Towards Quantum One-Time Memories from Stateless Hardware

Anne Broadbent[1], Sevag Gharibian[2], and Hong-Sheng Zhou[3]

[1]Department of Mathematics and Statistics, University of Ottawa, Ontario, Canada

[2]Department of Computer Science, Paderborn University, Germany, and Virginia Commonwealth University, USA

[3]Department of Computer Science, Virginia Commonwealth University, Virginia, USA

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 making at most $0.114n$ adaptive queries to the token (for n the key size), in the quantum universal composability framework, but leave open the question of security against a polynomial amount of 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 give two impossibility results showing certain assumptions in our scheme cannot be relaxed.

1 Introduction

Theoretical cryptography centers around 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; Dam+05], or make *trusted setups* available to honest players [Kil88; BFM88; Can01; Can+02; IPS08; PR08; LPV09; MPR09; MPR10; MR11; KMQ11; Kra+14], 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.

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

Anne Broadbent: abroadbe@uottawa.ca
Sevag Gharibian: sevag.gharibian@upb.de
Hong-Sheng Zhou: hszhou@vcu.edu

implement statistically secure OTPs for any efficiently computable program in the universal composability (UC) framework [Goy+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; Goy+10; Cho+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.*, [Goy+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. We propose what is, to our knowledge, the first prepare-and-measure quantum protocol that constructs OTMs from stateless hardware tokens. For this protocol, we are able to rigorously prove information theoretic security against an adversary making a *linear* (in n, the security parameter) number of adaptive queries to the token. While we conjecture that security holds also for *polynomially* many queries, note that already in this setting of linearly many adaptive queries, our protocol achieves something impossible classically (*i.e.*, classically, obtaining security against a linear number of queries is impossible). We also show stand-alone security against a malicious sender.

Historical Note. We proposed the concept that quantum information could provide a "stateless to stateful" transformation in a preliminary version of this work [BGZ15]; however, that work claimed security against a *polynomial* number of token queries, obtained via a reduction from the interactive to the non-interactive setting. We thank an anonymous referee for catching a subtle, but important bug which appears to rule out the proof approach of [BGZ15]. The current paper hence employs a different proof approach, which models interaction with the token as a "quantum game" via semidefinite programming (further details below). Since our original paper was posted, recent work [Chu+19] has shown an alternate quantum "stateful to stateless" transformation via quantum money constructions [BDS18]. Specifically, in [Chu+19], security against a polynomial number of queries is achieved, albeit with respect to a new definition of "OTMs relative to an oracle" (while the security results of the present paper are with respect to the well-established simulation-based definition of [Goy+10; Kat07]). Furthermore, [Chu+19] directly applies known quantum money constructions, which require difficult-to-prepare highly entangled states. Our focus here, in contrast, is to take a "first-principles" approach and build a technologically simple-to-implement scheme which requires no entanglement, but rather the preparation of just one of four single qubit states, $|0\rangle, |1\rangle, |+\rangle, |-\rangle$. Indeed, the two works are arguably complementary in that the former focuses primarily on *applications* of "stateful" single-use tokens, while our focus is on the most technologically simple way to *implement* such "stateful" tokens.

Construction. Our construction is inspired by Wiesner's *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, $|\psi\rangle$ (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.[1] The honest use of the OTM is thus intuitive: for choice bit $b = 0$, the user measures each qubit of the quantum key in the rectilinear basis to obtain an n-bit string y, and inputs (b, y) into the hardware token. If $b = 1$, the same process is applied, but with measurements in the diagonal basis.

Assumption. Crucially, we assume the hardware token accepts *classical* input only (alternatively and equivalently, the token immediately measures its quantum input in the standard basis), *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 present in Wiesner's proposal for quantum money [Wie83], and is often given a physical explanation in terms of the no-cloning theorem [WZ82] or 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 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. We conjecture, however, that a similar statement holds for any $m \in \text{poly}(n)$, *i.e.*, that the protocol is secure against polynomially many queries.

To show security against linearly many queries, we exploit the semidefinite programming-based quantum games framework of Gutoski and Watrous (GW) [GW07] to model interaction with the token. Intuitively, GW is useful for our setting, since it is general enough to model multiple rounds of adaptive queries to the token, even when the receiver holds quantum "side information" in the form of $|\psi\rangle$. We describe this technique in Sections 2.1 and 3.4, and provide full details in 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 at most* cn *queries for any* $c < 0.114$.

As stated above, we conjecture that our protocol is actually secure against polynomially many adaptive queries. However, we are unable to show this claim using our present proof techniques, and hence leave this question open. Related to this, we make the following comments: (1) As far as we are aware, the Main Theorem above is the only known formal proof of any type of security for conjugate coding in the interactive setting with $\Omega(1)$ queries. Moreover, as stated earlier, classical security against $\Omega(1)$ queries is trivially impossible. (2) Our proof introduces the GW semidefinite programming framework from quantum interactive proofs to the study of conjugate coding-based schemes. This framework allows

[1] We note that a simple modification using a classical one-time pad could be used to make *both* the quantum state and hardware token independent of s_0 and s_1: the token would output one of two uniformly random bits r_0 and r_1, which could each be used to decrypt a single bit, s_0 or s_1.

handling multiple challenges in a unified fashion: arbitrary quantum operations by the user, classical queries to the token, and the highly non-trivial assumption of quantum side information for the user (the "quantum key" state sent to the user.)

Towards security against polynomially many queries. Regarding the prospects of proving security against polynomially many adaptive queries, we generally believe it requires a significant new insight into how to design a "good" feasible solution to the primal semidefinite program (SDP) obtained via GW. However, in addition to our proof for linear security (Theorem C.5), in Appendix D we attempt to give evidence towards our conjecture for polynomial security. Namely, Appendix D.1 first simplifies the SDPs obtained from GW, and derives the corresponding dual SDPs. We remark these derivations apply for any instantiation of the GW framework, *i.e.* they are not specific to our setting, and hence may prove useful elsewhere. In Appendix D.2, we then give a feasible solution Y (Equation (128)) to the dual SDP. While Y is simple to state, it is somewhat involved to analyze. A heuristic analysis suggests Y's dual objective function value has roughly the behavior needed to show security, *i.e.* the value scales as $m/\sqrt{2^n}$, for m queries and n key bits. If Y were to be the *optimal* solution to the dual SDP, this would strongly suggest the optimal cheating probability is essentially $m/\sqrt{2^n}$. However, we explicitly show Y is not optimal, and so $m/\sqrt{2^n}$ is only a *lower bound* on the optimal cheating probability[2]. Nevertheless, we conjecture that while Y is not optimal, it is *approximately* optimal (see Conjecture D.2 for a precise statement); this would imply the desired polynomial security claim. Unfortunately, the only techniques we are aware of to show such approximate optimality involve deriving a better primal SDP solution, which appears challenging.

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; Pas+12], a reduction from oblivious transfer to bit commitment [Ben+92; Dam+09] and to other primitives such as "cut-and choose" functionality [Feh+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 [Dam+05; 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 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 (SDP) approaches have been applied to analyze security of conjugate coding [MVW13] for quantum money, in the setting of one round of interaction with a "stateful" bank. SDPs are also the technical tool we adopt for our proof (Section 3.4 and Appendix C), though here we require the more advanced quantum games SDP framework of Gutoski and Watrous [GW07] to deal with multiple adaptive interactions with stateless tokens. Reference [Pas+12] has also made use of Gavinsky's [Gav12] quantum retrieval games framework.

Continuing with proof techniques, somewhat similar to [Pas+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 [Pas+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 [Pas+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. Thus, [MVW13], for example, may be viewed as using a "stateful" verifier, whereas our focus here is on a "stateless" verifier (*i.e.*, a token).

Also, we note that prior work has achieved oblivious transfer using quantum information, together with some assumption (*e.g.*, bit commitment [Ben+92] or bounded quantum storage [Dam+05]). These protocols typically use an interaction phase similar to the "commit-and-open" protocol of [Ben+92]; because we are working in the non-interactive setting, these techniques appear to be inapplicable.

[2]Indeed, an attack in the Breidbart basis breaks our scheme with probability $2^{-0.228n}$, as observed by David Mestel; see Section D.2.

Finally, Liu [Liu14a; Liu14b; Liu15] has given stand-alone secure OTMs using quantum information in the *isolated-qubit model*. Liu's approach is nice in that it avoids the use of trusted setups. In return, however, Liu must use the isolated-qubit model, which restricts the adversary to perform only single-qubit operations (no entangling gates are permitted); this restriction is, in some sense, necessary if one wants to avoid trusted setups, as a secure OTM in the plain quantum model cannot exist (see Section 4). In contrast, in the current work we allow unbounded and unrestricted quantum adversaries, but as a result require a trusted setup. In addition, we remark the security notion of OTMs of [Liu14a; Liu14b; Liu15] is weaker than the simulation-based notion studied in this paper, and it remains an interesting open question whether the type of OTM in [Liu14a; Liu14b; Liu15] is secure under composition (in the current work, the UC framework gives us security under composition for free).

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, which is widely used as the "benchmark" for a "physically feasible" quantum protocol (in this model, one needs only the ability to prepares single-qubit states $|0\rangle, |1\rangle, |+\rangle, |-\rangle$, and to perform single-qubit projective measurements. In particular, no entangled states are required, and in principle no quantum memory is required, since qubits can be measured one-by-one as they arrive). 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 manufacture (e.g. analogous to an RFID tag) 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 (recall we show security for a linear number of queries, and conjecture security for polynomially many queries). Finally, our security analysis is in the quantum UC framework against a corrupted receiver; this means our protocol can be easily composed with many others; for example, 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[5]), 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 that 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

[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 [Pas+12].

[4] This is common in the oracle model of quantum computation, where a function $f : \{0,1\}^n \mapsto \{0,1\}$ is implemented via the (self-inverse) unitary mapping $U_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle$.

[5] The term "measure-and-access" here is not to be confused with "prepare-and-measure". We define the former in Section 4.2 to mean a protocol in which one measures a given quantum resource state to extract a classical key, which is then used for a desired purpose. "Prepare and measure", in contrast, is referring to the fact that our scheme is easy to implement; the preparer of the token just needs to prepare single-qubit states, and an honest user simply measures them.

this to be the case, but finding a formal proof has been elusive. (See discussion under "Towards security against polynomially many adaptive queries" above for details.)

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?[6]

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? In Section 4, we briefly discuss why it may be difficult to extend the techniques of our impossibility results straightforwardly when the adversary does *not* have access to the inverse of the token.

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.

Organization. We begin in Section 2 with preliminaries, including the ideal functionalities for an OTM and stateless token, background on quantum channels, semidefinite programming, and the Gutoski-Watrous framework for quantum games. 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. Appendix D gives a simplification of the GW SDP, derives its dual, and gives a dual feasible solution which we conjecture to be approximately optimal (formally stated in Conjecture D.2). Finally, the security proof for a lemma in Section 4 can be found in Appendix E.

2 Preliminaries

Notation. Two binary distributions $\mathbf{X}$ and $\mathbf{Y}$ are *indistinguishable*, denoted $\mathbf{X} \approx \mathbf{Y}$, if

$$|\Pr(X_n = 1) - \Pr(Y_n = 1)| \leq \mathrm{negl}(n). \tag{1}$$

We define single-qubit $|0\rangle_+ = |0\rangle$ and $|1\rangle_+ = |1\rangle$, so that $\{|0\rangle_+, |1\rangle_+\}$ form the *rectilinear basis*. We 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 $\theta = \theta_1, \theta_2, \ldots, \theta_n \in \{+, \times\}^n$, define $|x\rangle_\theta = \bigotimes_{i=1}^n |x_i\rangle_{\theta_i}$. The Hadamard gate in quantum information is $H = (1\ 1;\ 1\ -1)/\sqrt{2}$. It maps $H|0\rangle_+ = |0\rangle_\times$, $H|1\rangle_+ = |1\rangle_\times$, $H|0\rangle_\times = |0\rangle_+$, and $H|1\rangle_\times = |1\rangle_+$. For $\mathcal{X}$ a finite dimensional complex Hilbert space, $\mathcal{L}(\mathcal{X})$, $\mathrm{Herm}(\mathcal{X})$, $\mathrm{Pos}(\mathcal{X})$, 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 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 $\mathcal{F}_{\mathtt{OTM}}$ 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 $\mathcal{F}_{\mathtt{OTM}}$ as an *OTM token*.

Stateless hardware. The original work of Katz [Kat07] introduces the ideal functionality $\mathcal{F}_{\mathtt{wrap}}$ to model stateful tokens in the UC-framework. In the ideal model, a party that wants to create a token,

[6] We note that this would require a different protocol, since in our current construction, a cheating sender could program the token to abort based on the user's input.

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.

sends a 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.

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.

Although the environment and adversary are unbounded, we specify that stateless hardware can be queried only a polynomial number of times. This is necessary; otherwise the hardware token model is vacuous (with unbounded queries, the entire input-output behavior of stateless hardware can be extracted).

Quantum channels. We now review quantum channels. A basic background in quantum information is assumed, see e.g. [NC00] for a standard reference. 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). Such maps take density operators to density operators. 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 $J(\Phi) = \sum_{i,j} \Phi(|i\rangle\langle j|) \otimes |i\rangle\langle j|$. 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 (GW) 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 Hermiticity-preseving linear map $\Phi : \mathcal{L}(\mathcal{X}) \mapsto \mathcal{L}(\mathcal{Y})$, 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 (66) will for simplicity not be written in precisely the form above, but can without loss of generality be made so.

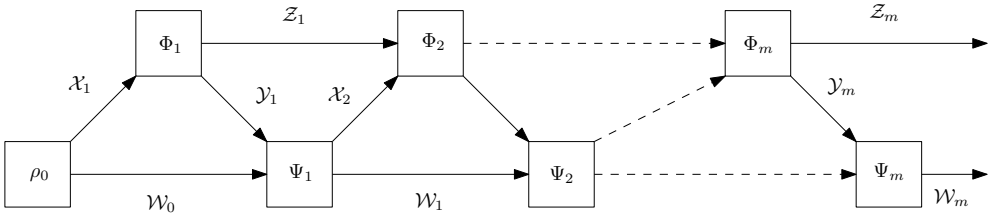

Figure 1: A general interaction between two quantum parties.

2.1 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 1. Here, we imagine one party, A, prepares an initial 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 $\Lambda = \{\Lambda_0, \Lambda_1\}$, meaning $\Lambda_0, \Lambda_1 \succeq 0$, $\Lambda_0 + \Lambda_1 = I$) in order to decide whether to reject (Λ_0) or accept (Λ_1). As done in [GW07], we may assume without loss of generality[7] that all channels are given by linear isometries[8] A_k, i.e. $\Phi_k(X) = A_k X A_k^\dagger$. 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 1).

Intuitively, since our protocol (Section 3.1) will begin with the token sending the user a quantum key $|x\rangle_\theta$, we will later model the token as a *measuring co-strategy*, and the user of the token as a *strategy*. The advantage to doing so is that the GW framework allows one to (recursively) characterize any such strategy (resp., co-strategy) via a set of linear (in)equalities and positive semi-definite constraints. (In this sense, the GW framework generalizes the Choi-Jamiołkowski representation for channels to a "Choi-Jamiołkowski" representation for strategies/co-strategies.) To state these constraints, we first write down the Choi-Jamiołkowski (CJ) representation of a strategy (resp., measuring co-strategy) from [GW07].

CJ representation of (non-measuring) strategy. The CJ representation of a strategy $(A_1, \ldots, A_m)$ is given by matrix [GW07]

$$\mathrm{Tr}_{\mathcal{Z}_m}(\mathrm{vec}(A)\,\mathrm{vec}(A)^\dagger), \tag{2}$$

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,

$$A := (I_{\mathcal{Y}_1 \otimes \cdots \otimes \mathcal{Y}_{m-1}} \otimes A_m) \cdots (A_1 \otimes I_{\mathcal{X}_2 \otimes \cdots \otimes \mathcal{X}_m}), \tag{3}$$

and the $\mathrm{vec} : \mathcal{L}(\mathcal{S}, \mathcal{T}) \mapsto \mathcal{T} \otimes \mathcal{S}$ 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 $\Lambda := \{\Lambda_0, \Lambda_1\}$ denote a POVM with reject and accept measurement operators Λ_0 and Λ_1, respectively. A measuring strategy which ends with a measurement with respect to POVM Λ replaces, for $\Lambda_a \in \Lambda$, Equation (2) with [GW07]

$$\begin{aligned} Q_a &:= \mathrm{Tr}_{\mathcal{Z}_m}((\Lambda_a \otimes I_{\mathcal{Y}_1 \otimes \cdots \otimes \mathcal{Y}_m})\,\mathrm{vec}(A)\,\mathrm{vec}(A)^\dagger) && (4)\\ &= \mathrm{Tr}_{\mathcal{Z}_m}(\mathrm{vec}((\sqrt{\Lambda_a} \otimes I_{\mathcal{Y}_1 \otimes \cdots \otimes \mathcal{Y}_m})A)\,\mathrm{vec}((\sqrt{\Lambda_a} \otimes I_{\mathcal{Y}_1 \otimes \cdots \otimes \mathcal{Y}_m})A)^\dagger) && (5)\\ &=: \mathrm{Tr}_{\mathcal{Z}_m}(\mathrm{vec}(B_a)\,\mathrm{vec}(B_a)^\dagger). && (6) \end{aligned}$$

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.1, all operators we derive will be symmetric with respect to the standard basis, and hence taking this transpose will be unnecessary.)

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

[8] A linear isometry $A \in \mathcal{L}(\mathcal{S}, \mathcal{T})$ satisfies $A^\dagger A = I_{\mathcal{S}}$, generalizing the notion of unitary maps to non-square matrices.

Optimization characterization over strategies and co-strategies. With CJ representations for strategies and co-strategies in hand, one can formulate [GW07] the optimal probability with which a strategy can force a corresponding co-strategy to output a desired result as follows. Fix any Q_a from a measuring co-strategy $\{Q_0, Q_1\}$, as in Equation (6). Then, Corollary 7 and Theorem 9 of [GW07] show that the maximum probability with which a (non-measuring) strategy can force the co-strategy to output result a is given by

$$\begin{aligned}
\text{min:} \quad & p && && (7)\\
\text{subject to:} \quad & Q_a \preceq pR_m && && (8)\\
& R_k = P_k \otimes I_{\mathcal{Y}_k} && \text{for } 1 \le k \le m && (9)\\
& \mathrm{Tr}_{\mathcal{X}_k}(P_k) = R_{k-1} && \text{for } 1 \le k \le m && (10)\\
& R_0 = 1 && && (11)\\
& R_k \in \mathrm{Pos}(\mathcal{Y}_{1,\ldots,k} \otimes \mathcal{X}_{1,\ldots,k}) && \text{for } 1 \le k \le m && (12)\\
& P_k \in \mathrm{Pos}(\mathcal{Y}_{1,\ldots,k-1} \otimes \mathcal{X}_{1,\ldots,k}) && \text{for } 1 \le k \le m && (13)\\
& p \in [0,1] && && (14)
\end{aligned}$$

Intuition. The minimum p denotes the optimal "success" probability, meaning the optimal probability of forcing the co-strategy to output a (Theorem 9 of [GW07]). The variables above, in addition to p, are $\{R_i\}$ and $\{P_i\}$, where the optimization is happening over all m-round co-strategies R_m satisfying Equation (8). How do we enforce that R_m encodes such an m-round co-strategy? This is given by the (recursive) Equations (9)-(13). Specifically, Corollary 7 of [GW07] states that R_m is a valid m-round co-strategy if and only if all of the following hold: (1) $R_m \succeq 0$, (2) $R_m = P_m \otimes I_{\mathcal{Y}_m}$ for $P_m \succeq 0$ and $\mathcal{Y}_m$ the last incoming message register to the co-strategy, (3) $\mathrm{Tr}_{\mathcal{X}_m}(P_m)$ is a valid $m-1$ round co-strategy (this is the recursive part of the definition). An intuitive sense as to why conditions (2) and (3) should hold is as follows: For any m-round co-strategy R_m, let R_{m-1} denote R_m restricted to the first $m-1$ rounds. Then, to operationally obtain R_{m-1} from R_m, the co-strategy first ignores the last incoming message in register $\mathcal{Y}_m$. This is formalized via a partial trace over $\mathcal{Y}_m$, which (once pushed through the CJ formalism[9]) translates into the $\otimes I_{\mathcal{Y}_k}$ term in Equation (9). Since the co-strategy is now ignoring the last *incoming* message $\mathcal{Y}_m$, any measurement it makes after $m-1$ rounds is independent of the last *outgoing* message $\mathcal{X}_m$. Thus, we can trace out $\mathcal{X}_m$ as well, obtaining a co-strategy R_{m-1} on just the first $m-1$ rounds; this is captured by Equation (10).

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 3.1). In Section 3.3, we describe the Simulator and prove Theorem 3.1 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 $\mathcal{F}_{\mathtt{wrap}}$ 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} acts as follows:
  - The sender chooses uniformly random $x \in_R \{0,1\}^n$ and $\theta \in_R \{+,\times\}^n$, and prepares $|x\rangle_\theta$. Based on tuple (s_0, s_1, x, θ), the sender then prepares the program M as in **Program 1**.
  - The sender sends $|x\rangle_\theta$ to the receiver.
  - The sender sends (create, M) to functionality $\mathcal{F}_{\mathtt{wrap}}$, and the functionality sends create to notify the receiver.

[9] Recall that the CJ representation of the trace map is the identity matrix (up to scaling).

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 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. 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. 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 details in Appendix B).

3.3 UC-Security against a corrupt receiver

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

Theorem 3.1. *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 3.1, we must construct and analyze an appropriate simulator, which we now do.

3.3.1 The simulator

In order to prove Theorem 3.1, for an adversary $\mathcal{A}$ that corrupts the receiver, we 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 3.2 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 is at most cn for any $c < 0.114$. 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 $O(2^{2m-0.228n})$. 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 3.2, 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

$$|\psi\rangle = \frac{1}{2^n} \sum_{z \in |0,1\rangle^{2n}} |\psi_z\rangle_R |z\rangle_T. \tag{15}$$

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

$$\rho_R := \frac{1}{2^{2n}} \sum_{z \in \{0,1\}^{2n}} |\psi_z\rangle\langle\psi_z|. \tag{16}$$

Theorem 3.2. *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 proof of Theorem 3.2 is in Appendix C. Its intuition can be sketched as follows.

Proof intuition. The challenge in analyzing security of the protocol is the fact that the receiver (a.k.a. the user) is not only given adaptive query access to the token, but also a copy of the quantum "resource state" ρ_R, which it may arbitrarily tamper with (in any manner allowed by quantum mechanics) while making queries. Luckily, the GW framework [GW07] (Section 2.1)) is general enough to model such "queries with quantum side information". The framework outputs an SDP, Γ (Equation (17)), the optimal value of which will encode the optimal cheating probability for a cheating user of our protocol. Giving a feasible solution for Γ will hence suffice to upper bound this cheating probability, yielding Theorem 3.2.

Coherently modeling quantum queries to the token. To model the interaction between the token and user, we first recall that all queries to the token must be classical by assumption. To model this process *coherently* in the GW framework, we hence imagine (solely for the purposes of the security analysis) that the token behaves as follows:

1. It first sends state ρ_R to the user.

2. When it receives as ith query a quantum state ρ_i from the user, it sends response string r_i to the user, and "copies" ρ_i via transversal CNOT gates to a private memory register $\mathcal{W}_i$, along with r_i. It does not access ρ_i again throughout the protocol, and only accesses r_i again in Step 3. For clarity, the token runs a classical circuit, and in the formal setup of Appendix C (see Remark (C.2)), the token conditions each response r_i solely on the current incoming message, ρ_i.

3. After all rounds of communication, the token "measures" its stored responses $(r_1, \ldots, r_m)$ in the Z-basis to decide whether to accept (user successfully cheated[10]) or reject (user failed to cheat).

The "copying" phase of Step 2 accomplishes two tasks: First, since the token will never read the "copies" of ρ_i again, the principle of deferred measurement [NC00] implies the transversal CNOT gates effectively simulate measuring ρ_i in the standard basis. In other words, without loss of generality the user is reduced to feeding a classical string $\widetilde{y}$ to the token. Second, we would like the entire security analysis to be done in a unified fashion in a single framework, the GW framework. To this end, we want the token itself to "decide" at the end of the protocol whether the user has successfully cheated (i.e. extracted both secret bits). Storing all responses r_i in Step 2 allows us to simulate such a final measurement in Step 3. We reiterate that, crucially, once the token "copies" ρ_i and r_i to W_i, it (1) never accesses (i.e. reads or writes to) ρ_i again and (2) only accesses r_i again in the final standard basis measurement of Step 3. Together, these ensure all responses r_i are independent, as required for a stateless token. A more formal justification is in Remark C.2 of Appendix C.

Formalization in GW framework. To place the discussion thus far into the formal GW framework, we return to Figure 1. The bottom "row" of Figure 1 will depict the token's actions, and the top row the user's actions. As outlined above, 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 $\widetilde{y}_i$ (which is, as discussed above a classical string without loss of generality) to the token via register $\mathcal{Y}_i$. Given any such query $\widetilde{y}_i$ in round i, the token applies its own TPCP map Ψ_i to determine how to respond to the query. In our protocol, the Ψ_i correspond to coherently applying a classical circuit, i.e. a sequence of unitary gates mapping the standard basis to itself. Specifically, their action 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", as recall from above that in the security analysis we model each Ψ_i as classically copying $(\widetilde{y}_i, r_i)$ to a distinct private register W_i.) Finally, after receiving the mth query $\widetilde{y}_m$ in register $\mathcal{Y}_m$, we imagine the token makes a measurement (not depicted in Fig. 1) based on the query responses $(r_1, \ldots, r_m)$ it returned; if the user managed to extract both s_0 and s_1 via queries, then the token "accepts"; otherwise it "rejects". (Again, we are using the fact that in our security analysis, the token keeps a history of all its responses r_i, solely for the sake of this final measurement.)

[10] We model the token as "accepting" when the user successfully cheats, so that a feasible solution to the semidefinite program Γ of Equation (17) correctly *upper bounds* the probability of said cheating. Formally, in the GW framework (Section 2.1), we will let Λ_1 denote this accepting measurement for the token; see Appendix C.

With this high-level setup in place, the output of the GW framework is a semidefinite program[11], denoted Γ (see Appendix C for further details):

$$\begin{aligned}
\text{min:} \quad & p && && (17)\\
\text{subject to:} \quad & Q_1 \preceq R_{m+1} && && (18)\\
& R_k = P_k \otimes I_{\mathcal{Y}_k} && \text{for } 1 \leq k \leq m+1 && (19)\\
& \mathrm{Tr}_{\mathcal{X}_k}(P_k) = R_{k-1} && \text{for } 1 \leq k \leq m+1 && (20)\\
& R_0 = p && && (21)\\
& R_k \in \mathrm{Pos}(\mathcal{Y}_{1,\ldots,k} \otimes \mathcal{X}_{1,\ldots,k}) && \text{for } 1 \leq k \leq m+1 && (22)\\
& P_k \in \mathrm{Pos}(\mathcal{Y}_{1,\ldots,k-1} \otimes \mathcal{X}_{1,\ldots,k}) && \text{for } 1 \leq k \leq m+1 && (23)
\end{aligned}$$

Above, Q_1 encodes the actions of the token, i.e. the co-strategy in the bottom row of Figure 1. The variable p denotes an upper bound on the optimal cheating probability (*i.e.*, the probability with which both s_0 and s_1 are extracted), subject to linear constraints (Equations (19)-(23)) which enforce that operator R_{m+1} encodes a valid co-strategy (see Section 2.1). Theorem 9 of [GW07] now says that the minimum p above encodes precisely the optimal cheating probability for a user which is constrained only by the laws of quantum mechanics. Since Γ is a minimization problem, to upper bound the cheating probability it hence suffices to give a feasible solution $(p, R_1, \ldots, R_{m+1}, P_1, \ldots, P_{m+1})$ for Γ, which will be our approach.

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 (3) of Appendix C, one uses tensor products with identity matrices appropriately). Then, the Choi-Jamiołkowski representation of A is given by [GW07] (see Section 2.1)

$$\mathrm{Tr}_{\mathcal{Z}_m}(\mathrm{vec}(A)\,\mathrm{vec}(A)^\dagger), \tag{24}$$

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 2.1.) However, since in our security analysis, we imagine the token also makes a final measurement via some POVM $\Lambda = \{\Lambda_0, \Lambda_1\}$, whereupon obtaining outcome Λ_1 the token "accepts", and upon outcome Λ_0 the token rejects, we require a slightly more complicated setup. Letting $B_1 := \Lambda_1 A$, we define Q_1 as [GW07]

$$Q_1 = \mathrm{Tr}_{\mathcal{Z}_m}(\mathrm{vec}(B_1)\,\mathrm{vec}(B_1)^\dagger). \tag{25}$$

The full derivation of Q_1 in our setting takes a few steps (App. C). Here, we state a slightly simplified version of Q_1 for exposition with intuition:

$$Q_1 = \frac{1}{4^n} \sum_{\text{"successful" } r} |r_m\rangle\langle r_m|_{\mathcal{X}_{m+1}} \otimes \cdots \otimes |r_1\rangle\langle r_1|_{\mathcal{X}_2} \otimes \tag{26}$$

$$\left(\sum_{\substack{\text{messages } \widetilde{y} \text{ and keys } z \\ \text{consistent with } r}} |\widetilde{y}_m\rangle\langle\widetilde{y}_m|_{\mathcal{Y}_m} \otimes \cdots \otimes |\widetilde{y}_1\rangle\langle\widetilde{y}_1|_{\mathcal{Y}_1} \otimes |\psi_z\rangle\langle\psi_z|_{\mathcal{X}_1} \right). \tag{27}$$

Above, recall each string r_i denotes the response of the token given the ith query $\widetilde{y}_i$ from the user; hence, the corresponding projectors in Q_1 act on spaces $\mathcal{X}_2$ through $\mathcal{X}_{m+1}$. We say r is "successful" if it encodes the user successfully extracting both secret bits from the token. Each string $\widetilde{y}_i \in \{0,1\}^{n+1}$ denotes the

[11] Note the optimization in Equation (17) differs from that in Equation (7). This is because, technically, Equation (17) is not yet an SDP due to the quadratic constraint $Q_a \preceq pR_m$. It is, however, easily seen to be equivalent to the SDP in Equation (7). We thank Jamie Sikora for pointing this out to us.

ith query sent from the user to the token, where each $\widetilde{y}_i = b_i \circ y_i$ 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. The inner summation is over all messages $\widetilde{y}$ and keys z such that the token correctly returns response r_i given both $\widetilde{y}_i$ and z.

Upper bounding p. To now upper bound p, we 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.3 of Appendix C.1. Here, we give a solution which attains $p \in O(2^{2m-0.228n})$, as claimed in Theorem 3.2 (and formally proven in Theorem C.5 of Appendix C.1). Namely, we set

$$R_{m+1} = \frac{1}{N} \sum_{\text{“successful”}\ r} |r_m\rangle\langle r_m|_{\mathcal{X}_{m+1}} \otimes \cdots \otimes |r_1\rangle\langle r_1|_{\mathcal{X}_2} \otimes I_{Y_1 \otimes \cdots \otimes Y_m} \otimes \frac{1}{2^n} I_{\mathcal{X}_1}, \tag{28}$$

where intuitively N is the total number of strings r corresponding to successful cheating, and recall n is the key size. This satisfies constraint (19) of Γ due to the identity term $I_{Y_1 \otimes \cdots \otimes Y_m}$. The renormalization factor of $(N2^n)^{-1}$ above ensures that tracing out all $\mathcal{X}_i$ registers yields $R_0 = 1$ in constraint (21) of Γ. We are thus reduced to choosing the minimum p such that constraint (18) 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.1, which says

$$\lambda_{\max}(Q_1) = \frac{2}{4^n} \left(1 + \frac{1}{\sqrt{2}}\right)^n. \tag{29}$$

Combining this bound on $\lambda_{\max}(Q_1)$ with the parameters of R_{m+1} above now yields the desired claim that $p \in O(2^{2m-0.228n})$. For (say) $m \geq n$ this bound is vacuous, and thus does not suffice to show even the trivial bound $p \leq 1$ for all m, as stated. (See Lemma C.3 for a feasible solution attaining $p \leq 1$ for all m.) However, for $m < 0.114n$ queries, the bound is fruitful, yielding the probability that a user of the token successfully cheats and thus that the simulation fails is exponentially small in the key size, n. Simplifications of the GW SDP, the derivation of its dual SDP, and a conjectured approximately optimal dual feasible solution are given in Appendix D.

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).

Note that if, on the other hand, the receiver is *not* given access to the token’s inverse operation, it is unlikely for a straightforward adaption of our no-go techniques to go through. This is because, in the most general case where the token is an arbitrary unitary U, which the receiver may apply as a black box, simulating $U^{-1} = U^\dagger$ appears difficult. For example, Theorem 3 of Quintino, Dong, Shimbo, Soeda, and Murao [Qui+19] shows that any *exact* implementation of $U^\dagger$ (even with an adaptive protocol) which (1) succeeds with probability $p > 0$ and (2) where p is independent of the choice of U, requires $k \geq d - 1$ uses of U. In our setting, d is exponential in the number of qubits, and thus so is k. Indeed, inverting arbitrary U would entail, as a special case, inverting arbitrary classical permutations, which

appears difficult. For example, Fefferman and Kimmel [FK18] use precisely this idea (i.e. an in-place permutation oracle, to which one does not have access to the inverse) to prove an oracle separation between two quantum generalizations of NP, Quantum-Classical Merlin Arthur and Quantum Merlin Arthur. We stress, however, that the works of [Qui+19; FK18] are for rather general unitaries U, whereas here we have a very specific choice of U (i.e. the token's implementation). For such a specialized U, it remains possible that a no-go theorem could still hold, even without black-box access to $U^\dagger$.

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[12] 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 $b \in \{0,1\}^m$, $O_f|y\rangle|b\rangle = |y\rangle|b \oplus f(y)\rangle$. 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 generalizes to OTMs that output s_i with probability at least $1 - \epsilon$ for small $\epsilon > 0$; for this, the Gentle Measurement Lemma [Win99] can be used to show that both bits can 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, we do 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 observed superposition queries to the token prevent an OTM from being secure. One can also ask how simple a hardware token with classical queries can be, while still allowing a secure OTM. Below, we consider such a strengthening in which the token is forced to have a bounded number of keys.

To formalize this, we 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 $f(y) = s_i$. Can 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 $1/\Delta^2$.*

[12] This formalization models O_f as a classical function f which can be queried in superposition, since the aim of this paper is to consider "easy-to-manufacture" tokens. However, our impossibility arguments in Section 4 trivially extend to the case when the token is modelled by an arbitrary unitary U_f.

We conclude that if a secret bit b_i has (say) at most polynomially many keys, then any measure-and-access OTM can be broken with at least inverse polynomial probability. The proof is given in Appendix E. In this sense, at least 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. Note that, 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.

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. We thank Kai-Min Chung and Jamie Sikora for related discussions, and David Mestel for observing that the bound of Equation (148) is not asymptotically optimal. AB acknowledges support 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. SG acknowledges support by NSF grants CCF-1526189 and CCF-1617710. HSZ acknowledges support by NSF grant CNS-1801470 and a Google Faculty Research Award.

A Universal Composition (UC) Framework

We consider simulation-based security. The Universal Composability (UC) framework was proposed by Canetti [Can01], culminating a long sequence of simulation-based security definitions (*c.f.* [GMW87; GL91; MR92; Bea91; Can00]); please see also [PW01; PS04; Can+07; 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], and then the quantum UC model by Unruh [Unr10].

A.1 Classical UC Model ([Can01])

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 $\mathsf{EXEC}[\mathcal{Z}, \mathcal{A}, \pi]$ to 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 $\mathcal{Z}$,*

$$\mathsf{EXEC}[\mathcal{Z}, \mathcal{A}, \pi] \approx \mathsf{EXEC}[\mathcal{Z}, \mathcal{S}, \pi'] \tag{30}$$

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 [Can01])**.** *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[13] 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 a 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 $\mathcal{Z}$,*

$$\mathsf{EXEC}[\mathcal{Z}, \mathcal{A}, \Pi] \approx \mathsf{EXEC}[\mathcal{Z}, \mathcal{S}, \Pi'] \tag{31}$$

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*.*

[13] 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

In order to define stand-alone security against a malicious sender (Definition B.2), in our context, we closely follow definitions given in prior work [DNS10], which we now recall. (Note that, instead of considering the *approximate* case for security, we are able to use the *exact* one.)

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 $\mathscr{A}$, $\mathscr{A}_i : \mathcal{L}(\mathcal{A}_{i-1}) \mapsto \mathcal{L}(\mathcal{A}_i), (1 \leq i \leq n)$.*
4. *An n-tuple of quantum operations $(\mathscr{B}_1, \dots \mathscr{B}_n)$ for $\mathscr{B}$, $\mathscr{B}_i : \mathcal{L}(\mathcal{B}_{i-1}) \mapsto \mathcal{L}(\mathcal{B}_i), (1 \leq i \leq n)$.*
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: $\mathcal{O}_i : \mathcal{L}(\mathcal{A}_i^{\mathcal{O}} \otimes \mathcal{B}_i^{\mathcal{O}}) \mapsto \mathcal{L}(\mathcal{A}_i^{\mathcal{O}} \otimes \mathcal{B}_i^{\mathcal{O}})$, $(1 \leq i \leq n)$.*

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 $\rho_{\text{in}} \in \mathrm{D}(\mathcal{A}_0 \otimes \mathcal{B}_0 \otimes \mathcal{R})$ where $\mathcal{R}$ is a system of dimension $\dim \mathcal{A}_0 \dim \mathcal{B}_0$, is:

$$[\mathscr{A} \circledast \mathscr{B}](\rho_{\text{in}}) = (\mathbb{1}_{\mathcal{L}(\mathcal{A}_n' \otimes \mathcal{B}_n' \otimes \mathcal{R})} \otimes \mathcal{O}_n)(\mathscr{A}_n \otimes \mathscr{B}_n \otimes \mathbb{1}_{\mathcal{R}}) \dots (\mathbb{1}_{\mathcal{L}(\mathcal{A}_1' \otimes \mathcal{B}_1' \otimes \mathcal{R})} \otimes \mathcal{O}_1)(\mathscr{A}_1 \otimes \mathscr{B}_1 \otimes \mathbb{1}_{\mathcal{R}})(\rho_{\text{in}})\,. \quad (32)$$

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.

Definition B.2. *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},*

$$\mathrm{Tr}_{\mathcal{B}_n \otimes \mathcal{R}}(\tilde{\mathscr{A}} \circledast \mathscr{B}) = \mathrm{Tr}_{\mathcal{B}_n \otimes \mathcal{R}}(\mathcal{S} \circledast \mathscr{B})\,. \quad (33)$$

We note that Definition B.2 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 B.3. *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,

$$\mathrm{Tr}_{\mathcal{B}_n \otimes \mathcal{R}}(\tilde{\mathscr{A}} \circledast \mathscr{B}) = \mathscr{A}(Tr_{\mathcal{B}_0 \otimes \mathcal{R}}(\rho_{\mathrm{in}})) = \mathcal{S}(Tr_{\mathcal{B}_0 \otimes \mathcal{R}}(\rho_{\mathrm{in}}))\,. \tag{34}$$

This concludes the proof. □

C Security Analysis for the Token

We now provide the technical result (Theorem 3.2) 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; as outlined in Section 3.4, to make it formal, in Section C.1 we model the user's interaction with the token via the Gutoski-Watrous (GW) framework for quantum games [GW07]. The resulting formal statement we desire, which immediately yields the informal claim below, is given in Theorem C.5.

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.

C.1 Security against a linear number of token queries: Primal SDP

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 2.1. As outlined in Section 3.4, we shall treat the token as the *co-strategy* and the user as the *strategy*. An overview of how all operators introduced below fit together is given in Figure 3, which may be periodically referred to as the reader progresses through this section.

Basics of our model. We proceed as follows. As depicted in Figure 1, the token (co-strategy) begins by preparing state $\rho_0 \in \mathcal{L}(\mathcal{X}_1 \otimes \mathcal{W}_0)$, and sending message $\mathcal{X}_1$ (which contains ρ_R from Equation (15) 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 $\Sigma = \left\{0, 1, \overline{0}, \overline{1}\right\}$ 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 ith query. Formally, the size of each register $\mathcal{X}_i$ for $i \geq 2$ is hence three qubits. We will deviate from Figure 1 in one respect — we assume the token also returns the response to the final query, m, 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. This will be spelled out formally below once we defined the isometries A_i for the protocol.

Before doing so, let us 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 $\widetilde{y} = b \circ y \in \{0,1\}^{n+1}$ (we henceforth write $\widetilde{y} = by$ for brevity), the first bit of which is

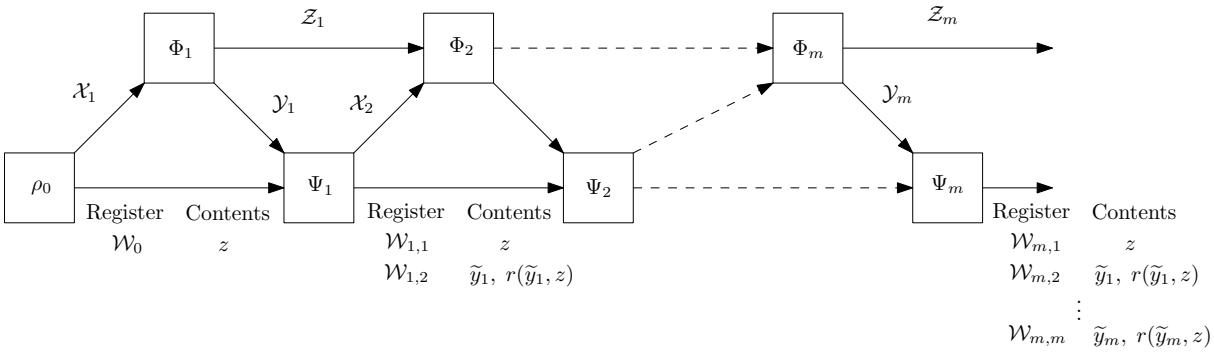

Figure 2: A reproduction of Figure 1 with additional details regarding the token's private memory contents (bottom row, horizontal arrows pointing to the right) in each round. For example, in round $k = 0$, $\mathcal{W}_0$ contains the secret key, z. In round $k = 1$, $\mathcal{W}_{1,1}$ contains z and $\mathcal{W}_{1,2}$ contains $\widetilde{y}_1$ and $r(\widetilde{y}_1, z)$. Here, $r(\widetilde{y}_k, z) \in \Sigma$ denotes whether the token accepted or rejected query string $\widetilde{y}_k$ assuming secret key z. Note the secret key z is passed along from round to round (otherwise, the token cannot correctly decide its response in a round k given query string $\widetilde{y}_k$).

the choice bit. We shall refer to $\widetilde{y}$ 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 $\widetilde{y}$ which cause the token to return response $\overline{0}$, $\overline{1}$, 0, or 1, respectively.

Defining the isometries A_k. Recall from Section 2.1 that the GW model begins by capturing the actions of a co-strategy as a sequence of linear isometries, A_k. To define these A_k, we first construct operators $\Delta_k(z) : \mathcal{Y}_k \mapsto \mathcal{X}_{k+1} \otimes \mathcal{W}_{k,k+1}$ (i.e. which map an incoming message in $\mathcal{Y}_k$ to the token to an outgoing message in $\mathcal{X}_{k+1}$ and private data to store in $\mathcal{W}_{k,k+1}$) for $k \in \{1, \ldots, m\}$ as follows:

$$\Delta_k(z) = \sum_{\widetilde{y} \in A_{\overline{0}}(z)} |\overline{0}0\rangle_{\mathcal{X}_{k+1}} |\widetilde{y}\overline{0}\rangle_{\mathcal{W}_{k,k+1}} \langle \widetilde{y}|_{\mathcal{Y}_k} + \tag{35}$$

$$\sum_{\widetilde{y} \in A_{\overline{1}}(z)} |\overline{1}0\rangle_{\mathcal{X}_{k+1}} |\widetilde{y}\overline{1}\rangle_{\mathcal{W}_{k,k+1}} \langle \widetilde{y}|_{\mathcal{Y}_k} + \tag{36}$$

$$\sum_{\widetilde{y} \in A_0(z)} |0s_0\rangle_{\mathcal{X}_{k+1}} |\widetilde{y}0\rangle_{\mathcal{W}_{k,k+1}} \langle \widetilde{y}|_{\mathcal{Y}_k} + \tag{37}$$

$$\sum_{\widetilde{y} \in A_1(z)} |1s_1\rangle_{\mathcal{X}_{k+1}} |\widetilde{y}1\rangle_{\mathcal{W}_{k,k+1}} \langle \widetilde{y}|_{\mathcal{Y}_k}. \tag{38}$$

The intuition is as follows. In round k, we model the token as (coherently) making the following classical computation: Upon input $|\widetilde{y}\rangle_{\mathcal{Y}_k}$ from the user (which consists of a choice bit b and candidate key y), the token sends its response in $\mathcal{X}_{k+1}$ 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 s, which is set to 0 by default for failed queries), and classically copies (i.e. via transversal CNOT gates) both the input $\widetilde{y}$ and response from Σ into $\mathcal{W}_k$ (the private memory of the token). Recall from Section 3.4 that coherently keeping this local copy of $\widetilde{y}$, which is never accessed again, simulates a measurement of $\mathcal{Y}_k$ in the standard basis (by the principle of deferred measurement [NC00]). The response from Σ is also locally stored in $\mathcal{W}_k$, solely for the token to be able to decide at the end of the protocol whether the user successfully extracted both secret bits. (More details on this below after the isometries A_k are defined.)

Before finally defining isometries A_k, let us further elaborate on how the token's private memory spaces $\mathcal{W}_k$ is modelled (illustrated in Figure 2).

- $\mathcal{W}_0$ contains the secret key $z \in \{0,1\}^{2n}$ of the token (i.e. the token knows what the secret key is).
- 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, i.e. the token should know the secret key in *all* rounds), and

Operator	Description
$\Delta_k(z)$	1. Sends back token's response to user's kth query $\widetilde{y}_k$, *conditioned on* key z 2. Copies all data sent above to token's private register 3. Forwards token's existing private memory contents to next round
A_k	"Bootstraps" $\Delta_k(z)$ by summing over all possible keys z Note A_0 also has special role of sending quantum key $\|\psi_z\rangle$ to user
A	The operator obtained by taking the product of all A_k
B_1	The operator A projected down onto the space of all "accepting" strategies, i.e. where the token's private memory in the last round, $\mathcal{W}_m$, contains $\|0\rangle$ and $\|1\rangle$ in some $\mathcal{W}'_i$ and $\mathcal{W}'_j$ for $i \neq j$, respectively
Q_1	The operator B_1 is reshaped into a column vector via vec() mapping, then the token's private memory in the last round, $\mathcal{W}_m$, is traced out.

Figure 3: An overview of how all operators in our instantiation of the GW framework fit together.

– $\mathcal{W}_{k,r}$ for $r \geq 2$ contains a copy in the standard basis of the user's $(r-1)$st query string (string $\widetilde{y}$), as well as the token's response from Σ for said query.

Note that an additional technical reason for storing $\widetilde{y}$ above is that it ensures $\Delta_k(z)^\dagger \Delta_k(z) = I$, so that each A_i defined shortly is an isometry. We remark that 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 stores nothing other than the secret key z in its private memory (i.e. registers $\mathcal{W}_{k,r}$ for $r \geq 2$ exist only for our security analysis, not the actual implementation of the token).

We are now ready to define isometries A_k for round k of the token's actions, where $1 < k \leq m$:

$$A_0 = \frac{1}{2^n} \sum_{z \in \{0,1\}^{2n}} |\psi_z\rangle_{\mathcal{X}_1} |z\rangle_{\mathcal{W}_{0,1}} \tag{39}$$

$$A_1 = \sum_{z \in \{0,1\}^{2n}} \Delta_1(z)_{\mathcal{Y}_1, \mathcal{X}_2, \mathcal{W}_{1,2}} \otimes |z\rangle_{\mathcal{W}_{1,1}} \langle z|_{\mathcal{W}_{0,1}} \tag{40}$$

$$A_k = \sum_{z \in \{0,1\}^{2n}} \Delta_k(z)_{\mathcal{Y}_k, \mathcal{X}_{k+1}, \mathcal{W}_{k,k+1}} \otimes |z\rangle_{\mathcal{W}_{k,1}} \langle z|_{\mathcal{W}_{k-1,1}} \bigotimes_{r=2}^{k} I_{\mathcal{W}_{k,r}, \mathcal{W}_{k-1,r}} \tag{41}$$

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 $1 \leq k \leq m$. The intuition is as follows:

- 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 in register $\mathcal{X}_1$.
- Each A_i for $1 \leq k \leq m$ consists of terms $\Delta_k(z)$ and $|z\rangle\langle z|$. The latter simply copies forward the secret key z from round $i-1$ to i from private register $\mathcal{W}_{k-1,1}$ to $\mathcal{W}_{k,1}$, ensuring the token always knows z. Recall the term $\Delta_k(z)$, defined in Equation (35), captures the token reading a message $\widetilde{y}$ from the user in $\mathcal{Y}_k$, measuring it in the standard basis (simulated by copying string $\widetilde{y}$ to a private register $\mathcal{W}_{k,k+1}$), returning an appropriate response to the user in register $\mathcal{X}_{k+1}$, and storing a copy of the kth response from Σ to the user in the private register $\mathcal{W}_{k,k+1}$.

Remark C.2. *It is in the definition of the A_i above that it is now formally clear that, although in our analysis the token stores additional data in its private register $\mathcal{W}$ (in addition to the private key, z), the token's response on incoming kth message $\widetilde{y}$ depends* solely on register $\mathcal{W}_{k,1}$*, which contains only the secret key, z. (This is most easily seen through Equation 35, where the only "bra" vector is $\langle\widetilde{y}|_{\mathcal{Y}_k}$, meaning the corresponding response $|\overline{0}0\rangle_{\mathcal{X}_{k+1}}$ depends only on $\widetilde{y}$ and the string z (since the term $\overline{0}$ depends on the summation criterion $\widetilde{y} \in A_{\overline{0}}(z)$, which depends on z). Thus, the effective interactive behavior of the token is indeed stateless, as desired.*

Combining isometries A_i to get A. Having defined isometries A_i, their product now yields operator A from Equation (3) (where we reorder the $\mathcal{X}$ and $\mathcal{W}$ registers to clarify that incoming message $\mathcal{Y}_k$ results in outgoing message $\mathcal{X}_{k+1}$):

$$A = \frac{1}{2^n} \sum_{z\in\{0,1\}^{2n}} \sum_{\widetilde{y}_1,\ldots,\widetilde{y}_m\in\{0,1\}^{n+1}} |\widetilde{y}_1 r(\widetilde{y}_1,z)\rangle_{\mathcal{W}_{m,2}} \otimes \cdots \otimes |\widetilde{y}_m r(\widetilde{y}_m,z)\rangle_{\mathcal{W}_{m,m+1}} \otimes \tag{42}$$

$$|r(\widetilde{y}_m,z) s_{r(\widetilde{y}_m,z)}\rangle_{\mathcal{X}_{m+1}} \langle\widetilde{y}_m|_{\mathcal{Y}_m} \otimes |r(\widetilde{y}_{m-1},z) s_{r(\widetilde{y}_{m-1},z)}\rangle_{\mathcal{X}_m} \langle\widetilde{y}_{m-1}|_{\mathcal{Y}_{m-1}} \otimes \cdots \otimes \tag{43}$$

$$|r(\widetilde{y}_1,z) s_{r(\widetilde{y}_1,z)}\rangle_{\mathcal{X}_2} \langle\widetilde{y}_1|_{\mathcal{Y}_1} \otimes |\psi_z\rangle_{\mathcal{X}_1} \otimes |z\rangle_{\mathcal{W}_{m,1}}, \tag{44}$$

where $r(\widetilde{y},z) \in \Sigma$ denotes whether the token accepted or rejected query string $\widetilde{y}$ assuming secret key z, and $s_{r(\widetilde{y},z)} \in \{0,1\}$ is the secret bit returned by the token corresponding to response $r(\widetilde{y},z) \in \Sigma$ (recall we set $s_{r(\widetilde{y},z)} = 0$ if $r(\widetilde{y},z) \in \{\overline{0},\overline{1}\}$).

Defining operator Q_1. In order to next define operator Q_1 from Equation (6), 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 on its private memory 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 $\mathcal{W}_{m,k}$ for $2 \le k \le m+1$, which hold the values from Σ (*i.e.*, the responses $r(\widetilde{y}_{k-1},z)$). Then, a *successful* user makes at least one correct 0-query and at least one correct 1-query (where a j-query refers to a query for choice bit j).

We define the "accepting" measurement operator Λ_1, corresponding to a successful user, as follows. Λ_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, Λ_1 projects onto set

$$T := \{t \in \Sigma^m \mid t \text{ contains at least one 0 and one 1}\}. \tag{45}$$

To use this definition of Λ_1 to write down B_1, we require further terminology. Define for any $t \in T$ and fixed key $z \in \{0,1\}^{2n}$, the set of all consistent sequences of query strings $\widetilde{y}_i \in \{0,1\}^{n+1}$ as:

$$Y_t = \left\{(\widetilde{y},z) \in \{0,1\}^{m(n+1)} \times \{0,1\}^{2n} \mid r(\widetilde{y}_i,z) = t_i \text{ for } \widetilde{y}_i \text{ the } i\text{th block of } (n+1) \text{ bits in } \widetilde{y}\right\}. \tag{46}$$

(For example, the second block of $(n+1)$ bits of $0^{n+1}1^{n+1}$ is 1^{n+1}.) In words, Y_t is the set of all strings $\widetilde{y}_1\widetilde{y}_2\cdots\widetilde{y}_m z$ such that the response of the token on query i, $r(\widetilde{y}_i,z) \in \Sigma$, is consistent with t_i. Using this, define relation $R \subseteq \Sigma^m \times \{0,1\}^{m(n+1)} \times \{0,1\}^{2n}$ such that

$$(t,\widetilde{y},z) \in R \text{ if and only if } [t \in T \text{ and } (\widetilde{y},z) \in Y_t]. \tag{47}$$

In words, a triple $(t,\widetilde{y},z) \in R$ if for a secret key z and query string $\widetilde{y}$, $t \in T \subseteq \Sigma^m$ encodes the correct set of m responses from the token (where recall T is the set of all "successful" response strings).

Recall from Equation (6) that to define Q_1, we required B_1, which in turn required Λ_1 and A. With the latter two in hand, we can now define $B_1 = (\sqrt{\Lambda_1} \otimes I)A = (\Lambda_1 \otimes I)A$ as (where recall $t_i = r(\widetilde{y}_i,z)$)

$$B_1 = \frac{1}{2^n} \sum_{(t,\widetilde{y},z)\in R} |\widetilde{y}_1 t_1\rangle_{\mathcal{W}_{m,2}} \otimes \cdots \otimes |\widetilde{y}_m t_m\rangle_{\mathcal{W}_{m,m+1}} \otimes \tag{48}$$

$$|t_m s_{t_m}\rangle_{\mathcal{X}_{m+1}} \langle\widetilde{y}_m|_{\mathcal{Y}_m} \otimes |t_{m-1} s_{t_{m-1}}\rangle_{\mathcal{X}_m} \langle\widetilde{y}_{m-1}|_{\mathcal{Y}_{m-1}} \otimes \cdots \otimes |t_1 s_{t_1}\rangle_{\mathcal{X}_2} \langle\widetilde{y}_1|_{\mathcal{Y}_1} \otimes \tag{49}$$

$$|\psi_z\rangle_{\mathcal{X}_1} \otimes |z\rangle_{\mathcal{W}_{m,1}}. \tag{50}$$

By definition of the vec mapping (Section 2.1),

$$\begin{aligned}\mathrm{vec}(B_1) =& \frac{1}{2^n} \sum_{(t,\widetilde{y},z)\in R} |\widetilde{y}_1 t_1\rangle_{\mathcal{W}_{m,2}} \otimes \cdots \otimes |\widetilde{y}_m t_m\rangle_{\mathcal{W}_{m,m+1}} \otimes && (51)\\ & |t_m s_{t_m}\rangle_{\mathcal{X}_{m+1}} |\widetilde{y}_m\rangle_{\mathcal{Y}_m} \otimes |t_{m-1} s_{t_{m-1}}\rangle_{\mathcal{X}_m} |\widetilde{y}_{m-1}\rangle_{\mathcal{Y}_{m-1}} \otimes \cdots \otimes |t_1 s_{t_1}\rangle_{\mathcal{X}_2} |\widetilde{y}_1\rangle_{\mathcal{Y}_1} \otimes && (52)\\ & |z\rangle_{\mathcal{W}_{m,1}} \otimes |\psi_z\rangle_{\mathcal{X}_1}. && (53)\end{aligned}$$

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

$$\begin{aligned}Q_1 =& \frac{1}{2^{2n}} \sum_{(t,\widetilde{y},z)\in R} |t_m s_{t_m}\rangle\langle t_m s_{t_m}|_{\mathcal{X}_{m+1}} \otimes |\widetilde{y}_m\rangle\langle\widetilde{y}_m|_{\mathcal{Y}_m} \otimes \cdots \otimes && (54)\\ & |t_1 s_{t_1}\rangle\langle t_1 s_{t_1}|_{\mathcal{X}_2} \otimes |\widetilde{y}_1\rangle\langle\widetilde{y}_1|_{\mathcal{Y}_1} \otimes |\psi_z\rangle\langle\psi_z|_{\mathcal{X}_1}. && (55)\end{aligned}$$

Note that we have crucially used the fact that queries to the token are *classical strings*. Namely, since the token implicitly measures its input in the standard basis (modelled by copying each string $\widetilde{y}_i$ to register $\mathcal{W}_i$), the partial trace over $\mathcal{W}_m$ annihilates all cross terms in $\mathrm{vec}(B_1)\,\mathrm{vec}(B_1)^*$. Thus, Q_1 is conveniently simplified to a *mixture* over $(t, \widetilde{y}, z) \in R$, which is further block-diagonal with respect to all registers other than $\mathcal{X}_1$.

For convenience, we permute subsystems to rewrite:

$$\begin{aligned}Q_1 =& \frac{1}{4^n} \sum_{t\in T} |t_m s_{t_m}\rangle\langle t_m s_{t_m}|_{\mathcal{X}_{m+1}} \otimes \cdots \otimes |t_1 s_{t_1}\rangle\langle t_1 s_{t_1}|_{\mathcal{X}_2} \otimes && (56)\\ & \left(\sum_{(\widetilde{y},z)\in Y_t} |\widetilde{y}_m\rangle\langle\widetilde{y}_m|_{\mathcal{Y}_m} \otimes \cdots \otimes |\widetilde{y}_1\rangle\langle\widetilde{y}_1|_{\mathcal{Y}_1} \otimes |\psi_z\rangle\langle\psi_z|_{\mathcal{X}_1} \right). && (57)\end{aligned}$$

The SDP. Having set up all required operators for the GW framework, Equation (7) of Section 2.1 now yields the optimal probability with which a cheating user can succeed; we reproduce Equation (7) below for convenience. Note the subsystem ordering of Q_1 below is not that of Equation (57), 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.

$$\begin{aligned}\text{min:} \quad & p && && (58)\\ \text{subject to:} \quad & Q_1 \preceq p R_{m+1} && && (59)\\ & R_k = P_k \otimes I_{\mathcal{Y}_k} && \text{for } 1 \le k \le m+1 && (60)\\ & \mathrm{Tr}_{\mathcal{X}_k}(P_k) = R_{k-1} && \text{for } 1 \le k \le m+1 && (61)\\ & R_0 = 1 && && (62)\\ & R_k \in \mathrm{Pos}(\mathcal{Y}_{1,\ldots,k} \otimes \mathcal{X}_{1,\ldots,k}) && \text{for } 1 \le k \le m+1 && (63)\\ & P_k \in \mathrm{Pos}(\mathcal{Y}_{1,\ldots,k-1} \otimes \mathcal{X}_{1,\ldots,k}) && \text{for } 1 \le k \le m+1 && (64)\\ & p \in [0,1] && && (65)\end{aligned}$$

In the analysis below, we shall sometimes analyze the optimization above, which we shall denote Γ'. However, note that technically it is not yet an SDP due to the quadratic constraint $Q_1 \preceq p R_{m+1}$. It is, however, easily seen to be equivalent to the following bona fide SDP Γ:

$$\begin{aligned}\text{min:} \quad & p && && (66)\\ \text{subject to:} \quad & Q_1 \preceq R_{m+1} && && (67)\\ & R_k = P_k \otimes I_{\mathcal{Y}_k} && \text{for } 1 \le k \le m+1 && (68)\\ & \mathrm{Tr}_{\mathcal{X}_k}(P_k) = R_{k-1} && \text{for } 1 \le k \le m+1 && (69)\\ & R_0 = p && && (70)\\ & R_k \in \mathrm{Pos}(\mathcal{Y}_{1,\ldots,k} \otimes \mathcal{X}_{1,\ldots,k}) && \text{for } 1 \le k \le m+1 && (71)\\ & P_k \in \mathrm{Pos}(\mathcal{Y}_{1,\ldots,k-1} \otimes \mathcal{X}_{1,\ldots,k}) && \text{for } 1 \le k \le m+1 && (72)\end{aligned}$$

Above and henceforth, we use terminology $\mathcal{T}_{1\cdots k}$ to denote the space $\mathcal{T}_1 \otimes \cdots \otimes \mathcal{T}_k$.

Warmup: A "trivial" solution. We mentioned in Section 3.4 that obtaining a solution to Γ which obtains the trivial bound $p \leq 1$ is not trivial. (Sometimes with SDPs, a scaled identity operator gives a feasible solution obtaining the desired trivial bound on the objective value; this unfortunately does not work here.) Let us hence warm up by demonstrating a solution attaining the trivial bound $p \leq 1$.

Lemma C.3. *The SDP Γ has a feasible solution with $p = 1$.*

Proof. Recall from Equation (57) that

$$Q_1 = \frac{1}{4^n} \sum_{(t,\widetilde{y},z)\in R} |t,s\rangle\langle t,s|_{\mathcal{X}_{m+1\cdots 2}} \otimes \left(|\widetilde{y}_m\rangle\langle\widetilde{y}_m|_{\mathcal{Y}_m} \otimes \cdots \otimes |\widetilde{y}_1\rangle\langle\widetilde{y}_1|_{\mathcal{Y}_1}\right) \otimes |\psi_z\rangle\langle\psi_z|_{\mathcal{X}_1}, \tag{73}$$

where $t \in T \subseteq \Sigma^m$ and $s \in \{0,1\}^m$ are the resulting query responses and secret bits, respectively. (Recall from Equation (57) that, formally, we should write s_t, as each s depends on t; to save clutter and space below, however, we drop the subscript.) Observe that any fixed $\widetilde{y} \in \{0,1\}^{m(n+1)}$ and $z \in \{0,1\}^{2n}$ determine a *unique* query response string $t \in \Sigma^m$ (which may or may not be in T); denote this as $t(\widetilde{y}, z)$. Therefore,

$$Q_1 = \frac{1}{4^n} \sum_{\substack{\widetilde{y},z \\ \text{s.t. } t(\widetilde{y},z)\in T}} |t(\widetilde{y},z),s\rangle\langle t(\widetilde{y},z),s|_{\mathcal{X}_{m+1\cdots 2}} \otimes \left(|\widetilde{y}_m\rangle\langle\widetilde{y}_m|_{\mathcal{Y}_m} \otimes \cdots \otimes |\widetilde{y}_1\rangle\langle\widetilde{y}_1|_{\mathcal{Y}_1}\right) \otimes |\psi_z\rangle\langle\psi_z|_{\mathcal{X}_1}, \tag{74}$$

for $T \subseteq \Sigma^m$ defined as in Equation (45). Let us drop the constraint that $t(\widetilde{y}, z) \in T$, *i.e.* choose

$$R_{m+1} = \frac{1}{4^n} \sum_{\widetilde{y},z} |t(\widetilde{y},z),s\rangle\langle t(\widetilde{y},z),s|_{\mathcal{X}_{m+1\cdots 2}} \otimes \left(|\widetilde{y}_m\rangle\langle\widetilde{y}_m|_{\mathcal{Y}_m} \otimes \cdots \otimes |\widetilde{y}_1\rangle\langle\widetilde{y}_1|_{\mathcal{Y}_1}\right) \otimes |\psi_z\rangle\langle\psi_z|_{\mathcal{X}_1}. \tag{75}$$

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 constraints, clearly R_{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:

$$\frac{1}{4^n} \sum_{\widetilde{y}_m,\ldots,\widetilde{y}_1} \sum_z |t_{m-1}(\widetilde{y},z)s_{m-1}\rangle\langle t_{m-1}(\widetilde{y},z)s_{m-1}|_{\mathcal{X}_{m\cdots 2}} \otimes \left(|\widetilde{y}_m\rangle\langle\widetilde{y}_m|_{\mathcal{Y}_m} \otimes \cdots \otimes |\widetilde{y}_1\rangle\langle\widetilde{y}_1|_{\mathcal{Y}_1}\right) \otimes |\psi_z\rangle\langle\psi_z|_{\mathcal{X}_1}, \tag{76}$$

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

$$\mathrm{Tr}_{\mathcal{X}_{m+1}}(R_{m+1}) = \frac{1}{4^n} \sum_{\widetilde{y}_{m-1},\ldots,\widetilde{y}_1} \sum_z |t_{m-1}(\widetilde{y},z)s_{m-1}\rangle\langle t_{m-1}(\widetilde{y},z)s_{m-1}|_{\mathcal{X}_{m\cdots 2}} \otimes \tag{77}$$

$$\left(I_{\mathcal{Y}_m} \otimes |\widetilde{y}_{m-1}\rangle\langle\widetilde{y}_{m-1}|_{\mathcal{Y}_{m-1}} \otimes \cdots \otimes |\widetilde{y}_1\rangle\langle\widetilde{y}_1|_{\mathcal{Y}_1}\right) \otimes |\psi_z\rangle\langle\psi_z|_{\mathcal{X}_1}. \tag{78}$$

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

$$\frac{1}{4^n} I_{\mathcal{Y}_{m+1\cdots 1}} \otimes \sum_z |\psi_z\rangle\langle\psi_z|_{\mathcal{X}_1}. \tag{79}$$

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, which we state and prove first.

Lemma C.4. *For Q_1 in Equation (57), $\lambda_{\max}(Q_1) = \frac{2}{4^n}\left(1+\frac{1}{\sqrt{2}}\right)^n$.*

Proof. The factor of 4^{-n} in the claimed value for $\lambda_{\max}(Q_1)$ comes from the 4^{-n} appearing in Equation (57); 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 $t \in T$ and $\widetilde{y} \in \{0,1\}^{m(n+1)}$ defines the $(t,\widetilde{y})$-block of Q_1. (Formally, the $(t,\widetilde{y})$-block of Q_1 is given by $\Pi_{t,\widetilde{y}} Q_1 \Pi_{t,\widetilde{y}}$, where $\Pi_{t,\widetilde{y}} = |t\rangle\langle t|_{\mathcal{X}_{m+1\cdots 2}} \otimes |\widetilde{y}\rangle\langle\widetilde{y}|_{\mathcal{Y}}$.)

Lower bound. We first show lower bound $\lambda_{\max}(Q_1) \geq \frac{2}{4^n}(1+\frac{1}{\sqrt{2}})^n$. To do so, we demonstrate an explicit $t,\widetilde{y}$ such that the $(t,\widetilde{y})$-block has eigenvalue $\frac{2}{4^n}(1+\frac{1}{\sqrt{2}})^n$. Set $t = 0^{m-1}1$ (note $t \in T$) and $\widetilde{y} = \widetilde{y}_1 \ldots \widetilde{y}_m$ for $\widetilde{y}_1 = \widetilde{y}_2 = \cdots = \widetilde{y}_{m-1}$ and $\widetilde{y}_{m-1} \neq \widetilde{y}_m$ (note $\widetilde{y}_i \in \{0,1\}^{n+1}$), where the first bit of each of $\widetilde{y}_1,\ldots,\widetilde{y}_{m-1}$ is 0, and the first bit of $\widetilde{y}_m$ 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 t and $\widetilde{y}$, how many $|\psi_z\rangle \in (\mathbb{C}^2)^{\otimes n}$ exist such that $(t,\widetilde{y},z) \in R$?

To answer this, observe that the token enforces the following set of rules. Fix any $i \in \{1,\ldots,m\}$, and let $|\widetilde{y}_i(j)\rangle$ and $|\psi_z(j)\rangle$ denote the jth qubits of $\widetilde{y}_i$ 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 $t_i = 0$, then $\forall j \in \{1,\ldots,n\}$, either $|\psi_z(j)\rangle = |\widetilde{y}_i(j)\rangle$ or $|\psi_z(j)\rangle \in \{|+\rangle,|-\rangle\}$.
2. If $t_i = 1$, then $\forall j \in \{1,\ldots,n\}$, either $|\psi_z(j)\rangle = H|\widetilde{y}_i(j)\rangle$ or $|\psi_z(j)\rangle \in \{|0\rangle,|1\rangle\}$.
3. If $t_i = \overline{0}$, then $\exists j \in \{1,\ldots,n\}$ such that $|\psi_z(j)\rangle = |\overline{\widetilde{y}_i(j)}\rangle$.
4. If $t_i = \overline{1}$, then $\exists j \in \{1,\ldots,n\}$ such that $|\psi_z(j)\rangle = H|\overline{\widetilde{y}_i(j)}\rangle$.

Recall now that we set $t_1 = 0$ and $t_m = 1$, *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, $|\psi_z(k)\rangle \in \{|\widetilde{y}_1(k)\rangle, H|\widetilde{y}_m(k)\rangle\}$. Moreover, since $\widetilde{y}_1 = \widetilde{y}_2 = \cdots = \widetilde{y}_{m-1}$, it follows that for all k, both assignments for $|\psi_z(k)\rangle$ are consistent with t. We conclude that the $(t,\widetilde{y})$-block of Q_1 has the following operator in register $\mathcal{X}_1$:

$$\sigma = \bigotimes_{k=1}^{n} \left(|\widetilde{y}_1(k)\rangle\langle\widetilde{y}_1(k)| + H|\widetilde{y}_m(k)\rangle\langle\widetilde{y}_m(k)|H\right). \tag{80}$$

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.

Upper bound. 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 $(t,\widetilde{y})$-blocks. For any $t \in T$, there exist indices $i \neq j$ such that $\widetilde{y}_i$ and $\widetilde{y}_j$ are a successful 0- and 1-query, respectively. Without loss of generality, assume $i = 1$ and $j = m$. Then, as in the previous case, rules 1 and 2 imply that:

$$\forall k \in \{1,\ldots,n\}, \quad |\psi_z(k)\rangle \in \{|\widetilde{y}_1(k)\rangle, H|\widetilde{y}_m(k)\rangle\}. \tag{81}$$

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

- (Case 1: $t_i = 0$) 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 $\widetilde{y}_1$ and $\widetilde{y}_i$ disagree, the secret key must have encoded bit k in the X-basis. In other words, $|\psi_z(k)\rangle = H|\widetilde{y}_m(k)\rangle$ in Equation (81) (*i.e.* one of the two possibilities is eliminated). (If $\widetilde{y}_1 = \widetilde{y}_i$, on the other hand, no such additional constraint exists.)

- (Case 2: $t_i = \overline{0}$) In this case, query i is an unsuccessful 0-query. By Rule 3, there exists a bit k on which $\widetilde{y}_1$ and $\widetilde{y}_k$ disagree, and whose corresponding secret key bit was encoded in the Z basis. In other words, $|\psi_z(k)\rangle = |\widetilde{y}_1(k)\rangle$ in Equation (81) (*i.e.*, one of the two possibilities is eliminated).

The analysis for $\widetilde{y}_i$ being a 1-query is analogous. We conclude that for any $(t, \widetilde{y})$-block of Q_1, the operator in register $\mathcal{X}_1$ is of the form of σ from Equation (80), except that the some of the indices k may contain an operator consisting of only 1 summand (e.g. $|\widetilde{y}_1(k)\rangle\langle\widetilde{y}_1(k)|$ instead of $|\widetilde{y}_1(k)\rangle\langle\widetilde{y}_1(k)| + H|\widetilde{y}_m(k)\rangle\langle\widetilde{y}_m(k)|H$). Since the omitted summands are all positive semidefinite, however, we conclude the eigenvalue on any $(t, \widetilde{y})$-block is at most the eigenvalue of σ from Equation (80), *i.e.*, at most $\lambda_{\max}(Q_1) \leq \frac{2}{4^n}(1 + \frac{1}{\sqrt{2}})^n$, as claimed. □

We can now prove the main result of this section.

Theorem C.5. *The SDP Γ has a feasible solution with $p \in O(2^{2m-0.228n})$.*

Proof. As Q_1 in Equation (57) is block-diagonal in registers $\mathcal{X}_2, \ldots, \mathcal{X}_{m+1}$, consider solution (for T from Equation (45))

$$R_{m+1} = \frac{1}{|T|} \sum_{t \in T} |t_m s_{t_m}\rangle\langle t_m s_{t_m}|_{\mathcal{X}_{m+1}} \otimes \cdots \otimes |t_1 s_{t_1}\rangle\langle t_1 s_{t_1}|_{\mathcal{X}_2} \otimes I_{\mathcal{Y}_{1,\ldots,m}} \otimes \frac{1}{2^n} I_{\mathcal{X}_1}. \tag{82}$$

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

$$|\Sigma^m| = 4^m \tag{83}$$
$$\{t \in \Sigma^m \mid t \text{ does not contain a } 0\} = 3^m \tag{84}$$
$$\{t \in \Sigma^m \mid t \text{ does not contain a } 1\} = 3^m \tag{85}$$
$$\{t \in \Sigma^m \mid t \text{ does not contain a } 0 \text{ or a } 1\} = 2^m. \tag{86}$$

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

In order for R_{m+1} to be feasible, we must pick p such that $Q_1 \preceq pR_{m+1}$. 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 $(|T|\, 2^n)^{-1}$. Thus, we must choose p such that

$$\lambda_{\max}(Q_1) \leq \frac{p}{|T|\, 2^n}, \tag{87}$$

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

$$p \geq \frac{|T|}{2^n} \lambda_{\max}\left(4^n Q_1\right). \tag{88}$$

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

$$p = \frac{|T|}{2^{n-1}}\left(1 + \frac{1}{\sqrt{2}}\right)^n \approx |T| \cdot 2^{(-0.228)n+1}, \tag{89}$$

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

D Simplifying the Gutoski-Watrous SDP and its dual

D.1 Streamlining the primal and dual

We now simplify the general SDP (Equation (66) from the Gutoski-Watrous (GW) framework (note this simplification is independent of our particular application of the framework for OTMs, i.e. independent

of Q_1), and derive its dual SDP. For convenience, we begin by reproducing the following definitions, including the SDP Γ of Equation (66).

$$\begin{aligned}
\text{min:} \quad & p && && (90)\\
\text{subject to:} \quad & Q_1 \preceq R_{m+1} && && (91)\\
& R_k = P_k \otimes I_{\mathcal{Y}_k} && \text{for } 1 \leq k \leq m+1 && (92)\\
& \mathrm{Tr}_{\mathcal{X}_k}(P_k) = R_{k-1} && \text{for } 1 \leq k \leq m+1 && (93)\\
& R_0 = p && && (94)\\
& R_k \in \mathrm{Pos}(\mathcal{Y}_{1,\ldots,k} \otimes \mathcal{X}_{1,\ldots,k}) && \text{for } 1 \leq k \leq m+1 && (95)\\
& P_k \in \mathrm{Pos}(\mathcal{Y}_{1,\ldots,k-1} \otimes \mathcal{X}_{1,\ldots,k}) && \text{for } 1 \leq k \leq m+1 && (96)
\end{aligned}$$

$$(t, \widetilde{y}, z) \in R \text{ if and only if } [t \in T \text{ and } (\widetilde{y}, z) \in Y_t] \tag{97}$$

$$Q_1 = \frac{1}{4^n} \sum_{t \in T} |t_m s_{t_m}\rangle\langle t_m s_{t_m}|_{\mathcal{X}_{m+1}} \otimes \cdots \otimes |t_1 s_{t_1}\rangle\langle t_1 s_{t_1}|_{\mathcal{X}_2} \otimes \tag{98}$$

$$\left(\sum_{(\widetilde{y},z) \in Y_t} |\widetilde{y}_m\rangle\langle\widetilde{y}_m|_{\mathcal{Y}_m} \otimes \cdots \otimes |\widetilde{y}_1\rangle\langle\widetilde{y}_1|_{\mathcal{Y}_1} \otimes |\psi_z\rangle\langle\psi_z|_{\mathcal{X}_1} \right). \tag{99}$$

Guiding example: $m = 3$. We explicitly run through the construction for the first non-trivial case, $m = 3$ queries. The construction then generalizes straightforwardly to all $m \geq 2$. To begin, using the fact that $R_4 = P_4$ (since $\mathcal{Y}_4 = \mathbb{C}$, due to the fact that we assumed message $m+1$ from the user to the token is empty), Γ can be written:

$$\begin{aligned}
\text{min:} \quad & \mathrm{Tr}(P_1) && (100)\\
\text{subject to:} \quad & Q_1 - P_4 \preceq 0 && (101)\\
& -P_3 \otimes I_{\mathcal{Y}_3} + \mathrm{Tr}_{\mathcal{X}_4}(P_4) \preceq 0 && (102)\\
& -P_2 \otimes I_{\mathcal{Y}_2} + \mathrm{Tr}_{\mathcal{X}_3}(P_3) \preceq 0 && (103)\\
& -P_1 \otimes I_{\mathcal{Y}_1} + \mathrm{Tr}_{\mathcal{X}_2}(P_2) \preceq 0 && (104)
\end{aligned}$$

Above, we relaxed the equalities to inequalities[14], which intuitively makes it easier to guess feasible solutions to Γ. We also omitted the positive semidefinite constraints on all P_i, since[15] $P_4 \succeq Q_1 \succeq 0$ implies $P_i \succeq 0$ for all i. We now follow the standard Lagrange approach for deriving the dual SDP (see, e.g. [BV04]). Labelling equations (101),(102),(103),(104) with dual variables $Y_1, \ldots, Y_4$, respectively, the primal variables in the Lagrange dual function can be isolated as follows:

Primal variable	Factor
P_4	$-Y_1 + Y_2 \otimes I_{\mathcal{X}_4}$
P_3	$-\mathrm{Tr}_{\mathcal{Y}_3}(Y_2) + Y_3 \otimes I_{\mathcal{X}_3}$
P_2	$-\mathrm{Tr}_{\mathcal{Y}_2}(Y_3) + Y_4 \otimes I_{\mathcal{X}_2}$
P_1	$I_{\mathcal{X}_1} - \mathrm{Tr}_{\mathcal{Y}_1}(Y_4)$

[14] This is without loss of generality, as we briefly justify. Clearly, any feasible solution for equality constraints is also feasible for inequality constraints. For the converse direction, suppose a feasible solution for the inequality constraints satisfies $P_i \otimes I_{\mathcal{Y}_{i+1}} - \mathrm{Tr}_{\mathcal{X}_{i+1}}(P_{i+1}) = \Lambda_i \succeq 0$ for non-zero Λ_i; pick the smallest such i satisfying this condition. Then, redefining $P'_{i+1} := P_{i+1} + |\phi\rangle\langle\phi|_{\mathcal{X}_{i+1}} \otimes \Lambda_i \succeq 0$ for arbitrary unit vector $|\phi\rangle$ satisfies $P_i \otimes I_{\mathcal{Y}_{i+1}} - \mathrm{Tr}_{\mathcal{X}_{i+1}}(P'_{i+1}) = 0$, as desired. Note we can recurse this trick now from constraint i to $i+1$, since if $P_{i+1} \otimes I_{\mathcal{Y}_{i+2}} - \mathrm{Tr}_{\mathcal{X}_{i+2}}(P_{i+2}) \succeq 0$, then $P'_{i+1} \otimes I_{\mathcal{Y}_{i+2}} - \mathrm{Tr}_{\mathcal{X}_{i+2}}(P_{i+2}) \succeq 0$ (similarly for constraint $Q_1 \preceq P_{m+1}$). Thus, we obtain a new feasible solution for which all inequality constraints (except $Q_1 \preceq P_{m+1}$) hold with equality. Moreover, this process does not alter the assignment for P_1 (i.e. we never define P'_1); thus the objective function value remains unchanged.

[15] In our particular setting, it is clear that $Q_1 \succeq 0$. However, more generally in the GW framework, the operators $\{Q_a\}$ defining a measuring co-strategy all satisfy $Q_a \succeq 0$.

For clarity and as an example, this says the term $P_4(-Y_1 + Y_2 \otimes I_{\mathcal{X}_4})$ appears in the dual function. This yields dual SDP:

$$\begin{aligned}
\text{max:} \quad & \operatorname{Tr}(Y_1 Q_1) && (105)\\
\text{subject to:} \quad & -Y_1 + Y_2 \otimes I_{\mathcal{X}_4} = 0 && (106)\\
& -\operatorname{Tr}_{\mathcal{Y}_3}(Y_2) + Y_3 \otimes I_{\mathcal{X}_3} = 0 && (107)\\
& -\operatorname{Tr}_{\mathcal{Y}_2}(Y_3) + Y_4 \otimes I_{\mathcal{X}_2} = 0 && (108)\\
& I_{\mathcal{X}_1} - \operatorname{Tr}_{\mathcal{Y}_1}(Y_4) = 0 && (109)\\
& Y_1, Y_2, Y_3, Y_4 \succeq 0 && (110)
\end{aligned}$$

Now we make the following simplifications: (1) Replace Y_1 with $Y_2 \otimes I_{\mathcal{X}_4}$ (follows from Equation (106)), (2) drop the constraints $Y_3, Y_4 \succeq 0$ (since they are implied by $Y_2 \succeq 0$), and (3) relax the equalities to inequalities (which follows similar to the argument for the primal, except here we also require that we are maximizing with respect to Y_2 below). Hence, we obtain:

$$\begin{aligned}
\text{max:} \quad & \operatorname{Tr}(Y_2 \operatorname{Tr}_{\mathcal{X}_4}(Q_1)) && (111)\\
\text{subject to:} \quad & -\operatorname{Tr}_{\mathcal{Y}_3}(Y_2) + Y_3 \otimes I_{\mathcal{X}_3} \succeq 0 && (112)\\
& -\operatorname{Tr}_{\mathcal{Y}_2}(Y_3) + Y_4 \otimes I_{\mathcal{X}_2} \succeq 0 && (113)\\
& I_{\mathcal{X}_1} - \operatorname{Tr}_{\mathcal{Y}_1}(Y_4) \succeq 0 && (114)\\
& Y_2 \succeq 0 && (115)
\end{aligned}$$

Note the $Y_2 \succeq 0$ *cannot* be removed. Intuitively, this is because the constraint $I_{\mathcal{X}_1} - \operatorname{Tr}_{\mathcal{Y}_1}(Y_4) \succeq 0$ alone does not imply $Y_4 \succeq 0$. Rather, it is the constraint $Y_2 \succeq 0$ which forces $Y_4 \succeq 0$ here. Indeed, a sanity check in CVX for Matlab reveals removing the $Y_2 \succeq 0$ incorrectly yields an unbounded SDP.

We now repeat the process by taking the dual of the dual to arrive at a simplified primal as follows. (Note that the inequalities above now go in the other direction, since we are starting from the dual SDP.) Labelling the constraints above $R_1, \ldots, R_4$, we have factor table:

Dual variable	Factor
Y_2	$\operatorname{Tr}_{\mathcal{X}_4}(Q_1) - R_1 \otimes I_{\mathcal{Y}_3} + R_4$
Y_3	$\operatorname{Tr}_{\mathcal{X}_3}(R_1) - R_2 \otimes I_{\mathcal{Y}_2}$
Y_4	$\operatorname{Tr}_{\mathcal{X}_2}(R_2) - R_3 \otimes I_{\mathcal{Y}_1}$

This yields primal SDP (after omitting the redundant constraints $R_1, R_2, R_3, R_4 \succeq 0$):

$$\begin{aligned}
\text{min:} \quad & \operatorname{Tr}(R_3) && (116)\\
\text{subject to:} \quad & \operatorname{Tr}_{\mathcal{X}_4}(Q_1) - R_1 \otimes I_{\mathcal{Y}_3} \preceq 0 && (117)\\
& \operatorname{Tr}_{\mathcal{X}_3}(R_1) - R_2 \otimes I_{\mathcal{Y}_2} \preceq 0 && (118)\\
& \operatorname{Tr}_{\mathcal{X}_2}(R_2) - R_3 \otimes I_{\mathcal{Y}_1} \preceq 0 && (119)
\end{aligned}$$

Taking the dual of the primal now yields the previous dual; so it seems we are done. Relabelling variables for the primal and dual, we obtain the final $m = 3$ primal and dual SDPs, respectively:

$$\begin{aligned}
\text{min:} \quad & \operatorname{Tr}(P_1)\\
\text{subject to:} \quad & \operatorname{Tr}_{\mathcal{X}_4}(Q_1) - P_3 \otimes I_{\mathcal{Y}_3} \preceq 0\\
& \operatorname{Tr}_{\mathcal{X}_3}(P_3) - P_2 \otimes I_{\mathcal{Y}_2} \preceq 0\\
& \operatorname{Tr}_{\mathcal{X}_2}(P_2) - P_1 \otimes I_{\mathcal{Y}_1} \preceq 0
\end{aligned}
\qquad
\begin{aligned}
\text{max:} \quad & \langle Y_1, \operatorname{Tr}_{\mathcal{X}_4}(Q_1) \rangle\\
\text{subject to:} \quad & -\operatorname{Tr}_{\mathcal{Y}_3}(Y_1) + Y_2 \otimes I_{\mathcal{X}_3} \succeq 0\\
& -\operatorname{Tr}_{\mathcal{Y}_2}(Y_2) + Y_3 \otimes I_{\mathcal{X}_2} \succeq 0\\
& -\operatorname{Tr}_{\mathcal{Y}_1}(Y_3) + I_{\mathcal{X}_1} \succeq 0\\
& Y_1 \succeq 0
\end{aligned}$$

General case. The derivation above straightforwardly extends to the case of arbitrary $m \geq 2$, yielding primal and dual SDPs:

Primal SDP

$$\begin{aligned}
\text{min:} \quad & \mathrm{Tr}(P_1) && && (120)\\
\text{s.t.} \quad & \mathrm{Tr}_{\mathcal{X}_{m+1}}(Q_1) - P_m \otimes I_{\mathcal{Y}_m} \preceq 0 && && (121)\\
& \mathrm{Tr}_{\mathcal{X}_{i+1}}(P_{i+1}) - P_i \otimes I_{\mathcal{Y}_i} \preceq 0 && \forall i \in \{1, \ldots, m-1\} && (122)\\
& && && (123)
\end{aligned}$$

Dual SDP

$$\begin{aligned}
\text{max:} \quad & \langle Y_1, \mathrm{Tr}_{\mathcal{X}_{m+1}}(Q_1) \rangle && && (124)\\
\text{s.t.} \quad & -\mathrm{Tr}_{\mathcal{Y}_{m-i+1}}(Y_i) + Y_{i+1} \otimes I_{\mathcal{X}_{m-i+1}} \succeq 0 && \forall i \in \{1, \ldots, m\} && (125)\\
& Y_1 \succeq 0 && && (126)
\end{aligned}$$

where note for uniformity in stating the dual constraints, we define $Y_{m+1} := 1$ in the dual SDP.

D.2 An approximately optimal dual solution?

We now give a simple feasible solution $Y := \{Y_i\}$ to the dual SDP, whose objective function value appears to scale roughly as one might expect, *if* security were to hold for our OTM construction. While we can explicitly prove Y is *not* dual optimal (thus, it only yields a lower bound on the best cheating probability), we conjecture it is roughly optimal up to multiplicative factors (stated precisely in Conjecture D.2), which would in turn imply security against subexponentially many queries to the token, as desired.

A candidate dual solution. Recall that each $\mathcal{Y}_i$ register encodes a message from the receiver to the token, consisting of $n+1$ qubits. Let $d := 2^{n+1}$ denote the dimension of this space. Define solution $Y := \{Y_1, \ldots, Y_m\}$ via:

$$Y_i = \frac{1}{d^{m-i+1}} I_{\mathcal{Y}_{1\cdots m}, \mathcal{X}_{1\cdots m}}. \tag{127}$$

Note that $Y_1 \succeq 0$ trivially, and that Equation (125) holds with equality for all $i \in \{1, \ldots, m\}$. Thus, Y is a dual feasible solution. Moreover, it obtains objective function value

$$\beta := \frac{\mathrm{Tr}(Q_1)}{d^m} = \frac{|R|}{4^n d^m}, \tag{128}$$

where recall we defined relation R in Equation (47) via

$$(t, \widetilde{y}, z) \in R \text{ if and only if } [t \in T \text{ and } (\widetilde{y}, z) \in Y_t]. \tag{129}$$

The cardinality of R. To analyze β, we require an expression for $|R|$, given as follows.

Lemma D.1.

$$|R| = \left(2^{m(n+1)+n}\right) \sum_{\alpha=0}^{n} \binom{n}{\alpha} \left[1 - \left(1 - \frac{1}{2^{\alpha+1}}\right)^m - \left(1 - \frac{1}{2^{n-\alpha+1}}\right)^m + \left(1 - \frac{1}{2^{\alpha+1}} - \frac{1}{2^{n-\alpha+1}}\right)^m\right] \tag{130}$$

Proof. Recall again from Equation (47) that R is defined via

$$(t, \widetilde{y}, z) \in R \text{ if and only if } [t \in T \text{ and } (\widetilde{y}, z) \in Y_t], \tag{131}$$

where T is the set of successful query responses. For any quantum key $|\psi_z\rangle$, let α denote the number of qubits in $|\psi_z\rangle$ which are encoded in the Z basis. Note that fixing α partitions R into $n+1$ sets; let R_α denote the set in this partition corresponding to α Z-bits. We analyze each R_α independently first.

Computing $|R_\alpha|$ for fixed α. Fix any $0 \le \alpha \le n$, and any secret key $z \in \{0,1\}^{2n}$ with precisely α bits in the Z-basis; denote the resulting subset of R_α by $R_{\alpha,z}$. We now ask: Conditioned on secret key z and token response $t \in \Sigma = \{0, 1, \overline{0}, \overline{1}\}$, how many query strings $\widetilde{y}$ are consistent with t?

- Case 1: $t = 0$. Since we have to get precisely all α bits correctly (i.e. those in the Z basis), and we can get up to $n - \alpha$ bits in the X basis incorrect, we have $2^{n-\alpha}$ choices for $\widetilde{y}$. (Note that the choice bit for $\widetilde{y}$ is forced to be 0 since $t = 0$.)
- Case 2: $t = 1$. This is analogous to $t = 0$, except now we can get the α Z-bits incorrect and the X-bits must be correct. Thus, there are 2^α strings $\widetilde{y}$.
- Case 3: $t = \overline{0}$. Since the X bits can be anything, and there is precisely one correct setting to the Z basis bits, we have $2^{n-\alpha}(2^\alpha - 1) = 2^n - 2^{n-\alpha}$ choices for $\widetilde{y}$.
- Case 4: $t = \overline{1}$. Analogous to the $t = \overline{0}$ case, we have $2^\alpha(2^{n-\alpha} - 1) = 2^n - 2^\alpha$ strings $\widetilde{y}$.

As a sanity check, note that summing the four values obtained above yields precisely $d = 2^{n+1}$ strings $\widetilde{y}$, which is the dimension of each register $\mathcal{Y}_i$, as desired.

To now obtain an expression for $|R_{\alpha,z}|$, recall that any set of m queries is successful if it contains at least one successful 0-query and one successful 1-query. Thus, using the inclusion-exclusion formula (intuition to follow):

$$|R_{\alpha,z}| = 2^{m(n+1)} - \sum_{a=0}^{m} \binom{m}{a} (2^\alpha)^a \left[\sum_{b=0}^{m-a} \binom{m-a}{b} \left(2^n - 2^{n-\alpha}\right)^b \left(2^n - 2^\alpha\right)^{m-a-b} \right] \tag{132}$$

$$- \sum_{a=0}^{m} \binom{m}{a} (2^{n-\alpha})^a \left[\sum_{b=0}^{m-a} \binom{m-a}{b} \left(2^n - 2^{n-\alpha}\right)^b \left(2^n - 2^\alpha\right)^{m-a-b} \right] \tag{133}$$

$$+ \sum_{b=0}^{m} \binom{m}{b} \left(2^n - 2^{n-\alpha}\right)^b \left(2^n - 2^\alpha\right)^{m-b}. \tag{134}$$

Above, the first term is the set of all query strings on m messages. The negative terms count the number of query strings with no successful 0-queries and no successful 1-queries, respectively. For the former, for example, we first choose a positions in which to put the successful 1-queries, and then distribute $\overline{0}$- and $\overline{1}$-queries among the remaining $m - a$ positions. The final, positive, term, counts the number of query strings with neither a successful 0- nor a successful 1-query. Inverting the binomial expansion $(a+b)^k = \sum_{l=0}^{k} \binom{k}{l} a^l b^{k-l}$, we can next write:

$$|R_{\alpha,z}| = 2^{m(n+1)} - \sum_{a=0}^{m} \binom{m}{a} (2^\alpha)^a \left[\left(2^n - 2^{n-\alpha}\right) + \left(2^n - 2^\alpha\right) \right]^{m-a} \tag{135}$$

$$- \sum_{a=0}^{m} \binom{m}{a} (2^{n-\alpha})^a \left[\left(2^n - 2^{n-\alpha}\right) + \left(2^n - 2^\alpha\right) \right]^{m-a} \tag{136}$$

$$+ \left[\left(2^n - 2^{n-\alpha}\right) + \left(2^n - 2^\alpha\right) \right]^m. \tag{137}$$

Applying the binomial expansion again yields

$$|R_{\alpha,z}| = 2^{m(n+1)} - \left[2^\alpha + \left(2^n - 2^{n-\alpha}\right) + \left(2^n - 2^\alpha\right) \right]^m \tag{138}$$

$$- \left[2^{n-\alpha} + \left(2^n - 2^{n-\alpha}\right) + \left(2^n - 2^\alpha\right) \right]^m \tag{139}$$

$$+ \left[\left(2^n - 2^{n-\alpha}\right) + \left(2^n - 2^\alpha\right) \right]^m. \tag{140}$$

Collecting like terms and factoring out $2^{m(n+1)}$ yields

$$|R_{\alpha,z}| = \left(2^{m(n+1)}\right) \left[1 - \left(1 - \frac{1}{2^{\alpha+1}}\right)^m - \left(1 - \frac{1}{2^{n-\alpha+1}}\right)^m + \left(1 - \frac{1}{2^{\alpha+1}} - \frac{1}{2^{n-\alpha+1}}\right)^m \right]. \tag{141}$$

Recall this was for any fixed secret key z. But for any fixed $0 \le \alpha \le n$, there are precisely $\binom{n}{\alpha} 2^\alpha 2^{n-\alpha} = \binom{n}{\alpha} 2^n$ choices of z with α qubits encoded in the Z-basis. Thus,

$$|R_\alpha| = \binom{n}{\alpha} \left(2^{m(n+1)+n}\right) \left[1 - \left(1 - \frac{1}{2^{\alpha+1}}\right)^m - \left(1 - \frac{1}{2^{n-\alpha+1}}\right)^m + \left(1 - \frac{1}{2^{\alpha+1}} - \frac{1}{2^{n-\alpha+1}}\right)^m \right]. \tag{142}$$

The final expression. The claim follows since the sets R_α partition R, and so $|R| = \sum_{\alpha=0}^{n} |R_\alpha|$. □

Using Lemma D.1 to heuristically bound β. Recall that our goal is to understand the dual value β from Equation (128) obtained by our dual solution, which in turn gives us a lower bound on the optimal cheating probability. So let us get a sense of how $|R|$ might scale asymptotically by deriving a heuristic approximation. To begin, applying the Hölder inequality to $|R|$ in Lemma D.1 yields

$$|R| \leq \left(2^{m(n+1)+2n}\right) \cdot \max_{\alpha} \left[1 - \left(1 - \frac{1}{2^{\alpha+1}}\right)^m - \left(1 - \frac{1}{2^{n-\alpha+1}}\right)^m + \left(1 - \frac{1}{2^{\alpha+1}} - \frac{1}{2^{n-\alpha+1}}\right)^m\right] \quad (143)$$

We shall assume[16] the maximum is attained for $\alpha = n/2$. For this choice of α, recalling that for large x,

$$\left(1 - \frac{1}{x}\right)^m \approx e^{-\frac{m}{x}}, \quad (144)$$

in the large n limit the term in the square brackets above is approximately $\left[1 - 2e^{-\frac{m}{2^{n/2+1}}} + e^{-\frac{m}{2^{n/2}}}\right]$. Hence, we may bound

$$|R| \leq \left(2^{m(n+1)+2n}\right) \left[1 - 2e^{-\frac{m}{2^{n/2+1}}} + e^{-\frac{m}{2^{n/2}}}\right] \quad (145)$$

$$\leq \left(2^{m(n+1)}\right) 4^n \left[1 - e^{-\frac{m}{2^{n/2}}}\right] \quad (146)$$

$$\approx d^m 4^n \frac{m}{2^{n/2}}, \quad (147)$$

for $m \ll n$, and where in the last line we used $d = 2^{n+1}$. Plugging this into Equation (128), we get precisely the type of behavior we want:

$$\beta = \frac{|R|}{4^n d^m} \lesssim \frac{m}{2^{n/2}}. \quad (148)$$

Thus, for polynomial m, the objective function value obtained by our dual solution from Equation (127)is exponentially small in the number of key bits, n (under the heuristic approximations made in this derivation).

The dual solution is not optimal. Naturally, this raises the question of whether our dual solution Y from Equation (127) is optimal. If it were, then a matching primal solution can in principle be found (it is easy to see that Slater's constraint qualification holds for the primal and dual, and so strong duality holds), and thus the optimal cheating probability would be approximately that given in Equation (148).

Unfortunately, Y is provably not dual optimal. Specifically, for $m = 2$ and $n = 1$ (2 queries, 1 key bit), the primal optimal value is ≈ 0.85 and for $m = 3, n = 1$, it is[17] 1 (both numerical values calculated via CVX in Matlab). However, evaluating $|R|$ in Lemma D.1 for these values of m and n yields $\beta = 0.25$ and $\beta = 0.46875$, respectively.

Moreover, the heuristic and loose upper bound on the objective function value of Y from Equation (148) is asymptotically not optimal[18], since the naive cheating strategy in which an adversary independently measures each qubit of $|\psi_z\rangle$ in basis $\left\{\cos\frac{\pi}{8}|0\rangle + \sin\frac{\pi}{8}|1\rangle, -\sin\frac{\pi}{8}|0\rangle + \cos\frac{\pi}{8}|1\rangle\right\}$ successfully obtains classical key $x \in \{0,1\}^n$ (see Program 1) with probability $(\cos^2\frac{\pi}{8})^n \approx 2^{-0.228n}$.

Thus, the dual solution Y of Equation (127) is not optimal. However, as the heuristic derivation of β from Equation (148) rather naturally led to the desired type of bound on the cheating probability, we conjecture that Y is *approximately* optimal, in the following sense.

[16] This assumption appears to hold in numerical calculations over various values of α.

[17] With $m = 3$ and $n = 1$, it is trivial to break the OTM construction. Namely, first measure the quantum key $|\psi_z\rangle$ in the standard basis and make an honest 0-query to extract the first secret bit. Then, since $|\psi_z\rangle$ is only 1 qubit, we can use brute force to make two 1-queries to the token with the only two possible candidate keys in the X-basis, 0, or 1.

[18] We thank David Mestel for bringing this to our attention.

Conjecture D.2. *The optimal values for the primal and dual SDPs of Section D.1 are, up to multiplicative scaling by some function $f(m,n) \in O(m^c 2^{(1/2-\epsilon)n})$ for constants $c > 0$ and $0 < \epsilon < 1/2$, equal to $\beta = \frac{|R|}{4^n d^m}$.*

If Conjecture D.2 holds, then our protocol would be secure in the sense that the optimal cheating probability would scale as $\text{poly}(m)/2^{\Theta(n)}$.

E 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

$$|\phi_1\rangle := U_i|\psi\rangle_{AB}|0\rangle_C. \tag{149}$$

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

$$|\phi_2\rangle := |\phi_y\rangle_A|y\rangle_B|0\rangle_C. \tag{150}$$

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.*,

$$|\phi_3\rangle := |\phi_y\rangle_A|y\rangle_B|s_i\rangle_C. \tag{151}$$

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 (149), which we can express as

$$|\phi_1\rangle = \sum_{y\in\{0,1\}^n} \alpha_y|\psi_y\rangle_A|y\rangle_B|0\rangle_C. \tag{152}$$

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 $f(y) = s_0$. 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 (150), obtaining y' with probability at least $1/\Delta$. Feeding y' into O_f yields s_0. Having obtained y', we have that $|\langle\phi_1|\phi_2\rangle|^2 \geq 1/\Delta$, implying

$$\left|\langle\psi|U_0^\dagger|\phi_{y'}\rangle|y'\rangle\right|^2 \geq 1/\Delta, \tag{153}$$

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

$$|\psi_{\text{approx}}\rangle = \beta_1|\psi_{\text{good}}\rangle + \beta_2|\psi_{\text{good}}^\perp\rangle, \tag{154}$$

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

$$\langle\psi_{\text{approx}}|\Pi_{\text{good}}|\psi_{\text{approx}}\rangle \geq |\beta_1|^2 \geq \frac{1}{\Delta}, \tag{155}$$

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*. 2012, pp. 41–60. DOI: 10.1145/2213977.2213983.

[BB84] Charles H. Bennett and Gilles Brassard. “Quantum cryptography: Public key distribution and coin tossing”. In: *International Conference on Computers, Systems and Signal Processing* (1984), pp. 175–179. DOI: 10.1016/j.tcs.2014.05.025.

[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”. In: *Journal of Cryptology* 4.2 (1991), pp. 75–122. DOI: 10.1007/BF00196771.

[Ben+92] Charles H. Bennett, Gilles Brassard, Claude Crépeau, and Marie-Hélène Skubiszewska. “Practical Quantum Oblivious Transfer”. In: *CRYPTO’91*. Ed. by Joan Feigenbaum. Vol. 576. LNCS. Springer, Aug. 1992, pp. 351–366.

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. “Non-Interactive Zero-Knowledge and Its Applications”. In: *20th ACM STOC*. ACM Press, 1988, pp. 103–112. DOI: 10.1145/62212.62222.

[BGS13] Anne Broadbent, Gus Gutoski, and Douglas Stebila. “Quantum One-Time Programs”. In: *CRYPTO 2013, Part II*. Ed. by Ran Canetti and Juan A. Garay. Vol. 8043. LNCS. Springer, Aug. 2013, pp. 344–360. DOI: 10.1007/978-3-642-40084-1˙20.

[BGZ15] Anne Broadbent, Sevag Gharibian, and Hong-Sheng Zhou. “Quantum One-Time Memories from Stateless Hardware”. arXiv:1511.01363. 2015.

[BM82] Manuel Blum and Silvio Micali. “How to Generate Cryptographically Strong Sequences of Pseudo Random Bits”. In: *23rd FOCS*. IEEE Computer Society Press, Nov. 1982, pp. 112–117. DOI: 10.1137/0213053.

[BS16] Anne Broadbent and Christian Schaffner. “Quantum Cryptography Beyond Quantum Key Distribution”. In: *Designs, Codes and Cryptography* 78.1 (2016), pp. 351–382. DOI: 10.1007/s10623-015-0157-4.

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

[Can00] Ran Canetti. “Security and Composition of Multiparty Cryptographic Protocols”. In: *Journal of Cryptology* 13.1 (2000), pp. 143–202. DOI: 10.1007/s001459910006.

[Can01] Ran Canetti. “Universally Composable Security: A New Paradigm for Cryptographic Protocols”. In: *42nd FOCS*. IEEE Computer Society Press, Oct. 2001, pp. 136–145. DOI: 10.1109/SFCS.2001.959888.

[Can+02] Ran Canetti, Yehuda Lindell, Rafail Ostrovsky, and Amit Sahai. “Universally composable two-party and multi-party secure computation”. In: *34th ACM STOC*. ACM Press, May 2002, pp. 494–503. DOI: 10.1145/509907.509980.

[Can+07] Ran Canetti, Yevgeniy Dodis, Rafael Pass, and Shabsi Walfish. “Universally Composable Security with Global Setup”. In: *TCC 2007*. Ed. by Salil P. Vadhan. Vol. 4392. LNCS. Springer, Feb. 2007, pp. 61–85. DOI: 10.1007/978-3-540-70936-7˙4.

[CGS08] Nishanth Chandran, Vipul Goyal, and Amit Sahai. “New Constructions for UC Secure Computation Using Tamper-Proof Hardware”. In: *EUROCRYPT 2008*. Ed. by Nigel P. Smart. Vol. 4965. LNCS. Springer, Apr. 2008, pp. 545–562. DOI: 10.5555/1788414.1788445.

[Cho+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: *TCC 2014*. Ed. by Yehuda Lindell. Vol. 8349. LNCS. Springer, Feb. 2014, pp. 638–662. DOI: 10.1007/978-3-642-54242-8˙27.

[Cho75] Man-Duen Choi. “Completely positive linear maps on complex matrices”. In: *Linear Alg. Appl.* 10 (1975), p. 285. DOI: 10.1016/0024-3795(75)90075-0.

[Chu+19] Kai-Min Chung, Marios Georgiou, Ching-Yi Lai, and Vassilis Zikas. “Cryptography with Disposable Backdoors”. In: *Cryptography* 3.3 (2019), p. 22. DOI: 10.3390/cryptography3030022.

[CM97] Christian Cachin and Ueli Maurer. “Unconditional security against memory-bounded adversaries”. In: *Advances in Cryptology - CRYPTO 1997*. LNCS. Springer, 1997, pp. 292–306. DOI: 10.1007/BFb0052243.

[Dam+05] 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*. IEEE, 2005, pp. 449–458. DOI: 10.1109/SFCS.2005.30.

[Dam+09] Ivan Damgård, Serge Fehr, Carolin Lunemann, Louis Salvail, and Christian Schaffner. "Improving the Security of Quantum Protocols via Commit-and-Open". In: *CRYPTO 2009*. Ed. by Shai Halevi. Vol. 5677. LNCS. Springer, Aug. 2009, pp. 408–427. DOI: 10.1007/978-3-642-03356-8˙24.

[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. Springer, 2010, pp. 685–706. DOI: 10.1007/978-3-642-14623-7˙37.

[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*. Vol. 7417. LNCS. Springer, 2012, pp. 794–811. DOI: 10.1007/978-3-642-32009-5˙46.

[DS13] Ivan Damgård and Alessandra Scafuro. "Unconditionally Secure and Universally Composable Commitments from Physical Assumptions". In: *ASIACRYPT 2013, Part II*. Ed. by Kazue Sako and Palash Sarkar. Vol. 8270. LNCS. Springer, Dec. 2013, pp. 100–119. DOI: 10.1007/978-3-642-42045-0˙6.

[Feh+13] Serge Fehr, Jonathan Katz, Fang Song, Hong-Sheng Zhou, and Vassilis Zikas. "Feasibility and Completeness of Cryptographic Tasks in the Quantum World". In: *TCC 2013*. Ed. by Amit Sahai. Vol. 7785. LNCS. Springer, Mar. 2013, pp. 281–296. DOI: 10.1007/978-3-642-36594-2˙16.

[FK18] Bill Fefferman and Shelby Kimmel. "Quantum vs. Classical Proofs and Subset Verification". In: *43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018)*. Ed. by Igor Potapov, Paul Spirakis, and James Worrell. Vol. 117. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018, 22:1–22:23. DOI: 10.4230/LIPIcs.MFCS.2018.22.

[Gav12] Dmitry Gavinsky. "Quantum Money with Classical Verification". In: *Computational Complexity (CCC), 2012 IEEE 27th Annual Conference on*. 2012, pp. 42–52. DOI: 10.1109/CCC.2012.10.

[GKR08] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. "One-Time Programs". In: *CRYPTO 2008*. Ed. by David Wagner. Vol. 5157. LNCS. Springer, Aug. 2008, pp. 39–56. DOI: 10.1007/978-3-540-85174-5˙3.

[GL91] Shafi Goldwasser and Leonid A. Levin. "Fair Computation of General Functions in Presence of Immoral Majority". In: *CRYPTO'90*. Ed. by Alfred J. Menezes and Scott A. Vanstone. Vol. 537. LNCS. Springer, Aug. 1991, pp. 77–93. DOI: 10.1007/3-540-38424-3˙6.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. "How to Play any Mental Game or A Completeness Theorem for Protocols with Honest Majority". In: *19th ACM STOC*. Ed. by Alfred Aho. ACM Press, May 1987, pp. 218–229. DOI: 10.1145/3335741.3335755.

[Goy+10] Vipul Goyal, Yuval Ishai, Amit Sahai, Ramarathnam Venkatesan, and Akshay Wadia. "Founding Cryptography on Tamper-Proof Hardware Tokens". In: *TCC 2010*. Ed. by Daniele Micciancio. Vol. 5978. LNCS. Springer, Feb. 2010, pp. 308–326. DOI: 10.1007/978-3-642-11799-2˙19.

[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)*. 2007, pp. 565–574. DOI: 10.1145/1250790.1250873.

[Hei27] Werner Heisenberg. "Schwankungserscheinungen und Quantenmechanik". In: *Zeitschrift fuer Physik* 40.7 (July 1927), pp. 501–506. ISSN: 14346001. DOI: 10.1007/BF01440827.

[HSS11] Sean Hallgren, Adam Smith, and Fang Song. "Classical Cryptographic Protocols in a Quantum World". In: *CRYPTO 2011*. Ed. by Phillip Rogaway. Vol. 6841. LNCS. Springer, Aug. 2011, pp. 411–428. DOI: 10.1007/978-3-642-22792-9˙23.

[IPS08] Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. "Founding Cryptography on Oblivious Transfer - Efficiently". In: *CRYPTO 2008*. Ed. by David Wagner. Vol. 5157. LNCS. Springer, Aug. 2008, pp. 572–591. DOI: 10.1007/978-3-540-85174-5˙32.

[Jam72] Andrzej Jamiołkowski. "Linear Transformations which preserve trace and positive semi-definiteness of operators". In: *Rep. Math. Phys.* 3 (1972), p. 275. DOI: 10.1016/0034-4877(72)90011-0.

[Kat07] Jonathan Katz. "Universally Composable Multi-party Computation Using Tamper-Proof Hardware". In: *EUROCRYPT 2007*. Ed. by Moni Naor. Vol. 4515. LNCS. Springer, May 2007, pp. 115–128. DOI: 10.1007/978-3-540-72540-4˙7.

[Kil88] Joe Kilian. "Founding Cryptography on Oblivious Transfer". In: *20th ACM STOC*. ACM Press, May 1988, pp. 20–31. DOI: 10.1145/62212.62215.

[KMQ11] Daniel Kraschewski and Jörn Müller-Quade. "Completeness Theorems with Constructive Proofs for Finite Deterministic 2-Party Functions". In: *TCC 2011*. Ed. by Yuval Ishai. Vol. 6597. LNCS. Springer, Mar. 2011, pp. 364–381. DOI: 10.1007/978-3-642-19571-6˙22.

[Kra+14] Daniel Kraschewski, Hemanta K. Maji, Manoj Prabhakaran, and Amit Sahai. "A Full Characterization of Completeness for Two-Party Randomized Function Evaluation". In: *EUROCRYPT 2014*. Ed. by Phong Q. Nguyen and Elisabeth Oswald. Vol. 8441. LNCS. Springer, May 2014, pp. 659–676. DOI: 10.1007/978-3-642-55220-5˙36.

[Liu14a] Yi-Kai Liu. "Building one-time memories from isolated qubits". In: *ITCS 2014*. Ed. by Moni Naor. ACM, Jan. 2014, pp. 269–286. DOI: 10.1145/2554797.2554823.

[Liu14b] Yi-Kai Liu. "Single-Shot Security for One-Time Memories in the Isolated Qubits Model". In: *CRYPTO 2014, Part II*. Ed. by Juan A. Garay and Rosario Gennaro. Vol. 8617. LNCS. Springer, Aug. 2014, pp. 19–36. DOI: 10.1007/978-3-662-44381-1˙2.

[Liu15] Yi-Kai Liu. "Privacy Amplification in the Isolated Qubits Model". In: *EUROCRYPT 2015, Part II*. Ed. by Elisabeth Oswald and Marc Fischlin. Vol. 9057. LNCS. Springer, Apr. 2015, pp. 785–814. DOI: 10.1007/978-3-662-46803-6˙26.

[LPV09] Huijia Lin, Rafael Pass, and Muthuramakrishnan Venkitasubramaniam. "A unified framework for concurrent security: universal composability from stand-alone non-malleability". In: *41st ACM STOC*. Ed. by Michael Mitzenmacher. ACM Press, 2009, pp. 179–188. DOI: 10.1145/1536414.1536441.

[Mau92] Ueli M. Maurer. "Protocols for Secret Key Agreement by Public Discussion Based on Common Information". In: *Advances in Cryptology - CRYPTO 1992*. Vol. 740. LNCS. Springer, 1992, pp. 461–470. DOI: 10.1007/3-540-48071-4˙32.

[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: *TCC 2009*. Ed. by Omer Reingold. Vol. 5444. LNCS. Springer, Mar. 2009, pp. 256–273. DOI: 10.1007/978-3-642-00457-5˙16.

[MPR10] Hemanta K. Maji, Manoj Prabhakaran, and Mike Rosulek. "A Zero-One Law for Cryptographic Complexity with Respect to Computational UC Security". In: *CRYPTO 2010*. Ed. by Tal Rabin. Vol. 6223. LNCS. Springer, Aug. 2010, pp. 595–612. DOI: 10.1007/978-3-642-14623-7˙32.

[MR11] Ueli Maurer and Renato Renner. "Abstract Cryptography". In: *ICS 2011*. Ed. by Bernard Chazelle. Tsinghua University Press, Jan. 2011, pp. 1–21. DOI: 10.1.1.402.6462.

[MR92] Silvio Micali and Phillip Rogaway. "Secure Computation (Abstract)". In: *CRYPTO'91*. Ed. by Joan Feigenbaum. Vol. 576. LNCS. Springer, Aug. 1992, pp. 392–404. DOI: 10.1007/3-540-46766-1˙32.

[MVW13] Abel Molina, Thomas Vidick, and John Watrous. "Optimal Counterfeiting Attacks and Generalizations for Wiesner's Quantum Money". In: *Theory of Quantum Computation, Communication, and Cryptography*. Ed. by Kazuo Iwama, Yasuhito Kawano, and Mio Murao. Vol. 7582. LNCS. Springer, 2013, pp. 45–64. DOI: 10.1007/978-3-642-35656-8˙4.

[NC00] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[Pas+12] Fernando Pastawski, Norman Y Yao, Liang Jiang, Mikhail D Lukin, and J Ignacio Cirac. "Unforgeable noise-tolerant quantum tokens". In: *Proceedings of the National Academy of Sciences* 109.40 (2012), pp. 16079 –16082. DOI: 10.1073/pnas.1203552109.

[PR08] Manoj Prabhakaran and Mike Rosulek. "Cryptographic Complexity of Multi-Party Computation Problems: Classifications and Separations". In: *CRYPTO 2008*. Ed. by David Wagner. Vol. 5157. LNCS. Springer, Aug. 2008, pp. 262–279. DOI: 10.1007/978-3-540-85174-5˙15.

[PS04] Manoj Prabhakaran and Amit Sahai. "New notions of security: Achieving universal composability without trusted setup". In: *36th ACM STOC*. Ed. by László Babai. ACM Press, June 2004, pp. 242–251. DOI: 10.1145/1007352.1007394.

[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*. IEEE, 2001, pp. 184–200. DOI: 10.1109/SECPRI.2001.924298.

[Qui+19] Marco Túlio Quintino, Qingxiuxiong Dong, Atsushi Shimbo, Akihito Soeda, and Mio Murao. "Reversing Unknown Quantum Transformations: Universal Quantum Circuit for Inverting General Unitary Operations". In: *Phys. Rev. Lett.* 123 (21 2019), p. 210502. DOI: 10.1103/PhysRevLett.123.210502.

[Ren08] Renato Renner. "Security of Quantum Key Distribution". PhD thesis. ETH Zürich, Sept. 2008, pp. 1–127. DOI: 10.1142/S0219749908003256.

[Unr10] Dominique Unruh. "Universally Composable Quantum Multi-party Computation". In: *EUROCRYPT 2010*. Ed. by Henri Gilbert. Vol. 6110. LNCS. Springer, May 2010, pp. 486–505. DOI: 10.1007/978-3-642-13190-5˙25.

[Unr13] Dominique Unruh. "Everlasting Multi-party Computation". In: *CRYPTO 2013, Part II*. Ed. by Ran Canetti and Juan A. Garay. Vol. 8043. LNCS. Springer, Aug. 2013, pp. 380–397. DOI: 10.1007/978-3-642-40084-1˙22.

[Unr14] Dominique Unruh. "Revocable Quantum Timed-Release Encryption". In: *EUROCRYPT 2014*. Ed. by Phong Q. Nguyen and Elisabeth Oswald. Vol. 8441. LNCS. Springer, May 2014, pp. 129–146. DOI: 10.1007/978-3-642-55220-5˙8.

[Wat11] John Watrous. *Lecture 7: Semidefinite programming*. Latest version available at: `https://cs.uwaterloo.ca/~watrous/TQI-notes/`. 2011.

[Wie83] Stephen Wiesner. "Conjugate coding". In: *ACM SIGACT News* 15.1 (1983). Original article written circa 1970., pp. 78–88. DOI: 10.1145/1008908.1008920.

[Win99] Andreas Winter. "Coding theorem and strong converse for quantum channels". In: *IEEE Transactions on Information Theory* 45 (1999), pp. 2481–2485. DOI: 10.1109/18.796385.

[WST08] Stephanie Wehner, Christian Schaffner, and Barbara M. Terhal. "Cryptography from Noisy Storage". In: *Physical Review Letters* 100.22 (June 2008), p. 220502. DOI: 10.1103/PhysRevLett.100.220502.

[WW10] Stephanie Wehner and Andreas Winter. "Entropic uncertainty relations—a survey". In: *New J. Phys.* 12.2 (2010), p. 025009. ISSN: 1367-2630. DOI: 10.1088/1367-2630/12/2/025009.

[WZ82] William K. Wootters and Wojciech H. Zurek. "A single quantum cannot be cloned". In: *Nature* 299.5886 (1982), pp. 802–803. DOI: 10.1038/299802a0.

[Yao82] Andrew Chi-Chih Yao. "Theory and Applications of Trapdoor Functions". In: *23rd FOCS*. IEEE Computer Society Press, Nov. 1982, pp. 80–91. DOI: 10.1109/SFCS.1982.45.