ARTICLE OPEN

Check for updates

QEnclave - A practical solution for secure quantum cloud computing

Yao Ma[1,2]✉, Elham Kashefi[1,3], Myrto Arapinis[3], Kaushik Chakraborty[3] and Marc Kaplan[2]

We introduce a secure hardware device named a QEnclave that can secure the remote execution of quantum operations while only using classical controls. This device extends to quantum computing from the classical concept of a secure enclave that isolates a computation from its environment to provide privacy and tamper-resistance. Remarkably, our QEnclave only performs single qubit rotations but can nevertheless be used to secure an arbitrary quantum computation even if the qubit source is controlled by an adversary. More precisely, by attaching a QEnclave to a quantum computer, a remote client controlling the QEnclave can securely delegate its computation to the server solely using classical communication. We investigate the security of our QEnclave by modeling it as an ideal functionality named remote state rotation (RSR). We show that this resource, similar to the previously introduced functionality of remote state preparation, allows blind delegated quantum computing with perfect security. Our proof under the Abstract Cryptography framework shows the construction of remote state preparation from remote state rotation while preserving security. An immediate consequence is the weakening of the requirements for blind delegated computation. While previous delegated protocols relied on a client that can either generate or measure quantum states, we show that this same functionality can be achieved with a client that only transforms quantum states without generating or measuring them.

npj Quantum Information (2022)8:128 ; https://doi.org/10.1038/s41534-022-00612-5

INTRODUCTION

Quantum computing is an emerging field of computation technology that promises to produce faster algorithms for solving computational problems[1,2]. Many government agencies and large companies like Google, IBM, and Amazon are putting efforts into building a programmable quantum device that can outperform existing classical computers[3,4]. Some of them have already managed to develop small-scale quantum computers and provide cloud services allowing users to delegate their quantum computations[5–8].

Although this form of delegated quantum computation (DQC) services is very useful in practice, for education and research, for example, running algorithms on untrusted quantum hardware raises important privacy issues. A major challenge of DQC is to ensure the privacy of the client's computation who doesn't have any quantum computation capability. In this paper, we address this issue by introducing a quantum hardware assumption, namely quantum trusted execution environment (Quantum TEE) and showing how it can be used to implement privacy-preserving DQC, even with a fully-classical client.

In the classical world, a trusted execution environment (TEE) can be figured as a secure processor that executes code in an isolated environment and prevents malicious access from the rest of the device. Global Platform initially proposed standardization to ensure the protection of stored applications and data[9]. In practice, a TEE is designed to isolate the trusted execution of the software layer from the untrusted area, also called rich execution environment (REE). It is based on a combination of hardware architecture and cryptographic protection. It allows to control the flow of information between applications in multiple environments with different root-of-trust. In more advanced scenarios, TEEs have been used for blockchain[10], privacy-preserving machine learning[11,12], or cloud services[13,14].

The goal of delegated computation is to allow a computationally bounded client to assign some computation to a computationally powerful but untrusted server while maintaining the privacy of data. This is relevant, especially in the case of high-performance computing in the cloud. A similar question arises with universal quantum computers becoming available in the near future. Even though we have recently been witnessing spectacular developments, it is expected that scalable quantum computers will remain hard to build and expensive for a long time. It is very likely that they will only be accessed remotely, exactly like supercomputers are nowadays. In this context, DQC enables a client with limited quantum capabilities to delegate a computation to a quantum server while maintaining the correctness and privacy of the computation.

The first efficient universal protocol for secure (blind) delegated quantum computation was introduced in[15] see recent reviews for other similar protocols[16,17]. However, these protocols all assume a quantum channel between the client and the server, which for some quantum hardware platforms such as superconducting or cold atom qubits might prove to be impractical, at least in the near future. For this reason, the construction of an efficient, private, and secure DQC protocol using only classical communication will be extremely important. Given the impossibility of achieving information secure delegated computing using only classical communication[18] other assumptions must be considered. Recent breakthroughs based on post-quantum secure trapdoor one-way functions paved the way for developing entirely new approaches toward fully-classical client protocols for emerging quantum servers[19–21]. Nevertheless, the challenge for these protocols is the huge server overhead. This is due to the fact that one has to ensure the quantum circuit implementing the required masking protocol based on the learning with error (LWE) encryption[22] remains unhackable both classically and quantumly.

[1]Laboratoire d'Informatique de Paris 6 (LIP6), Sorbonne Université, Paris, France. [2]VeriQloud, Montrouge, France. [3]School of Informatics, University of Edinburgh, Edinburgh, UK. ✉email: yao.ma@lip6.fr

That leads to current proposals that require an order of 1000 server qubits for masking a single gate of the target client computation.

Our work explores a different approach based on the hardware security assumption to derive a practical secure DQC protocol with a fully-classical client setting. We explore the modular approach introduced in ref. [23] that defines the remote state preparation (RSP) as the main building block for DQC protocol. It is worth noting that in ref. [24] an RSP protocol was also proposed using a classical channel between client and server but assuming a resource called measurement buffer, which externalizes a quantum state measurement from the server-side. However, such a resource can not be realized classically, as was proven in ref. [25]. Indeed, it is known that it is impossible to construct a composable secure RSP protocol using only a classical channel between the client and the server without any hardware assumption, which confirms our approach to be the only way forward to construct an efficient DQC protocol with a classical client from the RSP module. One could also take a different approach to define a hardware security module that securely implements the measurement buffer (on the server-side) and then uses the protocol introduced in ref. [24]. However, there are two fall-backs for such protocol. First, it is desired that the hardware assumption be as simple as possible and as we discuss later, securing the measuring device leads to an unnecessarily complicated architecture. A more severe issue, however, is that, as mentioned before, due to the usage of LWE-based encryption, the protocol of ref. [24] requires a huge overhead on the server-side.

With these constraints in mind, we introduce our Quantum TEE, called QEnclave, as a practical way to make DQC secure with a classical client. Remarkably only one call to our simple hardware module is enough to create one remote blind qubit. Our QEnclave only transforms single qubit states without generating or measuring them. Nevertheless, it can be composed with the universal blind quantum computing protocol of ref. [15] to achieve secure DQC with perfect blindness (assuming minimal hardware assumption) whiles using only classical communication between the client and the server with optimal server overhead. Surprisingly, the blindness of the protocol holds even if the server controls the qubit source.

RESULTS

QEnclave as an ideal functionality: remote state rotation

The contributions of our work are twofold. The first one is the introduction of an ideal functionality named remote state rotation (RSR). The only operation performed by this functionality is to rotate a quantum state with arbitrary angles chosen uniformly at random from a fixed set. Compared to the other ideal functionalities, RSR is even weaker . While RSP generates quantum states by itself, RSR only allows rotations of single qubit states generated by the server.

Definition 1. (See Fig. 1) The ideal resource named remote state rotation for blindness (RSR_B) has two interfaces A and B. After receiving a single qubit state ρ_{in} from interface B, it performs a rotation $Z(\theta)$ with θ chosen uniformly at random from the set $\{0, \frac{\pi}{4}, \ldots \frac{7\pi}{4}\}$. It then outputs ($\rho_{out}$) at the server's interface and the angle θ at the client's interface.

Similar to RSP_B and $MRSP_B$ (See Section Remote state preparation for DQC for further details), this functionality removes any quantum capability for the client. In particular, using RSR_B removes the assumption of a quantum communication channel between the client and the server.

We further define a two-party protocol $\pi = (\pi_A, \pi_B)$ to prepare quantum states with RSR_B in which π_A only receives the angle θ from the interface A of RSR_B, and π_B takes as input a classical bit c and a quantum state from the server.

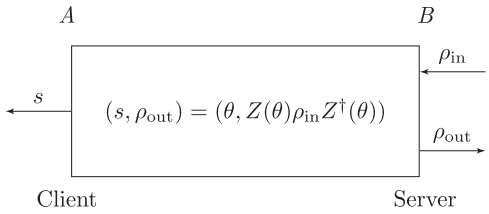

Fig. 1 Ideal Functionality of QEnclave: Remote State Rotation. Remote State Rotation (RSR) perform sarbitrary single qubit rotation with angel θ on income quantum state, outputs the angle at interface A and the post-rotation quantum state at interface B.

If $c = 0$, the server is honest, and π_B accepts $|+\rangle\langle+|$ as input from the server. If $c = 1$, the dishonest server prepares an arbitrary quantum state $\rho = \Omega(|+\rangle\langle+| \otimes \rho_{aux})\Omega^{\dagger}$. Here, Ω is an arbitrary unitary representing the server's deviation, and ρ_{aux} is an auxiliary state of the server.

After tracing out the auxiliary state of ρ, we get ρ_{in}, the input to RSR_B, which is a single qubit chosen by the dishonest server. In particular, this state can be entangled with the server's auxiliary system.

As a result, we show how to build RSP from RSR in the abstract cryptography (AC) framework[26]. In combination with previous results on the security of RSP, it implies that a classical client, using RSR, can achieve DQC with perfect blindness solely relying on classical communication even if the source that generates the state is compromised. In Section Composability of remote state rotation, it contains the detailed proof.

In practice, RSR reduces the client's quantum technology requirements compared to previously proposed RSP resources, usually requiring the client to generate or measure quantum states. This makes this functionality of independent interest for the study of practical quantum cryptographic protocols.

QEnclave: a practical design with secure processors

Our second contribution consists of a proposal to build our QEnclave using a standard classical TEE, with a protection of the flow between TEE and the quantum device that implements the single qubit rotations. That is to say, the QEnclave implements the ideal functionality RSR and uses the enclave and its hardware assumption to ensure the security of the construction. Moreover, it communicates with the client classically and returns a quantum state to the server, as shown in Fig. 2. For convenience, we assume that the client can choose the input angles uniformly at random, rather than letting the QEnclave choose them (as in RSR_B). This transformation does not change the security since, in our setup, both the client and QEnclave are expected to be honest.

By abstracting the functionality of secure processors as attested execution G_{att}, we demonstrate how G_{att} securely constructs outsourcing computation protocol under composition. By considering a simple two-party outsourcing computation $F_{outsrc}[C, S]$ with target function $y = f(x)$, where the client C outsources f and x with encoding and finally obtains the output y while the server S or any other adversary only knows the size of inputs and outputs ($|f + x|$, $|y|$) during the computation process. A G_{att}-hybrid protocol $\mathtt{Prot}_{outsrc}$ is given in ref. [27] and proven to UC-realized F_{outsrc} when C is honest and S is a static adversary. The probabilistic polynomial time indistinguishability of ideal-world and real-world executions is reduced to the decisional Diffie-Hellman (DDH) assumption for secure key exchange[28] and authenticated encryption. The indistinguishability is also equivalent under the AC framework without instantiating a DDH-based secure key exchange protocol but assuming the existence of a secure key exchange between

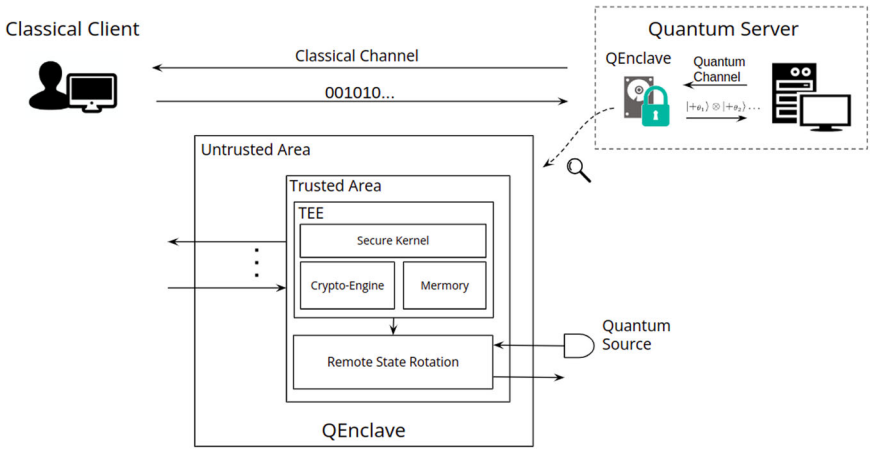

Fig. 2 Specification of QEnclave. The structure of QEnclave on the server's side is divided into two areas: the trusted and untrusted areas. The rotation angles from the client can not be revealed until the rotation is performed inside the trusted area. The quantum source is external to the QEnclave.

C and G_{att}. In practice, as long as the key exchange scheme is post-quantum secure, G_{att}-based RSR is feasible in terms of security.

Furthermore, a quantum-safe digital signature scheme Σ[29,30] is necessary for the remote attestation scheme since we assume that the quantum server is potentially malicious. Meanwhile, more practical remote attestation schemes provide post-quantum security[31].

The confidentiality consists in hiding the rotation angles chosen by the client. The requirement of using quantum-safe encryption makes symmetric schemes more appropriate than asymmetric ones for this task. Instead of a key exchange protocol based on DDH, there are other key encapsulation mechanism (KEM) schemes[32–34] available to share a secret key between the client and the QEnclave and proven to be secure against a quantum polynomial adversary for now.

Protocol 1. QEnclave-based RSP Protocol for Blindness with $prog_{rsr}$ G_{att}-enabled QEnclave Program $prog_{rsr}$:

On input ("keyex",p_k):
- let $(k, C_k) = \text{KEM.Enc}(p_k)$, seal k and return C_k

On input* ("compute",C_t):
- let $(f^\theta, x) := \text{AES.Dec}(k, C_t)$
- assert decryption success, C_t not seen before
- let $\theta_0 \ldots \theta_n := f^\theta(x)$, $\theta_0 \ldots \theta_n$ is applied to quantum states from an external source:

$$|e\rangle = Z_1(\theta_1) \otimes \ldots \otimes Z_n(\theta_n)|+\rangle^{\otimes n}$$

Server S:

On receive ("keyex",p_k) from C:
- let eid $:= G_{att}.\texttt{install}(\text{sid}, prog_{rsr})$
- let $(C_k, \sigma) := G_{att}.\texttt{resume}(\text{eid},(\text{"keyex"},p_k))$ and send (eid, C_k, σ) to C

On receive* ("compute", C_t) from C:
- let $|e\rangle = G_{att}.\texttt{resume}(\text{eid},(\text{"compute"},C_t))$ and keep $|e\rangle$ for further computation

Client C:

On initialize:
- let $(p_k, s_k) \leftarrow \text{KEM.KeyGen}(1^\lambda)$, $\texttt{mpk} := G_{att}.\texttt{getpk}()$
- send ("keyex", p_k) to S, await (eid, C_k, σ) from S
- assert $\Sigma.\text{Vf}_{\text{mpk}}((\text{sid}, \text{eid}, prog_{rsr}, C_k), \sigma)$
- let $k = \text{KEM.Dec}(s_k, C_k)$

On receive* ("compute",f^θ, x):
- let $C_t := \text{AES.Enc}(k, f^\theta, x)$ and send ("compute", C_t) to S

Once the secure channel is established between the client and QEnclave, the client can send the encrypted rotation angles to the QEnclave. QEnclave decrypts them and encodes the initial quantum state from the external source using the classical angles chosen by the client. At this stage, we assume that the trusted area, which includes the secure processor and minimal quantum devices. It leads to a remote state preparation protocol for delegated quantum computation with blindness using the QEnclave and classical communication between the client and the server. We summarize all the steps in Protocol 1 with a post-quantum secure digital signature scheme Σ with signature σ and a post-quantum secure key encapsulation mechanism KEM scheme with the key derivation function for generating a symmetric key for use, e.g., here we instantiate it as advanced encryption standard (AES).

Moreover, we assume that the communication between the secure processor and the quantum device is protected against a tampering server in terms of confidentiality. While this assumption may seem strong, the idea of sealing hardware components into a tamper-proof box is already well spread in the world of hardware security. In particular, the FIPS-140 certification for hardware security modules (HSM) includes criteria for physical tamper-evidence (level 2 certification), physical tamper-resistance (level 3), or even robustness against environmental attacks (level 4).

For the possible noise of quantum rotation inside QEnclave on the engineering level, the security definition of UBQC in terms of blindness[16] permits irrelevant errors to the secret information of the client. That is to say, the UBQC protocol remains secure while there are unrelated errors on top of the process. Since QEnclave is expected to retain the confidentiality of the client's secret information. Hence, the security of UBQC follows intuitively in theory. Still, this noise can not be verified throughout the protocol of RSR composing with UBQC. Also, since it is a multi-disciplined concept, especially more engineering-related. Plenty of open questions require inputs from different disciplines, e.g., controlling theory to solve. We leave this as one of our future work.

While our proposal of QEnclave implements the RSR ideal functionality, we have left aside a number of potential attacks that stem from the physical realization. Implicitly, we assume that QEnclave is fabricated correctly by a certified manufacturer, which ensures that an adversary cannot subvert the device before it is installed on the server. Besides, we exclude some hardware-dependent attacks in our work e.g., specific side-channel attacks on specific enclave products. Finally, we have not yet considered the possibility of counterfeiting the QEnclave.

The rest of the paper is organized as follows: We firstly conclude our paper with a discussion in Section Discussion. Here, we broadly discuss how our QEnclave can lead to a verifiable UBQC protocol[35] in Section QEnclave and verifiable quantum computation, the analysis of implementation with current quantum computing technologies in Section QEnclave and quantum computing technologies and other potential applications of our QEnclave in Section QEnclave and other applications; in Section Preliminaries, we recall the basic concepts and notations used in our work; in Section Remote state preparation for DQC, we introduce the functionalities for RSP used in our construction and discuss their composable security in the abstract cryptography framework; in Section Composability of remote state rotation, we give formal proofs of security to show how RSR_B builds a blind DQC protocol.

DISCUSSION

We introduced a functionality called remote state rotation that can be used to achieve secure delegated quantum computing in a practical way compatible with the currently available quantum hardware platform in the cloud. Moreover, we have proposed a realistic hardware assumption of trustworthy quantum operations with classical secrets to circumvent the impossibility results of[18,25] of implementing a composable RSR with a classical channel only. Our proposed ideal functionality with simple rotations lowers the minimal requirement on the client's operations while keeping minimal overhead on the server-side. Finally, we gave a complete specification of QEnclave that implements the RSR functionality using a secure processor to control the quantum devices required for the blindness of delegated quantum computation.

QEnclave and verifiable quantum computation

Besides privacy, another desirable property of delegated quantum computing is verifiability. In general, a DQC protocol is verifiable if the client can verify the result from the server (See Definition 4). A verifiable universal blind quantum computing protocol was proposed in ref. [35] where the client could insert in the target computation a set of trap qubits that are isolated from the computation. This construction ensures that the measurement results of trap qubits are always deterministic and can be used as a test of the correctness of the entire computation as they are known only by the client.

Adapting the same approach for RSR is not trivial as a malicious server controlling the source is now enabled to perform correlated attacks before and after the call to RSR. Hence the proof technique from ref. [35] does not directly apply. In principle, such deviations can be chosen to affect certain types of computation qubits but leave trap qubits unchanged, then change the execution of the protocol but remain unnoticed by the client at the same time, which means the protocol is not verifiable. However, there exist many other approaches to verifiability, such as the ones based on self-testing that might prove more suitable for RSR. We leave this question open for future work. It is worth mentioning that one could trivially add a trusted measurement device or a trusted source to the construction of the QEnclave to remove the possibility of such a correlated attack implementing directly the RSP resource instead. It will define directly an efficient classical client verifiable delegated computing protocol with an extended hardware assumption addressing the current challenge of demonstrating certifiable quantum supremacy. However, we believe keeping the QEnclave construction as simple as possible is a more interesting option to be explored.

QEnclave and quantum computing technologies

In this subsection, we discuss the integration of the QEnclave in different quantum computing technologies. Our current QEnclave only implements a single qubit rotation, and it interacts with the server's quantum computer while residing at the server's computing facility. Therefore, QEnclave always requires a quantum communication channel to interact with the source and the server's quantum computer. The linear optics-based photonics platforms are efficient for both quantum communication and single qubit rotation[36]. Hence we predict that the photonics-based platform would be ideal for implementing the QEnclave. Such QEnclaves would fit perfectly with photonics-based quantum computing technologies.

However, for the other kinds of quantum computing technologies, like ion traps-based processors, or superconducting-based qubits, we need to use an external interface for the interaction between the server and the QEnclave. Note that a promising approach to scaling ion-trap quantum computers to arbitrarily large numbers of qubits is to use many similar ion-trap processors (nodes) connected together in a modular network. Such a quantum network can produce ion-photon entanglement[37]. A naive idea for designing the interface between QEnclave and ion-trap-based quantum computer would be to use such an ion-photon entanglement to teleport the outcome of the QEnclave to the ion qubits. The detailed description of such an interface is beyond the scope of this paper. We leave this interesting study for our future work.

QEnclave and other applications

In this section, we discuss the other protocols that can exploit QEnclave rather than UBQC protocols. In general, the concept of QEnclave can be used for any client-server-based protocol[38] with quantum communication channels. First of all, QEnclave can be exploited in the prepare-and-send universal blind quantum computation with multiple clients to replace the multiple quantum communication channels from a server to the clients[39] for scalability. In this protocol, the security of blindness that the server doesn't learn the delegated computation and its input/output is guaranteed against either a dishonest server or a coalition of dishonest clients. In the case of the dishonest server, the protocol is equivalent to thinking of all honest clients as one with multiple input qubits. The blindness of DQC with QEnclave can be obtained intuitively.

In the case of the coalition of dishonest clients, the quantum channels among clients are replaced by RSP by QEnclave on the server-side. Meanwhile, since the clients are assumed to have secure access to verifiable secret sharing (VSS) in the protocol as a classical multiparty computation protocol, by committing classically every round of rotation angles during the RSP stage via VSS, the correctness of committed values can be verified by the honest server and the rest of honest clients. Note that the restriction to performing multiparty quantum computation for blindness with this protocol is that the collusion of a dishonest server and clients is impossible.

Secondly, in terms of quantum homomorphic encryption (QHE), which is formalized by ref. [40], it permits an evaluation of quantum circuits on encrypted quantum data in the DQC setting. Furthermore, a protocol of prepare-and-send QHE is proposed in[41] with quantum communication between client and server. Unlike UBQC, the quantum circuit is not hidden from the server, but the client can verify the computation by the decryption of the output. However, the composition of QEnclave and QHE is tricky since QHE requires the encryption and decryption of quantum data with confidentiality and integrity, as well as a trusted quantum source. Alternatively, one can put the encryption, evaluation, and decryption circuit fully inside the QEnclave. With such a powerful assumption, any classical client can run a secure QHE protocol using just classical communication. However, our primary goal here is to reduce the assumptions on the QEnclave functionality, i.e., we try to make the quantum circuit inside the

QEnclave as simple as possible. For example, the QEnclave contains only a single qubit rotation gate in our current setup. Making the QEnclave circuit simple for the QHE without losing security is challenging and beyond the scope of this paper. However, this is an interesting direction for our future research.

Finally, we think it can be relevant to use it in quantum money schemes[42], especially the protocol[43] considers that the bank mints the quantum states used as banknotes on the user's side and verifies their validity using only classical interactions. It matches our definition of remote state preparation once the problem of verifiability is also addressed. Then using a QEnclave, a bank might be able to authenticate the banknote by remotely performing quantum operations but using only classical communication.

METHODS

Preliminaries

Trusted execution environment. A TEE is a tamper-resistant processing environment that runs on a kernel[44] separated from its environment, named the rich execution environment (REE). It can be treated as a secure processor that guarantees the authenticity of the executed code, the integrity of the run-time states, and the confidentiality of its code and data. It can also provide remote attestations of its trustworthiness to third parties. A TEE should resist all software and physical attacks performed on the system's main memory. On the one hand, the OS and most of the applications are executed in the REE might be easily tampered with by the virus, trojans, malware, tools of rooting/reflashing, keystrokes logging, etc. On the other hand, running applications in the TEE is less efficient than on the REE.

There are many ways to implement a TEE in practice[45]. The smartcards we use daily are a prototype of TEE, with the smartcards themselves being the trusted area while the peripherals (e.g., POS terminal) do not need to be trusted[46]. Smartcards are completely isolated, providing high levels of trust, but are also very limited due to their size. The second type of familiar TEE is the trusted platform module (TPM)[47]. A TPM is a co-processor specialized for cryptographic tasks, including key generation, encryption, decryption, etc. The trusted components should include isolated engines for cryptography (e.g., SHA-1 engine, RSA engine, HMAC engine, etc.) and a random number generator. In addition, a TPM includes an isolated execution engine, platform configuration registers, and persistent memory for identification.

Apart from smartcards and TPMs, another type of TEE consists of designing processors with different execution environments and allowing inter-communication among environments with flow control (Fig. 3). Intel SGX, for example, allows users to instantiate a secure processor (enclave) to protect an application[48]. The code from outside the enclave cannot alter the application inside the enclave, even if executed with high privileges. Intel SGX also includes security measures such as remote attestation, crypto-based memory protection, sealing, etc. Another example is ARM TrustZone, which is implemented in most ARM processors nowadays. The system bus with ARM TrustZone uses an extended protected NS bit to distinguish the instructions of the trusted area from the untrusted area[49]. ARM TrustedZone can also protect specific peripherals by hiding them from untrusted applications.

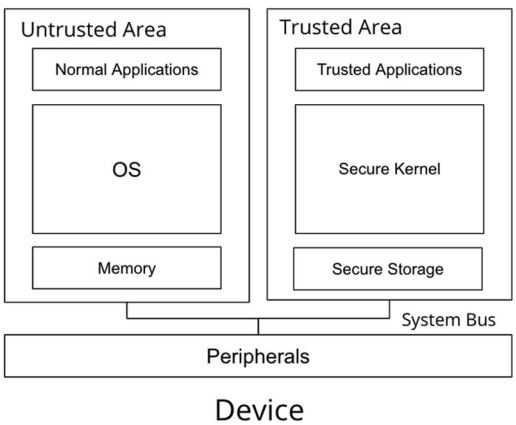

Fig. 3 TEE with co-existing execution environments. The trusted area on the right-hand side includes all trusted components used for executing trusted functions. Everything outside the trusted area are treated to be untrusted.

We introduce a feature that is important for our discussion: remote attestation. Remote attestation is a mechanism that allows proving the TEE integrity of a prover to a verifier. It provides an attestation signed by the TEE manufacturer. For instance, consider a client (verifier) aiming to delegate some application to the TEE on the server (prover) side. The client can challenge the server to provide him with an attestation signed specifically by the TEE manufacturer allowing the establishment of an authenticated channel between the client and the TEE before running a trusted application. The identity and hash of the TEE is a proof of integrity, signed with a hard-coded built-in private key[50]. The proof sent back by the server allows the client to verify the authenticity of the attestation message. Once the attestation is verified, the trusted application runs securely inside the TEE. It also allows anonymous attestation, where a user can verify an attestation generated by a valid enclave without identifying which one. The remote attested execution schemes are given in previous works[27,51] to capture the properties of enclave-like secure processors in the real world.

We exploit the abstraction of anonymous attested execution (See Functionality 1) as introduced in ref. [27] to formalize cryptographically the secure processors. G_{att} is parameterized by a signature scheme Σ and a register `reg` that captures all parties P that equips with a secure enclave. For the activation points of G_{att}, the starred ones are reentrant activation points, otherwise, it can be only executed once. In the registry stage, the secure processor enables the distribution of the manufacturer's public key of key pair (`mpk, msk`) to P upon the query. For enclave operations, the activation point `install` denotes an installation of enclave application with a program prog from P, it generates an identifier eid to P for identifying the enclave instance; the activation point `resume` enables the execution of prog upon the input inp by G_{att}. G_{att} then signs the output outp to be attested with *msk* using Σ. The attestation σ is returned to P for verification.

Functionality 1. Anonymous Attested Execution $G_{att}[\Sigma, \texttt{reg}]$

Registry:

// initialization

On initialize: (`mpk, msk`):= Σ. `KeyGen`(1^λ), $T = 0$

// public query interface

On receive* `getpk`() from some P : send `mpk` to P

Enclave Operations:

//install an enclave program

On receive* `install(idx, prog)` from some $P \in$ `reg`:

- if P is honest, assert idx = sid
- generate nonce eid $\in \{0, 1\}^\lambda$, store T[eid, P] := (eid, prog, 0), send eid to P

//resume an enclave program

On receive* `resume`(eid, `inp`) from some $P \in$ `reg`:

- let (idx, prog, mem) := T[eid, P], abort if not found
- let (`outp`, mem) := prog(`inp`, mem), update T[eid, P] := (idx, prog, mem)
- σ := Σ. $\texttt{Sig}_{\texttt{msk}}$(idx, eid, prog, `outp`), and send (`outp`, σ) to P

Quantum tools. We introduce the basic concepts required here. Interested readers can refer to standard textbooks on this topic[52]. In quantum computation, a quantum bit or (qubit) is a quantum system analogous to a classical bit. It lives in a two-dimensional Hilbert space H. In particular, the qubits of the computational basis of H are denoted as:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

More generally, the state of an arbitrary qubit is described as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ where $|\alpha|^2 + |\beta|^2 = 1$. An alternative basis called the Hadamard basis consists of the following qubits:

$$|+\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \qquad |-\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

We will in particular make use of the transform $Z(\theta)$ that maps $|\pm\rangle$ to $|\pm_\theta\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm e^{i\theta}|1\rangle)$. A quantum state can also be described by its density matrix $\rho = |\psi\rangle\langle\psi|$. Density matrices also capture mixed states of the form $\rho = \Sigma_s p_s |\psi_s\rangle\langle\psi_s|$ where p_s is a probability over pure states $|\psi_s\rangle\langle\psi_s|$.

For multiple qubit systems, two states $|v\rangle$ and $|w\rangle$ in two Hilbert spaces V and W with dimension n and m can be assembled as $|v\rangle \otimes |w\rangle$, or simply $|vw\rangle$, which lives in $V \otimes W$, a $n \cdot m$ dimensional Hilbert space. A quantum system $|u\rangle$ is called separable if it can be written $|v\rangle \otimes |w\rangle$. A multiple qubit system that is not separable is entangled. For example, $|\phi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ is an entangled state.

The measurement of a quantum state is defined by a set of operators $\{M_i\}$ satisfying $\sum_i M_i^\dagger M_i = I$ with its conjugate transpose operator $M^\dagger$. The probability of getting measurement result i on quantum state $|\psi\rangle$ is:

$$P(i) = \langle\psi|M_i^\dagger M_i|\psi\rangle = \langle\psi|M_i|\psi\rangle. \tag{1}$$

In particular, if $B = \{|u\rangle, |v\rangle\}$ is a basis of qubit states, then the measurement defined by the operators $\{|u\rangle\langle u|, |v\rangle\langle v|\}$ is usually referred to as a projection onto basis B.

The transformation of a quantum state can be described by a unitary operator U. These transforms preserve the norm of a vector and hence map a quantum state onto another quantum state.

We use the letters $X/Y/Z$ to denote some particular unitary operators called Pauli operators. For single qubit, the Pauli operators, as well as identity I, are given in the following matrices:

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, Y = \begin{pmatrix} 0 & \mathrm{i} \\ -\mathrm{i} & 0 \end{pmatrix}, Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2}$$

The other operators relevant here are the Hadamard (H) gate, which maps the computational basis to the Hadamard basis, and the control-U (CU) gates, which use two qubits as input: a control qubit and target qubit. It operates U on the target qubit when the control qubit is set to $|1\rangle$.

Furthermore, A completely positive and trace-preserving (CPTP) map $\mathcal{E}$ is a generalization of unitary operators to density matrices. It can always be described as a linear combination of Kraus operators $\{E_k\}$, which can be written as $E_k = \Sigma_i a_i \sigma_i$, where a_i is a complex number and σ_i is a Pauli operator.

Finally, we briefly introduce a model of quantum computation called measurement-based quantum computation (MBQC), originally proposed by Raussendorf and Briegel[53–55]. The DQC protocols discussed in our work are well described in the MBQC computation model. In this model, a computation is described by a series of commands involving single qubits or two qubits: preparations of single qubits in the state $|+\rangle$; entanglements of two qubits with the CZ operator; measurements on single qubits with basis $|+_\theta\rangle$ and $|-_\theta\rangle$ with measurement results 0 and 1 respectively, corrections on single qubits with operators X, Z depending on signals[56].

The entangled state used for computation in MBQC is called a graph state. An MBQC computation is a sequence of commands on a graph state that includes a subset of input and output qubits. In the family of graph state, cluster states are introduced in ref. [54] and brickwork states introduced in ref. [15] are proved to be universal for MBQC.

Abstract cryptography. The abstract cryptography (AC, also called Constructive Cryptography) framework was introduced in ref. [26] by Maurer and Renner for getting composable security properties. Compared to UC framework[57,58] that is built in a bottom-up approach, the AC framework is formalized with a top-down approach, where it considers the highest level of abstraction first, then the lower levels to instantiate particular objects of the protocol. UC can be realized by instantiating the abstraction of the AC framework. However, it is not our goal to compare different approaches in this paper but to show the idea behind composable security.

In the AC framework, the functionality is called a resource. A resource has a set of interface I corresponding to the parties that the resource interacts with. Since we focus on two-party communication between the client and the server as our protocol, our resources have two interfaces $I = \{A, B\}$ to the client and the server respectively.

A protocol $\pi = \{\pi_i\}_{i \in I}$ is a set of converters indexed by I. A converter has two interfaces—an inside interface and an outside interface, where the inside interface is connected to the resource and the outside interface is connected to the outside world. Intuitively, a dishonest party in a protocol has more access to the functionalities of a resource than an honest one. We denote by $\perp$ a filter used to enforce the honest behavior of a party. In this case, the functionalities accessed by the dishonest party are so-called filtered functionalities.

An important concept of the AC framework is the distinguisher (D), which measures the distance between two resources. For instance, consider a resource R and a protocol π_A, π_B, and denote $\pi_A R \pi_B$ as their composition. We say that two resources R, S are ε-closed, or $R \approx_\varepsilon S$ if there is no distinguisher D that can distinguish between R and S with an advantage greater that ε. If ε is negligible, we say that we can construct S from R with the protocol π_A, π_B. Furthermore, if the resource S is secure, we say that the resource R securely constructs S. The following definition formally defines this.

Definition 2. (See ref. [26]) Given two resources R and S, we say that a protocol $\pi = \{\pi_A, \pi_B\}$ constructs S from R within ε if the two following properties are satisfied:

- Correctness:

$$\pi_A R \pi_B \approx_\varepsilon S \perp, \tag{3}$$

- Security: if there exists a converter, where it is called a simulator σ such that

$$\pi_A R \approx_\varepsilon S\sigma. \tag{4}$$

We denote this:

$$R \xrightarrow{\pi, \varepsilon} S \tag{5}$$

Delegated quantum computing

In a client-server DQC protocol, a client with limited computational power asks a server to run a quantum computation, whose result is then returned to the client. There exist two types of DQC protocols. The first ones are prepare-and-send protocols, in which the client prepares a certain number of quantum states and sends them to the server. The second class is receive-and-measurement protocols[59], where the client receives single qubits from the server and measures them.

When delegating its computation, a client expects some security guarantees. The first one is blindness, which means that the server does not learn anything about the computation, input, and output. The second one is verifiability, which means a client can verify the correctness of the result returned by the server.

Ideal functionalities of DQC. The following definition from ref. [60] specifies an ideal resource for two-party delegated quantum computing with blindness.

Definition 3. (See Fig. 4a) The ideal resource for DQC S^{blind} provides both correctness and blindness. It takes an input ψ_A at the client's interface, and at the server's interface, a filtered control bit c (set by default to 0) and a pair that consists of a state ψ_B and a description of a CPTP map $\mathcal{E}$. It outputs the allowed leak ℓ^{ψ_A} at the server's interface. If $c = 0$, it outputs the correct result $U(\psi_A)$ at the client's interface; otherwise, it outputs the server's choice, $\mathcal{E}(\psi_{AB})$

The blindness means there is at most ℓ^{ψ_A} of information leaked to the server during the interactions. The other property of DQC that we are

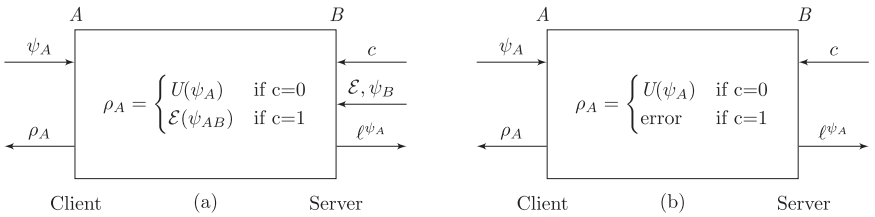

Fig. 4 DQC ideal resources with blindness and DQC ideal resources with both blindness and verifiability. The filtered control bit $c = 0/1$ denotes the honest/dishonest behavior of the server. The filtered functionalities with the input ψ_B, E and output ℓ^{ψ_A} of resource (**a**), and the output ℓ^{ψ_A} of resource (**b**) are accessible only to the dishonest server with $c = 1$.

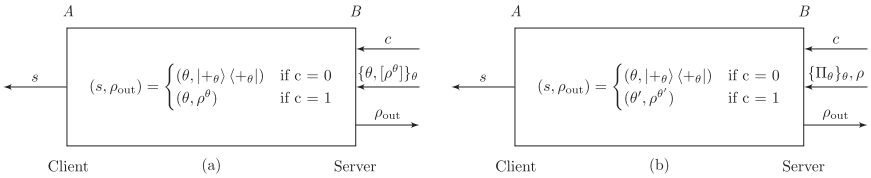

Fig. 5 RSP ideal resources for blindness and measured-based RSP for blindness. The filtered control bit $c = 0/1$ denotes the honest/dishonest behavior of the server. The filtered functionalities with the input $\{(\theta, [\rho^\theta])\}_\theta$ of resource (**a**), and $\{\Pi_\theta\}, \rho$ of resource (**b**) are accessible only to the dishonest server with $c = 1$.

interested in is verifiability. It means that if a dishonest server returns an incorrect result, the probability that the client accepts it is negligible. The following definition formalizes the definition of verifiable DQC.

Definition 4. (See Fig. 4b) The ideal resource DQC resource $S_{\text{verif}}^{\text{blind}}$ provides correctness, blindness and verifiability. It takes an input ψ_A at the client's interface and filtered control bits c (set by default to 0) at the server's interface. It outputs the allowed leak ℓ^{ψ_A} at the server's interface. If $c = 0$, it simply outputs $U(\psi_A)$ at the client's interface. If $c = 1$, it outputs an error message at the client's interface.

Universal blind quantum computation. Universal delegated quantum computation (UBQC), originally introduced in ref. [15] is a quantum computation model whose operations can easily be described in the MBQC model. At the start of a UBQC protocol, the client produces a sequence of single qubit states of the form $|+_\theta\rangle$ with θ chosen uniformly at random from $\{0, \frac{\pi}{4}, \ldots \frac{7\pi}{4}\}$. After receiving N such qubits from the client through a quantum channel, the server entangles them to build a brickwork state.

The computational stage is interactive and uses only classical communication. During this stage, the client continuously sends the measurement angle for each qubit to the server, which returns the measurement result to the client. The client then computes the following measurement angle. At the end of the computation, the server returns the quantum outputs to the client. Dunjko, Fitzsimons, Portmann, and Renner[60] showed the security of a UBQC protocol providing perfect blindness in the AC framework.

Remote state preparation for DQC

In this section, we review the works on ideal functionalities of RSP and their security in the AC framework. Using remote state preparation (RSP) as an ideal functionality allows replacing a quantum channel between a client and a server with a classical one.

Remote state preparation for blindness. The UBQC protocol introduced above requires the server to get a number of states of the form $|+_\theta\rangle\langle+_\theta|$. These states are then entangled as a brickwork state. Dunjko and Kashefi[23] have introduced the concept of weak correlations, which is a necessary and sufficient condition on the set of states sent by the client to obtain the blindness of the protocol. The following theorem formally introduces this notion.

Theorem 1. (See[23]) The UBQC protocol with classical input and computation of size N, where the client's preparation stage is replaced by the preparation of N states of the form σ_{AB}^i

$$\sigma_{AB}^i = \frac{1}{|\Theta|} \Sigma_{\theta_i \in \Theta} |\theta_i\rangle\langle\theta_i| \otimes \rho_i^{\theta_i}, \tag{6}$$

is blind if and only if the following conditions hold:

1. ρ^θ is a normalized quantum state, for all θ,
2. $\rho^\theta + \rho^{\theta+\pi} = \rho^{\theta'} + \rho^{\theta'+\pi}$ for all $\theta, \theta' \in \Theta (\Theta \in \{0, \frac{\pi}{4}, \ldots \frac{7\pi}{4}\})$,
3. $|\Theta|$ is the size of the set Θ.

The ideal resource random RSP for blindness is specified as follows[23]. If the server is honest, the functionality outputs $|+_\theta\rangle\langle+_\theta|$ to the server. If not, it takes as input from the server the classical description of a quantum state $[\rho^\theta]$ and outputs the corresponding quantum state ρ^θ to the server. In both cases, the client receives the classical angle θ. It is formalized in the following definition:

Definition 5. (See Fig. 5a) The ideal resource random remote state preparation for blindness, denoted RSP_B, has two interfaces, A to the client and B to the server. The resource chooses an angle of rotation θ uniformly at random from the set $\{0, \frac{\pi}{4}, \ldots \frac{7\pi}{4}\}$. There is a filtered functionality at interface B and a classical bit c. If $c = 0$, the server is honest and the resource outputs a state $|+_\theta\rangle\langle+_\theta|$ on B. If $c = 1$, the ideal functionality takes as input the set $\{(\theta, [\rho^\theta])\}_\theta$ from the server.

If the states provided by the server do not satisfy the conditions from Theorem 1, RSP_B ignores the input and waits for a new valid set. Once the set is received, the functionality outputs ρ^θ at B. In both case, RSP_B outputs the angle θ at the client's interface.

Dunjko and Kashefi also introduced another resource that is better suited for our purpose. It is a variant of RSP_B allows more operations by a dishonest server.

Definition 6. (See Fig. 5b) The ideal resource measurement-based remote blind state preparation (MRSP_B) has two interfaces A and B. The resource chooses an angle of rotation θ uniformly at random from the set $\{0, \frac{\pi}{4}, \ldots \frac{7\pi}{4}\}$. There is a filtered functionality at interface B and a classical bit c. If $c = 0$, the server is honest and the resource outputs a state $|+_\theta\rangle\langle+_\theta|$ on B. If $c = 1$, the ideal functionality takes as input the descriptions of eight positive operators $\{\Pi_\theta\}$, such that for all θ in $\{0, \frac{\pi}{4}, \ldots \frac{7\pi}{4}\}$, $\Pi_\theta + \Pi_{\theta+\pi} = I$. In addition, it accepts an arbitrary quantum state ρ of the same dimension as the operator Π_θ.

If the server's input does not satisfy the properties of Theorem 1, MRSP_B ignores it and waits for a new valid set. Once a valid input is received, MRSP_B applies the measurement $\Pi_\theta, \Pi_{\theta+\pi}$ corresponding to the chosen angle θ to ρ. Finally, MRSP_B outputs the measurement result θ', whose value is either θ or $\theta + \pi$, at the client's interface and the post-measurement state $\rho^{\theta'}$ at the server's interface.

The connection between these two ideal resources follows from the construction of MRSP_B from RSP_B, which preserves both correctness and security. Consider a trivial protocol $\pi = (\pi_A, \pi_B)$ in which π_A does nothing and π_B fixes the classical bit to $c = 0$. Following the conditions of Definition 2, it was shown that:

$$\pi_A \text{RSP}_B \pi_B = \text{MRSP}_B \perp \quad \textit{and} \quad \pi_A \text{RSP}_B = \text{MRSP}_B \sigma_B. \tag{7}$$

The inputs and outputs of these two ideal resources are trivially equivalent in the honest case, which implies correctness. To prove the security, the authors provided a simulator σ_B and showed that the outputs of A and B are the same actions for $\pi_A \text{RSP}_B$ and $\text{MRSP}_B \sigma_B$. Moreover, the authors show that RSP_B and MRSP_B can be used for UBQC. This leads to a perfect blind DQC without a quantum channel between the client and the server. In this context, perfect blindness means that the protocol leaks nothing more than what is strictly required (such as the size of the computation). The formal definition can be found in ref. [23].

The following theorem formalizes this argument for MRSP_B.

Theorem 2. (See ref. [23]) The UBQC protocol in which the client has access to the ideal functionality MRSP_B rather than to a quantum channel and a random generator of the $|+_\theta\rangle$ states, exactly constructs DQC with perfect blindness.

Limitations of RSP with only classical channel. While RSP_B and MRSP_B remove the need for a quantum communication channel between the client and the server, we have not discussed how these resources can be implemented. For example, a fully-classical blind DQC protocol could be obtained by implementing one of the two resources with a classical

communication channel. This idea is investigated by ref. [25]. They introduce the following definition.

Definition 7. An ideal resource S is said to be ε-classical-realizable if it is realizable from a classical channel C, i.e. if there exists a protocol $\pi = (\pi_A, \pi_B)$ between two parties interacting classically such that:

$$C \xrightarrow{\pi, \varepsilon} S \tag{8}$$

In order to prove the composable security of ε-classical-realizable RSP, we need to show that no unbounded adversary can learn information on θ by accessing only the right interface B. Unfortunately, the authors show that there is no describable remote state preparation protocol with composable security. In this context, describable means extracting a classical approximate description of a quantum state $[\rho]$ by accessing the state ρ on interface B. Since a protocol using only classical communication is describable, there is no classical-realizable RSP with composable security. It implies that UBQC with classical-realizable RSP cannot be composable and secure.

As a result, it is necessary to make additional assumptions to remove the quantum interaction between the client and the server. While[25] considers additional computational assumptions to bound the adversary's power, we take a different approach, introducing additional hardware assumptions such as tamper-proof quantum operations to get a secure DQC protocol with blindness using only classical communication.

Composability of remote state rotation

In this section, we elaborate on the composability of RSR in the AC framework. Compared to the other ideal functionalities of RSP, RSR is even weaker. While RSP generates quantum states by itself, RSR only allows rotations of single qubit states generated by the server.

We show the construction of DQC with RSR_B that achieves perfect blindness in two steps. First, In Lemma 1, we prove that the outcome of RSR_B satisfies the conditions for the blindness of Theorem 1. Then, in Theorems 3 and 4, we show the security of DQC with blindness obtained from RSR_B.

Lemma 1. For any quantum states ρ_{in} as used as input of RSR_B, the outcome system of the client and the server σ_{AB} satisfies the conditions of weak correlation of UBQC.

Proof. For simplicity, we first assume that ρ_{in} is not entangled with the server's auxiliary system. Without loss of generality, we get $\rho_{\mathrm{in}} = |\alpha|^2|0\rangle\langle 0| + \alpha\beta^*|0\rangle\langle 1| + \alpha^*\beta|1\rangle\langle 0| + |\beta|^2|1\rangle\langle 1|$. In this case, the output of RSR_B ρ^θ is

$$\rho^\theta = |\alpha|^2|0\rangle\langle 0| + e^{-i\theta}\alpha\beta^*|0\rangle\langle 1| + e^{i\theta}\alpha^*\beta|1\rangle\langle 0| + |\beta|^2|1\rangle\langle 1|. \tag{9}$$

For any θ in the set $\{0, \frac{\pi}{4}, \ldots \frac{7\pi}{4}\}$, we thus have

$$\rho^\theta + \rho^{\theta+\pi} = 2|\alpha|^2|0\rangle\langle 0| + 2|\beta|^2|1\rangle\langle 1|. \tag{10}$$

Since this is independent of θ, the state satisfies the weak correlation conditions.

In the general case, ρ_{in} can be entangled with the server's auxiliary system. we thus write $\rho'_{\mathrm{in}} = |\alpha|^2|0\rangle\langle 0| \otimes |\psi_0\rangle\langle\psi_0| + \alpha\beta^*|0\rangle\langle 1| \otimes |\psi_0\rangle\langle\psi_1| + \alpha^*\beta|1\rangle\langle 0| \otimes |\psi_1\rangle\langle\psi_0| + |\beta|^2|1\rangle\langle 1| \otimes |\psi_1\rangle\langle\psi_1|$, where $|\psi_0\rangle$ and $|\psi_1\rangle$ are states of the server's auxiliary system.

After the rotation of RSR_B on the first subsystem, we get the following entangled state:

$$\begin{aligned}\rho^\theta = {} & |\alpha|^2|0\rangle\langle 0| \otimes |\psi_0\rangle\langle\psi_0| + e^{-i\theta}\alpha\beta^*|0\rangle\langle 1| \otimes |\psi_0\rangle\langle\psi_1| \\ & + e^{i\theta}\alpha^*\beta|1\rangle\langle 0| \otimes |\psi_1\rangle\langle\psi_0| + |\beta|^2|1\rangle\langle 1| \otimes |\psi_1\rangle\langle\psi_1|.\end{aligned} \tag{11}$$

For any θ in the set $\{0, \frac{\pi}{4}, \ldots \frac{7\pi}{4}\}$, we have

$$\rho^\theta + \rho^{\theta+\pi} = 2|\alpha|^2|0\rangle\langle 0| \otimes |\psi_0\rangle\langle\psi_0| + 2|\beta|^2|1\rangle\langle 1| \otimes |\psi_1\rangle\langle\psi_1|. \tag{12}$$

Since the result $\rho^\theta + \rho^{\theta+\pi}$ is again independent of θ, the joint state of the client and the server also satisfy the weak correlation conditions for any state σ_{AB}.

We now prove the security of RSR_B with the UBQC protocol. We prove it by showing that the resource MRSP_B introduced in Definition 6 can be constructed from RSR_B. Since MRSP_B can be composed with a UBQC protocol to get DQC with perfect blindness, so does RSR_B

Theorem 3. The protocol $\pi = (\pi_A, \pi_B)$ introduced above with ideal resource RSR_B constructs the ideal resource MRSP_B.

Proof. We show that both the correctness and the security condition are satisfied. More precisely, proving the security amounts to showing that a distinguisher cannot distinguish MRSP_B from the protocol. This translates into the following equations for a simulator σ_B and the protocol $\pi = (\pi_A, \pi_B)$ with RSR_B.

$$\pi_A \mathrm{RSR}_B \pi_B \approx_\varepsilon \mathrm{MRSP}_B \perp, \tag{13}$$

and

$$\pi_A \mathrm{RSR}_B \approx_\varepsilon \mathrm{MRSP}_B \sigma_B. \tag{14}$$

For correctness, when the server is honest, the ideal resources RSR_B and MRSP_B both output an angle θ at interface A and its corresponding quantum state $|+_\theta\rangle\langle+_\theta|$ at interface B. Equation (13) is thus immediately satisfied.

For security, we introduce the simulator σ_B, defined as follows: It accepts and sends $c = 1$ to MRSP_B, as well as a set of projectors $\{\Pi^\theta\}$, where $\Pi^\theta = |+_{-\theta}\rangle\langle+_{-\theta}|$. After receiving a quantum system ρ from the server, the simulator takes the input ρ_{in} of the same dimension as $\{\Pi^\theta\}$ and generates a qubit $|0\rangle$. A CNOT gate is applied to these two qubits, where ρ_{in} is used as the control qubit ($|\phi_1\rangle$) and $|0\rangle$ the target bit ($|\phi_2\rangle$). This gives the simulator state ($\rho_{\sigma_B} = |\phi_{12}\rangle\langle\phi_{12}|$). Finally, σ_B sends the first qubit $|\phi_1\rangle$ back as the outcome state to the server, whereas the second qubit, $|\phi_2\rangle$, is sent to the resource MRSP_B.

We show that the outcome is similar to the expression obtained in Lemma 1. Again, we start by considering the case where ρ_{in} is not entangled with the server's auxiliary system. We then obtain the following expression for $|\phi'_{12}\rangle$ after the operation of MRSP_B:

$$\begin{aligned}|\phi'_{12}\rangle &= \frac{\Pi^\theta_2}{\sqrt{\langle\phi_{12}|\Pi^\theta_2|\phi_{12}\rangle}}|\phi_{12}\rangle \\ &= \tfrac{1}{\sqrt{2}}(|0\rangle + e^{-i\theta}|1\rangle)(\langle 0| + e^{i\theta}\langle 1|)(\alpha|00\rangle + \beta|11\rangle) \\ &= \tfrac{1}{\sqrt{2}}(\alpha|00\rangle + e^{-i\theta}\alpha|01\rangle + e^{i\theta}\beta|10\rangle + \beta|11\rangle)\end{aligned} \tag{15}$$

We obtain the outcome of the simulator by tracing out the second quantum subsystem.

$$\begin{aligned}\rho_1 &= Tr_2(|\phi'_{12}\rangle\langle\phi'_{12}|) \\ &= |\alpha|^2|0\rangle\langle 0| + e^{-i\theta}\alpha\beta^*|0\rangle\langle 1| + e^{i\theta}\alpha^*\beta|1\rangle\langle 0| + |\beta|^2|1\rangle\langle 1|\end{aligned} \tag{16}$$

The outcome quantum state is exactly the same as the result as the outcome of RSR_B in Eq. (9). Since a similar calculation holds for the projector $\Pi^{\theta+\pi}$, the outcome joint state of the client and the server of MRSP_B is the same as RSR_B.

Consider now an arbitrary entangled state $\alpha|0\rangle|\psi_0\rangle + \beta|1\rangle|\psi_1\rangle$. The simulator σ_B takes the first single qubit subsystem as the control qubit, and performs the same operation as in the previous case. After the operation of MRSP_B, we have:

$$\begin{aligned}|\phi'_{12}\rangle &= \frac{\Pi^\theta_2}{\sqrt{\langle\phi_{12}|\Pi^\theta_2|\phi_{12}\rangle}}|\phi_{12}\rangle \\ &= \tfrac{1}{\sqrt{2}}(|0\rangle + e^{-i\theta}|1\rangle)(\langle 0| + e^{i\theta}\langle 1|)(\alpha|0\rangle|\psi_0\rangle|0\rangle + \beta|1\rangle|\psi_1\rangle|1\rangle) \\ &= \tfrac{1}{\sqrt{2}}(\alpha|0\rangle|\psi_0\rangle|0\rangle + e^{-i\theta}\alpha|0\rangle|\psi_0\rangle|1\rangle + e^{i\theta}\beta|1\rangle|\psi_1\rangle|0\rangle + \beta|1\rangle|\psi_1\rangle|1\rangle)\end{aligned} \tag{17}$$

Then, after tracing out the second qubit, we obtain:

$$\begin{aligned}\rho_1 &= Tr_2(|\phi'_{12}\rangle\langle\phi'_{12}|) \\ &= |\alpha|^2|0\rangle\langle 0| \otimes |\psi_0\rangle\langle\psi_0| + e^{-i\theta}\alpha\beta^*|0\rangle\langle 1| \otimes |\psi_0\rangle\langle\psi_1| \\ &\quad + e^{i\theta}\alpha^*\beta|1\rangle\langle 0| \otimes |\psi_1\rangle\langle\psi_0| + |\beta|^2|1\rangle\langle 1| \otimes |\psi_1\rangle\langle\psi_1|.\end{aligned} \tag{18}$$

Again, The output quantum state is exactly equal to ρ^θ specified in Eq. (11). In consequence, the resource RSR_B is perfectly indistinguishable from the resource MRSP_B, that is, Equations (13) and (14) are satisfied with $\epsilon = 0$.

Finally, combining the fact that we can perfectly construct MRSP_B from RSR_B with Theorem 2, we obtain the following result.

Theorem 4. The UBQC protocol with the client accessing the RSR_B constructs the ideal functionality of DQC with perfect blindness.

Received: 22 September 2021; Accepted: 5 August 2022;
Published online: 05 November 2022

REFERENCES

1. Shor, P. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.* **26**, 1484–1509 (1997).
2. Aharonov, D., Jones, V. & Landau, Z. A polynomial quantum algorithm for approximating the jones polynomial. In *Proc. Thirty-Eighth Annual ACM Symposium on Theory of Computing* 427–436 (Association for Computing Machinery, 2006).
3. Arute, F. et al. Quantum supremacy using a programmable superconducting processor. *Nature* **574**, 505–510 (2019).
4. IBM. Quantum computing https://www.ibm.com/quantum-computing (2019).
5. Alsina, D. & Latorre, J. I. Experimental test of mermin inequalities on a five-qubit quantum computer. *Phys. Rev. A* **94**, 012314 (2016).
6. Devitt, S. J. Performing quantum computing experiments in the cloud. *Phys. Rev. A* **94**, 032329 (2016).
7. Hebenstreit, M., Alsina, D., Latorre, J. I. & Kraus, B. Compressed quantum computation using a remote five-qubit quantum computer. *Phys. Rev. A* **95**, 052339 (2017).
8. Wang, Y., Li, Y., Yin, Z.-q. & Zeng, B. 16-qubit IBM universal quantum computer can be fully entangled. *npj Quantum Inf.* **4**, 46 (2018).
9. GlobalPlatform. TEE system architecture v1.2 https://globalplatform.org/wp-content/uploads/2017/01/GPD_TEE_SystemArch_v1.2_PublicRelease.pdf (2018).
10. Lind, J. et al. Teechain: a secure payment network with asynchronous blockchain access. In *Proc. 27th ACM Symposium on Operating Systems Principles*, SOSP '19 63–79 (Association for Computing Machinery, 2019).
11. Grover, K., Tople, S., Shinde, S., Bhagwan, R. & Ramjee, R. Privado: practical and secure DNN inference with enclaves. Preprint at arxiv http://arxiv.org/abs/1810.00602 (2019).
12. Ohrimenko, O. et al. Oblivious multi-party machine learning on trusted processors. In *25th USENIX Security Symposium (USENIX Security 16)* 619–636 (USENIX Association, 2016).
13. Baumann, A., Peinado, M. & Hunt, G. Shielding applications from an untrusted cloud with Haven. *ACM Trans. Comput. Syst.* **33**, 1–26 (2015).
14. Schuster, F. et al. VC3: trustworthy data analytics in the cloud using SGX. In *2015 IEEE Symposium on Security and Privacy* 38–54 (IEEE, 2015).
15. Broadbent, A., Fitzsimons, J. & Kashefi, E. Universal blind quantum computation. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science* 517–526 (IEEE, 2009).
16. Fitzsimons, J. F. Private quantum computation: an introduction to blind quantum computing and related protocols. *npj Quantum Inf.* **3**, 23 (2017).
17. Gheorghiu, A., Kapourniotis, T. & Kashefi, E. Verification of quantum computation: an overview of existing approaches. *Theory of Comput. Syst.* **63**, 715–808 (2019).
18. Aaronson, S., Cojocaru, A., Gheorghiu, A. & Kashefi, E. Complexity-theoretic limitations on blind delegated quantum computation. In *46th ICALP 2019*, vol. 132 of *Leibniz International Proceedings in Informatics (LIPIcs)* 1–6 (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019).
19. Mahadev, U. Classical homomorphic encryption for quantum circuits. In *IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)* 332–338 (IEEE Computer Society, 2018).
20. Mahadev, U. Classical verification of quantum computations. In *IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)* 259–267 (IEEE Computer Society, 2018).
21. Cojocaru, A., Colisson, L., Kashefi, E. & Wallden, P. QFactory: Classically-Instructed Remote Secret Qubits Preparation. In: *Advances in Cryptology – ASIACRYPT 2019*. Lecture Notes in Computer Science, vol. 11921 (eds Galbraith, S. & Moriai, S.). https://doi.org/10.1007/978-3-030-34578-5_22 (Springer, Cham, 2019).
22. Regev, O. On lattices, learning with errors, random linear codes, and cryptography. In *Proc. Thirty-seventh Annual ACM Symposium on Theory of Computing* 84–93 (ACM, 2005).
23. Dunjko, V. & Kashefi, E. Blind quantum computing with two almost identical states. Preprint at arxiv https://arxiv.org/abs/1604.01586 (2016).
24. Gheorghiu, A. & Vidick, T. Computationally-secure and composable remote state preparation. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)* 1024–1033 (IEEE, 2019).
25. Badertscher, C. et al. Security Limitations of Classical-Client Delegated Quantum Computing. In: *Advances in Cryptology – ASIACRYPT 2020*. Lecture Notes in Computer Science, vol 12492 (eds Moriai, S. & Wang, H.). https://doi.org/10.1007/978-3-030-64834-3_23 (Springer, Cham, 2020).
26. Maurer, U. & Renner, R. Abstract cryptography. In *ICS* (2011).
27. Pass, R., Shi, E. & Tramèr, F. Formal Abstractions for Attested Execution Secure Processors. In: *Advances in Cryptology – EUROCRYPT 2017*. Lecture Notes in Computer Science, vol 10210 (eds Coron, JS. & Nielsen, J.). https://doi.org/10.1007/978-3-319-56620-7_10 (Springer, Cham, 2017).
28. Maurer, U., Tackmann, B. & Coretti, S. Key exchange with unilateral authentication: composable security definition and modular protocol design. *IACR Cryptology ePrint Archive* **2013**, 555 (2013).
29. Akleylek, S., Bindel, N., Buchmann, J., Krämer, J. & Marson, G.A. An Efficient Lattice-Based Signature Scheme with Provably Secure Instantiation. In: *Progress in Cryptology – AFRICACRYPT 2016*. Lecture Notes in Computer Science, vol 9646 (eds Pointcheval, D., Nitaj, A. & Rachidi, T.). https://doi.org/10.1007/978-3-319-31517-1_3 (Springer, Cham, 2016).
30. Buchmann, J., Dahmen, E. & Hülsing, A. XMSS - A Practical Forward Secure Signature Scheme Based on Minimal Security Assumptions. In: *Post-Quantum Cryptography*. Lecture Notes in Computer Science, vol 7071 (eds Yang, B. Y.). https://doi.org/10.1007/978-3-642-25405-5_8 (Springer, Berlin, Heidelberg, 2011).
31. Liu, X., Misoczki, R. & Sastry, M. R. Remote attestation for low-end prover devices with post-quantum capabilities. In *Proc. Eighth ACM Conference on Data and Application Security and Privacy* 84–94 (ACM, 2018).
32. Baldi, M., Barenghi, A., Chiaraluce, F., Pelosi, G. & Santini, P. LEDAkem: A Post-quantum Key Encapsulation Mechanism Based on QC-LDPC Codes. In: P*ost-Quantum Cryptography*. Lecture Notes in Computer Science, vol 10786 (eds Lange, T. & Steinwandt, R.). https://doi.org/10.1007/978-3-319-79063-3_1 (Springer, Cham, 2018).
33. Bindel, N., Brendel, J., Fischlin, M., Goncalves, B. & Stebila, D. Hybrid Key Encapsulation Mechanisms and Authenticated Key Exchange. In: *Post-Quantum Cryptography*. Lecture Notes in Computer Science, vol 11505 (eds Ding, J. & Steinwandt, R.). https://doi.org/10.1007/978-3-030-25510-7_12 (Springer, Cham, 2019).
34. Wang, W. & Stöttinger, M. Post-quantum secure architectures for automotive hardware secure modules. *IACR Cryptol. ePrint Arch.* **2020**, 26 (2020).
35. Fitzsimons, J. F. & Kashefi, E. Unconditionally verifiable blind quantum computation. *Phys. Rev. A* **96**, 012303 (2017).
36. Carolan, J. et al. Universal linear optics. *Science* **349**, 711–716 (2015).
37. Stute, A. et al. Tunable ion–photon entanglement in an optical cavity. *Nature* **485**, 482–485 (2012).
38. VeriQloud. Quantum protocol zoo. https://wiki.veriqloud.fr/index.php?title=Main_Page (2019).
39. Kashefi, E. & Pappa, A. Multiparty delegated quantum computing. *Cryptography* **1**, 12 (2017).
40. Broadbent, A. & Jeffery, S. Quantum Homomorphic Encryption for Circuits of Low T-gate Complexity. In: *Advances in Cryptology -- CRYPTO2015*. Lecture Notes in Computer Science, vol 9216 (eds Gennaro, R. & Robshaw, M). https://doi.org/10.1007/978-3-662-48000-7_30 (Springer, Berlin, Heidelberg, 2015).
41. Dulek, Y., Schaffner, C. & Speelman, F. Quantum homomorphic encryption for polynomial-size circuits. *Theory Comput.* **14**, 1–45 (2018).
42. Wiesner, S. Conjugate coding. *ACM SIGACT News* **15**, 78–88 (1983).
43. Radian, R. & Sattath, O. Semi-quantum money. In *Proc. 1st ACM Conference on Advances in Financial Technologies* 132–146 (Association for Computing Machinery, 2019).
44. Sabt, M., Achemlal, M. & Bouabdallah, A. Trusted execution environment: what it is, and what it is not. In *2015 IEEE Trustcom/BigDataSE/ISPA* 57–64 (IEEE, 2015).
45. González, J. *Operating System Support for Run-Time Security with a Trusted Execution Environment*. Ph.D. thesis (2015).
46. Kömmerling, O. & Kuhn, M. G. Design principles for tamper-resistant smartcard processors. In *Proc. USENIX Workshop on Smartcard Technology on USENIX Workshop on Smartcard Technology* 2 (USENIX Association, 1999).
47. TPM 2.0 library. https://trustedcomputinggroup.org/resource/tpm-library-specification/ (2019).
48. Intel® software guard extensions developer guide. https://download.01.org/intel-sgx/linux-1.7/docs/Intel_SGX_Developer_Guide.pdf (2016).
49. ARM security technology building a secure system using TrustZone technology. https://developer.arm.com/documentation/PRD29-GENC-009492/c (2022).
50. Sailer, R., Jaeger, T., Zhang, X. & van Doorn, L. Attestation-based policy enforcement for remote access. In *Proc. 11th ACM conference on Computer and communications security - CCS '04* 308 (ACM Press, 2004).
51. Barbosa, M., Portela, B., Scerri, G. & Warinschi, B. Foundations of hardware-based attested computation and application to SGX. In *2016 IEEE European Symposium on Security and Privacy* 245–260 (IEEE, 2016).
52. Nielsen, M. A. & Chuang, I. L. *Quantum Computation and Quantum Information* (Cambridge Univ. Press, 2010).
53. Raussendorf, R. & Briegel, H. J. A one-way quantum computer. *Phys. Rev. Lett.* **86**, 5188–5191 (2001).
54. Raussendorf, R., Browne, D. E. & Briegel, H. J. Measurement-based quantum computation on cluster states. *Phys. Rev. A.* **68**, 022312 (2003).
55. Briegel, H. J., Browne, D. E., Dür, W., Raussendorf, R. & Nest, M. V. D. Measurement-based quantum computation. *Nat. Phys.* **5**, 19–26 (2009).
56. Danos, V., Kashefi, E. & Panangaden, P. The measurement calculus. *J. ACM* **54**, 8 (2007).
57. Canetti, R. Universally composable security: a new paradigm for cryptographic protocols. In *Proc. 42nd IEEE Symposium on Foundations of Computer Science* 136–145 (IEEE, 2001).
58. Canetti, R., Dodis, Y., Pass, R. & Walfish, S. Universally Composable Security with Global Setup. In: Theory of Cryptography. Lecture Notes in Computer Science, vol

4392 (eds Vadhan, S. P.). https://doi.org/10.1007/978-3-540-70936-7_4 (Springer, Berlin, Heidelberg, 2007).
59. Hayashi, M. & Morimae, T. Verifiable measurement-only blind quantum computing with stabilizer testing. *Phys. Rev. Lett.* **115**, 220502 (2015).
60. Dunjko, V., Fitzsimons, J.F., Portmann, C. & Renner, R. Composable Security of Delegated Quantum Computation. In: *Advances in Cryptology –ASIACRYPT 2014*. Lecture Notes in Computer Science, vol 8874 (eds Sarkar, P. & Iwata, T.). https://doi.org/10.1007/978-3-662-45608-8_22 (Springer, Berlin, Heidelberg, 2014).

ACKNOWLEDGEMENTS

Y.M. is very grateful to Léo Colisson for many useful discussions about the security proofs. This work is supported by grants from Région Ile-de-France and Veriqloud, as well as Innovate UK-funded project called AirQKD: product of a UK industry pipeline, Grant Number 106178.

AUTHOR CONTRIBUTIONS

Y.M contributed to the modeling definitions and security proofs. K.C. contributed to the construction of protocols. E.K., M.A., and M.K supervised the work. All authors contributed to the manuscript writing.

COMPETING INTERESTS

The authors declare no competing interests

ADDITIONAL INFORMATION

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41534-022-00612-5.

Correspondence and requests for materials should be addressed to Yao Ma.

Reprints and permission information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.