Composable Security in the Bounded-Quantum-Storage Model

Stephanie Wehner[1] and Jürg Wullschleger[2]

[1] California Institute of Technology, IQI, Pasadena CA 91125, USA
[2] University of Bristol, University Walk, Bristol BS8 1TW, United Kingdom

Abstract. We give a new, simulation-based, definition for security in the bounded-quantum-storage model, and show that this definition allows for sequential composition of protocols. Damgård *et al.* (FOCS '05, CRYPTO '07) showed how to securely implement bit commitment and oblivious transfer in the bounded-quantum-storage model, where the adversary is only allowed to store a limited number of qubits. However, their security definitions did only apply to the standalone setting, and it was not clear if their protocols could be composed. Indeed, we show that these protocols are *not* composable in our framework without a small refinement. We then prove the security of their randomized oblivious transfer protocol with our refinement. Secure implementations of oblivious transfer and bit commitment follow easily by a (classical) reduction to randomized oblivious transfer.

1 Introduction

Secure two-party computation [1] allows two mutually distrustful players to jointly compute the value of a function without revealing more information about their inputs than can be inferred from the function value itself. The primitive known as oblivious transfer (OT) [2,3,4] is thereby of particular importance: *any* two-party computation can be implemented, if this primitive is available [5,6]. Another important primitive in this context is *bit commitment* (BC) [7]. But since bit commitment can be implemented from oblivious transfer, a direct implementation of bit commitment is only important if we cannot implement oblivious transfer itself, or if we want to improve efficiency. In oblivious transfer, the sender (Alice) chooses two bits x_0 and x_1, the receiver (Bob) chooses a bit c. The protocol of oblivious transfer allows Bob to retrieve x_c in such a way that Alice cannot gain any information about c. At the same time, Alice can be ensured that Bob only retrieves x_c, but no information about x_{1-c}.

Unfortunately, BC and OT are impossible to implement securely without any additional assumptions, even in the quantum model [8,9]. This result holds even in the presence of the so-called superselection rules [10]. Exact trade-offs on how well we can implement BC in the quantum world can be found in [11]. To circumvent this problem (classically and quantumly), we thus need to assume that the adversary is limited. In the classical case, one such limiting assumption is that the adversary is *computationally bounded*. In the quantum model, it is

L. Aceto et al. (Eds.): ICALP 2008, Part II, LNCS 5126, pp. 604–615, 2008.

also possible to securely implement both protocols provided that an adversary cannot measure more than a fixed number of qubits simultaneously [12]. String commitments can be obtained with very weak security parameters [13].

The Bounded-Quantum-Storage Model. In the quantum case, it is *very* difficult to store states even for a very short period of time. This leads to the protocol presented in [14,15], which show how to implement BC and OT if the adversary is not able to store *any* qubits at all. In [16,17], these ideas have been generalized to the *bounded-quantum-storage model*, where the adversary is computationally unbounded and allowed to have an unlimited amount of *classical* memory. However, he is only allowed a limited amount of *quantum* memory. The honest players do not require any quantum storage at all, making the protocols implementable using present day technology.

Security Definitions and Composability. As cryptographic protocols are almost never executed on their own, it is important that they remain secure when they are composed. [18,19,20] introduced *simulation-based* security definitions and showed that they can be composed *sequentially*, i.e, at any point in time at most one protocol is running. A stronger security definition called *universal composability* has been introduced in [21,22,23]. It guarantees that protocols can be securely composed in an arbitrary way (also concurrently) in any environment.

Based on earlier an earlier definition of security in the quantum setting [24], a simulation-based security definition has been presented in [25], however no composability theorem was proven. Universal composability in the quantum world has been introduced in [26], and independently in [27]. In [28], it has been shown that classical protocols are universally composable using their *classical* definitions, are secure against *quantum* adversaries.

1.1 Contribution

In [17], protocols for OT and BC have been presented and shown to be secure against adversaries who have bounded quantum storage. However, the proofs only guarantee security in a standalone setting, and it was not clear whether these protocols remain secure when they are composed with other protocols. Indeed, the following simple example shows that in some situations, the protocols presented in [16,17] do not guarantee security in a strong sense. (However, Fehr and Schaffner [29] recently showed that the original definitions still allow for some weak form of composability.) Suppose the adversary receives a large number of halves of EPR-pairs from the environment as his auxiliary input. He can then effectively enlarge his own quantum memory by teleporting quantum states to the environment, which has unlimited memory. The classical communication needed to teleport can be part of the adversary's classical storage that he later outputs. In the case of the protocol presented in [17] (where the security depends on the fact that the adversary does not know in which basis to measure before his quantum memory bound is applied) this allows the environment to distinguish easily between the real and the ideal setting.

We present a formal model for secure two-party computation in the bounded-quantum-storage model and show that our model implies that secure protocols are sequentially composable. Then, we slightly modify the protocol for randomized OT presented [17] by introducing a second memory bound and prove the security of the protocol in our model.

In the full version of this work, we give well-known *classical* reductions of BC and OT to randomized OT. An important consequence is that *any* secure function evaluation can be achieved in the bounded-quantum-storage model. This follows from the fact that the proof of [28] carries over to our model, which means that any classical protocol that is secure in the classical universal composability model is also secure in our model. Therefore, we can use the protocol from [30] (based on [6]) to implement any secure function evaluation [1].

2 Preliminaries

We use the term *computational basis* to refer to the basis given by $\{|0\rangle, |1\rangle\}$. We write $+$ for the computational basis, and let $|0\rangle_+ = |0\rangle$ and $|1\rangle_+ = |1\rangle$. The *Hadamard basis* is denoted by $\times$, and given by $\{|0\rangle_\times, |1\rangle_\times\}$, where $|0\rangle_\times = (|0\rangle + |1\rangle)/\sqrt{2}$ and $|1\rangle_\times = (|0\rangle - |1\rangle)/\sqrt{2}$. For a string $x \in \{0,1\}^n$ encoded in bases $b \in \{+,\times\}^n$, we write $|x\rangle_b = |x_1\rangle_{b_1}, \ldots, |x_n\rangle_{b_n}$. We also use 0 to denote $+$, and 1 to denote $\times$. Finally, we use $x_{|c}$ to denote the sub-string of an encoded string x consisting of all x_i where $b_i = c$.

We use the font $\mathcal{A}$ to label a quantum register, corresponding to a Hilbert space $\mathcal{A}$. A *quantum channel* from $\mathcal{A}$ to $\mathcal{B}$ is a completely positive trace preserving (CPTP) map $\Lambda : \mathcal{A} \to \mathcal{B}$. We also call a map from $\mathcal{A}$ to itself a *quantum operation*. Any quantum operation on the register $\mathcal{A}$ can be phrased as a unitary operation on $\mathcal{A}$ and an additional ancilla register $\mathcal{A}'$, where we trace out $\mathcal{A}'$ to obtain the actions of the quantum operation on register $\mathcal{A}$ [32]. We use $\mathbb{S}(\mathcal{A})$ to refer to the set of all quantum states in $\mathcal{A}$, and $\mathbb{T}(\mathcal{A})$ to refer to the set of all Hermitian matrices in $\mathcal{A}$. We use $\mathbf{U}$ to refer to a quantum operation, upper case letters X to refer to classical random variables, the font $\mathbb{S}$ for a set, and the font A to refer to a player in the protocol.

Our ability to distinguish two quantum states $\rho, \rho' \in \mathbb{S}(\mathcal{H})$ is determined by their *trace distance* defined as $D(\rho, \rho') := \frac{1}{2} \operatorname{Tr} |\rho - \rho'|$, where $|A| = \sqrt{A^\dagger A}$. The triangle inequality holds. I.e., for all ρ, ρ' and ρ'', we have $D(\rho, \rho'') \leq D(\rho, \rho') + D(\rho', \rho'')$. We also write $\rho \equiv_\varepsilon \rho'$, if $D(\rho, \rho') \leq \varepsilon$. For all practical purposes, $\rho \equiv_\varepsilon \rho$ means that the state ρ' behaves like the state ρ, except with probability ε [33]. For any quantum channel Λ, we have $D(\Lambda(\rho), \Lambda(\rho')) \leq D(\rho, \rho')$. Let $\rho_{AB} \in \mathbb{S}(\mathcal{A} \otimes \mathcal{B})$ be classical on $\mathcal{A}$, i.e. $\rho_{AB} = \sum_{x \in \mathcal{X}} P_X(x) |x\rangle\langle x| \otimes \rho_x$ for some distribution P_X over a finite set $\mathcal{X}$. We say that A *is ε-close to uniform with respect to* B, if $D(\rho_{AB}, \mathbb{I}_A/d \otimes \rho_B) \leq \varepsilon$, where $d = \dim(\mathcal{H}_A)$.

For random variables X and Y with joint distribution P_{XY}, the *smooth conditional min-entropy* [34] can be expressed in terms of an optimization over events

[1] Note that because our implementation of OT is physical, the results presented in [31] cannot be applied, as explained in [30] on page 11.

$\mathcal{E}$ occurring with probability at least $1-\varepsilon$. Let $P_{X\mathcal{E}|Y=y}(x)$ be the probability that $\{X=x\}$ *and* $\mathcal{E}$ occur conditioned on $Y=y$. We have $H_{\min}^{\varepsilon}(X|Y) = \max_{\mathcal{E}:\Pr(\mathcal{E})\geq 1-\varepsilon} \min_y \min_x(-\log P_{X\mathcal{E}|Y=y}(x))$.The smooth min-entropy allows us to use the following chain rule.

Lemma 1 (Chain Rule [34]). *For all random variables X, Y, Z and for all $\varepsilon, \varepsilon' > 0$, $H_{\min}^{\varepsilon+\varepsilon'}(X|YZ) \geq H_{\min}^{\varepsilon}(XY \mid Z) - \log|\mathbb{Y}| - \log(1/\varepsilon')$.*

We also need the monotonicity of the smooth min-entropy, $H_{\min}^{\varepsilon}(XY \mid Z) \geq H_{\min}^{\varepsilon}(X \mid Z)$. A function $h : \mathbb{S} \times \mathbb{X} \to \{0,1\}^{\ell}$ is called a *two-universal hash function* [35], if for all $x_0 \neq x_1 \in \mathbb{X}$, we have $\Pr[h(S,x_0) = h(S,x_1)] \leq 2^{-\ell}$ if S is uniform over $\mathbb{S}$. We thereby say that a random variable S is *uniform over* a set $\mathbb{S}$ if S is chosen from $\mathbb{S}$ according to the uniform distribution. The following theorem is from [17], stated slightly differently than in [33,36].

Theorem 1 (Privacy Amplification [33,36]). *Let X and Z be (classical) random variables distributed over $\mathbb{X}$ and $\mathbb{Z}$, and let Q be a random state of q qubits. Let $h : \mathbb{S} \times \mathbb{X} \to \{0,1\}^{\ell}$ be a two-universal hash function and let S be uniform over $\mathbb{S}$ and independent from X and Z. If $\ell \leq H_{\min}^{\varepsilon'}(X \mid Z) - q - 2\log(1/\varepsilon)$, then $h(S,X)$ is $(\varepsilon + 2\varepsilon')$-close to uniform with respect to (S,Z,Q).*

The following lemma that we prove in the full version follows from the uncertainty relation presented in [17].

Lemma 2. *Let $X \in \{0,1\}^n$ be a uniform random string, let $B \in \{+,\times\}^n$ be a uniform random basis. Let $|X\rangle_B = (|X_1\rangle_{B_1}, \ldots, |X_n\rangle_{B_n})$ be a state of n qubits, and let K be the outcome of an arbitrary measurement of $|X\rangle_B$, which does not depend on X and B. Then, for any ε, we have $H_{\min}^{\varepsilon}(X|BK) \geq \frac{n}{2} - 10\sqrt[3]{n^2 \log \frac{1}{\varepsilon}}$, which is positive if $n > 8000\log(1/\varepsilon)$.*

3 Security in the Bounded-Quantum-Storage Model

We now give a definition of offline-security in the bounded-quantum-storage model, and show that it allows protocols to be composed *sequentially* (at any given time only one sub-protocol is executed). More detail can be found in the long version of our paper. Our definitions are closely related to [25].

We look at the following setting: Two *players*, A and B, execute a *protocol* $\mathbf{P} = (\mathbf{P}_{\mathsf{A}}, \mathbf{P}_{\mathsf{B}})$, where $\mathbf{P}_{\mathsf{A}}$ is the program executed by A and $\mathbf{P}_{\mathsf{B}}$ the program executed by B. Before the first round, each program receives an input (that might be entangled with the input of the other player) and stores it. In each round, each program may first send/receive messages to/from a given functionality $\mathbf{G}$, then apply a quantum operation to its current internal storage (including the message space), and finally send/receive further messages at the end of each round. $\mathbf{G}$ defines the communication resources available between the players, modeled as an interactive quantum functionality. It may contain a classical and/or a quantum communication channel, or other functionalities such as oblivious transfer

or bit commitment. Finally, in the last step of the protocol each program outputs an output value. The execution of $\mathbf{P}$ using $\mathbf{G}$ (denoted by $\mathbf{P}(\mathbf{G})$) is a quantum channel, which takes the input of both parties to the output of both parties.

Players may be *honest*, which means that they follow the protocol, or they may be *corrupted*. All corrupted players belong to the *adversary*, $\mathbb{A} \subset \{\mathsf{A}, \mathsf{B}\}$. Note that we can ignore the case where both players are corrupted. To simplify the proofs, we assume the set $\mathbb{A}$ to be *static*, i.e., it is already fixed before the protocol starts. We take the adversary to be *active*, i.e., he may not follow the protocol. The adversary $\mathbb{A} = \{p\}$ may replace his part of the protocol $\mathbf{P}_p$ by another program $\mathbf{A}_p$. Opposed to $\mathbf{P}_p$, $\mathbf{A}_p$ receives some *auxiliary (quantum) input* at the start of the protocol that may also be entangled with the environment. This input can be given to the adversary from the environment, but also come from the output of an honest player from a previous run of the protocol. At the end of the protocol, the adversary may return a (quantum) output to the environment. There is no communication between the adversary and the environment between the beginning and the end of the protocol. After receiving the (quantum) output, the environment tries to distinguish the protocol from the ideal setting based on its knowledge of its own input and output to and from the adversary.

We do not restrict the computational power of $\mathbf{A}_p$ in any way, however we do limit its internal quantum storage to a certain *memory-bound* of m qubits. We call such an $\mathbf{A}_p$ *m-bounded*. $\mathbf{A}_p$ is allowed to perform arbitrary quantum operations in each round of the protocol. However after receiving his input, and after every round, all of his internal memory is measured, except for m qubits. He may, however, store an unlimited amount of classical information.

The *ideal functionality*, denoted by $\mathbf{F}$, defines what functionality we expect the protocol to implement. In this paper, we only consider *non-interactive* functionalities, i.e., both players can send it input only once at the beginning, and obtain the output only once at the end. These functionalities have the form of a quantum channel. To make the definitions more flexible, we allow $\mathbf{F}$ to look differently depending on whether both players are honest, or either A or B belongs to the adversary. So the ideal functionality is in fact a *collection of functionalities*, $\mathbf{F} = (\mathbf{F}_\emptyset, \mathbf{F}_{\{\mathsf{A}\}}, \mathbf{F}_{\{\mathsf{B}\}})$. $\mathbf{F}_\emptyset$ denotes the functionality for the case when both players are honest, and $\mathbf{F}_{\{\mathsf{A}\}}$ and $\mathbf{F}_{\{\mathsf{B}\}}$ for the cases when A or B respectively are dishonest. As a honest player does not know whether the other player is also honest or not, we require that $\mathbf{F}_{\{\mathsf{A}\}}$ ($\mathbf{F}_{\{\mathsf{B}\}}$) and $\mathbf{F}_{\{\emptyset\}}$ must look the same from him. We also require that $\mathbf{F}_{\{\mathsf{A}\}}$ and $\mathbf{F}_{\{\mathsf{B}\}}$ allow the adversary to play honestly, i.e., they must be at least as good for the adversary as the functionality $\mathbf{F}_\emptyset$.

As we formally define in the long version, we say that a protocol $\mathbf{P}$ having access to the functionality $\mathbf{G}$ implements a functionality $\mathbf{F}$, if the following conditions are satisfied: First of all, we require that output of the protocol is ε-close to that of $\mathbf{F}$, if both players are honest. Second, for $\mathbb{A} = \{p\}$, we require that the adversary attacking the protocol has basically no advantage over attacking $\mathbf{F}$ directly. We thus require that for every m-bounded program $\mathbf{A}_p$, there exists an s-bounded program $\mathbf{S}_p$ (called the *simulator*), such that the overall outputs of both situations are ε-close, for all inputs. For simplicity, we do not make any

restrictions on the efficiency of the simulators[2]. Also, we do not require him to use the adversary $\mathbf{A}_p$ as a black-box: $\mathbf{S}_p$ may be constructed from scratch, under full knowledge of the behavior of $\mathbf{A}_p$.

It is important to note that we allow the simulator to execute some or all actions of $\mathbf{A}_p$ in a single round. This will allow the simulator to execute $\mathbf{A}_p$ *without* a memory bound being applied: Recall, that a memory bound is applied only after each round. This model is motivated by the physically realistic assumption that such memory bounds are introduced by adding specific waiting times after each round. Since the adversary is computationally unbounded, he would essentially also be able to perform any computation before the memory bound is applied and hence the simulator does not gain any more powers than the adversary. In particular, this does not give the simulator any memory.

However, in order to make protocols composable with other protocols in our model, we do require the simulator to be memory-bounded as well. The amount of memory required by the simulator gives a bound on the *virtual* memory the adversary seems to have by attacking the real protocol instead of the ideal one. Ideally, we would like $\mathbf{S}_p$ to use the same amount of memory as $\mathbf{A}_p$.

An important property of our definition is that it allows protocols to be composed. The following theorem shows that in a secure protocol that is based on an ideal, non-interactive functionality $\mathbf{G}$ and some other functionalities $\mathbf{G}'$, we can replace $\mathbf{G}$ with a secure implementation of $\mathbf{G}$, without making the protocol insecure. We thereby denote the concatenation of the functionalities $\mathbf{G}$ and $\mathbf{G}'$ by $\mathbf{G}\|\mathbf{G}'$. The theorem requires that $\mathbf{G}$ is called sequentially, i.e., that no other sub-protocols are running parallel to $\mathbf{G}$. The proof uses the same idea as in the classical case [20].

Theorem 2 (Sequential Composition Theorem). *Let* $\mathbf{F}$ *and* $\mathbf{G}$ *be non-interactive, and* $\mathbf{G}'$ *and* $\mathbf{H}$ *arbitrary functionalities. Let* $\mathbf{P}(\mathbf{G}\|\mathbf{G}')$ *be a protocol that calls* $\mathbf{G}$ *sequentially and that implements* $\mathbf{F}$ *with error at most* ε_1 *secure against* m_1*-bounded adversaries using* s_1*-bounded simulators, and let* $\mathbf{Q}(\mathbf{H})$ *be a protocol that implements* $\mathbf{G}$ *with error at most* ε_2 *secure against* m_2*-bounded adversaries using* s_2*-bounded simulators, where* $m_2 \geq s_1$*. Then* $\mathbf{P}(\mathbf{Q}(\mathbf{H})\|\mathbf{G}')$ *implements* $\mathbf{F}$ *with error at most* $\varepsilon_1 + \varepsilon_2$*, secure against* $\min(m_1, m_2)$*-bounded adversaries using* s_2*-bounded simulators.*

4 Randomized Oblivious Transfer

We now apply our framework to the randomized OT protocol presented in [16]. In particular, we prove security with respect to the following definition of randomized oblivious transfer. We show in the long version how to obtain the standard notion of OT from randomized OT. Note that in our version of randomized OT, also the choice bit c of the receiver is randomized.

Definition 1 (Randomized oblivious transfer). $\binom{2}{1}$-ROT^{ℓ} *(or, if* ℓ *is clear from the context,* ROT*) is defined as* $\mathsf{ROT} = (\mathsf{ROT}_{\emptyset}, \mathsf{ROT}_{\{A\}}, \mathsf{ROT}_{\{B\}})$*, where*

[2] Recall the adversary is computationally unbounded as well.

- $\mathsf{ROT}_{\emptyset}$*: The functionality chooses uniformly at random the value* $(x_0, x_1) \in_R \{0,1\}^{2\ell}$ *and* $c \in_R \{0,1\}$*. It sends* (x_0, x_1) *to* A *and* (c, y) *to* B *where* $y = x_c$*.*
- $\mathsf{ROT}_{\{\mathsf{A}\}}$*: The functionality receives* $(x_0, x_1) \in \{0,1\}^{2\ell}$ *from* A*. Then, it chooses* $c \in_R \{0,1\}$ *uniformly at random and sends* (c, y) *to* B*, where* $y = x_c$*.*
- $\mathsf{ROT}_{\{\mathsf{B}\}}$*: The functionality receives* $(c, y) \in \{0,1\} \times \{0,1\}^{\ell}$ *from* B*. Then, it sets* $x_c = y$*, chooses* $x_{1-c} \in_R \{0,1\}^{\ell}$ *uniformly at random, and sends* (x_0, x_1) *to* A*.*

The protocol $\mathsf{BQS\text{-}OT} = (\mathsf{BQS\text{-}OT_A}, \mathsf{BQS\text{-}OT_B})$ uses a noiseless unidirectional quantum channel Q-Comm, and a noiseless unidirectional classical channel Comm, both from the sender to the receiver. Let $h : \mathcal{R} \times \{0,1\}^n \to \{0,1\}^{\ell}$ be a two-universal hash function. A memory bound is applied before step 1, and between step 2 and 3. The sender (A) and receiver (B) execute:

Protocol 1: $\mathsf{BQS\text{-}OT_A}$

1. Choose $x \in_R \{0,1\}^n$ and $b \in_R \{0,1\}^n$ uniformly at random.
2. Send $|x\rangle_b := (|x_1\rangle_{b_1}, \ldots, |x_n\rangle_{b_n})$ to Q-Comm, where $|x_i\rangle_{b_i}$ is x_i encoded in the basis b_i.
3. Choose $r_0, r_1 \in_R \mathcal{R}$ uniformly at random and send (b, r_0, r_1) to Comm.
4. Output $(s_0, s_1) := (h(r_0, x_{|0}), h(r_1, x_{|1}))$, where $x_{|j}$ is the string of all x_i where $b_i = j$.

Protocol 2: $\mathsf{BQS\text{-}OT_B}$

1. Choose $c \in_R \{0,1\}$ uniformly at random.
2. Receive the qubits $(q_1, \ldots, q_n)$ from Q-Comm and measure them in the basis c, which gives output $x' \in \{0,1\}^n$.
3. Receive (b, r_0, r_1) from Comm.
4. Output $(c, y) := (c, h(r_c, x'_{|c}))$, where $x'_{|c}$ is the string of all x'_i where $b_i = c$.

Security against the sender. We first consider the case when the sender, A, is dishonest. This case turns out to be quite straightforward and closely follows the proof given in [17]. We use the following letters to refer to the different classical and quantum registers available to the adversary: Let $\mathcal{Q}$ denote the quantum register. Note that since we assume that our adversary's memory is m-bounded, the size of $\mathcal{Q}$ does not exceed m. Let $\mathcal{M}_Q$ and $\mathcal{M}_K$ denote the quantum and classical registers, that hold the messages sent to the receiver. Let $\mathcal{K}$ denote the classical input register of the adversary. Finally, let $\mathcal{A}$ denote an auxiliary quantum register. Recall from Section 2, that any quantum operation on $\mathcal{Q}$ and $\mathcal{M}_Q$ can be implemented by a unitary followed by a measurement on an additional register $\mathcal{A}$. Wlog we let $\mathcal{A}$ and $\mathcal{M}_Q$ be measured in the computational basis to enforce a memory bound, and $\mathcal{Q}$ be the sole quantum memory.

To model quantum and classical input that a malicious A may receive, we let $\mathcal{Q}$ start out in any state ρ_{in}, unknown to the simulator. Likewise, $\mathcal{K}$ may contain some classical input k_{in} of A. Wlog we assume that all other registers start out in a fixed state of $|0\rangle$. We can then describe the actions of A by a single unitary

$\mathbf{A}_\mathsf{A}$ defined by

$$\mathbf{A}_\mathsf{A}(\underbrace{\rho_{\text{in}}}_{\mathcal{Q}} \otimes \underbrace{|0\rangle\langle 0|}_{\mathcal{A}} \otimes \underbrace{k_{\text{in}}}_{\mathcal{K}} \otimes \underbrace{|0\rangle\langle 0|}_{\mathcal{M}_Q} \otimes \underbrace{|0\rangle\langle 0|}_{\mathcal{M}_K})\mathbf{A}_\mathsf{A}^\dagger = \underbrace{\rho_{\text{out}}}_{\mathcal{Q},\mathcal{A}} \otimes \underbrace{k_{\text{in}}}_{\mathcal{K}} \otimes \underbrace{\rho_{x_b}}_{\mathcal{M}_Q} \otimes \underbrace{|br_0r_1\rangle\langle br_0r_1|}_{\mathcal{M}_K}.$$

Note that without loss of generality $\mathbf{A}_\mathsf{A}$ leaves $\mathcal{K}$ unmodified: since $\mathcal{K}$ is classical we can always copy its contents to $\mathcal{A}$ and let all classical output be part of $\mathcal{A}$. To enforce the memory bound, assume wlog that $\mathcal{A}$ and $\mathcal{M}_Q$ are now measured completely in the computational basis. We now show that for any adversary $\mathbf{A}_\mathsf{A}$ there exists an appropriate simulator $\mathbf{S}_\mathsf{A}$.

Lemma 3. *Protocol BQS-OT is secure against dishonest A.*

Proof. Let $\mathbf{S}_\mathsf{A}$ be defined as follows: $\mathbf{S}_\mathsf{A}$ runs $\mathbf{A}_\mathsf{A}$. Note that $\mathbf{S}_\mathsf{A}$ can effectively skip the wait time required for the memory bound to take effect, since he can execute $\mathbf{A}_\mathsf{A}$ in one round before his memory bound is applied, where we refer to Section 3 for an important discussion and justification of this procedure. The simulator then measures register $\mathcal{M}_Q$ in the basis determined by $\mathcal{M}_K$. This allows him to compute $s_0 = h(r_0, x_{|0})$ and $s_1 = h(r_1, x_{|1})$. $\mathbf{S}_\mathsf{A}$ then sends s_0 and s_1 to $\mathsf{ROT}_{\{\mathsf{A}\}}$. It is clear that since the simulator based his measurement on $\mathcal{M}_K$, s_0 and s_1 are consistent with the run of the protocol. Furthermore, note that $\mathbf{S}_\mathsf{A}$ did not need to touch register $\mathcal{Q}$ at all. We can thus immediately conclude that the environment can tell no difference between the real protocol and the ideal setting. □

Security against the receiver. The proof of security against a dishonest receiver requires a more careful treatment of the quantum input given to the adversary. The main idea behind our proof is that the memory bound in fact *fixes* a classical bit c. Our main challenge is to find a c that the simulator can calculate and that is consistent with the adversary and his input, while keeping the output state of the adversary intact. To do so, we use a generalization of the *min-entropy splitting lemma* in [17], which in turn is based on an earlier version of [37]. It states that if two random variables X_0 and X_1 together have high min-entropy, then we can define a random variable C, such that X_{1-C} has at least half of the original min-entropy. To find C, one must know the distributions of X_0 and X_1. In the following generalization, we do *not* exactly know the distribution of X_0 and X_1, since we assume that its distribution also depends on an unknown random variable J, distributed over a domain of the size 2^β. $\beta = 0$ gives the min-entropy splitting lemma in [17].

Lemma 4 (Generalized Min-Entropy Splitting Lemma). *Let $\varepsilon \geq 0$, and $0 < \beta < \alpha$. Let J be a random variable over $\{0, \ldots, 2^\beta - 1\}$, and let X_0, X_1 and K be random variables such that $H_{\min}^\varepsilon(X_0X_1 \mid KJ) \geq \alpha$. Let $f(x_1, k) = 1$, if there exists a $j \in \{0, \ldots, 2^\beta - 1\}$ such that $P_{X_1|KJ}(x_1, k, j) \geq 2^{-(\alpha-\beta)/2}$, and 0 otherwise, and let $C := f(X_1, K)$. We have $H_{\min}^\varepsilon(X_{1-C}C \mid KJ) \geq \frac{\alpha-\beta}{2}$.*

Proof. Let S_k^j be the set of values x_1 for which $P_{X_1|KJ}(x_1, k, j) \geq 2^{-(\alpha-\beta)/2}$. We have $|S_k^j| \leq 2^{(\alpha-\beta)/2}$, since all values in S_k^j have a probability that is at least $2^{-(\alpha-\beta)/2}$. Let $S_k := \bigcup_j S_k^j$. We have $|S_k| \leq 2^\beta \cdot 2^{(\alpha-\beta)/2} = 2^{(\alpha+\beta)/2}$.

Let $K = k$ and $J = j$. Because $C = 0$ implies that $X_1 \notin S_k$, and thus also that $X_1 \notin S_k^j$, we have $P_{X_1 C|KJ}(x_1, 0, k, j) < 2^{-(\alpha-\beta)/2}$. It follows from the assumption that there exists an event $\mathcal{E}$ with probability $1 - \varepsilon$ such that for all x_0, x_1, k and j, we have $P_{X_0 X_1 \mathcal{E}|KJ}(x_0, x_1, k, j) \leq 2^{-\alpha}$. Hence $P_{X_0 C \mathcal{E}|KJ}(x_0, 1, k, j) = \sum_{x_1 \in S_k} P_{X_0 X_1 \mathcal{E}|KJ}(x_0, x_1, k, j) \leq 2^{(\alpha+\beta)/2} \cdot 2^{-\alpha} = 2^{-(\alpha-\beta)/2}$. □

We now describe the actions of the adversary. Let $\mathcal{M}$ denote the register holding the quantum message he receives from the sender in step 2. Let his registers $\mathcal{Q}$, $\mathcal{A}$ and $\mathcal{K}$ be initialized as above. We can now describe the actions of the adversary by two unitaries, where a memory bound is applied after the first. The action of the adversary following step 2 can be described as a unitary $\mathbf{A}_{\mathsf{B}}^{(1)}$ as before. Note we can again assume that $\mathbf{A}_{\mathsf{B}}^{(1)}$ leaves $\mathcal{K}$ unmodified. To enforce the memory bound, we now let register $\mathcal{M}$ and $\mathcal{A}$ be measured in the computational basis. We use $\rho_{\text{out}} \in \mathcal{Q}$ to denote the adversary's quantum output, and $k_{\text{out}} \in \mathcal{M} \otimes \mathcal{A}$ to denote his classical output. After the memory bound is applied, the receiver obtains additional information from the sender. The actions of the adversary after step 3 can then be described by a unitary $\mathbf{A}_{\mathsf{B}}^{(2)}$ followed by a measurement of quantum registers $\mathcal{M}$ and $\mathcal{A}$ in the computational basis.

First, we analyze the case where the adversary's auxiliary quantum input is a pure state of β qubits. Note that this means that the adversary cannot be entangled with the environment. Then we extend it, by allowing the adversary some arbitrary mixed quantum auxiliary input.

Lemma 5. *Protocol BQS-OT is secure against dishonest B with an error of at most* 5ε*, if he receives a pure state quantum (auxiliary) input, and his quantum memory is bounded before step 1 by* β *qubits, and between step* 2 *and* 3 *by* m *qubits, for*

$$8\ell + 2\beta + 4m \leq n - 20\sqrt[3]{n^2 \log \frac{1}{\varepsilon}} - 12 \log \frac{1}{\varepsilon} - 4.$$

Proof. Let K_{in} be the classical auxiliary input the adversary receives, and let $|j\rangle$ for $j \in \{0, \ldots, 2^\beta - 1\}$ be a basis for the quantum auxiliary input. Any fixed auxiliary input $|j\rangle$ and k_{in} fixes a distribution $P_{X_0 X_1 K|J=j}$, where K is the classical value the adversary has after second memory bound. The choice of input state $|\Psi_{\text{in}}\rangle$ thus defines the distribution of J. First of all, the simulator simulates the actions of the sender following steps 1 and 2, using a random string X and a random basis B. The simulator then applies $\mathbf{A}_{\mathsf{B}}^{(1)}$, which gives him some classical output K_{out}, and a quantum state ρ_{out}. It follows from the uncertainty relation of Lemma 2 that $H_{\min}^{\varepsilon}(X \mid BK_{\text{out}}K_{\text{in}}) \geq \alpha$ for $\alpha := n/2 - 10\sqrt[3]{n^2 \log(1/\varepsilon)}$. Let $(X_0, X_1) := X$, where $X_0 := X_{|0}$ and $X_1 := X_{|1}$ are the substrings of X defined in the same way as in the protocol.

It follows from Lemma 4 and the fact that the simulator holds a description of $\mathbf{A}^{(1)}$, $K = (B, K_{\text{out}}, K_{\text{in}})$ and X_0, X_1 that he can calculate the value $C := f(X_1, K)$. This means that the simulator can construct a linear transformation $\mathbf{S}_{\mathsf{B}}$ acting on registers $\mathcal{Q}$, $\mathcal{M}$, $\mathcal{A}$, $\mathcal{K}$, $\mathcal{X}$, $\mathcal{B}$, $\mathcal{R}$, and $\mathcal{C}$ combining the actions of $\mathbf{A}_{\mathsf{A}}^{(1)}$ and the choice of c using the function f as defined in the min-entropy splitting

Lemma 4. We have

$$\mathbf{S}_{\mathsf{B}}\Big(\sum_j \alpha_j \underbrace{|j\rangle}_{\mathcal{Q}} \otimes \underbrace{|x_b\rangle}_{\mathcal{M}} \otimes \underbrace{|0\rangle}_{\mathcal{A}} \otimes \underbrace{|k_{\text{in}}\rangle}_{\mathcal{K}} \otimes \underbrace{|x\rangle}_{\mathcal{X}} \otimes \underbrace{|b\rangle}_{\mathcal{B}} \otimes \underbrace{|r_0, r_1\rangle}_{\mathcal{R}} \otimes \underbrace{|0\rangle}_{\mathcal{C}} \otimes \underbrace{|0\rangle}_{\mathcal{Y}}\Big) =$$

$$\sum_{q,m_1,a_1} \alpha_{q,m_1,a_1} \underbrace{|q\rangle}_{\mathcal{Q}} \otimes \underbrace{|m_1\rangle}_{\mathcal{M}} \otimes \underbrace{|a_1\rangle}_{\mathcal{A}} \otimes \underbrace{|k_{\text{in}}\rangle}_{\mathcal{K}} \otimes \underbrace{|x\rangle}_{\mathcal{X}} \otimes \underbrace{|b\rangle}_{\mathcal{B}} \otimes \underbrace{|r_0, r_1\rangle}_{\mathcal{R}} \otimes \underbrace{|c\rangle}_{\mathcal{C}} \otimes \underbrace{|s_0, s_1\rangle}_{\mathcal{Y}})$$

for any pure state input $|\Psi_{\text{in}}\rangle = \sum_j \alpha_j |j\rangle$. Wlog, all registers except $\mathcal{Q}$ are now measured in the computational basis as the memory bound takes effect. It is an important consequence of our generalized min-entropy splitting lemma that the simulator can measure register $\mathcal{C}$ in the computational basis to extract c without causing any disturbance to the quantum output: Note that the definition of f did not take j into account explicitly and hence $\mathcal{C}$ is not entangled with the quantum output. From Lemma 4 we thus have that $H^{\varepsilon}_{\min}(X_{1-C}C \mid K) \geq \frac{\alpha-\beta}{2}$. The simulator now chooses R_0 and R_1 uniformly at random and calculates $S_0 = h(R_0, X_0)$ and $S_1 = h(R_1, X_1)$. Since R_0 and R_1 are independent of X_0, X_1 and C, we have $H^{\varepsilon}_{\min}(X_{1-C}C \mid K) = H^{\varepsilon}_{\min}(X_{1-C}C \mid R_C K)$. Using the chain rule from Lemma 1 and the monotonicity of $H^{\varepsilon}_{\min}$, we obtain $H^{2\varepsilon}_{\min}(X_{1-C} \mid CR_C K S_C) \geq \frac{\alpha-\beta}{2} - \ell - 1 - \log\frac{1}{\varepsilon}$. By using the privacy amplification Theorem 1, we get that S_{1-C} is 5ε close to uniform with respect to $(R_0, R_1, C, S_C, B, K_{\text{out}}, K_{\text{in}})$ and ρ_{out} if $\ell \leq \frac{\alpha-\beta}{2} - \ell - 1 - \log\frac{1}{\varepsilon} - m - 2\log\frac{1}{\varepsilon}$. By replacing α and rearranging the terms we get the claimed equation.

The simulator now sets $Y := S_C$, and sends (C, Y) to $\mathsf{ROT}_{\{\mathsf{B}\}}$. To complete the simulation, he runs $\mathbf{A}^{(2)}_{\mathsf{A}}$ as the adversary would have. Note that the simulator did not require any more memory than the adversary itself, i.e., we can take $\mathbf{S}_{\mathsf{B}}$ to be m-bounded as well. Clearly, the simulator determined C solely from the classical output of the adversary and thus the adversary's output state in the simulated run is equal to the original output state of the adversary $\rho_{\text{out}} \otimes k_{\text{out}}$. Since the only difference between the simulation and the real execution is that in the simulation, S_{1-C} is chosen completely at random, the simulation is 5ε-close to the output of the real protocol. □

It remains to address the case where the receiver gets a mixed state quantum input. This is the case where the adversary receives a state that is entangled with the environment. Note that this means that we must decrease the size of the adversary's memory: If he could receive an entangled state of β qubits as input, he could use it to increase his memory to $m + \beta$ qubits by teleporting β qubits to the environment, and storing the remaining m. Hence, we now have to take the adversary to be m'-bounded, where $m' := m - \beta$. Luckily, using a a similar argument as in [38], we can now extend the argument given above: Note that for any pure state input $|\Psi\rangle = |\Psi_{\text{in}}\rangle \otimes k_{\text{in}}$, the output of the simulated adversary is *exactly* $\Lambda(|\Psi\rangle\langle\Psi|)$, where Λ is the adversary's channel. Since $\{|\Psi\rangle\langle\Psi| \,||\Psi\rangle \in \mathcal{Q} \otimes \mathcal{K}, \||\Psi\rangle\| = 1\}$ spans all of $\mathbb{T}(\mathcal{Q} \otimes \mathcal{K})$ and the map given by the simulation procedure is the same as Λ on all inputs, we can conclude that the complete map is equal to Λ. Note that the simulator does not need to consider the β qubits

that the adversary might have teleported to the environment: we can essentially view it as part of the original adversary's quantum memory, and the simulator bases his decision solely on the classical output of the adversary. Hence,

Lemma 6. *Protocol* BQS-OT *is secure against dishonest* B *with an error of at most* 5ε*, if he receives a quantum (auxiliary) input, and his quantum memory is bounded before step 1 by* β *qubits and between step* 2 *and* 3*, by* m *qubits, for* $8\ell + 6\beta + 4m \leq n - 20\sqrt[3]{n^2 \log \frac{1}{\varepsilon}} - 12 \log \frac{1}{\varepsilon} - 4$.

Theorem 3. *Protocol* BQS-OT(Q-Comm$\|$Comm) *implements* $\binom{2}{1}$-ROT$^{\ell}$ *with an error of at most* 5ε*, secure against* m*-bounded adversaries using* m*-bounded simulators, if* $8\ell + 10m \leq n - 20\sqrt[3]{n^2 \log \frac{1}{\varepsilon}} - 12 \log \frac{1}{\varepsilon} - 4$.

Acknowledgments

We thank S. Desrosiers and C. Schaffner for useful comments, and D. Unruh for a kind explanation of his work. SW is supported by NSF grant PHY-0456720. JW is supported by the EPSRC. Part of this work was done while SW was a PhD student at CWI, Amsterdam, and JW was a PhD student at ETH Zürich, and during a 3 month visit at McGill University, Montreal, Quebec.

References

1. Yao, A.C.: Protocols for secure computations. In: 23rd IEEE FOCS, pp. 160–164 (1982)
2. Wiesner, S.: Conjugate coding. SIGACT News 15(1), 78–88 (1983)
3. Rabin, M.O.: How to exchange secrets by oblivious transfer. Technical Report TR-81, Harvard Aiken Computation Laboratory (1981)
4. Even, S., Goldreich, O., Lempel, A.: A randomized protocol for signing contracts. Commun. ACM 28(6), 637–647 (1985)
5. Kilian, J.: Founding cryptography on oblivious transfer. In: Proceedings of the 20th STOC, pp. 20–31 (1988)
6. Crépeau, C., van de Graaf, J., Tapp, A.: Committed oblivious transfer and private multi-party computation. In: Coppersmith, D. (ed.) CRYPTO 1995. LNCS, vol. 963, pp. 110–123. Springer, Heidelberg (1995)
7. Blum, M.: Coin flipping by telephone a protocol for solving impossible problems. SIGACT News 15(1), 23–27 (1983)
8. Mayers, D.: Unconditionally secure quantum bit commitment is impossible. Physical Review Letters 78, 3414–3417 (1997)
9. Lo, H.K., Chau, H.F.: Is quantum bit commitment really possible? Physical Review Letters 78, 3410–3413 (1997)
10. Kitaev, A., Mayers, D., Preskill, J.: Superselection rules and quantum protocols. Physical Review A 69, 052326 (2004)
11. Spekkens, R., Rudolph, T.: Degrees of concealment and bindingness in quantum bit commitment protocols. Physical Review A 65, 012310 (2002)
12. Salvail, L.: Quantum bit commitment from a physical assumption. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 338–353. Springer, Heidelberg (1998)
13. Buhrman, H., Christandl, M., Hayden, P., Lo, H.K., Wehner, S.: Security of quantum bit string commitment depends on the information measure. Physical Review Letters 97, 250501 (2006)

14. Bennett, C.H., Brassard, G., Crépeau, C., Skubiszewska, H.: Practical quantum oblivious transfer. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 351–366. Springer, Heidelberg (1992)
15. Crépeau, C.: Quantum oblivious transfer. J. of Mod. Opt. 41(12), 2455–2466 (1994)
16. Damgård, I., Fehr, S., Salvail, L., Schaffner, C.: Cryptography in the Bounded Quantum-Storage Model. In: 46th IEEE FOCS, pp. 449–458 (2005)
17. Damgård, I., Fehr, S., Renner, R., Salvail, L., Schaffner, C.: A tight high-order entropic uncertainty relation with applications in the bounded quantum-storage model. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622. Springer, Heidelberg (2007)
18. Micali, S., Rogaway, P.: Secure computation. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 392–404. Springer, Heidelberg (1992)
19. Beaver, D.: Foundations of secure interactive computing. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 377–391. Springer, Heidelberg (1992)
20. Canetti, R.: Security and composition of multiparty cryptographic protocols. Journal of Cryptology 13(1), 143–202 (2000)
21. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: 42th IEEE FOCS, pp. 136–145 (2001)
22. Pfitzmann, B., Waidner, M.: A model for asynchronous reactive systems and its application to secure message transmission. In: IEEE SP, p. 184 (2001)
23. Backes, M., Pfitzmann, B., Waidner, M.: A universally composable cryptographic library (2003), `http://eprint.iacr.org/2003/015`
24. van de Graaf, J.: Towards a formal definition of security for quantum protocols. Ph.D. thesis (1998), `http://www.cs.mcgill.ca/~crepeau/PS/these-jeroen.ps`
25. Smith, A.: Multi-party quantum computation. Masters Thesis (2001), quant-ph/0111030
26. Ben-Or, M., Mayers, D.: General security definition and composability for quantum and classical protocols (2004), quant-ph/0409062
27. Unruh, D.: Simulatable security for quantum protocols (2004), quant-ph/0409125
28. Unruh, D.: Formal security in quantum cryptology. Student research project, Institut für Algorithmen und Kognitive Systeme. University of Karlsruhe (2002)
29. Fehr, S., Schaffner, C.: Composing quantum protocols in a classical environment (2008), arxiv:0804.1059
30. Estren, G.: Universally composable committed oblivious transfer and multi-party computation assuming only basic black-box. M.Sc. thesis, School of Computer Science. McGill University (2004)
31. Canetti, R., Lindell, Y., Ostrovsky, R., Sahai, A.: Universally composable two-party and multi-party secure computation. In: 34th STOC, pp. 494–503 (2002)
32. Hayashi, M.: Quantum Information: An introduction. Springer, Heidelberg (2006)
33. Renner, R., König, R.: Universally composable privacy amplification against quantum adversaries. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 407–425. Springer, Heidelberg (2005)
34. Renner, R., Wolf, S.: Simple and tight bounds for information reconciliation and privacy amplification. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 199–216. Springer, Heidelberg (2005)
35. Carter, J.L., Wegman, M.N.: Universal classes of hash functions. Journal of Computer and System Sciences 18, 143–154 (1979)
36. Renner, R.: Security of Quantum Key Distribution. PhD thesis, ETH Zurich, Switzerland (2005), `http://arxiv.org/abs/quant-ph/0512258`
37. Wullschleger, J.: Oblivious-transfer amplification. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515. Springer, Heidelberg (2007)
38. Watrous, J.: Zero-knowledge against quantum attacks (2005), quant-ph/0511020