Secure Quantum Two-Party Computation: Impossibility and Constructions

Michele Ciampi[1], Alexandru Cojocaru[1], Elham Kashefi[1,2], and Atul Mantri[3]

[1] *School of Informatics, University of Edinburgh*
[2] *Laboratoire d'Informatique de Paris 6 (LIP6), Sorbonne Université*
[3] *Joint Center for Quantum Information and Computer Science (QuICS), University of Maryland*

October 15, 2020

Abstract

Secure two-party computation considers the problem of two parties computing a joint function of their private inputs without revealing anything beyond the output of the computation. In this work, we take the first steps towards understanding the setting in which the two parties want to evaluate a joint quantum functionality while using only a classical communication channel between them. Our first result indicates that it is in general *impossible* to realize a two-party quantum functionality against malicious quantum adversaries with black-box simulation, relying only on classical channels. The negative result stems from reducing the existence of a black-box simulator to the existence of an extractor for classical proof of quantum knowledge, which in turn leads to violation of the quantum no-cloning.

Towards the positive results, we first introduce the notion of *Oblivious Quantum Function Evaluation* (OQFE). An OQFE is a two-party quantum cryptographic primitive with one fully classical party (Alice) whose input is (a classical description of a) quantum unitary, U, and a quantum party (Bob) whose input is a quantum state, ψ. In particular, Alice receives the classical output corresponding to the measurement of $U(\psi)$ while Bob receives no output. At the same time, the functionality guarantees that Bob remains oblivious to Alice's input U, while Alice learns nothing about ψ more than what can be learned from the output of the computation. We present two concrete constructions, one secure against semi-honest parties and the other secure against malicious parties. Due to the no-go result mentioned above, we consider what is arguably the best possible notion obtainable in our model with respect to malicious adversaries: *one-sided simulation* security. This notion protects the input of one party (the quantum Bob) in the standard simulation-based sense, and protects the privacy of the other party's input (the classical Alice). We realize our protocol relying on the assumption of quantum secure injective homomorphic trapdoor one-way functions, which in turn rely on the learning with errors problem. As a result, we put forward a first, simple and modular construction of secure one-sided quantum two-party computation and quantum oblivious transfer over classical networks.

Contents

1 Introduction

Secure multi-party computation (MPC) [2, 3] allows multiple distrusting parties to jointly compute any function of their joint input, such that no information is leaked about their private inputs apart from what can be inferred from the correct output of the computation [4, 5]. From the security point of view, it is well known that MPC is not possible with information-theoretic security, nonetheless, the field of MPC is actively studied in the last several decades both from the theoretical and practical side, relaxing the security to computational and/or trusted-third party scenarios (in both honest and dishonest majority of adversaries). Recently, an MPC protocol has been proposed that guarantees security against quantum adversaries [6]. MPC in the quantum world, is still relatively less studied in comparison to its classical counterpart. Secure quantum multiparty computation (QMPC) and the case of quantum 2-party computation (Q2PC) have been originally proposed in [7, 8] and [9, 10] and further explored in [11, 12, 13, 14]. Here we will only consider the case of two mutually distrusting parties.

All the previous works either realize post-quantum secure classical functionalities over classical networks or require all the involved parties to possess quantum resources and are based on quantum networks. This means a two-way quantum communication channel is a must if we want to implement a joint quantum function and hence, puts a rather heavy burden on the parties involved. Nonetheless, similar to the classical setting, we know that information-theoretic secure QMPC is not possible. Therefore, the security achievable in QMPC protocols with the quantum channel are at best computational, despite requiring parties to have powerful quantum devices and access to the quantum channel. Therefore, a natural question regarding the trade-off between the functionality achieved and the resources needed is the following:

Do all parties involved in (computationally) secure quantum MPC *protocols require quantum devices and need to share quantum channels between them?*

Getting insights into the minimum requirements for QMPC protocols is important not only from the foundational perspective, but will also lay the stepping stone for the practical QMPC protocols. Another common theme in almost all the prior works is that they are not modular and are therefore non-trivial to further optimise in terms of resources. For example, we know how MPC can be realised in the classical world based on oblivious transfer (as OT is necessary and sufficient for 2PC and MPC [15]). However, a similar module for QMPC and Q2PC is yet to be identified[1] and remains an important open problem. Ideally, we would like to construct a primitive that achieves the quantum functionality equivalent of classical oblivious transfer, that would be universal for Q2PC, and which would require only classical communication channels. Furthermore, analysing the security of such functionality in sequential or concurrent composition would help to guarantee the security of more complex protocols for Q2PC and QMPC. Keeping in mind this grand vision, we take the first step towards this direction and present a modular construction of a quantum 2PC protocol based entirely on a classical channel without compromising the security of either party. In particular, we focus in this paper on the functionality of secure function evaluation, an important class of 2PC protocols in the classical world [4]. In such a setting, a client wants to evaluate an input x of a function f, in such a way that only the clients get the output. Moreover, the server should not learn anything about the client's input, and the client should not learn anything about f beyond the output $f(x)$.

We present an analogue of secure function evaluation in the quantum setting and we call it *oblivious quantum function evaluation* (OQFE). Informally, the OQFE functionality consists of two parties: a user Alice and a server Bob that allows Alice to evaluate a quantum unitary U_i from a set of unitaries $\mathcal{U}$ (known to both parties) on Bob's private input state ψ. Additionally, OQFE guarantees that Alice learns nothing more than the output state $U_i(\psi)$, and Bob gets no information about the index i of Alice at the end of the protocol.

Our work also initiates the study towards the following questions: *Is it possible to achieve* Q2PC *with the same level of security when the parties only share a classical communication channel? What are the minimum cryptographic assumptions required to construct* OQFE *and therefore,* Q2PC*?* We settle the first question with a no-go result on achieving black-box simulation-based security against quantum adversaries. This notion of security guarantees the existence of a simulator that works by having only black-box access to the adversary. We circumvent the impossibility, by considering the weaker notion of one-sided simulation.

[1] There are works on quantum oblivious transfer but the aim in those works is to achieve classical functionality using quantum resources.

This notion protects the input of one party (the quantum Bob) in the standard simulation-based sense and protects the privacy of the other party's input (the classical Alice).

Regarding the second question, we provide a connection between OQFE and some fundamental quantum primitives such as secure remote state preparation (RSP) [16, 1, 17, 18]. In particular, we present an explicit construction of OQFE from RSP (and in turn, of quantum OT from OQFE) which can be further reduced to the existence of injective homomorphic trapdoor one-way functions (OWFs). Classically, we know that there exists a black-box separation between OT and OWFs [19]. However, if the parties are equipped with quantum resources and share quantum channel, then one-way functions are sufficient and the black-box reduction no longer holds [20]. It remains an open problem if secure OT can be constructed from OWFs in the quantum world where parties only share a classical channel.

Our modular construction of OQFE from RSP establishes the latter primitive as a candidate for a universal primitive. It is worth mentioning that due to the direct link, any further optimisation of the complexity of RSP will also provide answers to the complexity of OQFE. This, in turn, will enhance our understanding of the question of minimal complexity-theoretic resources required for important cryptographic primitives such as delegated quantum computing [21] and classical verification of quantum computing [17, 22] in addition to quantum multi-party computation.

1.1 Our Contributions

All the results in this work concern secure quantum two-party computation (Q2PC) over a classical channel. Nonetheless, we divide our contributions into two sets. The first one is independent of any protocol and uncovers a fundamental trade-off between the achievable security and the resources required in any Q2PC. The second one is more concrete in nature and we present a candidate construction for quantum two-party computation with classical Alice and quantum Bob and further analyse the security under different threat models.

Security of Q2PC with classical channel. What is the best possible security, achievable when a quantum two-party computation is implemented over classical networks? We answer this question at the highest level of abstraction, i.e. without going into the details of the implementation, particular functionality, and model of computation. Doing so, we analyze the difficulty underlying the process of achieving simulation-based security with black-box access to the adversary. To elaborate on this further, firstly, we formalize a connection between the existence of a simulator for any malicious adversary in (quantum) multi-party computation and the existence of an extractor (with the same run-time) in (quantum) proofs/arguments of (quantum) knowledge. This can be informally summarised as follows.

Theorem 1 (Informal) *Any secure black-box two-party (quantum) computation is also a proof of (quantum) knowledge with negligible knowledge error.*

As a consequence of this relation, we present a no-go result for the black-box fully-simulatable security of any Q2PC that only requires classical communication channels between the parties.[2]

Corollary 1 (Informal) *Secure quantum two-party computation over classical channel with black-box simulation is impossible.*

The negative result arises essentially from the connection to classical proof of quantum knowledge where the existence of an extractor is used to build an algorithm for cloning arbitrary quantum states. Since the latter is an impossible task even with unbounded devices, this makes our argument robust against extractor run-time. Due to these limitations, we resort to a one-sided simulation framework.

Oblivious Quantum Function Evaluation. We introduce the notion of oblivious quantum function evaluation (OQFE): a quantum analogue of secure function evaluation that only relies on a classical network between the parties. For the sake of simplicity, we first present a simplified case of OQFE that we call

[2] Our result also applies to a quantum multi-party computation where any two-party only share a classical channel.

1-out-of-2 OQFE. The constructions for 1-out-of-2 OQFE and OQFE are based on one-bit teleportation and measurement-based model of quantum computation (in particular, the second one is inspired by the universal blind quantum computation protocol [23]). In 1-out-of-2 OQFE, we present two concrete protocols one of which is non-interactive, secure against semi-honest Alice and protects the input of Alice against a malicious Bob. Then, in the second protocol, we uplift the security of the protocol against malicious Alice achieving the best possible security: one-sided simulation. Finally, we present the extension of 1-out-of-2 OQFE to full OQFE and we obtain the following result.

Theorem 7 (Informal) *There exists an* OQFE *protocol (Protocol 7.1) that securely computes the* OQFE *functionality with one-sided simulation.*

All the protocols in this work only require a classical channel instead of a quantum channel between Alice and Bob. This task is achieved using recent (computationally-secure) classical-client remote state preparation protocols as a sub-module. Specifically, given the underlying connection to remote state preparation, this reduces the cryptographic complexity of OQFE to injective homomorphic trapdoor quantum one-way functions. Additionally, as a side and direct application of OQFE, we give a simple construction of a quantum oblivious transfer protocol that relies only on a classical channel.

1.2 Related Works

The first work that studied the question of MPC in the quantum domain is [8], where a secure QMPC protocol is proposed based on the ideas of verifiable quantum secret sharing of the inputs of all the parties and later extended in [8, 7, 24].

In a series of works by Dupuis et al. [9, 10] the setting of two-party quantum computation is presented using the tools from classical MPC and quantum authentication codes developed in [25]. This protocol is generalized to the multi-party setting with the dishonest majority in a recent work by Dulek et al. [14]. In a recent work of [26], the authors propose a garbling scheme for quantum circuits.

A different approach inspired by delegated quantum computing [23, 27] towards secure two-party computation, similar to a quantum analog of Yao's protocol from classical MPC [4], is studied in [11, 12] and towards (composable) secure multi-party quantum computation in [28, 29, 30].

All previous works on secure two-party and multi-party quantum computation considers both quantum and classical communication channel shared between parties. However, in our work, we propose a secure two-party quantum computation protocol that only requires a classical channel and completely mitigates the need for a quantum channel, as well as quantum resources from one of the parties.

From a cryptographic complexity point of view, Unruh in [31], building upon the works of quantum oblivious OT [20] and MPC [32], proposed a UC-secure protocol for classical multi-party computations using only commitments and a quantum channel. This is known to be impossible in a purely classical setting [19]. It is not clear whether a quantum channel is necessary to achieve MPC with quantum parties just relying on commitments. However, in this work, we show that a quantum channel is necessary to achieve general two-party quantum computation in the setting of malicious parties (and hence in the UC security model as well). Furthermore, we relax the security requirement to one-sided simulation and show that classical channel is indeed sufficient albeit our construction requires primitives stronger than quantum one-way functions.

1.3 Overview of Our Results and Techniques

We describe the approach to our results and give a brief overview of the tools and techniques involved.

Limitations of black-box secure Q2PC.

It is known that if the functionality of MPC is classical then one-sided protocols cannot be information-theoretically secure in the quantum world (using quantum resources) [33, 34, 35]. Our first result takes a step forward in generalising this result for quantum functionalities. In particular, we show that one-sided quantum protocols over a completely classical channel cannot be secure in the full-simulation secure sense even against *computationally-bounded* adversaries. This features a striking trade-off between the resources needed to achieve the desired level of security for quantum two-party computation (Q2PC). We start by establishing a connection between black-box-security for two-party computation and proofs/arguments of knowledge for both classical and quantum two-party functionalities. Classically, this connection is folklore, nonetheless, we formalize here this relationship in a unified manner taking into account both the classical and quantum two-party functionalities.

The high-level idea behind the reduction of black-box-security of two-party computation to the existence of proof of knowledge (e.g. Theorem 1) is the following. We assume that the two-party functionality is secure in the black-box simulation-based model (as per Definition 9). Consider one of the parties say, Alice, is honest while Bob is fully malicious. This immediately implies the existence of a simulator Sim for this functionality that uses the adversary (Bob's state) in a black-box fashion. Next, we construct an extractor $\mathsf{Extract}$ which calls Sim in an oracle manner to extract the witness state. For a quantum relation (such as QMA-relation, Definition 7) such a witness would be quantum, unlike the classical relations. Note that the computational power of simulator Sim also differs in the classical and the quantum world. As a result, we obtain an extractor whose run-time is expected PPT in the classical and QPT in the quantum setting, respectively. A formal proof of how $\mathsf{Q2PC}$ implies proof of quantum knowledge is provided in Section 3 and the proof of how $\mathsf{2PC}$ implies proof of knowledge is presented in Appendix D, as it essentially follows the same argument. Building upon the previous result, we then present the impossibility of one-sided two-party quantum protocols based on a classical channel (Corollary 1). This is possible due to the above reduction and a recent result on classical proofs of quantum knowledge [36]. To further refine this argument, note that, as a result of the above connection, a $\mathsf{Q2PC}$ with a classical channel would imply classical proofs of quantum knowledge. Let us assume that at least one of the parties in $\mathsf{Q2PC}$ functionality is quantum (this is the most general setting possible) and is malicious. Then we claim that the existence of $\mathsf{Q2PC}$ over a classical channel for such a functionality in the black-box simulation security reduces to cloning an arbitrary quantum state. The latter task is not possible even with unbounded computational power. Loosely speaking, this is because the security of such a functionality over classical channel implies the existence of a simulator Sim which can be invoked by an extractor $\mathsf{Extract}$ to extract quantum states of malicious parties using only classical transcript. For some cheating strategies, such an algorithm ($\mathsf{Extract}^{\mathsf{Sim}}$) outputs multiple copies of an unknown quantum state using only classical messages as the input. This is not possible even with a non-black-box definition of proof of quantum knowledge provided in [36] and the no-go result is independent of the computational power of $\mathsf{Extract}$. A formal argument is presented in Section 3.

Next, we move towards our positive results and propose a construction for one-sided two-party quantum computation. We further analyze the security in the one-sided simulation model, which is (arguably) the best possible scenario given the above limitation.

Our Constructions.

The one-sided two-party classical-quantum setup consists of two parties - Alice ($\mathfrak{A}$) and Bob ($\mathfrak{B}$) - that have their own private inputs and wish to perform a joint computation, but where only one of the parties receives the output. In the remainder of this work, we will use the following convention: a) the joint computation is described by both parties' inputs, and b) Alice would be the (classical) party who receives the output. One of the potential applications of such a setting is the following. Imagine a scenario where one of the parties, Bob, has a quantum database and the other party, Alice, queries the database in such a way that i) Bob would like to keep the entries of the database secure except the one which is queried, and ii) Alice would like to maintain the privacy of her requested query. We call such functionality as the *oblivious quantum function evaluation* (OQFE) in analogy with secure function evaluation in the classical setting. The OQFE ideal functionality is provided in Section 4 along with the security model.

Informally, the description of this functionality (as shown in Figure 1) can be understood as follows:

1. Alice has as input (a classical description of) a quantum function f, where f has quantum input and classical output.

2. Bob has as input a quantum state ψ.

3. Alice obtains $f(\psi)$ and "learns nothing" more than this information. At the same time, Bob receives no output and "learns nothing" about f.

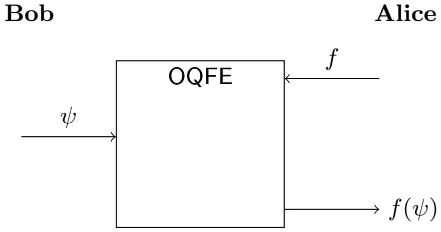


Figure 1: Oblivious quantum function evaluation functionality with two parties - classical Alice and quantum Bob.

In general, ψ could be an arbitrary quantum state (pure as well mixed) and since Alice is classical, f denotes the quantum map that consists of a unitary U followed by measurement in the computational basis. We provide a modular approach towards the construction of an OQFE protocol and prove its security in the one-sided simulation-based framework. At a high level, we first give a protocol that achieves privacy against quantum Bob and (statistical) security against semi-honest Alice. Then, in the second construction, using cryptographic tools such as secure commitment schemes and zero-knowledge proof of knowledge, we uplift the security to full simulation-based security against malicious Alice.

To this end, in Section 4.2, we present a simplified functionality called 1-out-of-2 OQFE (Definition 12), where Alice's input is a single bit b and Bob's input is a single qubit state ψ. The target computation is 1 out of 2 possible functions f_0 and f_1. This functionality ensures that Alice obtains $f_b(\psi)$ without "learning" anything about the other function applied on Bob's input (i.e. $f_{1\oplus b}(\psi)$), while Bob "learns" nothing about Alice's input b. The notion of learning is formalised using one-sided simulation framework (Section 3.2). Aside from being instrumental in the construction and security proofs of the full OQFE protocol, this simple functionality of 1-out-of-2 OQFE can be of independent interest. For instance, we show in Appendix F that it can lead to the construction of quantum oblivious transfer over classical channel.

In Section 5, we present a concrete protocol for 1-out-of-2 OQFE that is non-interactive and achieves security against semi-honest Alice. This protocol sets the stage for 1-out-of-2 OQFE functionality that is one-sided simulation-based secure, which in turn is extended to a full OQFE protocol with one-sided simulation security. The central idea behind the construction of this protocol (Protocol 5.2) is inspired by simple one-bit teleportation circuits and relies on a remote state preparation sub-module to eliminate the need for quantum communication between Alice and Bob. While the security of this construction holds only against semi-honest Alice, we can show that it holds in the statistical regime. On the other hand, for Bob, we show that the privacy of Alice's input is achieved given the hardness of the LWE problem, where this computational security arises from the use of classical-client RSP. To sum up, the result of Section 5 is given by the following theorems.

Theorems 3 and 4 (Informal) *The 1-out-of-2 OQFE Protocol 5.2 achieves privacy against malicious Bob and is statistically secure against semi-honest Alice.*

To uplift the security from semi-honest Alice to a malicious Alice, we need to be able to validate the transcripts Alice is sending to Bob during the run of the protocol. Specifically, in Section 6, we first show that this can be reduced to ensuring that Alice generates correctly the public key of the trapdoor function employed within the RSP procedure. As a result, we need to impose that Alice runs the correct key generation algorithm and at the same time she should not use a "bad" randomness in executing this algorithm. We follow standard techniques from the (classical) modern cryptography literature. In order to ensure the first condition, Alice will prove to Bob that the key is an output of the public key generation algorithm using a zero-knowledge proof of knowledge procedure. Then, for the second condition, Alice and Bob will run a variant of a coin-tossing protocol, in which Alice sends a commitment of a random string to Bob. Upon receiving the commitment, Bob replies to Alice with another randomness. Finally, Alice proves using a zero-knowledge proof system that the key was generated using the key generation algorithm and the randomness computed as the exclusive or (XOR) of the two random strings chosen by Alice and Bob. These tools will ensure the security against malicious Alice, however, it is also important to demand the zero-knowledge property and the hiding property of the commitment scheme to remain secure against quantum Bob. The result can be summarised with the following theorem and is proved in Section 6.

Theorem 5 (Informal) *The* 1*-out-of-*2 OQFE *Protocol 6.1 securely computes the* 1*-out-of-*2 OQFE *functionality with one-sided simulation.*

In Section 7, we extend the construction of Protocol 6.1 to the full OQFE protocol with the desired level of security.

More concretely, our construction for one-sided simulation secure OQFE protocol is based on the measurement-based model of quantum computation, by combining the blind quantum computation protocol [23] with RSP as subroutines. Additionally, we employ the following cryptographic tools: commitment scheme and zero-knowledge proof of knowledge to enforce the honest behavior of Alice. This is done in a similar manner and with the help of proof techniques we developed for the simplified setting of 1-out-of-2 OQFE. We obtain the following result for our full OQFE construction.

Theorem 7 (Informal) *The* OQFE *Protocol 7.1 securely computes the* OQFE *functionality with one-sided simulation.*

Finally, we would like to remark that the cryptographic complexity of our Q2PC proposal can be reduced to the cryptographic assumptions required to achieve classical-client RSP. More specifically, our construction can be instantiated with injective homomorphic trapdoor quantum one-way functions and leaves the possibility of realizing Q2PC over a classical channel with only quantum one-way functions as an important open problem.

1.4 Organization of the paper

In Section 2, we define the notations and definitions of black-box simulation-based two party (quantum) functionality followed by relevant classical as well as quantum cryptographic primitives. In Section 3, we present an impossibility result for simulation-based quantum two-party over classical channel in the black-box model and discuss the issue concerning security against both parties. Subsection 3.2 describes our (relaxed) security model of one-sided simulation. In Section 4, we present a simple construction for 1-out-of-2 oblivious quantum function evaluation. Section 5 and Section 6 analyse the security of 1-out-of-2 oblivious quantum function evaluation in semi-honest and malicious setting. An extension from 1-out-of-2 oblivious quantum function evaluation to full oblivious quantum function evaluation along with its security is presented in Section 7.

2 Preliminaries

2.1 Notations

In this paper when we talk about distributions being indistinguishable for any PPT adversary we will use the symbol $\approx_c$, if they are indistinguishable for any QPT distinguisher, we will use $\approx_q$ and if they are indistinguishable for an unbounded adversary we will use $\approx_u$. Additionally, when testing for equality we will use directly the symbol "=". Let $\mathcal{A}$ and $\mathcal{B}$ be two Hilbert spaces. The set $\mathsf{L}(\mathcal{A}, \mathcal{B})$ is the set of all linear maps from $\mathcal{A}$ to $\mathcal{B}$. The set $\mathsf{L}(\mathcal{A}) = \mathsf{L}(\mathcal{A}, \mathcal{A})$ is the set of all linear maps on $\mathcal{A}$ and the mapping $\varphi : \mathsf{L}(\mathcal{A}) \mapsto \mathsf{L}(\mathcal{B})$ is also called super-operator. If φ is completely positive and preserves the trace then such a super-operator is also known as quantum operation or CPTP map. We denote identity operator as $\mathbb{I}$ and $\mathcal{A} \otimes \mathcal{B}$ denotes the space of two such quantum registers. For more details on the quantum background we refer the readers to [37].

In the rest of the section, we define black-box two-party quantum computing functionality along with the definitions of classical and quantum primitives used in this work including proof of quantum knowledge. Some parts of this section are taken from [38, 39, 40]. The remaining definitions related to interactive quantum machines and (quantum) oracle are presented in Appendix A.

2.2 Secure 2-party Computation

We extend the definition of black-box 2-party computation from [41] to quantum functionalities. Let F be a joint quantum function $F : \mathsf{L}(\mathcal{A}_{in}, \mathcal{B}_{in}) \mapsto \mathsf{L}(\mathcal{A}_{out}, \mathcal{B}_{out})$ with input x and y from Alice and Bob, resp. Let $\mathsf{REAL}_{\Pi,\mathcal{A}(z),i}(x, y, 1^\lambda)$ and $\mathsf{IDEAL}_{F,\mathcal{S}(z),i}(x, y, 1^\lambda)$ be the output in the real and ideal execution for F when the adversary $\mathcal{A}$ is controlling party $i \in \{0, 1\}$ with the auxiliary input z.

Definition 1 (Black-box (Quantum) 2-party computation)**.** *We say that a protocol* Π *securely computes* F *if for every* $i \in \{0, 1\}$*, every non-uniform quantum-polynomial-time (*QPT*) adversary* $P_i^\star$ *controlling* P_i *in the real model, there exists a non-uniform quantum polynomial-time adversary* Sim_i *(having black-box access to* P_i*) for the ideal world such that:*

$$\{\mathsf{REAL}_{\Pi,P_i^\star(z)}(1^\lambda)\}_{z\in\{0,1\}^\lambda} \approx_q \{\mathsf{IDEAL}_{F,\mathsf{Sim}_i(z)}(1^\lambda)\}_{z\in\{0,1\}^\lambda}$$

where $\mathsf{REAL}_{\Pi,P_i^\star(z)}(1^\lambda)$ *denotes the distribution of the output of the adversary* $P_i^\star$ *(controlling the* P_i*) after a real execution of protocol* Π*.* $\mathsf{IDEAL}_{F,\mathsf{Sim}_i(z)}(1^\lambda)$ *denotes the analogous distribution in an ideal execution with a trusted party that computes* F *for the parties and hands the output to them.*

2.3 Classical and Quantum Cryptographic Primitives

Polynomial time relation Rel is a subset of $\{0, 1\}^* \times \{0, 1\}^*$ such that membership of (x, w) in Rel can be decided in time polynomial $|x|$. For a polynomial-time relation Rel, we define the NP language $L_{\mathsf{Rel}} := \{x \mid \exists w \text{ such that } (x, w) \in Rel\}$.

Definition 2 (Proof/Argument system)**.** *A pair of* PPT *interactive algorithms* $\Pi = (\mathsf{K}, \mathsf{P}, \mathsf{V})$ *constitutes a* proof system *(resp., an* argument system*) for an* NP*-language* L*, if the following condition holds:*

Completeness: *For every* $x \in L$ *and* w *such that* $(x, w) \in \mathsf{Rel_L}$*, it holds that for every* $\mathsf{crs} \leftarrow \mathsf{K}(1^n)$*:*

$$\Pr[\langle \mathsf{P}(w), \mathsf{V}\rangle(\mathsf{crs}, x) = 1] = 1.$$

Soundness: *For every interactive (resp., PPT interactive) algorithm* $\mathsf{P}^\star$*, there exists a negligible function* ν *such that for* $\mathsf{crs} \leftarrow \mathsf{K}(1^n)$ *and for every* $x \notin L$ *and every* z*:*

$$\Pr[\langle \mathsf{P}^\star(z), \mathsf{V}\rangle(\mathsf{crs}, x) = 1] < \mathsf{negl}(n)\,.$$

Definition 3. *Let* $(\mathsf{K}, \mathsf{P}, \mathsf{V})$ *be an interactive proof (argument) system for an NP-language* L*. We say that* $(\mathsf{K}, \mathsf{P}, \mathsf{V})$ *is* post-quantum zero-knowledge *with respect to an auxiliary input if for every quantum polynomial-size interactive machine* $\mathsf{V}^\star$ *there exists a black-box*[3] *quantum polynomial-time* $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$ *such that:*

$$\{\mathsf{crs} \leftarrow \mathsf{K}(1^n), \langle \mathsf{P}(w), \mathsf{V}^\star(z)\rangle(\mathsf{crs}, x)\}_{(x,w)\in\mathsf{Rel}_L, z\in\{0,1\}^\star} \approx \{\mathsf{Sim}_1(1^n), \mathsf{Sim}_2(\mathsf{crs}, x, z)\}_{x\in L, z\in\{0,1\}^\star}$$

Instantiation. A post-quantum zero-knowledge argument system can be instantiated from the scheme proposed in [42]. This scheme also satisfies the property of the argument of knowledge for the case of PPT adversaries.

Definition 4 (Remote State Preparation (Informal))**.** *A resource* S *is a Remote State Preparation resource if it outputs on the right interface (Bob's interface) a quantum state* ρ *and on the left interface (Alice's interface) a classical description of a state* ρ' *such that the states* ρ *and* ρ' *are close in trace distance.*

Instantiation. A remote state preparation primitive can be instantiated using the QFactory protocol [1]. This protocol is described in

Section 5.

Next, we define proof of knowledge in both the classical and quantum setting.

[3] The simulator has only black-box access to the adversary $\mathsf{V}^\star$.

A *Proof of Knowledge (PoK)* is an interactive proof system for some relation R such that if the verifier accepts some input x with high enough probability, then she is "convinced" that the prover "knows" some witness w such that $(x,w) \in R$. This notion is formalized by requiring the existence of an efficient *extractor* $\mathsf{Extract}$, that is able to return a witness for x when given oracle access to the prover (including the ability to rewind its actions, in the classical case).

Definition 5 (Proof of Knowledge[4])**.** *A pair $(\mathcal{P},\mathcal{V})$ of* PPT *interactive machines is a* proof of knowledge with knowledge error $k(\cdot)$ *for polynomial-time relation* Rel *if the following properties hold:*

- Completeness: *for every $(x,w) \in \mathsf{Rel}$, it holds that*

$$\Pr[\langle \mathcal{P}(w), \mathcal{V}\rangle(x) = 1] = 1 - \mathsf{negl}(|x|).$$

- Knowledge Soundness: *there exists a probabilistic oracle machine* $\mathsf{Extract}$*, called the* extractor*, running in expected probabilistic polynomial time, such that for every interactive machine $\mathcal{P}^\star$ and for every input x accepted by $\mathcal{V}$ when interacting with $\mathcal{P}^\star$ with probability $\epsilon(x) > k(x)$, we have*

$$\Pr\Big(((x,w) \in R) : w \leftarrow \mathsf{Extract}^{\mathcal{P}^\star}(x)\Big) \geq p\left(\epsilon(x) - k(x), \frac{1}{\mathsf{poly}(|x|)}\right).$$

The notion of the argument of knowledge is essentially the same but it requires the knowledge soundness property to hold against PPT adversaries and for a sufficiently long input [43].

Definition 6 (Quantum Proof of Classical Knowledge [38])**.** *Let $R \subseteq \mathcal{X} \times \mathcal{Y}$ be a relation. A proof system (P,V) for R is a Quantum Proof of Knowledge for R with knowledge error κ if there exists a polynomial $p > 0$ and a quantum polynomial-time machine* $\mathsf{Extract}$ *such that for any quantum interactive machine P^*, any $\mu \in \mathbb{N}$, any polynomial $l > 0$, any instance $x \in \{0,1\}^n$ for $n = \mathsf{poly}(\mu)$ and any state ρ: if the execution $(P^*(x,\rho), V(x))$ returns 1 with probability $\varepsilon > \kappa(\mu)$, we have:*

$$\Pr\Big(((x,\sigma) \in R) : \sigma \leftarrow \mathsf{Extract}^{|P^*(x,\rho)\rangle}(x)\Big) \geq p\left(\varepsilon - \kappa(\mu), \frac{1}{n}\right),$$

where $\mathsf{Extract}^{|P^*(x,\rho)\rangle}$ is the extractor $\mathsf{Extract}$ with oracle access to (quantum) P^* and ρ is some (quantum) side information held by P^*.
Recently, proof systems have also been extended to QMA-relations in [39, 40]. The main difference from Quantum Proofs of (classical) Knowledge is that in the case of QMA relations, the notion of a witness is in a different manner than NP relations. For any $0 \leq \gamma \leq 1$, a quantum relation is defined as follows:

$$R_{Q,\gamma} = \{(x,\sigma) : Q \text{ accepts } (x,\sigma) \text{ with probability at least } \gamma\}.$$

The parameter γ quantifies the expected success probability for the verifier and roughly speaking, γ is a measure of the "quality" of a witness $|\psi\rangle$ (or mixture thereof, as represented by the density matrix σ) that is sufficient for the witness to be acceptable with respect to the relation R. More formally, we have:

Definition 7 (QMA-relation)**.** *Let A be a problem in* QMA *(See Definition 16), and let Q be a* QPT *verifier, with completeness α and soundness β. Then, we say that $R_{Q,\gamma}$ is a* QMA*-relation such that the following holds*

1. *(Completeness)* $(x, |\psi\rangle) \in R_{Q,\alpha} \implies \Pr\big[Q_{|x|}(x,|\psi\rangle) = 1\big] \geq \alpha$
2. *(Soundness)*$(x,\rho) \notin R_{Q,\beta} \implies \Pr\big[Q_{|x|}(x,\rho) = 1\big] < \beta$.

Additionally, the extractor $\mathsf{Extract}$ is also allowed to abort (i.e. output a "$\perp$" state) and we require that either the extractor returns "$\perp$" or it returns a witness of a certain quality. One can formalize this by assuming that the output of extractor $\mathsf{Extract}$ is measured according to $\{|\perp\rangle\langle\perp|, \mathbb{I} - |\perp\rangle\langle\perp|\}$. Finally, we require the outcome of this measurement corresponding to non-abort state (σ) be least inverse-polynomial probability, and that, conditioned on the latter outcome, the post-measurement state be a witness (of a certain quality).

[4] This definition can be easily extended to the CRS model. In that case the extractor would have the additional power of programming the CRS. We make this algorithm explicit when it is required for our constructions.

Definition 8 (Proof of Quantum Knowledge [39, 40]). *Let $R_{Q,\gamma}$ be a* QMA *relation. A proof system (P, V) is a Proof of Quantum Knowledge for $R_{Q,\gamma}$ with knowledge error $\kappa(n) > 0$ and quality q, if there exists a polynomial $p > 0$ and a polynomial-time machine* Extract *such that for any quantum interactive machine P^* that makes V accept some instance x of size n with probability at least $\varepsilon > \kappa(n)$, we have:*

$$Pr\left[\left((x,\sigma) \in R_{Q,q(\varepsilon,\frac{1}{n})}\right) : \sigma \leftarrow \mathsf{Extract}^{|P^*(x,\rho)\rangle}(x)\right] \geq p\left(\varepsilon - \kappa(n), \frac{1}{n}\right).$$

3 Impossibility of Simulation-based Quantum Two-Party Computation over Classical Channel

In this section, we present a relationship between the simulator Sim in black-box (quantum) 2-party computation (Definition 9) and extractor Extract from proofs of (quantum) knowledge (Definition 5, Definition 8). For sake of simplicity, we only present the results for proof of knowledge but a similar result holds for arguments of knowledge.

We consider the quantum zero-knowledge functionality $\mathcal{F}_{\mathsf{QZK}}$ that is parametrised by QMA-relation $R_{Q,\gamma}$ (Definition 7) and takes inputs only from one party called the prover. Given input statement x and a witness ρ, $\mathcal{F}_{\mathsf{QZK}}$ checks if $(x, \rho) \in R_{Q,\gamma}$. If that is the case, then $\mathcal{F}_{\mathsf{QZK}}$ outputs 1 to the second party (called verifier) with probability γ, else it outputs 0.

Theorem 1. *Any 2-party quantum computation protocol Π that realizes the functionality $\mathcal{F}_{\mathsf{QZK}}$ accordingly to Definition 1 is also a proof of quantum knowledge (acc. to Definition 8) with negligible knowledge error κ and non-negligible quality γ.*

Proof. To prove our theorem we need to show that Π satisfies the property of completeness and knowledge soundness. The property of completeness comes immediately from the Definition 1. Hence, we just need to exhibit an extractor Extract. By assumption, we know that for every malicious party acting as the prover $\mathcal{P}^\star$ in Π, there exists an QPT simulator $\mathsf{Sim}_{\mathcal{P}}$ such that:

$$\{\mathsf{REAL}_{\Pi,\mathcal{P}^\star(z)}(1^\lambda)\}_{z\in\{0,1\}^\lambda} \approx \{\mathsf{IDEAL}_{\mathcal{F}_{\mathsf{QZK}},\mathsf{Sim}_{\mathcal{P}}(z)}(1^\lambda)\}_{z\in\{0,1\}^\lambda}$$

where $\mathsf{REAL}_{\Pi,\mathcal{P}^\star(z)}(1^\lambda)$ denotes the distribution of the output of the adversarial prover $\mathcal{P}^\star$ after a real execution of protocol Π. $\mathsf{IDEAL}_{\mathcal{F}_{\mathsf{QZK}},\mathsf{Sim}_{\mathcal{P}}(z)}(1^\lambda)$ denotes the analogous distribution in an ideal execution with a trusted party that computes $\mathcal{F}_{\mathsf{QZK}}$ for the parties and hands the output to them.

Our proof of quantum knowledge extractor Extract simply runs $\mathsf{Sim}_{\mathcal{P}}$, and when $\mathsf{Sim}_{\mathcal{P}}$ provides the input (x, ρ) to the ideal functionality $\mathcal{F}_{\mathsf{QZK}}$, Extract measures the final outcome with $\{|\bot\rangle\langle\bot|, \mathbb{I} - |\bot\rangle\langle\bot|\}$ and checks if $(x, \sigma) \in R_{Q,\gamma}$ where σ is the post-measurement state corresponding to the latter projector. We note that $\mathsf{Sim}_{\mathcal{P}}$ makes only black-box use of the adversary $\mathcal{P}$, hence, so does Extract (that makes black-box use of $\mathsf{Sim}_{\mathcal{P}}$).

We now just need to argue that Extract runs in QPT time and it extracts a valid witness with probability negligibly close to p, where p is the probability that the verifier, in the real world experiment, outputs 1. Without loss of generality, we assume that $|x|$ and λ are polynomially related. By assumption, we know that $\mathsf{Sim}_{\mathcal{P}}$ successfully extracts the input ρ such that $(x, \rho) \in R_{Q,\gamma}$ from the adversarial prover $\mathcal{P}^\star$ with probability (at least) negligible close to p. Indeed, if this does not hold, then the security of Π would not hold. Hence, the probability that $\mathsf{Sim}_{\mathcal{P}}$ successfully extracts a witness is negligibly close to the probability $p \cdot \gamma$, for which the verifier outputs 1 in the real world. □

Our result holds for arguments of knowledge and computational secure black-box two-party computation. A similar relationship between secure 2PC and proofs of knowledge holds in the classical setting. For completeness, we provide the full proof for the classical counterpart in Appendix D.

Definition 9 (Q2PC over classical channel). *An m-round (classical Alice, quantum Bob) quantum protocol* $\mathsf{P} = (\mathfrak{A}, \mathfrak{B}, m)$ *over classical communication channel consists of:*

1. *input spaces S_0 and S_0' consisting of inputs ψ_A and ψ_B for parties $\mathfrak{A}$ and $\mathfrak{B}$.*

2. *memory spaces* $\mathsf{S} \coloneqq (S_1, \ldots, S_m)$ *for* $\mathfrak{A}$ *and* $\mathsf{S}' \coloneqq (S'_1, \ldots, S'_m)$ *for* $\mathfrak{B}$ *and (classical) communication spaces* $\mathsf{N} \coloneqq (N_1, \ldots, N_m)$ *and* $\mathsf{N}' \coloneqq (N'_1, \ldots, N'_m)$.
3. *an m-tuple of stochastic operations* $(\mathcal{A}_1, \ldots, \mathcal{A}_m)$ *for* $\mathfrak{A}$*, where* $\mathcal{A}_1 : \mathsf{L}(S_0) \mapsto \mathsf{L}(S_1 \otimes N_1)$*, and* $\mathcal{A}_i : \mathsf{L}(S_{i-1} \otimes N'_{i-1}) \mapsto \mathsf{L}(S_i \otimes N_i)$*,* $(2 \leq i \leq m)$.
4. *an m-tuple of quantum operations* $(\mathcal{B}_1, \ldots, \mathcal{B}_m)$ *for* $\mathfrak{B}$*, where* $\mathcal{B}_i : \mathsf{L}(S'_{i-1} \otimes N_i) \mapsto \mathsf{L}(S'_i \otimes N'_i)$*,* $(1 \leq i \leq m-1)$ *and* $\mathcal{B}_m : \mathsf{L}(S'_{m-1} \otimes N_m) \mapsto \mathsf{L}(S'_m)$.

3.1 Security of Q2PC with Classical Channel

In this work, we only deal with security proofs in the black-box simulation paradigm. That is, we require the simulator to be an oracle algorithm that makes oracle calls to the adversary. Next, we elaborate on the limitations that such a high level of security brings forward for secure two-party quantum computation, especially when the interaction is restricted to be completely classical. This highlights a trade-off between the security and resources required for quantum primitives. A similar trade-off has been recently studied in [18] in the context of the composable security of delegation protocols.

Recall that the quantum two-party computation as defined in Definition 9 consists of a completely classical Alice and a quantum Bob. Since information-theoretic secure quantum two-party is not possible, the best one could hope for, is to weaken the security of one of the parties to be computational. This is discussed below in more detail.

Simulator for Malicious Quantum Bob implies an Extractor in Classical proofs [Argument] of Quantum Knowledge. In the previous section, we showed that (quantum) two-party computation implies proof of (quantum) knowledge. If we restrict the communication between the two parties (Alice and Bob) to be completely classical (as in Definition 9) then the security of quantum two-party computation would imply a classical argument of quantum knowledge between a classical verifier and a quantum prover. Essentially this means we require an extractor Extract that extracts or reconstructs the quantum witness ρ from a classical transcript. This seems unlikely as also mentioned in the recent work of Vidick and Zhang [36]. The authors propose a definition of classical proofs/argument of knowledge where the extractor is non-black-box i.e. extractor is allowed to make use of the prover's internal state by defining an abstract party called *intermediary*. Additionally, the new definition also comes with a property, relevant to our result, that is a nondestructive classical argument of quantum knowledge, which is proven to be impossible for nontrivial states. Roughly speaking, a protocol is called a *nondestructive* if all the intermediate measurements when acted on a quantum state ρ keep the state ρ unchanged.

Corollary 1. *Secure quantum two-party computation over classical channel with black-box simulation is not possible.*

Proof. We choose the same notations for Alice and Bob as presented in Definition 9 and assume that only Bob has input while Alice gets the output. Furthermore, we assume that Alice behaves honestly. This trivially satisfies the requirement for $\mathcal{F}_{\mathsf{QZK}}$ functionality (as required in Theorem 1). Let us define a cheating strategy in the following way. We take an unclonable state in the input space S'_0 of Bob, while the general quantum operations $(B_1, \ldots, B_m)$ performed on Bob's end are such that they are Identity on the quantum registers and generate random classical transcripts N'. More formally, $\mathcal{B}_i : \mathsf{L}(S'_{i-1} \otimes N_i) \mapsto \mathsf{L}(S'_{i-1} \otimes N'_i)$, for every $i \in [m]$. Such operation guarantees that Bob's initial input state in the register S'_0 remains unchanged during the intermediate quantum operations and therefore, the protocol satisfies the nondestructive property. Intuitively, all the intermediate state $(S'_1, \ldots, S'_m)$ on Bob's side remains the same and all the information about cheating strategy is contained in the classical messages N'. For this cheating strategy, the security of Q2PC in simulation-based notion (i.e. the existence of simulator) reduces to the existence of an extractor for a nondestructive classical argument of knowledge. The latter is impossible as shown in Claim 4.2 of [36]. □

Relaxing Security Definition: One-sided Simulation. Since Bob is computationally powerful it is reasonable to expect computational security against QPT Bob. There are two possible ways of going forward to relax the security definition. One is to consider the non-black-box extractors and the other is to require only input privacy against QPT Bob. It is important to note that our no-go result via the cheating strategy mentioned earlier also rules out the non-black-box extractors defined in [36]. Also, as mentioned earlier our approach in this work is to focus only on black-box simulation techniques for which we provide a negative result. A natural step in relaxing the two-sided simulation is to show what is commonly known as one-sided simulation. The notion of *one-sided simulation security* is usually used when only one party receives output while the other learns nothing. This captures exactly our setup, where only Alice receives the output of the computation, while Bob must learn nothing about Alice's input.

When a party should learn nothing, we can define privacy via indistinguishability similar to encryption. Specifically, it suffices to require that Bob is not able to distinguish between two protocol runs corresponding to two different inputs of Alice. That is, consider a protocol/functionality where Alice receives output while Bob learns nothing. Roughly speaking, when Bob is corrupted we require that Bob should not be able to learn anything about Alice's input and formalize this via *indistinguishability.* Before presenting a formal definition, we look into the case of security against dishonest Alice and the issues concerning black-box simulation-based definition. This is discussed in detail below.

Using Quantum Proof of Classical Knowledge as a Subroutine. When Alice is corrupted we require the existence of a simulator that can fully simulate its view. Ideally, we would like to present security against unbounded Alice or against computationally bounded QPT Alice. Following Theorem 1 for our setup of classical Alice this implies the existence of an extractor for the quantum proof of classical knowledge or quantum arguments of classical knowledge, respectively. Defining and invoking such proofs/arguments of knowledge against quantum adversaries bring in different levels of difficulty as highlighted in [38].

To elaborate further, classically, the probability of a successful extraction by an extractor, as defined in 5, could be increased by repeating the procedure (polynomially) many times. However, a similar trick does not work straightforwardly for a general quantum prover because the measurement performed to extract the witness state could possibly disturb the internal state of the prover, making it impossible to simulate the side information that the prover had originally in the subsequent simulations. The implication of not being able to amplify the extractor's success probability restricts in invoking the quantum proof [argument] of knowledge as a subroutine.

Security against Alice. However, due to difficulty in obtaining proper quantum proofs [argument] of knowledge, in this work, we discuss the simulation-based security against computationally-bounded PPT Alice. However, our results should be easily lifted to the case of statistical or QPT adversary if there is any further advancement in the quantum proof or argument of knowledge (which can be used as a subroutine) in the future. One could also in principle obtain input privacy against Alice to obtain statistical input-privacy or post-quantum security. Since Alice is classical and given the way we construct our protocol[5], we believe it is rather straightforward for our concrete protocol to be generalized. For the rest of the paper, we are only concerned with the non-trivial case of simulation-based security.

Having discussed all the security issues, we present the one-sided simulation security definition for a one-sided[6] quantum two-party computation that is used in the remaining part of this work.

3.2 One-sided Simulation.

Let $\mathcal{F}$ be a joint quantum function $\mathcal{F} : \mathsf{L}(\mathcal{A}_{in}, \mathcal{B}_{in}) \mapsto \mathsf{L}(\mathcal{A}_{out})$ with input x and y from Alice ($\mathfrak{A}$) and Bob ($\mathfrak{B}$), resp. and output only on Alice's side. Let $\mathsf{REAL}_{\Pi,\mathcal{A}(z),i}(x, y, 1^\lambda)$ and $\mathsf{IDEAL}_{\mathcal{F},\mathcal{S}(z),i}(x, y, 1^\lambda)$ be the output in the real and ideal execution for $\mathcal{F}$ when the adversary $\mathcal{A}$ is controlling party $i \in \{\mathfrak{A}, \mathfrak{B}\}$ with the the auxiliary input z. The $view^{\mathcal{A},i}_{\Pi,\mathcal{A}(z)}(x, y, z, 1^\lambda)$ denote the view of the adversary $\mathcal{A}$ after a real execution of Π.

Definition 10 (One-Sided Secure Realization of $\mathcal{F}$)**.** *We say a protocol* $\Pi = (\mathfrak{A}, \mathfrak{B})$ *securely computes* $\mathcal{F}$ *with one-sided simulation (adapted from Def.2.6.2 in [44])) if for all inputs* $(x, y) \in D(A_{in} \otimes B_{in} \otimes R)$ *we have:*

[5] Bob one-time pads all the messages sent to Alice in the Protocol 5.2 and Protocol 7.1 while still preserving the correctness. Therefore, the input privacy of Bob against a malicious Alice reduces to analysing the information gain only for the final round.

[6] Our work only deals with one-sided computation but the definition can be easily generalized to the setting where both parties receive output.

1. *For every non-uniform PPT adversary $\mathcal{A}$ controlling $\mathfrak{A}$ in the real model, there exists a non-uniform PPT adversary $\mathcal{S}$ for the ideal model, such that:*
$$\{\mathsf{REAL}_{\Pi,\mathcal{A}(z),\mathfrak{A}}(x,y,1^\lambda)\}_{x,y,z,\lambda} \approx_c \{\mathsf{IDEAL}_{\mathcal{F},\mathcal{S}(z),\mathfrak{A}}(x,y,1^\lambda)\}_{x,y,z,\lambda} \tag{1}$$

2. *For every non-uniform QPT adversary $\mathcal{A}$ controlling $\mathfrak{B}$, we have that the following distributions are indistinguishable to any QPT distinguisher $\mathcal{D}$:*
$$\begin{aligned}&\{view^{\mathcal{A},\mathfrak{B}}_{\Pi,\mathcal{A}(z)}(x,y,z,1^\lambda)\}_{x,x',y,z,\lambda} \approx_q \{view^{\mathcal{A},\mathfrak{B}}_{\Pi,\mathcal{A}(z)}(x',y,1^\lambda)\}_{x,x',y,z,\lambda}\\ &\textit{where } |x| = |x'|.\end{aligned} \tag{2}$$

4 One-sided Two-party Quantum Computation over Classical Channel

In the rest of the work, we will consider a special case of two-party quantum computation that is one-sided, meaning that Alice has the input and Bob controls the function to be performed on the input. Inspired by the secure function evaluation functionality in classical world, we define an m-round, one-sided, two-party quantum functionality over classical channel and call it *oblivious quantum function evaluation* (OQFE)-(Definition 9). The ideal functionality for OQFE is formally defined as follows.

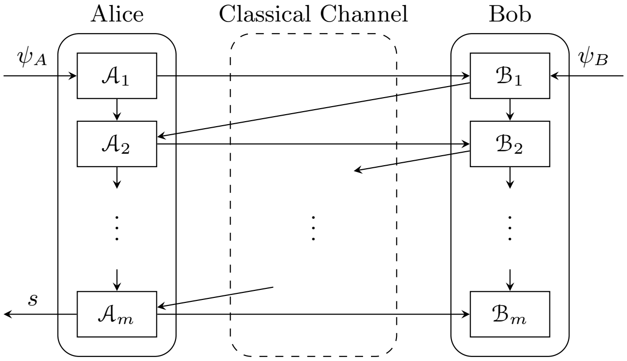


Figure 2: One-sided two-party quantum protocol with classical Alice and quantum Bob. The left and the right box represents the parties Alice (classical) and Bob (quantum) and the center box depicts the classical communication channel between them.

Definition 11. *(Ideal Functionality $\mathcal{F}_{\mathsf{OQFE}}$) An Oblivious Quantum Function Evaluation functionality $\mathcal{F}_{\mathsf{OQFE}}$ is defined as follows: When both the parties ($\mathfrak{A}$ and $\mathfrak{B}$) are honest then Oblivious Quantum Function Evaluation takes as input (a classical description of) a quantum function f from Alice and a quantum state ψ from Bob, respectively and outputs $f(\psi)$ at Alice's side. This is represented pictorially in Figure 1.*

In the previous section, we showed that it is not possible to attain the full security of Definition 1 for a quantum two-party protocol over a classical network. To show this we presented an adversary $\mathfrak{B}'$ in P for which there cannot exist any simulator Sim_B in the ideal world independent of the (statistical or computational) indistinguishability measure. We note that the same no-go argument remains valid for OQFE as well. As mentioned before, a natural step forward is to relax the security requirement to one-sided simulation. We present a concrete OQFE protocol and analyse the security in the case when either of the party is malicious. In particular, we show simulation-based security against a PPT Alice and input privacy against a QPT Bob. Such compromise seems inevitable given our previous result and the challenges to construct a quantum proof of knowledge, which shows that full simulation-based security in the presence of malicious adversaries is not possible.

4.1 Simplified Scenario: 1-out-of-2 OQFE

Before describing a candidate construction for oblivious quantum function evaluation (OQFE) that satisfies the above definition, we introduce a simplified functionality 1-out-of-2 oblivious quantum function evaluation, which can be understood as Alice having 2 possible functions f_0 and f_1 that she can choose from. The reason for introducing 1-out-of-2-OQFE is that it is simple in construction and security analysis and sets the stage for more complex OQFE functionality.

We model 1-out-of-2 oblivious quantum function evaluation functionality, denoted as Ξ_{OQFE}, in the following way: Bob's private quantum computation is parameterized by a quantum state $|\psi_{in}\rangle$ and a set of angles (ϕ_0, ϕ_1). The measurement outputs s_i are then given by the computational basis measurement of the unitary $R_x(\phi_i)$ on the input state $|\psi_{in}\rangle$ for $i \in \{0, 1\}$. For simplicity, we choose $\phi_0 = 0$ and $\phi_1 = \pi/2$. Therefore, the functionality Ξ_{OQFE} is given by:

$$((s_0, s_1), b) \rightarrow (\lambda, s_b, \epsilon) \tag{3}$$

where λ denotes the empty string, (1- ϵ) is the probability of Alice obtaining the correct outcome and:

$$\begin{aligned} s_0 &= M_Z |\psi_{in}\rangle, \\ s_1 &= M_Z R_x\left(-\frac{\pi}{2}\right) |\psi_{in}\rangle \end{aligned} \tag{4}$$

4.2 Ideal Functionality: 1-out-of-2 OQFE

Definition 12. *(Ideal Functionality Ξ_{OQFE}) A 1-out-of-2 Oblivious Quantum Function Evaluation functionality Ξ_{OQFE} is defined as follows. When both the parties ($\mathfrak{A}$ and $\mathfrak{B}$) are honest then 1-out-of-2 Oblivious Quantum Function Evaluation takes the input b and (s_0, s_1) from Alice and Bob, respectively and outputs s_b at Alice's side. Here (s_0, s_1) corresponds to measurement output of Bob's (server's) private quantum computation and b denotes the choice of computation Alice (user) wishes to retrieve.*

In the following sections, we give two constructions for 1-out-of-2-OQFE protocols in which b and (s_0, s_1) are the inputs of Alice and Bob, respectively, and s_b is the output of Alice.

1. The first protocol, denoted as π_{SH}, is non-interactive in nature i.e. requires only two-message between Alice and Bob and we prove the simulation-based security against any semi-honest adversaries as well as privacy against malicious Bob.

2. Building upon the first construction, the second protocol, denoted as π_{MAL}, crucially relies on the properties of a zero-knowledge argument of knowledge as well as commitment schemes to achieve simulation-based security against fully malicious Alice. This, however, comes at the cost of an increase in the number of rounds for the overall protocol.

The most important feature of these protocols is that Alice and Bob only need to share a classical channel between them and are modular in design. This means advancement in any of the techniques can easily upgrade the security of our protocols. To the best of our knowledge, this is the first secure construction of any classical client - quantum server two-party quantum computation.

5 1-out-of-2 OQFE: Semi-Honest Alice

Our first construction is a 2-message, non-interactive protocol for semi-honest Alice and can be further seen as a two-step process: a simple 2-party quantum computing protocol over a quantum channel inspired from one-bit teleportation circuit [45, 46] and replacing the quantum channel with a classical-client remote state preparation protocol [1, 17]. The protocol consists of two parties - Alice and Bob, where Alice's input is represented by an index i and Bob have a quantum input $|\psi_{in}\rangle$. As mentioned before, we will consider Alice to be the party who receives the output at the end of the protocol. Without loss of generality, we assume that the goal of Alice and Bob is to perform a unitary operation U_i from a set $\mathcal{U}$ on Bob's private input state $|\psi_{in}\rangle$. Here, the set of unitaries $\mathcal{U} := \{U_i\}_i$ is known to both parties and we denote the output of the joint computation as $|\psi_{out}\rangle$, where $|\psi_{out}\rangle := U_i\,|\psi_{in}\rangle$. We emphasize that our protocols hold for Bob having as input a mixed state as well, but for simplicity of presentation, we will denote Bob's input as a pure state.

For simplicity, we present the protocol where Alice's input is denoted with a bit and Bob's input is a single qubit quantum state. Alice's input bit b is encoded using an angle ϕ_b, where the bit $b = 0$ and $b = 1$ corresponds to the angle $\phi_0 = 0$ and $\phi_1 = \pi/2$, respectively. Bob's private input is ρ_{in}. In step 1 and step 2, Alice encrypts her private bit b (parameterized with the angle ϕ_b) using one-time pad with random key θ_2 and r_A in the following way:

$$\delta := \phi_b + \theta_2 \cdot \frac{\pi}{2} + r_A \cdot \pi = (b + \theta_2) \cdot \frac{\pi}{2} + r_A \cdot \pi = \frac{\pi}{2} \cdot [b + \theta_2 + 2r_A \bmod 4] \tag{5}$$

where θ_2 and r_A are uniformly chosen random bits. Hereafter, we will drop the $\pi/2$ coefficient and refer to δ as $\delta = b + \theta_2 + 2r_A \bmod 4$. Alice remotely prepares on Bob's side the following single qubit state:

$$|\psi_A\rangle = |+_\theta\rangle := \frac{1}{\sqrt{2}}(|0\rangle + e^{i\theta}\,|1\rangle) \tag{6}$$

where $\theta \in \{0, \pi/2, \pi, 3\pi/2\}$ is an angle represented as a two bit string $(\theta_1, \theta_2) \in \{0,1\}^2$, where θ_2 is exactly the bit used above by Alice. This is achieved by following the steps described in the remote state protocol (RSP), presented in Protocol 5.1 [7]. Alice transmits to Bob the classical message δ along with the classical messages required in Protocol 5.1.

Protocol 5.1 4-states QFactory: classical delegation of $|+_\theta\rangle$ states ($\theta \in \{0, \pi/2, \pi, 3\pi/2\}$) ([1])

Requirements: A 2-regular trapdoor one-way family $\mathcal{F}$ and homomorphic hardcore predicate $\{h_k\}$.

1. Preimages superposition
   - (a) Alice runs the algorithm $(k, t_k) \leftarrow \text{Gen}_{\mathcal{F}}(1^n)$.
   - (b) Alice instructs Bob to prepare one register at $\otimes^n H\,|0\rangle$ and second register initiated at $|0\rangle^m$.
   - (c) Bob receives k from Alice and applies U_{f_k} using the first register as control and the second as target.
   - (d) Bob measures the second register in the computational basis, obtains the outcome y. The combined state is given by $(|x\rangle + |x'\rangle) \otimes |y\rangle$ with $f_k(x) = f_k(x') = y$ and $y \in \text{Im}\, f_k$.
2. Output preparation
   - (a) Bob applies U_{h_k} on the preimage register $|x\rangle + |x'\rangle$ as control and another qubit initiated at $|0\rangle$ as target. Then, measures all the qubits, but the target in the $\{\frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)\}$ basis, obtaining the outcome $b = (b_1, ..., b_n)$. Bob applies on the unmeasured qubit (representing the output state) the operation $HR(-\pi/2)$. Now, Bob returns both y and b to Alice.
   - (b) Alice using the trapdoor t_k computes the preimages of y:
   - (c) Then compute: $\theta_2 := h_k(x) \oplus h_k(x')$ and $\theta_1 := (\theta_2 \cdot \langle b, x \oplus x'\rangle) \oplus h_k(x)h_k(x')$
3. **Outputs**: The quantum state that Bob has generated is (with overwhelming probability [8]) the state $|+_\theta\rangle$, where $\theta \in \{0, \pi/2, \pi, 3\pi/2\}$ state described using the two bits (θ_1, θ_2), where θ_2 is also known as the basis of the state. The output of Alice is the classical description (θ_1, θ_2).

In step 3, upon receiving the messages from Alice, Bob applies on the state $|\psi_A\rangle$ (resulted from the RSP procedure) a controlled Pauli X where the control is a random bit r_B and also updates the classical message δ to $\delta' := (-1)^{r_B}\delta$. These operations enable Bob to ensure security against malicious Alice while preserving correctness. The purpose of this step is to hide the classical messages sent from Bob during the run of the protocol. In step 4, Bob initializes an ancillary register, $|\psi_B\rangle$, in the $|+\rangle$ state and performs entangling operations, controlled-Z (CZ) gates, as shown in Fig. 3. In the end, Bob measures the first two registers corresponding to his private input state and to the quantum state obtained from the RSP procedure. The two measurements are performed in the Hadamard and $\{|\pm_{\delta'}\rangle\}$ basis, respectively, where $|\pm_{\delta'}\rangle := R(-\delta')|\pm\rangle$

[7] It is important to emphasize the exact construction of remote state preparation used here relies on a two-way communication classical channel.

[8] the probability comes from the probability of $\mathcal{F}$ being a 2-regular homomorphic-hardcore family of functions

and $R(-\delta')$ is the rotation around z-axis with the angle δ'. This step results in the measurement outcomes m_0 and m_1 on Bob's side. Finally, Bob performs a correction operator $Z^{m_0}X^{m_1}$ on the ancillary register and measures it in computational basis to obtain $\bar{s_b}$. Bob sends $m_0 \oplus r_B$ and $\bar{s_b}$ to Alice.

Protocol 5.2 1-out-of-2 OQFE Protocol, π_{SH}, with Semi-Honest Classical Alice

Inputs: Bob: single qubit state $|\psi_{in}\rangle$ and Alice: $b \in \{0,1\}$
Output: Alice: s_b, where $s_b := M_Z[R_x\left(-\frac{\pi}{2} \cdot b\right)|\psi_{in}\rangle]$

1. Alice and Bob follows Step 1 (a) - (d) and Step 2 (a) of the Protocol 5.1. Alice obtains θ_2 and Bob obtains $|\psi_A\rangle$ (See, Eq. 6).
2. Alice encodes her input as $\phi_b := b \cdot \frac{\pi}{2}$, uniformly samples $r_A \overset{\$}{\leftarrow} \{0,1\}$, computes the angle δ (See, Eq. 5) and sends δ to Bob.
3. Bob samples uniformly $r_B \overset{\$}{\leftarrow} \{0,1\}$ and updates the quantum state, $|\psi_A\rangle$, as well as the classical message, δ, to obtain: $|\psi_0\rangle := X^{r_B}|\psi_A\rangle$ and $\delta' := (-1)^{r_B}\delta$.
4. Bob performs the following entangling operations on his private input register $|\psi_{in}\rangle$, the updated state $|\psi_0\rangle$, and the ancillary register $|\psi_B\rangle$: $(\mathbb{I} \otimes CZ)(CZ \otimes \mathbb{I})(|\psi_{in}\rangle \otimes |\psi_0\rangle \otimes |\psi_B\rangle)$, where $|\psi_0\rangle$ is in the state $|+\rangle$.
5. Bob performs the measurement of first register in the X-basis and the second register in the (X,Y)-plane with an angle δ' to obtain the measurement outcomes $m_0 \in \{0,1\}$ and $m_1 \in \{0,1\}$.
6. Bob applies $X^{m_1}Z^{m_0}$ to the resulting quantum state to obtain:
$$|out_b\rangle = X^{\theta_1 \oplus r_A \oplus (m_0 \oplus r_B)\cdot b}[HR(-\phi_b)H|\psi_{in}\rangle] \tag{7}$$
7. Bob measures $|out_b\rangle$ in the computational basis and obtains a measurement outcome $\bar{s_b}$, updates $m_0' := m_0 \oplus r_B$ and sends $(m_0', \bar{s_b})$ to Alice.
8. Alice computes (efficiently) θ_1 from m_{qf} and the trapdoor t_k and performs the following (classical) operation to get her desired outcome: $s_b := \bar{s_b} \oplus \theta_1 \oplus r_A \oplus m_0' \cdot b$.

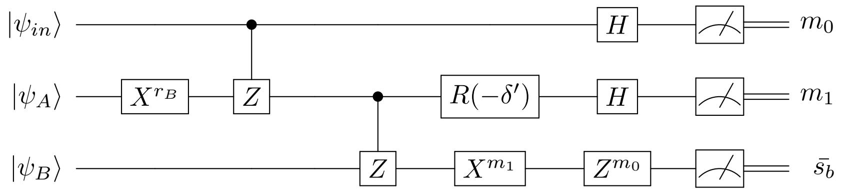


Figure 3: Quantum computations performed by Bob in steps 4-8 of Protocol 5.2.

Theorem 2 (Correctness)**.** *In an honest run of* 1*-out-of-*2 OQFE *Protocol 5.2, when both parties follow the protocol specifications, Alice obtains the outcome* $s_b = M_Z[R_x\left(-b \cdot \frac{\pi}{2}\right)|\psi_{in}\rangle]$*, where* b *is Alice's input and* $|\psi_{in}\rangle$ *is Bob's input.*

Proof Sketch. Protocol 5.2 can be seen as two consecutive run of one-bit teleportation circuit [45] distributed between Alice and Bob. In particular, Alice controls the quantum state of the second wire and the measurement angle while Bob controls the state of the first of the circuit as shown in Figure 3. The state of the third wire is fixed in $|+\rangle$ state and known to both the parties. We first show that there is no effect of Bob's randomness r_B (Step 3) on the correctness. Recall that $R(-\delta') = X^{r_B}R(-\delta)$ and moving $R(-\delta')$ on the second wire towards left-hand side to merge with X^{r_B} will cancel the randomness added by Bob on the second wire, however this commutation will give rise to an additional Pauli term Z^{r_B} on the first wire. This is because of the following identity: $CZ_{12}(\mathbb{I} \otimes X^{r_B})_{12}(\mathbb{I} \otimes R(-\delta))_{12} = (Z^{r_B} \otimes X^{r_B})_{12}(\mathbb{I} \otimes R(-\delta))_{12}CZ_{12}$. This additional Pauli byproduct Z^{r_B}, after commuting with Hadamard towards the right, essentially dictates the need for correction at Bob's end in Step 7. Next, we note that $|\psi_A\rangle = |+_\theta\rangle$ (by Eqn. 6), where θ is Alice's private key. The effect of this randomness is partially cancelled by the $R(-\delta)$ as $R(-\delta)|+_\theta\rangle = R(-\delta + \theta)|+\rangle$ and the rest of the randomness is carried over the measurement outcome which is corrected by Alice (Step 8). Roughly speaking, these two steps allows Alice and Bob to hide their input while still maintaining correctness. For full proof see Appendix E.

Theorem 3 (Simulation-based statistical security against semi-honest Alice)**.** *The* 1*-out-of-*2 OQFE *Protocol,* π_{SH}*, (Protocol 5.2) securely computes* Ξ_{OQFE} *in the presence of semi-honest adversary Alice.*

Proof. We need to prove that there exists a PPT simulator $\mathcal{S}^A$ that given Alice's input and output can simulate the view of Alice in π_{SH}, such that the output of the simulator and the real view of Alice are indistinguishable to any *unbounded distinguisher*:

$$\{\mathcal{S}^A(b, s_b)\}_{\substack{b\in\{0,1\}\\ |\psi_{in}\rangle\in\mathcal{H}_2}} \approx_u \{view_A^{\pi_{\mathsf{SH}}}(b, |\psi_{in}\rangle)\}_{\substack{b\in\{0,1\}\\ |\psi_{in}\rangle\in\mathcal{H}_2}} \tag{8}$$

The view of Alice in π_{SH} consists of an input b, the randomness r^A and all the messages received from Bob during the protocol. More specifically, we have:

$$view_A^{\pi_{\mathsf{SH}}}(b, |\psi_{in}\rangle)\} = \{b, r^A, (m_{qf}, m'_0, \bar{s}_b)\} \tag{9}$$

The internal random coins r^A consists of the random bit r_A and the randomness used for the algorithm $Gen_{\mathcal{F}}$, denoted by r^A_f.

We construct the simulator $\mathcal{S}^A$ in the following way:

```
$\mathcal{S}^A(b, s_b)$
1 :  $\tilde{r}^A \leftarrow_\$ \mathcal{R}^A$ // $\mathcal{R}^A$ is the space of randomness of Alice, and $\tilde{r}^A$ includes $\tilde{r}^A_f$ and $\tilde{r}_A$
2 :  $\tilde{\bar{s}}_b \leftarrow_\$ \{0,1\}$
3 :  $(\tilde{k}, \tilde{t}_k, hp) \leftarrow Gen_{\mathcal{F}}(1^\lambda)$
4 :  $\tilde{y} \leftarrow_\$ Im(f_{\tilde{k}})$ , $\tilde{m} \leftarrow_\$ \{0,1\}^n$
5 :  $\tilde{m}_{qf} \leftarrow (\tilde{y}, \tilde{m})$
6 :  $\tilde{\theta}_1 \leftarrow Inv_{\mathcal{F}}(\tilde{m}_{qf}, \tilde{t}_k)$
7 :  if $(b == 1)$ then
8 :    $\tilde{\gamma} \leftarrow \tilde{\bar{s}}_b \oplus s_b \oplus \tilde{\theta}_1 \oplus \tilde{r}_A$ // $\tilde{\gamma} := \tilde{m}'_0 \cdot b$ can be computed from Step 8 of Protocol 5.2.
9 :    $\tilde{m}'_0 \leftarrow \tilde{\gamma}$
10 : else
11 :   $\tilde{\gamma} \leftarrow_\$ \{0,1\}$
12 :   $\tilde{m}'_0 \leftarrow \tilde{\gamma}$
13 : return $(b, \tilde{r}^A, (\tilde{m}_{qf}, \tilde{m}'_0, \tilde{\bar{s}}_b))$
```

In order to prove that the output of $\mathcal{S}^A$ and the real view of Alice are indistinguishable, we need to show that the following distributions D_1 and D_2 are indistinguishable (where D_1 corresponds to Alice's view and D_2 with $\mathcal{S}^A$'s output):

$$\begin{aligned} D_1 &= \{b, r_A, m_{qf}, m'_0, \bar{s}_b\}_{b,|\psi_{in}\rangle} \\ D_2 &= \{b, \tilde{r}_A, \tilde{m}_{qf}, \tilde{m}'_0, \tilde{\bar{s}}_b\}_{b,|\psi_{in}\rangle} \end{aligned} \tag{10}$$

From step 7 of simulator $\mathcal{S}^A$ and Step 8 of Protocol 5.2, we can also write the distribution D_1 and D_2 as:

$$\begin{aligned} D'_1 &= \{b, r_A, m_{qf}, m'_0, \bar{s}_b\}_{b,|\psi_{in}\rangle} \\ D'_2 &= \{b, \tilde{r}_A, \tilde{m}_{qf}, \tilde{\gamma}, \tilde{\bar{s}}_b\}_{b,|\psi_{in}\rangle} \end{aligned} \tag{11}$$

Independent of the value of the bit b, m'_0 and $\tilde{\gamma}$ are indistinguishable.

Since (r_A, $\tilde{r}_A$) and ($\bar{s}_b$, $\tilde{\bar{s}}_b$) are sampled uniformly and independently at random from {0,1}, the task of the distinguisher can be equivalently seen as distinguishing these two distributions:

$$\begin{aligned} D''_1 &= \{b, m_{qf}, s_b \oplus \theta_1\}_{b,|\psi_{in}\rangle} \\ D''_2 &= \{b, \tilde{m}_{qf}, s_b \oplus \tilde{\theta}_1\}_{b,|\psi_{in}\rangle} \end{aligned} \tag{12}$$

Since θ_1 and $\tilde{\theta}_1$ are generated from m_{qf} and $\tilde{m}_{qf}$, respectively, which are sampled uniformly and independently at random, the task of distinguisher can be equivalently seen as distinguishing these two distributions:

$$\begin{aligned} D_1''' &= \{b, m_{qf}\}_{b,|\psi_{in}\rangle} \\ D_2''' &= \{b, \tilde{m}_{qf}\}_{b,|\psi_{in}\rangle} \end{aligned} \tag{13}$$

Finally, in the real protocol m_{qf} consists of $y \in Im(f_k)$ and a bitstring $m \in \{0,1\}^n$. They represent outcomes of Bob's measurements inside QFactory protocol (thus before $|\psi_{in}\rangle$ was even used) and irrespective of b, in an honest run they occur with equal probability: $\Pr[y] = \frac{1}{|\mathrm{Im}\, f|}$, $\Pr[m] = \frac{1}{2^n}$. Therefore, as in $\tilde{m_{qf}}$ we sample $\tilde{y}$ uniformly from the the image of $f_{k'}$ and $\tilde{m}$ uniformly at random from $\{0,1\}^n$, this makes m_{qf} and $\tilde{m_{qf}}$ statistically indistinguishable. This shows that D_1' and D_2' are indistinguishable against the unbounded distinguished $\mathcal{D}$, which concludes the proof.

□

Theorem 4 (Privacy against Malicious Bob)**.** *The* 1*-out-of-*2 OQFE *Protocol 5.2* π_{SH} *is private against malicious Bob.*

Proof. To show that the π_{SH} is private against Bob, it suffices to show that Bob cannot distinguish between the cases when Alice has input $b = 0$ or input $b = 1$.

In other words, we need to show that: For any QPT *Bob** (interacting with Alice in π_{SH}) and for any auxiliary input z, we have:

$$\{view_{Bob^*}(Bob^*(z), Alice(0))\} \approx_q \{view_{Bob^*}(Bob^*(z), Alice(1))\} \tag{14}$$

The view of a malicious *Bob** in π_{SH} when he has auxiliary input z and Alice has input b is defined as:

$$view_{Bob^*}(Bob^*(z), Alice(b)) = (z, r^B, (k, \delta)) \tag{15}$$

where r^B is the randomness of *Bob**.
The transcript received by *Bob** from Alice during π_{SH} consists of:

1. k - the public key obtained by Alice when running $Gen_{\mathcal{F}}$ corresponding to a 2-regular trapdoor (post-quantum) function f_k;
2. δ - represents the measurement angle that Bob is instructed to use in the quantum computations he is performing in Stage 3 of the Protocol 5.2;

The internal random tape of *Bob** contains r_B. We will prove by contradiction that if there exists a QPT distinguisher that can distinguish between the 2 views of *Bob** in the cases Alice's input is $b = 0$ and respectively $b = 1$, then there exists a QPT algorithm that can break the *4-states basis blindness* property (Definition 21) of QFactory, or equivalently, the hardcore property of the basis bit θ_2. More specifically, we assume that there exists a QPT algorithm $\mathcal{A}$ that on input (r^B, k, δ) can output Alice's input b with probability $\frac{1}{2} + \frac{1}{p}$ and we will construct an algorithm $\mathcal{A}'$ that can break the hardcore property of the basis θ_2 with probability $\frac{1}{2} + \frac{1}{p}$. This implies that if $\mathcal{A}$ succeeds to distinguish the 2 views with inverse polynomial probability, the same applies to the hard-core predicate property, and hence we reach a contradiction.

$\mathcal{A}'(k)$

```
$r_B \leftarrow_\$ \{0,1\}$ , $r_A \leftarrow_\$ \{0,1\}$ , $b \leftarrow_\$ \{0,1\}$ , $B_2 \leftarrow_\$ \{0,1\}$
$\tilde{\delta} \leftarrow b + B_2 + 2r_A \bmod 4$
$\tilde{b} \leftarrow \mathcal{A}(r_B, k, \tilde{\delta})$
if $(\tilde{b} = b)$ then $\tilde{\theta_2} \leftarrow B_2$
else $\tilde{\theta_2} \leftarrow B_2 \oplus 1$
return $\tilde{\theta_2}$
```

Now to compute the probability that $\mathcal{A}$ break the hardcore predicate, we first consider 2 cases: i) $B_2 = \theta_2$ and ii) $B_2 \neq \theta_2$, where as B_2 is sampled uniformly, each occur with probability $\frac{1}{2}$. The first case corresponds to the view of the protocol when Alice's input is b and the second case corresponds to the view of the protocol when Alice's input is $1 \oplus b$. Therefore, we have:

$$\begin{aligned}
&Pr[\mathcal{A}'(k) = \theta_2] = Pr[\mathcal{A}'(k) = \theta_2 \,|\, B_2 = \theta_2] \cdot Pr[B_2 = \theta_2] + \\
&+ Pr[\mathcal{A}'(k) = \theta_2 \,|\, B_2 = 1 \oplus \theta_2] \cdot Pr[B_2 = 1 \oplus \theta_2] \\
&= \frac{1}{2}(Pr[\mathcal{A} \text{ outputs } b \,|\, \text{Alice input} = b] + Pr[\mathcal{A} \text{ outputs } 1 \oplus b \,|\, \text{Alice input} = 1 \oplus b]) \\
&= \frac{1}{2}\left(\frac{1}{2} + \frac{1}{p} + \frac{1}{2} + \frac{1}{p}\right) = \frac{1}{2} + \frac{1}{p}
\end{aligned}$$

□

6 1-out-of-2 OQFE: Malicious Alice

In this section, we present a protocol π_{MAL} that is one-sided simulatable against malicious parties.

The main reason why the previous Protocol 5.2 is not secure against a malicious Alice is that we have no guarantee that the key k sent in the first round represents a valid key. That is, Alice could generate k using an algorithm different from $Gen_{\mathcal{F}}$, or using $Gen_{\mathcal{F}}$ with some *bad randomness* (a non-uniformly chosen random string). To solve the first problem, we could just let Alice prove that k is a key computed using the algorithm $Gen_{\mathcal{F}}$ via a zero-knowledge protocol. In this way, Alice protects the trapdoor of k and Bob can be convinced about the validity of k. Unfortunately, this does not prevent Alice from using "bad" randomness. To tackle the second problem, we let Alice and Bob engage in a sort of coin-tossing protocol. In more details, Alice sends a commitment of a string r_f^A, Bob sends a random string r_f^B in the clear in the second round, and Alice proves, always using a zero-knowledge scheme, that the key k of the trapdoor function is generated accordingly to $Gen_{\mathcal{F}}$, using as the randomness the xor of the string committed from Alice (r_f^A) and the string sent from Bob (r_f^B).

Given that Bob is a quantum adversary, we need a post-quantum hiding commitment scheme in addition to the quantum zero-knowledge property. If we can employ a zero-knowledge proof in combination with a post-quantum hiding and statistically binding commitment we would get statistical security against malicious Alice in the one-sided simulation framework. Unfortunately, we are not aware of any post-quantum secure zero-knowledge proof of knowledge protocol

For this reason, in this section, we assume Alice to be a PPT adversary, hence we need the binding of the commitment scheme and the soundness of the zero-knowledge scheme to hold against PPT adversaries (argument of knowledge). The commitment scheme can be instantiated from any non-interactive statistically hiding commitment scheme. Alternatively, we can rely on the protocol proposed by Baum et al [47]. In this commitment scheme, the binding property is based on the Module-LWE assumption and the Hiding property is based on the Module-SIS assumption (which are generalizations of the LWE and SIS problems to polynomial rings), but under specific choices of parameters, the scheme can be made statistical binding. Since we want to prove that our protocol is simulatable against malicious Alice, we require the post-quantum secure zero-knowledge protocol to enjoy the property of the argument of knowledge. One zero-knowledge protocol that has all these features is the one proposed in [42].[9]

We now formally prove the security of our protocol against Alice requiring the output distribution of the simulator and the adversarial Alice to be computationally indistinguishable. However, we note that relying on a proof of knowledge zero-knowledge (instead of an argument of knowledge) it is easy to extend our result to the statistical case.

More precisely, our scheme makes use of the following tools:

1. Non-interactive post-quantum hiding, computationally binding commitment scheme $\mathsf{COM} = (Com, Dec)$;
2. A 2-regular Trapdoor One-Way Function $\mathcal{F} = (Gen_{\mathcal{F}}, Eval_{\mathcal{F}}, Inv_{\mathcal{F}})$.

[9] In [42] the authors do not explicitly prove the property of AoK for their protocol but acknowledge that the AoK extractor would work as for the protocol provided in [48].

3. An argument of knowledge post-quantum zero-knowledge protocol $\Pi := (G_{\mathsf{ZK}}, P_{\mathsf{ZK}}, V_{\mathsf{ZK}})$ for the NP-relation:
$$Rel_f = \{(x = (k, r_f^B, com_f), w = (r_f^A, dec_f)\}$$
such that dec_f is the decommitment of com_f and k is (part of) the output of probabilistic algorithm $Gen_{\mathcal{F}}$ when run with internal random coins $r_f = r_f^A \oplus r_f^B$.

We refer to the Protocol 6.1 for the formal description of our scheme.

Protocol 6.1 1-out-of-2 OQFE Protocol, π_{MAL}, against Malicious Alice

Common input: $\sigma \leftarrow G_{\mathsf{ZK}}(1^\lambda)$

Private Inputs:

1. Sender (Bob): single qubit state $|\psi_{in}\rangle$
2. Receiver (Alice): $b \in \{0, 1\}$

1. **Alice's Verification and Setup**
   - 1.1 Alice samples uniformly at random r_f^A from $\{0,1\}^\lambda$.
   - 1.2 Alice runs $Com(r_f^A) \rightarrow (com_f, dec_f)$ and sends com_f to Bob.
   - 1.3 Bob samples r_f^B uniformly at random from $\{0,1\}^\lambda$ and sends it to Alice.
   - 1.4 Alice now run computes $r_f = r_f^A \oplus r_f^B$. She then runs $Gen_{\mathcal{F}}$ using internal random coins r_f and obtains (k, t_k, hp).
   - 1.5 Alice now sends k to Bob and runs the interactive algorithm P_{ZK} on input the CRS σ, the statement to be proven $x = (k, r_f^B, com_f)$ and the witness $w = (r_f^A, dec_f)$.
   - 1.6 Bob runs the interactive algorithm V_{ZK} on input the CRS σ and the statement x. Let c be the output of V_{ZK}. If $c = 0$ then Bob aborts, otherwise he waits to receive another message from Alice.
   - 1.7 Alice assigns $\theta_2 = hp$ and encodes her input as $\phi_b := b \cdot \frac{\pi}{2}$, uniformly samples $r_A \xleftarrow{\$} \{0,1\}$ and computes the angle δ (See, Eq 5) and sends δ to Bob. Bob continues to the next stage.
2. **4-states QFactory** (Protocol 5.1)
3. **Computation on Bob's side** (Steps 5-8 of Protocol 5.2)
4. **Output on Alice's side** (Step 9 of Protocol 5.2)

Output: Alice obtains: $s_b = M_Z[R_x\left(-\frac{\pi}{2} \cdot b\right)|\psi_{in}\rangle]$.

Theorem 5. *Protocol π_{MAL} securely computes Ξ_{OQFE} with one-sided simulation.*

The correctness of the modified protocol (Theorem 2) follows from the correctness of the commitment scheme COM and the correctness of Π.

To complete the proof we need to prove the following two lemmata: Lemma 1 and Lemma 2.

Lemma 1 (Simulation-based (computational) security against malicious Alice)**.** *The protocol π_{MAL} is simulation-based secure against malicious Alice.*

Proof. We need to show that for any PPT adversary $Alice^*$, there exists a PPT adversary $\mathcal{S}$ for the ideal model such that:

$$\{\mathsf{IDEAL}_{\Xi_{\mathsf{OQFE}}, \mathcal{S}(z), Alice}(b, |\psi_{in}\rangle)\}_{b, |\psi_{in}\rangle, z} \approx_c \{\mathsf{REAL}_{\pi_{\mathsf{MAL}}, Alice^*(z), Alice}(b, |\psi_{in}\rangle)\}_{b, |\psi_{in}\rangle, z}$$

In other words, to show that π_{MAL} is simulation-based secure against malicious receiver $Alice^*$, we have to prove that there exists a PPT simulator S, that by having access only to the ideal functionality Ξ_{OQFE}, can simulate the output of any malicious $Alice^*$ who runs one execution of π_{MAL} with an honest sender Bob. The simulator S having oracle access to $Alice^*$, will run as a sender Bob in the real protocol π_{MAL}. By studying the real protocol, we notice that the 2 messages that $Alice^*$ sends to Bob (that she could be cheating about) are k (the public key of the 2-regular trapdoor function) and δ (the angle that $Alice^*$ instructs Bob to use in his computation and which should depend on her input b). One of the important elements to prove the security is the Argument of Knowledge property of Π, which guarantees the existence of an extractor that, given an acceptable proof for an NP statement x, it extracts the witness for x with overwhelming probability. At a high level, our simulator works as follows. Upon receiving a proof Π from *Alice**, the simulator runs the PoK extractor of Π thus obtaining r_f^A, computes $r_f = r_f^A \oplus r_f^B$ and runs the algorithm $Gen_{\mathcal{F}}$ with internal random coins r_f. The output of $Gen_{\mathcal{F}}$ corresponds to (k, t_k, hp), where $\theta_2 = hp$. Now using δ^* and θ_2, the simulator can compute $d := \delta^* - \theta_2 \bmod 4$ and finally to extracts b^* by computing $b^* = d \bmod 2$.

Before providing the formal description of our simulator, we assume, without loss of generality, that the AoK extractor of Π consists of two algorithms: (E_1, E_2). E_1, on input the security parameter outputs a CRS σ and an auxiliary information ξ (if any). E_2 on input σ, ξ and the theorem x interacts (in a black-box way) with the malicious prover to extract the witness for x. Formally, the simulator S does the following steps.

1. Computes $(\sigma, \xi) \leftarrow E_1(1^n)$ where σ represents the CRS
2. Receives com_f from *Alice**
3. Samples r_f^B uniformly at random and sends it to *Alice**
4. Receives k and defines the statement $x = (k, r_f^B, com_f)$
5. Runs $E_2(\sigma, \xi, x)$, where $x = (k, r_f^B, com_f)$ thus obtaining $w = (r_f^A, dec_f)$
6. Verifies the decommitment phase, namely if $Dec(com_f, r_f^A, dec_f) = 1$.
7. If not, this means that *Alice** cheated during the commitment and we abort
8. Computes $r_f = r_f^A \oplus r_f^B$
9. Runs $Gen_{\mathcal{F}}$ using the randomness r_f, and since the randomness is fixed the output of $Gen_{\mathcal{F}}$ is deterministic.
10. Denote this output $(\bar{k}, t_k, hp)$ and assign $\theta_2 = hp$, as in the real protocol
11. Upon receiving σ^*, computes $d = \delta^* - \theta_2 \mod 4$
12. Computes the input of *Alice** as: $b^* = d \mod 2$
13. Invokes the ideal functionality Ξ_{OQFE} on input b^* and obtains s_{b^*}
14. Runs the simulator for semi-honest Alice (see the proof of Theorem 4): $S^A(b^*, s_{b^*})$ using the randomness r_f to compute the last round

We now show this simulator S is a good simulator, i.e. the output of S is computationally indistinguishable from the output of *Alice** in the real-world experiment. We note that there are only two scenarios in which the simulator would fail:

1. *Alice**, to generate the public key k, uses a randomness $\bar{r}_f^A$ different than the one she committed to (r_f^A), or k is not in the domain of $Gen_{\mathcal{F}}$.
2. *Alice** biases the randomness used to run $Gen_{\mathcal{F}}$ by constructing an opening for the commitment which depends on r_f^B.

Loosely speaking, the simulator fails if *Alice** breaks the *binding* of the commitment or the argument of knowledge property of Π.

i In the first scenario, we can immediately use *Alice** to construct a reduction to the AoK property of Π.

ii In the second scenario, we want to use the malicious Alice to construct a reduction to the binding of the commitment scheme. The reduction works as follows.

   (a) Interact with *Alice** as Bob would do, and upon receiving the proof Π, run the extractor to obtain the opening of the commitment.

(b) Rewind $Alice^*$ and sample a randomness $r_f^{B'}$ such that $r_f^{B'} \neq r_f^B$. Then send again $r_f^{B'}$ to $Alice^*$.

(c) As a result we receive from $Alice^*$ a new tuple $(k', \delta^{*\prime})$.

(d) Run again the extractor of Π on input (σ, x', zkp') where $x' = (k', r_f^B, com_f)$ and obtain a new witness $w' = (r_f^{A'}, dec_f')$.

(e) If $r_f^{A'} \neq r_f^A$, then this means we have found 2 decommitments (r_f^A, dec_f) and $(r_f^{A'}, dec_f')$ for the same commitment com_f with $r_f^A \neq r_f^{A'}$, and as a result we break the binding property of COM. If $r_f^{A'} \neq r_f^A$ then restart from the beginning.

□

Lemma 2 (Privacy against Malicious Bob)**.** *The* 1*-out-of-*2*-*OQFE *Protocol* π_{MAL} *is private against malicious Bob.*

Proof. The proof follows directly from Theorem 4 and the quantum secure zero-knowledge property of Π.

□

Finally, as a direct application of 1-out-of-2-OQFE, we show in Appendix F a construction for Oblivious Transfer between a classical Alice and a quantum Bob over a classical channel where the security is in the semi-honest model, statistical against Alice and post-quantum against Bob.

7 Oblivious Quantum Function Evaluation

In this section we describe the oblivious quantum function evaluation (OQFE) protocol achieving one-sided simulation security. It is important to emphasize that the OQFE functionality is equivalent to a quantum 2-party computation (Q2PC) protocol between a fully classical party Alice and a quantum party Bob, where only Alice receives the quantum computation output. The equivalence follows directly by considering the Q2PC where the target computation is a universal function. Specifically, the Q2PC computation is chosen as a quantum function g which takes 2 inputs: a classical input x (Alice's input) and a quantum input $|\psi\rangle$ (Bob's input) and the result of the computation $g(x, |\psi\rangle)$ is obtained only by Alice (as Alice is classical, the output of the Q2PC will also be classical).

As the two functionalities are equivalent, from a security point of view our Quantum 2PC protocol achieves the same security as the OQFE protocol, namely: privacy against Bob (he learns nothing about Alice's input x) and simulation security against Alice (she learns nothing more than $g(x, |\psi\rangle)$). Due to this relation, in this section, we will only focus on the OQFE setting.

The OQFE protocol, denoted as π_{OQFE}, represents a generalization of the 1-out-of-2 OQFE construction and similar to the Protocol 6.1, we require the following primitives:

1. A commitment scheme $\mathsf{COM} = (Com, Dec)$ that is hiding against quantum adversaries and computationally biding.

2. A trapdoor one-way function $\mathcal{F} = (Gen_{\mathcal{F}}, Eval_{\mathcal{F}}, Inv_{\mathcal{F}})$ for the construction of OQFE.

3. An argument of knowledge post-quantum zero-knowledge protocol $\Pi^\star := (G^\star_{\mathsf{ZK}}, P^\star_{\mathsf{ZK}}, V^\star_{\mathsf{ZK}})$ for the NP-relation: $\mathsf{Rel} = \{com, (dec, m) : \mathsf{Dec}(com, dec, m) = 1\}$ described earlier.

4. An argument of knowledge post-quantum zero-knowledge protocol $\Pi := (G_{\mathsf{ZK}}, P_{\mathsf{ZK}}, V_{\mathsf{ZK}})$ for the NP-relation: Rel_f described earlier.

5. An argument system post-quantum zero-knowledge protocol $\Pi := (G'_{\mathsf{ZK}}, P'_{\mathsf{ZK}}, V'_{\mathsf{ZK}})$ for the NP-relation: $Rel' = \{(\delta, \pi, s^Z, s^X, com), (r, \theta, \phi', dec) : \delta = \phi' + \theta + r\pi \text{ and } \phi' = (-1)^{s^X}\phi + s^Z\pi \text{ and } \mathsf{Dec}(com, dec, \phi) = 1\}$.

Additionally, we require the universal blind quantum computation (UBQC) protocol with classical output (Protocol 2 of [23]) as a sub-module. The UBQC protocol is interactive and is based on the measurement-based model of quantum computation. In this model, one can represent an arbitrary quantum function f equivalently as a tuple $(\mathcal{G}_{n\times m}, \Phi, g)$ where $\mathcal{G}_{n\times m}$ is a highly entangled quantum state (often represented as a graph with dimension (n, m) and is also known as graph state), a sequence of classical angles: $\Phi := \{\phi_{i,j}\}$ for $i \in [n]$ and $j \in [m]$, where $\phi_{i,j} \in \{0, \frac{\pi}{4}, \cdots, \frac{7\pi}{4}\}$, and g denotes a set of bits dictating the dependency sets ($s^X_{i,j}$ and $s^Z_{i,j}$), known to both parties, which are required to perform certain Pauli corrections to obtain the desired deterministic computation. We assume that the bits corresponding to g are known to both Alice and Bob and for our purposes we can as well ignore it. Similarly, we can fix the graph state $\mathcal{G}_{n\times m}$, except its dimension (n, m), to say brickwork state [23] or cluster state [49] as both of them are known to be universal for quantum computation with (X, Y)-plane measurements [49, 50]. Similarly, we parametrize Bob's input quantum state $|\psi\rangle$ as: $\Phi_{in} = \{\phi_{0,j}\}_{j\in[m]}$ (modulo graph state and flow bits). For more details on measurement-based quantum computation we refer the readers to this excellent tutorial [51].

Protocol 7.1 OQFE Protocol, π_{OQFE}, with Classical Alice and Quantum Bob

Common input: $\sigma \leftarrow G_{\mathsf{ZK}}(1^\lambda), \sigma^\star \leftarrow G^\star_{\mathsf{ZK}}(1^\lambda), \sigma' \leftarrow G'_{\mathsf{ZK}}(1^\lambda)$

Private Inputs:

1. Sender (Bob): $|\psi\rangle$ an m-qubit state represented as the set of angles $\Phi_{in} := \{\phi_{0,j}\}_j$.
2. Receiver (Alice): f an n-qubit unitary represented as the set of angles $\Phi := \{\phi_{i,j}\}_{i,j}$ of a one-way quantum computation over a brickwork state/cluster state [50], of the size $n \times m$, along with the dependencies X and Z obtained via flow construction [52].

1. **Preliminary phase**

   1.1 Alice samples uniformly at random $r^A_{f,i,j} \leftarrow \{0,1\}^\lambda$ and $r_{i,j} \leftarrow \{0,1\}$ for $i \in [n]$ and $j \in [m]$.

   1.2 For each $i \in [n]$ and $j \in [m]$, Alice computes $\mathsf{Com}(r^{(i,j)}_{fA}) \rightarrow (com^{(i,j)}_f, dec^{(i,j)}_f)$ and $\mathsf{Com}(\phi_{i,j}) \rightarrow (com^{(i,j)}, dec^{(i,j)})$ and sends $(com^{(i,j)}_f, com^{(i,j)})$ to Bob.

   1.3 For each $i \in [n]$ and $j \in [m]$, Alice runs $P^\star_{\mathsf{ZK}}$ on input $\sigma^\star$ the statement $x := com^{(i,j)}$ and the witness $(dec^{(i,j)}, \phi_{i,j})$, and Bob runs $V^\star_{\mathsf{ZK}}$ on input $\sigma^\star$ and the statement x. If $V^\star_{\mathsf{ZK}}$ outputs 0 then Bob stops, otherwise he continues with the following steps.

   1.4 For each $i \in [n]$ and $j \in [m]$, Bob samples $r^B_{f,i,j}$ uniformly at random from $\{0,1\}^\lambda$. We denote by $r^B_f := \{r^B_{f,i,j}\}_{i,j}$. Bob sends r^B_f to Alice.

   1.5 Alice computes $r_{f,i,j} = r^A_{f,i,j} \oplus r^B_{f,i,j}$. She then runs $Gen_{\mathcal{F}}$ $n \cdot m$ times using internal random coins $r_{f,i,j}$ and obtains $(k^{(i,j)}, t^{(i,j)}_k, hp^{(i,j)})$ for $i \in [n]$ and $j \in [m]$. Denote by $k := \{k^{(i,j)}\}_{i,j}$ is the concatenation of the $n \cdot m$ public keys.

   1.6 For each $i \in [n]$ and $j \in [m]$:

      1.6.1 Alice runs P_{ZK} on input $\sigma, x^{(i,j)} = (k^{(i,j)}, r^{(i,j)}_{fB}, com^{(i,j)}_f), w = (r^{(i,j)}_{fA}, dec^{(i,j)}_f)$ and Bob runs V_{ZK} on input $\sigma, x^{(i,j)}$.

      1.6.2 If V_{ZK} outputs 0 then Bob aborts. Otherwise, he continues to the next step.

2. **QFactory and UBQC**

   2.1 For each $i \in [n]$ and $j \in [m]$, Alice on input $t^{(i,j)}_k$, and Bob on input $k^{(i,j)}$ run an instances of 8-states QFactory protocol[10] (in sequence) to obtain $\theta_{i,j}$ on client's side and $\left|+_{\theta_{i,j}}\right\rangle$ on server's side, where $\theta_{i,j} \leftarrow \mathbb{Z}_{\frac{\pi}{4}}$, $i \in [n]$, $j \in [m]$. For each qubit $\left|+_{\theta_{i,j}}\right\rangle$ Bob samples uniformly $r^B_{i,j} \xleftarrow{\$} \{0,1\}$ and applies controlled Pauli-X, $X^{r^B_{i,j}}$ to $\left|+_{\theta_{i,j}}\right\rangle$.

   2.2 Bob entangles all these qubits by applying controlled-Z gates between them in order to create a graph state $\mathcal{G}_{n\times m}$

   2.3 For $j \in [m]$ and $i \in [n]$:

[10] An 8-states QFactory protocol is combination of two runs of 4-states QFactory Protocol 5.1 given in [1]. The only difference between the QFactory Protocol 5.1, and the one used in our construction is that Alice does not execute the first step (a), since the key for the trapdoor OWFs has been already generated as described in the previous steps.

2.3.1 Alice computes $\delta_{i,j} = \phi'_{i,j} + \theta_{i,j} + r_{i,j}\pi$, where $\phi'_{i,j} = (-1)^{s^X_{i,j}}\phi_{i,j} + s^Z_{i,j}\pi$ and $s^X_{i,j}$ and $s^Z_{i,j}$ are computed using the previous measurement outcomes and the X and Z dependency sets. Alice then sends the measurement angle $\delta_{i,j}$ to Bob.

– Alice runs P'_{ZK} on input the CRS σ, the statement to be proven $x := (\delta_{i,j}, \pi_{i,j}, s^Z_{i,j}, s^X_{i,j}, com_{i,j})$ and the witness $w := (r_{i,j}, \theta_{i,j}, \phi'_{i,j}, dec_{i,j})$, and Bob runs the interactive algorithm V_{ZK}, on input the CRS σ and the statement x. Let b be the output of V'_{ZK}. If $b = 0$ then Bob aborts, otherwise he continues as follows.

2.3.2 Bob updates the measurement angles $\delta'_{i,j} := (-1)^{r^B_{i,j}}\delta_{i,j}$ (as in Protocol 6.1). Bob measures the qubit $|+_{\theta_{i,j}}\rangle$ in the basis $\{|+_{\delta'_{i,j}}\rangle, |-_{\delta'_{i,j}}\rangle\}$ and obtains a measurement outcome $s_{i,j} \in \{0,1\}$. Bob updates $s'_{i,j} = s_{i,j} \oplus r^B_{i,j}$. Bob sends the updated measurement result $s'_{i,j}$ to Alice.

2.3.3 Alice computes $\bar{s}_{i,j} = s'_{i,j} \oplus r_{i,j}$.

Output: Alice obtains the output $f(|\psi\rangle)$ as the concatenation of $\{\bar{s}_{i,m}\}_i$.

Theorem 6 (Correctness)**.** *In an honest run of the* OQFE *Protocol 7.1, when both parties follow the protocol specifications, Alice obtains the outcome $f(|\psi\rangle)$, where f is Alice's input and $|\psi\rangle$ is Bob's input.*

Proof. The correctness of OQFE Protocol 7.1 follows from the correctness proof of 1-out-of-2 OQFE (Theorem 2) and the correctness of the UBQC protocol [23]. □

Theorem 7. *Protocol π_{OQFE} securely computes $\mathcal{F}_{\mathsf{OQFE}}$ with one-sided simulation.*

To complete the proof we need to prove the following two lemmata: Lemma 3 and Lemma 4.

Lemma 3 (Privacy against Malicious Bob)**.** *The* OQFE *Protocol 7.1 is private against malicious Bob.*

Proof. This results from the privacy against Bob of the protocol resulting from combining the quantum-client UBQC protocol with QFactory proven in Theorem 5.3 of [18] together with the zero-knowledge property of Π and Π', and the hiding of the commitment scheme. □

Lemma 4 (Simulation-based Security Malicious Alice)**.** *The* OQFE *Protocol 7.1 is simulation-based secure against malicious Alice.*

Proof. We need to show that for any adversary $Alice^*$ there exists a PPT adversary $\mathcal{S}$ for the ideal model such that:

$$\{\mathsf{IDEAL}_{\mathcal{F}_{\mathsf{OQFE}},\mathcal{S}(z),Alice}(\Phi, \Phi_{in})\}_{\Phi,\Phi_{in},z} \approx_c \{\mathsf{REAL}_{\pi_{\mathsf{OQFE}},Alice^*(z),Alice}(\Phi, \Phi_{in})\}_{\Phi,\Phi_{in},z}$$

The proof will follow closely the steps of the proof of Theorem 5, hence we only provide a sketch of the proof highlighting the main differences.

We have to prove that there exists a PPT simulator S, that by having access only to the ideal functionality $\mathcal{F}_{\mathsf{OQFE}}$, can simulate the output of any malicious $Alice^*$ who runs one execution of π_{OQFE} with an honest sender Bob. The simulator S having oracle access to $Alice^*$, will run as a sender Bob in the real protocol. The simulator runs the argument of knowledge extractor for $\Pi^\star$ (which exists by definition), and extracts the input of the adversary (i.e., it extracts $\phi_{i,j}$ for all $i \in [n], j \in [m]$). The simulator now can invoke the ideal functionality $\mathcal{F}_{\mathsf{OQFE}}$ to obtain the output. Then the simulator extracts the trapdoors for all the trapdoor OWFs keys following the same procedure of the simulator of Theorem 5, to make sure that $Alice^\star$ was behaving honestly.

From this point on we are guaranteed that $Alice^\star$ behaves honestly by the soundness of Π'. Moreover, the soundness of Π' guarantees that the input that $Alice^\star$ is using to compute the values $\{\delta_{i,j}\}_{i\in[n],j\in[m]}$ is compatible with the input $\{\phi_{i,j}\}_{i\in[n],j\in[m]}$ extracted by the simulator. Hence, the simulator can act as the semi-honest simulator for the UBQC stage of the protocol. □

Acknowledgments MC acknowledges support from H2020 project PRIVILEDGE #780477. EK acknowledges support from the EPSRC Verification of Quantum Technology grant (EP/N003829/1), the EPSRC Hub in Quantum Computing and Simulation (EP/T001062/1), grant FA9550-17-1-0055 and the UK Quantum Technology Hub: NQIT grant (EP/M013243/1) and funding from the EU Flagship Quantum Internet Alliance (QIA) project. AM gratefully acknowledges funding from the AFOSR MURI project "Scalable Certification of Quantum Computing Devices and Networks". This work was partly done while AM was at the University of Edinburgh, UK where it was supported by EPSRC Verification of Quantum Technology grant (EP/N003829/1).

References

[1] Alexandru Cojocaru, Léo Colisson, Elham Kashefi, and Petros Wallden. Qfactory: Classically-instructed remote secret qubits preparation. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology – ASIACRYPT 2019*, pages 615–645, 2019.

[2] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In Alfred V. Aho, editor, *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*, pages 218–229. ACM, 1987.

[3] Andrew Chi-Chih Yao. How to generate and exchange secrets. In *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*, pages 162–167. IEEE, 1986.

[4] Andrew C Yao. Protocols for secure computations. In *Foundations of Computer Science, 1982. SFCS'08. 23rd Annual Symposium on*, pages 160–164. IEEE, 1982.

[5] Oded Goldreich. *Foundations of cryptography: volume 2, basic applications.* Cambridge university press, 2009.

[6] Amit Agarwal, James Bartusek, Vipul Goyal, Dakshita Khurana, and Giulio Malavolta. Post-quantum multi-party computation in constant rounds. *preprint arXiv:2005.12904*, 2020.

[7] Michael Ben-Or, Claude Crépeau, Daniel Gottesman, Avinatan Hassidim, and Adam Smith. Secure multiparty quantum computation with (only) a strict honest majority. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 249–260. IEEE, 2006.

[8] Claude Crépeau, Daniel Gottesman, and Adam Smith. Secure multi-party quantum computation. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 643–652. ACM, 2002.

[9] Frédéric Dupuis, Jesper Buus Nielsen, and Louis Salvail. Secure two-party quantum evaluation of unitaries against specious adversaries. In *Annual Cryptology Conference*, pages 685–706. Springer, 2010.

[10] Frédéric Dupuis, Jesper Buus Nielsen, and Louis Salvail. Actively secure two-party evaluation of any quantum operation. In *Annual Cryptology Conference*, pages 794–811, 2012.

[11] Elham Kashefi and Petros Wallden. Garbled quantum computation. *Cryptography*, 1(1):6, 2017.

[12] Elham Kashefi, Luka Music, and Petros Wallden. The quantum cut-and-choose technique and quantum two-party computation. *arXiv preprint arXiv:1703.03754*, 2017.

[13] Zhiyuan Sun, Qin Li, Fang Yu, and Wai Hong Chan. Application of blind quantum computation to two-party quantum computation. *International Journal of Theoretical Physics*, 57(6):1864–1871, 2018.

[14] Yfke Dulek, Alex B Grilo, Stacey Jeffery, Christian Majenz, and Christian Schaffner. Secure multi-party quantum computation with a dishonest majority. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 729–758. Springer, 2020.

[15] Joe Kilian. Founding cryptography on oblivious transfer. In Janos Simon, editor, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 20–31. ACM, 1988.

[16] Alexandru Cojocaru, Léo Colisson, Elham Kashefi, and Petros Wallden. On the possibility of classical client blind quantum computing. *arXiv preprint arXiv:1802.08759*, 2018.

[17] Alexandru Gheorghiu and Thomas Vidick. Computationally-secure and composable remote state preparation. *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1024–1033, 2019.

[18] Christian Badertscher, Alexandru Cojocaru, Léo Colisson, Elham Kashefi, Dominik Leichtle, Atul Mantri, and Petros Wallden. Security limitations of classical-client delegated quantum computing. *arXiv preprint arXiv:2007.01668*, 2020.

[19] Yael Gertner, Sampath Kannan, Tal Malkin, Omer Reingold, and Mahesh Viswanathan. The relationship between public key encryption and oblivious transfer. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 325–335, 2000.

[20] Charles H Bennett. Quantum Crytography. In *Proc. IEEE Int. Conf. Computers, Systems, and Signal Processing, Bangalore, India, 1984*, pages 175–179, 1984.

[21] Joseph F Fitzsimons. Private quantum computation: an introduction to blind quantum computing and related protocols. *npj Quantum Information*, 3(1):23, 2017.

[22] Urmila Mahadev. Classical verification of quantum computations. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 259–267. IEEE, 2018.

[23] Anne Broadbent, Joseph Fitzsimons, and Elham Kashefi. Universal blind quantum computation. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 517–526. IEEE, 2009.

[24] Victoria Lipinska, Jérémy Ribeiro, and Stephanie Wehner. Secure multi-party quantum computation with few qubits. *arXiv preprint arXiv:2004.10486*, 2020.

[25] Dorit Aharonov, Michael Ben-Or, and Elad Eban. Interactive proofs for quantum computations. *arXiv preprint arXiv:0810.5375*, 2008.

[26] Zvika Brakerski and Henry Yuen. Quantum garbled circuits. *arXiv preprint arXiv:2006.01085*, 2020.

[27] Vedran Dunjko, Joseph F Fitzsimons, Christopher Portmann, and Renato Renner. Composable security of delegated quantum computation. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 406–425, 2014.

[28] Elham Kashefi and Anna Pappa. Multiparty delegated quantum computing. *Cryptography*, 1(2):12, 2017.

[29] Monireh Houshmand, Mahboobeh Houshmand, Si-Hui Tan, and Joseph Fitzsimons. Composable secure multi-client delegated quantum computation. *preprint arXiv:1811.11929*, 2018.

[30] Atul Mantri. Secure Delegated Quantum Computing, PhD thesis, 2019.

[31] Dominique Unruh. Universally composable quantum multi-party computation. In *Advances in Cryptology–EUROCRYPT 2010*, pages 486–505. Springer, 2010.

[32] Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. Founding cryptography on oblivious transfer–efficiently. In *Annual international cryptology conference*, pages 572–591. Springer, 2008.

[33] Roger Colbeck. Impossibility of secure two-party classical computation. *Physical Review A*, 76(6):062308, 2007.

[34] Louis Salvail, Christian Schaffner, and Miroslava Sotáková. On the power of two-party quantum cryptography. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 70–87. Springer, 2009.

[35] Harry Buhrman, Matthias Christandl, and Christian Schaffner. Complete insecurity of quantum protocols for classical two-party computation. *Physical review letters*, 109(16):160501, 2012.

[36] Thomas Vidick and Tina Zhang. Classical proofs of quantum knowledge. *arXiv preprint arXiv:2005.01691*, 2020.

[37] Michael A Nielsen and Isaac Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, 2000.

[38] Dominique Unruh. Quantum proofs of knowledge. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 135–152, 2012.

[39] Anne Broadbent and Alex B Grilo. Zero-knowledge for qma from locally simulatable proofs. *arXiv preprint arXiv:1911.07782*, 2019.

[40] Andrea Coladangelo, Thomas Vidick, and Tina Zhang. Non-interactive zero-knowledge arguments for qma, with preprocessing. In *Annual International Cryptology Conference*, pages 799–828. Springer, 2020.

[41] Rafail Ostrovsky, Silas Richelson, and Alessandra Scafuro. Round-optimal black-box two-party computation. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, volume 9216 of *Lecture Notes in Computer Science*, pages 339–358. Springer, 2015.

[42] Ivan Damgård, Serge Fehr, and Louis Salvail. Zero-knowledge proofs and string commitments withstanding quantum attacks. In *CRYPTO*, volume 3152 of *Lecture Notes in Computer Science*, pages 254–272. Springer, 2004.

[43] Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. In Ernest F. Brickell, editor, *Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16-20, 1992, Proceedings*, volume 740 of *Lecture Notes in Computer Science*, pages 390–420. Springer, 1992.

[44] Carmit Hazay and Yehuda Lindell. *Efficient Secure Two-Party Protocols: Techniques and Constructions.* Springer-Verlag, Berlin, Heidelberg, 1st edition, 2010.

[45] Xinlan Zhou, Debbie W Leung, and Isaac L Chuang. Methodology for quantum logic gate construction. *Physical Review A*, 62(5):052316, 2000.

[46] Daniel Gottesman and Isaac L Chuang. Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations. *Nature*, 402(6760):390, 1999.

[47] Carsten Baum, Ivan Damgård, Vadim Lyubashevsky, Sabine Oechsner, and Chris Peikert. More efficient commitments from structured lattice assumptions. In *Security and Cryptography for Networks*, pages 368–385, 2018.

[48] Ivan Damgård. Efficient concurrent zero-knowledge in the auxiliary string model. In Bart Preneel, editor, *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques*, volume 1807 of *Lecture Notes in Computer Science*, pages 418–430. Springer, 2000.

[49] Robert Raussendorf, Daniel E Browne, and Hans J Briegel. Measurement-based quantum computation on cluster states. *Physical Review A*, 68(2):022312, 2003.

[50] Atul Mantri, Tommaso F Demarie, and Joseph F Fitzsimons. Universality of quantum computation with cluster states and (X, Y)-plane measurements. *Scientific Reports*, 7:42861, 2017.

[51] Dan E Browne and Hans J Briegel. One-way quantum computation-a tutorial introduction. *arXiv preprint quant-ph/0603226*, 2006.

[52] Vincent Danos and Elham Kashefi. Determinism in the one-way model. *Physical Review A*, 74(5):052310, 2006.

[53] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, pages 700–718. Springer Berlin Heidelberg, 2012.

[54] Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh Vazirani, and Thomas Vidick. Certifiable randomness from a single quantum device. *arXiv preprint arXiv:1804.00640*, 2018.

A Computational Complexity Classes

We provide some well-known relations and the class of languages associated with it. Classically, a relation over finite sets $\{0,1\}^* \times \{0,1\}^*$ is a subset $R \subseteq \{0,1\}^* \times \{0,1\}^*$, and the language associated with R is $L_R = \{x : \exists y : (x,y) \in R\}$.

Definition 13 (NP)**.** *The class* NP *consists of all languages $L \subseteq \{0,1\}^*$ for which there exists a uniformly generated family of classical, deterministic, poly-size circuits $\{V_x : x \in \{0,1\}^*\}$ and a polynomial m, such that the following holds:*

1. *(Completeness) For all $x \in L$ there exists an $m(|x|)$-bit witness w such that $V_x(w) = 1$*
2. *(Soundness) For all $x \notin L$ and for all $m(|x|)$-bit witness w, $V_x(w) = 0$.*

Definition 14 (MA)**.** *The class* MA *consists of all languages $L \subseteq \{0,1\}^*$ for which there exists a uniformly generated family of classical, randomized, poly-size circuits $\{V_x : x \in \{0,1\}^*\}$ and a polynomial m, such that the following holds:*

1. *(Completeness) For all $x \in L$ there exists an $m(|x|)$-bit witness w such that $\Pr(V_x(w) = 1) \geq 2/3$*
2. *(Soundness) For all $x \notin L$ and for all $m(|x|)$-bit witness w, $\Pr(V_x(w) = 0) \geq 1/3$.*

Definition 15 (MA-relation)**.** *A relation R is an MA-relation if there is a* PPT *Verifier V such that:*

1. *(Completeness) $(x,w) \in L_R \implies \Pr\big[V_{|x|}(x,w) = 1\big] \geq 2/3$*
2. *(Soundness) $x \notin L_R \implies \Pr\big[V_{|x|}(x,w) = 1\big] \leq 1/3$.*

where $V = \{V_n\}$ are the uniformly generated family of circuits.

In the quantum case we replace the "witness" w (the second argument) with a quantum state $|\psi\rangle$ and define the class QMA with polynomial-size quantum circuits $Q = \{Q_n\}_{n\in\mathbb{N}}$ such that for every n, Q_n takes as input a string $x \in \{0,1\}^n$ and a quantum state σ on $p(n)$ qubits (for some polynomial $p(n)$) and returns a single bit as output.

Definition 16 (QMA)**.** *The class* QMA *consists of all languages $L \subseteq \{0,1\}^*$ for which there exists a uniformly generated family of quantum poly-size circuits $\{Q_x : x \in \{0,1\}^*\}$ and a polynomial m, p where each V_x has $m(|x|)$ input qubits, $k(|x|)$ auxiliary qubits and its output is given by the first output qubit such that the following holds:*

1. *(Completeness) For all $x \in L$ there exists an $m(|x|)$-qubit witness $|\psi\rangle$ such that $\Pr(V_x \text{ accepts } |\psi\rangle) \geq 2/3$*
2. *(Soundness) For all $x \notin L$ and for all $m(|x|)$-qubit witness $|\psi\rangle$, $\Pr(V_x \text{accepts} |\psi\rangle) \leq 2/3$.*

Note that the completeness and soundness can be amplified to $1 - 2^{-poly(|x|)}$ and $2^{-poly(|x|)}$, respectively.

Interactive quantum machines [38, 40]. An *interactive quantum machine* is a machine M with two registers: a register S for its internal state, and a register N for sending and receiving messages (the network register). Upon activation, M expects in N a message, and in S the state at the end of the previous activation. At the end of the current activation, N contains the outgoing message of M, and S contains the new internal state of M. A machine M gets as input: a security parameter $\mu \in \mathbb{N}$, a classical input $x \in \{0,1\}^*$, and quantum input $|\Phi\rangle$, which is stored in S. Formally, machine M is specified by a family of circuits $\{M_{\mu x}\}_{\mu\in\mathbb{N}, x\in\{0,1\}^*}$, and a family of integers $\{r_{\mu x}\}_{\mu\in\mathbb{N}, x\in\{0,1\}^*}$. $M_{\mu x}$ is the quantum circuit that M performs on the registers S and N upon invocation. $r_{\mu x}$ determines the total number of messages/invocations. We might omit writing the security parameter when it is clear from the context. We say that M is *quantum-polynomial-time* (QPT) if the circuit $M_{\mu x}$ has polynomial size in $\mu + |x|$, the description of the circuit is computable in deterministic polynomial time in $\mu + |x|$ given μ and x, and $r_{\mu,x}$ is polynomially bounded in μ and x.

Usually, both these registers are assumed to be quantum but for this work, we require the register N to be strictly classical. In this work, we model one of the parties (Bob) as a quantum interactive machine while the other party (Alice) is only required to perform classical (stochastic) operations. We will denote the interactive machines for Alice and Bob as $\mathfrak{A}$ and $\mathfrak{B}$, respectively, with internal registers S and S' and the classical network register as N and N', respectively. Finally, we assume that Alice (completely classical party) sends the first message as well as receives the last message.

Oracle access to an interactive (quantum) machine. We say that a (quantum) algorithm A has oracle access to an interactive (quantum) machine M (and we write this as A^M, or sometimes $A^{|M\rangle}$ to emphasize that M is a quantum machine and that oracle access includes the ability to apply the inverse of M) to mean the following. Besides the security parameter and its own classical input x, we allow A to execute the quantum circuit $M_{\mu x}$ specifying M, and its inverse (recall that these act on the internal register S and on the network register N of M). Moreover, we allow A to provide and read messages from M (formally, we allow A to act freely on the network register N). We do not allow A to act on the internal register S of M, except via $M_{\mu x}$ or its inverse.

B Cryptographic Primitives

Definition 17 (k-regular)**.** *A deterministic function $f\colon \mathcal{D} \to \mathcal{R}$ is **k-regular** if $\forall y \in \operatorname{Im} f$, we have $|f^{-1}(y)| = k$.*

Definition 18 (Trapdoor One-Way Function)**.** *A family of functions $\{f_k : \mathcal{D} \to \mathcal{R}\}$ is a **trapdoor function** if:*

- *There exists a PPT algorithm* `Gen` *which on input 1^n outputs (k, t_k), where k represents the index of the function.*
- *$\{f_k : \mathcal{D} \to \mathcal{R}\}_{k\in\mathcal{K}}$ is a family of one-way functions, namely:*
  - *There exists a PPT algorithm that can compute $f_k(x)$ for any index k, outcome of the PPT parameter-generation algorithm Gen and any input $x \in \mathcal{D}$;*
  - *Any QPT algorithm $\mathcal{A}$ can invert f_k with at most negligible probability over the choice of k:*
    $$\Pr_{\substack{k\leftarrow Gen(1^n)\\ x\leftarrow\mathcal{D}\\ rc\leftarrow\{0,1\}^*}} [f(\mathcal{A}(k, f_k(x)) = f(x)] \le \mathsf{negl}(n)$$
    where rc represents the randomness used by $\mathcal{A}$
- *There exists a PPT algorithm* `Inv`*, which on input t_k (which is called the trapdoor information) output by* `Gen`*(1^n) and $y = f_k(x)$ can invert y (by returning all preimages of y[11]) with overwhelming probability over the choice of (k, t_k) and uniform choice of x.*

[11] While in the standard definition of trapdoor functions it suffices for the inversion algorithm `Inv` to return one of the preimages of any output of the function, in our case we require a two-regular trapdoor function where the inversion procedure returns both preimages for any function output.

Instantiation. A trapdoor one-way function can be instantiated from the construction of [53] and a 2-regular variant can be found in [1].

Definition 19 (Hardcore Predicate)**.** *A function $hc \colon \mathcal{D} \to \{0,1\}$ is a **hardcore predicate** for a function f if:*

- *There exists a PPT algorithm that, for any input x, can compute $hc(x)$;*
- *Any QPT algorithm $\mathcal{A}$ when given $f(x)$, can compute $hc(x)$ with negligible better than $1/2$ probability: $\Pr_{\substack{x \leftarrow \mathcal{D}(n) \\ rc \leftarrow \{0,1\}^*}} [\mathcal{A}(f(x), 1^n) = hc(x)] \leq \frac{1}{2} + \mathsf{negl}(n)$, where rc is the randomness used by $\mathcal{A}$;*

Definition 20 (Commitment Scheme)**.** *$CS = (Sen, Rec)$ is a 2-phase protocol between 2 polynomial-time interactive algorithms: sender Sen and receiver Rec. In the commitment phase Sen with input m interacts with Rec to produce a commitment com and the private output d of Sen.*

1. ***Correctness**: On the decommitment phase, Rec on input m and d accepts m as decommitment of com.*
2. ***Computational (post-quantum) Hiding**: For any QPT adversary Rec^* interacting with Sen, the probability distributions describing the output of Rec^*: $\{\langle Sen(0), Rec^* \rangle\}$ and $\{\langle Sen(1), Rec^* \rangle\}$ are computationally indistinguishable.*
3. ***Statistical Binding**: For any commitment com generated during the commitment phase by a malicious unbounded sender Sen^*, there exists negligible negl such that Sen^* with probability at most negl outputs 2 decommitments (m_0, d_0) and (m_1, d_1) with $m_0 \neq m_1$ such that Rec accepts both decommitments.*

*For **non-interactive commitment scheme** (Com, Dec) we use the notation:*

1. *Commitment phase: $Com(m) \to (com, dec)$, where com is the commitment of the message m and dec is the corresponding decommitment information.*
2. *Decommitment phase: $Dec(com, dec, m) = 1$*

Instantiation. A post-quantum hiding, statistical binding commitment scheme can be instantiated from the scheme proposed in [47].

C Security and Construction of required primitives for QFactory from [1]

In any run of the protocol, honest or malicious, the state that Alice believes that Bob has is the one described in Protocol 5.1. Therefore, the task that a malicious Bob wants to achieve, is to be able to guess, as good as it can, the description of the output state that Alice (based on the public communication) thinks Bob has produced. In particular, in our case, Bob needs to guess the bit θ_2 (corresponding to the basis) of the (honest) output state.

Definition 21 (4 states basis blindness)**.** *We say that a protocol (π_A, π_B) achieves **basis-blindness** with respect to an ideal list of 4 states*
$S = \{S_{\theta_1,\theta_2}\}_{(\theta_1,\theta_2) \in \{0,1\}^2}$ if:

- *S is the set of states that the protocol outputs, i.e.:*

$$\Pr\left[|\phi\rangle = S_{B_1 B_2} \in S \mid ((\theta_1, \theta_2), |\phi\rangle) \leftarrow (\pi_A \| \pi_B)\right] \geq 1 - \mathsf{negl}(n)$$

- *and no information is leaked about the index bit θ_2 of the output state of the protocol, i.e for all QPT adversary $\mathcal{A}$:*

$$\Pr\left[\theta_2 = \tilde{\theta_2} \mid ((\theta_1, \theta_2), \tilde{\theta_2}) \leftarrow (\pi_A \| \mathcal{A})\right] \leq 1/2 + \mathsf{negl}(n)$$

Theorem 8 (4-states QFactory is secure ([1])). *Protocol 5.1 satisfies 4-states basis blindness with respect to the ideal list of states:*
$S = \{\left|+\right\rangle, \left|-\right\rangle, \left|+_{\pi/2}\right\rangle, \left|+_{3\pi/2}\right\rangle\}$.

An 8-state QFactory protocol producing states in the set $\{k\pi/4\}_{k\in\{0,\dots,7\}}$ can be obtained from 2 runs of 4-state QFactory Protocol 5.1 given in [1].

C.1 Generic Construction

We will denote with g the injective, homomorphic, post-quantum OWF and with h the hardcore predicate of g, which is also homomorphic with respect to the operation of g. In more details, we require:
Consider a fixed element of the domain of g, x_0 and for now a public function h having the same domain as g
Then, we define the function $f(x,c) = g(x + c \cdot x_0)$, where $c \in \{0,1\}$. As g is injective, we can see that f is 2-regular (and one-way). Now, this function needs to be constructed and applied by Bob, but we don't want to reveal him the value of x_0, which is where the homomorphic property (for a one-time operation) steps in:

$$g(x) * g(x_0) = g(x + x_0) \text{ for any operations “*” and “+”}$$

Then, to compute f: $f(x,c) = g(x) * (c \cdot g(x_0))$, therefore it is sufficient to send him $g(x_0)$ (and the description of g) for Bob to apply f. And as g is one-way, then x_0 is also hidden from Bob.
Then, after applying a unitary corresponding to function f and a series of measurements, Bob obtains the quantum state: $H^{B_1} X^{B_2} \left|0\right\rangle \in \{\left|0\right\rangle, \left|1\right\rangle, \left|+\right\rangle, \left|-\right\rangle\}$, a single qubit gate whose description is represented by the 2 bits B_1 and B_2. We call B_1 the basis bit, and B_2 the output bit. The target is to ensure that B_1 is completely hidden from Bob.
The formal description of the 2 bits is the following:

$$\begin{aligned} B_1 &= h(x + x_0) \oplus h(x) \\ B_2 &= (B_1 \cdot \langle b, (x \oplus (x + x_0))\rangle) \oplus h(x)h(x + x_0) \end{aligned} \tag{16}$$

where x is a randomly chosen preimage of g and b is a random bit-string.
But now, as we were saying we wanted to ensure B_1 is completely hidden from Bob who only received from Alice $g(x_0)$.
The, if we impose that h is homomorphic in the sense:

$$h(x + x_0) \oplus h(x) = h(x_0)$$

, then we have:

$$B_1 = h(x_0)$$

And if additionally, we impose that h is a hardcore predicate with respect to function g, then Bob while he has $g(x_0)$ he knows nothing about $B_1 = h(x_0)$.
Moreover, as $B_1 = h(x_0)$, then Alice knows from the very beginning the value of the basis bit.

C.2 Function Description

The generation algorithm Gen_F will output:

1. k - the public description of a 2-regular trapdoor (post-quantum) function f_k;

2. t_k - the trapdoor information corresponding to f_k;

3. hp - a hardcore predicate associated with f_k

To construct Gen_F we rely on a family of injective homomorphic trapdoor functions $\mathcal{G} = \{g_{k'}\}_{k'}$ and a hardcore predicate h for $\mathcal{G}$.

```
$\mathtt{Gen}_{\mathcal{F}}(1^{\lambda})$
$(K', t_{K'}) \leftarrow_{\$} Gen_{\mathcal{G}}(1^{\lambda})$
$z_0 \leftarrow_{\$} Dom(g_{K'})$
$y_0 \leftarrow g_{K'}(z_0)$
$t_k \leftarrow (t_{K'}, z_0)$
$k \leftarrow (K', y_0)$
$f_k(z, c) := g_{K'}(z) + c \cdot y_0 = g_{K'}(z + c \cdot z_0)$ , where $c \in \{0, 1\}$
$hp \leftarrow h(z_0)$
return $(k, t_k, hp)$
```

To construct the injective homomorphic trapdoor functions $\mathcal{G} = \{g_{k'}\}_{k'}$ we rely on the construction of [53].

In other words, to sample a function f_k, we first sample a matrix $K' \in \mathbb{Z}_q^{m \times n}$ using the construction of [53] (that provides an injective and trapdoor function), a uniform vector $s_0 \in \mathbb{Z}_q^n$, an error $e_0 \in \mathbb{Z}_q^m$ according to a small Gaussian[12] and a random bit d_0, and we compute:

$$\begin{aligned} z_0 &= (s_0, e_0, d_0) \\ y_0 &= K's_0 + e_0 + d_0 \times \begin{pmatrix} \frac{q}{2} & 0 & \dots & 0 \end{pmatrix}^T \\ g_{K'}(s, e, d) &= K's + e + d \times \begin{pmatrix} \frac{q}{2} & 0 & \dots & 0 \end{pmatrix}^T \end{aligned} \tag{17}$$

The function f_k will then be defined as follow:

$$f_k(s, e, c, d) = K's + e + c \times y_0 + d \times \begin{pmatrix} \frac{q}{2} & 0 & \dots & 0 \end{pmatrix}^T \tag{18}$$

Note that c and d are bits, and the error e is chosen in a bigger space[13] than e_0 to ensure that the function f_{K,y_0} has two preimages with good probability. Moreover, if we define $h(s, e, c, d) = d$, it is easy to see that the hardcore property will directly come from the fact that under LWE assumption, no adversary can distinguish a LWE instance $K's_0 + e_0$ from a random vector, so it is not possible to know if we added or not a constant vector.

D From Secure Two-party Computation to Argument of Knowledge

Let $F(x_1, x_2)$ be a two-party functionality run between parties P_1 holding input x_1 and P_2 holding input x_2. In the ideal world, P_i with $(i \in \{1, 2\})$ sends its input x_i to the F and obtains only $y = F(x_1, x_2)$. We say that a protocol Π securely realizes F if the view of any malicious PPT $P_i^{\star}$ executing Π with an honest P_j with $i \neq j$ combined with the output of P_j (if any) can be simulated by a PPT simulator that has only access to F and has oracle access to $P_i^{\star}$.

This security requirement is formalized via the ideal/real-world paradigm. In the ideal world, the functionality is implemented by a trusted party that takes the inputs from P_1 and P_2 and provides the output to them and is therefore secure by definition. A real-world protocol Π securely realizes the ideal functionality F, if the following conditions hold.

The joint view of the output of any malicious sender $P_i^{\star}$ running one execution of Π with P_{i-1} and the output of P_{i-1} can be simulated by an expected PPT simulator Sim that has only access to the ideal world functionality F and oracle access to $P_i^{\star}$.

Definition 22 (Black-box 2-party computation [41])**.** *Let F be any* PPT *two inputs functionality. We say that a protocol Π securely computes F with if for every $b \in \{0, 1\}$, every non-uniform* PPT *adversary $P_i^{\star}$ controlling P_i in the real model, there exists a non-uniform expected* PPT *adversary Sim_i (having black-box access to P_i) for the ideal world such that:*

[12] but big enough to make sure the function is secure

[13] but small enough to make sure the partial functions $f(\cdot, \cdot, c, \cdot)$ are still injective

$$\{\mathsf{REAL}_{\Pi,P_i^\star(z)}(1^\lambda)\}_{z\in\{0,1\}^\lambda} \approx \{\mathsf{IDEAL}_{F,\mathsf{Sim}_i(z)}(1^\lambda)\}_{z\in\{0,1\}^\lambda}$$

where $\mathsf{REAL}_{\Pi,P_i^\star(z)}(1^\lambda)$ *denotes the distribution of the output of the adversary* $P_i^\star$ *(controlling the* P_i*) after a real execution of protocol* Π*.* $\mathsf{IDEAL}_{F,\mathsf{Sim}_i(z)}(1^\lambda)$ *denotes the analogous distribution in an ideal execution with a trusted party that computes* f *for the parties and hands the output to the them.*

D.1 2-party Computation implies Arguments of Knowledge

In this section, we consider the zero-knowledge functionality $\mathcal{F}_{\mathsf{ZK}}$. $\mathcal{F}_{\mathsf{ZK}}$ is parametrised by an NP-relation R and takes inputs only from one party called the prover. $\mathcal{F}_{\mathsf{ZK}}$, on input a statement x and a witness w, checks if $(x, w) \in R$. If that is the case, then $\mathcal{F}_{\mathsf{ZK}}$ outputs 1 to the second party (called verifier), else it outputs 0.

Theorem 9. *Any 2-party computation protocol* Π *that realizes the functionality* $\mathcal{F}_{\mathsf{ZK}}$ *accordingly to Def 22 is also an argument of knowledge with negligible knowledge error.*

Proof. To prove our theorem we need to show that Π satisfies the property of completeness and knowledge soundness. We property of completeness comes immediately from the Def. 22. Hence, we just need to exhibit an extractor $\mathsf{Extract}$. By assumption, we know that for every malicious party acting as the prover $\mathcal{P}^\star$ in Π, there exists an expected PPT simulator Sim_P such that

$$\{\mathsf{REAL}_{\Pi,\mathcal{P}^\star(z)}(1^\lambda)\}_{z\in\{0,1\}^\lambda} \approx \{\mathsf{IDEAL}_{\mathcal{F}_{\mathsf{ZK}},\mathsf{Sim}_P(z)}(1^\lambda)\}_{z\in\{0,1\}^\lambda}$$

where $\mathsf{REAL}_{\Pi,\mathcal{P}^\star(z)}(1^\lambda)$ denotes the distribution of the output of the adversarial prover $\mathcal{P}^\star$ after a real execution of protocol Π. $\mathsf{IDEAL}_{\mathcal{F}_{\mathsf{ZK}},\mathsf{Sim}_{\mathcal{P}}(z)}(1^\lambda)$ denotes the analogous distribution in an ideal execution with a trusted party that computes $\mathcal{F}_{\mathsf{ZK}}$ for the parties and hands the output to the them.

Our PoK extractor $\mathsf{Extract}$, then simply runs $\mathsf{Sim}_{\mathcal{P}}$, and when $\mathsf{Sim}_{\mathcal{P}}$ provides the input (x, w) to the ideal functionality $\mathcal{F}_{\mathsf{ZK}}$, $\mathsf{Extract}$ checks if $(x, w) \in R$, and if it is the case then it outputs w. We note that $\mathsf{Sim}_{\mathcal{P}}$ makes only black-box use of the adversary P, hence, so does $\mathsf{Extract}$ (that makes black-box use of $\mathsf{Sim}_{\mathcal{P}}$).

We now just need to argue that $\mathsf{Extract}$ runs in expected polynomial time and it extracts a valid witness with probability negligibly close to p, where p is the probability that the verifier, in the real world experiment, outputs 1. Without loss of generality, we assume that $|x|$ and λ are polynomially related.

By assumption, we know that Sim_P successfully extracts the input w such that $(x, w) \in R$ from the adversarial prover $\mathcal{P}^\star$ with probability (at least) negligible close to p. Indeed, if this does not hold, then neither the security of Π would hold.

Hence, the probability that Sim_P successfully extracts a witness is negligibly close to the probability p, for which the verifier outputs 1 in the real world. □

E 1-out-of-2 OQFE Correctness: Proof of Theorem 2

Theorem 2 (Correctness) *In an honest run of* 1*-out-of-*2 OQFE *Protocol 5.2, when both parties follow the protocol specifications, Alice obtains the outcome* $s_b = M_Z[R_x\left(-b \cdot \frac{\pi}{2}\right)|\psi_{in}\rangle]$*, where* b *is Alice's input and* $|\psi_{in}\rangle$ *is Bob's input.*

Proof. Using the correctness of the 4-states QFactory protocol (Protocol 5.1), Alice fixes $\theta_2 \in \{0, 1\}$ and Bob obtains at the end of the protocol the state $|out_{qf}\rangle = |+_\theta\rangle$ where θ can be described using the 2 bits $\theta_1\theta_2 \in \{0, 1\}^2$.

Now let us examine the computations performed in Steps 3-7 by Bob.

First Bob computes:

$$|\psi_0\rangle = X^{r_B}|+_\theta\rangle = \left|+_{(-1)^{r_B}\theta}\right\rangle \tag{19}$$

Then in Step 4, he performs:

$$(M_Z \otimes I_2)(H_1 \otimes I_2)(CZ\,|\psi_{in}\rangle \otimes |\psi_0\rangle) \tag{20}$$

If the measurement outcome is m_0 the unmeasured qubit $|\psi_1\rangle$ becomes:

$$|\psi_1\rangle = X^{m_0} R((-1)^{m_0 \oplus r_B}\theta) H |\psi_{in}\rangle \tag{21}$$

By using the notation $m'_0 = m_0 \oplus r_B$:

$$|\psi_1\rangle = X^{m_0} R((-1)^{m'_0}\theta) H |\psi_{in}\rangle \tag{22}$$

Bob uses now the measurement angle received from Alice:

$$\delta = \phi_b + \theta_2 \cdot \frac{\pi}{2} + r_A \cdot \pi = (b + \theta_2) \cdot \frac{\pi}{2} + r_A \cdot \pi \tag{23}$$

Then, Bob computes computes $\delta' = (-1)^{r_B} \cdot \delta$ and performs the following quantum measurement in Step 5:

$$(M_Z \otimes I_2)(H_1 R_1(-\delta') \otimes I_2)\ [CZ(|\psi_1\rangle \otimes |+\rangle)] \tag{24}$$

If the measurement outcome is m_1 the unmeasured qubit $|out'_c\rangle$ is equal to:

$$|out'_b\rangle = X^{m_1} H R(-\delta') |\psi_1\rangle \tag{25}$$

By replacing δ' and $|\psi_1\rangle$ we get:

$$\begin{aligned}
|out'_b\rangle &= X^{m_1} H R\left(-(-1)^{r_B}((b+\theta_2)\cdot\frac{\pi}{2} + r_A \cdot \pi)\right) X^{m_0} R\left((-1)^{m'_0}\theta\right) H |\psi_{in}\rangle \\
&= X^{m_1} Z^{m_0} H R\left[-(-1)^{r_B \oplus m_0}((b+\theta_2)\cdot\frac{\pi}{2} + r_A \cdot \pi)\right] \cdot \\
&\cdot R\left[(-1)^{m'_0}(\theta_1 \cdot \pi + \theta_2 \cdot \frac{\pi}{2})\right] H |\psi_{in}\rangle \\
&= X^{m_1} Z^{m_0} H R\left[(-1)^{m'_0}(-(b+\theta_2)\cdot\frac{\pi}{2} - r_A \cdot \pi + \theta_1 \cdot \pi + \theta_2 \cdot \frac{\pi}{2})\right] H |\psi_{in}\rangle \\
&= X^{m_1} Z^{m_0} H R\left[-(-1)^{m'_0} b \cdot \frac{\pi}{2} + (-1)^{m'_0}(\theta_1 - r_A)\cdot \pi\right] H |\psi_{in}\rangle \\
&= X^{m_1} Z^{m_0} H R\left[-(-1)^{m'_0} b \cdot \frac{\pi}{2}\right] R\left[(-1)^{m'_0}(\theta_1 - r_A)\cdot \pi\right] H |\psi_{in}\rangle
\end{aligned} \tag{26}$$

Using the relations: $R((-1)^{b_1} b_2 \pi) = R(b_2\pi) = Z^{b_2}$ and $R((-1)^{b_1} b_2 \frac{\pi}{2}) = Z^{b_1 b_2} R(b_2 \frac{\pi}{2})$, for any $b_1, b_2 \in \{0,1\}$ we get:

$$\begin{aligned}
|out'_b\rangle &= X^{m_1} Z^{m_0} H Z^{m'_0 \cdot b} R\left(-b \cdot \frac{\pi}{2}\right) R\left[(-1)^{m'_0}(\theta_1 - r_A)\cdot \pi\right] H |\psi_{in}\rangle \\
&= X^{m_1 \oplus (m'_0 \cdot b)} Z^{m_0} H R\left[-b \cdot \frac{\pi}{2}\right] Z^{\theta_1 \oplus r_A} H |\psi_{in}\rangle \\
&= X^{m_1 \oplus (m'_0 \cdot b) \oplus \theta_1 \oplus r_A} Z^{m_0} H R\left(-b \cdot \frac{\pi}{2}\right) H |\psi_{in}\rangle
\end{aligned} \tag{27}$$

Then, Bob applies the final corrections in Step 6:

$$\begin{aligned}
|out_b\rangle &= X^{m_1} Z^{m_0} |out'_b\rangle \\
|out_b\rangle &= X^{(m'_0 \cdot b) \oplus \theta_1 \oplus r_A} H R\left(-b \cdot \frac{\pi}{2}\right) H |\psi_{in}\rangle
\end{aligned} \tag{28}$$

In step 7, Bob measures this quantum state in the computational basis and sends the outcome $\bar{s_b}$ together with $m'_0 = m_0 \oplus r_B$ and m_{qf} to Alice.

$$\begin{aligned}
\bar{s_b} = M_Z |out_b\rangle &= M_Z X^{(m'_0 \cdot b) \oplus \theta_1 \oplus r_A} H R\left(-b \cdot \frac{\pi}{2}\right) H |\psi_{in}\rangle \\
\bar{s_b} &= [(m'_0\ \cdot\ b)\ \oplus\ \theta_1\ \oplus\ r_A] \oplus M_Z H R\left(-b \cdot \frac{\pi}{2}\right) H |\psi_{in}\rangle
\end{aligned} \tag{29}$$

On Alice side, in Step 8, she first uses her trapdoor key t_k and computes from m_{qf} the value of θ_1. Then, we can see that Alice by computing: $s_b = \bar{s_b} \oplus (m'_0\ \cdot\ b)\ \oplus\ \theta_1\ \oplus\ r_A$ she will obtain:

$$s_b = M_Z H R\left(-b \cdot \frac{\pi}{2}\right) H |\psi_{in}\rangle = M_Z R_x\left(-b \cdot \frac{\pi}{2}\right) |\psi_{in}\rangle \tag{30}$$

□

F 1-out-of-2 OT from 1-out-of-2 OQFE

Oblivious transfer (OT) is an important two-party cryptographic primitive that is used in a wide range of other (complex) cryptographic protocols including secure function evaluation, key exchange, etc.

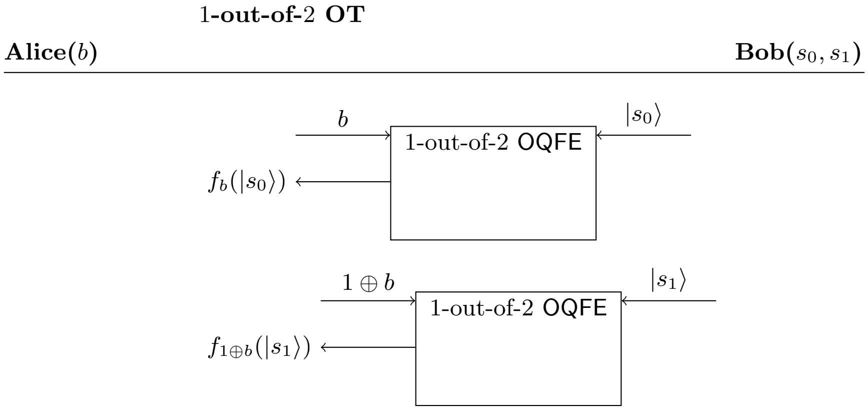
Definition 23. *(Ideal Functionality $\mathcal{F}_{\mathsf{OT}}$) A 1-out-of-2 oblivious transfer functionality, ot, is defined as follows. When both the parties ($\mathfrak{A}$ and $\mathfrak{B}$) are honest then $\mathcal{F}_{\mathsf{OT}}$ takes the input b and (s_0, s_1) from Alice and Bob, respectively and outputs s_b at Alice's side and nothing, $\perp$, on Bob's side.*

$$((s_0, s_1), b) \to (\perp, s_b) \tag{31}$$

We briefly sketch a simple constructions of 1-out-of-2 OT from 1-out-of-2 OQFE. The security can be shown similar to the 1-out-of-2 OQFE case, however, we won't present rigorous security analysis in this work. This is because our main idea here is to show that one can in principle obtain other important cryptographic primitives such as1-out-of-2 OQFE rather than providing a non-trivial construction of OT. Interestingly, our construction of quantum OT relies only on a classical channel and a single quantum party. To be precise, our construction relies on assumption from classical-client remote state preparation, in addition to the classical channel, which in turn can be reduced to variants of trapdoor claw-free family [54, 17] or injective homomorphic one-way functions [1]. We, therefore, leave further exploration of OQFE applications and their use cases for future work.

F.1 Our Construction

Our construction is modular in design and require only 2 runs of 1-out-of-2 OQFE, where Alice is classical and Bob is quantum, as shown in the figure below. Bob encodes his classical input (s_0, s_1) as quantum states, simply as $(|s_0\rangle, |s_1\rangle)$. Alice's input are b and $b \oplus 1$ corresponding to two runs of OQFE. If $b = 0$, Alice obtains $f_0(|s_0\rangle)$ and $f_1(|s_1\rangle)$; otherwise, she obtains $f_1(|s_0\rangle)$ and $f_0(|s_1\rangle)$ and we note that $f_0(|s_b\rangle) = s_b$.



```
if (b = 0) return
  $s_0 \leftarrow f_0(|s_0\rangle)$
else return
  $s_1 \leftarrow f_0(|s_1\rangle)$
```

Let's assume the case of semi-honest adversaries. Loosely speaking, Alice cannot obtain any information from $f_1(|s_b\rangle)$ and this can be seen by expanding the output obtained by Alice:

$$\begin{aligned} f_1(|s_b\rangle) &= M_Z R_X\left(-\frac{\pi}{2}\right)|s_b\rangle = M_Z\left[\frac{1-i}{2}|s_b\rangle + \frac{1+i}{2}|s_b \oplus 1\rangle\right] \\ Pr[f_1(|s_b\rangle) = 0] &= \frac{|1-i|^2}{4} + \frac{|1+i|^2}{4} = \frac{1}{2} = Pr[f_1(|s_b\rangle) = 1]. \end{aligned} \tag{32}$$

Moreover, it can also be shown that Bob cannot learn anything from the 2 runs of 1-out-of-2 OQFE with Alice's inputs b and $b \oplus 1$. To see this, note that the only messages received by Bob are, classical and depending on the values of b, the 2 angles - δ for the first run and δ' for the second run along with classical transcript corresponding to remote state preparation. The latter encodes θ_2 and θ_2', however, from the security of QFactory (Theorem 8) we have that θ_2 and θ_2' are independent and unknown to Bob. Additionally, r_A and $r_A{}'$ are uniformly and independently sampled by Alice. Recall that δ and δ' encrypts (with one-time pad) Alice's input b using θ_2, r_A' and θ_2', r_A', respectively, in the following way:

$$\begin{aligned}
\delta &= b + \theta_2 + 2r_A \bmod 4 \\
\delta' &= 1 + b + \theta_2' + 2r_A{}' \bmod 4
\end{aligned} \tag{33}$$

Therefore, δ and δ' does not reveal anything about b to Bob. The above informal argument about security only works in the semi-honest case, because for instance if Alice runs both runs of 1-out-of-2 OQFE with the same input b, then she obtains both secrets s_0 and s_1. Security against malicious Alice can be achieved using similar tools and techniques of commitment scheme and zero knowledge proof of knowledge techniques that is employed to uplift the security of OQFE (section 6).