(B_a)\,\mathrm{vec}(B_a)^\dagger). \end{aligned} \tag{17}$$

To convert this to a *co*-strategy, one takes the transpose of the operators defined above (with respect to the standard basis). (Note: In our use of the GW framework in Section C.3, all operators we derive will be symmetric with respect to the standard basis, and hence taking this transpose will be unnecessary.)

Optimization characterization over strategies and co-strategies. Fix any Q_a from a measuring co-strategy $\{Q_0, Q_1\}$, as in Equation (17). Then, the maximum probability with which a (non-measuring) strategy can force the co-strategy to output result a is given by

$$\begin{aligned} \text{min:} \quad & p & & \\ \text{subject to:} \quad & Q_a \preceq pR_m & & \\ & R_k = P_k \otimes I_{\mathcal{Y}_k} & & \text{for } 1 \le k \le m \\ & \mathrm{Tr}_{\mathcal{X}_k}(P_k) = R_{k-1} & & \text{for } 1 \le k \le m \\ & R_0 = 1 & & \\ & R_k \in \mathrm{Pos}(\mathcal{Y}_{1,\ldots,k} \otimes \mathcal{X}_{1,\ldots,k}) & & \text{for } 1 \le k \le m \\ & P_k \in \mathrm{Pos}(\mathcal{Y}_{1,\ldots,k-1} \otimes \mathcal{X}_{1,\ldots,k}) & & \text{for } 1 \le k \le m \\ & p \in [0,1] & & \end{aligned} \tag{18}$$

C.3 Security against a linear number of token queries

To show security of our hardware token implementation (Program 1) against a linear number of queries, we now model a user's interaction with the token as an interactive game between two parties using the GW framework of Section C.2. We shall treat the token as the *co-strategy* and the user as the *strategy*.

We proceed as follows. As depicted in Figure C.2, the token (co-strategy) begins by preparing state $\rho_0 \in \mathcal{L}(\mathcal{X}_1 \otimes \mathcal{W}_0)$, and sending message $\mathcal{X}_1$ to the user. The user then makes m queries, each

via a distinct register $\mathcal{Y}_i$ for $i \in \{1, \ldots, m\}$. For each query made, we model the token as returning two strings: (1) a symbol in set $\Sigma = \{0, 1, \overline{0}, \overline{1}\}$ where 0 and 1 denote successful 0- and 1-queries, respectively, and $\overline{0}$ and $\overline{1}$ denote unsuccessful 0- and 1-queries, respectively, and (2) a bit b which is set to 0 for a failed query, or secret bit b_i for a successful i-query. Formally, the size of each register $\mathcal{X}_i$ for $i \geq 2$ is hence three qubits. We will deviate from Figure C.2 in one respect — we assume the token returns the response to query m as well via a register $\mathcal{Y}_{m+1}$; this does not affect the success or failure of the user (as the latter makes no further queries at this point), but helps streamline the analysis. After this last response is sent out, the token measures the string $s \in \Sigma^m$ of responses it sent back to the user, and "accepts" if and only if s contains at least one 0 and one 1. (More on this history of responses kept by the token and how to maintain the stateless property of the token to come.)

Let us next introduce the terminology used in this section for discussing the secret key held by the token. Namely, recall in Program 1 that the token keeps secret key data $x \in \{0,1\}^n$ and $\theta \in \{+, \times\}^n$. Here, we shall replace these by a single string $z \in \{0,1\}^{2n}$, such that bits $2i$ and $2i+1$ of z specify the basis and value of conjugate coding qubit i, for $i \in \{1, \ldots, n\}$ (i.e. $z_{2i} = \theta_i$ and $z_{2i+1} = x_i$). We shall call z the *secret key*. For consistency, we shall rename the *quantum key* $|x\rangle_\theta$ from Program 1 by $|\psi_z\rangle \in (\mathbb{C}^2)^{\otimes n}$, i.e. $|x\rangle_\theta = |\psi_z\rangle$. Next, in Program 1 the token takes inputs $b \in \{0,1\}$ and $y \in \{0,1\}^n$, for b the choice bit and y the claimed measured value. In this section, we shall simply concatenate these as one string $x = by \in \{0,1\}^{n+1}$, the first bit of which is the choice bit. We shall refer to x as a *query string*. With these definitions in hand, for each secret key $z \in \{0,1\}^{2n}$, we define a partition $A_{\overline{0}}(z)$, $A_{\overline{1}}(z)$, $A_0(z)$, $A_1(z)$ of $\{0,1\}^{n+1}$, which correspond to the sets of query strings x which cause the token to return response $\overline{0}$, $\overline{1}$, 0, or 1, respectively.

We can now begin to set up the GW framework. To define linear isometries A_k, we first construct operators $\Delta_k(z) : \mathcal{Y}_k \mapsto \mathcal{X}_{k+1} \otimes \mathcal{W}_{k,k+1}$ for $k \in \{1, \ldots, m\}$ as follows:

$$\begin{aligned}
\Delta_k(z) \quad = \quad & \sum_{x \in A_{\overline{0}}(z)} |\overline{0}0\rangle_{\mathcal{X}_{k+1}} |x\overline{0}\rangle_{\mathcal{W}_{k,k+1}} \langle x|_{\mathcal{Y}_k} + \\
& \sum_{x \in A_{\overline{1}}(z)} |\overline{1}0\rangle_{\mathcal{X}_{k+1}} |x\overline{1}\rangle_{\mathcal{W}_{k,k+1}} \langle x|_{\mathcal{Y}_k} + \\
& \sum_{x \in A_0(z)} |0b_0\rangle_{\mathcal{X}_{k+1}} |x0\rangle_{\mathcal{W}_{k,k+1}} \langle x|_{\mathcal{Y}_k} + \\
& \sum_{x \in A_1(z)} |1b_1\rangle_{\mathcal{X}_{k+1}} |x1\rangle_{\mathcal{W}_{k,k+1}} \langle x|_{\mathcal{Y}_k}.
\end{aligned}$$

Above, recall the register $\mathcal{Y}_k$ denotes the kth message sent by the user to the token, $\mathcal{X}_k$ the kth response sent by the token back to the user (the first symbol of which denotes accept/reject via a symbol from Σ, and the second symbol of which is the corresponding secret bit, which is set to 0 by default for failed queries), and $\mathcal{W}_k$ denotes the private memory of the token, which we now discuss further.

Let us now elaborate on how the token's private memory spaces $\mathcal{W}_k$ is modelled. First, $\mathcal{W}_0$ contains the secret key $z \in \{0,1\}^{2n}$ of the token. Then, each $\mathcal{W}_k$ register for $k > 0$ is split into $k+1$ parts: $\mathcal{W}_{k,1}$ contains a copy of z (this allows us to pass forward z from one round of interaction to the next), and $\mathcal{W}_{k,r}$ for $r \geq 2$ contains a copy in the standard basis of the user's $(r-1)$st query string, as well as the token's response from Σ. These copies are kept for two reasons. First, it

simulates measuring each message from the user in the standard basis[7], as required by the token. Second, keeping x ensures $\Delta_k(z)^\dagger \Delta_k(z) = I$, so that each A_i defined shortly is an isometry. Note that, crucially, the contents of $\mathcal{W}_{k,r}$ for $r \geq 2$ are never accessed again[8] by the token in any future iteration (this is the definition of the token being *stateless*, and is formally captured by the definition of the terms A_i shortly). Also, while the size of $\mathcal{W}$ grows with m in our security analysis here, the actual token does not have growing memory requirements since it simply discards the results of each measurement of the user's message once it returns a symbol from Σ in each round of communication.

We can now define isometries A_i (intuition to follow) for round i of the token's actions, where $1 < k \leq m$:

$$\begin{aligned}
A_0 &= \frac{1}{2^n} \sum_{z \in \{0,1\}^{2n}} |\psi_z\rangle_{\mathcal{X}_1} |z\rangle_{\mathcal{W}_{0,1}} \\
A_1 &= \sum_{z \in \{0,1\}^{2n}} \Delta_1(z)_{\mathcal{Y}_1, \mathcal{X}_2, \mathcal{W}_{1,2}} \otimes |z\rangle_{\mathcal{W}_{1,1}} \langle z|_{\mathcal{W}_{0,1}} \\
A_k &= \sum_{z \in \{0,1\}^{2n}} \Delta_k(z)_{\mathcal{Y}_k, \mathcal{X}_{k+1}, \mathcal{W}_{k,k+1}} \otimes |z\rangle_{\mathcal{W}_{k,1}} \langle z|_{\mathcal{W}_{k-1,1}} \bigotimes_{r=2}^{k} I_{\mathcal{W}_{k,r}, \mathcal{W}_{k-1,r}}
\end{aligned}$$

Here, $A_0 : \mathbb{C} \mapsto \mathcal{X}_1 \otimes \mathcal{W}_0$, and $A_k : \mathcal{Y}_k \otimes \mathcal{W}_{k-1} \mapsto \mathcal{X}_{k+1} \otimes \mathcal{W}_k$ for $1 \leq k \leq m$. Intuitively, the isometry A_0 captures the token choosing an initial secret key z uniformly at random and preparing corresponding quantum key $|\psi_z\rangle$, which it sends to the user. Each A_i for $1 \leq k \leq m$ captures the token reading a message from the user in $\mathcal{Y}_k$ and measuring it in the standard basis (simulated by copying string x to a private register $\mathcal{W}_{k,k+1}$), and returning an appropriate response to the user in register $\mathcal{X}_{k+1}$ (note this response depends only on the contents of $\mathcal{Y}_k$, *i.e.*, on the kth message, since the token is stateless). It also stores a copy of the kth response from Σ to the user in the private register $\mathcal{W}_{k,k+1}$; as mentioned before, this information is not accessed by the token in deciding any future messages $\mathcal{X}_{k+i}$, but is used in our analysis to define the accepting measurement P_1.

Having defined isometries A_i, their product now yields operator A from Equation (16) (where we reorder the $\mathcal{X}$ and $\mathcal{W}$ registers to clarify that incoming message $\mathcal{Y}_k$ results in outgoing message $\mathcal{X}_{k+1}$):

$$\begin{aligned}
A = \frac{1}{2^n} \sum_{z \in \{0,1\}^{2n}} \sum_{x_1, \ldots, x_m \in \{0,1\}^{n+1}} & |x_1 A(x_1, z)\rangle_{\mathcal{W}_{m,2}} \otimes \cdots \otimes |x_m A(x_m, z)\rangle_{\mathcal{W}_{m,m+1}} \otimes \\
& |A(x_m, z) b_{A(x_m,z)}\rangle_{\mathcal{X}_{m+1}} \langle x_m|_{\mathcal{Y}_m} \otimes |A(x_{m-1}, z) b_{A(x_{m-1},z)}\rangle_{\mathcal{X}_m} \langle x_{m-1}|_{\mathcal{Y}_{m-1}} \otimes \cdots \otimes \\
& |A(x_1, z) b_{A(x_1,z)}\rangle_{\mathcal{X}_2} \langle x_1|_{\mathcal{Y}_1} \otimes |\psi_z\rangle_{\mathcal{X}_1} \otimes |z\rangle_{\mathcal{W}_{m,1}},
\end{aligned}$$

[7]Normally, the GW framework allows quantum messages to be exchanged between parties, but in our setting the token only accepts classical query strings. To force the user to send classical strings, the token can simulate measurement of the user's query qubits by simply creating a "local copy" of said qubits via local controlled-NOT gates, i.e. the token employs the principle of deferred measurement.

[8]Strictly speaking, as mentioned earlier, this is not quite accurate — the contents of $\mathcal{W}_{k,r}$ *are* accessed during the final measurement made by the token in determining whether the user succeeded in cheating. This, however, is just a technical construct for our analysis, which allows us to formalize what it means for the user to "successfully cheat". The final measurement does not affect any of the token's previous responses $\mathcal{X}_i$, thus maintaining the stateless property of the token. For clarity, in the actual protocol itself, the token does not keep any private memory, and makes no "final" measurement as done in our analysis.

where $A(x,z) \in \Sigma$ denotes whether the token accepted or rejected query string x assuming secret key z, and $b_{A(x,z)} \in \{0,1\}$ is the secret bit returned by the token corresponding to $A(x,z) \in \Sigma$.

In order to next define operator Q_1 from Equation (17), we model what it means for a cheating user of the token to "succeed". As mentioned earlier, this is formalized by having the token make a final measurement after the protocol concludes, in order to determine whether the user has successfully extracted both secret bits via queries. Formally, for convenience, let $\mathcal{W}'$ denote the tensor product of the registers in $\mathcal{W}_{m,r}$ for $2 \leq r \leq m+1$ which hold the values from Σ (*i.e.*, which hold terms $A(x_{r-1}, z)$). Then, a *successful* user makes at least one correct 0 query and at least one correct 1 query. Our accepting measurement operator P_1, which corresponds to a successful user, is thus defined as follows. P_1 maps $\mathcal{W}'$ to itself, and is a projector onto the set of strings with some $i \neq j$ such that $\mathcal{W}'_i$ is set to $|0\rangle$ and $\mathcal{W}'_j$ is set to $|1\rangle$. In other words, P_1 projects onto set

$$S := \{s \in \Sigma^m \mid s \text{ contains at least one 0 and one 1}\}. \tag{19}$$

To now use this definition of P_1 to write down B_1, we first need further terminology. Define for any $s \in S$ and fixed key $z \in \{0,1\}^{2n}$, the set of all consistent sequences of query strings $x_i \in \{0,1\}^{n+1}$ as:

$$X_s = \Big\{(x,z) \in \{0,1\}^{m(n+1)} \times \{0,1\}^{2n} \mid A(x_i,z) = s_i \text{ for } x_i \text{ the } i\text{th block of } (n+1) \text{ bits in } x\Big\}.$$

(For clarity and as an example, the second block of $(n+1)$ bits of $0^{n+1}1^{n+1}$ is 1^{n+1}.) Finally, define relation $R \subseteq \Sigma^m \times \{0,1\}^{m(n+1)} \times \{0,1\}^{2n}$ such that

$$(s,x,z) \in R \text{ if and only if } (x,z) \in X_s.$$

In words, a triple $(s,x,z) \in R$ if for a secret key z and query string x, $s \in \Sigma^m$ is the (unique) correct response string from the token.

With these definitions in place, we can finally define $B_1 = (\sqrt{P_1} \otimes I)A = (P_1 \otimes I)A$ as (where recall $s_i = A(x_i,z)$)

$$\begin{aligned} B_1 \;=\; \frac{1}{2^n} \sum_{(s,x,z)\in R} & |x_1 s_1\rangle_{\mathcal{W}_{m,2}} \otimes \cdots \otimes |x_m s_m\rangle_{\mathcal{W}_{m,m+1}} \otimes \\ & |s_m b_{s_m}\rangle_{\mathcal{X}_{m+1}} \langle x_m|_{\mathcal{Y}_m} \otimes |s_{m-1} b_{s_{m-1}}\rangle_{\mathcal{X}_m} \langle x_{m-1}|_{\mathcal{Y}_{m-1}} \otimes \cdots \otimes |s_1 b_{s_1}\rangle_{\mathcal{X}_2} \langle x_1|_{\mathcal{Y}_1} \otimes \\ & |\psi_z\rangle_{\mathcal{X}_1} \otimes |z\rangle_{\mathcal{W}_{m,1}}. \end{aligned}$$

Thus,

$$\begin{aligned} \mathrm{vec}(B_1) \;=\; \frac{1}{2^n} \sum_{(s,x,z)\in R} & |x_1 s_1\rangle_{\mathcal{W}_{m,2}} \otimes \cdots \otimes |x_m s_m\rangle_{\mathcal{W}_{m,m+1}} \otimes \\ & |s_m b_{s_m}\rangle_{\mathcal{X}_{m+1}} |x_m\rangle_{\mathcal{Y}_m} \otimes |s_{m-1} b_{s_{m-1}}\rangle_{\mathcal{X}_m} |x_{m-1}\rangle_{\mathcal{Y}_{m-1}} \otimes \cdots \otimes |s_1 b_{s_1}\rangle_{\mathcal{X}_2} |x_1\rangle_{\mathcal{Y}_1} \otimes \\ & |z\rangle_{\mathcal{W}_{m,1}} \otimes |\psi_z\rangle_{\mathcal{X}_1}. \end{aligned}$$

It follows that $Q_1 = \mathrm{Tr}_{\mathcal{W}_m}(\mathrm{vec}(B_1)\,\mathrm{vec}(B_1)^*)$ equals

$$\begin{aligned} Q_1 \;=\; \frac{1}{2^{2n}} \sum_{(s,x,z)\in R} & |s_m b_{s_m}\rangle\langle s_m b_{s_m}|_{\mathcal{X}_{m+1}} \otimes |x_m\rangle\langle x_m|_{\mathcal{Y}_m} \otimes \cdots \otimes \\ & |s_1 b_{s_1}\rangle\langle s_1 b_{s_1}|_{\mathcal{X}_2} \otimes |x_1\rangle\langle x_1|_{\mathcal{Y}_1} \otimes |\psi_z\rangle\langle\psi_z|_{\mathcal{X}_1}. \end{aligned}$$

Above, note that we have crucially used the fact that queries to the token are *classical strings*; this allows us to reduce Q_1 to a *mixture* over $(s, x, z) \in R$ (*i.e.*, all cross-terms in $\mathrm{vec}(B_1)\,\mathrm{vec}(B_1)^*$ disappear once we trace out $\mathcal{W}_m$).

Permuting subsystems, we can hence write:

$$\begin{aligned} Q_1 \;=\; & \frac{1}{4^n} \sum_{s \in S} |s_m b_{s_m}\rangle\langle s_m b_{s_m}|_{\mathcal{X}_{m+1}} \otimes \cdots \otimes |s_1 b_{s_1}\rangle\langle s_1 b_{s_1}|_{\mathcal{X}_2} \otimes \\ & \left(\sum_{(x,z) \in X_s} |x_m\rangle\langle x_m|_{\mathcal{Y}_m} \otimes \cdots \otimes |x_1\rangle\langle x_1|_{\mathcal{Y}_1} \otimes |\psi_z\rangle\langle\psi_z|_{\mathcal{X}_1} \right). \end{aligned} \tag{20}$$

Having set up all required operators for the GW framework, Equation (18) of Section C.2 now yields the optimal probability with which a cheating user can succeed; we reproduce Equation (18) below for ease of exposition. Note the subsystem ordering of Q_1 below is not that of Equation (20), but rather $Q_1 \in \mathrm{Pos}(\mathcal{Y}_{1,\ldots,m} \otimes \mathcal{X}_{1,\ldots,m+1})$ below; we have omitted explicitly including the permutation effecting this reordering to avoid clutter. Also, to account for the slight asymmetry in our protocol (the token sends out $m+1$ messages $\mathcal{X}_i$, whereas the user only sends m messages $\mathcal{Y}_i$), we add a dummy space $\mathcal{Y}_{m+1} = \mathbb{C}$ which models an empty $(m+1)$st message from the user to the token.

$$\begin{aligned} \text{min:} \quad & p \\ \text{subject to:} \quad & Q_1 \preceq p R_{m+1} \\ & R_k = P_k \otimes I_{\mathcal{Y}_k} && \text{for } 1 \leq k \leq m+1 \\ & \mathrm{Tr}_{\mathcal{X}_k}(P_k) = R_{k-1} && \text{for } 1 \leq k \leq m+1 \\ & R_0 = 1 \\ & R_k \in \mathrm{Pos}(\mathcal{Y}_{1,\ldots,k} \otimes \mathcal{X}_{1,\ldots,k}) && \text{for } 1 \leq k \leq m+1 \\ & P_k \in \mathrm{Pos}(\mathcal{Y}_{1,\ldots,k-1} \otimes \mathcal{X}_{1,\ldots,k}) && \text{for } 1 \leq k \leq m+1 \\ & p \in [0,1] \end{aligned}$$

While this optimization is not an SDP due to the quadratic constraint $Q_1 \preceq p R_{m+1}$, it is easily seen to be equivalent to the following SDP Γ:

$$\begin{aligned} \text{min:} \quad & p \\ \text{subject to:} \quad & Q_1 \preceq R_{m+1} \\ & R_k = P_k \otimes I_{\mathcal{Y}_k} && \text{for } 1 \leq k \leq m+1 \\ & \mathrm{Tr}_{\mathcal{X}_k}(P_k) = R_{k-1} && \text{for } 1 \leq k \leq m+1 \\ & R_0 = p \\ & R_k \in \mathrm{Pos}(\mathcal{Y}_{1,\ldots,k} \otimes \mathcal{X}_{1,\ldots,k}) && \text{for } 1 \leq k \leq m+1 \\ & P_k \in \mathrm{Pos}(\mathcal{Y}_{1,\ldots,k-1} \otimes \mathcal{X}_{1,\ldots,k}) && \text{for } 1 \leq k \leq m+1 \end{aligned} \tag{21}$$

Note we have dropped the constraint $p \in [0,1]$; this is redundant, as we now show. Henceforth, for brevity we shall use terminology $\mathcal{T}_{1\cdots k}$ to denote the space $\mathcal{T}_1 \otimes \cdots \otimes \mathcal{T}_k$.

Lemma C.2. *The SDP Γ has a feasible solution with $p = 1$.*

Proof. Recall from Equation (20) that

$$Q_1 = \frac{1}{4^n} \sum_{(s,x,z)\in R} |s,b\rangle\langle s,b|_{\mathcal{X}_{m+1\cdots 2}} \otimes \Big(|x_m\rangle\langle x_m|_{\mathcal{Y}_m} \otimes \cdots \otimes |x_1\rangle\langle x_1|_{\mathcal{Y}_1}\Big) \otimes |\psi_z\rangle\langle\psi_z|_{\mathcal{X}_1},$$

where $s \subseteq \Sigma^m$ and $b \in \{0,1\}^m$ are the resulting query responses and secret bits, respectively. Observe that any fixed $x \in \{0,1\}^{m(n+1)}$ and $z \in \{0,1\}^{2n}$ determine a *unique* query response string $s \in \Sigma^m$; denote this as $s(x,z)$. Therefore,

$$Q_1 = \frac{1}{4^n} \sum_{\substack{x,z \\ \text{s.t. } s(x,z)\in S}} |s(x,z),b\rangle\langle s(x,z),b|_{\mathcal{X}_{m+1\cdots 2}} \otimes \Big(|x_m\rangle\langle x_m|_{\mathcal{Y}_m} \otimes \cdots \otimes |x_1\rangle\langle x_1|_{\mathcal{Y}_1}\Big) \otimes |\psi_z\rangle\langle\psi_z|_{\mathcal{X}_1},$$

for $S \subseteq \Sigma^m$ defined as in Equation (19). Let us drop the constraint that $s(x,z) \in S$, *i.e.* choose

$$R_{m+1} = \frac{1}{4^n} \sum_{x,z} |s(x,z),b\rangle\langle s(x,z),b|_{\mathcal{X}_{m+1\cdots 2}} \otimes \Big(|x_m\rangle\langle x_m|_{\mathcal{Y}_m} \otimes \cdots \otimes |x_1\rangle\langle x_1|_{\mathcal{Y}_1}\Big) \otimes |\psi_z\rangle\langle\psi_z|_{\mathcal{X}_1}.$$

Clearly, $Q_1 \preceq p \cdot R_{m+1}$ for $p = 1$, since we added positive semidefinite terms to Q_1 to get R_{m+1}. Thus, if R_{m+1} satisfies the remaining primal constraints, then it has objective function value $p = 1$.

To see that R_{m+1} satisfies the primal constraints, clearly Q_{m+1} has I in register $\mathcal{Y}_{m+1}$ (recall $\mathcal{Y}_{m+1} = \mathbb{C}$, so this just means $\mathcal{Y}_{m+1}$ is trivially set to 1). Let us now trace out $\mathcal{X}_{m+1}$; we require that register $\mathcal{Y}_{m-1}$ now also contains the identity. For this, $\mathrm{Tr}_{\mathcal{X}_{m+1}}(R_{m+1})$ equals:

$$\frac{1}{4^n} \sum_{x_m,\ldots,x_1} \sum_z |s_{m-1}(x,z)b_{m-1}\rangle\langle s_{m-1}(x,z)b_{m-1}|_{\mathcal{X}_{m\cdots 2}} \otimes \Big(|x_m\rangle\langle x_m|_{\mathcal{Y}_m} \otimes \cdots \otimes |x_1\rangle\langle x_1|_{\mathcal{Y}_1}\Big) \otimes |\psi_z\rangle\langle\psi_z|_{\mathcal{X}_1},$$

where for brevity we use $s_{m-1}(x,z)b_{m-1}$ to denote the first $m-1$ queries. But since we discarded the mth symbol of $s(x,z)$, registers $\mathcal{Y}_m$ and $\mathcal{X}_1$ are now independent. Thus, bringing in the sum over x_m,

$$\begin{aligned}\mathrm{Tr}_{\mathcal{X}_{m+1}}(Q_{m+1}) \;=\; & \frac{1}{4^n} \sum_{x_{m-1},\ldots,x_1} \sum_z |s_{m-1}(x,z)b_{m-1}\rangle\langle s_{m-1}(x,z)b_{m-1}|_{\mathcal{X}_{m\cdots 2}} \otimes \\ & \Big(I_{\mathcal{Y}_m} \otimes |x_{m-1}\rangle\langle x_{m-1}|_{\mathcal{Y}_{m-1}} \otimes \cdots \otimes |x_1\rangle\langle x_1|_{\mathcal{Y}_1}\Big) \otimes |\psi_z\rangle\langle\psi_z|_{\mathcal{X}_1}.\end{aligned}$$

In a similar fashion, tracing out registers $\mathcal{X}_{m\cdots 2}$ will yield operator

$$\frac{1}{4^n} I_{\mathcal{Y}_{m+1\cdots 1}} \otimes \sum_z |\psi_z\rangle\langle\psi_z|_{\mathcal{X}_1}.$$

Finally, tracing out $\mathcal{X}_1$ yields $I_{\mathcal{Y}_{m\cdots 1}}$, since there are 4^n possible quantum key states $|\psi_z\rangle$. Hence, R_{m+1} is a feasible solution. □

An upper bound on the cheating probability. We now give a feasible solution to SDP Γ which yields the claimed security against a linear number of queries. Its proof of correctness relies on Lemma C.4, stated and proven subsequently.

Theorem C.3. *The SDP Γ has a feasible solution with $p \in O(2^{2m-0.228n})$.*

Proof. As Q_1 in Equation (20) is block-diagonal in registers $\mathcal{X}_2, \ldots, \mathcal{X}_{m+1}$, consider solution (for S from Equation (19))

$$R_{m+1} \quad = \quad \frac{1}{|S|} \sum_{s \in S} |s_m b_{s_m}\rangle\langle s_m b_{s_m}|_{\mathcal{X}_{m+1}} \otimes \cdots \otimes |s_1 b_{s_1}\rangle\langle s_1 b_{s_1}|_{\mathcal{X}_2} \otimes I_{\mathcal{Y}_{1,\ldots,m}} \otimes \frac{I}{2^n}_{\mathcal{X}_1}.$$

(Aside: Recall that $\mathcal{X}_1$ is an n-qubit register above, hence the 2^n renormalization factor.) Note that

$$\begin{aligned} |\Sigma^m| &= 4^m \\ \{s \in \Sigma^m \mid s \text{ does not contain a } 0\} &= 3^m \\ \{s \in \Sigma^m \mid s \text{ does not contain a } 1\} &= 3^m \\ \{s \in \Sigma^m \mid s \text{ does not contain a } 0 \text{ or a } 1\} &= 2^m. \end{aligned}$$

Thus, by the inclusion-exclusion principle, $|S| = 4^m - 2 \cdot 3^m + 2^m$.

In order for R_{m+1} to be feasible, we must pick p such that $Q_1 \preceq pR_{m+1}$ (recall this is equivalent to the SDP formulation Γ). Since Q_1 is block-diagonal on registers $\mathcal{X}_2 \cdots \mathcal{X}_{m+1}$, it suffices to identify its block with the largest eigenvalue. In fact, each corresponding block for R_{m+1} has eigenvalue $(|S|\, 2^n)^{-1}$. Thus, we must choose p such that

$$\lambda_{\max}(Q_1) \leq \frac{p}{|S|\, 2^n},$$

or equivalently, due to the 4^{-n} factor in Q_1,

$$p \geq \frac{|S|}{2^n} \lambda_{\max}\left(4^n Q_1\right).$$

By Lemma C.4, $\lambda_{\max}(Q_1) = \frac{2}{4^n}\left(1 + \frac{1}{\sqrt{2}}\right)^n$. Thus, we can set

$$p = \frac{|S|}{2^{n-1}}\left(1 + \frac{1}{\sqrt{2}}\right)^n \approx |S| \cdot 2^{(-0.228)n+1},$$

and since $|S| \in \Theta(4^m)$, the cheating probability satisfies $p \in O(2^{2m-0.228n})$. □

Lemma C.4. *For Q_1 in Equation (20), $\lambda_{\max}(Q_1) = \frac{2}{4^n}\left(1 + \frac{1}{\sqrt{2}}\right)^n$.*

Proof. The factor of 4^{-n} in the claimed value for $\lambda_{\max}(Q_1)$ comes from the 4^{-n} appearing in Equation (20); we henceforth thus ignore this 4^{-n} term in this proof by redefining Q_1 as $4^n Q_1$. We shall also ignore the b_i terms in Q_1, as they shall play no role in the analysis. Now, since Q_1 is block-diagonal (with respect to the standard basis) on registers $\mathcal{X}_2,\ldots,\mathcal{X}_{m+1}$, $\mathcal{Y}_1,\ldots,\mathcal{Y}_m$, it suffices to characterize the largest eigenvalue of any block. We shall say that any fixed $s \in S$ and $x \in \{0,1\}^{m(n+1)}$ defines the (s,x)-block of Q_1. (Formally, the (s,x)-block of Q_1 is given by $\Pi_{s,x} Q_1 \Pi_{s,x}$, where $\Pi_{s,x} = |s\rangle\langle s|_{\mathcal{X}_{m+1\cdots 2}} \otimes |x\rangle\langle x|_{\mathcal{Y}}$.)

We begin by demonstrating an explicit s, x such that the (s,x)-block has eigenvalue $\frac{2}{4^n}(1 + \frac{1}{\sqrt{2}})^n$ (*i.e.* $\lambda_{\max}(Q_1) \geq \frac{2}{4^n}(1 + \frac{1}{\sqrt{2}})^n$). Set $s = 0^{m-1}1$ (note $s \in S$) and $x = x_1 \ldots x_m$ for $x_1 = x_2 = \cdots = x_{m-1}$ and $x_{m-1} \neq x_m$ (note $x_i \in \{0,1\}^{n+1}$), where the first bit of each of $x_1, \ldots, x_{m-1}$ is 0, and the

first bit of x_m is 1. In words, we are modelling $m-1$ successful (and identical) 0-queries in the Z-basis, followed by a single successful 1-query in the X-basis. The question now is: Given s and x, how many $|\psi_z\rangle \in (\mathbb{C}^2)^{\otimes n}$ exist such that $(s,x,z) \in R$?

To answer this, observe that the token enforces the following set of rules. Fix any $i \in \{1,\ldots,m\}$, and let $|x_i(j)\rangle$ and $|\psi_z(j)\rangle$ denote the jth qubits of x_i and ψ_z, respectively. Then we have rules (where H denotes the 2×2 Hadamard matrix, and $\overline{b}$ denotes the complement of bit b):

1. If $s_i = 0$, then $\forall j \in \{1,\ldots,n\}$, either $|\psi_z(j)\rangle = |x_i(j)\rangle$ or $|\psi_z(j)\rangle \in \{|+\rangle,|-\rangle\}$.
2. If $s_i = 1$, then $\forall j \in \{1,\ldots,n\}$, either $|\psi_z(j)\rangle = H|x_i(j)\rangle$ or $|\psi_z(j)\rangle \in \{|0\rangle,|1\rangle\}$.
3. If $s_i = \overline{0}$, then $\exists j \in \{1,\ldots,n\}$ such that $|\psi_z(j)\rangle = |\overline{x_i(j)}\rangle$.
4. If $s_i = \overline{1}$, then $\exists j \in \{1,\ldots,n\}$ such that $|\psi_z(j)\rangle = H|\overline{x_i(j)}\rangle$.

Recall now that we set $s_1 = 0$ and $s_m = 1$, *i.e.* the first query was a successful Z-basis query and the last query was a successful X-basis query. Applying rules 1 and 2 above thus yields that for all indices k, $|\psi_z(k)\rangle \in \{|x_1(k)\rangle, H|x_m(k)\rangle\}$. Moreover, since $x_1 = x_2 = \cdots = x_{m-1}$, it follows that for all k, both assignments for $|\psi_z(k)\rangle$ are consistent for $|\psi_z(k)\rangle$. We conclude that the (s,x)-block of Q_1 has the following operator in register $\mathcal{X}_1$:

$$\sigma = \bigotimes_{k=1}^{m} \left(|x_1(k)\rangle\langle x_1(k)| + H|x_m(k)\rangle\langle x_m(k)|H\right). \tag{22}$$

But for any $b,c \in \{0,1\}$, $\lambda_{\max}(|b\rangle\langle b| + H|c\rangle\langle c|H) = 1 + \frac{1}{\sqrt{2}}$ (see, e.g., [MVW13]). Thus, $\lambda_{\max}(\sigma) = (1+\frac{1}{\sqrt{2}})^n$, as claimed.

We next show a matching upper bound of $\lambda_{\max}(Q_1) \leq \frac{2}{4^n}(1+\frac{1}{\sqrt{2}})^n$ among all (s,x)-blocks. For any $s \in S$, there exist indices $i \neq j$ such that x_i and x_j are a successful 0- and 1-query, respectively. Without loss of generality, assume $i = j = 1$. Then, as in the previous case, rules 1 and 2 imply that:

$$\forall k \in \{1,\ldots,n\}, \quad |\psi_z(k)\rangle \in \{|x_1(k)\rangle, H|x_m(k)\rangle\}. \tag{23}$$

Consider now any x_i for $1 < i < m$, and suppose without loss of generality that x_i is a 0-query, *i.e.* its first bit is set to 0. There are two cases to analyze:

- (Case 1: $s_i = 0$) In this case, both query 1 and query i are successful 0-queries; thus, they must agree on *all* secret key bits which were encoded in the Z basis. It follows from Rule 1 that for any bit k on which x_1 and x_i disagree, the secret key must have encoded bit k in the X-basis. In other words, $|\psi_z(k)\rangle = H|x_m(k)\rangle$ in Equation (23) (*i.e.* one of the two possibilities is eliminated). (If $x_1 = x_i$, on the other hand, no such additional constraint exists.)
- (Case 2: $s_i = \overline{0}$) In this case, query i is an unsuccessful 0-query. By Rule 3, there exists a bit k on which x_1 and x_k disagree, and whose corresponding secret key bit was encoded in the Z basis. In other words, $|\psi_z(k)\rangle = |x_1(k)\rangle$ in Equation (23) (*i.e.*, one of the two possibilities is eliminated).

The analysis for x_i being a 1-query is analogous. We conclude that for any (s,x)-block of Q_1, the operator in register $\mathcal{X}_1$ is of the form of σ from Equation (22), except that the some of the

indices k may contain an operator consisting of only 1 summand (e.g. $|x_1(k)\rangle\langle x_1(k)|$ instead of $|x_1(k)\rangle\langle x_1(k)| + H|x_m(k)\rangle\langle x_m(k)|H$). Since the omitted summands are all positive semidefinite, however, we conclude the eigenvalue on any (s,x)-block is at most the eigenvalue of σ from Equation (22), *i.e.*, at most $\lambda_{\max}(Q_1) \leq \frac{2}{4^n}(1+\frac{1}{\sqrt{2}})^n$, as claimed. □

D Proof of Lemma 4.3

Proof. Observe first that an honest receiver Alice wishing to extract s_i acts as follows. She applies a unitary $U_i \in \mathcal{U}(A \otimes B)$ to get state

$$|\phi_1\rangle := U_i|\psi\rangle_{AB}|0\rangle_C. \tag{24}$$

She then measures B in the computational basis and postselects on result $y \in \{0,1\}^n$, obtaining state

$$|\phi_2\rangle := |\phi_y\rangle_A|y\rangle_B|0\rangle_C. \tag{25}$$

She now treats y as a "key" for s_i, *i.e.*, she applies O_f to $B \otimes C$ to obtain her desired bit s_i, *i.e.*,

$$|\phi_3\rangle := |\phi_y\rangle_A|y\rangle_B|s_i\rangle_C. \tag{26}$$

A malicious receiver Bob wishing to extract s_0 and s_1 now acts similarly to the rewinding strategy for superposition queries. Suppose without loss of generality that s_0 has at most Δ keys. Then, Bob first applies U_0 to prepare $|\phi_1\rangle$ from Equation (24), which we can express as

$$|\phi_1\rangle = \sum_{y\in\{0,1\}^n} \alpha_y|\psi_y\rangle_A|y\rangle_B|0\rangle_C. \tag{27}$$

for $\sum_y |\alpha_y|^2 = 1$. Since measuring B next would allow us to retrieve s_0 in register C with certainty, we have that all y appearing in the expansion above satisfy $f(y) = s_i$. Moreover, since s_0 has at most Δ keys, there exists a key y' such that $|\alpha_{y'}|^2 \geq 1/\Delta$. Bob now measures B in the computational basis to obtain $|\phi_2\rangle$ from Equation (25), obtaining y' with probability at least $1/\Delta$. Feeding y' into O_f yields s_1. Having obtained y', we have that $|\langle\phi_1|\phi_2\rangle|^2 \geq 1/\Delta$, implying

$$\left|\langle\psi|U_0^\dagger|\phi_{y'}\rangle|y'\rangle\right|^2 \geq 1/\Delta, \tag{28}$$

i.e., Bob now applies $U_0^\dagger$ to recover a state with "large" overlap with initial state $|\psi\rangle$.

To next recover s_1, define $|\psi_{\text{good}}\rangle := U_1|\psi\rangle$ and $|\psi_{\text{approx}}\rangle := U_1U_0^\dagger|\phi_{y'}\rangle|y'\rangle$. Bob applies U_1 to obtain

$$|\psi_{\text{approx}}\rangle = \beta_1|\psi_{\text{good}}\rangle + \beta_2|\psi_{\text{good}}^\perp\rangle, \tag{29}$$

where $\sum_i |\beta_i|^2 = 1$, $\langle\psi_{\text{good}}|\psi_{\text{good}}^\perp\rangle = 0$, and $|\beta_1|^2 \geq 1/\Delta$. Define $\Pi_{\text{good}} := \sum_{y\in\{0,1\}^n \text{ s.t. } f(y)=s_1} |y\rangle\langle y|$. Then, the probability that measuring B in the computational basis now yields a valid key for s_1 is

$$\langle\psi_{\text{approx}}|\Pi_{\text{good}}|\psi_{\text{approx}}\rangle \geq |\beta_1|^2 \geq \frac{1}{\Delta}, \tag{30}$$

where we have used the fact that $\Pi_{\text{good}}|\psi_{\text{good}}\rangle = |\psi_{\text{good}}\rangle$ (since an honest receiver can extract s_1 with certainty). We conclude that Bob can extract both s_0 and s_1 with probability at least $1/\Delta^2$. □

References

[AC12] 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.

[BB84] Charles H. Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. *International Conference on Computers, Systems and Signal Processing*, pages 175–179, 1984.

[BBCS92] Charles H. Bennett, Gilles Brassard, Claude Crépeau, and Marie-Hélène Skubiszewska. Practical quantum oblivious transfer. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 351–366. Springer, August 1992.

[BDS18] Shalev Ben-David and Or Sattath. Quantum tokens for digital signatures. arXiv:1609.09047, 2018.

[Bea91] Donald Beaver. Secure multiparty protocols and zero-knowledge proof systems tolerating a faulty minority. *Journal of Cryptology*, 4(2):75–122, 1991.

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications. In *20th ACM STOC*, pages 103–112. ACM Press, May 1988.

[BGS13] Anne Broadbent, Gus Gutoski, and Douglas Stebila. Quantum one-time programs. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 344–360. Springer, August 2013.

[BGZ15] Anne Broadbent, Sevag Gharibian, and Hong-Sheng Zhou. Quantum one-time memories from stateless hardware. arXiv:1511.01363, November 2015.

[BM82] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo random bits. In *23rd FOCS*, pages 112–117. IEEE Computer Society Press, November 1982.

[BS16] Anne Broadbent and Christian Schaffner. Quantum cryptography beyond quantum key distribution. *Designs, Codes and Cryptography*, 78(1):351–382, 2016.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[Can00a] Ran Canetti. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology*, 13(1):143–202, 2000.

[Can00b] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. Cryptology ePrint Archive, Report 2000/067, 2000. `http://eprint.iacr.org/2000/067`.

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd FOCS*, pages 136–145. IEEE Computer Society Press, October 2001.

[CDPW07] Ran Canetti, Yevgeniy Dodis, Rafael Pass, and Shabsi Walfish. Universally composable security with global setup. In Salil P. Vadhan, editor, *TCC 2007*, volume 4392 of *LNCS*, pages 61–85. Springer, February 2007.

[CGLZ18] Kai-Min Chung, Marios Georgiou, Ching-Yi Lai, and Vassilis Zikas. Cryptography with dispensable backdoors. eprint:2018/352, 2018.

[CGS08] Nishanth Chandran, Vipul Goyal, and Amit Sahai. New constructions for UC secure computation using tamper-proof hardware. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 545–562. Springer, April 2008.

[Cho75] Man-Duen Choi. Completely positive linear maps on complex matrices. *Linear Alg. Appl.*, 10:285, 1975.

[CKS+14] Seung Geol Choi, Jonathan Katz, Dominique Schröder, Arkady Yerukhimovich, and Hong-Sheng Zhou. (efficient) universally composable oblivious transfer using a minimal number of stateless tokens. In Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 638–662. Springer, February 2014.

[CLOS02] Ran Canetti, Yehuda Lindell, Rafail Ostrovsky, and Amit Sahai. Universally composable two-party and multi-party secure computation. In *34th ACM STOC*, pages 494–503. ACM Press, May 2002.

[CM97] Christian Cachin and Ueli Maurer. Unconditional security against memory-bounded adversaries. In *Advances in Cryptology - CRYPTO 1997*, LNCS, pages 292–306. Springer, 1997.

[DFL+09] Ivan Damgård, Serge Fehr, Carolin Lunemann, Louis Salvail, and Christian Schaffner. Improving the security of quantum protocols via commit-and-open. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 408–427. Springer, August 2009.

[DFSS05] Ivan Damgård, Serge Fehr, Louis Salvail, and Christian Schaffner. Cryptography in the bounded quantum-storage model. In *Symposium on Foundations of Computer Science - FOCS 2005*, pages 449–458. IEEE, 2005.

[DNS10] Frédéric Dupuis, Jesper Buus Nielsen, and Louis Salvail. Secure two-party quantum evaluation of unitaries against specious adversaries. In *Advances in Cryptology – Proc. CRYPTO 2010*, LNCS, pages 685–706. Springer, 2010.

[DNS12] Frédéric Dupuis, Jesper Buus Nielsen, and Louis Salvail. Actively secure two-party evaluation of any quantum operation. In *Advances in Cryptology – Proc. CRYPTO 2012*, volume 7417 of *LNCS*, pages 794–811. Springer, 2012.

[DS13] Ivan Damgård and Alessandra Scafuro. Unconditionally secure and universally composable commitments from physical assumptions. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part II*, volume 8270 of *LNCS*, pages 100–119. Springer, December 2013.

[FKS+13] Serge Fehr, Jonathan Katz, Fang Song, Hong-Sheng Zhou, and Vassilis Zikas. Feasibility and completeness of cryptographic tasks in the quantum world. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 281–296. Springer, March 2013.

[Gav12] Dmitry Gavinsky. Quantum money with classical verification. In *Computational Complexity (CCC), 2012 IEEE 27th Annual Conference on*, pages 42–52, June 2012.

[GIS+10] Vipul Goyal, Yuval Ishai, Amit Sahai, Ramarathnam Venkatesan, and Akshay Wadia. Founding cryptography on tamper-proof hardware tokens. In Daniele Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 308–326. Springer, February 2010.

[GKR08] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. One-time programs. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 39–56. Springer, August 2008.

[GL91] Shafi Goldwasser and Leonid A. Levin. Fair computation of general functions in presence of immoral majority. In Alfred J. Menezes and Scott A. Vanstone, editors, *CRYPTO'90*, volume 537 of *LNCS*, pages 77–93. Springer, August 1991.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In Alfred Aho, editor, *19th ACM STOC*, pages 218–229. ACM Press, May 1987.

[GW07] Gus Gutoski and John Watrous. Toward a general theory of quantum games. In *Proceedings of the 39th ACM Symposium on Theory of Computing (STOC 2007)*, pages 565–574, 2007. arXiv:quant-ph/0611234v2.

[Hei27] Werner Heisenberg. Schwankungserscheinungen und quantenmechanik. *Zeitschrift fuer Physik*, 40(7):501–506, July 1927.

[HSS11] Sean Hallgren, Adam Smith, and Fang Song. Classical cryptographic protocols in a quantum world. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 411–428. Springer, August 2011.

[IPS08] Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. Founding cryptography on oblivious transfer - efficiently. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 572–591. Springer, August 2008.

[Jam72] Andrzej Jamiołkowski. Linear transformations which preserve trace and positive semi-definiteness of operators. *Rep. Math. Phys.*, 3:275, 1972.

[Kat07] Jonathan Katz. Universally composable multi-party computation using tamper-proof hardware. In Moni Naor, editor, *EUROCRYPT 2007*, volume 4515 of *LNCS*, pages 115–128. Springer, May 2007.

[Kil88] Joe Kilian. Founding cryptography on oblivious transfer. In *20th ACM STOC*, pages 20–31. ACM Press, May 1988.

[KMPS14] Daniel Kraschewski, Hemanta K. Maji, Manoj Prabhakaran, and Amit Sahai. A full characterization of completeness for two-party randomized function evaluation. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 659–676. Springer, May 2014.

[KMQ11] Daniel Kraschewski and Jörn Müller-Quade. Completeness theorems with constructive proofs for finite deterministic 2-party functions. In Yuval Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 364–381. Springer, March 2011.

[Liu14a] Yi-Kai Liu. Building one-time memories from isolated qubits. In Moni Naor, editor, *ITCS 2014*, pages 269–286. ACM, January 2014.

[Liu14b] Yi-Kai Liu. Single-shot security for one-time memories in the isolated qubits model. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part II*, volume 8617 of *LNCS*, pages 19–36. Springer, August 2014.

[Liu15] Yi-Kai Liu. Privacy amplification in the isolated qubits model. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 785–814. Springer, April 2015.

[LPV09] Huijia Lin, Rafael Pass, and Muthuramakrishnan Venkitasubramaniam. A unified framework for concurrent security: universal composability from stand-alone non-malleability. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 179–188. ACM Press, May / June 2009.

[Mau92] Ueli M. Maurer. Protocols for secret key agreement by public discussion based on common information. In *Advances in Cryptology - CRYPTO 1992*, volume 740 of *LNCS*, pages 461–470. Springer, 1992.

[MPR09] Hemanta K. Maji, Manoj Prabhakaran, and Mike Rosulek. Complexity of multi-party computation problems: The case of 2-party symmetric secure function evaluation. In Omer Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 256–273. Springer, March 2009.

[MPR10] Hemanta K. Maji, Manoj Prabhakaran, and Mike Rosulek. A zero-one law for cryptographic complexity with respect to computational UC security. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 595–612. Springer, August 2010.

[MR92] Silvio Micali and Phillip Rogaway. Secure computation (abstract). In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 392–404. Springer, August 1992.

[MR11] Ueli Maurer and Renato Renner. Abstract cryptography. In Bernard Chazelle, editor, *ICS 2011*, pages 1–21. Tsinghua University Press, January 2011.

[MVW13] Abel Molina, Thomas Vidick, and John Watrous. Optimal counterfeiting attacks and generalizations for wiesner's quantum money. In Kazuo Iwama, Yasuhito Kawano, and Mio Murao, editors, *Theory of Quantum Computation, Communication, and Cryptography*, volume 7582 of *Lecture Notes in Computer Science*, pages 45–64. Springer Berlin Heidelberg, 2013.

[PR08] Manoj Prabhakaran and Mike Rosulek. Cryptographic complexity of multi-party computation problems: Classifications and separations. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 262–279. Springer, August 2008.

[PS04] Manoj Prabhakaran and Amit Sahai. New notions of security: Achieving universal composability without trusted setup. In László Babai, editor, *36th ACM STOC*, pages 242–251. ACM Press, June 2004.

[PW01] Birgit Pfitzmann and Michael Waidner. A model for asynchronous reactive systems and its application to secure message transmission. In *Proc. 22nd IEEE Symposium on Security & Privacy (S&P) 2001*, pages 184–200. IEEE, 2001. Full version available at `http://eprint.iacr.org/2000/066`.

[PYJ+12] Fernando Pastawski, Norman Y Yao, Liang Jiang, Mikhail D Lukin, and J Ignacio Cirac. Unforgeable noise-tolerant quantum tokens. *Proceedings of the National Academy of Sciences*, 109(40):16079 –16082, 2012.

[Ren08] Renato Renner. *Security of Quantum Key Distribution*. PhD thesis, ETH Zürich (Switzerland), September 2008.

[Unr10] Dominique Unruh. Universally composable quantum multi-party computation. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 486–505. Springer, May 2010.

[Unr13] Dominique Unruh. Everlasting multi-party computation. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 380–397. Springer, August 2013.

[Unr14] Dominique Unruh. Revocable quantum timed-release encryption. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 129–146. Springer, May 2014.

[Wat11] John Watrous. Lecture 7: Semidefinite programming, 2011. Latest version available at: `https://cs.uwaterloo.ca/~watrous/CS766/LectureNotes/07.pdf`.

[Wie83] Stephen Wiesner. Conjugate coding. *ACM SIGACT News*, 15(1):78–88, 1983. Original article written circa 1970.

[Win99] Andreas Winter. Coding theorem and strong converse for quantum channels. *IEEE Transactions on Information Theory*, 45:2481–2485, 1999.

[WST08] Stephanie Wehner, Christian Schaffner, and Barbara M. Terhal. Cryptography from noisy storage. *Physical Review Letters*, 100(22):220502, June 2008.

[WW10] Stephanie Wehner and Andreas Winter. Entropic uncertainty relations—a survey. *New J. Phys.*, 12(2):025009, Feb 2010.

[WZ82] William K. Wootters and Wojciech H. Zurek. A single quantum cannot be cloned. *Nature*, 299(5886):802–803, 1982.

[Yao82] Andrew Chi-Chih Yao. Theory and applications of trapdoor functions. In *23rd FOCS*, pages 80–91. IEEE Computer Society Press, November 1982.