Indistinguishability Obfuscation of Null Quantum Circuits and Applications

James Bartusek
University of California, Berkeley, CA, USA

Giulio Malavolta
Max Planck Institute for Security and Privacy, Bochum, Germany

Abstract

We study the notion of indistinguishability obfuscation for null quantum circuits (quantum null-iO). We present a construction assuming:

- The quantum hardness of learning with errors (LWE).
- Post-quantum indistinguishability obfuscation for classical circuits.
- A notion of “dual-mode” classical verification of quantum computation (CVQC).

We give evidence that our notion of dual-mode CVQC exists by proposing a scheme that is secure assuming LWE in the quantum random oracle model (QROM).

Then we show how quantum null-iO enables a series of new cryptographic primitives that, prior to our work, were unknown to exist even making heuristic assumptions. Among others, we obtain the first witness encryption scheme for QMA, the first publicly verifiable non-interactive zero-knowledge (NIZK) scheme for QMA, and the first attribute-based encryption (ABE) scheme for BQP.

2012 ACM Subject Classification Theory of computation → Computational complexity and cryptography

Keywords and phrases Obfuscation, Witness Encryption, Classical Verification of Quantum Computation

Digital Object Identifier 10.4230/LIPIcs.ITCS.2022.15


1 Introduction

The goal of program obfuscation [33, 10] is to convert an arbitrary circuit $C$ into an unintelligible but functionally equivalent circuit $\tilde{C}$. Recent work has shown that program obfuscation enables a series of new remarkable applications (e.g. [24, 42, 27, 12]), establishing obfuscation as a central object in cryptography.

Yet, the scope of obfuscation has so far been concerned almost exclusively with classical cryptography. The advent of quantum computing has motivated researchers [1, 5] to ask whether program obfuscation is a meaningful notion also in a quantum world:

Can we obfuscate quantum circuits? Is this notion useful at all?

Unfortunately, results on the matter are largely negative [5, 8, 3], barring a few schemes for restricted function classes of questionable usefulness [6, 20]. At present, it is unclear whether obfuscation of quantum circuits in its most general form can exist at all. The goal of this work is to make progress on this question. Our contributions are twofold.

1.1 Quantum Null-iO and Witness Encryption for QMA

We show that, assuming LWE, post-quantum indistinguishability obfuscation (iO) for classical circuits, and (trapdoor) dual-mode classical verification of quantum computation (CVQC), there exists an obfuscation scheme for null quantum circuits, i.e., any polynomial-size quantum circuit that rejects all inputs with overwhelming probability. The following statement summarizes our main result.
Theorem 1 (Informal). Assuming the hardness of the LWE problem, the existence of post-quantum iO for classical circuits, and a (trapdoor) dual-mode CVQC protocol, there exists quantum null-iO.

While the first assumption is standard, and the second seems to some extent necessary, dual-mode CVQC is a non-standard cryptographic building block that we introduce in this work. Loosely speaking, a CVQC protocol is dual-mode if there is a standard mode in which the scheme is correct, and a simulation mode in which there do not exist any accepting proofs for no instances (though in this mode, the scheme may not necessarily be correct for yes instances). These modes must be computationally indistinguishable even given the verification key.

Actually, we do not know of any construction of CVQC that satisfies this dual-mode property, so we instead relax the property to a “trapdoor” variant, where there exists a trapdoor setup algorithm (computationally indistinguishable from the original one) that satisfies the dual-mode property. We show that this relaxation suffices to construct quantum null-iO (along with LWE and post-quantum iO for classical circuits). To establish the feasibility of this building block, we present a construction of trapdoor dual-mode CVQC secure assuming the hardness of the learning with errors (LWE) problem in the quantum random oracle model (QROM).

Theorem 2 (Informal). Assuming the hardness of the LWE problem, there exists a trapdoor dual-mode CVQC protocol in the QROM.

As a cautionary note for the reader, we stress that our security proof relies on the programmability of the QROM. Furthermore, the quantum null-iO construction requires one to obfuscate the circuit computing the hash function. As such, combining this dual-mode CVQC protocol with our previous compiler, does not result in a provably secure construction of quantum null-iO under standard cryptographic assumptions, not even in the (Q)ROM. Nevertheless, our work establishes the first plausible candidate construction of quantum null-iO for which, prior to our work, we lacked even a heuristic scheme. We view the existence of a dual-mode CVQC in the QROM as evidence that our construction template is sound, and we leave constructing quantum null-iO under more conservative assumptions as a fascinating open problem (more discussion on this in Section 2.3).

We also propose an alternative construction of quantum null-iO secure only assuming LWE problem but with respect to an oracle. The oracle that we consider is entirely classical, but queriable in superposition. In fact, we show that the scheme is secure assuming LWE and post-quantum virtual black-box obfuscation of a particular classical circuit.

Witness Encryption for QMA. Applying a well-known transformation, we obtain witness encryption [25] for QMA as a corollary. Importantly, our scheme has an entirely classical encryption algorithm: Any classical user can encrypt a message $m$ with respect to the membership of some statement $x$ in a language $L \in \text{QMA}$. The message $m$ can be (quantumly) decrypted by anyone possessing (multiple copies of) a valid witness $|\psi\rangle \in R_L(x)$.

1.2 New Applications

We show that witness encryption for QMA with classical encryption enables a series of new cryptographic primitives, thereby positioning witness encryption for QMA (and consequently quantum null-iO) as a central catalyst in quantum cryptography. Most of our results are obtained via classical synthesis of quantum programs: We compress an exponential number
of quantum programs into a small classical circuit via the use of classical iO. We give an overview of the implications of our results below. We remark that, prior to our work, we did not even have a heuristic candidate for any of the primitives that we obtain.

(1) **NIZK for QMA:** We present the first construction of NIZK [13] for QMA. A (quantum) prover can efficiently produce a zero-knowledge certificate $\pi$ that a certain statement $x \in \mathcal{L}$, where $\mathcal{L}$ is any language in QMA. This certificate is publicly verifiable with respect to a publicly-known common reference string (CRS). Prior to our work, all non-interactive proof systems for QMA [19, 22, 4, 21] were in the secret parameters model or the designated verifier setting, i.e. the verifier (or additionally the prover) needed some secret information not accessible to the other party.

This resolves an outstanding open problem in the area (see e.g. [44] for a discussion on the barriers to achieving public verifiability). In addition, our NIZK scheme satisfies several properties of interest, namely, (i) it is statistically zero-knowledge, (ii) the verification algorithm (and CRS) is fully classical, and (iii) the verification algorithm is succinct, i.e. its runtime is independent of the size of the witness. In fact, we obtain the first succinct non-interactive argument (zk-SNARG) for QMA.

This primitive also implies the first classical verification of quantum computation (CVQC) scheme with succinct verifier (and large CRS) that is publicly verifiable, and thus reusable. This improves over the privately-verifiable scheme of [21] where the large CRS is not reusable, though we note that their CRS is actually a common random string as opposed to the common reference string required for our protocol.

(2) **ZAPR for QMA:** We show how to transform our NIZK for QMA scheme into a (publicly verifiable) two-round statistically witness-indistinguishable argument (ZAPR) for QMA. Our transformation is generic and can be thought of as a quantum analogue of the Dwork-Naor compiler [23], in the setting of computational soundness.

(3) **ABE for BQP:** We obtain a ciphertext-policy ABE [43, 32] scheme for BQP (bounded-error quantum polynomial-time) computation. In ciphertext-policy ABE for BQP, anyone can encrypt a message $m$ with respect to some BQP language, represented by a quantum circuit $Q$. The key authority can generate decryption keys associated with any attribute $x$. The ciphertext can then be decrypted if and only if evaluating $Q$ on $x$ produces 1 (which, by QMA amplification, can happen with either overwhelming probability or negligible probability). Interestingly, all algorithms except for the decryption circuit are fully classical. This is the first example of an ABE scheme for functionalities beyond classical computations.

The scheme satisfies the standard notion of payload-hiding. That is, the message $m$ is hidden, though the policy $Q$ is revealed by the ciphertext. We then show that we can upgrade the security of the scheme via a generic transformation to predicate-encryption security [29] (i.e. the policy $Q$ is hidden from the evaluator if they are only in possession of keys for rejecting attributes). We achieve this via a construction of lockable obfuscation [30, 46] for quantum circuits from LWE.

(4) **Constrained PRF for BQP:** We present a construction of a pseudorandom function (PRF) [14, 36, 15] where one can issue constrained keys associated to a quantum circuit $Q$. Such keys can evaluate the PRF on an input $x$ if and only if evaluating $Q$ on $x$ returns 1 with overwhelming probability. Otherwise, the output of the PRF on $x$ looks pseudorandom. The scheme is fully collusion-resistant, i.e. security is preserved even if an unbounded number of constrained keys is issued.

(5) **Secret Sharing for Monotone QMA:** Finally, as a direct application of our witness encryption scheme, we show how to construct a secret sharing scheme for access structures in monotone QMA.
2 Technical Overview

We give a cursory overview of the techniques introduced by our work and we provide some informal intuition on how we achieve our results. For precise statements and a formal analysis, we refer the reader to the full version [11].

2.1 How to Obfuscate Quantum Circuits

Before delving into the specifics of our approach, we highlight a few reasons why known techniques for obfuscating classical circuits do not seem to be directly portable to the quantum setting. For obvious reasons, we restrict this discussion to schemes that plausibly retain security in the presence of quantum adversaries. Recent proposals [18, 26, 45] follow the split fully homomorphic encryption (split-FHE) approach [17], which we loosely recall here. The obfuscator computes

$$\text{FHE}(C) \xrightarrow{\text{Eval}(x \cdot \cdot)} \text{FHE}(C(x)) \xrightarrow{\text{Hint}(sk \cdot \cdot)} h$$

where the decryption hint $h$ is specific to the ciphertext encoding $C(x)$. The obfuscated circuit consists of $(\text{FHE}(C), h)$ and the evaluator can recompute the homomorphic evaluation and use the decryption hint $h$ to recover $C(x)$. It turns out that, as long as $|h| \ll |C(x)|$, this primitive alone is enough to build full-fledged obfuscation. One crucial aspect of this paradigm is that the split-FHE evaluation algorithm is deterministic, which allows the obfuscator and the evaluator to converge to the exact same ciphertext.

Translating this approach to the quantum setting seems to get stuck at a very fundamental level: Known FHE schemes for quantum circuits [38] have an inherently randomized homomorphic evaluation algorithm. This means that the evaluator and the obfuscator would most likely end up with a different ciphertext even though they are computing the same function. This makes the hint $h$ (which is ciphertext-specific) completely useless to recover the output. A similar barrier also emerges in other generic transformations, such as [28], which are at the foundations of many existing obfuscation schemes.

Reducing to Classical Obfuscation. As a direct approach seems to be out of reach of current techniques, in this work we take a different route. Following the template of [24], our high-level idea is to outsource the quantum computation to some untrusted component of the scheme and instead obfuscate only the circuit that verifies that the computation was carried out correctly. The important point here is that verifying the correctness of quantum computation is a much easier task than performing the computation itself. In fact, it was recently shown [39] that the validity of any BQP computation can be verified by a completely classical algorithm, assuming the quantum hardness of the LWE problem. Furthermore, recent works [4, 21] have shown that the protocol can be collapsed to two rounds (in the random oracle model): On input a quantum circuit $Q$, the verifier produces some public parameters $pp$, which can be used by the prover (holding a quantum state $|\psi\rangle$) to compute a classical proof $\pi$. The verifier can then locally verify $\pi$ using some secret information $r$ which was sampled together with $pp$.

Using these classical verification of quantum computation (CVQC) protocols without any additional modification would allow us to implement the scheme as outlined above. However, we would not get any meaningful notion of privacy for the obfuscated circuit, since the prover needs to evaluate the circuit $Q$ in plain. Thus, we will need to turn these protocol blind: The prover is able to prove that $Q(|\psi\rangle) = y$ obliviously, without knowing $Q$. This can be done in a canonical way, using fully homomorphic encryption for quantum circuits (QFHE) with classical keys [38, 16].
Challenges Towards Provable Security. Although everything seems to fall in place, there is a subtle aspect that makes our attempt not sound: We implicitly assumed that the CVQC protocol is resettably secure. If the prover is given access to a circuit implementing a (obfuscated) verifier, nothing prevents it from rewinding it in an attempt to extract the verifier’s secret. Once this is leaked, the prover can fool the verifier into accepting false statements, ultimately learning some information about the obfuscated circuit. This is not only a theoretical concern, instead one can show concrete attacks against all known CVQC protocols (more discussion on this later). While this class of attacks seems to be hard to prevent in general, we observe that, if we restrict our attention to null (i.e. always rejecting) circuits, then this concern disappears. This is not a coincidence: Any two-round CVQC protocol that is one-time sound is automatically many-time sound for reject-only circuits, since it is easy to simulate the responses of the verifier (always reject).

Another challenge that we need to resolve is that of provable security: The classical obfuscation only provides us with the weak guarantee of computational indistinguishability for functionally equivalent (classical) circuits. Even if we restrict our attention to null circuits, our scheme still needs to hardwire the verifier’s secret in the obfuscated verifier circuit. That is, to obfuscate a null circuit \( Q \), we publish

\[
\text{pp and Obf (} \Pi_{\text{null-IO}}(\pi) : \text{Return CVQC. Verify}(\text{pp}, \pi, r))
\]

where \((\text{pp}, r) \leftarrow \text{CVQC.KeyGen}(1^\lambda, Q)\) is sampled by the obfuscator. To show security, we cannot simply switch the obfuscated circuit to reject all inputs: Since the CVQC protocol is only computationally sound, valid proofs \( \pi \) for false statements always exist, they are just hard to find. In particular, this means that the circuit \( \Pi_{\text{null-IO}} \) as defined above is not functionally equivalent to an always rejecting (classical) circuit.

Quantum Null iO from Trapdoor Dual-Mode CVQC. To address the above issue, we introduce the notion of dual-mode CVQC. As mentioned earlier, a dual-mode CVQC supports an alternative parameter generation algorithm \( \text{SimGen} \) where for any null circuit \( Q \) and \((\text{pp}, r) \leftarrow \text{CVQC.SimGen}(1^\lambda, Q)\), there do not exist any proofs \( \pi \) that accept with respect to the verification key \( r \). Furthermore, the parameters and verification key generated by \( \text{SimGen} \) should be computationally indistinguishable from those generated by \( \text{KeyGen} \).

Unfortunately, the two-message CVQC protocol mentioned above does not satisfy this dual-mode property.\(^1\) However, we observe that a weaker property, which we call the trapdoor dual-mode property, both suffices for quantum null-iO, and can be shown to exist in the quantum random oracle model. In a trapdoor dual-mode CVQC, the standard key generation algorithm \( \text{KeyGen} \) does not support the dual-mode property. However, there exists a “trapdoor” parameter generation algorithm \( \text{CVQC.TdGen} \) that returns the public parameters \( \text{pp}_{\text{td}} \) along with a secret key \( r_{\text{td}} \) and a trapdoor \( \text{td} \). The operation of the protocol in this trapdoor setting does actually satisfy the dual-mode property. In full detail, a trapdoor dual-mode CVQC satisfies the following properties:

- (Setup Indistinguishability) For all circuits \( Q \), the distributions

\[
\text{CVQC.KeyGen}(1^\lambda, Q) \approx_c (\text{pp}_{\text{td}}, r_{\text{td}})
\]

are computationally indistinguishable, where \((\text{pp}_{\text{td}}, r_{\text{td}}, \text{td}) \leftarrow \text{CVQC.TdGen}(1^\lambda, Q)\).

\(^1\) Note in particular that such a property actually implies publicly-verifiable CVQC, for which there were no known constructions prior to this work.
Indistinguishability Obfuscation of Null Quantum Circuits and Applications

- (Verification Equivalence) The algorithms

\[ \text{CVQC.Verify}(Q, r_{td}) \equiv \text{CVQC.TdVerify}(Q, r_{td}) \]

are functionally equivalent.\(^2\)

Moreover, in the trapdoor setting the scheme should satisfy the dual-mode property, using parameter generation algorithm SimGen.

- (Dual-Mode) For any null circuit \( Q \), the distributions

\[ (\text{pp}_{td}, r_{td}) \approx_c (\text{pp}_{sim}, r_{sim}) \]

are computationally indistinguishable, where \( (\text{pp}_{td}, \text{sk}_{td}, r_{td}) \leftarrow \text{CVQC.TdGen}(1^\lambda, Q) \) and \( (\text{pp}_{sim}, r_{sim}) \leftarrow \text{CVQC.SimGen}(1^\lambda, Q) \). Moreover, the circuit \( \text{CVQC.TdVerify}(Q, \cdot, r_{sim}) \) has no accepting inputs.

Deferring for the moment the discussion on how to actually construct a trapdoor dual-mode CVQC, we now argue that the above properties suffice for constructing quantum null-iO. Our obfuscation scheme will make use of quantum fully-homomorphic encryption to switch the parameter generation to use

\[ \text{QFHE.Obfuscation}(\cdot) \]

four hybrids. This establishes that the obfuscation of \( \text{QFHE} \) to switch the parameter generation to use

\[ \text{null-iO} \].

Finally, applying a known transformation, we obtain a witness encryption scheme for QMA as an immediate corollary.

A Trapdoor Dual-Mode CVQC Protocol in the QROM. What is left to be shown is an instantiation of a trapdoor dual-mode CVQC protocol. While we do not know of a scheme secure in the standard model (in fact, we do not yet know how to construct any two-message CVQC secure in the standard model, even without the trapdoor dual-mode

\(^2\) The astute reader may wonder why we cannot use the trapdoor mode as actual verification procedure, since they are anyway functionally equivalent. Looking ahead to our construction, the reason is that the trapdoor mode will become available only in the QROM.
property), we present a construction secure assuming the hardness of the LWE problem in the QROM. In fact, we show how to compile any two-message CVQC protocol (such as the one discussed above), into a trapdoor dual-mode CVQC. We make use of a random oracle $H : \{0,1\}^* \rightarrow \{0,1\}^{\lambda+1}$ that may be queried in superposition. The modified protocol simply consists of running the CVQC prover and hashing the resulting proof

$$\pi \leftarrow \text{CVQC.Prove}(pp, |\psi\rangle) \text{ and } h = H(\pi)$$

whereas the verification algorithm checks the consistency of $h = H(\pi)$, in addition to running the CVQC.\text{Verify} algorithm on $\pi$.

In the trapdoor setting, we move the computation of the verification algorithm CVQC.\text{Verify}(pp, \pi, r) into the specification of the random oracle $H$. That is, we replace the last bit of the random oracle output with an encryption of CVQC.\text{Verify}(pp, \pi, r), under a secret key $td$ that functions as the trapdoor. Then, the trapdoor verification algorithm no longer requires the CVQC secret parameters: It can instead use $td$ to decrypt the last bit of the random oracle output $h$ in order to uncover the result of CVQC.\text{Verify}(pp, \pi, r). To implement this, we use a quantum-secure PRF $F$ with key $td$. On input some proof $\pi$, we set the first $\lambda$ bits of the random oracle to be uniformly sampled and the last bit to

$$F(td, \pi) \oplus \text{CVQC.Verify}(pp, \pi, r).$$

Note that the verifier (TdVerify) can equivalently check the validity of $\pi$ by simply recomputing $F(td, \pi)$ and unmasking the response that was already computed in the random oracle.

Now, it remains to show how we obtain the dual-mode property in the trapdoor setting. This follows by letting SimGen simply be the same as TdGen, except that CVQC.\text{Verify}(pp, \pi, r) is replaced with 0 always in the implementation of the random oracle. That is, the last bit of the random oracle output is always $F(td, \pi)$, and thus, verification using $td$ will always output 0. To show that this is computationally indistinguishable from CVQC.TdGen for any null circuit $Q$, we observe that any adversary that can distinguish these oracles can be used to break the soundness of CVQC. This follows by specifying a reduction that measures one of the adversary’s oracle queries to obtain an accepting proof with noticeable probability.

### 2.2 Applications

Next, we explore some applications of our newly constructed null-iO for quantum circuits.

Since it is the weaker primitive, we are going to use witness encryption for QMA (with classical ciphertexts) as the starting point for all of our primitives.

**NIZK for QMA.** Our first result is a construction of NIZK arguments for QMA with public verifiability. To build up some intuition about the protocol, consider the simplified setting where we have a single fixed statement $x$. We can then define the common reference string to be

$$(vk, \text{WE.Enc}(x, \sigma)) \text{ such that } \text{Verify}(vk, \sigma, x) = 1$$

where $vk$ is a verification key of a signature scheme. Anyone with a valid witness $|\psi\rangle$ for $x$ can recover the signature by decrypting the ciphertext. Then anyone can verify the validity of $x$ by simply verifying the signature $\sigma$ under $vk$. Note that this computation is entirely classical and succinct: It’s runtime does not depend on the computation needed to compute $|\psi\rangle \in R_C(x)$. To extend this approach to an exponential number of statements, we exploit
the fact that our witness encryption scheme has a \textit{completely classical} encryption procedure. Our idea is to place in the common reference string the obfuscation of the (classical) circuit

$$\Pi_{\text{NIZK}}[sk, k](x) : \text{Return } \text{WE.Enc}(x, \sigma; \text{PRF}(k, x))$$

where PRF is a puncturable PRF [42]. The prover algorithm can then evaluate the obfuscated circuit on $x$ to obtain $\text{WE.Enc}(x, \sigma)$ and proceed as before. Such an approach can be shown to be sound via a standard puncturing argument.

\textbf{ZAPR for QMA.} The next question that we ask is whether we can reduce the trust on the setup and obtain some meaningful guarantees also in the presence of a maliciously generated common reference string. Known generic transformations [23] do not apply to our case, since the NIZK that we obtain is an argument, i.e. it has computational soundness. Our saving grace is again the fact that our NIZK has completely classical setup and verification procedures. This allows us to leverage powerful tools from the literature of (classical) zero-knowledge. We adopt a dual-track approach (reminiscent of the Naor-Yung [41] paradigm) where we define the setup to sample two copies ($\text{crs}_0, \text{crs}_1$), the image of a one-way function $y$, and a \textit{classical} non-interactive witness indistinguishable (NIWI) proof that either $\text{crs}_0$ or $\text{crs}_1$ is correctly generated.

At this point, it is still unclear whether we have any privacy guarantee, since one of the two strings can be maliciously generated and can therefore leak some information about the witness, if naively used by the prover. Thus, instead of having the prover directly compute the proofs $(\pi_0, \pi_1)$, we let it compute a classical NIWI for the statement

$$\begin{cases} \exists (\pi_0, \pi_1, z) \text{ such that:} & \text{Verify}(\text{crs}_0, \pi_0, x) = 1 \text{ OR} \\ & \text{Verify}(\text{crs}_1, \pi_1, x) = 1 \text{ OR} \\ & \text{OWF}(z) = y. \end{cases}$$

Since the verification algorithm of our NIZK scheme is classical, then so is the above statement. By the witness indistinguishability of the NIWI, the verifier cannot distinguish whether the prover inverted the one-way function or possesses a valid proof. Proving soundness requires more work, since our NIZK scheme is only \textit{computationally sound}. To get around this, we further augment the scheme with a statistically hiding sometimes-binding (SBSH) commitment [35, 31, 9]: This tool allows the prover to commit to its witness, which is statistically hidden, except with some (exponentially) small probability where the commitment is efficiently extractable. We can then set the parameters of our primitives to be sufficiently large (i.e. use complexity leveraging) to ensure that whenever the extraction even happens, it still leads to a contradiction to the soundness of the NIZK or to the one-wayness of OWF.

\textbf{ABE for BQP.} We next show how our witness encryption scheme for QMA yields the first ABE scheme for quantum functionalities. For starters, consider again the simplified setting where the key authority issues a single key for a fixed attribute $x$. The witness encryption suggests a natural ABE encryption procedure for a policy encoded by a quantum circuit $Q$: We compute $\text{WE.Enc}((Q, x), m)$, where the statement $(Q, x)$ returns 1 if and only if $Q(x) = 1$. Note that this is technically a BQP statement (the witness is publicly computable), but a witness encryption for QMA is also a witness encryption for BQP. The challenge is now to define an encryption algorithm that does not need to take the attribute $x$ as an input. Instead of publishing the ciphertext directly, the encrypter obfuscate the (classical) circuit

$$\Pi_{\text{ABE}}[vk, k](x, \sigma) : \text{If } \text{Verify}(vk, \sigma, x) = 1 \text{ return } \text{WE.Enc}((Q, x), m; \text{PRF}(k, x))$$
where PRF is a puncturable PRF and vk is the verification key for a signature scheme. A secret key for an attribute x simply consists of a signature σx on x, computed with a signing key held by the key authority. This way, the holder of a key for an attribute x can only extract witness encryption ciphertexts associated with x. Some additional work is needed in order to obtain a provably secure scheme, but the main ideas are already present in this outline.

The scheme as described so far does not hide the circuit Q, i.e. it only satisfies the notion of payload hiding. We show a generic compiler that transforms any quantum ABE with payload-hiding security into one with predicate encryption security, i.e. where the circuit Q is hidden to the holders of keys for rejecting attributes. This result is obtained by introducing the notion of quantum lockable obfuscation and presenting a construction under LWE.

**Constrained PRF for BQP.** Given the above ABE scheme for BQP, one can easily turn any puncturable PRF into a constrained PRF. For convenience, here we consider a key-policy ABE, which can be obtained from the scheme as described above via universal (quantum) circuits. The public parameters of the PRF are augmented with an obfuscated circuit

\[ \Pi_{\text{PRF}}[k, \tilde{k}] : \text{Return } \text{ABE.Enc}(x, \text{PRF}(k, x); \text{PRF}(\tilde{k}, x)) \]

where \( \tilde{k} \) is an independently sampled key. Note that anyone can query such circuit on any attribute x, however only the holder of a key for a policy Q such that \( Q(x) = 1 \) (with overwhelming probability) can recover the PRF output \( \text{PRF}(\tilde{k}, x) \) by decrypting the resulting ciphertext. Furthermore, observe that the functionality specified above is entirely classical, and therefore classical obfuscation suffices.

**Secret Sharing for Monotone QMA.** It is well-known that witness encryption for NP implies the existence of a secret sharing scheme for monotone NP [37]. The high-level idea is to assign to each party \( P_i \) the opening of a perfectly binding commitment \( c_i = \text{Com}(i; r_i) \) encoding the index corresponding to the party. Then one can publish a witness encryption for the statement

\[ \{ \exists (I \subseteq P, r_1, \ldots, r_{|I|}) \text{ such that: } I \in \mathcal{L} \text{ AND } \forall i \in I : c_i = \text{Com}(i; r_i) \} \]

where I, parsed as a binary string, forms a statement in a NP-complete language \( \mathcal{L} \) with witness w. It is not hard to show that decrypting the witness encryption (i.e. reconstructing the secret) can only be done by an authorized set of parties holding the witness w. We show that this construction naturally generalizes to the QMA setting, when given a witness encryption scheme for QMA.

### 2.3 Discussion and Open Problems

We discuss two clear open problems that are suggested by this work. We identify barriers towards making progress on each problem with our current approach.

**Obfuscation Beyond Null Circuits.** In this work, we only consider obfuscating the CVQC verification circuit in the setting where each instance the prover can query will be rejecting (with high probability). One could also consider obfuscating the verification circuit in the setting where the prover can query on an accepting instance, which would help in constructing fully-fledged iO for all quantum circuits. We expect obfuscation for general quantum circuits to have a variety of applications and we consider it a fascinating problem in its own right.
Unfortunately, it turns out that this approach is in general insecure and concrete attacks exist against all known constructions of CVQC. We provide a high-level description of these attacks in the full version [11]. The main source of trouble appears to be the lack of resettable security of CVQC protocols. That is, an attacker is able to extract the verifier’s secret by observing its responses on accepting instances.

**Quantum Null-iO from Standard Assumptions.** Another natural question is whether one can obtain quantum null-iO from standard cryptographic assumptions. Following our approach, this would require instantiating the dual-mode CQVC in the standard model (as opposed to the QROM). We wish to stress that the usage of the QROM that we make is non-standard in two ways: (i) the trapdoor verification procedure is only well-defined in the QROM, and crucially relies on programming the random oracle,\(^3\) (ii) the classical obfuscated circuit needs to hardwire the description of the hash function. It follows that the resulting quantum null-iO does not achieve provable security under standard cryptographic assumptions, not even in the (Q)ROM.

Note that even without the dual-mode property, two-message CVQC protocols are only known in the QROM. One approach towards evading this barrier could be to instantiate the base CVQC protocol with the two-message protocol of [40] (with quantum first message), which is statistically sound. This would result in a valid quantum null-iO since the first message can be computed by the obfuscator and sent along with the obfuscated circuit. However, even in this case, the verification circuit in [40] will accept exponentially many proofs even for no instances, as the underlying delegation of quantum computation protocol has a probabilistic verifier (that chooses which Hamiltonian terms to measure on each copy of the history state). Even though soundness can be driven to negligible by parallel repetition, this also rapidly increases the proof size. Thus, attempting to hybrid over each proof will fail, since the number of hybrids will be much larger than inverse of the soundness error. Given this barrier, we leave constructing any of the primitives discussed in this work in the standard model and from standard cryptographic assumptions, as an intriguing open problem.

In the full version [11], we present an alternative construction of quantum null-iO assuming classical virtual-blackbox (VBB) obfuscation. While it is known that VBB obfuscation is in general impossible [10], classical VBB obfuscation has been used as a heuristic method to analyze the security of certain schemes. Recent examples include fully-homomorphic encryption for RAM programs [34] and one-shot signatures [7]. As another example, Aaronson and Christiano [2] made use of ideal classical obfuscation to establish the feasibility of public-key quantum money. This influential result inspired a fruitful line of research, including a result by Zhandry [47] that showed how to instantiate their original approach from post-quantum indistinguishability obfuscation.

**References**


\(^3\) In particular, this means that the trapdoor verification procedure does not exist in any standard model instantiation of the CVQC scheme (with a concrete hash function). While it is true that the trapdoor verification procedure is only used in the security proof when using dual-mode CVQC scheme to construct quantum null-iO, this discussion makes it clear that we currently do not even have any standard model candidates of a dual-mode CVQC.


Indistinguishability Obfuscation of Null Quantum Circuits and Applications


