Quantum Garbled Circuits

Zvika Brakerski†
Weizmann Institute of Science
Rehovot, Israel

Henry Yuen‡
Columbia University
New York, NY, USA

ABSTRACT
In classical computing, garbled circuits (and their generalization known as randomized encodings) are a versatile cryptographic tool with many applications such as secure multiparty computation, delegated computation, depth-reduction of cryptographic primitives, complexity lower-bounds, and more. Quantum analogues of garbled circuits were not known prior to this work.

In this work, we introduce a definition of quantum randomized encodings and present a construction which allows us to efficiently garble any quantum circuit, assuming the existence of quantum-secure one-way functions. Our construction has comparable properties to the best known classical garbling schemes. We can also achieve perfect information-theoretic security albeit with blowup in the size of the garbled circuits.

We believe that quantum garbled circuits and quantum randomized encodings can be an instrumental concept and building block for quantum computation and in particular quantum cryptography. We present some applications, including a conceptually-simple zero-knowledge proof system for QMA, a protocol for private simultaneous messages, functional encryption, and more.

CCS CONCEPTS
• Theory of computation → Quantum complexity theory;
• Cryptographic primitives: Cryptographic protocols.

KEYWORDS
quantum randomized encodings, quantum garbled circuits

ACM Reference Format:
Zvika Brakerski and Henry Yuen. 2022. Quantum Garbled Circuits. In Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing (STOC ’22), June 20–24, 2022, Rome, Italy. ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3519935.3520073

1 INTRODUCTION
The notion of randomized encoding (RE) is central in cryptographic research and allowed for numerous advances in various different contexts, perhaps most famously in the context of secure computation. A randomized encoding of a function f is another function  ̂f, computed probabilistically, such that on every input x, the output  ̂f(x) can be recovered from  ̂f(x), and no other information about x is conveyed by  ̂f(x). RE is useful when the encoding  ̂f is easier to compute in some sense than f itself. 1 As a simple example, consider the n-bit XOR function  ̂f(x) = f(x1, . . . , xn) = x1 ⊕ · · · ⊕ xn, this function requires a log(n)-depth circuit to compute. In contrast, consider the function  ̂f which takes as input x = (x1, . . . , xn), in addition to n random bits r = (r1, . . . , rn), and has an n-bit output:

  ̂f(x; r) = (rn ⊕ x1 ⊕ r1, r1 ⊕ x2 ⊕ r2, . . . , rn−1 ⊕ xn ⊕ rn).

The function  ̂f can be computed in constant depth and we argue that it constitutes an encoding of f in the following sense. First, the value f(x) can be derived from  ̂f(x; r), for any r simply by applying XOR to all output bits. This is the decoding of f(x) out of its encoding. 2 Second, for any x, if r is sampled at random then the distribution  ̂f(x; r) depends only on f(x) and not on x itself, in the sense that it is possible to sample from this distribution using the value f(x). In the example above this is done by sampling a set of n random bits whose XOR equals f(x). This is the simulation of  ̂f(x; r) out of the decoded value. Simulation implies privacy by showing that  ̂f(x; r) does not reveal any information about x except for the value of f(x). One often considers statistical or even computational privacy where the simulator samples from a distribution that is statistically or computationally indistinguishable from  ̂f(x; r).

Indeed, the XOR example is quite simple, but the famous garbled circuit construction of Yao [53] (see also [13, 49]) shows that a similar property can be achieved for any circuit. In particular, any circuit admits an encoding whose depth is independent of the circuit complexity. Furthermore, the encoding is decomposable: each output bit of  ̂f depends on at most a single bit of x. In other words, for all r the function  ̂f(·; r) is a linear function. The immediate consequence was that if one can construct secure two-party computation protocols for linear functions (a notion known in cryptography as "oblivious transfer" or OT), then it can be extended to securely compute any function f. Concretely, assume that party A holds some multi-bit value x, and party B holds a multi-bit value y. The goal is for party A to learn f(x, y) where f is some function, but nothing else about y. 3 To do this, party B samples a random r and prepares a description of a sequence of linear functions corresponding to the output bits of  ̂f(·; y; r). The parties then use OT in order to allow party A to learn exactly  ̂f(x, y; r). This, as we explained, conveys

1Supported by the Binational Science Foundation (Grant No. 2016726), and by the European Union Horizon 2020 Research and Innovation Program via ERC Project REACT (Grant 756482) and via Project PROMETHEUS (Grant 780701).
2Supported by AFOSR award FA9550-21-1-0040 and NSF CAREER Award 2144219.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires specific permission and/or a fee. Request permissions from permissions@acm.org.
STOC ’22, June 20–24, 2022, Rome, Italy.
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9264-8/22/06...
https://doi.org/10.1145/3519935.3520073

804
no more information than the value \( f(x, y) \) alone. Since OT can be very round efficient (can be computed in two message in many settings), this would imply round efficiency for secure computation for arbitrary functions.

Garbled circuits have been used time and again to leverage limited cryptographic capabilities (ones that only apply to linear or constant depth functions) to achieve cryptographic functionalities for arbitrary computations. This includes numerous applications to secure multi-party computation such as \([13, 15, 29, 30, 37, 38, 41, 53]\) and probably hundreds more. functional encryption \([32, 33, 50]\), key-dependent message security \([5, 11]\), homomorphic encryption \([31, 46]\), just to name a few. The notion of CRE (of which garbled circuits are a special case) was formally defined by Applebaum, Ishai and Kushilevitz in \([9, 10]\) and we refer the readers to an extensive survey by Applebaum \([8]\) for additional details and references. In Appendix A we provide a brief comparison between RE and other cryptographic notions (multiparty computation, homomorphic encryption, and obfuscation) which may be helpful for the reader who is not familiar with the cryptographic literature.

Given the richness and utility of randomized encodings in cryptography and theoretical computer science, it is very natural to ask whether there exists a quantum analogue of randomized encodings. Despite its appeal, this question has remained open, and as far as we know the notion was not even formally defined in the literature before.

The full version of this paper is available at \(https://arxiv.org/abs/2006.01085\). Readers are encouraged to refer to the full version.

### 1.1 Our Contributions

In this paper we introduce the notion of randomized encodings in the quantum setting, propose a construction, and analyze it. Our definition is an adaptation of the classical one: the quantum randomized encoding (QRE) of a quantum operation \( F \) (represented as a quantum circuit) and a quantum state \( x \) is another quantum state \( \hat{F}(x) \) satisfying two properties:

1. **Correctness.** The quantum state \( F(x) \) can be decoded from \( \hat{F}(x) \).
2. **Privacy.** The encoding \( \hat{F}(x) \) reveals no information about \( x \) apart from the output \( F(x) \).

As in the classical case, privacy is formalized using the notion of a simulator. Correctness and privacy should hold even with respect to quantum auxiliary input - which is necessary for many cryptographic applications. We also define decomposability for QRE as an encoding where each input (qu)bit is acted on separately.³

We then present a construction of a decomposable QRE, which we call the Quantum Garbled Circuits scheme:

**Theorem 1.1 (Main result, informal).** Suppose CRE is a classical DRE scheme with perfect correctness, information-theoretic (resp. computational) privacy, and polynomial time decoding. Then there exists a decomposable CRE scheme QGC with the following properties:

1. QGC has perfect correctness and polynomial-time decoding.

(2) QGC uses CRE as a black box, and has information-theoretic (resp. computational) privacy.

(3) If the encoding procedure of CRE can be computed in \( \text{NC}^0 \), then the encoding procedure of QGC can be computed in constant quantum depth.⁵

Similarly to classical garbled circuits, our construction operates in a gate-by-gate manner, where each gate (including "input gates") is garbled using a "gadget", thus achieving decomposability with respect to both inputs and gates. If some input (qu)bit turns out to be a classical bit, then the corresponding gate gadget is a classical linear function.

We provide an overview of the construction and the ideas behind its analysis in Section 2. We refer the reader to Section 4.2 for a formal statement of our main result.

### 1.2 Applications

Many of the applications of RE in the classical setting could carry over to the quantum setting. We briefly describe some potential applications, and formally elaborate on one of them (a zero-knowledge \( \Sigma \) protocol) in the technical part of the paper.

**A New Zero-Knowledge \( \Sigma \)-Protocol for QMA.** We show that it is possible to obtain a public-coin zero-knowledge proof protocol for QMA with the special \( \Sigma \) ("Sigma") form. In a \( \Sigma \)-protocol, the prover sends a first message, then the verifier responds with a random string as challenge, and then the prover sends a last message. These protocols are appealing in many settings and in particular are amenable, in the classical world, to "compression" into a non-interactive zero-knowledge (NIZK) protocol \([28]\).

Broadbent and Grilo \([20]\) recently presented the first ZK \( \Sigma \)-protocol for QMA. However, we show that using QGC, we can adapt a classical protocol almost effortlessly, substituting a classical for a quantum garbled circuit. Our protocol has additional favorable features compared to \([20]\), such as having the "delayed input" property where only the prover’s last message depends on the instance and witness.

**PSM.** Private Simultaneous Messages (PSM) \([27]\) is a form of secure multi-party computation in which a set of parties (say two) hold individual inputs \( x, y \) for a two input function \( f(\cdot, \cdot) \). The parties are required to each send a single message \( m_x, m_y \) respectively to a third party, which should learn the value \( f(x, y) \) and nothing else. The two parties do not know each other’s input but are allowed to share a random tape. This primitive has been very useful in the design of secure multi-party computation protocols.

Prior to this work it was not known whether PSM for quantum functions and quantum inputs was at all possible - regardless of communication complexity. QGC immediately implies an PSM protocol in the quantum setting.

**Delegation.** QGC implies that any quantum computation can be delegated (in 2 messages) using a constant depth verifier. This is done by simply sending a QGC that computes the circuit and if it accepts then output some pre-designated random string that is known to the verifier. We note that a recent paper of Morimae \([44]\) showed that 2-message verifiable quantum computing is possible.

³In our main construction, the encoding is a quantum circuit that does not involve measurements and is computable in the class \( \text{QNC}^0 \) (i.e. the class of constant-depth quantum circuits with unbounded fan-out gates, see \([36, 43]\)).
Two-Party Secure Computation. It is fairly straightforward to adapt to the quantum setting Yao’s protocol for two-message secure two-party computation in a “semi-honest” setting (see Appendix B.1 for details). A followup work by Bartusek et al. [12] relied on our QGC to minimize the round complexity of secure two-party computation under a fully malicious security definition, thus demonstrating their usefulness in this setting. One could consider further applications in the context of MPC such as improved quantum MPC in the multi-party setting and in the malicious setting [23, 26].

Functional Encryption. It was noticed in [50] that decomposable RE schemes imply a limited form of functional encryption (FE). This is an encryption scheme where there are multiple secret keys, associated with functions, so that when key_\text{f} decrypts \text{Enc}(x) the output is f(x). In a similar way, QGC can be directly used to construct private-key quantum functional encryption schemes, which to our knowledge have not been studied before. See Appendix B.2 for more details.

Obfuscating Quantum Circuits. If it is possible to construct a QRE with classical encoding for classical inputs and function descriptions, then it would allow to construct quantum indistinguishability obfuscation (iO) from the classical variant, similarly to how classical RE is leveraged to obtain classical iO [6, 16]. See discussions on the barriers towards classical garbling in Appendix B.3. Constructing iO for quantum circuits is one of the intriguing open problems in the context of quantum cryptography, and with the connections between iO and RE in the classical setting, one would hope that QRE could be a useful tool in establishing it.

1.3 Our Techniques

Our aim is to encode the function \( f \) in a decomposable manner. Drawing inspiration from classical garbled circuits, our garbled circuit will include an encrypted version of the inputs, then for each gate we provide a gadget that converts an encrypted input to an encrypted output for that specific gate. Finally for the output gate we provide the keys to decrypt the resulting value.

Our first challenge is to devise the right notion of “encryption” for the task. In the classical setting, the gate gadget includes an encrypted version of the truth table of the gate which allows to map encrypted inputs to encrypted outputs. A quantum gate seems not to have a natural notion of “truth table” that will be applicable in this setting. To address this challenge, we observe that quantum teleportation can be used as a notion of encryption that has seemingly-desirable properties as we explain below.

Quantum teleportation allows two parties that share two ends of an EPR pair to communicate a quantum state to one another using classical communication. The sender applies a certain measurement to the “near end” of the EPR pair, together with the state to be sent. The classical outcome of the measurement is sent to the other party and allows it to perform a “correction” on the “far end” of the EPR pair and recover the state that was sent. A simple observation is that prior to receiving the classical information, the “far end” can be thought of as an encrypted version of the original state, where the key is the classical information that allows recovery. Furthermore, this method of encryption makes sense also in terms of semantics of the system. It turns out that it is possible to first apply a gate (for the Clifford+T basis) on the “far end” of the EPR pair, and then perform the correction on the output of the gate. We view this as the correct quantum analog of the encrypted truth table. Namely, we apply a gate to an encrypted version of the input in such a way that given the correct key, the output can be decrypted. The above can be recognized as the famous computation by teleportation paradigm [35].

Indeed, if our circuit contained only one layer of gates, the above intuition could have been made to work. The input would have been teleported onto “near ends” of EPR pairs, and the “far ends” would have been fed into the gates. Then, based on the classical measurements of the teleportation, it would have been possible to compute the correct “decryption keys”. We note that it is important not to send the teleportation measurements in the clear (as those would allow to simply decrypt the input, whereas the security requirement requires that only the output of the circuit is revealed), but rather compute locally the proper output decryption keys and only output those. We note that the computation of the decryption key is a completely classical computation that can be performed in low depth by using a classical garbled circuit.

Our next challenge, therefore, is how to propagate the encrypted state properly down the circuit. Unfortunately, it is not possible to simply teleport the output of the first gate into a new “near end” of an EPR pair (whose “far end” is connected to the next gate). The reason is that reversing the order of the correction procedure and the gate application changes the type of encryption: whereas if we first perform a correction and then apply a gate the decryption of our scheme is a Pauli operation, when we reverse the order we end up with a decryption which is a Clifford operation. (For the uninformed reader, it suffices to know that Clifford operations are a strict superset of Pauli operations.) To summarize, our gate gadget produces outputs under a different type of encryption than the one that it takes as input, which creates a mismatch.

As an important side-note, we point out that if all gates in our circuit are Clifford then the problem above does not arise, and the output of the gate gadget is natively Pauli-encrypted. This already implies a method for garbling for the restricted case of Clifford circuits. Since we want to address general circuits, our next challenge, is to come up with an “encryption converter” from Clifford encryption to Pauli encryption. To address this challenge, we consider the circuit that does the conversion “in the clear”. It takes the keys to the Clifford encryption, uses it to decrypt the value, and then teleports this value along the “near end” of the next EPR, thus associating the “far end” of this EPR with a properly Pauli encrypted version of this value. It is important to notice that we cannot compute this circuit at garbling time because the Clifford decryption keys are a function of all prior gates, and we need to garble each gate individually. However, importantly, the encryption converter is an efficient classical function of the Clifford decryption key. We can therefore devise a classical garbled circuit for the classical function that takes the Clifford decryption key and produces the encryption converter circuit.

This hints, therefore, that we may want to garble the encryption conversion circuit itself (not just its classical description). This will
essentially complete our construction. We address this challenge in two steps. First, we notice that if the encryption converter was a Clifford circuit, then we could have garbled it using a method that we call “group randomizing RE”, and draws from classical garbling literature [40] as well as quantum secure computation literature [2, 23, 25, 26]. Second, we utilize the fact that performing a measurement (which is not even a unitary operation) is equivalent to applying a gate chosen from a distribution over Cliffords.3 Putting these two ideas together we can quantumly garble the encryption converter and complete our construction.

We provide a more detailed technical overview in Section 2.

1.4 Other Related Work

We mention some related work on adapting the notion of randomized encodings/garbled circuits to the quantum setting. In [39], Kashefi and Wallden present an interactive, multi-round protocol for verifiable, blind quantum computing, that is inspired by Yao’s garbled circuits. The motivation for their protocol comes from wanting a protocol where a weak quantum client delegates a quantum computation to a powerful quantum server, while still maintaining verifiability.

In a recent paper [55] (which builds on prior work [54]), Zhang presents a blind delegated quantum computation protocol that is (partially) “succinct”: it is an interactive protocol with an initial quantum phase whose complexity is independent of the computation being delegated, and the second phase is completely classical (with communication and round complexity that depends on the size of the computation). The security of the protocol is proved in the random oracle model. The construction and analysis appear to use ideas from classical garbled circuits.

Both the work of [39] and [55] focus on protocols for delegated quantum computation, and both protocols involve a large number of rounds of interaction that grow with the size of the computation being delegated. In contrast, the focus of our work is on studying the notion of quantum randomized encodings (in which the number of rounds of interaction is constant).

2 DETAILED TECHNICAL OVERVIEW

We provide an overview of our Quantum Garbled Circuits construction and its analysis. We refer to the technical sections for the formal presentation and proofs. In what follows, we use bolded variables such as $q$ to denote a quantum state (via its density matrix), and for a unitary $U$ we write $U(q)$ to denote the application of $U$ on $q$, i.e. the state $UqU^\dagger$ (see Section 3 for more details about notation). We let $F$ denote a quantum operation acting on $n$ qubits which is computed by a circuit consisting of gates $G_1, G_2, \ldots, G_m$ drawn from a universal gate set (specifically, Clifford + T). Let $x$ denote an $n$-qubit state.

2.1 The Starting Point: Computation via Teleportation

The starting point of our approach to quantum RE is computation by teleportation (also known as gate teleportation), an idea that is common to many prior results on protocols for delegated quantum computation and computing on encrypted data [19, 21, 24, 35]. Recall that an EPR pair is the two-qubit state $|EPR\rangle = \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle)$. We use the notation $e = (e_1, e_2)$ to denote the density matrix $|EPR\rangle \langle EPR|$, and when we write $e_j$ by itself we mean the reduced density matrix on the $j$th qubit.

The idea behind gate teleportation is as follows. Let $x$ denote a qubit density matrix, and let $G$ denote a single-qubit gate (the generalization to multi-qubit gates is straightforward). One way of applying $G$ to $x$ is to first apply $G$ to one half of an EPR pair $e = (e_1, e_2)$, resulting in $(I \otimes G)(e) = (e_1, G(e_2))$. Then applying the standard teleportation circuit on qubits $x$ and $e_1$ (the “input half” of the EPR pair) and obtaining measurement outcomes $(a, b)$.

\[
x \xrightarrow{\text{I}} a \quad b
\]

Figure 1: Teleportation circuit.

The qubit $G(e_2)$ (the “output half”) collapses to $G(X^a(Z^b(x)))$ where $X$ and $Z$ are the Pauli bit-flip and phase-flip matrices respectively. The measurement outcomes $(a, b)$ are often called teleportation keys. We can rewrite the state as $E(G(x))$ where $E$ is the unitary $GX^aZ^bG^\dagger$, which we call an error. By performing the correction operation $E^\dagger$ on this state, we get the intended state $G(x)$.

We can apply this approach to an entire quantum circuit $F$ consisting of gates $G_1, G_2, \ldots, G_m$. For every wire of the circuit (which one can think of as a line segment in a circuit diagram in between the gates, as well as the segments for the inputs/outputs qubits), we introduce a new EPR pair. Each gate $G_i$ of the circuit is applied to the “output” ends of the EPR pairs corresponding to the wires that lead into gate $G_i$. Let $s$ denote the density matrix corresponding to this entangled state; intuitively it represents the circuit $F$ in quantum state form. See Figure 2 for a depiction of the correspondence between $F$ and the EPR pairs (for clarity we omit the EPR pairs corresponding to the input and output wires).

\[
G_1 \quad G_2 \quad G_4
\]

(a) An example of a quantum circuit $F$.

\[
G_1 \quad G_2 \quad G_4
\]

(b) A representation of resource state associated with $F$.

Figure 2: Decomposing a circuit into EPR pairs.

Suppose we didn’t care about maintaining privacy of the circuit and input. Consider the pair $(x, s)$, which forms an encoding of
We need a way to ensure that the evaluator computes the product $O$ on actual teleportation keys.

We use classical randomized encodings to solve this problem.

2.2 Using Classical Randomized Encodings

The key idea in our construction is the following: the evaluator also gets a classical randomized encoding of the function $f(a, b)$ that maps teleportation keys to the unitary $O$. Furthermore, the teleportation circuits are modified so that, instead of the measurement outcomes being two bits $(a, b)$, they are two long random strings $(f_x, a, f_z, b)$ called labels which encode the teleportation keys with respect to the classical randomized encoding scheme. These labels, combined with the randomized encoding of the function $f$, allows the evaluator to compute $O = f(a, b)$ and learn nothing else.

Let us be more precise. Fix a decomposable randomized encoding scheme $(f, a, b) \mapsto \hat{f}(a, b)$; one can instantiate this with Yao’s garbled circuits, for example. Because of decomposability, we can write

$$\hat{f}(a, b; r) = (\hat{f}_{\text{off}}, f_x, a, f_z, b)$$

where $r$ is a random string, $\hat{f}_{\text{off}}$ is a string (known as the “offline encoding”) that only depends on $f$ and $r$, $f_{x,a}$ is a string that only depends on $a$, $f_z$ and $r$, and $f_{z,b}$ is a string that only depends on $b$, $f$, and $r$. Borrowing terminology from Yao’s garbled circuits, we call $f_{x,a}$ and $f_{z,b}$ labels for $a$ and $b$, respectively.

We now describe the modified teleportation circuit, which we call the encrypted teleportation gadget. Fix labels $\ell = (\ell_x, \ell_z, \ell_{x,0}, \ell_{z,0}, \ell_{x,1})$ and bits $s = (s_x, s_z), t = (t_x, t_z) \in \{0, 1\}^2$. The gadget, denoted by $\text{TP}_{r,f,l,s}$, is depicted below, where for a string $r \in \{0, 1\}^k$, the gate $X(r)$ denotes the $k$-qubit gate that applies a bit-flip gate $X$ to qubit $i$ if $r_i = 1$.

```
\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{teleportation_gadget.png}
\caption{Encrypted teleportation gadget.}
\end{figure}
```

The functionality of this circuit is to perform the teleportation measurement on qubits $(x, e_1)$ as before, but instead of obtaining teleportation keys $(a, b) \in \{0, 1\}^2$ as usual, the middle two wires yield teleportation labels $(f_{x,a}, f_{z,b})$ which indicate the Pauli error $X^aZ^b$ on the teleported qubit. As long as the randomness used to choose $\ell = (\ell_x, \ell_z, \ell_{x,0}, \ell_{z,0}, \ell_{x,1})$ and the bits $s = (s_x, s_z), t = (t_x, t_z)$ are kept secret from the evaluator, then this circuit completely hides all information about the teleportation keys $(a, b)$, and thus the teleported qubit is completely randomized (from the evaluator’s point of view).

Now that the teleportation circuits act on a larger number of qubits, we also modify the quantum state $Q$ so that the randomizers $A_w$ also act on the same ancilla qubits. We now describe the evaluation process. Consider an internal wire $w$ with a single-qubit gate $G$ at the end. To evaluate $G$, the evaluator uses teleportation...
labels \((\ell_{x,a}, \ell_{z,b})\) from a teleportation on a predecessor wire, along with the offline encoding \(f_{\text{off}}\) of a function \(f(a, b)\) that computes the correct-then-teleport operation \(O_{\omega} = \text{TP}_{\ell} \cdot R \cdot A_{\omega} \). The evaluator applies \(O_{\omega}\) to the state, which cancels out the randomizer \(A_{\omega}\), applies the correction \(R\), and applies the next teleportation gadget. The evaluator can then measure the ancilla qubits of this new gadget to get the labels for the next teleportation, and so on.

Thus, in addition to the state \(Q\), the encoding of \((F, x)\) also includes the offline encoding \(f_{\text{off}}\) of the functions \(f\) that compute the correct-then-teleport operations. For the sake of this overview we omitted many technical details that arise in the encoding and evaluation; for example the correct-then-teleport operations \(O\) are different for the input and output wires. The functions \(f\) get more complex when it comes to evaluating multi-qubit gates. However this presents the main idea.

We now mention an important point. The correct-then-teleport operations \(O\) are Clifford unitaries; this is because the teleportation gadgets, the corrections \(R\), and the randomizers are all Clifford. This means that first, the unitaries \(O\) have a canonical description that can be efficiently computed \([1, 34]\); thus the functions \(f\) can be computed by (classical) circuits whose size is polynomial in the number of qubits of \(O\). Second, since the randomizers are uniformly over the Clifford group, the \(O\) operations are also uniformly random Cliffrords. This fact is crucial for obtaining the privacy properties of this encoding, which we discuss next.

2.3 Group-Randomizing QRE and Privacy of the Scheme

The scheme we just sketched satisfies functionality (because the evaluator can obtain \(F(x)\) from the encoding); we now explain why it satisfies the privacy property. The privacy properties of our randomized encoding scheme are established via the existence of a simulator, which is an efficient procedure \(\text{Sim}\) that takes as input a quantum state \(F(x)\) for some circuit \(F\) and state \(x\), and produces another quantum state that is indistinguishable from the randomized encoding \(\hat{F}(x)\). This formalizes the idea that the only thing that can be learned from the randomized encoding \(\hat{F}(x)\) is the value \(F(x)\).

The existence of a simulator for our scheme relies on an idea which we call group-randomizing quantum randomized encoding.

Group-Randomizing QRE. We pause to explain group-randomizing QRE in greater generality than is needed for our Quantum Garbled Circuits scheme, because we believe that it is a useful conceptual framework for thinking about applications of randomized QRE and may be useful for other applications of QRE (see, e.g., the work of \([12]\) for an application of group-randomizing QRE to secure multiparty computation).

Let \(G\) denote a finite subgroup of the \(n\)-qubit unitary group (for example, the Clifford group). Let \(C\) denote a circuit computing a unitary in \(G\), and let \(x\) denote a quantum input. The encoding of \(C(x)\) is a pair \(\hat{C}(x) = (A(x), CA^\dagger)\), where \(A\) is a uniformly random element of \(G\), and we assume that \(CA^\dagger\) is described using some canonical circuit representation for elements of \(G\). Clearly, the output \(C(x)\) can be decoded from this encoding: the decoder simply applies \(CA^\dagger\) to \(A(x)\). This encoding is also perfectly private:

for a uniformly random \(A \in G\), the unitary \(CA^\dagger\) is also uniformly distributed in \(G\). Thus, a simulator on input \(y\) can output \((\hat{B}(y), B)\) for a uniformly random element \(B \in G\). When \(y = C(x)\) this yields the same distribution as the encoding \(\hat{C}(x)\). We note that this method is conceptually similar to the well-known classical RE techniques for branching programs using matrix randomization \([40]\).

Intuition for the Privacy Property. We now illustrate how the group randomization idea is used. For simplicity assume that there is only one single-qubit gate \(G\) in the computation acting on a qubit state \(x\). Imagine that the evaluator receives an encoding which consists of

1. Labels \((\ell_{x,a}, \ell_{z,b})\) corresponding to bits \((a, b) \in \{0, 1\}^2\),
2. The offline encoding \(f_{\text{off}}\), and
3. The quantum state \(R = A(E(x), s)\) where \(A\) is a Clifford randomizer, \(E = X^aZ^b\) is a Pauli error, and

\[
\begin{align*}
& s = (e_1, G(R_{u,p}(e_2)), 0 \cdots 0) \\
& \text{with } (e_1, e_2) \text{ being an EPR pair.}
\end{align*}
\]

The function \(f\) computes the map \((a, b) \mapsto O = TP_{\ell^\prime} \cdot E^\dagger \cdot A^\dagger\) where \(TP_{\ell^\prime}\) is an encrypted teleportation gadget.

If the evaluator were honest, then it can use the classical randomized encoding \((f_{\text{off}}, \ell_{x,a}, \ell_{z,b})\) to obtain the (classical description of) \(O\); it then applies \(O\) to the state \(R\), yielding the state \(TP_{\ell^\prime}(x, s)\), with which one can perform gate teleportation to obtain another set of labels \((\ell_{x^\prime,a^\prime}, \ell_{z,b^\prime})\), etc.

We claim that, regardless of whether the evaluator is honest, the evaluator could have simulated the encoding by itself using only the state \(TP_{\ell^\prime}(x, s)\). To do so, it first samples a uniformly random Clifford randomizer \(B\), and applies \(B^\dagger\) to the state, yielding the pair

\[
\left(\hat{B}^\dagger \left( TP_{\ell^\prime}(x, s) \right), B \right).
\]

This is distributed identically to

\[
\left( R, O \right)
\]

because \(O = TP_{\ell^\prime} \cdot E^\dagger \cdot A^\dagger\) is a uniformly random Clifford unitary and \(O(R) = TP_{\ell^\prime}(x, s)\).

We now appeal to the privacy of the classical randomized encoding: the distribution of \((f_{\text{off}}, \ell_{x,a}, \ell_{z,b})\) can be simulated given only the output \(O\); thus by applying the classical simulator to \((2.1)\), we get a distribution that is indistinguishable from

\[
\left( R, \hat{f}_{\text{off}}, \ell_{x,a}, \ell_{z,b} \right)
\]

which is precisely the distribution of the encoding received by the evaluator.

Note that the notion of indistinguishability here is derived from the privacy guarantees of the classical randomized encoding scheme. If the classical randomized encoding is information-theoretically private, so is this quantum randomized encoding. If it classical randomized encoding scheme relies on computational assumptions, then so does the quantum randomized encoding scheme.
Privacy of the Scheme. The argument sketched above can be generalized to a circuit with many gates. We now discuss several further aspects about the privacy properties of our Quantum Garbled Circuits scheme.

An important feature of the randomized encoding $\hat{F}(x)$ is that it hides the specific names of the gates being applied in a circuit $F$, and only reveals the topology of the circuit $F$. This feature automatically implies the privacy of the randomized encoding: this means that it is not possible to distinguish between the encoding of $F$ on input $x$, or the encoding of a circuit $E$ with the same topology as $F$, but with all identity gates, with input $F(x)$. In both cases, the decoding process (which is a public procedure which does not require any secrets) produces the output $F(x)$. Thus, there is a canonical choice of simulator for such a randomized encoding: given input $F(x)$, it computes the randomized encoding $\hat{E}(F(x))$, which is indistinguishable from the randomized encoding $\hat{F}(x)$ via the circuit hiding property.

We now discuss a subtle point. We have described the evaluation procedure as involving measurements: the evaluator is supposed to measure the middle two wires of the teleportation gadgets to obtain the labels needed to evaluate the group-randomizing QRE in a subsequent teleportation. However, there is no guarantee that a malicious evaluator (who is trying to learn extra information from the encoding) will perform these measurements. The encoding of a gate technically gives a superposition over all possible labels of a teleportation gadget, and a malicious evaluator could try to perform some coherent operation to extract information about labels for different teleportation keys simultaneously (which would in turn compromise the privacy of the classical RE).

In our scheme, we in fact define the honest evaluation procedure $\text{Dec}$ (which stands for “decode”) to be unitary. The randomization bits $s_x, s_z, t_x, t_z$ in the teleportation gadgets, which are never revealed to the evaluator, effectively force a measurement of the teleportation gadget labels. In particular, randomizing over the $s_z, t_z$ bits destroys any coherence between the different labels in the superposition, which means that the evaluator (even a malicious one) can only get labels for a single teleportation key per teleportation gadget at a time.

The unitarity of $\text{Dec}$ is used in the privacy analysis in the following way: we show that given the randomized encoding of $F(x)$ and $E(F(x))$, applying a unitary evaluation procedure $\text{Dec}$ to both states yields outputs that are indistinguishable (statistical or computational, depending on the privacy properties of the classical RE scheme). Thus, applying the inverse unitary $\text{Dec}^{-1}$ to the outputs preserves the indistinguishability (up to a loss related to the complexity of $\text{Dec}$), and shows that the encodings of $F(x)$ and $E(F(x))$ are indistinguishable.

2.4 Complexity of the Scheme

We have sketched how the QRE scheme presented above satisfies functionality and privacy. However, it does not have the desired efficiency property of having constant-depth encoding (i.e. the encoding process is highly parallel). This is because, as described, the encoder has to apply the Clifford randomizers. If a Clifford randomizer acts on $k$ qubits, then it requires $\Theta(k^2 / \log k)$ gates in general [1, 34], and therefore the depth of the Clifford circuit must grow with $k$. The number of qubits that a randomizer acts on depends on the length of labels used in the corresponding teleportation gadgets, which is in turn determined by the size of a classical randomized encoding.

If we use a classical RE scheme with information-theoretical privacy, then the label lengths will grow doubly-exponentially with the depth of the quantum circuit $F$ being encoded. The reason for this extravagant blow-up is the following. In the information-theoretic setting, the label lengths of known classical REs scale polynomially with the classical function being encoded. Recall that the classical function outputs the description of a Clifford circuit. Thus, for each layer of the quantum circuit $F$, the associated label lengths in the encoding are going to be roughly quadratic in the label lengths of subsequent layers. This implies that the label lengths for the first layer of the circuit are going to scale doubly-exponentially with the depth. If we use classical REs with computational privacy, then the label lengths are independent of the complexity of the function $f$, and only depend polynomially on the security parameter. In either case, the encoding complexity of the QRE will not be constant depth; the complexity of the randomizers is too large.

Our final Quantum Garbled Circuits scheme achieves constant-depth encoding – specifically, the encoding is computable in the class $\text{QNC}^0$, which is the quantum analogue of the class $\text{NC}^0$ (see Section 3.2.3 for formal definitions of these models). Instead of using randomizers from the general Clifford group, we show that it suffices to use randomizers from a much simpler subgroup of unitaries; this is a class of depth-one circuits composed of a tensor product of single- and two-qubit Clifford gates in fixed locations. See the full version for details.

Finally, we discuss the complexity of the Quantum Garbled Circuits encoding. In the setting of information-theoretic privacy, the size of the encoding grows doubly-exponentially with the depth of the circuit, and in the setting of computational privacy, the size of the encoding is polynomial in the security parameter and the size of the circuit. See Section 4.2 for a precise accounting of the QRE complexity.

We note that there is an interesting gap between the complexity of classical RE and quantum RE in the information-theoretic regime. Information-theoretic classical RE has encoding complexity that scales “only” exponentially with the depth of the computation being encoded, whereas in the quantum case we have a doubly-exponential scaling. Is this gap inherent? We leave this as an open question.

2.5 Another Simple QRE Using Group-Randomizing QRE

We note that it is possible to construct a simpler QRE scheme for circuits using the group-randomizing QRE in different and arguably more straightforward manner. This construction does not have the low complexity property, or the gate-by-gate encoding property as our main construction presented above, but it is simpler and carries conceptual resemblance to classical branching program RE via matrix randomization as the well known Kilian RE [40].

The idea is to use the "magic state" representation of quantum circuits [18]. At a high level and using the terminology of this work,
[18] shows that any quantum circuit can be represented (without much loss in size and depth) by a layered circuit as follows. Each layer consists of a unitary Clifford circuit with two types of outputs. Some of the output qubits are passed to the next layer as inputs (hence each layer has fewer inputs than its predecessor), and some of them are measured and the resulting classical string determines the gates that will be applied in the next layer (the last layer of course contains no measured qubits). The inputs to the first layer consist of the input to the original circuit, and in addition some auxiliary qubits, each of which is independently sampled from an efficiently samplable distribution over single-qubit quantum states (called “magic state”).

Given our methodology above, one can straightforwardly come up with a QRE for such circuits. Given a circuit in the aforementioned layered form, and an input, the encoding is computed as follows.

(1) Generate the required number of magic states and concatenate them with the given input.

(2) If the circuit contains only one layer, i.e., is simply a Clifford circuit, use group-randomizing encoding (there is no need for classical RE in this case).

(3) If the circuit contains more than one layer, consider the last layer (that produces the output), we refer to the layer before last as the “predecessor layer”.

(a) Generate (classical) randomness that will allow to apply group-randomizing QRE on the last layer (including decomposable classical RE of the classical part of the encoding). This includes a randomizing Clifford $R$ and randomness for classical RE.

(b) Modify the description of the predecessor layer so that instead of outputting its designated output, it essentially outputs the QRE of the last layer. Specifically, modify the predecessor layer as follows. For the outputs that are passed to the next layer, add an application of $R$ before the values are actually output (since $R$ is Clifford, the layer remains a Clifford layer).

For the outputs that are to be measured, add a $Z$-twirl (i.e. $Z^s$ for a random $s$) followed by a Clifford circuit that selects between the two labels of the classical RE of the following (i.e. last) layer description. Also always output the (fixed classical) offline part of the classical RE.

This transformation maintains the invariant that the new last layer (which is the augmented predecessor layer) is a Clifford circuit where the identity of the gates is determined by a classical value that comes from predecessor layers.

(c) Remove the last layer from the circuit and continue recursively.

Correctness and security follow from those of the group randomizing QRE and the $Z$-twirl. While this QRE is not natively decomposable, it can be made decomposable by adding a single layer of teleportation-based encoding (as in our full-fledged scheme) at the input. Interestingly, the only quantum operation required in this QRE is an application of a random Clifford on the input (more accurately, extended input containing the actual input and a number of auxiliary qubits in a given fixed state).

Carefully keeping track of the lengths of the labels of the classical RE would imply again that for perfect security we may incur up to a double-exponential blowup of the label length as a function of the depth. In the computational setting, the blowup is only polynomial.

In terms of efficiency of encoding, we tried to present the scheme in a way that would make it easiest to verify its correctness and security, but an efficiency-oriented description would allow to encode all layers in parallel. This is because the modification to each layer only depends on the randomness of the QRE of the next layer, which can be sampled ahead of time.

3 PRELIMINARIES

3.1 Notation

Registers. A register is a named Hilbert space $\mathbb{C}^d$ for some dimension $d$. We denote registers using sans-serif font such as $a, b, c,$ etc. Let $w_1, w_2, \ldots, w_n$ be a collection of registers. For a subset $S = \{i_1, \ldots, i_k\} \subseteq [n]$, we write $w_S$ to denote the union of registers $\bigcup_{i \in S} w_i$. We write $\dim(a)$ to denote the dimension of register $a$.

Density Matrices. We use bolded variables to denote density matrices. For example, we write $a$ to denote a density matrix on register $a$. When referring to a collection $(a, b, c)$ of density matrices simultaneously, we are referring to a joint state on the registers $abc$, and when we refer to (say) $a$ alone, we are referring to the reduced density matrix of that state on register $a$. In general, the state $(a, b, c)$ may be highly entangled across its registers. If we want to emphasize that some parts of the state are unentangled with others, we will write (for example) $a \otimes (b, c)$. This indicates a state that is unentangled between registers $a$ and $bc$.

Quantum Operations. Given a quantum operation (a completely positive, trace-preserving map) $F$ mapping register $a$ to register $a'$ and a collection of density matrices $(a, b)$, we write $(F(a), b)$ to denote the density matrix on registers $a'b$ that is the result of applying $F \otimes I$ to the density matrix $(a, b)$. Given quantum operations $F$ and $G$ that act on disjoint qubits, we write $(F, G)$ to denote the product operation $F \otimes G$. For a unitary $U$, we also write $U(x)$ as shorthand for $U x U^\dagger$.

Quantum Circuits and Their Descriptions. Throughout this paper we will talk about quantum circuits both as quantum operations (e.g., a unitary), and as classical descriptions of a sequence of gates. Formally, one is an algebraic object and the other is a classical string (using some reasonable encoding format for quantum circuits). To distinguish between the two presentations we write $\text{Ckt}$ to denote a classical description of a circuit (i.e., a sequence of gates on some
number of qubits), and use sans-serif font such as Ckt to denote the corresponding unitary.

EPR Pair. We let \([EPR]\) denote the maximally entangled state on two qubits, i.e., \(
\frac{1}{\sqrt{2}} \left( |00\rangle + |11\rangle \right)\).

Distinguishability of Quantum States. We say that two quantum states \(a, b\) on the same number of qubits are \((t, \epsilon)\)-indistinguishable if for all auxiliary quantum states \(q\) that are unentangled with either \(a\) or \(b\), for all quantum circuits \(D\) of size \(t\),
\[
\left| P[D(a \otimes q) = 1] - P[D(b \otimes q) = 1] \right| \leq \epsilon.
\]

We often write \(a \approx_{(t, \epsilon)} b\) to denote this. This notion of indistinguishability satisfies the triangle inequality: if \(a, b, c\) are quantum states such that \(a \approx_{(t, \epsilon_1)} b\) and \(b \approx_{(t, \epsilon_2)} c\), then we have \(a \approx_{(t, \epsilon_1 + \epsilon_2)} c\).

Furthermore, if \(a \approx_{(t, \epsilon)} b\) and \(U\) is a quantum circuit of size \(s\), then \(U(a) \approx_{(t+s, \epsilon)} U(b)\) and \(U(a) \approx_{(t-s, \epsilon)} U(b)\). This is because if there was a circuit \(D\) of size \(t-s\) that could distinguish between the two with advantage more than \(\epsilon\), then there exists a circuit \(D' = D \circ U\) of size \(t\) that could distinguish between \(a\) and \(b\) with advantage more than \(\epsilon\).

3.2 Quantum Gates and Circuits

3.2.1 Universal Gate Set. A universal set of gates is \(C_2 \cup \{T\}\), i.e., the set of two-qubit Clifford gates along with
\[
T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}.
\]
The \(T\) gate is an example of a unitary from the third level of the Clifford hierarchy, as formalized in the following fact.

FACT 1. For all \(a, b \in \{0, 1\}\) it holds that \(TX^a Z^b = P^{a} X^a Z^b T\).

3.2.2 Classical Circuits. Here we briefly review classical circuits.

A classical circuit topology \(T\) consists of a directed acyclic graph (DAG) where the nodes are divided into input terminals, placeholder gates, and output terminals. Input terminals have in-degree 0 and arbitrary out-degree (i.e., they are source nodes). Output terminals have in-degree 1 and out-degree 0 (i.e., they are sink nodes). Placeholder gates have constant in-degree and out-degree (without loss of generality this constant can be 2 while incurring only a constant blowup in size and depth compared to any other constant). A classical circuit \(C\) with topology \(T\) is simply an assignment of boolean functionalities to the placeholder gates.

The depth of a circuit is simply the length of the longest path from an input terminal to an output terminal. The size of a circuit is the number of wires (i.e. edges) in the circuit topology.

An important class of circuits are constant-depth circuits. These are captured by the complexity class \(\NC^0\), which technically consists of function families \((f_n)\) that can be computed by a family of polynomial-size circuits whose depth is bounded by a constant (i.e. does not grow with \(n\)). As we shall see in Section 3.3, the class \(\NC^0\) captures the complexity of encoding in classical randomized encoding schemes.

3.2.3 Quantum Circuits and Their Topology. A quantum circuit topology \(\mathcal{T}\) is a tuple that represents the circuit structure as a directed acyclic graph, the gate interconnections interconnections (but without the actual gate identities, i.e. without telling which quantum operator each gate performs). See full version for a formal definition.

Quantum Circuits. A general quantum circuit \(F\) is a pair \((\mathcal{T}, \mathcal{G})\) where \(\mathcal{T}\) is a circuit topology and \(\mathcal{G}\) represents the actual functionality of the gates.

By the Stinespring dilation theorem, every quantum operation can be realized as a general quantum circuit. We associate the complexity of a quantum operation with the size of the quantum circuit that implements it, with respect to a universal set of gates.

For this work, we specify a universal set of gates (specifically, we choose to work with the universal set \(C_2 \cup \{T\}\) as described in Section 3.2.1).

Constant-Depth Quantum Circuits. The main model of constant-depth quantum circuits that we consider in this paper are \(\QNC^0\) circuits, which are constant-depth circuits consisting of one- or two-qubit gates, as well as fan-out gates of arbitrary arity, which copy a control qubit to a number of target qubits (i.e., a fan-out gate with fan-out \(k\) performs the following transformation: \(|x, y_1, \ldots, y_k\rangle \mapsto |x, y_1 \otimes y_2 \otimes \ldots, y_k \otimes x\rangle\)). This is a natural analogue of classical \(\NC^0\) circuits, yet is surprisingly powerful: functions such as \textsc{parity} can be computed in this model \([36, 43]\). We will show that \(\QNC^0\) captures the complexity of the encoding procedures of our quantum randomized encoding scheme.

3.3 Classical Randomized Encoding

We define classical randomized encoding schemes and their properties. See the survey by Applebaum \([8]\) for details and references.

\textbf{Definition 3.1 (Classical Randomized Encoding).} Let \(f \in \{0, 1\}^n \rightarrow \{0, 1\}^m\) be some function. The function \(\hat{f} : \{0, 1\}^n \times \{0, 1\}^t \rightarrow \{0, 1\}^m\) is a \((t, \epsilon)\)-private classical randomized encoding (CRE) of \(f\) if there exist a deterministic function \(\text{CDec}\) (called a decoder) and a randomized function \(\text{CSim}\) (called a simulator) with the following properties.

- \textbf{Correctness.} For all \(x, r\) it holds that \(f(x) = \text{CDec}(\hat{f}(x; r))\).\(^{12}\)
- \textbf{\((t, \epsilon)\)-Privacy.} For all \(x\) and for all circuits \(D\) of size \(t\) it holds that\(^{12}\)
\[
\left| P[D(\hat{f}(x; r)) = 1] - P[D(\text{CSim}(f(x)) = 1) \right| \leq \epsilon
\]

where the second probability is over the randomness of the simulator \(\text{CSim}\). The case of \(\epsilon = 0\) is called perfect privacy.

The encoding \(\hat{f}\) is a decomposable CRE (DCRE) of \(f\) if there exist functions \(\hat{f}_{\text{off}}(r)\) (called the offline part of the encoding) and \(\text{lab}_{i, n}(r)\) (called the label functions) for all \(i \in [n], b \in \{0, 1\}\), such that for all \((x, r)\),
\[
\hat{f}(x; r) = \left( \hat{f}_{\text{off}}(r), (\text{lab}_{i, n}(r))_{i \in [n]} \right).
\]

\(^{12}\)This is known as \textit{perfect correctness} and is the only notion of correctness considered in this work.
Remark 1. We refer to the second input $r$ of $\hat{f}$ as the randomness of the encoding, and we use a semicolon to distinguish it from the deterministic input $x$. We will sometimes write $\hat{f}(x)$ to denote the random variable $f(x; r)$ induced by sampling $r$ from the uniform distribution. Furthermore, we say that the value $\hat{f}(x)$ is the randomized encoding of a function $f$ and a deterministic input $x$.

Note that as presented in Definition 3.1, there is no requirement that the randomized encoding $\hat{f}$ can be efficiently computed from the original function $f$. Furthermore, the decoder $\text{CDec}$ and simulator $\text{CSim}$ are technically allowed to depend arbitrarily on the function $f$ be encoded. However, it is a highly desirable feature that randomized encodings be efficiently computable given a description of $f$, and also have a universality property (see, e.g., Section 7.6.2 of [7]), where the encoding $\hat{f}(x)$ hides information not just about the input $x$, but also about the function $f$. This is formalized by requiring that the decoding and simulation procedures depend only partially on $f$. In many cases, including in this work, they should only depend on the topology of the circuit computing $f$.

4 QUANTUM RANDOMIZED ENCODING – DEFINITION AND EXISTENCE

4.1 Definition

We propose the following quantum analogue of randomized encoding.

Definition 4.1 (Quantum Randomized Encoding). Let $F(x)$ be a quantum operation that maps $n$ qubits to $m$ qubits. The quantum operation $\hat{F}(x;r)$ where $r$ is classical randomness is a $(t, e)$-private quantum randomized encoding (QRE) of $F$ if there exist quantum operations $\text{Dec}$ (called the decoder) and $\text{Sim}$ (called the simulator) with the following properties.

- **Correctness.** For all quantum states $(x, q)$ and all randomness $r$, it holds that $(\text{Dec}(\hat{F}(x;r), q) = (\hat{F}(x), q)$.
- **$(t, e)$-Privacy.** For all quantum states $(x, q)$, we have
  \[
  \left(\hat{F}(x;r), q\right) \approx_{(t, e)} \left(\text{Sim}(F(x)), q\right)
  \]
  where the state on the left-hand side is averaged over $r$. The case of $e = 0$ is called perfect privacy.

The encoding $\hat{F}$ is a decomposable QRE (DQRE) if there exists a quantum state $\epsilon$ (called the resource state of the encoding), an operation $\hat{F}_{\text{off}}$ (called the offline part of the encoding) and a collection of input encoding operations $\hat{F}_1, \ldots, \hat{F}_n$ such that for all inputs $x = (x_1, \ldots, x_n)$,
\[
\hat{F}(x;r) = (\hat{F}_{\text{off}}, \hat{F}_1, \ldots, \hat{F}_n)(x, r, \epsilon)
\]
where the functions $\hat{F}_{\text{off}}, \hat{F}_1, \ldots, \hat{F}_n$ act on disjoint subsets of qubits from $\epsilon, x$ (but can depend on all bits of $r$), each $\hat{F}_i$ acts on the $i$th qubit of the input $x$, and $\hat{F}_{\text{off}}$ does not act on any of the qubits of $x$.

Similarly to the classical case, we refer to the second input $r$ of $\hat{F}$ as the randomness of the encoding. We will often write $\hat{F}(x)$ to denote the quantum state $\hat{F}(x;r)$ when $r$ is sampled from the uniform distribution. Furthermore, we say that the *quantum state* $\hat{F}(x)$ is the randomized encoding of the operation $F$ and an input $x$.

One can see that this definition of quantum randomized encoding is syntactically similar to Definition 3.1, with a couple differences. First, the correctness and privacy properties involve the pair $(x, q)$. We refer the reader to Section 3.1 for the full explanation of the quantum random variable notion; but in short $(x, q)$ represents a bipartite density matrix with an $x$ part, and a $q$ part, and these parts may be entangled. The $q$ part is never acted upon by the decoder or simulator, but distinguishability is measured with respect to the encoding of $x$ as well as $q$, which we think of as quantum side information. In other words, correlations between the input and an external system are preserved through the encoding, decoding, and simulation.

A second difference involves the definition of decomposable QRE. In addition to receiving a random string $r$, the randomized encoding also receives an auxiliary quantum state $\epsilon$ (that is independent of the input $x$). The definition allows for any resource state $\epsilon$, but in this paper we focus on decomposable QREs where the resource state $\epsilon$ is a low-complexity state, such as a collection of EPR pairs.

Remark 2. We emphasize that the resource state $\epsilon$ is generated by the encoder, and is not preshared between the encoder and decoder.

Furthermore, similar to the classical setting, it is highly desirable to have efficient QRE schemes that are universal with respect to some property of the quantum operations being encoded, say the topology of some circuit implementation of them. This motivates the following definition of universal QRE scheme, in analogy to the classical setting (see full version for details).

Definition 4.2 (Universal QRE Schemes for Circuits). Let $C$ denote a class of general quantum circuits\(^\text{13}\) and let $\mathcal{R}$ denote an equivalence relation over $C$. An $(t, e)$-private and efficient $\mathcal{R}$-universal QRE scheme for the class $C$ is a tuple of polynomial-time quantum algorithms $(\text{Enc}, \text{Dec}, \text{Sim})$ such that given a circuit $F \in C$ (here we identify the circuit with the function it computes),

- **Efficient Encoding.** For all quantum inputs $x$ and randomness $r$, $\text{Enc}(F, x; r)$ computes a quantum randomized encoding $\hat{F}(x;r)$.
- **Correctness.** For all quantum states $(x, q)$ and randomness $r$ it holds that $(\hat{F}(x), q) = (\text{Dec}(c, \hat{F}(x;r)), q)$ where $c$ denotes the equivalence class of $F$ in $\mathcal{R}$.
- **$(t, e)$-Privacy.** For all quantum states $(x, q)$, we have
  \[
  \left(\hat{F}(x;r), q\right) \approx_{(t, e)} \left(\text{Sim}(c, F(x)), q\right)
  \]
  where the state on the left-hand side is averaged over the randomness $r$, and $c$ denotes the equivalence class of $F$ in $\mathcal{R}$.

Furthermore, we say that the $\mathcal{R}$-universal QRE scheme $(\text{Enc}, \text{Dec}, \text{Sim})$ is decomposable if the randomized encoding $\hat{F}(x)$ is decomposable and if the input encoding operations $\hat{F}_i$ only depend on the equivalence class of $F$ in $\mathcal{R}$.

For the remainder of this paper, when we speak of encoding a quantum operation $F$, we are referring to encoding a specific circuit implementation of $F$. Furthermore, in this paper we will be focused on topologically-universal QRE schemes – in other words, the equivalence relation $\mathcal{R}$ is such that two circuits $F, F'$ are equivalent\(^\text{13}\).

\(^\text{13}\)See Section 3.2.3 for the definition of general quantum circuits.
if they have the same topology (see Section 3.2.3 for the definition of quantum circuit topology).

4.2 Our Main Result: Existence of Decomposable Quantum Randomized Encodings

Our main result is an efficient topologically-universal decomposable QRE scheme, which we call Quantum Garbled Circuits. We use classical decomposable RE as a building block.

**Lemma 4.3 (Quantum Garbled Circuits Scheme).** Let CRE be an efficient topologically-universal and label-universal classical CRE scheme such that for classical circuits $f$ of size $s$ and depth $d$, the time complexity of encoding $f$ is $c(d, s)$ and the length of the labels is $k(d, s)$. Furthermore, suppose that the encoding of CRE can be computed by NC$^0$ circuits, and that the scheme is $(t, \epsilon)$-private with respect to quantum adversaries.

Recursively define $k_0 = O(1)$, $k_1 = k(O(1), O(k_0^2))$, and define $c_i = c(O(1), O(k_{i-1}^2))$. (All the $O(\cdot)$’s refer to universal constants.)

Then there exists an efficient topologically-universal decomposable CRE scheme $\text{QRE} = (\text{Enc}, \text{Dec}, \text{Sim})$ that satisfies the following properties:

- **Efficiency.** For every operation $F$ computable by a size-$s$ and depth-$d$ quantum circuit and for every quantum input $x$, the encoding $\text{Enc}(F, x; r)$ is computable by a QNC$^0$ circuit of size $O(c_d \cdot s)$. The QNC$^0$ encoding circuit takes as input a string of random bits $r$, the quantum input $x$, and a collection of EPR pairs. Furthermore, the input encoding operations $\hat{F}_i$ can be computed by QNC$^0$ circuits of size $O(k_d)$. The running time of $\text{Dec}$ and $\text{Sim}$ is $O(c_d \cdot s)$.

- **Classical Inputs.** If an input qubit $x_i$ is classical, then $\text{Enc}(F, x; r)$ is computable by a classical circuit.

- **Privacy.** The scheme CRE is $(t', \epsilon')$-private where $t' = t - \text{poly}(c_d) \cdot s$ and $\epsilon' = \epsilon \cdot s$. Here, $s$ and $d$ refer to the size and depth of the circuit being encoded.

**Remark 3.** We can apply our encoding scheme to a universal circuit rather than to $F$ itself, and consider the classical description of $F$ as an additional input. This would incur some overhead due to the use of the universal circuit but will have properties that may be useful in some settings. In particular, the dependence of the encoding on the description of $F$ becomes very simple since the description is classical, the encoding of the input $F$ also becomes classical. Furthermore, if the input $x$ is classical as well, then the quantum part of the encoding $\hat{F}(x)$ is independent of both $F$ and $x$ and can be generated beforehand as a "resource state" that is given to the encoder.

The proof of Lemma 4.3 is presented in the full version. By instantiating CRE in Lemma 4.3 with existing classical RE schemes, the following theorems immediately follow (see full version for details).

**Theorem 4.4 (Information Theoretic DQRE).** There exists an efficient topologically-universal decomposable CRE scheme $\text{QRE} = (\text{Enc}, \text{Dec}, \text{Sim})$ with the following properties:

- **Efficiency.** For every operation $F$ computable by a size-$s$ and depth-$d$ quantum circuit and for every quantum input $x$, the encoding $\text{Enc}(F, x; r)$ is computable by a QNC$^0$ circuit of size $\text{poly}(2^{sd}) \cdot s$. The QNC$^0$ encoding circuit takes as input a string of random bits $r$, the quantum input $x$, and a collection of EPR pairs. Furthermore, the input encoding operations $\hat{F}_i$ can be computed by QNC$^0$ circuits of size $\text{poly}(2^{sd})$. The running time of $\text{Dec}$ and $\text{Sim}$ is $\text{poly}(2^{sd}) \cdot s$.

- **Classical Inputs.** If an input qubit $x_i$ is classical, then $\text{Enc}(F, x; r)$ is computable by a classical circuit.

- **Perfect Information-Theoretic Privacy.** The scheme has perfect privacy against the class of all distinguishers.

**Theorem 4.5 (Computational DQRE).** Assume there exists a length doubling pseudorandom generator (PRG) G that is secure against polynomial-time quantum adversaries. There exists an efficient topologically-universal decomposable CRE scheme $\text{QRE} = (\text{Enc}, \text{Dec}, \text{Sim})$, which implicitly depends on a security parameter $\lambda$, and has the following properties:

- **Efficiency.** For every operation $F$ computable by a size-$s$ and depth-$d$ quantum circuit and for every quantum input $x$, the encoding $\text{Enc}(F, x; r)$ is computable by a QNC$^0$ circuit of size $\text{poly}(\lambda) \cdot s$. The QNC$^0$ encoding circuit takes as input a string of random bits $r$, the output $G(r)$ of the PRG, the quantum input $x$, and a collection of EPR pairs. Furthermore, the input encoding operations $\hat{F}_i$ can be computed by QNC$^0$ circuits of size $\text{poly}(\lambda)$. The running time of $\text{Dec}$ and $\text{Sim}$ is $\text{poly}(\lambda) \cdot s$.

- **Classical Inputs.** If an input qubit $x_i$ is classical, then $\text{Enc}(F, x; r)$ is computable by a classical circuit.

- **Computational Privacy.** For every polynomial $t(\lambda)$, there exists a negligible function $\epsilon(\lambda)$ such that the scheme is $(t'(\lambda), \epsilon'(\lambda, s))$-private with respect to quantum adversaries, where $t'(\lambda, s) = t(\lambda) - \text{poly}(\lambda, s)$ and $\epsilon'(\lambda, s) = \epsilon(\lambda) \cdot s$ with $s$ being the size of the circuit being encoded.

**ACKNOWLEDGMENTS**

We thank Sanjam Garg and Vinod Vaikuntanathan for insightful discussions, and Tomoyuki Morimae and Nico Döttling for valuable feedback on our work. We thank Chinmay Nirkhe for lengthy discussions about quantum garbled circuits. We thank Nir Bitansky and Omri Shmueli for discussions on quantum zero-knowledge. We thank anonymous conference reviewers for their helpful feedback. We thank the Simons Institute for the Theory of Computing – much of this work was performed while the authors were visiting the Institute as a part of the Summer Cluster on Quantum Computing (2018) and the Semester Programs on Quantum Computing and Lattice Cryptography (2020).

A COMPARISON WITH RELATED CRYPTOGRAPHIC NOTIONS

We now compare quantum randomized encoding with other cryptographic primitives of a similar nature. While we focus on the quantum variants of these primitives, the distinctions are the same as in the classical versions.

---

14 As in the classical case, the PRG is used in a black-box manner. If we allow non-black-box use of the PRG and if it can be computed by a $O(\log \lambda)$-depth circuit, then the encoding circuit can be made fully in QNC$^0$ that takes as input only $x$, the randomness $r$, and EPR pairs.
Secure Multiparty Computation. The goal of secure multiparty computation (MPC) is to allow a number of parties, each with their own private input $x_i$, to jointly compute a function $f(x_1, \ldots, x_n)$ such that no party can learn about others’ inputs. There is a close connection between MPC and REs, in that REs can often be used to accomplish secure MPC. Indeed, Yao’s garbled circuits scheme from the 80’s was presented as a technique for achieving secure 2-party computation, and its distillation into a separate primitive with concrete properties occurred later. Many protocols for secure quantum MPC have been constructed over the years (see, e.g., [14, 23, 25, 26, 51]).

While RE is useful for constructing MPC (and sometimes the other way around), and while the security in both notions is defined using the simulation paradigm, inherently they are very different. While MPC is a communication protocol between multiple parties with inputs, RE is not a protocol and it only considers a single input (one could imagine RE as a single-message protocol where an encoder with an input sends a message to a decoder without an input). Since in the context of RE there is only a single party, there is always a trivial solution of computing the function locally. Therefore usually in RE we are concerned with other properties of the construction beyond its security, such as the complexity of encoding.

Homomorphic Encryption. Fully homomorphic encryption (FHE) is a method to compute functions on inputs that are encrypted, without having to ever decrypt the information. FHE and RE also share some commonalities, and there are contexts where both techniques are used to accomplish secure computation and delegation of computation. However there are intrinsic differences between these two concepts. (See also [8, Remark 1.4].)

In FHE, a client encrypts an input $x$ and sends $\text{Enc}(x)$ to a remote evaluator. The evaluator can then compute $\text{Enc}(f(x))$ for any function $f$ – from the given ciphertext, without learning anything about $x$ or $f(x)$. It sends $\text{Enc}(f(x))$ back to the client, who can use its secret decryption key to recover $f(x)$. It is important that the evaluator does not have access to the decryption key, because otherwise it will be able to learn the original input $x$.

With REs, the client sends an encoding of both a function $f$ and an input $x$, from which the evaluator can compute the value $f(x)$ in the clear. The evaluator doesn’t have to send any messages back to the client, and furthermore the evaluator cannot derive an encoding of $g(x)$ for some unrelated function $g$.

If all we require is decomposable RE, then one can achieve this under minimal assumptions such as the existence of one-way functions (or even unconditionally in some cases, depending on the desired complexity properties). FHE (and homomorphic encryption in general) is only known under stronger assumptions (and cannot be achieved with unconditional security). In particular, candidates for classical and quantum FHE often rely on the hardness of the learning with errors problem (or related problems). Quantum FHE was considered recently in [17, 21, 24, 42].

Program Obfuscation. In program obfuscation, an obfuscator encodes a function $f$ into an obfuscated program $\text{Enc}(f)$, which is sent to an evaluator. Using the obfuscated program, the evaluator can compute $f(x)$ for any choice of input $x$. The security requirement, intuitively, is that nothing about $f$ is revealed beyond its input-output functionality. The most commonly studied notions of obfuscation are virtual black-box (VBB) obfuscation and indistinguishability obfuscation (iO), which differ in how they hide the function $f$.

While an obfuscated program can be evaluated on multiple inputs (as m mant as the user wishes), in RE the encoding fixes both the function and an input, so it can be thought of as a “one-time obfuscation”. As explained in [8, Section 4.4], the obfuscation of the program which has $x$ hardwired and evaluates $f$ on it, constitutes a RE of $f(x)$. On the other hand, REs can be used to “bootstrap” obfuscators to have superior complexity properties.

There has been a formalization of quantum program obfuscation [3], but it is not yet known whether general quantum obfuscation can be achieved assuming only classical obfuscation. Broadbent and Kazmi have recently showed how to achieve indistinguishability obfuscation for quantum circuits with low $T$-gate count (at most logarithmically many) [22]. As we mention in Appendix B, a classical RE scheme for quantum circuits can be combined with a classical obfuscator to imply a quantum obfuscator.

B ON APPLICATIONS OF QRE

B.1 Two-Party Secure Computation

Applications to MPC in the context of round reduction also seem to follow. In particular, one can use our construction to obtain an analogue of Yao’s original 2-message two-party MPC protocol using (classical) oblivious transfer (OT) as a building block. We recall that in OT, we have a receiver with a bit $b$, and a sender with two strings $r_0, r_1$, and in the end of the execution the receiver learns $r_b$ and the sender learns nothing about $b$.

We can consider two parties $A, B$, each of which holding a quantum input, $x, y$ respectively, and they wish to jointly compute a quantum operation $F$ on their inputs whose output is delivered to $A$. This can be done as follows. Party $A$ encrypts its input with a classical key using a quantum one-time pad ($\text{QOTP}$, [4]), it sends the encrypted input to $B$ and conducts an OT protocol as a receiver, for each bit of the classical pad for the QOTP. Party $B$ considers a quantum functionality $F'$ that takes as input an encrypted $x$, (unencrypted) $y$, and classical QOTP key. On this input, $F'$ first decrypts $x$ and then applies the original $F$ on $x, y$. Party $B$ creates a decomposable QRE of $F'$, plugging in the encrypted $x$ that was received from $A$, and its own unencrypted input $y$. We recall that for classical input bits, our QRE is classical and decomposable, which means that for each classical input bit we can generate two labels $r_0, r_1$, such that if the value of the bit is $b$ then the part of the encoding that depends on this bit is $r_b$. This means that party $B$ can send the parts of the encoding that it can compute, and complete the OT protocol as a sender with values $r_0, r_1$ for each bit of classical input. This will allow party $A$ to obtain the encoding of $F$, apply the decoding procedure and learn the intended output.

It appears that one should be able to prove security of such a protocol in the specious model [25], which is the mildest model of security in the quantum setting, if the underlying classical OT

\[15\text{Interestingly, contrary to the classical setting, this does not seem to immediately imply a protocol for the setting where both parties receive an output with the same round complexity (if we consider a general quantum operation). One additional round message seems to be needed in this case.}\]
primitive is also spescious secure. However, a formal proof is
tangent to the scope of this work and one should only treat this
as a candidate until a formal proof is presented. We also note that
protocols with comparable round complexity can be achieved using
quantum fully homomorphic encryption [17, 21, 24, 42].

B.2 Functional Encryption
It was noticed in [50] that decomposable RE schemes imply a lim-
ited form of functional encryption (FE). This is an encryption scheme
where there are multiple secret keys, associated with functions, so
that when skf decrypts Enc(x) the output is f(x). In the classical
setting RE implies FE without “collusion resistance” (i.e. an adver-
sary should not be allowed to obtain more than a single functional
key). It was then showed [33] how to extend this technique to FE
with “bounded collusion”. This construction seem to carry over to
the quantum setting using our QRE scheme, under the appropriate
definition of FE. However, some definitional work is required in or-
der to formally substantiate the definition and show the connection
in the quantum setting.

A more ambitious goal is to construct succinct FE schemes (even
with bounded collusion) and so-called “reusable” garbled circuits
which are function-private symmetric-key FE, analogously to the
classical constructions in [32] (but possibly using different tech-
nique). One obstacle that seems to prevent direct application is the
absence of a quantum attribute-based encryption schemes that are
a central building block in that construction.

B.3 Classical Garbling for Quantum Circuits
There are some barriers towards achieving classical garbling. The
first is that purely classical encodings of quantum randomized en-
codings cannot exist unless the output of the decoding is classical
(rather than being a general quantum state as allowed by our defi-
nition). This was shown in a recent paper by Morimae [44], who
provided a very simple and elegant argument of this; essentially, he
showed that otherwise the classical encoding could be used to clone
a set of non-orthogonal states. Even if we modify the definition of
QRE to restrict to the case of classical outputs, there is a complexity-
theoretic constraint on this possibility: Applebaum showed that
any language decidable by circuits that admit efficient RE with
information-theoretic security falls into the class \( \text{SZK} \subseteq \text{PH} \) [7].
Therefore, if we could achieve QRE with statistical security for
polynomial size quantum circuits, this would imply \( \text{BQP} \subseteq \text{SZK} \).
On the other hand, the oracle separation between BQP and PH by
Raz and Tal [48] suggests that this inclusion is unlikely. As
presented, our Quantum Garbled Circuits scheme achieves statistical
security for all quantum circuits of depth \( O(\log \log n) \), but with
small modifications can handle an interesting subclass of quantum
circuits of depth \( O(\log n) \) that is not obviously classically simulable,
and thus languages computed by this subclass are not obviously
contained in \( \text{SZK} \). This suggests that obtaining entirely classical
encoding of quantum circuits cannot be achieved with statistical
security; whether it can be achieved with computational security
remains an intriguing open problem.

REFERENCES
[1] Scott Aaronson and Daniel Gottesman. 2004. Improved Simulation of Stabilizer
Circuits. CoRR quant-ph/0406196 (2004). http://arxiv.org/abs/quant-ph/0406196
[2] Dorit Aharonov, Michael Ben-Or, Elad Eban, and Umrao Mahadev. 2017. In-
teractive proofs for quantum computations. arXiv preprint arXiv:1704.04487
(2017).
[3] Gorjan Alagic and Bill Fefferman. 2016. On quantum obfuscation. arXiv preprint
arXiv:1602.01771 (2016).
[4] Andreas Ambainis, Michele Mosca, Alain Tapp, and Ronald de Wolf. 2000. Private
Quantum Channels. In 41st Annual Symposium on Foundations of Computer
Science, FOCS 2000, 12-14 November, Redondo Beach, California, USA. 547–
553. https://doi.org/10.1109/SFCS.2000.893142
[5] Benny Applebaum. 2011. Key-Dependent Message Security: Generic Ampli-
fication and Completeness. In Advances in Cryptology - EUROCRYPT 2011 -
30th Annual International Conference on the Theory and Applications of Cryp-
tographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings ( Lecture
Notes in Computer Science, Vol. 6632), Kenneth G. Paterson (Ed.). Springer, 527–546.
https://doi.org/10.1007/978-3-642-20465-4_29
[6] Benny Applebaum. 2014. Bootstrapping Obfuscators via Fast Pseudorandom
Functions. In Advances in Cryptology - ASIACRYPT 2014 - 19th International
Conference on the Theory and Application of Cryptology and Information Security
Kaoxiang, Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II ( Lecture
Notes in Computer Science, Vol. 8874), Palash Sarkar and Tetsu Iwata (Eds.). Springer,
162–172. https://doi.org/10.1007/978-3-662-45608-8_9
[7] Benny Applebaum. 2014. Cryptography in Constant Parallel Time. Springer.
https://doi.org/10.1007/978-3-642-17367-7
[8] Benny Applebaum. 2017. Garbled Circuits as Randomized Encodings of Functions:
A Primer. In Tutorials on the Foundations of Cryptography, Yehuda Lindell (Ed.).
Springer International Publishing, 1–44. https://doi.org/10.1007/978-3-319-57048-8_1
[9] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. 2004. Cryptography in NC0.
In 45th Symposium on Foundations of Computer Science (FOCS 2004), 17-19 October
2004, Rome, Italy. Proceedings. 166–175. https://doi.org/10.1109/FOCS.2004.20
[10] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. 2006. Computationally
Private Randomizing Polynomials and Their Applications. Computational Com-
plexity 15, 2 (2006), 155–162. https://doi.org/10.1007/s00037-006-0021-9
[11] Roaz Barak, Itthach Dolev, and Yuval Ishai. 2018. Bound key-
dependent message security. In Annual International Conference on the Theory and
Applications of Cryptographic Techniques. Springer, 423–444.
[12] James Bartusek, Andrea Coladangelo, Dakshita Khurana, and Fermi Ma. 2021.
On the Round Complexity of Secure Quantum Computation. In Advances in
Cryptography - CRYPTO 2021 - 41st Annual International Cryptology Conference,
CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I ( Lecture
Notes in Computer Science, Vol. 12812), Tal Malkin and Chris Peikert (Eds.). Springer,
406–435. https://doi.org/10.1007/978-3-030-84242-0_15
[13] Donald Beaver, Silvio Micali, and Phillip Rogaway. 1990. The Round Complexity
of Secure Protocols (Extended Abstract). In Proceedings of the 22nd Annual ACM
Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA.
503–513. https://doi.org/10.1145/102216.102287
[14] Michael Ben-Or, Claude Crépeau, Daniel Gottesman, Avinatan Hassidim, and
Adam Smith. 2006. Secure multiparty quantum computation with (only) a strict
honest majority. In 2006 47th Annual IEEE Symposium on Foundations of Computer
Science (FOCS'06). IEEE, 249–268.
[15] Fabrice Benhamouda and Huijia Lin. 2018. \&-Round Multiparty Computation from
\&-Round Oblivious Transfer via Garbled Interactive Circuits, See [45], 500–
532. https://doi.org/10.1007/978-3-319-78375-8_17
[16] Nir Bitansky, Ran Canetti, Jiaming Chen, Justin Holmgren, and Umur Kerecz. 2013.
Indistin-
guizability Oblfuscation for RAM Programs and Succinct Randomized
Encodings. SIAM J Comput. 47, 3 (2018), 1123–1160. https://doi.org/10.1137/15M105063
[17] Zvika Brakerski. 2018. Quantum FHE (Almost) As Secure As Classical. In Advances
in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference,
Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part III ( Lecture
Notes in Computer Science, Vol. 10991), Hao Wang and Alexandre Baly (Eds.).
Springer, 67–95. https://doi.org/10.1007/978-3-319-96878-0_3
[18] Sergey Bravy and Alexei Kitaev. 2005. Universal quantum computation with
ideal Clifford gates and noisy ancillas. Physical Review A. 72, 2 (Feb. 2005), 022316.
https://doi.org/10.1103/PhysRevA.71.022316
[19] Anne Broadbent, Joseph Fitzsimons, and Elham Kashefi. 2009. Universal Blind
Quantum Computation. In 2009 40th Annual Symposium on Foundations of Computer
Science, IEEE, 517–526.
[20] Anne Broadbent and Alex B. Grilo. 2020. QMA-hardness of consistency of local
density matrices with applications to quantum zero-knowledge. In 2020 IEEE 61st
Annual Symposium on Foundations of Computer Science (FOCS). IEEE, 196–205.
816
