Secure Two-Party Quantum Evaluation of Unitaries Against Specious Adversaries

Frédéric Dupuis[1⋆], Jesper Buus Nielsen[2], and Louis Salvail[3⋆⋆]

[1] Institute for Theoretical Physics, ETH Zurich, Switzerland
dupuis@phys.ethz.ch
[2] DAIMI, Aarhus University, Denmark
jbn@cs.au.dk
[3] Université de Montréal (DIRO), QC, Canada
salvail@iro.umontreal.ca

Abstract. We show that any two-party quantum computation, specified by a unitary which simultaneously acts on the registers of both parties, can be securely implemented against a quantum version of classical semi-honest adversaries that we call specious.
We first show that no statistically private protocol exists for swapping qubits against specious adversaries. The swap functionality is modeled by a unitary transform that is not sufficient for universal quantum computation. It means that universality is not required in order to obtain impossibility proofs in our model. However, the swap transform can easily be implemented privately provided a classical bit commitment scheme.
We provide a simple protocol for the evaluation of any unitary transform represented by a circuit made out of gates in some standard universal set of quantum gates. All gates except one can be implemented securely provided one call to swap made available as an ideal functionality. For each appearance of the remaining gate in the circuit, one call to a classical AND-box is required for privacy. The AND-box can easily be constructed from oblivious transfer. It follows that oblivious transfer is universal for private evaluations of unitaries as well as for classical circuits.
Unlike the ideal swap, AND-boxes are classical primitives and cannot be represented by unitary transforms. It follows that, to some extent, this remaining gate is the hard one, like the AND gate for classical two-party computation.

1 Introduction

In this paper, we address the problem of privately evaluating some unitary transform U upon a joint quantum input state held by two parties. Since unitaries model what quantum algorithms are implementing, we can see this problem as a natural extension of secure two-party evaluation of functions to the quantum

⋆ Supported by Canada's NSERC Postdoctoral Fellowship Program.
⋆⋆ Supported by Canada's NSERC discovery grant, MITACS, and the QuantumWorks networks(NSERC).

realm. Suppose that a state $|\phi_{\mathtt{in}}\rangle \in \mathcal{A} \otimes \mathcal{B}$ is the initial shared state where Alice holds register $\mathcal{A}$ and Bob holds register $\mathcal{B}$. Let $U \in \mathrm{U}(\mathcal{A} \otimes \mathcal{B})$ be some unitary transform acting upon $\mathcal{A}$ and $\mathcal{B}$. What cryptographic assumptions are needed for a private evaluation of $|\phi_{\mathtt{out}}\rangle = U|\phi_{\mathtt{in}}\rangle$ where *private* means that each player learns no more than in the ideal situation depicted in Fig. 1? Of course, answers to this question depend upon the adversary we are willing to tolerate.

In [21], it was shown that unitaries cannot be used to implement classical cryptographic primitives. Any non-trivial primitive implemented by unitaries will necessarily leak information toward one party. Moreover, this leakage is available to a weak class of adversaries that can be interpreted as the quantum version of classical semi-honest adversaries. It follows that quantum two-party computation of unitaries cannot be used to implement classical cryptographic primitives. This opens the possibility that the cryptographic assumptions needed for private evaluations of unitaries are weaker than for their classical counterpart. So, what classical cryptographic assumptions, if any, are required to achieve privacy in our setting? Are there unitaries more difficult to evaluate privately than others?

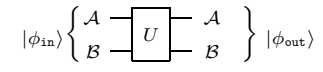


Fig. 1. Ideal Functionality for unitary U.

In this work, we answer these questions against a class of weak quantum adversaries, called specious, related to classical semi-honest adversaries. We say that a quantum adversary is specious if at any step during the execution of a protocol, it can provide a judge with some state that, when joined with the state held by the honest player, will be indistinguishable from a honest interaction. In other words, an adversary is specious if it can pass an audit with success at any step. Most known impossibility proofs in quantum cryptography apply when the adversary is restricted to be specious. Definitions similar to ours have been proposed for the quantum setting and usually named semi-honest. However, translating our definition to the classical setting produces a strictly stronger class of adversaries than semi-honest[4], as demonstrated in Appendix B which justifies not adopting the term *semi-honest*. We propose the name *specious* as the core of the definition is that the adversary must appear to act honestly.

Contributions. First, we define two-party protocols for the evaluation of unitaries having access to oracle calls. This allows us to consider protocols with security relying on some ideal functionalities in order to be private. We then say that a protocol is in the *bare model* if it does not involve any call to an ideal functionality. We then formally define what we mean by specious adversaries.

[4] As an example, assume there exist public key cryptosystems where you can sample a public key without learning the secret key. Then this is a semi-honest oblivious transform: The receiver, with choice bit c, samples pk_c in the normal way and learns its corresponding secret key and samples pk_{1-c} without learning its secret key. He sends (pk_0, pk_1). Then the sender sends $(E_{pk_0}(m_0), E_{pk_1}(m_1))$ and the receiver decrypts $E_{pk_c}(m_c)$. This is not secure against a specious adversary who can sample pk_{1-c} along with its secret key sk_{1-c} and then delete sk_{1-c} before the audit.

Privacy is then defined via simulation. We say that a protocol for the two-party evaluation of unitary U is private against specious adversaries if, for any joint input state and at any step of the protocol, there exists a simulator that can reproduce the adversary's view having only access to its own part of the joint input state. Quantum simulation must rely on a family of simulators for the view of the adversary rather than one because quantum information does not accumulate but can vanish as the protocol evolves. For instance, consider the trivial protocol that let Alice send her input register to Bob so that he can apply locally $|\phi_{\mathtt{out}}\rangle = U|\phi_{\mathtt{in}}\rangle$ before returning her register. The final state of such a protocol is certainly private, as Bob cannot clone Alice's input and keep a copy, yet at some point Bob had access to Alice's input thus violating privacy. No simulator can possibly reproduce Bob's state after he received Alice's register without having access to her input state.

Second, we show that no protocol can be shown statistically private against specious adversaries in the bare model for a very simple unitary: the swap gate. As the name suggests, the swap gate simply permutes Alice's and Bob's input states. Intuitively, the reason why this gate is impossible is that at some point during the execution of such protocol, one party that still has almost all its own input state receives a non-negligible amount of information (in the quantum sense) about the other party's input state. At this point, no simulator can possibly re-produce the complete state held by the receiving party since a call to the ideal functionality only provides access to the other party's state while no call to the ideal functionality only provides information about that party's own input. Therefore, any simulator cannot re-produce a state that contains information about the input states of both parties. It follows that cryptographic assumptions are needed for the private evaluation of unitaries against specious adversaries. On the other hand, a classical bit commitment is sufficient to implement the swap privately in our model.

Finally, we give a very simple protocol for the private evaluation of any unitary based on ideas introduced by [11, 10] in the context of fault tolerant quantum computation. Our construction is similar to Yao's original construction in the classical world[26, 13]. We represent any unitary U by a quantum circuit made out of gates taken from the universal set $\mathcal{UG} = \{X, Y, Z, \mathsf{CNOT}, \mathsf{H}, \mathsf{P}, \mathsf{R}\}$ [17]. The protocol evaluates each gate of the circuit upon shared encrypted input where the encryption uses the Pauli operators $\{X, Y, Z\}$ together with the identity. In addition to the Pauli gates X, Y, and Z, gates CNOT, H, and P can easily be performed over encrypted states without losing the ability to decrypt. Gates of that kind belong to what is called the *Clifford group*. The CNOT gate is the only gate in $\mathcal{UG}$ acting upon more than one qubit while the R-gate is the only one that does not belong to the Clifford group. In order to evaluate it over an encrypted state while preserving the ability to decrypt, we need to rely upon a classical ideal functionality computing securely an additive sharing for the AND of Alice's and Bob's input bits. We call this ideal functionality an AND-box. Upon input $x \in \{0, 1\}$ for Alice and $y \in \{0, 1\}$ for Bob, it produces $a \in_R \{0, 1\}$ and $b \in \{0, 1\}$ to Alice and Bob respectively such that $a \oplus b = x \wedge y$. An AND-box can be ob-

tained from any flavor of oblivious transfer and is defined the same way than an NL-box[18, 19] without the property that its output can be obtained before the input of the other player has been provided to the box (i.e., NL-boxes are non-signaling). The *equivalence* between AND-boxes, NL-boxes, and oblivious transfer is discussed in [25]. At the end of the protocol, each part of the shared key allowing to decrypt the output must be exchanged in a fair way. For this task, Alice and Bob rely upon an ideal swap functionality called SWAP. The result is that any U can be evaluated privately upon any input provided Alice and Bob have access to one AND-box per R-gate and one call to the an ideal swap. If the circuit happens to have only gates in the Clifford group then only one call to an ideal swap is required for privacy. In other words, SWAP is universal for the private evaluation of circuits in the Clifford group (i.e., those circuits having no R-gate) and itself belongs to that group (SWAP is not a classical primitive). To some extent, circuits in the Clifford group are the *easy* ones. Privacy for circuits containing R-gates however needs a classical cryptographic primitive to be evaluated privately by our protocol. It means that AND-boxes are universal for the private evaluation of any circuit against specious adversaries. We don't know whether there exist some unitary transforms that are universal for the private evaluation of any unitary against specious adversaries.

Previous works. All impossibility results in quantum cryptography we are aware of apply to classical primitives. In fact, the impossibility proofs usually rely upon the fact that an adversary with a seemingly honest behavior can force the implementation of classical primitives to behave quantumly. The result being that implemented that way, the primitive must leak information to the adversary. This is the spirit behind the impossibility of implementing oblivious transfer securely using quantum communication[14]. In that same paper the impossibility of any one-sided private evaluation of non-trivial primitives was shown. All these results can be seen as generalizations of the impossibility of bit commitment schemes based on quantum communication[15, 16]. The most general impossibility result we are aware of applies to any *non-trivial* two-party classical function[21]. It states that it suffices for the adversary to *purify* its actions in order for the quantum primitive to leak information. An adversary purifying its actions is specious as defined above. None of these impossibility proofs apply to quantum primitives characterized by some unitary transform applied to joint quantum inputs. Blind quantum computation is a primitive that shows similarities to ours. In [6], a protocol allowing a client to get its input to a quantum circuit evaluated blindly has been proposed. The security of their scheme is unconditional while in our setting almost no unitary allows for unconditional privacy.

An unpublished work of Smith[23] shows how one can devise a private protocol for the evaluation of any unitary that seems to remain private against all quantum adversaries. However, the techniques used require strong cryptographic assumptions like homomorphic encryption schemes, zero-knowledge and witness indistinguishable proof systems. The construction is in the spirit of protocols for multiparty quantum computation[4, 8] and fault tolerant quantum circuits[22,

2]. Although our protocol only guarantees privacy against specious adversaries, it is obtained using much weaker cryptographic assumptions.

Organization. We introduce protocols for the two-party evaluation of unitaries in Sect. 2.1. In Sect. 3, we define the class of specious quantum adversaries and in Sect. 3.3, we define privacy. We show in Sect. 4 that no private protocol exists for swap. The description of our protocol follows in Sect. 5 and the proof of privacy is in Appendix E.

2 Preliminaries

The N-dimensional complex Euclidean space (i.e., Hilbert space) will be denoted by $\mathcal{H}_N$. We denote quantum registers using calligraphic typeset $\mathcal{A}$. As usual, $\mathcal{A}\otimes\mathcal{B}$ denotes the space of two such quantum registers. We write $\mathcal{A}\approx\mathcal{B}$ when $\mathcal{A}$ and $\mathcal{B}$ are such that $\dim(\mathcal{A}) = \dim(\mathcal{B})$. A register $\mathcal{A}$ can undergo transformations as a function of time; we denote by $\mathcal{A}_i$ the state of space $\mathcal{A}$ at time i. When a quantum computation is viewed as a circuit accepting input in $\mathcal{A}$, we denote all wires in the circuit by $\mathtt{w}\in\mathcal{A}$. If the circuit accepts input in $\mathcal{A}\otimes\mathcal{B}$ then the set of all wires is denoted $\mathtt{w}\in\mathcal{A}\cup\mathcal{B}$.

The set of all linear mappings from $\mathcal{A}$ to $\mathcal{B}$ is denoted by $\mathrm{L}(\mathcal{A},\mathcal{B})$ while $\mathrm{L}(\mathcal{A})$ stands for $\mathrm{L}(\mathcal{A},\mathcal{A})$. To simplify notation, for $\rho\in\mathrm{L}(\mathcal{A})$ and $M\in\mathrm{L}(\mathcal{A},\mathcal{B})$ we write $M\cdot\rho$ for $M\rho M^\dagger$.

We denote by $\mathrm{Pos}(\mathcal{A})$ the set of positive semi-definite operators in $\mathcal{A}$. The set of positive semi-definite operators with trace 1 acting on $\mathcal{A}$ is denoted $\mathrm{D}(\mathcal{A})$; $\mathrm{D}(\mathcal{A})$ is the set of all possible quantum states for register $\mathtt{A}$. An operator $A\in\mathrm{L}(\mathcal{A},\mathcal{B})$ is called a *linear isometry* if $A^\dagger A = \mathbb{1}_{\mathcal{A}}$. The set of unitary operators (i.e., linear isometries with $\mathcal{B}=\mathcal{A}$) acting in $\mathcal{A}$ is denoted by $\mathrm{U}(\mathcal{A})$. The identity operator in $\mathcal{A}$ is denoted $\mathbb{1}_{\mathcal{A}}$ and the completely mixed state in $\mathrm{D}(\mathcal{A})$ is denoted by $\mathbb{I}_{\mathcal{A}}$. For any positive integer $N>0$, $\mathbb{1}_N$ and $\mathbb{I}_N$ denote the identity operator respectively the completely mixed state in $\mathcal{H}_N$. When the context requires, a pure state $|\psi\rangle\in\mathcal{AB}$ will be written $|\psi\rangle^{\mathcal{AB}}$ to make explicit the registers in which it is stored.

A linear mapping $\Phi:\mathrm{L}(\mathcal{A})\mapsto\mathrm{L}(\mathcal{B})$ is called a *super-operator* since it belongs to $\mathrm{L}(\mathrm{L}(\mathcal{A}),\mathrm{L}(\mathcal{B}))$. Φ is said to be *positive* if $\Phi(A)\in\mathrm{Pos}(\mathcal{B})$ for all $A\in\mathrm{Pos}(\mathcal{A})$. The super-operator Φ is said to be *completely positive* if $\Phi\otimes\mathbb{1}_{\mathrm{L}(\mathcal{Z})}$ is positive for every choice of the Hilbert space $\mathcal{Z}$. A super-operator Φ can be physically realized or is *admissible* if it is completely positive and preserves the trace: $\mathrm{tr}(\Phi(A)) = \mathrm{tr}(A)$ for all $A\in\mathrm{L}(\mathcal{A})$. We call such a super-operator a *quantum operation*. Any quantum operation $\Phi:\mathrm{L}(\mathcal{A})\mapsto\mathrm{L}(\mathcal{B})$ can be written in its Kraus form $\{E_j\}_{j=1}^{\dim(\mathcal{A})\cdot\dim(\mathcal{B})}$ where $E_j\in\mathrm{L}(\mathcal{A},\mathcal{B})$ for every j such that $\Phi(\rho)=\sum_j E_j\rho E_j^\dagger$, for any $\rho\in\mathrm{Pos}(\mathcal{A})$ and where $\sum_j E_j^\dagger E_j = \mathbb{1}_{\mathcal{B}}$. Another way to represent any quantum operation is through a linear isometry $W\in\mathrm{L}(\mathcal{A},\mathcal{B}\otimes\mathcal{Z})$ such that $\Phi(\rho)=\mathrm{tr}_{\mathcal{Z}}(W\cdot\rho)$, for some extra space $\mathcal{Z}$. Any such isometry W can be implemented by a physical process as long as the resource to

implement the space $\mathcal{Z}$ is available. This is just a unitary transform in $\mathrm{U}(\mathcal{A}\otimes\mathcal{Z})$ where the system in $\mathcal{Z}$ is initially in known state $|0_{\mathcal{Z}}\rangle$.

For two states $\rho_0, \rho_1 \in \mathrm{D}(\mathcal{A})$, we denote by $\Delta(\rho_0, \rho_1)$ the trace norm distance between ρ_0 and ρ_1: $\Delta(\rho_0, \rho_1) := \frac{1}{2}\|\rho_0 - \rho_1\|$. If $\Delta(\rho_0, \rho_1) \leq \varepsilon$ then any quantum process applied to ρ_0 behaves exactly as for ρ_1 except with probability at most ε [20].

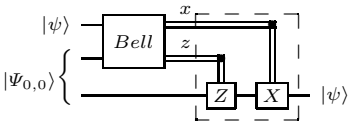


Fig. 2. The teleportation circuit

We let C_1 be the Pauli group (the set of tensor products of the three Pauli matrices X, Y, and Z, see Appendix A, and the 2×2 identity matrix $\mathbb{1}_2$). Furthermore, C_{i+1} is then defined recursively for $i \geq 1$ as $C_{i+1} \equiv \{U | UC_1U^\dagger \in C_i\}$, where C_2 is called the Clifford group.

The Bell measurement is a complete orthogonal measurement on two qubits made out of the measurement operators $\{|\Psi_{0,0}\rangle\langle\Psi_{0,0}|, |\Psi_{0,1}\rangle\langle\Psi_{0,1}|, |\Psi_{1,0}\rangle\langle\Psi_{1,0}|, |\Psi_{1,1}\rangle\langle\Psi_{1,1}|\}$ where $|\Psi_{0,0}\rangle := \frac{1}{\sqrt{2}}(|00\rangle+|11\rangle)$, $|\Psi_{0,1}\rangle := \frac{1}{\sqrt{2}}(|00\rangle-|11\rangle)$, $|\Psi_{1,0}\rangle := \frac{1}{\sqrt{2}}(|01\rangle+|10\rangle)$, and $|\Psi_{1,1}\rangle := \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$. The outcome $|\Psi_{x,z}\rangle$ of the Bell measurement is identified by the two classical bits $(x, z) \in \{0, 1\}^2$. The quantum one-time-pad is a perfectly secure encryption of quantum states[3]. It encrypts a qubit $|\psi\rangle$ as $X^x Z^z |\psi\rangle$, where the key is two classical bits, $(x, z) \in \{0, 1\}^2$ and $X^0Z^0 = \mathbb{1}$, $X^0Z^1 = Z$, $X^1Z^0 = X$ and $X^1Z^1 = Y$ are the Pauli operators. Quantum teleportation[5] can be used to implement the quantum one-time-pad. Consider the teleportation circuit in Fig. 2. If the state to encrypt is $|\psi\rangle$ then the state of the lower wire before entering the out-dashed box is the encryption of $|\psi\rangle$ under a uniformly random key produced by the Bell measurement. The two gates inside the dashed-box is the decryption circuit.

2.1 Modeling two-party strategies

Consider an interactive two-party strategy $\Pi^{\mathscr{O}}$ between parties $\mathscr{A}$ and $\mathscr{B}$ and oracle calls $\mathscr{O}$. $\Pi^{\mathscr{O}}$ can be modeled by a sequence of quantum operations for each player together with some oracle calls also modeled by quantum operations. Each quantum operation in the sequence corresponds to the action of one party at a certain step of the strategy. The following definition is a straightforward adaptation of n-turn interactive quantum strategies as described in [12]. The main difference is that here, we provide a joint input state to both parties and that quantum transmissions taking place during the execution is modeled by a quantum operation; one that is moving a state on one party's side to the other party.

Definition 2.1. *A n–*step two party strategy with oracle calls *denoted $\Pi^{\mathscr{O}} = (\mathscr{A}, \mathscr{B}, \mathscr{O}, n)$ consists of:*

1. *input spaces $\mathcal{A}_0$ and $\mathcal{B}_0$ for parties $\mathscr{A}$ and $\mathscr{B}$ respectively,*
2. *memory spaces $\mathcal{A}_1, \ldots, \mathcal{A}_n$ and $\mathcal{B}_1, \ldots, \mathcal{B}_n$ for $\mathscr{A}$ and $\mathscr{B}$ respectively,*
3. *an n-tuple of quantum operations $(\mathscr{A}_1, \ldots, \mathscr{A}_n)$ for $\mathscr{A}$, $\mathscr{A}_i : \mathrm{L}(\mathcal{A}_{i-1}) \mapsto \mathrm{L}(\mathcal{A}_i)$, $(1 \leq i \leq n)$,*

4. *an n-tuple of quantum operations $(\mathscr{B}_1, \ldots, \mathscr{B}_n)$ for $\mathscr{B}$, $\mathscr{B}_i : \mathrm{L}(\mathcal{B}_{i-1}) \mapsto \mathrm{L}(\mathcal{B}_i)$, $(1 \leq i \leq n)$,*
5. *memory spaces $\mathcal{A}_1, \ldots, \mathcal{A}_n$ and $\mathcal{B}_1, \ldots, \mathcal{B}_n$ can be written as $\mathcal{A}_i = \mathcal{A}_i^{\mathscr{O}} \otimes \mathcal{A}'_i$ and $\mathcal{B}_i = \mathcal{B}_i^{\mathscr{O}} \otimes \mathcal{B}'_i$, $(1 \leq i \leq n)$, and $\mathscr{O} = (\mathscr{O}_1, \mathscr{O}_2, \ldots, \mathscr{O}_n)$ is an n-tuple of quantum operations: $\mathscr{O}_i : \mathrm{L}(\mathcal{A}_i^{\mathscr{O}} \otimes \mathcal{B}_i^{\mathscr{O}}) \mapsto \mathrm{L}(\mathcal{A}_i^{\mathscr{O}} \otimes \mathcal{B}_i^{\mathscr{O}})$, $(1 \leq i \leq n)$.*

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

$$\begin{aligned}[\mathscr{A} \circledast \mathscr{B}](\rho_{\mathrm{in}}) :=& (\mathbb{1}_{\mathrm{L}(\mathcal{A}'_n \otimes \mathcal{B}'_n \otimes \mathcal{R})} \otimes \mathscr{O}_n)(\mathscr{A}_n \otimes \mathscr{B}_n \otimes \mathbb{1}_{\mathcal{R}}) \\ & \ldots (\mathbb{1}_{\mathrm{L}(\mathcal{A}'_1 \otimes \mathcal{B}'_1 \otimes \mathcal{R})} \otimes \mathscr{O}_1)(\mathscr{A}_1 \otimes \mathscr{B}_1 \otimes \mathbb{1}_{\mathcal{R}})(\rho_{\mathrm{in}}) \ .\end{aligned}$$

Step i of the strategy corresponds to the actions of $\mathscr{A}_i$ and $\mathscr{B}_i$ followed by the oracle call $\mathscr{O}_i$.

Note that we consider input states defined on the input systems together with a reference system $\mathcal{R}$; this allows us to show the correctness and privacy of the protocol not only for pure inputs, but also for inputs that are entangled with a third party. This is the most general case allowed by quantum mechanics.

A two-party strategy is therefore defined by quantum operation tuples $(\mathscr{A}_1, \ldots, \mathscr{A}_n)$, $(\mathscr{B}_1, \ldots, \mathscr{B}_n)$, and $(\mathscr{O}_1, \ldots, \mathscr{O}_n)$. These operations also define working spaces $\mathcal{A}_0, \ldots, \mathcal{A}_n, \mathcal{B}_0, \ldots, \mathcal{B}_n$ together with the input-output spaces to the oracle calls $\mathcal{A}_i^{\mathscr{O}}$ and $\mathcal{B}_i^{\mathscr{O}}$ for $1 \leq i \leq n$.

A *communication oracle* from Alice to Bob is modeled by having $\mathcal{A}_i^{\mathscr{O}} \approx \mathcal{B}_i^{\mathscr{O}}$ and letting $\mathscr{O}_i$ move the state in $\mathcal{A}_i^{\mathscr{O}}$ to $\mathcal{B}_i^{\mathscr{O}}$ and erase $\mathcal{A}_i^{\mathscr{O}}$. Similarly for communication in the other direction. We define a *bare model* protocol to be one which only uses communication oracles.

3 Specious Quantum Adversaries

3.1 Protocols for two-party evaluation

Let us consider two-party protocols for the quantum evaluation of unitary transform $U \in \mathrm{U}(\mathcal{A}_0 \otimes \mathcal{B}_0)$ between parties $\mathscr{A}$ and $\mathscr{B}$ upon joint input state $\rho_{\mathrm{in}} \in \mathrm{D}(\mathcal{A}_0 \otimes \mathcal{B}_0 \otimes \mathcal{R})$. We define these protocols as two-party interactive strategies with placeholder for the output as follows:

Definition 3.1. *A* two-party protocol $\Pi_U^{\mathscr{O}} = (\mathscr{A}, \mathscr{B}, \mathscr{O}, n)$ for $U \in \mathrm{U}(\mathcal{A}_0 \otimes \mathcal{B}_0)$ *is an n–step two-party strategy with oracle calls, where $\mathcal{A}_n \approx \mathcal{A}_0$ and $\mathcal{B}_n \approx \mathcal{B}_0$. It is said to be ε–*correct *if*

$$\Delta\left([\mathscr{A} \circledast \mathscr{B}](\rho_{\mathrm{in}}), (U \otimes \mathbb{1}_{\mathcal{R}}) \cdot \rho_{\mathrm{in}}\right) \leq \varepsilon \quad \text{for all } \rho_{\mathrm{in}} \in \mathrm{D}(\mathcal{A}_0 \otimes \mathcal{B}_0 \otimes \mathcal{R}) \ .$$

We denote by Π_U a two-party protocol in the bare model *where, without loss of generality, we assume that $\mathscr{O}_{2i+1}$ $(0 \leq i \leq \lfloor \frac{n}{2} \rfloor)$ implements a communication channel from $\mathscr{A}$ to $\mathscr{B}$ and $\mathscr{O}_{2i}$ $(1 \leq i \leq \lfloor \frac{n}{2} \rfloor)$ implements a communication channel from $\mathscr{B}$ to $\mathscr{A}$. Communication oracles are said to be* trivial.

In other words, a two-party protocol $\Pi_U^{\mathscr{O}}$ for unitary U is a two-party interactive strategy where, at the end, the output of the computation is stored in the memory of the players. $\Pi_U^{\mathscr{O}}$ is correct if, when restricted to the output registers (and $\mathcal{R}$), the final quantum state shared by $\mathscr{A}$ and $\mathscr{B}$ is $(U \otimes \mathbb{1}_{\mathcal{R}}) \cdot \rho_{\text{in}}$.

As it will become clear when we discuss privacy in Sect. 3.3, we need to consider the joint state at any step during the evolution of the protocol. We define,

$$\begin{aligned} \rho_1(\rho_{\text{in}}) &:= (\mathbb{1}_{\mathrm{L}(\mathcal{A}'_1 \otimes \mathcal{B}'_1 \otimes \mathcal{R})} \otimes \mathscr{O}_1)(\mathscr{A}_1 \otimes \mathscr{B}_1 \otimes \mathbb{1}_{\mathrm{L}(\mathcal{R})})(\rho_{\text{in}}), \\ \rho_{i+1}(\rho_{\text{in}}) &:= (\mathbb{1}_{\mathrm{L}(\mathcal{B}'_{i+1} \otimes \mathcal{A}'_{i+1} \otimes \mathcal{R})} \otimes \mathscr{O}_{i+1})(\mathscr{A}_{i+1} \otimes \mathscr{B}_{i+1} \otimes \mathbb{1}_{\mathrm{L}(\mathcal{R})})(\rho_i(\rho_{\text{in}})) \ , \quad (1) \end{aligned}$$

for $1 \leq i < n$. We also write the final state of $\Pi_U^{\mathscr{O}}$ upon input state ρ_{in} as $\rho_n(\rho_{\text{in}}) = [\mathscr{A} \circledast \mathscr{B}](\rho_{\text{in}})$.

3.2 Modeling Specious Adversaries

Intuitively, a specious adversary acts in any way apparently indistinguishable from the honest behavior, in the sense that no audit can distinguish the behavior of the adversary from the honest one.

More formally, a specious adversary in $\Pi_U^{\mathscr{O}} = (\mathscr{A}, \mathscr{B}, \mathscr{O}, n)$ may use an arbitrary large quantum memory space. However, at any step $1 \leq i \leq n$, the adversary can transform its own current state to one that is indistinguishable from the honest joint state. These transforms are modeled by quantum operations, one for each step of the adversary in $\Pi_U^{\mathscr{O}}$, and are part of the adversary's specification. We denote by $(\mathscr{T}_1, \ldots, \mathscr{T}_n)$ these quantum operations where $\mathscr{T}_i$ produces a valid transcript at the end of the i–th step.

Let $\tilde{\mathscr{A}}$ and $\tilde{\mathscr{B}}$ be adversaries in $\Pi_U^{\mathscr{O}}$. We denote by $\Pi_U^{\mathscr{O}}(\tilde{\mathscr{A}}) = (\tilde{\mathscr{A}}, \mathscr{B}, \mathscr{O}, n)$ and $\Pi_U^{\mathscr{O}}(\tilde{\mathscr{B}}) = (\mathscr{A}, \tilde{\mathscr{B}}, \mathscr{O}, n)$ the resulting n–step two-party strategies. We denote by $\tilde{\rho}_i(\tilde{\mathscr{A}}, \rho_{\text{in}})$ the state defined in (1) for protocol $\Pi_U^{\mathscr{O}}(\tilde{\mathscr{A}})$ and similarly by $\tilde{\rho}_i(\tilde{\mathscr{B}}, \rho_{\text{in}})$ that state for protocol $\Pi_U^{\mathscr{O}}(\tilde{\mathscr{B}})$.

Adding the possibility for the adversary to be ε-*close* to honest, we get the following definition:

Definition 3.2. *Let $\Pi_U^{\mathscr{O}} = (\mathscr{A}, \mathscr{B}, \mathscr{O}, n)$ be an n–step two-party protocol with oracle calls for $U \in \mathrm{U}(\mathcal{A}_0 \otimes \mathcal{B}_0)$. We say that:*

- *$\tilde{\mathscr{A}}$ is* ε–specious *if $\Pi_U^{\mathscr{O}}(\tilde{\mathscr{A}}) = (\tilde{\mathscr{A}}, \mathscr{B}, \mathscr{O}, n)$ is an n–step two-party strategy with $\tilde{\mathcal{A}}_0 = \mathcal{A}_0$ and there exists a sequence of quantum operations $(\mathscr{T}_1, \ldots, \mathscr{T}_n)$ such that:*
  1. *for every $1 \leq i \leq n$, $\mathscr{T}_i : \mathrm{L}(\tilde{\mathcal{A}}_i) \mapsto \mathrm{L}(\mathcal{A}_i)$,*
  2. *for every input state $\rho_{\text{in}} \in \mathrm{D}(\mathcal{A}_0 \otimes \mathcal{B}_0 \otimes \mathcal{R})$, and for all $1 \leq i \leq n$,*

$$\Delta\left((\mathscr{T}_i \otimes \mathbb{1}_{\mathrm{L}(\mathcal{B}_i \otimes \mathcal{R})})\left(\tilde{\rho}_i(\tilde{\mathscr{A}}, \rho_{\text{in}})\right), \rho_i(\rho_{\text{in}})\right) \leq \varepsilon \ .$$

- *$\tilde{\mathscr{B}}$ is* ε–specious *if $\Pi_U^{\mathscr{O}}(\tilde{\mathscr{B}}) = (\mathscr{A}, \tilde{\mathscr{B}}, \mathscr{O}, n)$ is a n–step two-party strategy with $\tilde{\mathcal{B}}_0 = \mathcal{B}_0$ and there exists a sequence of quantum operations $(\mathscr{T}_1, \ldots, \mathscr{T}_n)$ such that:*

1. *for every* $1 \leq i \leq n$, $\mathscr{T}_i : \mathrm{L}(\tilde{\mathcal{B}}_i) \mapsto \mathrm{L}(\mathcal{B}_i)$,
2. *for every input state* $\rho_{\mathrm{in}} \in \mathrm{D}(\mathcal{A}_0 \otimes \mathcal{B}_0 \otimes \mathcal{R})$, *and for all* $1 \leq i \leq n$,

$$\Delta\left(\left(\mathbb{1}_{\mathrm{L}(\mathcal{A}_i \otimes \mathcal{R})} \otimes \mathscr{T}_i\right)\left(\tilde{\rho}_i(\tilde{\mathscr{B}}, \rho_{\mathrm{in}})\right), \rho_i(\rho_{\mathrm{in}})\right) \leq \varepsilon \ .$$

If a party is $\varepsilon(m)$*–specious with* $\varepsilon(m)$ *negligible for* m *a security parameter then we say that this party is* statistically specious.

3.3 Privacy

Privacy for $\Pi_U^{\mathscr{O}}$ is defined as the ability for a simulator, having only access to the adversary's input and the ideal functionality U, to reproduce the state of the adversary at any step in the execution of $\Pi_U^{\mathscr{O}}$. Our definition is similar to the one introduced in [24] for statistical zero-knowledge proof systems.

A simulator for an adversary in $\Pi_U^{\mathscr{O}}$ is represented by a sequence of quantum operations $(\mathscr{S}_i)_{i=1}^n$, where $\mathscr{S}_i$ re-produces the view of the adversary after step i. $\mathscr{S}_i$ initially receives the adversary's input and has access to the ideal functionality for U evaluated upon the joint input of the adversary and the honest player. Because of no-cloning, a simulator calling U loses its input, and the input might be required to simulate e.g. early steps in the protocol, so we have to allow that $\mathscr{S}_i$ does not call U. For this purpose we introduce a bit $q_i \in \{0,1\}$. When $q_i = 0$, $\mathscr{S}_i$ does not call U and when $q_i = 1$, $\mathscr{S}_i$ must first call the ideal functionality U before performing some post-processing. More precisely,

Definition 3.3. *Let* $\Pi_U^{\mathscr{O}} = (\mathscr{A}, \mathscr{B}, \mathscr{O}, n)$ *be an* n*–step two-party protocol for* $U \in \mathrm{D}(\mathcal{A}_0 \otimes \mathcal{B}_0)$*. Then,*

- $\mathscr{S}(\tilde{\mathscr{A}}) = \langle(\mathscr{S}_1, \ldots, \mathscr{S}_n), q\rangle$ is a simulator for adversary $\tilde{\mathscr{A}}$ in $\Pi_U^{\mathscr{O}}$ *if it consists of:*
  1. *a sequence of quantum operations* $(\mathscr{S}_1, \ldots, \mathscr{S}_n)$ *where for* $1 \leq i \leq n$, $\mathscr{S}_i : \mathrm{L}(\mathcal{A}_0) \mapsto \mathrm{L}(\tilde{\mathcal{A}}_i)$,
  2. *a sequence of bits* $q \in \{0,1\}^n$ *determining if the simulator calls the ideal functionality at step* i*:* $q_i = 1$ *iff the simulator calls the ideal functionality.*
- *Similarly,* $\mathscr{S}(\tilde{\mathscr{B}}) = \langle(\mathscr{S}_1, \ldots, \mathscr{S}_n), q'\rangle$ is a simulator for adversary $\tilde{\mathscr{B}}$ in $\Pi_U^{\mathscr{O}}$ *if it consists of:*
  1. *a sequence of quantum operations* $(\mathscr{S}_1, \ldots, \mathscr{S}_n)$ *where for* $1 \leq i \leq n$, $\mathscr{S}_i : \mathrm{L}(\mathcal{B}_0) \mapsto \mathrm{L}(\tilde{\mathcal{B}}_i)$
  2. *a sequence of bits* $q' \in \{0,1\}^n$ *determining if the simulator calls the ideal functionality at step* i*:* $q'_i = 1$ *iff the simulator calls the ideal functionality.*

Given an input state $\rho_{\mathrm{in}} \in \mathrm{D}(\mathcal{A}_0 \otimes \mathcal{B}_0 \otimes \mathcal{R})$, we define the $\tilde{\mathscr{A}}$'s respectively $\tilde{\mathscr{B}}$'s simulated views as:

$$\nu_i(\tilde{\mathscr{A}}, \rho_{\mathrm{in}}) := \mathrm{tr}_{\mathcal{B}_0}\left(\left(\mathscr{S}_i \otimes \mathbb{1}_{\mathrm{L}(\mathcal{B}_0 \otimes \mathcal{R})}\right)\left((U^{q_i} \otimes \mathbb{1}_{\mathcal{R}}) \cdot \rho_{\mathrm{in}}\right)\right) \ ,$$

$$\nu_i(\tilde{\mathscr{B}}, \rho_{\text{in}}) := \text{tr}_{\mathcal{A}_0}\left((\mathbb{1}_{\text{L}(\mathcal{A}_0\otimes\mathcal{R})} \otimes \mathscr{S}_i)\left((U^{q'_i} \otimes \mathbb{1}_{\mathcal{R}}) \cdot \rho_{\text{in}}\right)\right) \ .$$

We say that protocol $\Pi_U^{\mathscr{O}}$ is private against specious adversaries if there exits a simulator for the view at any step of any such adversary. In more details,

Definition 3.4. *Let $\Pi_U^{\mathscr{O}} = (\mathscr{A}, \mathscr{B}, \mathscr{O}, n)$ be a protocol for $U \in \text{U}(\mathcal{A}_0 \otimes \mathcal{B}_0)$ and let $0 \leq \delta \leq 1$. We say that $\Pi_U^{\mathscr{O}}$ is* δ–private against ε–specious *$\tilde{\mathscr{A}}$ if there exists a simulator $\mathscr{S}(\tilde{\mathscr{A}})$ such that for all input states $\rho_{\text{in}} \in \text{D}(\mathcal{A}_0 \otimes \mathcal{B}_0 \otimes \mathcal{R})$ and for all $1 \leq i \leq n$, $\Delta\left(\nu_i(\tilde{\mathscr{A}}, \rho_{\text{in}}), \text{tr}_{\mathcal{B}_i}(\tilde{\rho}_i(\tilde{\mathscr{A}}, \rho_{\text{in}}))\right) \leq \delta$. Similarly, we say that Π_U is* δ–private against ε–specious *$\tilde{\mathscr{B}}$ if there exists a simulator $\mathscr{S}(\tilde{\mathscr{B}})$ such that for all input states $\rho_{\text{in}} \in \text{D}(\mathcal{A}_0 \otimes \mathcal{B}_0 \otimes \mathcal{R})$ and for all $1 \leq i \leq n$, $\Delta\left(\nu_i(\tilde{\mathscr{B}}, \rho_{\text{in}}), \text{tr}_{\mathcal{A}_i}(\tilde{\rho}_i(\tilde{\mathscr{B}}, \rho_{\text{in}}))\right) \leq \delta$. Protocol $\Pi_U^{\mathscr{O}}$ is* δ–private against ε–specious adversaries *if it is δ–private against both $\tilde{\mathscr{A}}$ and $\tilde{\mathscr{B}}$. For $\gamma > 0$, if $\Pi_U^{\mathscr{O}}$ is $2^{-\gamma m}$–private for $m \in \mathbb{N}^+$ a security parameter then we say that $\Pi_U^{\mathscr{O}}$ is* statistically private.

One should keep in mind that δ should be kept small compared to the number of rounds, since the protocol is only secure if we can ensure that, with high probability, the adversary cannot behave differently in the simulated world at *any* of the rounds. If δn is kept small, we can use the union bound over all the rounds to ensure this.

We show next that for some unitary, statistical privacy cannot be satisfied by any protocol in the bare model.

4 Unitaries with no private protocols

In this section, we show that no statistically private protocol for the swap gate exists in the bare model. The swap gate, denoted SWAP, is the following unitary transform:

$$\mathsf{SWAP} : |\phi_A\rangle^{\mathcal{A}_0}|\phi_B\rangle^{\mathcal{B}_0} \mapsto |\phi_B\rangle^{\mathcal{A}_0}|\phi_A\rangle^{\mathcal{B}_0} \ ,$$

for any one qubit states $|\phi_A\rangle \in \mathcal{A}_0$ and $|\phi_B\rangle \in \mathcal{B}_0$ (i.e., $\dim(\mathcal{A}_0) = \dim(\mathcal{B}_0) = 2$). Notice that SWAP is in the Clifford group since it can be implemented with three CNOT gates. It means that universality is not required (gates in the Clifford groups are not universal for quantum computation) for a unitary to be impossible to evaluate privately. The impossibility of SWAP essentially follows from no cloning.

Theorem 4.1 (Impossibility of swapping). *There is no correct and statistically private two-party protocol $\Pi_{\mathsf{SWAP}} = (\mathscr{A}, \mathscr{B}, \mathscr{O}, n(m))$ in the bare model.*

Using this line of reasoning, Theorem 4.1 can be extended to apply to any protocol for almost any unitary preventing both parties to recover their input states from its output.

Sufficient Assumptions for Private SWAP. A private protocol for SWAP in the bare model would exist if the players could rely on special relativity and a lower bound on their separation in space: they simply send their messages simultaneously. The fact that messages cannot travel faster than the speed of light ensures that the messages are independent of each other. It is also straightforward to devise a private protocol for SWAP based on commitment schemes. $\mathscr{A}$ sends one half EPR-pair to $\mathscr{B}$ while keeping the other half. $\mathscr{A}$ then teleports (without announcing the outcome of the measurement) her register and commits on the outcome of the Bell measurement. $\mathscr{B}$ sends his register to $\mathscr{A}$ before she opens her commitment. This allows $\mathscr{B}$ to reconstruct $\mathscr{A}$'s initial state.

5 The Protocol

We now describe a private protocol for the two-party evaluation of any unitary $U \in \mathrm{U}(\mathcal{A}_0 \otimes \mathcal{B}_0)$ denoted by $P_U^{\mathscr{O}} = (\mathscr{A}^*, \mathscr{B}^*, \mathscr{O}, n_U + 1)$ where U is represented by a circuit C_U with u gates in $\mathcal{UG}$. We slightly abuse the notation with respect to the parameter $n_U + 1$. Given circuit C_U, we let n_U be the number of oracle calls (including calls to communication oracles). Setting the last parameter to $n_U + 1$ instead of n_U comes from the fact that in our protocol, $\mathscr{A}^*$ and $\mathscr{B}^*$ have to perform a last operation each in order to get their outcome. These last operations do not involve a call to any oracle. Let G_j be the j-th gate in $C_U = G_u G_{u-1} \dots G_1$. The protocol is obtained by composing sub-protocols for each gate similarly to well-known classical constructions[26, 13]. Notice that $P_U^{\mathscr{O}}$ will not be presented in the form of Definition 3.1. $\mathscr{A}^*$ is not necessarily sending the first and the last messages. This can be done without consequences since we provide a simulation for each step where a message from the honest party is received or the output of a call to an ideal functionality is available. Putting $P_U^{\mathscr{O}}$ in the standard form of Definition 3.1 is straightforward and changes nothing to the proof of privacy.

The evaluation of each gate is performed over shared encrypted states. Each wire in C_U will be updated from initially holding the input $\rho_{\text{in}} \in \mathrm{D}(\mathcal{A}_0 \otimes \mathcal{B}_0 \otimes \mathcal{R})$ to finally holding the output $(U \otimes \mathbb{1}_{\mathcal{R}}) \cdot \rho_{\text{in}} \in \mathrm{D}(\mathcal{A}_0 \otimes \mathcal{B}_0 \otimes \mathcal{R})$. The state of wires $\mathtt{w} \in \mathcal{A}_0 \cup \mathcal{B}_0$ after the evaluation of G_j are stored at $\mathscr{A}^*$'s or $\mathscr{B}^*$'s according if $\mathtt{w} \in \mathcal{A}_0$ or $\mathtt{w} \in \mathcal{B}_0$. The shared encryption keys for wire $\mathtt{w} \in \mathcal{A}_0 \cup \mathcal{B}_0$ updated after the evaluation of G_j are denoted by $K^j_{\mathscr{A}^*}(\mathtt{w}) = (X^j_{\mathscr{A}^*}(\mathtt{w}), Z^j_{\mathscr{A}^*}(\mathtt{w})) \in \{0,1\}^2$ and $K^j_{\mathscr{B}^*}(\mathtt{w}) = (X^j_{\mathscr{B}^*}(\mathtt{w}), Z^j_{\mathscr{B}^*}(\mathtt{w})) \in \{0,1\}^2$ for $\mathscr{A}^*$ and $\mathscr{B}^*$ respectively and are held privately in internal registers of each party.

The final phase of the protocol is where a call to an ideal functionality is required. $\mathscr{A}^*$ and $\mathscr{B}^*$ exchange their own part of each encryption key for the other party's wires. In order to do this, the *key-releasing phase* invokes an ideal SWAP-gate as functionality: $\mathscr{O}_{n_U} : \mathrm{L}(\mathcal{A}^{\mathscr{O}}_{n_U} \otimes \mathcal{B}^{\mathscr{O}}_{n_U}) \mapsto \mathrm{L}(\mathcal{A}^{\mathscr{O}}_{n_U} \otimes \mathcal{B}^{\mathscr{O}}_{n_U})$, where $\mathscr{O}_{n_U}(\rho) := \mathsf{SWAP} \cdot \rho$. Upon joint input state $\rho_{\text{in}} \in \mathrm{D}(\mathcal{A}_0 \otimes \mathcal{B}_0 \otimes \mathcal{R})$, protocol $P_U^{\mathscr{O}(U)}$ runs the following phases:

Initialization: We assume that $\mathscr{A}^*$ and $\mathscr{B}^*$ have agreed upon a description of U by a circuit C_U made out of u gates $(G_1, \dots, G_u)$ in $\mathcal{UG}$. For all wires

$\mathtt{w} \in \mathcal{A}_0 \cup \mathcal{B}_0$, $\mathscr{A}^*$ and $\mathscr{B}^*$ set their initial encryption keys as $K^0_{\mathscr{A}^*}(\mathtt{w}) = (X^0_{\mathscr{A}^*}(\mathtt{w}), Z^0_{\mathscr{A}^*}(\mathtt{w})) := (0,0)$ and $K^0_{\mathscr{B}^*}(\mathtt{w}) = (X^0_{\mathscr{B}^*}(\mathtt{w}), Z^0_{\mathscr{B}^*}(\mathtt{w})) := (0,0)$ respectively.

Evaluation: For each gate number $1 \leq j \leq u$, $\mathscr{A}^*$ and $\mathscr{B}^*$ evaluate G_j as described in details below. This evaluation results in shared encryption under keys $K^j_{\mathscr{A}^*}(\mathtt{w}) = (X^j_{\mathscr{A}^*}(\mathtt{w}), Z^j_{\mathscr{A}^*}(\mathtt{w}))$ and $K^j_{\mathscr{B}^*}(\mathtt{w}) = (X^j_{\mathscr{B}^*}(\mathtt{w}), Z^j_{\mathscr{B}^*}(\mathtt{w}))$ for all wires $\mathtt{w} \in \mathcal{A}_0 \cup \mathcal{B}_0$, which at that point hold a shared encryption of $((G_j G_{j-1} \dots G_1) \otimes \mathbb{1}_{\mathcal{R}}) \cdot \rho_{\text{in}}$. Only the evaluation of the R-gate requires a call to an ideal functionality (i.e., an AND-BOX).

Key-Releasing: Let $\mathcal{A}^{\mathscr{O}}_{n_U}$ and $\mathcal{B}^{\mathscr{O}}_{n_U}$ be the set of registers holding respectively $K^u_{\mathscr{A}^*}(\mathtt{w}) = (X^u_{\mathscr{A}^*}(\mathtt{w}), Z^u_{\mathscr{A}^*}(\mathtt{w}))$ for $\mathtt{w} \in \mathcal{B}_0$ and $K^u_{\mathscr{B}^*}(\mathtt{w}) = (X^u_{\mathscr{B}^*}(\mathtt{w}), Z^u_{\mathscr{B}^*}(\mathtt{w}))$ for $\mathtt{w} \in \mathcal{A}_0$. We assume w.l.g that dimensions of both sets of registers are identical[5]:

1. $\mathscr{A}^*$ and $\mathscr{B}^*$ run the ideal functionality for the SWAP-gate upon registers $\mathcal{A}^{\mathscr{O}}_{n_U}$ and $\mathcal{B}^{\mathscr{O}}_{n_U}$.
2. $\mathscr{A}^*$ applies the decryption operator $K_{\mathscr{A}^*}(\mathtt{w}) = (X^u_{\mathscr{A}^*}(\mathtt{w}) \oplus X^u_{\mathscr{B}^*}(\mathtt{w}), Z^u_{\mathscr{A}^*}(\mathtt{w}) \oplus Z^u_{\mathscr{B}^*}(\mathtt{w}))$ to each of her wires $\mathtt{w} \in \mathcal{A}_0$.
3. $\mathscr{B}^*$ applies the decryption operator for key $K_{\mathscr{B}^*}(\mathtt{w}) = (X^u_{\mathscr{A}^*}(\mathtt{w}) \oplus X^u_{\mathscr{B}^*}(\mathtt{w}), Z^u_{\mathscr{A}^*}(\mathtt{w}) \oplus Z^u_{\mathscr{B}^*}(\mathtt{w}))$ to each of his wires $\mathtt{w} \in \mathcal{B}_0$.

In the following subsections 5.1 to 5.3, we describe the evaluation phase for each gate in $\mathcal{UG}$.

Swapping for key-releasing. Notice that the key-releasing phase only uses the SWAP-gate with classical input states. The reader might therefore wonder why this functionality is defined quantumly when a classical swap would work equally well. The reason is that, perhaps somewhat surprisingly, a classical swap is a potentially stronger primitive than a quantum swap. From a classical swap one can build a quantum swap by encrypting the quantum states with classical keys, exchange the encrypted states using quantum communication, and then using the classical swap to exchange the keys. Obtaining a classical swap from a quantum one, however, is not obvious. Suppose that registers $\mathcal{A}$ and $\mathcal{B}$ should be swapped classically while holding quantum states beforehand. These registers could be entangled with some purification registers before being swapped. Using a quantum swap between $\mathcal{A}$ and $\mathcal{B}$ will always leave these registers entangled with the purification registers until they become measured while a classical swap will ensure that $\mathcal{A}$ and $\mathcal{B}$ become unentangled with the purification registers after its invocation. In other words, a classical swap could prevent an adversary from exploiting entanglement in his attack.

The ideal AND-box functionality. As we are going to see next, a call to an ideal AND-box is required during the evaluation of the R-gate. Unlike the ideal SWAP used for key-releasing, the AND-box will be modeled by a purely classical primitive denoted AND-BOX. This is required for privacy of our protocol since

[5] Otherwise, add enough registers initially in state $|0\rangle$ to the smaller set.

any implementation of it by some unitary will necessarily leak[21]. The quantum operation implementing it will first measure the two one-qubit input registers in the computational basis in order to get classical inputs $x, y \in \{0,1\}$ for $\mathscr{A}^*$ and $\mathscr{B}^*$ respectively. The classical output bits are then set to $a \in_R \{0,1\}$ for $\mathscr{A}^*$ and $b = a \oplus xy$ for $\mathscr{B}^*$.

5.1 Computing over Encrypted States

Before the execution of G_{j+1} in C_U, $\mathscr{A}^*$ and $\mathscr{B}^*$ share an encryption of $\rho_j = ((G_j \cdot G_{j-1} \cdot \ldots \cdot G_1) \otimes \mathbb{1}_{\mathcal{R}}) \cdot \rho_{\text{in}}$ in registers[6] holding wires $\mathtt{w} \in \mathcal{A}_0 \cup \mathcal{B}_0$. Each wire $\mathtt{w} \in \mathcal{A}_0 \cup \mathcal{B}_0$ is encrypted by a shared quantum one-time pad as

$$\left(\left(\bigotimes_{\mathtt{w} \in \mathcal{A}_0 \cup \mathcal{B}_0} X^{X^j_{\mathscr{A}^*}(\mathtt{w}) \oplus X^j_{\mathscr{B}^*}(\mathtt{w})} Z^{Z^j_{\mathscr{A}^*}(\mathtt{w}) \oplus Z^j_{\mathscr{B}^*}(\mathtt{w})} \right) \otimes \mathbb{1}_{\mathcal{R}} \right) \cdot \rho_j \ , \tag{2}$$

where $K^j_{\mathscr{A}^*}(\mathtt{w}) := (X^j_{\mathscr{A}^*}(\mathtt{w}), Z^j_{\mathscr{A}^*}(\mathtt{w})) \in \{0,1\}^2$ and $K^j_{\mathscr{B}^*}(\mathtt{w}) := (X^j_{\mathscr{B}^*}(\mathtt{w}), Z^j_{\mathscr{B}^*}(\mathtt{w})) \in \{0,1\}^2$ are two bits of secret keys for $\mathscr{A}^*$ and $\mathscr{B}^*$ respectively. In other words, wires $\mathtt{w} \in \mathcal{A}_0 \cup \mathcal{B}_0$ are encrypted by $X^x Z^z$ where $x = X^j_{\mathscr{A}^*}(\mathtt{w}) \oplus X^j_{\mathscr{B}^*}(\mathtt{w})$ and $z = Z^j_{\mathscr{A}^*}(\mathtt{w}) \oplus Z^j_{\mathscr{B}^*}(\mathtt{w})$ are additive sharings for the encryption of $\mathtt{w}$. Then, evaluating G_{j+1} upon state (2) will produce a new sharing $K^{j+1}_A(\mathtt{w}) := (X^{j+1}_A(\mathtt{w}), Z^{j+1}_A(\mathtt{w}))$ and $K^{j+1}_B(\mathtt{w}) := (X^{j+1}_B(\mathtt{w}), Z^{j+1}_B(\mathtt{w}))$ for the encryption of state $\rho_{j+1} = (G_{j+1} \otimes \mathbb{1}_{\mathcal{R}}) \cdot \rho_j$. In the following, we describe how to update the keys for the wires involved in the current gate to be evaluated—all other wires retain their previous values.

5.2 Evaluation of Gates in the Pauli and Clifford Groups

Pauli gates. Non-trivial Pauli gates (i.e., X, Y, and Z) can easily be computed on encrypted quantum states since they commute or anti-commute pairwise. Let $G_{j+1} \in \{X, Y, Z\}$ be the Pauli gate to be executed on wire $\mathtt{w}$. We have:

$$G_{j+1} \left(X^{X^i_{\mathscr{A}^*}(\mathtt{w}) \oplus X^j_{\mathscr{B}^*}(\mathtt{w})} Z^{Z^j_{\mathscr{A}^*}(\mathtt{w}) \oplus Z^j_{\mathscr{B}^*}(\mathtt{w})} \right) = \pm \left(X^{X^j_{\mathscr{A}^*}(\mathtt{w}) \oplus X^j_{\mathscr{B}^*}(\mathtt{w})} Z^{Z^j_{\mathscr{A}^*}(\mathtt{w}) \oplus Z^j_{\mathscr{B}^*}(\mathtt{w})} \right) G_{j+1} \ .$$

It means that up to an irrelevant phase factor, it suffices for the owner of $\mathtt{w}$ to apply G_{j+1} without the need for neither party to update their shared keys, i.e., $K^{j+1}_{\mathscr{A}^*}(\mathtt{w}) := K^j_{\mathscr{A}^*}(\mathtt{w})$ and $K^{j+1}_{\mathscr{B}^*}(\mathtt{w}) := K^j_{\mathscr{B}^*}(\mathtt{w})$.

H, P, and CNOT on local wires. Now, suppose that $G_{j+1} \in \{\mathsf{H}, \mathsf{P}\}$. Each of these one-qubit gates applied upon wire $\mathtt{w}$ will be computed by simply letting the party owning $\mathtt{w}$ apply G_{j+1}. Since

$$\mathsf{H} \left(X^{X^j_{\mathscr{A}^*}(\mathtt{w}) \oplus X^j_{\mathscr{B}^*}(\mathtt{w})} Z^{Z^j_{\mathscr{A}^*}(\mathtt{w}) \oplus Z^j_{\mathscr{B}^*}(\mathtt{w})} \right) = \left(X^{Z^j_{\mathscr{A}^*}(\mathtt{w}) \oplus Z^j_{\mathscr{B}^*}(\mathtt{w})} Z^{X^j_{\mathscr{A}^*}(\mathtt{w}) \oplus X^j_{\mathscr{B}^*}(\mathtt{w})} \right) \mathsf{H} \ , \text{ and}$$

[6] To ease the notation in the following, we assume $\rho_j \in \mathrm{D}(\mathcal{A}_0 \otimes \mathcal{B}_0)$ rather than in $\mathrm{D}(\mathcal{A}_0 \otimes \mathcal{B}_0 \otimes \mathcal{R})$. It is easy to see that this can be done without loss of generality.

$$\mathsf{P}\left(X^{X^j_{\mathscr{A}^*}(\mathtt{w})\oplus X^j_{\mathscr{B}^*}(\mathtt{w})}Z^{Z^j_{\mathscr{A}^*}(\mathtt{w})\oplus Z^j_{\mathscr{B}^*}(\mathtt{w})}\right) = \left(X^{X^j_{\mathscr{A}^*}(\mathtt{w})\oplus X^j_{\mathscr{B}^*}(\mathtt{w})}Z^{X^j_{\mathscr{A}^*}(\mathtt{w})\oplus X^j_{\mathscr{B}^*}(\mathtt{w})\oplus Z^j_{\mathscr{A}^*}(\mathtt{w})\oplus Z^j_{\mathscr{B}^*}(\mathtt{w})}\right)\mathsf{P}\ ,$$

the encryption keys are updated as follows:

$$\begin{aligned}
\mathsf{H}: K^{j+1}_{\mathscr{A}^*} &= (X^{j+1}_{\mathscr{A}^*}(\mathtt{w}), Z^{j+1}_{\mathscr{A}^*}(\mathtt{w})) := (Z^i_{\mathscr{A}^*}(\mathtt{w}), X^j_{\mathscr{A}^*}(\mathtt{w}))\ ,\\
K^{j+1}_{\mathscr{B}^*} &= (X^{j+1}_{\mathscr{B}^*}(\mathtt{w}), Z^{j+1}_{\mathscr{B}^*}(\mathtt{w})) := (Z^j_{\mathscr{B}^*}(\mathtt{w}), X^j_{\mathscr{B}^*}(\mathtt{w}))\ ,\\
\mathsf{P}: K^{j+1}_{\mathscr{A}^*} &= (X^{j+1}_{\mathscr{A}^*}(\mathtt{w}), Z^{j+1}_{\mathscr{A}^*}(\mathtt{w})) := (X^j_{\mathscr{A}^*}(\mathtt{w}), X^j_{\mathscr{A}^*}(\mathtt{w})\oplus Z^j_{\mathscr{A}^*}(\mathtt{w}))\ ,\\
K^{j+1}_{\mathscr{B}^*} &= (X^{j+1}_{\mathscr{B}^*}(\mathtt{w}), Z^{j+1}_{\mathscr{B}^*}(\mathtt{w})) := (X^j_{\mathscr{B}^*}(\mathtt{w}), X^j_{\mathscr{B}^*}(\mathtt{w})\oplus Z^j_{\mathscr{B}^*}(\mathtt{w}))\ .
\end{aligned}$$

Any one-qubit gate in the Clifford group can be implemented the same way using their own commutation relations with the Pauli operators used for encryption. A CNOT-gate on local wires can be evaluated in a similar way. That is, whenever both wires $\mathtt{w}$ and $\mathtt{w}'$ feeding the CNOT belong to the same party. Assume that $\mathtt{w}$ is the control wire while $\mathtt{w}'$ is the target and that $\mathscr{A}^*$ holds them both(i.e., $\mathtt{w}, \mathtt{w}' \in \mathcal{A}_0$). Then, $\mathscr{A}^*$ simply applies CNOT on wires $\mathtt{w}$ and $\mathtt{w}'$. Encryption keys are updated as:

$$\begin{aligned}
\mathsf{CNOT}: K^{j+1}_{\mathscr{A}^*}(\mathtt{w}) &= (X^{j+1}_{\mathscr{A}^*}(\mathtt{w}), Z^{j+1}_{\mathscr{A}^*}(\mathtt{w})) := (X^j_{\mathscr{A}^*}(\mathtt{w}), Z^j_{\mathscr{A}^*}(\mathtt{w})\oplus Z^j_{\mathscr{A}^*}(\mathtt{w}'))\ ,\\
K^{j+1}_{\mathscr{A}^*}(\mathtt{w}') &= (X^{j+1}_{\mathscr{A}^*}(\mathtt{w}'), Z^{j+1}_{\mathscr{A}^*}(\mathtt{w}')) := (X^j_{\mathscr{A}^*}(\mathtt{w}')\oplus X^j_{\mathscr{A}^*}(\mathtt{w}), Z^j_{\mathscr{A}^*}(\mathtt{w}'))\ ,\\
K^{j+1}_{\mathscr{B}^*}(\mathtt{w}) &:= K^j_{\mathscr{B}^*}(\mathtt{w}) \text{ and } K^{j+1}_{\mathscr{B}^*}(\mathtt{w}') := K^j_{\mathscr{B}^*}(\mathtt{w}')\ .
\end{aligned}$$

When $\mathscr{B}^*$ holds both wires, the procedure is simply performed with the roles of $\mathscr{A}^*$ and $\mathscr{B}^*$ reversed.

Nonlocal CNOT. We now look at the case where $G_{j+1} = \mathsf{CNOT}$ upon wires $\mathtt{w}$ and $\mathtt{w}'$, one of which is owned by $\mathscr{A}^*$ while the other is owned by $\mathscr{B}^*$. In this case, interaction is unavoidable for the evaluation of the gate. Let us assume w.l.g that $\mathscr{A}^*$ holds the control wire $\mathtt{w}$ while $\mathscr{B}^*$ holds the target wire $\mathtt{w}'$ (i.e., $\mathtt{w} \in \mathcal{A}_0$ and $\mathtt{w}' \in \mathcal{B}_0$). We start from a construction introduced in [11] in the context of fault tolerant quantum computation.

The idea behind the sub-protocol is depicted in Fig. 3. The effect of the Bell measurement is to *teleport* the input state of wires $\mathtt{w}$ and $\mathtt{w}'$ *through* the CNOT-gate[11]. The input to the CNOT appearing in the circuit of Fig. 3 is independent of both input wires $\mathtt{w}$ and $\mathtt{w}'$ (they are just two half EPR-pairs).

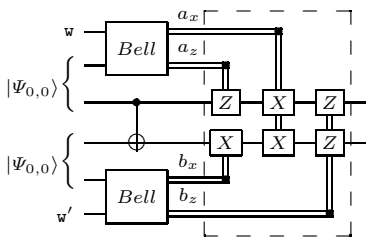


Fig. 3. Evaluation of CNOT.

The sub-protocol for the evaluation of CNOT simply consists in executing the circuit of Fig. 3 without the decryption part (i.e., the part inside the dotted rectangle). The state $|\xi\rangle := (\mathbb{1}_{\mathtt{A}} \otimes CNOT \otimes \mathbb{1}_{\mathtt{B}})|\Psi_{0,0}\rangle|\Psi_{0,0}\rangle$ can be prepared by one party. We let the holder of the *control wire* (i.e., $\mathscr{A}^*$ in Fig. 3) prepare $|\xi\rangle$ before sending its two rightmost registers to the other party. The decryption in the dotted-rectangle

is used to update the encryption keys according to the measurement outcomes (a_x, a_z, b_x, b_z):

$$\begin{aligned}
\mathsf{CNOT}: K^{j+1}_{\mathscr{A}^*}(\mathtt{w}) &:= (X^j_{\mathscr{A}^*}(\mathtt{w}) \oplus a_x, Z^j_{\mathscr{A}^*}(\mathtt{w}) \oplus a_z) \ , \\
K^{j+1}_{\mathscr{B}^*}(\mathtt{w}) &:= (X^i_{\mathscr{B}^*}(\mathtt{w}), Z^j_{\mathscr{B}^*}(\mathtt{w}) \oplus b_z) \ , \\
K^{j+1}_{\mathscr{A}^*}(\mathtt{w}') &:= (X^j_{\mathscr{A}^*}(\mathtt{w}') \oplus a_x, Z^j_{\mathscr{A}^*}(\mathtt{w}')) \ , \\
K^{j+1}_{\mathscr{B}^*}(\mathtt{w}') &:= (X^j_{\mathscr{B}^*}(\mathtt{w}') \oplus b_x, Z^j_{\mathscr{B}^*}(\mathtt{w}') \oplus b_z) \ .
\end{aligned}$$

As for all previous gates, the key updating phase is performed locally without the need for communication.

5.3 Evaluation of the R-Gate

The only gate left in $\mathcal{UG}$ is $G_{j+1} := \mathsf{R}$. We assume without loss of generality that $\mathscr{A}^*$ owns wire $\mathtt{w}$ upon which R is applied (i.e., $\mathtt{w} \in \mathcal{A}_0$). The subprotocol needs a call to an ideal AND-BOX in order to guarantee privacy during the key updating process. Observe first that the R-gate commutes with Pauli encryption operator Z. It means that applying the R-gate upon a state encrypted with Z produces the correct output state still encrypted with Z. However, the equality $\mathsf{R}\cdot X = e^{-i\pi/4} Y \mathsf{P}\cdot\mathsf{R}$ tells us that a P-gate should be applied for the decryption of the output when the input has been encrypted using X. This breaks the invariant that wires after each gate are all encrypted by Pauli operators. We remove the P-gate by converting it into a sequence of Pauli operators.

Suppose $\mathscr{A}$'s wire $\mathtt{w}$ is encrypted as usual by shared keys $K^j_{\mathscr{A}^*}(\mathtt{w}) := (X^j_{\mathscr{A}^*}(\mathtt{w}), Z^j_{\mathscr{A}^*}(\mathtt{w}))$, and $K^j_{\mathscr{B}^*}(\mathtt{w}) := (X^j_{\mathscr{B}^*}(\mathtt{w}), Z^j_{\mathscr{B}^*}(\mathtt{w}))$. Ignoring an irrelevant global phase, the result of applying R on wire $\mathtt{w}$ is

$$\begin{aligned}
&\mathsf{R} Z^{Z^j_{\mathscr{A}^*}(\mathtt{w}) \oplus Z^i_{\mathscr{B}^*}(\mathtt{w})} X^{X^j_{\mathscr{A}^*}(\mathtt{w}) \oplus X^j_{\mathscr{B}^*}(\mathtt{w})} = \\
&\quad Z^{Z^j_{\mathscr{A}^*}(\mathtt{w}) \oplus Z^j_{\mathscr{B}^*}(\mathtt{w}) \oplus X^j_{\mathscr{A}^*}(\mathtt{w}) \oplus X^j_{\mathscr{B}^*}(\mathtt{w})} X^{X^j_{\mathscr{A}^*}(\mathtt{w}) \oplus X^j_{\mathscr{B}^*}(\mathtt{w})} \mathsf{P}^{X^j_{\mathscr{A}^*}(\mathtt{w}) \oplus X^j_{\mathscr{B}^*}(\mathtt{w})} \mathsf{R} \ ,
\end{aligned} \tag{3}$$

To remove the P-gate, we let each party remove his part of $\mathsf{P}^{X^j_{\mathscr{A}^*}(\mathtt{w}) \oplus X^j_{\mathscr{B}^*}(\mathtt{w})}$ in a private interactive process. To do this, $\mathscr{A}^*$ picks random bits r and r', and $\mathscr{B}^*$ picks random bits s and s'. $\mathscr{A}^*$ applies the operator $X^r Z^{r'} P^{X^i_{\mathscr{A}^*}(\mathtt{w})}$ and sends the resulting quantum state to $\mathscr{B}^*$. $\mathscr{B}^*$ applies the operator $X^s Z^{s'} P^{X^j_{\mathscr{B}^*}(\mathtt{w})}$ and sends the result back to $\mathscr{A}^*$. The resulting protocol is shown in Fig. 4. It starts with $\mathscr{A}^*$ applying R upon the encrypted state before the one-round interactive process described above starts.

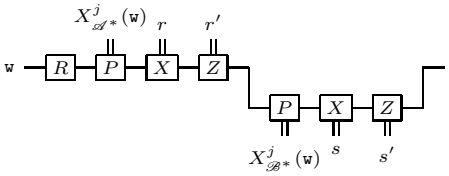


Fig. 4. Implementation of the R-gate.

After $\mathscr{A}^*$'s application of R, the resulting state is as described on the right-hand side of (3). At the end of the process (i.e., circuit of Fig. 4), the encryption

becomes:

$$Z^{s'}X^{s}\mathsf{P}^{X^{j}_{\mathscr{B}^*}(\mathtt{w})}Z^{r'}X^{r}\mathsf{P}^{X^{j}_{\mathscr{A}^*}(\mathtt{w})} \\ Z^{Z^{j}_{\mathscr{A}^*}(\mathtt{w})\oplus Z^{j}_{\mathscr{B}^*}(\mathtt{w})\oplus X^{j}_{\mathscr{A}^*}(\mathtt{w})\oplus X^{j}_{\mathscr{B}^*}(\mathtt{w})}X^{X^{j}_{\mathscr{A}^*}(\mathtt{w})\oplus X^{j}_{\mathscr{B}^*}(\mathtt{w})}\mathsf{P}^{X^{j}_{\mathscr{A}^*}(\mathtt{w})\oplus X^{j}_{\mathscr{B}^*}(\mathtt{w})} \ . \tag{4}$$

Since Z and P commute and $\mathsf{P}\cdot X = XZ\cdot\mathsf{P}$, we can re-write (4) (i.e., up to an irrelevant phase factor) as

$$Z^{s'\oplus r'\oplus r\cdot X^{j}_{\mathscr{B}^*}(\mathtt{w})}X^{s\oplus r}\mathsf{P}^{X^{j}_{\mathscr{A}^*}(\mathtt{w})+X^{j}_{\mathscr{B}^*}(\mathtt{w})} \\ Z^{Z^{j}_{\mathscr{A}^*}(\mathtt{w})\oplus Z^{j}_{\mathscr{B}^*}(\mathtt{w})\oplus X^{j}_{\mathscr{A}^*}(\mathtt{w})\oplus X^{j}_{\mathscr{B}^*}(\mathtt{w})}X^{X^{j}_{\mathscr{A}^*}(\mathtt{w})\oplus X^{j}_{\mathscr{B}^*}(\mathtt{w})X^{j}_{\mathscr{A}^*}(\mathtt{w})\oplus X^{j}_{\mathscr{B}^*}(\mathtt{w})} \ .$$

Using the fact that for $a,b\in\{0,1\}$, $\mathsf{P}^{a+b}=Z^{ab}\mathsf{P}^{a\oplus b}$, the previous equation can be re-written as

$$Z^{s'\oplus r'\oplus Z^{j}_{\mathscr{A}^*}(\mathtt{w})\oplus Z^{j}_{\mathscr{B}^*}(\mathtt{w})\oplus X_{\mathscr{A}^*}(\mathtt{w})\oplus X_{\mathscr{B}^*}(\mathtt{w})\oplus(r\oplus X^{j}_{\mathscr{A}^*}(\mathtt{w}))\cdot X^{j}_{\mathscr{B}^*}(\mathtt{w})} \\ X^{s\oplus r}\mathsf{P}^{X^{j}_{\mathscr{A}^*}(\mathtt{w})\oplus X^{j}_{\mathscr{B}^*}(\mathtt{w})}X^{X^{j}_{\mathscr{A}^*}(\mathtt{w})\oplus X^{j}_{\mathscr{B}^*}(\mathtt{w})}\mathsf{P}^{X^{j}_{\mathscr{A}^*}(\mathtt{w})\oplus X^{j}_{\mathscr{B}^*}(\mathtt{w})} . \tag{5}$$

Moving the leftmost P-gate to the right results in Pauli encryption,

$$Z^{s'\oplus r'\oplus X^{j}_{\mathscr{A}^*}(\mathtt{w})\oplus X^{j}_{\mathscr{B}^*}(\mathtt{w})\oplus Z^{j}_{\mathscr{A}^*}(\mathtt{w})\oplus Z^{j}_{\mathscr{B}^*}(\mathtt{w})\oplus(r\oplus X^{j}_{\mathscr{A}^*}(\mathtt{w}))\cdot X^{j}_{\mathscr{B}^*}(\mathtt{w})} \\ X^{s\oplus r\oplus X^{j}_{\mathscr{A}^*}(\mathtt{w})\oplus X^{j}_{\mathscr{B}^*}(\mathtt{w})} \ . \tag{6}$$

Encryption (6) is not a proper additive sharing since the Z-operator depends on $(r\oplus X^{j}_{\mathscr{A}^*}(\mathtt{w}))\cdot X^{j}_{\mathscr{B}^*}(\mathtt{w})$; the logical AND between a value known only by $\mathscr{A}^*$ (i.e., $r\oplus X^{j}_{\mathscr{A}^*}(\mathtt{w})$) and a value known only by $\mathscr{B}^*$ (i.e., $X^{j}_{\mathscr{B}^*}(\mathtt{w})$). To get back to an additive sharing, $\mathscr{A}^*$ and $\mathscr{B}^*$ can simply call the AND-BOX once with inputs $r\oplus X^{j}_{\mathscr{A}^*}(\mathtt{w})$ and $X^{j}_{\mathscr{B}^*}(\mathtt{w})$ respectively as depicted in Fig. 5. After this, $\mathscr{A}^*$ and $\mathscr{B}^*$ share a proper encryption of the resulting state. The new encryption key for $\mathscr{A}^*$'s wire $\mathtt{w}$ becomes:

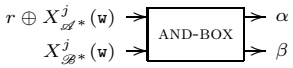


Fig. 5. $\alpha\oplus\beta = (r\oplus X^{j}_{\mathscr{A}^*}(\mathtt{w}))\cdot X^{j}_{\mathscr{B}^*}(\mathtt{w})$ from an AND-BOX.

$$\mathsf{R}: K^{j+1}_{\mathscr{A}^*}(\mathtt{w}) := (r\oplus X^{j}_{\mathscr{A}^*}(\mathtt{w}), r'\oplus\alpha\oplus Z^{j}_{\mathscr{A}^*}(\mathtt{w})\oplus X^{j}_{\mathscr{A}^*}(\mathtt{w})) \ , \\ K^{j+1}_{\mathscr{B}^*}(\mathtt{w}) := (s\oplus X^{j}_{\mathscr{B}^*}(\mathtt{w}), s'\oplus\beta\oplus Z^{j}_{\mathscr{B}^*}(\mathtt{w})\oplus X^{j}_{\mathscr{B}^*}(\mathtt{w})) \ .$$

5.4 On the Necessity of Swapping Privately

One may ask whether relying upon SWAP is necessary for the protocol to be private against specious adversaries. For instance, what would happen if one party announces the encryption keys before the other party? We now show that as soon as one party gets the other party's decryption key before having announced its own, a specious adversary can break privacy.

Consider the protocol for a quantum circuit made out of one single CNOT-gate. Suppose that $\mathscr{A}^*$ holds the control wire $\mathtt{w}$ while $\mathscr{B}^*$ holds the target wire $\mathtt{w}'$.

Suppose also the key-releasing phase first asks $\mathscr{B}^*$ to announce the encryption keys $K_{\mathscr{B}^*}(\mathtt{w})$ before $\mathscr{A}^*$ announces $K_{\mathscr{A}^*}(\mathtt{w}')$. Suppose $\tilde{\mathscr{A}}$'s input state is $|0\rangle$.

The adversary $\tilde{\mathscr{A}}$ can now act as follows. $\tilde{\mathscr{A}}$ runs the protocol for CNOT without performing the Bell measurement until she receives the encryption key b_z from $\mathscr{B}^*$. Clearly, $\tilde{\mathscr{A}}$'s behavior is specious up to that point since she could re-produce the honest state by just applying the Bell measurement on her input state stored in register $\mathcal{A}_0$. However, given b_z she could also in principle compute the CNOT upon any input state of her choice. This means that the state she holds after b_z has been announced and before applying her Bell measurement contains information about $\mathscr{B}^*$'s input. On the one hand, when $\tilde{\mathscr{A}}$'s input state is $|0\rangle$ no information whatsoever on $\mathscr{B}^*$'s input state should be available to her (i.e., in this case CNOT behaves like the identity). On the other hand, had her input state been $|-\rangle$, information about $\mathscr{B}^*$'s state would have become available since the control and target wires exchange their roles when the input states are in the Hadamard basis. However, when $\tilde{\mathscr{A}}$'s input state is $|0\rangle$, any simulation of her view can only call the ideal functionality with input state $|0\rangle$. It follows that no simulator can reproduce $\tilde{\mathscr{A}}$'s state right after the announcement of b_z.

6 Main Result and Open Questions

Putting Lemma E.1 and Lemma E.2 together gives the desired result:

Theorem 6.1 (Main Result). *Protocol $P_U^{\mathscr{O}}$ is statistically private against any statistically specious quantum adversary and for any $U \in \mathrm{U}(\mathcal{A}_0 \otimes \mathcal{B}_0)$. If U is in the Clifford group then the only non-trivial oracle call in $\mathscr{O}$ is one call to an ideal* SWAP*. If U is not in the Clifford group then $\mathscr{O}$ contains an additional oracle call to* AND-BOX *for each* R*-gate in the circuit for U.*

It should be mentioned that it is not too difficult to modify our protocol in order to privately evaluate quantum operations rather than only unitary transforms. Classical two party computation together with the fact that quantum operations can be viewed as unitaries acting in larger spaces can be used to achieve this extra functionality. Privacy can be preserved by keeping these extra registers encrypted after the execution of the protocol. We leave this discussion to the full version of the paper.

A few interesting questions remain open:

- It would be interesting to know whether there exists a unitary transform that can act as a universal primitive for private two-party evaluation of unitaries. This would allow to determine whether classical cryptographic assumptions are required for this task.
- Finally, is there a way to compile quantum protocols secure against specious adversaries into protocols secure against arbitrary quantum adversaries? An affirmative answer would allow to simplify greatly the design of quantum protocols. Are extra assumptions needed to preserve privacy against any adversary?

7 Acknowledgements

The authors would like to thank the referees for their comments and suggestions. We would also like to thank Thomas Pedersen for numerous helpful discussions in the early stage of this work.

References

1. *Physical Review Letters*, volume 78, April 1997.
2. D. Aharonov and M. Ben-Or. Fault-tolerant quantum computation with constant error. In *29th Annual ACM Symposium on Theory of Computing (STOC)*, pages 176–188, 1997.
3. Andris Ambainis, Michele Mosca, Alain Tapp, and Ronald de Wolf. Private quantum channels. In *41st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 547–553, 2000.
4. Michael Ben-Or, Claude Crépeau, Daniel Gottesman, Avinatan Hassidim, and Adam Smith. Secure multiparty quantum computation with (only) a strict honest majority. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 249–260, 2006.
5. Charles H. Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres, and William K. Wootters. Teleporting an unknown quantum state via dual classical and einstein-podolsky-rosen channels. *Physical Review Letters*, 68(21):1895–1899, March 1993.
6. Anne Broadbent, Joseph Fitzsimons, and Elham Kashefi. Universal blind quantum computation, December 2009. available at http://arxiv.org/abs/0807.4154.
7. Ran Canetti. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology*, 13(1):143–202, 2000.
8. Claude Crépeau, Daniel Gottesman, and Adam Smith. Secure multi-party quantum computation. In *34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 643–652, 2002.
9. Ivan B. Damgård, Serge Fehr, Carolin Lunemann, Louis Salvail, and Christian Schaffner. Improving the security of quantum protocols via commit-and-open. In *Advances in Cryptology—CRYPTO '09*, volume 5677 of *Lecture Notes in Computer Science*, pages 408–427. Springer, 2009. Full version available at: http://arxiv.org/abs/0902.3918.
10. Daniel Gottesman and Isaac L. Chuang. Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations. *Nature*, 402:390–393, November 1999.
11. Daniel Gottesman and Isaac L. Chuang. Quantum teleportation is a universal computational primitive. http://arxiv.org/abs/quant-ph/9908010, August 1999.
12. G. Gutoski and J. Watrous. Quantum interactive proofs with competing provers. In *22nd Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 3404 of *Lecture Notes in Computer Science*, pages 605–616. Springer, March 2005.
13. Joe Kilian. Founding cryptography on oblivious transfer. In *20th Annual ACM Symposium on Theory of Computing (STOC)*, pages 20–31, 1988.
14. Hoi-Kwong Lo. Insecurity of quantum secure computations. *Physical Review A*, 56(2):1154–1162, 1997.

15. Hoi-Kwong Lo and Hoi Fung Chau. Is quantum bit commitment really possible? In *Physical Review Letters* [1], pages 3410–3413.
16. Dominic Mayers. Unconditionally secure quantum bit commitment is impossible. In *Physical Review Letters* [1], pages 3414–3417.
17. Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge university press, 2000.
18. Sandu Popescu and Daniel Rohrlich. Quantum nonlocality as an axiom. *Foundations of Physics*, 24(3):379–385, 1994.
19. Sandu Popescu and Daniel Rohrlich. Causality and nonlocality as axioms for quantum mechanics. In *symposium on Causality and Locality in Modern Physics and Astronomy: Open Questions and Possible Solutions*, 1997. http://arxiv.org/abs/quant-ph/9709026.
20. Renato Renner and Robert König. Universally composable privacy amplification against quantum adversaries. In *Theory of Cryptography Conference (TCC)*, volume 3378 of *Lecture Notes in Computer Science*, pages 407–425. Springer, 2005.
21. Louis Salvail, Miroslava Sotáková, and Christian Schaffner. On the power of two-party quantum cryptography. In *Advances in Cryptology—ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 70–87. Springer, 2009.
22. Peter W. Shor. Fault-tolerant quantum computation. In *37th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 56–65, 1996.
23. Adam Smith. Techniques for secure distributed computing with quantum data. Presented at the Field's institute Quantum Cryptography and Computing workshop, October, 2006.
24. John Watrous. Limits on the power of quantum statistical zero-knowledge. In *43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 459–468, 2002.
25. Stefan Wolf and Jürg Wullschleger. Oblivious transfer and quantum non-locality. In *International Symposium on Information Theory (ISIT 2005)*, pages 1745–1748, 2005.
26. Andrew Yao. How to generate and exchange secrets. In *27th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 1986.

A Commutations Rules

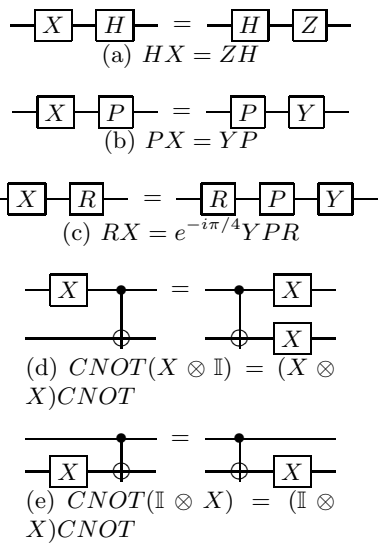
$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \ , \ \ Y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \ , \ \ Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \ ,$$

$$P = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \ , \ \ H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \ , \ \ R = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix} \ ,$$

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \ .$$

B Classical Definition of a Specious Adversary

In this section we briefly discuss the definition of an specious adversary and the definition of security against such an adversary, and we compare it to the notion of a semi-honest classical adversary to illustrate the difference.



(a) $HX = ZH$

(b) $PX = YP$

(c) $RX = e^{-i\pi/4}YPR$

(d) $CNOT(X \otimes \mathbb{I}) = (X \otimes X)CNOT$

(e) $CNOT(\mathbb{I} \otimes X) = (\mathbb{I} \otimes X)CNOT$

Fig. 6. Commutation relations for X.

B.1 Specious Adversary

As usual we let an n-party function $(y_1, \ldots, y_n) = f(x_1, \ldots, x_n)$ define n functions $y_i = f_i(x_1, \ldots, x_n)$.

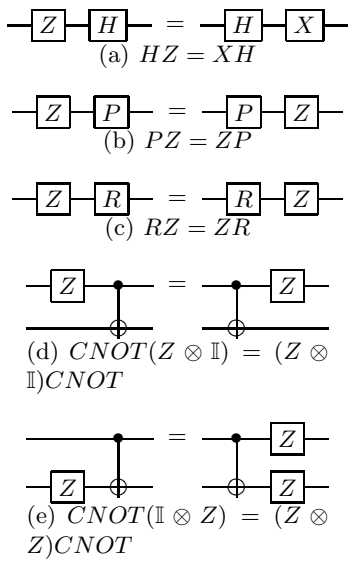
For our purpose, an n-party protocol $\pi = (\pi_1, \ldots, \pi_n)$ consists of n parties π_i connected by secure channels. If the protocol is for the h-hybrid model, for an n-party function h, there are additionally some designated rounds where each π_i must specify an input a_i to h. Then $(b_1, \ldots, b_n) = h(a_1, \ldots, a_n)$ is computed and each π_i is given back b_i. A receiving point in a protocol is a point where the parties just exchanged messages or just received outputs b_i from h.

For an n-party protocol π and for $H \subset \{1, \ldots, n\}$ we denote by π_H the set $\{\pi_i\}_{i \in H}$ of parties indexed by $i \in H$.

For an n-party protocol π and for $C \subset \{1, \ldots, n\}$ we denote by $\tilde{\pi}_C$ an adversary for π acting on behalf of parties indexed by $i \in C$. It receives the inputs, randomness and messages of all parties indexed by $i \in C$ and decides what messages they should send. By$(\pi_{\bar{C}}, \tilde{\pi}_C)$ we mean the protocol consisting of the parties π_i, $i \notin C$, running with the adversary $\tilde{\pi}_C$.

We use the following notation for vectors. We sometimes identify a vector $v = (v_1, \ldots, v_n)$ with the set $\{(i, v_i)\}_{i \in \{1,\ldots,n\}}$. For $S \subset \{1, \ldots, n\}$ we let v_S be the vector v restricted to indices in S, formally $v_S = \{(i, v_i)\}_{i \in S}$. For $S_1, S_2 \subset \{1, \ldots, n\}$ with $S_1 \cap S_2 = \emptyset$ we let $(v_{S_1}, v_{S_2}) = v_{S_1} \cup v_{S_2}$.

Definition B.1 (execution of (corrupted) protocol). *For an n-party protocol π and input $x = (x_1, \ldots, x_n)$, the distribution $\pi(x)$ is defined as follows: sample $r = (r_1, \ldots, r_n)$ uniformly at random. Run π on input x and randomness r. Let $y = (y_1, \ldots, y_n)$, where y_i is the output of party π_i, and let $\pi(x) = (x, y)$. For*



(a) $HZ = XH$

(b) $PZ = ZP$

(c) $RZ = ZR$

(d) $CNOT(Z \otimes \mathbb{I}) = (Z \otimes \mathbb{I})CNOT$

(e) $CNOT(\mathbb{I} \otimes Z) = (Z \otimes Z)CNOT$

Fig. 7. Commutation relations for Z.

an n-party protocol π, input $(x_1, \ldots, x_n)$, subset $C \subset \{1, \ldots, n\}$, adversary $\tilde{\pi}_C$ and $\tilde{\pi} = (\pi_{\bar{C}}, \tilde{\pi}_C)$, the distribution $\tilde{\pi}(x)$ is defined as follows: Sample $\tilde{r}_i$, $i \in C$, uniformly at random. Sample r_i, $i \notin C$, uniformly at random. Let $\tilde{\pi} = (\pi_{\bar{C}}, \tilde{\pi}_C)$. Run $\tilde{\pi}$ on input x and randomness $(r_{\bar{C}}, \tilde{r}_C)$. Let $\tilde{y}_C$ be the output of the adversary, let $y_{\bar{C}}$ be the outputs of the parties $\pi_{\bar{C}}$, and let $\tilde{\pi}(x) = (x, (y_{\bar{C}}, \tilde{y}_C))$.

Definition B.2 (specious adversary). *Let π be an n-party protocol, let $C \subset \{1, \ldots, n\}$, let $\tilde{\pi}_C$ be an adversary, let $\tilde{\pi} = (\pi_{\bar{C}}, \tilde{\pi}_C)$. We say that $\tilde{\pi}_C$ is specious in π if there exists a poly-time view simulator V such that for all inputs $x = (x_1, \ldots, x_n)$ and for all receiving points p in $\tilde{\pi}$ it holds that $D^{(p)}$ and $\tilde{D}^{(p)}$ have the same distribution, where the distribution $D^{(p)}$ is defined as follows: sample $r = (r_1, \ldots, r_n)$ uniformly at random. Run π on input x and randomness r until receiving point p. Let $M = (M_1, \ldots, M_n)$, where M_i is the messages sent and received by party π_i, and let $D^{(p)} = (x, r, M)$. The distribution $\tilde{D}^{(p)}$ is defined as follows: Sample r_i, $i \notin C$, uniformly at random. Sample $\tilde{r}_i$, $i \in C$, uniformly at random. Run $\tilde{\pi}$ on input x and randomness $(r_{\bar{C}}, \tilde{r}_C)$ until receiving point p. Let $\tilde{M}_C$ be the messages sent and received by the adversary $\tilde{\pi}_C$, let $M_{\bar{C}}$ be the messages sent and received by parties $\pi_{\bar{C}}$, let $(r_C, M_C) = V(p, x_C, \tilde{r}_C, \tilde{M}_C)$, and let $\tilde{D}^{(p)} = (x, (r_C, r_{\bar{C}}), (M_C, M_{\bar{C}}))$.*

Definition B.3 (specious security). *Let π be an n-party protocol and let f be an n-party function. By δ^f we denote the dummy protocol for f: it runs in the f-hybrid model and party δ_i^f on input x_i sends x_i to f, waits for the output y_i from f, outputs y_i and terminates. We say that π is a specious implementation of f against corruptions from adversary structure $\mathcal{C}$ if for all $C \in \mathcal{C}$ and all*

adversaries $\tilde{\pi}_C$ which are specious in π there exists an adversary $\tilde{\delta}_C^f$ which is specious in δ^f such that $(\pi_{\bar{C}}, \tilde{\pi}_C)(x) = (\delta_{\bar{C}}^f, \tilde{\delta}_C^f)(x)$.

The adversary $\tilde{\delta}_C^f$ is also called the simulator. It gets the input x_C and can then choose alternative inputs x'_C. Then it receives y'_C, where $y' = f(x_{\bar{C}}, x'_C)$, and outputs some $\tilde{y}_C$. In the dummy protocol, there is only one receiving point, namely after the ideal evaluation of f. So, for $\tilde{\delta}_C^f$ to be specious in δ^f it needs only be able to compute the correct view at this point. The correct view is y_C for $y = f(x)$, so a specious $\tilde{\delta}_C^f$ (in δ^f) can by definition compute y_C from x_C, x'_C and y'_C (and its own randomness if it is randomized). In words, being specious in the ideal process means that for all inputs x you give an alternative input to f which allows to reconstruct the right output.

Note that if we consider an n-party function f where all parties receive the same output, $f_i = f_j$, then it is clear that for $\tilde{\delta}_C^f$ to be specious it should hold that $f(x_{\bar{C}}, \tilde{x}_C) = f(x)$ for all inputs x, as $f(x_{\bar{C}}, \tilde{x}_C)$ is included in the messages received by $\delta_{\bar{C}}^f$. In words, for a function f with common output, being specious in the ideal process means that for all inputs x you give an alternative input to f which makes f give the right output; You can therefore only make insignificant changes to your true input.

B.2 Specious Adversaries can be Stronger than Semi-Honest Adversaries

In some settings a specious adversary is strictly stronger than a semi-honest adversary. We demonstrate this by first giving a protocol for one-out-of-two oblivious transfer (OT) which is secure against a poly-time semi-honest adversary, but insecure against a poly-time specious adversary. We then show that there exists a function f and a protocol π which is a perfectly secure implementation of f against an unbounded semi-honest adversary in the OT-hybrid model, but insecure against even a poly-time specious adversary. The first example exploits that a specious adversary can prepare its randomness in any way it wants. The second example exploits that a specious adversary can provide any input it wants to ideal functionalities (in our case the OT's of the OT-hybrid model) as long as it can later make it look as if it gave the right input.

Theorem B.4. *Under the computational assumption given below, there exists a protocol which is a secure implementation of oblivious transfer against a static, poly-time semi-honest adversary but which is insecure against a static, poly-time specious adversary.*

Assume that we have a family of trapdoor permutations, where the description of a random permutation is a random string. More formally:

- on input $n \in \mathbb{N}$ the generator G outputs (i, t), where i is uniformly random in some $\{0,1\}^\ell$, and G runs in poly-time in n.
- Each index $i \in \{0,1\}^\ell$ defines a permutation $p_i : \{0,1\}^n \to \{0,1\}^n$. Given $i \in \{0,1\}^\ell$ and $x \in \{0,1\}^n$ one can compute $y = p_i(x)$ in poly-time in n.

– Given t, where $(i,t) \leftarrow G(n)$ and $y \in \{0,1\}^n$ one can compute $x = p_i^{-1}(y)$ in poly-time in n.
– It holds for all poly-time algorithms A that the probability that it outputs $p_i^{-1}(y)$ on input (i,y), where $(i,t) \leftarrow G(n)$ and $y \stackrel{\$}{\in} \{0,1\}^n$, is negligible in n.

On security parameter n the protocol runs as follows:

1. The sender S has input two messages $m_0, m_1 \in \{0,1\}$.
2. The receiver R has input a choice bit $c \in \{0,1\}$.
3. R samples $(i_c, t_c) \leftarrow G(n)$ and $i_{1-c} \stackrel{\$}{\in} \{0,1\}^{|i_c|}$ and sends (i_0, i_1) to S.
4. S samples $x_0, x_1 \stackrel{\$}{\in} \{0,1\}^n$ and sends $(p_{i_0}(x_0), H(x_0) \oplus m_0)$ and $(p_{i_1}(x_1), H(x_1) \oplus m_1)$, where H is a (possibly randomized) hard-core bit for p.
5. R uses t_c to compute $m_c = H(p_{i_c}^{-1}(p_{i_c}(x_c))) \oplus (H(x_c) \oplus m_c)$.

It is straight-forward to prove that this protocol is computationally secure against a static semi-honest adversary in the stand-alone model[7]: The security for the receiver is perfect, and the receiver picks i_{1-c} as to not learn t_{1-c} and hence $H(x_{1-c}) \oplus m_{1-c}$ hides m_{1-c} in the sense of semantic security.

On the other hand it is clear that the protocol is not secure against a specious adversary: A specious adversary runs the protocol honestly, except that it prepares i_{1-c} by sampling $(i_{1-c}, t_{1-c}) \leftarrow G(n)$ and then uses t_{1-c} to learn m_{1-c}. The view simulator V adds i_{1-c} to the random string r such that an execution of R on r samples the uniformly random $i_{1-c} \stackrel{\$}{\in} \{0,1\}^\ell$.

Theorem B.5. *There exists a function f and a protocol π such that π is a perfectly secure implementation of f in the OT hybrid model against a static, unbounded semi-honest adversary, but insecure against a static, poly-time specious adversary.*

Proof. We look at a function $(a,b) \mapsto (x,y)$. Let a be a bit, let $b = (b_0, b_1)$ be two bits, and let $x = b_a$ and $y = \epsilon$. Consider the following protocol π: it contains two applications of OT, where in both $\mathscr{B}$ will offer input (b_0, b_1) and where in both $\mathscr{A}$ will input a. At the end $\mathscr{A}$ outputs b_a.

It is trivial that π is perfectly secure against a semi-honest adversary. It is, on the other hand, also clear that π is not secure against a specious adversary, as $\tilde{\mathscr{A}}$ can use selection bit $1-a$ in the second OT to learn b_{1-a} and then output (b_0, b_1). In the transcript α of received messages the view simulator V simply replaces b_{1-a} by b_a as the message received from the second OT, so $\tilde{\mathscr{A}}$ is indeed specious. It is also clear that no simulator for the ideal model (even if it was allowed active corruptions) can always output both b_0 and b_1. □

C Proof of Theorem 4.1

Suppose that there exists an ε-correct, ε-private protocol in the bare model for SWAP for sufficiently small ε; we will show that this implies that one of the

two players must *lose* information upon receiving a message, which is clearly impossible.

We will consider the following particular pure input state: $|\varphi\rangle := |\Psi_{0,0}\rangle^{\mathcal{A}_0\mathcal{R}_\mathcal{A}} \otimes |\Psi_{0,0}\rangle^{\mathcal{B}_0\mathcal{R}_\mathcal{B}}$, a maximally entangled state between $\mathcal{A}_0 \otimes \mathcal{B}_0$ and the reference system $\mathcal{R}_\mathcal{A} \otimes \mathcal{R}_\mathcal{B}$ that is broken down into two subsystems for convenience. Furthermore, we will consider the "purified" versions of the honest players for this protocol; in other words, we will assume that the super-operators $\mathscr{A}_1, \ldots, \mathscr{A}_n$ and $\mathscr{B}_1, \ldots, \mathscr{B}_n$ are in fact linear isometries and that therefore the players never discard any information unless they have to send it to the other party. The global state $\rho_i(\varphi)$ after step i is therefore a pure state on $\mathcal{A}_i \otimes \mathcal{B}_i \otimes \mathcal{R}_\mathcal{A} \otimes \mathcal{R}_\mathcal{B}$.

After step i of the protocol (i.e., after the ith message has been sent), Alice's state must either depend only on her own original input (if $q_i = 0$ for her simulator), or on Bob's original input (if $q_i = 1$). More precisely, by the definition of privacy (Definition 3.4), we have that

$$\Delta\left(\nu_i(\mathscr{A}, \varphi), \mathrm{tr}_{\mathcal{B}_i}[\rho_i(\varphi)]\right) \leq \varepsilon \ ,$$

where $\nu_i(\mathscr{A}, \varphi)$ is $\mathscr{A}$'s simulated view after step i and $\rho_i(\varphi)$ is the global state in the real protocol after step i. Now, suppose that $q_i = 0$, and let $|\xi\rangle \in \mathcal{A}_i \otimes \mathcal{R}_\mathcal{A} \otimes \mathcal{R}'_\mathcal{B} \otimes \mathcal{Z}$ be a purification of $\nu_i(\mathscr{A}, \varphi)$ with $\mathcal{Z}$ being the purifying system, and $\mathcal{R}_\mathcal{B}$ renamed for upcoming technical reasons. The pure state $|\xi\rangle \otimes |\Psi_{0,0}\rangle^{\mathcal{R}_\mathcal{B}\mathcal{B}_0}$ has the same reduced density matrix as $\nu_i(\mathscr{A}, \varphi)$ on $\mathcal{A}_i \otimes \mathcal{R}_\mathcal{A} \otimes \mathcal{R}_\mathcal{B}$. Hence, by Uhlmann's theorem, there exists a linear isometry $V : \mathcal{B}_i \to \mathcal{B}_0 \otimes \mathcal{Z} \otimes \mathcal{R}'_\mathcal{B}$ such that

$$V\nu_i(\mathscr{A}, \varphi)V^\dagger = |\xi\rangle\langle\xi| \otimes |\Psi_{0,0}\rangle\langle\Psi_{0,0}|^{\mathcal{B}_0\mathcal{R}_\mathcal{B}}$$

and hence

$$\Delta\left(V\rho_i(\varphi)V^\dagger, |\xi\rangle\langle\xi| \otimes |\Psi_{0,0}\rangle\langle\Psi_{0,0}|^{\mathcal{B}_0\mathcal{R}_\mathcal{B}}\right) \leq \sqrt{2\varepsilon} \ .$$

This means that if $q_i = 0$, then Bob is still capable of reconstructing his own input state after step i by applying V to his working register. Clearly, this means that $q'_i = 0$ (i.e., Bob's simulator must also not call SWAP), and therefore, by the same argument, Alice must also be able to reconstruct her own input with an isometry $V_A : \mathcal{A}_i \to \mathcal{B}_0 \otimes \mathcal{Z} \otimes \mathcal{R}'_\mathcal{A}$. The same argument also holds if $q_i = 1$: we then conclude that $q'_i = 1$ and that Alice and Bob must have each other's inputs; no intermediate situation is possible. We conclude that, at every step i of the protocol, $q_i = q'_i$.

Now, before the protocol starts, Alice must have her input, and Bob must have his, hence, $q_0 = q'_0 = 0$. At the end, the two inputs must have been swapped, which means that $q_n = q'_n = 1$; there must therefore be a step k in the protocol after which the two inputs are swapped but not before, meaning that $q_k = 1$ and $q_{k-1} = 0$. But at each step, only one player receives information, which means that at this step k, the player who received the message must lose the ability to reconstruct his own input, which is clearly impossible. □

D The Rushing Lemma

Specious adversaries are guaranteed to get the correct output state after the execution of a correct protocol. This implies that at the end of the protocol, any extra working registers (used to implement its attack) of any specious adversary are independent of the joint input state of the computation. In other words, no extra information is available to the adversary at the very end of the protocol. If the adversary can break the privacy of a protocol for the two party evaluation of unitaries then it must do so before the last step. The adversary must therefore rush to break privacy before the protocol ends.

Lemma D.1 (Rushing Lemma). *Let $\Pi_U^{\mathscr{O}} = (\mathscr{A}, \mathscr{B}, n)$ be a correct protocol for the two party evaluation of U. Let $\tilde{\mathscr{A}}$ be any ε–specious adversary in $\Pi_U^{\mathscr{O}}$. Then, there exist an isometry $T : \tilde{\mathcal{A}}_n \to \mathcal{A}_n \otimes \widehat{\mathcal{A}}$ and a mixed state $\tilde{\varrho} \in \mathrm{D}(\widehat{\mathcal{A}})$ such that for all joint input states $\rho_{\mathrm{in}} \in \mathrm{D}(\mathcal{A}_0 \otimes \mathcal{B}_0 \otimes \mathcal{R})$,*

$$\Delta\left((T \otimes \mathbb{1}_{\mathcal{B}_n \otimes \mathcal{R}})\left([\tilde{\mathscr{A}} \circledast \mathscr{B}](\rho_{\mathrm{in}})\right)(V^\dagger \otimes \mathbb{1}_{\mathcal{B}_n \otimes \mathcal{R}}), \tilde{\varrho} \otimes (U \otimes \mathbb{1}_{\mathcal{R}})\rho_{\mathrm{in}}(U^\dagger \otimes \mathbb{1}_{\mathcal{R}})\right) \leq 12\sqrt{2\varepsilon}. \tag{7}$$

The same also applies to any ε–specious adversary $\tilde{\mathscr{B}}$: there exists a $T : \tilde{\mathcal{B}}_n \to \mathcal{B}_n \otimes \widehat{\mathcal{B}}$ and a $\tilde{\varrho} \in \mathrm{D}(\widehat{\mathcal{B}})$ such that

$$\Delta\left((T \otimes \mathbb{1}_{\mathcal{A}_n \otimes \mathcal{R}})\left([\mathscr{A} \circledast \tilde{\mathscr{B}}](\rho_{\mathrm{in}})\right)(V^\dagger \otimes \mathbb{1}_{\mathcal{A}_n \otimes \mathcal{R}}), \tilde{\varrho} \otimes (U \otimes \mathbb{1}_{\mathcal{R}})\rho_{\mathrm{in}}(U^\dagger \otimes \mathbb{1}_{\mathcal{R}})\right) \leq 12\sqrt{2\varepsilon}, \tag{8}$$

for every ρ_{in}.

Proof. We shall only prove the statement for an ε–specious $\tilde{\mathscr{A}}$; the statement for an ε–specious $\tilde{\mathscr{B}}$ is identical. Furthermore, by convexity, it is sufficient to prove the theorem for pure ρ_{in}.

Consider any pair of pure input states $|\psi_1\rangle$ and $|\psi_2\rangle$ in $\mathcal{A}_0 \otimes \mathcal{B}_0 \otimes \mathcal{R}$. Now, let $\mathcal{R}' := \mathcal{R} \otimes \mathcal{R}_2$, where $\mathcal{R}_2 = \mathrm{span}\{|1\rangle, |2\rangle\}$ represents a single qubit, and define the state $|\psi\rangle := \frac{1}{\sqrt{2}}(|\psi_1\rangle|1\rangle + |\psi_2\rangle|2\rangle) \in \mathcal{A}_0 \otimes \mathcal{B}_0 \otimes \mathcal{R}'$. Note that $\mathrm{tr}_{\mathcal{R}_2}(|\psi\rangle\langle\psi|) = \frac{1}{2}|\psi_1\rangle\langle\psi_1| + \frac{1}{2}|\psi_2\rangle\langle\psi_2|$. Due to the correctness of the protocol and to the speciousness of $\tilde{\mathscr{A}}$, there exists a quantum operation $\mathscr{T}_n : \mathrm{L}(\tilde{\mathcal{A}}_n) \to \mathrm{L}(\mathcal{A}_n)$ such that

$$\Delta\left((\mathscr{T}_n \otimes \mathbb{1}_{\mathrm{L}(\mathcal{B}_n \otimes \mathcal{R}')})([\tilde{\mathscr{A}} \circledast \mathscr{B}](|\psi\rangle\langle\psi|)), (U \otimes \mathbb{1}_{\mathcal{R}'})|\psi\rangle\langle\psi|(U \otimes \mathbb{1}_{\mathcal{R}'})^\dagger\right) \leq 2\varepsilon.$$

Now, consider any isometry $T : \tilde{\mathcal{A}}_n \to \mathcal{A}_n \otimes \widehat{\mathcal{A}}$ such that $\mathscr{T}_n(\sigma) = \mathrm{tr}_{\widehat{\mathcal{A}}}(T\sigma T^\dagger)$ for every $\sigma \in \mathrm{L}(\tilde{\mathcal{A}}_n)$ — in other words, any operation that implements $\mathscr{T}_n$ while keeping any information that would otherwise be destroyed in $\widehat{\mathcal{A}}$. By Uhlmann's theorem, there must exist a state $\tilde{\varrho} \in \mathrm{D}(\widehat{\mathcal{A}})$ such that

$$\Delta\left((T \otimes \mathbb{1}_{\mathcal{B}_n \otimes \mathcal{R}'})\left([\tilde{\mathscr{A}} \circledast \mathscr{B}](|\psi\rangle\langle\psi|)\right)(T^\dagger \otimes \mathbb{1}_{\mathcal{B}_n \otimes \mathcal{R}'}), \tilde{\varrho} \otimes (U \otimes \mathbb{1}_{\mathcal{R}'})|\psi\rangle\langle\psi|(U^\dagger \otimes \mathbb{1}_{\mathcal{R}'})\right) \leq 2\sqrt{2\varepsilon}.$$

Now, the trace distance is monotonous under completely positive, trace non-increasing maps. In particular, we can apply the projector $P_1 = \mathbb{1}_{\mathrm{L}(\mathcal{A}_n \otimes \mathcal{B}_n \otimes \mathcal{R})} \otimes$

$|1\rangle\langle 1|$ to both states in the above trace distance and the inequality will still hold. In other words, we project both states onto $|1\rangle$ on $\mathcal{R}_2$, thereby turning $|\psi\rangle\langle\psi|$ into $\frac{1}{2}|\psi_1\rangle\langle\psi_1|$. Factoring out the $\frac{1}{2}$, we get that

$$\Delta\left((T\otimes \mathbb{1}_{\mathcal{B}_n\otimes\mathcal{R}})\left([\tilde{\mathscr{A}}\circledast\mathscr{B}](|\psi_1\rangle\langle\psi_1|)\right)(T\otimes \mathbb{1}_{\mathcal{B}_n\otimes\mathcal{R}})^\dagger, \tilde{\varrho}\otimes(U\otimes \mathbb{1}_{\mathcal{R}})|\psi_1\rangle\langle\psi_1|(U^\dagger\otimes \mathbb{1}_{\mathcal{R}})\right)\leq 4\sqrt{2\varepsilon}.$$

Likewise, projecting onto $|2\rangle$ yields

$$\Delta\left((T\otimes \mathbb{1}_{\mathcal{B}_n\otimes\mathcal{R}})\left([\tilde{\mathscr{A}}\circledast\mathscr{B}](|\psi_2\rangle\langle\psi_2|)\right)(T\otimes \mathbb{1}_{\mathcal{B}_n\otimes\mathcal{R}})^\dagger, \tilde{\varrho}\otimes(U\otimes \mathbb{1}_{\mathcal{R}})|\psi_2\rangle\langle\psi_2|(U^\dagger\otimes \mathbb{1}_{\mathcal{R}})\right)\leq 4\sqrt{2\varepsilon}.$$

Our only problem at this point is that $\tilde{\varrho}$ in principle depends on $|\psi_1\rangle$ and $|\psi_2\rangle$. However, repeating the above argument with $|\psi_1\rangle$ and $|\psi_3\rangle$ for any $|\psi_3\rangle$ will yield a $\tilde{\varrho}'$ with

$$\Delta\left((T\otimes \mathbb{1}_{\mathcal{B}_n\otimes\mathcal{R}})\left([\tilde{\mathscr{A}}\circledast\mathscr{B}](|\psi_1\rangle\langle\psi_1|)\right)(T^{\otimes}\mathbb{1}_{\mathcal{B}_n\otimes\mathcal{R}})^\dagger, \tilde{\varrho}'\otimes(U\otimes \mathbb{1}_{\mathcal{R}})|\psi_1\rangle\langle\psi_1|(U^\dagger\otimes \mathbb{1}_{\mathcal{R}})\right)\leq 4\sqrt{2\varepsilon}$$

and hence, by the triangle inequality, $\Delta(\tilde{\varrho},\tilde{\varrho}')\leq 8\sqrt{2\varepsilon}$. Therefore, for any state $|\varphi\rangle\in\mathcal{A}_0\otimes\mathcal{B}_0\otimes\mathcal{R}$, there exists a state $\tilde{\rho}\in\widehat{\mathcal{A}}$ with $\Delta(\tilde{\rho},\tilde{\varrho})\leq 8\sqrt{2\varepsilon}$ such that

$$\Delta\left((T\otimes \mathbb{1}_{\mathcal{B}_n\otimes\mathcal{R}})\left([\tilde{\mathscr{A}}\circledast\mathscr{B}](|\varphi\rangle\langle\varphi|)\right)(T\otimes \mathbb{1}_{\mathcal{B}_n\otimes\mathcal{R}})^\dagger, \tilde{\rho}\otimes(U\otimes \mathbb{1}_{\mathcal{R}})|\varphi\rangle\langle\varphi|(U^\dagger\otimes \mathbb{1}_{\mathcal{R}})\right)\leq 4\sqrt{2\varepsilon}.$$

The lemma then follows by the triangle inequality:

$$\begin{aligned}
&\Delta\left((T\otimes \mathbb{1}_{\mathcal{B}_n\otimes\mathcal{R}})\left([\tilde{\mathscr{A}}\circledast\mathscr{B}](|\varphi\rangle\langle\varphi|)\right)(T\otimes \mathbb{1}_{\mathcal{B}_n\otimes\mathcal{R}})^\dagger, \tilde{\varrho}\otimes(U\otimes \mathbb{1}_{\mathcal{R}})|\varphi\rangle\langle\varphi|(U^\dagger\otimes \mathbb{1}_{\mathcal{R}})\right)\\
&\leq \Delta\left((T\otimes \mathbb{1}_{\mathcal{B}_n\otimes\mathcal{R}})\left([\tilde{\mathscr{A}}\circledast\mathscr{B}](|\varphi\rangle\langle\varphi|)\right)(T\otimes \mathbb{1}_{\mathcal{B}_n\otimes\mathcal{R}})^\dagger, \tilde{\rho}\otimes(U\otimes \mathbb{1}_{\mathcal{R}})|\varphi\rangle\langle\varphi|(U^\dagger\otimes \mathbb{1}_{\mathcal{R}})\right)+\Delta(\tilde{\rho},\tilde{\varrho})\\
&\leq 4\sqrt{2\varepsilon}+8\sqrt{2\varepsilon}=12\sqrt{2\varepsilon}
\end{aligned}$$

□

E Proof of Privacy

In the following we prove the privacy $P_U^{\mathscr{O}}=(\mathscr{A}^*,\mathscr{B}^*,n_U+1)$ against specious quantum adversaries and that for any unitary $U\in\mathrm{U}(\mathcal{A}_0\otimes\mathcal{B}_0)$ represented by a quantum circuit C_U with u gates in $\mathcal{UG}$. We provide families of simulators $\mathscr{S}_{\tilde{\mathscr{A}}}$ and $\mathscr{S}_{\tilde{\mathscr{B}}}$ for any specious quantum adversaries $\tilde{\mathscr{A}}$ and $\tilde{\mathscr{B}}$ respectively. Since the protocol has n_U oracle calls, it is sufficient to provide simulators for each of these n_U steps since the final quantum operations (i.e., $\mathscr{A}_{n_U+1}$ and $\mathscr{B}_{n_U+1}$) are local. No simulator for a round occurring before the start of the *key-releasing* phase needs to call the ideal functionality for U. The output of these simulators will be shown identical to the adversary's view (i.e., the simulation is perfect) even if the adversary is arbitrarily malicious. Only the last simulator of each family needs to call to the ideal functionality for U. The last simulation produces a state that is essentially $\sqrt{\varepsilon}$–close to adversary's view provided it is ε–specious.

First, we show privacy of the *evaluation phase* before addressing privacy of the *key-releasing phase*. Privacy of the entire protocol will then follow.

E.1 Privacy of the Evaluation Phase

We start by showing privacy of protocol $P_U^{\mathscr{O}} = (\mathscr{A}^*, \mathscr{B}^*, n_U + 1)$ at all steps $1 \leq i \leq n_U - 1$ occurring during the *evaluation phase* of quantum circuit C_U implementing U with u gates in $\mathcal{UG}$. The last step of the evaluation phase is $n_U - 1$ since only one oracle call is left to complete the execution. This phase is the easy part of the simulation since all transmissions are independent of the joint input state $\rho_{\text{in}} \in \mathrm{D}(\mathcal{A}_0 \otimes \mathcal{B}_0 \otimes \mathcal{R})$. The theorem below provides a perfect simulation of any adversary's view generated during the evaluation of any gate in C_U. No call to the ideal functionality for U is required.

Theorem E.1 (Privacy of the Evaluation). *$P_U^{\mathscr{O}} = (\mathscr{A}^*, \mathscr{B}^*, n_U + 1)$ admits simulators $\mathscr{S}(\tilde{\mathscr{A}})$ and $\mathscr{S}(\tilde{\mathscr{B}})$ that do not call the ideal functionality for $U \in \mathrm{U}(\mathcal{A}_0 \otimes \mathcal{B}_0)$ such that for any joint input state $\rho_{\text{in}} \in \mathrm{D}(\mathcal{A}_0 \otimes \mathcal{B}_0 \otimes \mathcal{R})$, every $1 \leq i \leq n_U - 1$:*

$$\Delta\left(\nu_i(\tilde{\mathscr{A}}, \rho_{\text{in}}), \mathrm{tr}_{\mathcal{B}_i}\left(\tilde{\rho}_i(\tilde{\mathscr{A}}, \rho_{\text{in}})\right)\right) = 0 \text{ and } \Delta\left(\nu_i(\tilde{\mathscr{B}}, \rho_{\text{in}}), \mathrm{tr}_{\mathcal{A}_i}\left(\tilde{\rho}_i(\tilde{\mathscr{B}}, \rho_{\text{in}})\right)\right) = 0, \tag{9}$$

This holds against any adversaries $\tilde{\mathscr{A}}$ and $\tilde{\mathscr{B}}$, not necessarily specious.

Proof (Sketch). Without loss of generality, we prove privacy only against adversary $\tilde{\mathscr{A}}$. The protocol being symmetric, privacy against $\tilde{\mathscr{B}}$ follows. The proof proceeds by induction on the current gate G_j in the circuit $C_U := G_u G_{u-1} \dots G_1$ evaluated in $P_U^{\mathscr{O}}$. We provide simulators $\mathscr{S}_j^*$ producing $\tilde{\mathscr{A}}$'s view after the evaluation of G_j. During the execution of G_j, $\tilde{\mathscr{A}}$ may receive at most one message from $\mathscr{B}$ and may call the ideal AND-BOX at most once (when $G_j = \mathsf{R}$). It means that during the evaluation of G_j, no, one, or two simulations will be needed since it consists in no, one or two oracle calls out of which at most one is non-trivial. Let $s[j] \in \{0, 1, 2\}$ for $1 \leq j \leq u$ be the number of steps to be simulated during the evaluation of G_j. Let $i[0] := 0$ and $i[j] = s[j] + i[j-1]$ for $1 \leq j \leq u$ be all steps to simulate for the evaluation of $G_j G_{j-1} \cdots G_1$. In order to fulfill privacy as defined in Definition 3.3, each simulator $\mathscr{S}_j^*$ must be converted into $\mathscr{S}_{i[j-1]+1} \in \mathscr{S}(\tilde{\mathscr{A}})$ if $i[j] = i[j-1]+1$ (i.e., G_j requires only one step to be simulated and this step is a message from $\mathscr{B}^*$) and into $\{\mathscr{S}_{i[j-1]+1}, \mathscr{S}_{i[j-1]+2}\} \subseteq \mathscr{S}(\tilde{\mathscr{A}})$ if $i[j] = i[j-1]+2$ (i.e., $G_j = \mathsf{R}$ and therefore requires two simulation steps: one message from $\mathscr{B}^*$ and one call to AND-BOX). This conversion is performed the following way. We let $\mathscr{S}_{i[j-1]+1}$ run $\mathscr{S}_j^*$ until the $i[j-1]+1$–th step is reached. This step is necessarily a message transmitted from $\mathscr{B}^*$ to $\tilde{\mathscr{A}}$. If $i[j] = i[j-1]+2$ then $\mathscr{S}_{i[j]} := \mathscr{S}_j^*$ which corresponds to the simulation of $\tilde{\mathscr{A}}$'s view after the call to AND-BOX. We now provide $\mathscr{S}_j^*$ for each gate G_j in C_U. Notice that we do not explicitly simulates a communication from $\tilde{\mathscr{A}}$ to $\mathscr{B}^*$ since simulating this step can be performed from the simulation of the previous step together with $\tilde{\mathscr{A}}$ quantum operations at the current step.

$\mathscr{S}_1^*$ works as follows. It runs $\tilde{\mathscr{A}}$ on her part of the joint input state $\rho_{\text{in}} \in \mathrm{D}(\mathcal{A}_0 \otimes \mathcal{B}_0 \otimes \mathcal{R})$ until the first message from the other party is expected. If gate

G_1 does not involve any transmission from $\mathscr{B}^*$ then the simulation of gate G_1 is over (i.e., G_1 is in $\{X, Y, Z, \mathsf{H}, \mathsf{P}\}$ or a CNOT applied on local wires). Otherwise, it prepares the first message sent from $\mathscr{B}^*$. Of course, this message depends on G_1. We have the following three cases to address:

CNOT-gate: $\tilde{\mathscr{A}}$ holds the *target wire* while $\mathscr{B}^*$ holds the *control wire*. This case is the only one where $\tilde{\mathscr{A}}$ receives something from $\mathscr{B}^*$ during the computation of a CNOT-gate. $\mathscr{S}_1^*$ then works the obvious way in order to generate the first transmission from $\mathscr{B}^*$ to $\tilde{\mathscr{A}}$:

- $\mathscr{S}_1^*$ prepares $|\xi\rangle = (\mathbb{1}_2 \otimes \mathsf{CNOT} \otimes \mathbb{1}_2)|\Psi_{0,0}\rangle^{\mathcal{W}} \otimes |\Psi_{0,0}\rangle^{\mathcal{A}_1^{\mathscr{O}}}$ where $\mathcal{W}$ is a new working register for the simulator. $\mathscr{S}_1^*$ then sends register $\mathcal{A}_1^{\mathscr{O}}$ to $\tilde{\mathscr{A}}$. This simulates $\mathscr{B}^*$'s transmission to $\tilde{\mathscr{A}}$.
- The transmission prepared by $\mathscr{S}_1^*$ is in the same state as when $\tilde{\mathscr{A}}$ interacts with $\mathscr{B}^*$ upon any input state ρ_{in}. It follows that the output of $\mathscr{S}_1^*$ satisfies:

$$\Delta\left(\nu_1(\tilde{\mathscr{A}}, \rho_{\text{in}}), \text{tr}_{\mathcal{B}_1}\left(\tilde{\rho}_1(\tilde{\mathscr{A}}, \rho_{\text{in}})\right)\right) = 0,$$

for all input states $\rho_{\text{in}} \in \text{D}(\mathcal{A}_0 \otimes \mathcal{B}_0 \otimes \mathcal{R})$.

R-gate: $\tilde{\mathscr{A}}$ holds the register upon which the gate is executed. In this case, $\mathscr{S}_1^*$ provides $\tilde{\mathscr{A}}$ with $\mathscr{B}^*$'s as follows:

- $\mathscr{S}_1^*$ prepares and sends $\mathbb{1}_2 \in \text{D}(\mathcal{A}_1^{\mathscr{O}})$ to $\tilde{\mathscr{A}}$.
- $\mathscr{S}_1^*$ then call the ideal AND-BOX with a random input bit. Notice that $\tilde{\mathscr{A}}$ cannot distinguish this behavior from $\mathscr{B}^*$'s since an AND-box is non-signaling and can therefore not be used by one party to extract any information about the other party's input state (i.e., the output of one party can be generated before the input of the other party has been provided to the box).
- As in the case where $\tilde{\mathscr{A}}$ interacts with $\mathscr{B}^*$, the first message received from $\mathscr{S}_1^*$ is in state $\mathbb{1}_2$ and $\tilde{\mathscr{A}}$'s output of AND-BOX is independent of $\mathscr{B}$'s view. It follows that,

$$\Delta\left(\nu_1(\tilde{\mathscr{A}}, \rho_{\text{in}}), \text{tr}_{\mathcal{B}_1}\left(\tilde{\rho}_1(\tilde{\mathscr{A}}, \rho_{\text{in}})\right)\right) = 0,$$

for all input states $\rho_{\text{in}} \in \text{D}(\mathcal{A}_0 \otimes \mathcal{B}_0 \otimes \mathcal{R})$.

R-gate: $\mathscr{B}^*$ holds the register upon which the gate is executed. $\mathscr{S}_1^*$ provides $\tilde{\mathscr{A}}$ with $\mathscr{B}^*$'s first transmission the same way than in the previous case:

- $\mathscr{S}_1^*$ prepares and sends $\mathbb{1}_2 \in \text{D}(\mathcal{A}_1^{\mathscr{O}})$ to $\tilde{\mathscr{A}}$. This simulates $\mathscr{B}^*$ transmission to $\tilde{\mathscr{A}}$.
- $\mathscr{S}_1^*$ provides the AND-BOX with a fresh random bit as for in the previous case.
- As in the case where $\tilde{\mathscr{A}}$ interacts with $\mathscr{B}^*$, the first message received from $\mathscr{S}_1^*$ is in state $\mathbb{1}_2$ and $\tilde{\mathscr{A}}$'s output of AND-BOX is independent of $\mathscr{B}^*$'s view. It follows that,

$$\Delta\left(\nu_1(\tilde{\mathscr{A}}, \rho_{\text{in}}), \text{tr}_{\mathcal{B}_1}\left(\tilde{\rho}_1(\tilde{\mathscr{A}}, \rho_{\text{in}})\right)\right) = 0,$$

for all input states $\rho_{\text{in}} \in \text{D}(\mathcal{A}_0 \otimes \mathcal{B}_0 \otimes \mathcal{R})$.

Since the three cases above exhaust all possibilities for a transmission from $\mathscr{B}$ to $\tilde{\mathscr{A}}$, $\mathscr{S}_1^*$ satisfies (9).

Now, suppose by the induction hypothesis that $\mathscr{S}_{j-1}^*$ simulates perfectly up to and including the $j-1$-th step of the adversary $\tilde{\mathscr{A}}$ for $2 \leq j \leq n_U - 1$. We now show how to construct $\mathscr{S}_j^*$ simulating perfectly up to an including gate G_j. We construct $\mathscr{S}_j^*$ the obvious way. $\mathscr{S}_j^*$ runs $\mathscr{S}_{j-1}^*$ and then simulates $\tilde{\mathscr{A}}$ until $\mathscr{B}^*$'s next transmission occurring during the evaluation of G_j. If no such message occurs during the evaluation of G_j then $\mathscr{S}_j^*$ is done. Otherwise, the same three cases described above have to be considered. $\mathscr{S}_j^*$ provides $\tilde{\mathscr{A}}$ with $\mathscr{B}^*$'s transmission exactly the same way than for $\mathscr{S}_1^*$. The result follows easily. □

E.2 Privacy of the Key-Releasing Phase

In order to conclude the privacyof $P_U^{\mathscr{O}}$, families $\mathscr{S}_{\tilde{\mathscr{A}}}$ and $\mathscr{S}_{\tilde{\mathscr{B}}}$ need one more simulator each: $\mathscr{S}_{n_U} \in \mathscr{S}(\tilde{\mathscr{A}})$ and $\mathscr{S}'_{n_U} \in \mathscr{S}(\tilde{\mathscr{B}})$ corresponding to the simulation of the key-releasing phase. This time, these simulators need to query the ideal functionality for U and also need the adversary to be specious. We show that privacy of the key-releasing phase follows from the "Rushing Lemma" (Lemma D.1). The lemma tells us that as soon as the output is available to the honest player, it is too late for specious adversaries to break privacy. This is the role of the ideal SWAP to make sure that before the adversary gets the output of the computation, the information needed by the honest player to recover its own output has been given away by the adversary.

It should be mentioned that we're not explicitly simulating the final state of the adversary since simulating the SWAP allows also to get $\tilde{\mathscr{A}}$'s final state by simply adding $\tilde{\mathscr{A}}$'s last quantum operation to the simulated view. We therefore set step n_U in $P_U^{\mathscr{O}}$ to be the step reached after the call to SWAP. This abuses the notation a bit since after SWAP, $\tilde{\mathscr{A}}$ and $\mathscr{B}^*$ must each apply a final quantum operation with no more oracle call. We'll denote by $\tilde{\mathscr{A}}_{n_U+1}$ and $\mathscr{B}^*_{n_U+1}$ these last operations allowing to reconstruct the output of the computation (no comunication).

Lemma E.2. *For any ε-specious quantum adversaries $\tilde{\mathscr{A}}$ and $\tilde{\mathscr{B}}$ against $P_U^{\mathscr{O}} = (\mathscr{A}^*, \mathscr{B}^*, n_U + 1)$, there exist simulators $\mathscr{S}_{n_U} \in \mathscr{S}(\tilde{\mathscr{A}})$ and $\mathscr{S}'_{n_U} \in \mathscr{S}(\tilde{\mathscr{B}})$ such that for all $\rho_{\text{in}} \in \mathrm{D}(\mathcal{A}_0 \otimes \mathcal{B}_0 \otimes \mathcal{R})$,*

$$\Delta\left(\nu_{n_U}(\tilde{\mathscr{A}}, \rho_{\text{in}}), \mathrm{tr}_{\mathcal{B}_{n_U}}\left(\tilde{\rho}_{n_U}(\tilde{\mathscr{A}}, \rho_{\text{in}})\right)\right) \leq 24\sqrt{2\varepsilon} \quad \textit{and}$$
$$\Delta\left(\nu_{n_U}(\tilde{\mathscr{B}}, \rho_{\text{in}}), \mathrm{tr}_{\mathcal{A}_{n_U}}\left(\tilde{\rho}_{n_U}(\tilde{\mathscr{B}}, \rho_{\text{in}})\right)\right) \leq 24\sqrt{2\varepsilon}\,. \tag{10}$$

Simulators $\mathscr{S}_{n_U}$ and $\mathscr{S}'_{n_U}$ call the ideal functionality for U and imply the simulations of step $n_U + 1$ as well.

Proof (sketch). Once again, we only prove privacy against adversary $\tilde{\mathscr{A}}$. The privacy against $\tilde{\mathscr{B}}$ follows directly since the key-releasing phase is completely symmetric. The idea behind the proof is to run $\tilde{\mathscr{A}}$ and $\mathscr{B}^*$ upon a dummy joint input state until the end of the protocol. Since the adversary is specious, it can re-produce the honest state at the end. The Rushing Lemma tells us that at this point, the output of the computation is essentially in tensor product with all the other registers. Moreover, the state of all other registers is independent of the input state upon which the protocol is executed. The *dummy output* can then be replaced by the output of the ideal functionality for U before $\tilde{\mathscr{A}}$ goes back to the stage reached just after SWAP.

More formally, we define a simulator $\mathscr{S}_{n_U} \in \mathscr{S}(\tilde{\mathscr{A}})$ producing $\tilde{\mathscr{A}}$'s view just after the call to SWAP. Let $\tilde{\mathscr{A}}_{\mathsf{SWAP}} \in \mathrm{L}(\mathcal{A}_0, \tilde{\mathcal{A}}_{n_U})$ and $\mathscr{B}^*_{\mathsf{SWAP}} \in \mathrm{L}(\mathcal{B}_0, \tilde{\mathcal{B}}_{n_U})$ be the quantum operations run by $\tilde{\mathscr{A}}$ and $\mathscr{B}^*$ respectively until SWAP is executed. Notice that at this point, $\tilde{\mathscr{A}}$'s and $\mathscr{B}^*$'s registers do not have any further oracle registers since no more communication or oracle call will take place. Let $\tilde{A}_{n_U} \in \mathrm{L}(\tilde{\mathcal{A}}_{n_U}, \tilde{\mathcal{A}}_{n_U+1} \otimes \mathcal{Z})$ be the isometry implementing $\tilde{\mathscr{A}}$'s last quantum operation taking place after the call to SWAP (and producing the outcome) and let $B_{n_U} \in \mathrm{L}(\mathcal{B}_{n_U}, \mathcal{B}_{n_U+1} \otimes \mathcal{W})$ be the isometry implementing $\mathscr{B}^*$'s last quantum operation. Finally, let $T \in \mathrm{L}(\tilde{\mathcal{A}}_{n_U+1}, \mathcal{A}_{n_U+1} \otimes \hat{\mathcal{A}})$ be the isometry implementing $\mathscr{T}_{n_U+1}$ as defined in Lemma D.1 (i.e., the transcript produced at the very end of the protocol). As usual , let $\rho_{\mathrm{in}} \in \mathrm{D}(\mathcal{A}_0 \otimes \mathcal{B}_0 \otimes \mathcal{R})$ be the joint input state of $P^{\mathscr{O}}_U$. The simulator $\mathscr{S}_{n_U}$ performs the following operations:

1. $\mathscr{S}_{n_U}$ generates the quantum state $\sigma(\phi^*) = [\tilde{\mathscr{A}}_{\mathsf{SWAP}} \circledast \mathscr{B}^*_{\mathsf{SWAP}}](|\phi^*\rangle\langle\phi^*|) \in \mathrm{D}(\tilde{\mathcal{A}}_{n_U} \otimes \mathcal{B}_{n_U})$ implementing $\tilde{\mathscr{A}}$ interacting with $\mathscr{B}^*$ until SWAP is applied. The execution is performed upon a predetermined (dummy) arbitrary input state $|\phi^*\rangle \in \mathcal{A}_0 \otimes \mathcal{B}_0$.
2. $\mathscr{S}_{n_U}$ sets $\sigma'(\phi^*) = (T\tilde{A}_{n_U} \otimes B_{n_U}) \cdot \sigma(\phi^*) \in \mathrm{D}(\mathcal{A}_{n_U+1} \otimes \mathcal{B}_{n_U+1} \otimes \mathcal{Z} \otimes \hat{\mathcal{A}} \otimes \mathcal{W})$.
3. $\mathscr{S}_{n_U}$ replaces register $\mathcal{A}_{n_U+1} \approx \mathcal{A}_0$ by $\mathscr{A}^*$'s output of the ideal functionality for U evaluated upon ρ_{in}. That is, $\mathscr{S}_{n_U}$ generates the state $\sigma'(\rho_{\mathrm{in}}) = (U \otimes \mathbb{1}_{\mathcal{R}})\rho_{\mathrm{in}}(U \otimes \mathbb{1}_{\mathcal{R}})^\dagger \otimes \mathrm{tr}_{\mathcal{A}_{n_U+1}\mathcal{B}_{n_U+1}}(\sigma'(\phi^*)) \in \mathrm{D}(\mathcal{A}_{n_U+1} \otimes \mathcal{B}_{n_U+1} \otimes \mathcal{R} \otimes \mathcal{Z} \otimes \hat{\mathcal{A}} \otimes \mathcal{W})$.
4. $\mathscr{S}_{n_U}$ finally sets $\nu_{n_U}(\tilde{\mathscr{A}}, \rho_{\mathrm{in}}) = \mathrm{tr}_{\mathcal{B}_{n_U}\mathcal{W}}((T\tilde{A}_{n_U} \otimes \mathbb{1}_{\mathcal{B}_{n_U+1}\mathcal{R}})^\dagger \cdot \sigma'(\rho_{\mathrm{in}})) \in \mathrm{D}(\tilde{\mathcal{A}}_{n_U} \otimes \mathcal{R})$.

Notice that execution of the ideal SWAP ensures that the keys swapped are independent of each other and of the joint input state ρ_{in}. This is because for any input state, all these keys are uniformly distributed bits if they are outcomes of Bell measurements and otherwise are set to 0. By the Rushing Lemma D.1 and the fact that $\tilde{\mathscr{A}}$ is ε–specious, we have:

$$\Delta\left(\mathrm{tr}_{\mathcal{Z}\hat{\mathcal{A}}\mathcal{W}}\left(\sigma'(\phi^*)\right), \tilde{\varrho} \otimes U|\phi^*\rangle\langle\phi^*|U^\dagger\right) \leq 12\sqrt{2\varepsilon} \text{ and}$$

$$\Delta\left((\mathscr{T}_{n_U+1} \otimes \mathbb{1}_{\mathrm{L}(\mathcal{B}_{n_U+1})})\left([\tilde{\mathscr{A}} \circledast \mathscr{B}^*](\rho_{\mathrm{in}})\right), \tilde{\varrho} \otimes U\rho_{\mathrm{in}}U^\dagger\right) \leq 12\sqrt{2\varepsilon}.$$

It follows using the triangle inequality that,

$$\Delta\left((\mathscr{T}_{n_U+1} \otimes \mathbb{1}_{\mathrm{L}(\mathcal{B}_{n_U+1})})\left([\tilde{\mathscr{A}} \circledast \mathscr{B}^*](\rho_{\mathrm{in}})\right), \mathrm{tr}_{\mathcal{Z}\hat{\mathcal{A}}\mathcal{W}}\left(\sigma'(\rho_{\mathrm{in}})\right)\right) \leq 24\sqrt{2\varepsilon}. \quad (11)$$

Using the fact that isometries cannot increase the trace-norm distance and that $(T\tilde{A}_{n_U})^\dagger$ allows $\tilde{\mathscr{A}}$ to go back from the end of the protocol to the step reached after SWAP, we get from (11) that:

$$\Delta\left(\nu_{n_U}(\tilde{\mathscr{A}},\rho_{\text{in}}),\text{tr}_{\mathcal{B}_{n_U}}\left(\tilde{\rho}_{n_U}(\tilde{\mathscr{A}},\rho_{\text{in}})\right)\right)=\Delta\left((\mathscr{T}_{n_U+1}\otimes\mathbb{1}_{\text{L}(\mathcal{B}_{n_U+1})})\left([\tilde{\mathscr{A}}\circledast\mathscr{B}^*](\rho_{\text{in}})\right),\text{tr}_{\mathcal{Z}\hat{A}\mathcal{W}}\left(\sigma'(\rho_{\text{in}})\right)\right)$$
$$\leq 24\sqrt{2\varepsilon}.$$

The statement follows. □

Theorem E.1 and Lemma E.2 imply the privacy of $P_U^{\mathscr{O}}$ against specious adversaries and that for any $U\in\text{U}(\mathcal{A}_0\otimes\mathcal{B}_0)$ as stated in our main Theorem 6.1. When U is in the Clifford group, one call to an ideal SWAP is sufficient to ensure privacy. Unitaries in the Clifford group are, to some extent, the *easy* ones since although an ideal functionality is required for privacy, that functionality is unitary and belongs to the Clifford group rather than a classical cryptographic primitive. If U is not in the Clifford group however, one additional call to a classical AND-BOX is required for each R-gate. This is reminiscent to classical circuits with AND gates where oblivious transfer is required to be able to evaluate them privately. In order to implement a classical AND-BOX, commitments and quantum communication are sufficient and necessary [9, 14].