Quantum Garbled Circuits

Zvika Brakerski [*] Henry Yuen [†]

Abstract

We present a garbling scheme for quantum circuits, thus achieving a decomposable randomized encoding scheme for quantum computation. Specifically, we show how to compute an encoding of a given quantum circuit and quantum input, from which it is possible to derive the output of the computation and nothing else.

In the classical setting, garbled circuits (and randomized encodings in general) 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. However, a quantum analogue for garbling general circuits was not known prior to this work. We hope that our quantum randomized encoding scheme can similarly be useful for applications in quantum computing and cryptography.

The properties of our scheme are as follows:

- Our scheme has perfect correctness, and has perfect information-theoretic security if we allow the encoding size to blow-up considerably (double-exponentially in the depth of the circuit in the worst-case). This blowup can be avoided via computational assumptions (specifically, the existence of quantum-secure pseudorandom generators). In the computational case, the size of the encoding is proportional to the size of the circuit being garbled, up to a polynomial in the security parameter.
- The encoding process is decomposable: each input qubit can be encoded independently, when given access to classical randomness and EPR pairs.
- The complexity of encoding essentially matches the size of its output and furthermore it can be computed via a constant-depth quantum circuit with bounded-arity gates as well as quantum fan-out gates (which come "for free" in the classical setting). Formally this is captured by the complexity class $\mathbf{QNC}^0_f$.

To illustrate the usefulness of quantum randomized encoding, we use it to design a conceptually-simple zero-knowledge (ZK) proof system for the complexity class **QMA**. Our protocol has the so-called Σ format with a single-bit challenge, and allows the inputs to be delayed to the last round. The only previously-known ZK Σ-protocol for **QMA** is due to Broadbent and Grilo (FOCS 2020), which does not have the aforementioned properties.

[*]Weizmann Institute of Science, zvika.brakerski@weizmann.ac.il. Supported 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).

[†]University of Toronto. hyuen@cs.toronto.edu. Supported by a NSERC Discovery Grant.

Contents

1 Introduction

A *randomized encoding* (RE) of a function f is another function $\hat{f}$, computed probabilistically, such that on every input x, the output $f(x)$ can be recovered from $\hat{f}(x)$, and no other information about f or x is conveyed by $\hat{f}(x)$. A trivial example of a RE of a function f is f itself. Things become much more interesting when computing $\hat{f}(x)$ is simpler in some way than computing $f(x)$; for example, $\hat{f}(x)$ could be computed via a highly parallel process even if evaluating $f(x)$ itself requires a long sequential computation.

REs are central objects in cryptographic research and have proven useful in a multitude of settings: the most famous example of a RE is Yao's garbled circuits construction [Yao86], but it was only until the work of Applebaum, Ishai and Kushilevitz in [AIK04, AIK06] that the formal notion of randomized encodings was presented. Applications of RE range from secure multi-party computation, parallel cryptography, verifiable computation, software protection, functional encryption, key-dependent message security, program obfuscation and more. We refer the readers to an extensive survey by Applebaum [App17] for additional details and references. Interestingly, REs have also proved useful in recent circuit lower bounds [CR20].

A useful feature of many randomized encodings is *decomposability*: this is where a function f and a sequence of inputs $(x_1, \ldots, x_n)$ can be encoded in such a way that $\hat{f}(x_1, \ldots, x_n) = (\hat{f}_{\mathsf{off}}, \hat{f}_1, \ldots, \hat{f}_n)$ where $\hat{f}_{\mathsf{off}}$ (called the "offline" part of the encoding) depends only on f and the randomness r of the encoding, and $\hat{f}_i$ (the "online" part) only depends on x_i and the randomness r. Such randomized encodings are called *decomposable*.

A good illustration of the usefulness of decomposable REs (DREs) is the task of "private simultaneous messages" (PSM) introduced by Feige, Kilian and Naor [FKN94]. In a PSM protocol for computing a function f, a set of n separated players each have an input x_i and send a message m_i to a referee, who then computes the output value $y = f(x_1, \ldots, x_n)$. The messages m_i cannot reveal any information about the x_i's aside from the fact that $f(x_1, \ldots, x_n) = y$ (formally, the messages e_i can be simulated given y). The parties share a common random string r that is independent of their inputs and unknown to the referee, and the goal is to accomplish this task using minimal communication.

Using a DRE such as garbled circuits, the parties can simply send an encoding of the function f and their respective inputs respectively to the referee. In particular, using the point-and-permute garbled circuits scheme of Beaver, Micali and Rogaway [BMR90, Rog91], it is possible to construct DREs with perfect decoding correctness and perfect simulation security for any function f with complexity that scales with the formula size of f. Assuming the adversaries are computationally bounded, it is possible to reduce this complexity to scale polynomially with the circuit size of f. Thus, the PSM task can be performed efficiently using DREs.

In some cases, even non-decomposable REs can be useful. However, some other non-degeneracy condition should be imposed, since (as mentioned before) every function f is trivially a non-decomposable RE of itself. For example, if for some measure of complexity, the complexity of computing the encoding $\hat{f}$ is lower than the complexity of computing f, then this can be leveraged

for blind delegated computation: a verifier who wishes to compute $f(x)$ can first compute the encoding of a function $g(x)$ that outputs a random string r_0 if $f(x) = 0$, and otherwise outputs r_1. The server can evaluate the encoding $\hat{g}(x)$ to obtain either r_0 or r_1, which the verifier decodes to determine $f(x)$. As long as the complexity of encoding $g(x)$ is less than $f(x)$, this yields a non-trivial delegation scheme.

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.

1.1 Quantum Randomized Encodings

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 $\boldsymbol{x}$ is another quantum state $\hat{F}(\boldsymbol{x})$ satisfying two properties:

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

The privacy property is formalized by saying there is a simulator that, given $F(\boldsymbol{x})$, can compute the encoding $\hat{F}(\boldsymbol{x})$. We also refer to $\hat{F}$ as the encoding of F.

Furthermore, we also define what it means for a QRE to be *decomposable*: the encoding $\hat{F}(\boldsymbol{x})$ can be computed in a way that each qubit of the input $\boldsymbol{x}$ is encoded independently, and the encoding takes in as input $\boldsymbol{x}$, a classical random string r, and a sequence of EPR pairs $\boldsymbol{e}$. [1] (See Section 4.1 for a formal definition of (decomposable) QRE.)

For comparison of the notion of QRE with other cryptographic notions such as MPC, FHE and program obfuscation see Appendix A.

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 QRE 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* $\mathbf{NC}^0$*, then the encoding procedure of* QGC *can be computed in* $\mathbf{QNC}^0_f$ *(i.e. the class of constant-depth quantum circuits with unbounded fan-out gates).*

(See Section 4.2 for a formal statement of our main result). We elaborate on the properties of the QRE scheme below. It assumes the existence of a classical DRE scheme CRE with specific correctness, privacy, and complexity properties; examples of such schemes can be found in [BMR90, Rog91] (also see the survey in [App17]). In the case of computational privacy, we assume the existence of quantum-secure one-way functions.

[1] We recall that an EPR pair is the maximally entangled state $\frac{1}{\sqrt{2}}\big(|00\rangle + |11\rangle\big)$, the quantum analogue of a pair of classically correlated bits.

Correctness. The correctness property asserts that from an encoding $\hat{F}(x)$, it is possible to decode the output state $F(x)$ with probability 1. This is inherited from the perfect correctness property of CRE. Furthermore, the decoding procedure preserves quantum correlations with side information: if (x, y) denotes the joint state of the input x and some auxiliary quantum state y which may be *entangled* with x, then the joint state of the output and side information after decoding is $(F(x), y)$.

The decoding procedure also takes polynomial time in the size of encoding. This inherits the polynomial-time decoding complexity of CRE.

Privacy. The privacy property implies that there exists a quantum algorithm Sim such that $\mathsf{Sim}(F(x))$ is indistinguishable from the encoding $\hat{F}(x)$, and furthermore Sim runs in time polynomial in the size of the encoding $\hat{F}(x)$. The quantum scheme inherits its privacy from the classical scheme in a black box way: if CRE is secure against all (quantum) distinguishers of size S, then QGC is secure against distinguishers of size at most $S - \Lambda$ where Λ is the complexity of decoding $\hat{F}(x)$ (which is polynomial in the size of $\hat{F}(x)$). Note that perfect information-theoretic privacy corresponds to privacy against distinguishers of all sizes.

The privacy property also holds even when considering entangled side information: the joint state $(\mathsf{Sim}(F(x)), y)$ (which is computed by applying F, then Sim to the input x) is indistinguishable from $(\hat{F}(x), y)$.

Size of the Encoding. The *size* of the encoding of QGC is the number of qubits in $\hat{F}(x)$, which depends on the circuit size and depth of F, and also on the size of the classical encodings computed in the scheme CRE. For example, the classical DRE schemes from [BMR90, Rog91] with computational privacy have encoding size that scales polynomially with the circuit size of f. This translates to the size of $\hat{F}$ being polynomial in the circuit size of F. On the other hand, the known classical decomposable RE schemes with information-theoretic privacy all have encodings $\hat{f}$ that grow *exponentially* with the circuit *depth* of the function f. Using such a scheme as our CRE, the size of the corresponding quantum encoding $\hat{F}$ in our construction might grow even doubly-exponentially with the circuit depth of the quantum operation F. Note that this is the worst-case, the exact growth depends on the composition of gates in the circuit (see technical overview).[2] However we note that even in this case, the number of EPR pairs used in the encoding $\hat{F}(x)$ remains linear in the circuit size of F.

Complexity and Locality of Encoding. The decomposability of the encoding $\hat{F}(x)$ is analogous to the decomposability property of classical DRE schemes, where the encoding can be expressed as the following concatenation:

$$\hat{F}(x; r, e) = (\hat{F}_{\mathsf{off}}(r, e), \hat{F}_1(x_1; r, e), \ldots, \hat{F}_n(x_n; r, e))$$

where we indicate the dependency of the encoding on randomness string r and a sequence of EPR pairs e. The state $\hat{F}_{\mathsf{off}}(r, e)$ is called the "offline" part of the encoding that depends on F but not x, and $\{\hat{F}_j(x_j; r, e)\}_j$ forms the "online" part of the encoding where x_j is the j-th qubit of the n-qubit state x.

The encoding procedure of QGC is highly parallelizable. Suppose that the encoding procedure of CRE is computable in $\mathbf{NC}^0$ (which is the case for the randomized encoding schemes of [BMR90,

[2] We note that an earlier version of this work claimed that the growth in the information-theoretic setting is only (single) exponential. However we discovered an error in our original proof, and the correct analysis turns out to imply the aforementioned parameters. We discuss the causes for this in our technical overview below.

Rog91]). Then the offline part $\hat{F}_{\mathsf{off}}(r;\boldsymbol{e})$ can be computed by a $\mathbf{QNC}^0_{\mathrm{f}}$ circuit acting on $(r,\boldsymbol{e})$; a $\mathbf{QNC}^0_{\mathrm{f}}$ circuit is a constant-depth circuit composed of single- and two-qubit gates, as well as *fan-out gates* with unbounded arity, which implements the unitary $|x, y_1, \ldots, y_n\rangle \to |x, x \oplus y_1, \ldots, x \oplus y_n\rangle$. Similarly, $\hat{F}_j(\boldsymbol{x}_j; r, \boldsymbol{e})$ can be computed by a $\mathbf{QNC}^0_{\mathrm{f}}$ circuit acting on r, a constant number of qubits of $\boldsymbol{e}$, and the qubit $\boldsymbol{x}_j$ (but does not depend on the gates of F).

We note that in the quantum setting it is not so clear what is the "correct" analogue for the complexity class $\mathbf{NC}^0$, which is the class of functions that can be computed in constant-depth, or equivalently, functions where each output bit depends only on a constant number of input bits. This implicitly assumes that input bits can be replicated an arbitrary number of times (for example, all of the output bits may depend on one input bit). However, due to the No-Cloning Theorem we cannot assume that input qubits can be copied, and thus it seems reasonable to consider constant-depth quantum circuits augmented with fan-out gates. However, the fan-out gate does appear to yield unexpected power in the quantum setting: for example, the parity gate can be computed in $\mathbf{QNC}^0_{\mathrm{f}}$, while classically it is even outside $\mathbf{AC}^0$ (see e.g. [Moo99, HS05]). Nevertheless, $\mathbf{QNC}^0_{\mathrm{f}}$ circuits appear to be weaker than general polynomial-size quantum circuits and it may be reasonable to assume that it will be possible to implement the fan-out gate in "constant depth" in some quantum computing architectures (see [HS05] for discussion).

Classical Encoding for Classical Inputs. A desirable property that comes up in the quantum setting is to allow some of the parties to remain classical, even when performing a quantum task. In the RE setting, we would like to allow parties with a classical inputs to compute their encoding in a classical manner (and in particular with access only to the classical part of the randomness/EPR string). Our scheme indeed allows this type of functionality, and therefore allows applications such as quantum PSM (as discussed above) even when some of the parties are classical. Nevertheless, the encoding (and in particular the offline part that depends on the circuit) requires quantum computation.

Could the encoding of a quantum circuit and classical input be made entirely classical? As we will discuss later, this could be used to achieve general indistinguishability obfuscation for quantum circuits. However, there are certain complexity-theoretic constraints on this possibility: Applebaum showed that any language decidable by circuits that admit efficient RE with information-theoretic security falls into the class $\mathbf{SZK} \subseteq \mathbf{PH}$ [App14b]. Therefore, if we could achieve QRE with statistical security for polynomial size quantum circuits, this would imply $\mathbf{BQP} \subseteq \mathbf{SZK}$. On the other hand, the oracle separation between $\mathbf{BQP}$ and $\mathbf{PH}$ by Raz and Tal [RT19] suggests that this inclusion is unlikely.[3] 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 $\mathbf{SZK}$.[4] 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.

[3] We thank Vinod Vaikuntanathan for pointing this out to us.

[4] This subclass includes circuits such that the first $O(\log\log n)$ layers can have arbitrary 2-qubit gates, and the remaining layers are all Clifford gates.

1.2 Other Related Work

We mention some related work on adapting the notion of randomized encodings/garbled circuits to the quantum setting. In [KW17], 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 [Zha20] (which builds on prior work [Zha19]), 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 [KW17] and [Zha20] 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).

Finally, we mention that while our notion of quantum randomized encoding has many similarities with other commonly studied cryptographic notions such as secure multiparty computation (MPC) and homomorphic encryption, QRE is a distinct notion with different goals. We provide a more detailed comparison in Appendix A.

1.3 Application: A New Zero-Knowlege Σ-Protocol for QMA

To highlight the usefulness of the notion of QRE, we present an application to designing zero-knowledge (ZK) protocols for the complexity class **QMA**. Specifically we show how to easily obtain 3-round "sigma" (abbreviated by Σ) protocols for **QMA** using QRE as a black box, and in fact our construction achieves features that were not known before in the literature. We elaborate more below.

Zero-knowledge proofs [GMR89] is one of the most basic and useful notions in cryptography. Essentially, it is an interactive proof system where the verifier is guaranteed to learn nothing beyond the validity of the statement being proven. This is formalized by showing that for any accepting instance and any (possibly malicious) verifier, there exists a simulator which can generate a view which is indistinguishable from the actual view of the verifier in an interaction with an honest prover. The notion of indistinguishability depends on whether we are considering computational or statistical zero-knowledge.

In the classical setting, the canonical ZK protocol for **NP** was presented by Goldreich, Micali, and Wigderson [GMW91]. The protocol has a simple 3-message structure (known as the Σ format): the prover sends a message, the verifier sends a uniformly random challenge, the prover responds, and the verifier decides to accept or reject based on the transcript of the communication. Aside from their simplicity, Σ-protocols are also desirable because it one can then use the Fiat-Shamir heuristic [FS86] to make the protocol non-interactive, for example. In some cases, it is useful to have a Σ-protocol with a single challenge bit. In the classical case this implies a notion known as "special soundness" which is useful, for example, for constructing non-interactive zero-knowledge (NIZK) protocols [BFM88, FLS90]. Another useful feature is "delayed input", where the prover can produce the first message without any knowledge of the instance or the witness. This is useful for confirming well-formedness of the execution of a protocol while minimizing the number of

extra rounds of communication. In the classical setting Blum's Graph-Hamiltonicity protocol [Blu86, FLS90] has these properties and is thus often used.

Zero-knowledge proof systems for **QMA**, the quantum analogue of **NP**, have only been studied fairly recently, and known results are still few [BJSW16, VZ20, CVZ20, BG19, BS20]. Recently, Broadbent and Grilo [BG19] presented the first ZK Σ-protocol for **QMA**, achieving constant soundness error. Their protocol relies on a reduction to a special variant of the local Hamiltonians problem. It requires multi-bit challenges and does not seem to support delayed inputs.

In this work, we show a simple approach for obtaining a Σ-protocol with a single-bit challenge and delayed-input functionality, using quantum randomized encodings. Like [BG19], our protocol also has constant soundness error. In contrast to [BG19], we do not require a reduction to a specific **QMA**-complete problem. Our approach is similar to constructions of ZK protocols from randomized encoding in the classical setting [HV16].

Conceptually, the protocol is simple. Recall that a **QMA** problem L is defined by a (quantum polynomial time) verifier circuit V, which takes a classical instance x and a quantum witness $\boldsymbol{w}$ and decides (with all but negligible probability) whether x is a yes or no instance. We generically create a zero-knowledge protocol, where the basic idea is as follows. The prover creates a QRE of V, and sends it to the verifier, together with commitments to the labels and the randomness used to generate the QRE. The verifier sends a challenge bit b. Now for $b = 0$ simply all the commitments are opened and the verifier checks that indeed the proper circuit was encoded, if $b = 1$ then only the labels corresponding to the actual $x, \boldsymbol{w}$, and the verifier can thus check the value of V on them. The actual protocol is slightly more complicated since the QRE only has "labels" for classical inputs, and $\boldsymbol{w}$ needs to be hidden even given the labels. Therefore $\boldsymbol{w}$ is treated slightly differently than described above (essentially "teleported" into the circuit).

1.4 Future Directions and Open Problems

We end this section with several examples of future directions and open problems.

1. **Applications of QRE**. We presented one application in the form of a simple zero-knowledge protocol for **QMA**. Given the variety of applications of RE in classical cryptography, we anticipate that there is similarly many analogous applications in the quantum setting. We elaborate on several potential applications in Appendix B.

2. **Obtain statistically-private QRE for all log-depth circuits**. Our information-theoretic QRE has overhead that is *doubly-exponential* in the depth of the circuit being encoded (although as mentioned it should be possible to encode certain classes of circuits with "only" an exponential overhead). Thus there is a gap between what is achievable with classical RE (where it is possible to encode all log-depth circuits with statistical privacy). Can information-theoretic QRE be achieved for all log-depth circuits, or is the gap inherent? We note that it is not known whether statistically secure RE can be performed for all polynomial-size *classical* circuits.

3. **Completely classical encoding for quantum circuits**. Can the encoding of a quantum circuit be made completely classical? This would be very useful for obtaining obfuscation for quantum circuits, assuming classical obfuscation.

1.5 Paper Organization

Section 2 contains a technical overview of our contribution. Section 3 contains notation and preliminaries about quantum computation and classical randomized encoding. In Section 4 we define the notion of quantum randomized encoding, state some of its basic properties and state our main result. The details of our new zero-knowledge Σ-protocol appear in Section 5. Section 6 contains our Quantum Garbled Circuits construction and Section 7 contains proofs of correctness and privacy. We note that there is no dependence at all (or vice versa) between the last two sections and Section 5, and the order of reading them should be up to the reader's preference.

Acknowledgments

We thank Sanjam Garg and Vinod Vaikuntanathan for insightful discussions. 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).

2 Overview of Our Construction

We provide an overview of our techniques, we refer to the technical sections for the formal presentation and proofs. In what follows, we use bolded variables such as $\boldsymbol{q}$ to denote density matrices, and for a unitary U we write $U(\boldsymbol{q})$ to denote the state $U\boldsymbol{q}U^\dagger$ (see Section 3 for more details about notation).

2.1 Our Approach: Quantum Computation via Teleportation

The basis of our approach to quantum RE is *computation by teleportation*, an idea that is common to many prior results on protocols for delegated quantum computation and computing on encrypted data [BFK09, BJ15, DSS16]. We briefly review this concept.

Recall that quantum teleportation allows one party to transmit a qubit $\boldsymbol{q}$ to another party using only classical communication and a preshared EPR pair $\boldsymbol{e} = (\boldsymbol{e}_1, \boldsymbol{e}_2)$. Specifically, the sender performs a measurement on the qubit $\boldsymbol{q}$ and $\boldsymbol{e}_1$ to obtain two (uniformly distributed) classical bits a, b (often called "teleportation keys"). The receiver's qubit $\boldsymbol{e}_2$ collapses to $X^a Z^b(\boldsymbol{q})$ where X, Z are the bit-flip and phase-flip Pauli matrices respectively. Using the teleportation keys (a, b) the original qubit $\boldsymbol{q}$ can be recovered.

Teleportation can be used to apply gates: let G be a single-qubit unitary (the generalization to multi-qubit unitaries is straightforward), and suppose that the sender and receiver share the state $(\boldsymbol{e}_1, G(\boldsymbol{e}_2))$ instead, in which G is applied to the second half of an EPR pair. When the sender teleports the qubit $\boldsymbol{q}$ and obtains teleportation keys (a, b), the resulting state on the receiver's side is $G(X^a Z^b(\boldsymbol{q}))$. If G is a Clifford gate (i.e. a unitary that normalizes the Pauli group), then this is equal to $X^{a'} Z^{b'}(G(\boldsymbol{q}))$ for some updated keys (a', b') that are a deterministic function of (a, b) and G. We call $X^{a'} Z^{b'}$ the *Pauli error* on the state.

This already suggests a method of quantum randomized encoding for the class of Clifford circuits. Let C be a circuit consisting of gates $G_1, \ldots, G_m$. The encoding of the circuit C and an

n-qubit quantum input $\boldsymbol{x}$ can be computed in the following way:

1. Generate EPR pairs $\boldsymbol{e}^w = (\boldsymbol{e}_1^w, \boldsymbol{e}_2^w)$ for each wire w of the circuit C.[5]
2. For each gate G_i, if the input wires as specified by circuit C are v_1, v_2 (if G_i is a two-qubit gate, for example), then apply G_i to the "second halves" $(\boldsymbol{e}_2^{v_1}, \boldsymbol{e}_2^{v_2})$ of the corresponding EPR pairs. Note that after this operation the qubits $(\boldsymbol{e}_2^{v_1}, \boldsymbol{e}_2^{v_2})$ now store the output of G_i.
3. If wire v is connected to wire w via some gate, perform the teleportation measurement on the qubits $(\boldsymbol{e}_2^v, \boldsymbol{e}_1^w)$ to obtain classical teleportation keys (a_{vw}, b_{vw}).
4. If wire w is the i-th input wire to circuit C, then perform the teleportation measurement on qubits $(\boldsymbol{x}_i, \boldsymbol{e}_1^w)$ to obtain classical teleportation keys (a_i, b_i).
5. Compute from all the intermediate teleportation keys $(a_i, b_i)_i$ and $(a_{vw}, b_{vw})_{v,w}$ the final teleportation keys (a'_j, b'_j) corresponding to the j-th output qubit, for each j. The final teleportation keys are a deterministic function f_{corr} of all the intermediate teleportation keys, as well as the gates $G_1, \ldots, G_m$.

Each of the teleportation operations will yield uniformly random teleportation keys for each pair of connected wires, inducing Pauli errors that accumulate as the teleported state "moves" through the circuit. Since all gates are Clifford, the Pauli errors get adjusted in a deterministic way, and the resulting state in the qubits $(\boldsymbol{e}_2^w)$ for output wires w will be $X^{a'_1}Z^{b'_1} \otimes \cdots \otimes X^{a'_n}Z^{b'_n}(C(\boldsymbol{x}))$. This output state, along with the final teleportation keys $(a'_j, b'_j)_j$, yields a QRE of circuit C and input $\boldsymbol{x}$, because the final state $C(x)$ can be recovered from this, and it yields no information about the gates or the original input $\boldsymbol{x}$ (as long as the intermediate teleportation keys are not revealed). The *quantum* complexity of this encoding is quite low: preparing the EPR pairs, applying the gates, and applying the teleportation measurements can be parallelized and thus performed in constant-depth. However the *classical* complexity of this encoding is dominated by the complexity of computing the final teleportation keys $(a'_j, b'_j)_j$, which takes time that is linear in the size of the circuit C.

This complexity issue can be solved by leveraging *classical randomized encodings* (CREs): the encoder, instead of computing $(a'_j, b'_j)_j$ itself, computes a randomized encoding $\hat{f}_{\text{corr}}(\vec{k})$ of the function f_{corr} and intermediate teleportation keys $\vec{k}$. Using a decomposable RE scheme such as (classical) garbled circuits, it is possible to compute $\hat{f}_{\text{corr}}$ using a constant-depth circuit; this encoding corresponds of an offline part $\hat{f}_{\text{corr,off}}$ and an online part that consists of *labels* for each bit of the teleportation keys $\vec{k}$. Thus the overall quantum encoding of $C(\boldsymbol{x})$ will be the quantum state $X^{a'_1}Z^{b'_1} \otimes \cdots \otimes X^{a'_n}Z^{b'_n}(C(\boldsymbol{x}))$, along with the CRE $\hat{f}_{\text{corr}}(\vec{k})$. The decoder can compute from the CRE the final teleportation keys, and then recover $C(\boldsymbol{x})$.

We see that this yields a simple QRE for Clifford circuits. If the CRE used is decomposable, the QRE is decomposable as well: observe that the input qubits $\boldsymbol{x}$ are encoded separately from the encoding of the circuit (the input teleportation measurements and the computation of the input labels for $\hat{f}_{\text{corr}}$ can be done independently). Furthermore, the QRE has information-theoretic (resp. computational) privacy if the CRE has information-theoretic privacy (resp. computational).

[5] One can think of a *wire* as a line segment in a circuit diagram in between the gates, as well as the segments for the inputs/output qubits.

2.2 The Challenge: Going Beyond Clifford Gates

The real challenge comes from dealing with the case of non-Clifford gates in the circuit (such as the $T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$ gate).[6] The QRE described above does not work when one of the gates is a T gate; this is because the gate teleportation protocol induces a non-Pauli error: $T(X^a Z^b(\boldsymbol{q})) = X^{a'} Z^{b'} P^{a'}(T(\boldsymbol{q}))$, where $P = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$ is the phase gate (a Clifford gate). Thus we no longer have the invariant that the intermediate states of the teleportations are the masked by Pauli errors. This is problematic because the errors that are induced via the sequence of teleportations will no longer be simple to compute classically.

Instead, a phase error induced by a gate teleportation should be removed before the next teleportation. However, there appears to be a catch-22: in order to know whether there is a phase error, a teleportation measurement needs to be performed to get the keys (a, b). On the other hand, the teleportation can only be performed if one was certain that there was no phase error from previous teleportations! If the encoder wants to avoid performing a sequential computation, it appears that both the teleportation measurements and the corresponding Pauli/phase error corrections have to be performed by the evaluator – and this must be done in a way that does not violate the privacy of the encoding.

We now describe the key ideas used in our Quantum Garbled Circuits scheme to handle these issues.

Encrypted Teleportation Gadgets. First, to allow the teleportation measurements to be performed by the evaluator in a manner that maintains the privacy of the encoding, the encoder will apply *encrypted teleportation gadgets* between the connected EPR pairs. For simplicitly assume that G is a single-qubit gate in the circuit that connects wire v to wire w. The encoder applies gate G to the EPR qubit $\boldsymbol{e}_2^v$ as in the Clifford encoding, but instead of performing the teleportation measurements on the pair $(\boldsymbol{e}_2^v, \boldsymbol{e}_1^w)$, the encoder applies the following circuit to the two qubits as well as additional ancilla:[7]

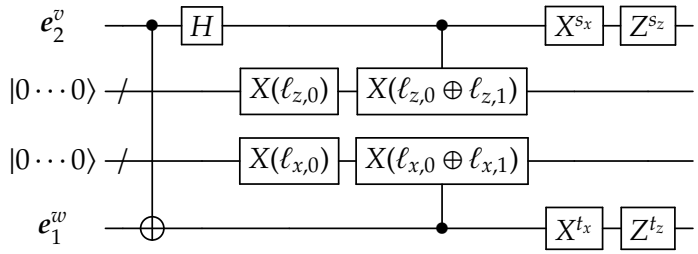


Figure 1: Encrypted teleportation gadget.

Here, $(\ell_{z,0}, \ell_{z,1}, \ell_{x,0}, \ell_{x,1})$ are random strings in $\{0,1\}^\kappa$ (we will say in a moment where they come from), and s_x, s_z, t_x, t_z are uniformly random bits. For a string $r \in \{0,1\}^\kappa$, the gate $X(r)$ denotes the κ-qubit gate that applies a bit-flip gate X to qubit i if $r_i = 1$.

The functionality of this circuit is to perform the teleportation measurement on $(\boldsymbol{e}_2^v, \boldsymbol{e}_1^w)$, but instead of obtaining teleportation keys $(a, b) \in \{0,1\}^2$ as usual, the middle two wire-bundles yield

[6] Recall that when augmenting the Clifford group with any non-Clifford such as the T gate, the resulting set of gates is universal for quantum computation.

[7] If G is a multiqubit gate, then the encoder will apply G to qubits $(\boldsymbol{e}_2^{v_1}, \ldots, \boldsymbol{e}_2^{v_p})$ before applying the teleportation gadget to each $(\boldsymbol{e}_2^{v_i}, \boldsymbol{e}_1^{w_i})$.

teleportation labels $(\ell_{z,a}, \ell_{x,b})$ which indicate the Pauli error $X^b Z^a$ on the teleported qubit.[8] As long as the randomness used to choose $(\ell_{z,0}, \ell_{z,1}, \ell_{x,0}, \ell_{x,1})$ and the bits s_x, s_z, 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).

Switching The Order of Correction and Teleportation. There still is the issue that before applying the encrypted teleportation gadget to a specific pair $(\boldsymbol{e}_2^v, \boldsymbol{e}_1^w)$, the input qubit may have a Pauli/phase error that needs correcting. For example, imagine that the qubit $\boldsymbol{e}_2^v$, originally the second half of an EPR pair, now stores a qubit $X^a Z^b(\boldsymbol{q})$ due to a previous teleportation. When we apply the gate G (which we again assume is a single-qubit gate for simplicity) to $\boldsymbol{e}_2^v$, we obtain the following correction: $G(X^a Z^b(\boldsymbol{q})) = R(G(\boldsymbol{q}))$ where R is a Pauli or phase correction. We need to avoid having the correction R before applying the teleportation gadget.

We show that we can essentially *switch* the order of the two operations: instead of performing the Pauli/phase correction and then applying the encrypted teleportation gadget, we show that a circuit similar to the teleportation gadget can be applied *first* by the encoder (obliviously to whether the correction R was needed or not), and *then* the consequences of this bold move can be dealt with by the encoder. More precisely, if we let $\mathsf{TP}_{\ell,s,t}$ denote the encrypted teleportation gadget, then there exist unitaries $\Lambda_1, \Lambda_2, \Lambda_3$ such that the following circuit identity holds:

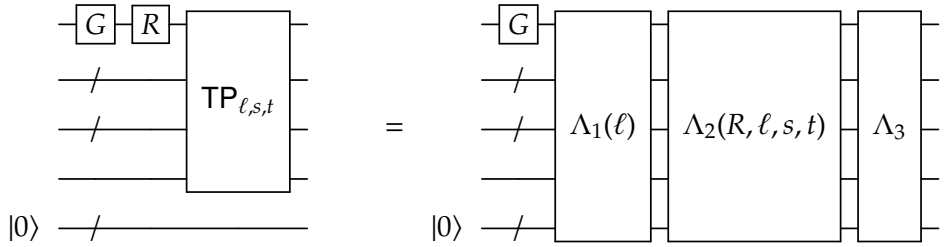


Figure 2: Switching the order of correction and teleportation.

Here, Λ_1 is a circuit that only depends on the labels ℓ. The circuit Λ_2 depends on all parameters R, ℓ, s, t. The circuit Λ_3 does not depend on any parameters. The circuits on the right-hand side use additional ancilla.

Thus, the encoder can first apply the gate G and then the circuit $\Lambda_1(\ell)$ to the connecting EPR pairs and ancilla qubits, because it knows the random labels ℓ. The idea is then to offload the task of applying the remaining circuits $\Lambda_2(R, \ell, s, t)$ and Λ_3 – which we call *correction gadgets* – to the evaluator in a way that hides the values of R, ℓ, s, t (and thus maintains the privacy of the encoding). Since Λ_3 doesn't depend on any parameters, the decoder can always automatically apply this. For Λ_2, the encoder computes a quantum randomized encoding of $\Lambda_2(R, \ell, s, t)$ — but this time we have an easier task because $\Lambda_2(R, \ell, s, t)$ is a Clifford circuit (in fact, it comes from a special subset of depth-1 Clifford circuits). In principle one could then recursively compute the QRE of Λ_2 using the Clifford scheme described above, but for our Quantum Garbled Circuits scheme we employ a different method called *group-randomizing QRE*.

Group-Randomizing QRE. In this section we explain group-randomizing QRE in greater generality than is needed for our Quantum Garbled Circuits scheme, because we believe that it is a

[8] The "label" terminology is inspired by how random labels are used to encrypt the contents of each wire in classical garbled circuits.

useful conceptual framework for thinking about randomizing computations and may be useful for other applications of QRE.

Let $\mathcal{G}$ denote a subgroup of the n-qubit unitary group. Let C denote a circuit computing a unitary in $\mathcal{G}$, and let x denote an input. The encoding of $C(x)$ is a pair $\hat{C}(x) = (R(x), CR^\dagger)$, where R is a uniformly random[9] element of $\mathcal{G}$, and we assume that $CR^\dagger$ is described using some canonical circuit representation for elements of $\mathcal{G}$. Clearly, the output $C(x)$ can be decoded from this encoding: the decoder simply applies $CR^\dagger$ to $R(x)$. This encoding is also perfectly private: for a uniformly random $R \in \mathcal{G}$, the unitary $CR^\dagger$ is also uniformly distributed in $\mathcal{G}$. Thus, a simulator on input y can output $(E^\dagger(y), E)$ for a uniformly random element $E \in \mathcal{G}$. 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 [Kil88].

For general unitaries, group-randomizing QRE is exorbitantly inefficient, because a general unitary (even after discretization) requires exponentially many random bits to describe. However when the group $\mathcal{G}$ is the Clifford group, for example, then this encoding is efficient: the n-qubit Clifford group is a finite group of order $2^{\Theta(n^2)}$, and a uniformly random element can be sampled in poly(n) time. Furthermore, there are efficient algorithms to compute canonical representations of Clifford circuits [Got98, AG04]. In particular, an n-qubit Clifford unitary can always be written as a Clifford circuit with $O(n^2)$ gates, and this is tight.

We now apply this group-randomizing framework to encoding the correction circuits Λ_2 described above. Let $\mathcal{G}$ denote a unitary subgroup that contains Λ_2; this will be a group of a special class of depth-1 Clifford circuits. The circuits Λ_2 depends on parameters R, ℓ, s, t. The parameters (ℓ, s, t) are randomness that is known to the encoder, so this can be be fixed. However, as discussed the correction unitary R is not known ahead of time to the encoder – it depends on the teleportation keys (a, b) of a *previous* gate teleportation, as well as the gate G that was involved. In fact R is a deterministic function of (a, b), and therefore the description of $\Lambda_2(R, \ell, s, t)$ is a deterministic function of (a, b).

To encode the circuit $\Lambda_2(R, \ell, s, t)$ with quantum input x, the encoder first samples a uniformly random element $A \in \mathcal{R}$. Then it applies the unitary A to the input x. This constitutes the quantum part of the encoding.

For the classical part: let $f_{\text{corr},A,\ell,s,t}(a, b)$ denote the function that computes a canonical description of the unitary $\Lambda_2(R, \ell, s, t) \cdot A^\dagger \in \mathcal{G}$. Since Λ_2 is a Clifford unitary on $O(\kappa^2)$ qubits (the ancillas at the bottom of Figure 2 consist of $O(\kappa^2)$ qubits), the function $f_{\text{corr},A,\ell,s,t}$ can be computed by a poly(κ)-sized circuit. The encoder can then efficiently compute a classical decomposable RE $\hat{f}_{\text{corr},A,\ell,s,t}$ of $f_{\text{corr},A,\ell,s,t}$. Since the input (a, b) is not known yet, the encoding consists of an offline part $\hat{f}_{\text{off}}$ and online labels $(\ell'_{z,0}, \ell'_{z,1}, \ell'_{x,0}, \ell'_{x,1})$ for the 4 possible values of (a, b). The classical part of the encoding of Λ_2 is simply the offline part $\hat{f}_{\text{off}}$.

Thus, the decoder has the encoding $(A(x), \hat{f}_{\text{off}})$. Suppose for the moment that the decoder had two out of the four labels $(\ell'_{z,a}, \ell'_{x,b})$ for some $(a, b) \in \{0, 1\}^2$. Then the tuple $(\hat{f}_{\text{off}}, \ell'_{z,a}, \ell'_{x,b})$ denotes the RE $\hat{f}_{\text{corr},A,\ell,s,t}(a, b)$, from which the value $f_{\text{corr},A,\ell,s,t}(a, b) = \Lambda_2(R, \ell, s, t) \cdot A^\dagger$ can be efficiently decoded. Given this, the decoder can then decode $\Lambda_2(R, \ell, s, t)(x)$, as desired. The privacy of this encoding follows from the group-randomizing property described above.

[9] For the purpose of this discussion we can always assume that $\mathcal{G}$ is finite.

2.3 Putting it Together

We now put everything together. For every pair v, w of wires in the circuit connected by a gate G, the encoder applies the circuit $\Lambda_1(\ell_{vw})$ to the EPR pairs corresponding to v, w. Then, it computes the encoding of $\Lambda_2(\cdot, \ell_{vw}, s_{vw}, t_{vw})$ as described. Here, s_{vw}, t_{vw} are uniform randomness sampled for every connected pair (v, w). The random labels ℓ_{vw} are generated from the classical RE $\hat{f}_{\text{corr}}$ of a Λ_2 circuit corresponding to a *subsequent* pair (v', w') of connected wires.

Thus, the decoder can sequentially decode each wire[10] of the circuit in the following manner: inductively assume that the decoder has two-of-four labels $(\ell'_{z,a}, \ell'_{x,b})$ resulting from a *previous* encrypted teleportation measurement. Then the decoder can decode the group-randomizing QRE of the Λ_2 circuits, and then apply the Λ_3 circuits. For each pair (v, w) of wires, the resulting state on the qubits $(\boldsymbol{e}_2^v, \boldsymbol{e}_1^w)$ will be $\Lambda_3 \cdot \Lambda_2(R, \ell, s, t) \cdot \Lambda_1(\ell) \cdot G$, which by Figure 2 is equivalent to $\mathsf{TP}_{\ell,s,t} \cdot R \cdot G$, the desired gate followed by "correct-then-teleport" operation[11] Measuring the middle wire-bundles of the teleportation gadget to obtain labels $(\ell_{z,d}, \ell_{x,e})$ sets the decoder up for decoding a subsequent wire pair (v', w').

The encoder also provides a "dictionary" of labels for the output qubits, so that the decoder can undo any final Pauli/phase errors. All together, the decoding is equivalent to a sequence of correct-then-teleport operations, followed by a final correction, which allows the decoder to recover the value $C(\boldsymbol{x})$.

The encoding is constant-depth. The classical RE used in the encoding of each Λ_2 circuit can be done all in parallel and in constant-depth (assuming the CRE scheme has constant-depth encoding). Then the quantum part of the encoding for each pair of wires (v, w) can be computed in constant-depth, because the Λ_1 circuits are constant-depth and the group $\mathcal{G}$ consists of constant-depth circuits.

In the setting of information-theoretic privacy, the size of the encoding grows doubly-exponentially with the depth of the circuit. This is because if the labels ℓ for a pair of wires is κ bits, then the corresponding teleportation gadget $\mathsf{TP}_{\ell,s,t}$ has size $\Theta(\kappa)$, and thus the corresponding Λ_2 circuit has size $\Theta(\kappa^2)$, which means that the complexity of computing the correction function f_{corr} is at least $\Theta(\kappa^2)$. This means that the labels for statistically-secure classical RE of f_{corr} are at least $\Omega(\kappa^2)$ bits long. Thus the label sizes square with each layer. In the worst-case, this means that we can only achieve efficient statistically-private encodings of $O(\log \log n)$-depth circuits.[12] With some simple modifications to this scheme, it is possible to handle certain circuits of larger depth, e.g., $O(\log n)$-depth circuits where all layers except for the first $O(\log \log n)$ layers consist of Clifford gates.

In the setting of computational privacy (e.g., where we assume the existence of pseudorandom generators), the size of the encoding is polynomial in the size of the circuit. This is because the label sizes of computationally-secure classical RE are *independent* of the complexity of the function being encoded; they only depend on the security parameter.

[10] This should be viewed as analogous to the decoding in classical garbled circuits.

[11] Again we assume that the gate G is a single-qubit gate; in the multiqubit case the gate G is "spread" across multiple $\boldsymbol{e}_2^v$ qubits.

[12] In a previous version of this paper, we erroneously claimed that the label sizes grow linearly with each layer in the information-theoretic setting, and concluded that the encoding size grows exponentially with the depth (which would match the encoding complexity of known information-theoretic classical RE schemes). We find it an interesting problem to determine whether we can avoid the doubly-exponential complexity blow-up for information-theoretic QRE.

2.4 Privacy of Our Scheme

The privacy properties of our randomized encoding scheme is established via the existence of a *simulator*, which is an efficient procedure Sim that takes as input a quantum state $F(\boldsymbol{x})$ for some quantum circuit F and state $\boldsymbol{x}$, and produces another quantum state that is indistinguishable from the randomized encoding $\hat{F}(\boldsymbol{x})$. This formalizes the idea that the only thing that can be learned from the randomized encoding $\hat{F}(\boldsymbol{x})$ is the value $F(\boldsymbol{x})$.

An important feature of the randomized encoding $\hat{F}(\boldsymbol{x})$ 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 $\boldsymbol{x}$, or the encoding of a circuit E with the same topology as F, but with all identity gates, with input $F(\boldsymbol{x})$. In both cases, the decoding process (which is a public procedure which does not require any secrets) produces the output $F(\boldsymbol{x})$. Thus, there is a canonical choice of simulator for such a randomized encoding: given input $F(\boldsymbol{x})$, it computes the randomized encoding $\hat{E}(F(\boldsymbol{x}))$, which is indistinguishable from the randomized encoding $\hat{F}(\boldsymbol{x})$ via the circuit hiding property.

We now discuss a subtle point. We have described the decoding/evaluation procedure as involving measurements: the decoder is supposed to measure the middle wire-bundles of the teleportation gadgets to obtain the labels needed to decode 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 decoding procedure Dec to be unitary. The randomization bits s_x, s_z, t_x, t_z in the teleportation gadgets, which are never revealed to the decoder, effectively *force* a measurement of the teleportation gadget labels. In particular, randomizing over the s_z, t_z bits destroy any coherence between the different labels in the superposition, which means that the decoder (even a malicious one) can only get labels for a single teleportation key per teleportation gadget at a time.

The unitarity of Dec is used in the privacy analysis in the following way: we show that given the randomized encoding of $F(\boldsymbol{x})$ and $E(F(\boldsymbol{x}))$, applying a unitary decoder 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 Dec^{-1} to the outputs preserves the indistinguishability (up to a loss related to the complexity of Dec), and shows that the encodings of $F(\boldsymbol{x})$ and $E(F(\boldsymbol{x}))$ are indistinguishable.

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 [Kil88].

The idea is to use the "magic state" representation of quantum circuits [BK05]. At a high level and using the terminology of this work, [BK05] 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).[13] 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)[14] 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).

[13] In fact, the [BK05] characterization is much more specific about what the layers look like and only uses very specific classical characterization, in particular each measured bit controls one gate, but for our purposes even the above suffices.

[14] Applying Z^s for a randomly chosen bit s removes all information except the information that is recoverable via measurement (in the computational basis). Therefore one can think of Z-twirl as equivalent to measurement (or as a randomized encoding of the measurement operation).

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.[15] 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 $\mathsf{a}, \mathsf{b}, \mathsf{c}$, etc. Let $\mathsf{w}_1, \mathsf{w}_2, \ldots, \mathsf{w}_n$ 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 $\bigcup_{i \in S} \mathsf{w}_i$. We write $\dim(\mathsf{a})$ to denote the dimension of register a.

We also use underbrackets to denote the registers associated with a state, e.g.,

$$\underbrace{|\psi\rangle}_{\mathsf{a}}, \qquad \text{and} \qquad \frac{1}{\sqrt{2}} \sum_e \underbrace{|e,e\rangle}_{\mathsf{vu}}.$$

The first denotes a pure state $|\psi\rangle$ in the register a, and the second denotes an EPR pair on registers vu, respectively.

Quantum Random Variables. A *quantum random variable (QRV)* $\boldsymbol{a}$ on a register a is a density matrix on register a. Note that we denote QRVs using bolded font. When referring to a collection $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$ of QRVs simultaneously, we are referring to the reduced density matrix of a global state on the registers abc – we say that $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$ is a *joint QRV*, which is also a QRV itself. We say that a QRV $\boldsymbol{a}$ is *independent* of a collection of QRVs $(\boldsymbol{b}, \boldsymbol{c})$ if the density matrix corresponding to $\boldsymbol{a}$ is in tensor product with the density matrix corresponding to $(\boldsymbol{b}, \boldsymbol{c})$. We denote this by $\boldsymbol{a} \otimes (\boldsymbol{b}, \boldsymbol{c})$.

Quantum Operations. Given a quantum operation F mapping register a to register a' and a collection of QRVs $(\boldsymbol{a}, \boldsymbol{b})$, we write $(F(\boldsymbol{a}), \boldsymbol{b})$ to denote the a density matrix on registers $\mathsf{a'b}$ that is the result of applying F to the density matrix $(\boldsymbol{a}, \boldsymbol{b})$. 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 $U(\boldsymbol{x})$ as shorthand for $U \boldsymbol{x} U^\dagger$.

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 $\widehat{\mathsf{Ckt}}$ 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.

[15] While the description above was completely sequential, one can notice that a blowup in labels of a certain gates does not effect other gates that are in the same level in the circuit, even though the encoding is described sequentially.

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

Distinguishability of Quantum States. We say that two quantum states $\boldsymbol{a}, \boldsymbol{b}$ on the same number of qubits are (t, ϵ)-*indistinguishable* if for all auxiliary quantum states $\boldsymbol{q}$ that are unentangled with either $\boldsymbol{a}$ or $\boldsymbol{b}$, for all quantum circuits D of size t,

$$\left| \mathbb{P}[D(\boldsymbol{a} \otimes \boldsymbol{q}) = 1] - \mathbb{P}[D(\boldsymbol{b} \otimes \boldsymbol{q}) = 1] \right| \leq \epsilon.$$

We often write $\boldsymbol{a} \approx_{(t,\epsilon)} \boldsymbol{b}$ to denote this. This notion of indistinguishability satisfies the triangle inequality: if $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$ are quantum states such that $\boldsymbol{a} \approx_{(t_1,\epsilon_1)} \boldsymbol{b}$, and $\boldsymbol{b} \approx_{(t_2,\epsilon_2)} \boldsymbol{c}$, then we have $\boldsymbol{a} \approx_{(\min(t_1,t_2),\epsilon_1+\epsilon_2)} \boldsymbol{c}$.

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

3.2 Quantum Gates and Circuits

3.2.1 Pauli, Clifford, and PX Groups

Pauli Group. The single-qubit *Pauli group* $\mathcal{P}$ consists of the group generated by the following Pauli matrices:

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad Y = \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix} \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

The n-qubit Pauli group $\mathcal{P}_n$ is the n-fold tensor product of $\mathcal{P}$.

Clifford Group. The n-qubit *Clifford group* $\mathcal{C}_n$ is defined to be the set of unitaries C such that

$$C\mathcal{P}_n C^\dagger = \mathcal{P}_n.$$

Elements of the Clifford group are generated by CNOT, Hadamard ($H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$), and Phase ($P = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$) gates.

The *third level of the n-qubit Clifford hierarchy* (Pauli and Clifford groups being the first and second, respectively) are unitaries U where

$$U\mathcal{P}_n U^\dagger = \mathcal{C}_n.$$

In other words, conjugating Paulis by U yield a Clifford element.

PX Group. The following subgroup of the Clifford group, which we call the PX-group, will be of particular interest to us. The group is defined as a group of single-qubit unitaries, and can be extended to multiple qubits by tensoring as usual. We define the single-qubit PX-group to be the group generated by the Pauli X gate and Phase P gate.

The aforementioned P operation is the "square root" of the Pauli Z, so $P^2 = Z$ (and therefore the Pauli group is a subgroup of the PX-group). The Pauli group is a strict subset of the PX-group (since the Pauli group does not contain P), and the PX-group itself is a strict subgroup of the single-qubit Clifford group (since the PX group does not contain the Hadamard gate).

Calculation shows that $PX = iXP^3$, which implies that any element in the PX-group can be written as $i^c X^a P^b$, where $a \in \{0,1\}$ and $b, c \in \{0,1,2,3\}$. This immediately implies that given a circuit that contains only I, X, P gates (and of course also $Z = P^2$ gates which are equivalent to two consecutive applications of P), it is possible to efficiently find its canonical representation as $i^c X^a P^b$. We refer to such a circuit as a "PX circuit". Given the canonical representation of a PX element, it is possible to find a canonical PX circuit that implements the operation of the PX circuit. Since the PX group is a group, applying a random PX operation on an arbitrary PX operation results in a random PX operation.

3.2.2 Universal Gate Set

A universal set of gates is $\mathcal{C}_2 \cup \{T\}$, 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 3.1. *For all $a, b \in \{0,1\}$ it holds that $TX^aZ^b = P^aX^aZ^bT$.*

3.2.3 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 $\mathbf{NC}^0$, 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 $\mathbf{NC}^0$ captures the complexity of encoding in classical randomized encoding schemes.

3.2.4 Quantum Circuits and Their Topology

Circuit Topology. A *quantum circuit topology* $\mathcal{T}$ is a tuple $(\mathcal{B}, \mathcal{I}, \mathcal{O}, \mathcal{W}, \texttt{inwire}, \texttt{outwire}, \mathcal{Z}, \mathcal{T})$ where

1. $G_{\mathcal{T}} = (\mathcal{B} \cup \mathcal{I} \cup \mathcal{O}, \mathcal{W})$ forms a directed acyclic graph (DAG) where the vertex set consists of the union of disjoint sets $\mathcal{B}$, $\mathcal{I}$, $\mathcal{O}$, and the edge set is $\mathcal{W}$.
2. The set of edges $\mathcal{W}$ are called *wires* of the circuit topology $\mathcal{T}$.
3. The set $\mathcal{I}$ is ordered, and consist of *input terminals*, and have in-degree 0, and out-degree 1. Throughout this paper we will overload notation and let $\mathcal{I}$ denote the subset of wires $\mathcal{W}$ that are incident to the input terminals.

4. The set O is ordered, and consist of *output terminals*, and have in-degree 1, and out-degree 0. Throughout this paper we will overload notation and let O denote the subset of wires $\mathcal{W}$ that are incident to the output terminals.

5. The vertices $\mathcal{B}$ are called *placeholder gates*, and for every $g \in \mathcal{B}$, the in-degree and out-degree of g are equal. We let p_g denote the degree, which we call the *arity* of the placeholder gate g.

6. For every $g \in \mathcal{B}$, we let $\texttt{inwire}(g)$ denote an ordering of the wires $w \in \mathcal{W}$ that enter g, and let $\texttt{outwire}(g)$ denote an ordering of the wires that exit g.

7. The set $\mathcal{Z}$ is a subset of $\mathcal{I}$ that denotes *zero qubits* (i.e. qubits to initialize in the state $|0\rangle$).

8. $\mathcal{T} \subseteq O$ denotes the set of *discarded qubits* (i.e. output qubits to trace out).

A topology thus specifies a quantum circuit in a natural way, except only "placeholder gates" are specified. Note that the number of input terminals must be equal to the number of output terminals; these correspond to the input and output wires of the circuit topology. We often let n denote the number of input (and therefore output) terminals. Furthermore, a circuit topology allows some input qubits to be initialized in the $|0\rangle$ state, and some output qubits to be traced out.

Given a topology $\mathcal{T}$, we define an *evaluation order* $\pi : \mathcal{B} \to |\mathcal{B}|$ to be a topological ordering of the "blank gates".

Quantum Circuits. A *general quantum circuit* F is a pair $(\mathcal{T}, \mathcal{G})$ where

$$\mathcal{T} = (\mathcal{B}, \mathcal{I}, O, \mathcal{W}, \texttt{inwire}, \texttt{outwire}, \mathcal{Z}, \mathcal{T})$$

is a circuit topology and $\mathcal{G}$ is a set of unitaries such that for every $g \in \mathcal{B}$, there is a corresponding p_g-qubit unitary $U_g \in \mathcal{G}$. We often write $g \in \mathcal{G}$ to denote the unitary itself. The *size* of a unitary circuit C is the number of wires $|\mathcal{W}|$. For an $(n - |\mathcal{Z}|)$-qubit state $\boldsymbol{w}$ supported on the qubits not indexed by $\mathcal{Z}$, we write $F(\boldsymbol{w})$ to denote the density matrix resulting from applying the gates $g \in \mathcal{G}$ to $(\boldsymbol{w}, 0)$ where the qubits indexed by $\mathcal{Z}$ are set to $|0\rangle$, and at the end the qubits indexed by $\mathcal{T}$ are traced out.

A *unitary quantum circuit* C is a circuit where the set $\mathcal{Z} = \mathcal{T} = \emptyset$. In other words, it maps n qubits to n qubits, and no qubits are discarded at the end.

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 choose to work with the universal set $\mathcal{C}_2 \cup \{T\}$ as described in Section 3.2.2.

Constant-Depth Quantum Circuits. The main model of constant-depth quantum circuits that we consider in this paper are $\mathbf{QNC}^0_f$ 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: $|x, y_1, \ldots, y_k\rangle \mapsto |x, y_1 \oplus x, y_2 \oplus x, \ldots, y_k \oplus x\rangle$). This is a natural analogue of classical $\mathbf{NC}^0$ circuits, yet is surprisingly powerful: functions such as PARITY can be computed in this model [Moo99, HS05]. We will show that $\mathbf{QNC}^0_f$ 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 survey by Applebaum [App17] for details and references.

Definition 3.2 (Classical Randomized Encoding)**.** Let $f \in \{0,1\}^n \to \{0,1\}^m$ be some function. The function $\hat{f} : \{0,1\}^n \times \{0,1\}^\ell \to \{0,1\}^{m'}$ is a (t,ϵ)-private *classical randomized encoding* (CRE) of f if there exist a deterministic function CDec (called a *decoder*) and a randomized function CSim (called a *simulator*) with the following properties.

- **Correctness.** For all x, r it holds that $f(x) = \mathsf{CDec}(\hat{f}(x;r))$.[16]
- (t,ϵ)**-Privacy.** For all x and for all circuits D of size t it holds that

$$\left| \mathbb{P}_r[D(\hat{f}(x;r)) = 1] - \mathbb{P}[D(\mathsf{CSim}(f(x)) = 1] \right| \leq \epsilon$$

  where the second probability is over the randomness of the simulator 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 $\hat{f}_{\mathsf{off}}(r)$ (called the *offline part of the encoding*) and $\mathsf{lab}_{i,b}(r)$ (called the *label functions*) for all $i \in [n], b \in \{0,1\}$, such that for all (x,r),

$$\hat{f}(x;r) = \left(\hat{f}_{\mathsf{off}}(r) \,,\, (\mathsf{lab}_{i,x_i}(r))_{i\in[n]} \right).$$

Remark 3.3. 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 $\hat{f}(x;r)$ 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 3.2, there is no requirement that the randomized encoding $\hat{f}$ can be efficiently computed from the original function f. Furthermore, the decoder CDec and simulator 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 [App14b]), 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 (see Section 3.2.3 for an overview of classical circuit topology). This motivates the following general definition.

Definition 3.4 (Universal RE Schemes for Circuits)**.** Let C denote a class of circuits and let $\mathcal{R}$ denote an equivalence relation over C. An (t,ϵ)*-private and efficient* $\mathcal{R}$*-universal RE scheme for the class* C is a tuple of polynomial-time algorithms $(\mathsf{CEnc}, \mathsf{CDec}, \mathsf{CSim})$ such that for all circuits $f \in C$ (here we identify the circuit with the function it computes),

- **Efficient Encoding**. For all x, r, $\mathsf{CEnc}(f, x; r)$ computes a randomized encoding $\hat{f}(x, r)$.
- **Correctness**. For all x and r it holds that $f(x) = \mathsf{CDec}(c, \hat{f}(x;r))$ where c denotes the equivalence class of f in $\mathcal{R}$.

[16] This is known as *perfect correctness* and is the only notion of correctness considered in this work.

- (t, ϵ)**-Privacy**. For all x and circuits D of size t it holds that

$$\left| \mathbb{P}_r[D(\hat{f}(x;r)) = 1] - \mathbb{P}[D(\mathsf{CSim}(c, f(x)) = 1] \right| \leq \epsilon \ .$$

  where c denotes the equivalence class of f in $\mathcal{R}$.

Furthermore, we say that the $\mathcal{R}$-universal RE scheme $(\mathsf{CEnc}, \mathsf{CDec}, \mathsf{CSim})$ is *decomposable* if the randomized encoding $\hat{f}(x;r)$ is decomposable, and furthermore we say that it is *label-universal* if the label functions $\mathsf{lab}_{i,b}(r)$ only depend on the equivalence class of f in $\mathcal{R}$.

Remark 3.5 (Universal RE and Universal Circuits)**.** For many classes of functions it is possible to achieve universality for RE using the notion of a universal circuit (or machine). If the class of functions admits a universal circuit (which depends on a property such as the topology but takes the remainder of the description as input), and this universal circuit itself belongs to the class that can be encoded using the RE scheme, then one can apply the RE to the universal circuit and consider the description of f as additional input to the circuit. This will result in a universal RE scheme, and if the original RE was decomposable then the resulting scheme will be decomposable with respect to both the input and the description of the function.

For the remainder of this paper, when we speak of encoding a function f, we are referring to encoding a specific *circuit implementation* of f (see Section 3.2.3). Furthermore, in this paper we will be focused on *topologically-universal RE schemes* – in other words, the equivalence relation $\mathcal{R}$ is such that two circuits f, f' are equivalent if they have the same topology (i.e. the same interconnection between gates, but possibly different gate functionality). Randomized encoding schemes in the literature are typically topologically-universal (for example Yao's garbled circuits scheme).

We note that label universality can be derived from $\mathcal{R}$-universality in a generic way (essentially by using a straightforward label-universal encoding of the multiplexer functions), but the constructions we cite from the literature will have this property even without this transformation.

Existing Decomposable Classical RE Schemes. We use decomposable CRE (DCRE) as a building block for our construction of quantum RE. In particular we rely on the following information theoretical and computational schemes [BMR90, Rog91, App17].

Theorem 3.6 (Information Theoretic DRE)**.** *There exists an efficient topologically-universal and label-universal DRE scheme* $\mathsf{CRE} = (\mathsf{CEnc}, \mathsf{CDec}, \mathsf{CSim})$ *with the following properties:*

- ***Efficiency.*** *For every function f computable by a size-s and depth-d classical circuit and for every input x, the encoding $\mathsf{CEnc}(f, x; r)$ is computable in time $\mathrm{poly}(2^d) \cdot s$. Furthermore, the label functions $\mathsf{lab}_{i,b}(r)$ are computable in time $\mathrm{poly}(2^d)$. The decoding and simulation algorithms CDec and CSim are also computable in time $\mathrm{poly}(2^d) \cdot s$.*

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

- ***Locality.*** *Every output bit of $\hat{f}(x;r) = \mathsf{CEnc}(f, x; r)$ depends on at most* 4 *bits of (x, r).*

We note that the locality and efficiency properties of the DRE scheme specified by Theorem 3.6 implies that the randomized encodings $\hat{f}(x;r)$ are computable by $\mathbf{NC}^0$ circuits that take as input x and the randomness r.

Theorem 3.7 (Computational DRE)**.** *Assume there exists a length doubling pseudorandom generator (PRG)* G *that is secure against polynomial time classical (resp. quantum) adversaries. There exists an efficient topologically-universal and label-universal DRE scheme* $\mathsf{CRE} = (\mathsf{CEnc}, \mathsf{CDec}, \mathsf{CSim})$*, which implicitly depends on a security parameter* λ*, and has the following properties:*

- ***Efficiency.*** *For every function* f *computable by a size-*s *classical circuit and for every input* x*, the encoding* $\mathsf{CEnc}(f, x; r)$ *is computable in time* $\mathrm{poly}(\lambda) \cdot s$*. Furthermore, the label functions* $\mathsf{lab}_{i,b}(r)$ *are computable in time* $\mathrm{poly}(\lambda)$*. The decoding and simulation algorithms* CDec *and* CSim *are also computable in time* $\mathrm{poly}(\lambda) \cdot s$*.*

- ***Computational Privacy.*** *For every polynomial* $t(\lambda)$*, there exists a negligible function* $\epsilon(\lambda)$ *such that the scheme is* $(t(\lambda), \epsilon(\lambda))$*-private against the class of size-*$t(\lambda)$ *classical (resp. quantum) distinguishers.*

- ***Locality.*** *Every output bit of* $\hat{f}(x; r) = \mathsf{CEnc}(f, x; r)$ *depends on at most* 4 *bits of* $(x, r, \mathsf{G}(r))$*. If the PRG* G *can be computed by a* $O(\log(\lambda))$*-depth circuit, then every output bit of* $\hat{f}(x; r)$ *can be made to depend on at most* 4 *bits of* (x, r)*, via non-black-box use of the PRG.*

We note that the locality and efficiency properties of the DRE scheme specified by Theorem 3.7 implies that the randomized encodings $\hat{f}(x; r)$ are computable by $\mathbf{NC}^0$ circuits that take as input x, the randomness r, and the output of the PRG G.

Remark 3.8. In the case of computational security, the RE scheme $(\mathsf{CEnc}, \mathsf{CDec}, \mathsf{CSim})$ may also depend on a security parameter λ. Since the security parameter will always be set and fixed in application, we do not explicitly point it out in our notation.

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

- **Correctness.** For all quantum states $(\boldsymbol{x}, \boldsymbol{q})$ and all randomness r, it holds that $(\mathsf{Dec}(\hat{F}(\boldsymbol{x}; r)), \boldsymbol{q}) = (F(\boldsymbol{x}), \boldsymbol{q})$.

- (t, ϵ)**-Privacy.** For all quantum states $(\boldsymbol{x}, \boldsymbol{q})$, we have

$$\left(\hat{F}(\boldsymbol{x}; r)\,,\ \boldsymbol{q}\right) \approx_{(t,\epsilon)} \left(\mathsf{Sim}(F(\boldsymbol{x}))\,,\ \boldsymbol{q}\right)$$

  where the state on the left-hand side is averaged over r. The case of $\epsilon = 0$ is called *perfect privacy*.

The encoding $\hat{F}$ is a *decomposable* QRE (DQRE) if there exists a quantum state $\boldsymbol{e}$ (called the *resource state of the encoding*), an operation $\hat{F}_{\mathsf{off}}$ (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 $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$,

$$\hat{F}(\boldsymbol{x}; r) = \left(\hat{F}_{\mathsf{off}}, \hat{F}_1, \hat{F}_2, \ldots, \hat{F}_n\right)(\boldsymbol{x}, r, \boldsymbol{e})$$

where the functions $\hat{F}_{\mathsf{off}}, \hat{F}_1, \ldots, \hat{F}_n$ act on disjoint subsets of qubits from $\boldsymbol{e}, \boldsymbol{x}$ (but can depend on all bits of r), each $\hat{F}_i$ acts on a single qubit $\boldsymbol{x}_i$, and $\hat{F}_{\mathsf{off}}$ does not act on any of the qubits of $\boldsymbol{x}$.

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 $\hat{F}(\boldsymbol{x})$ to denote the quantum state $\hat{F}(\boldsymbol{x}; r)$ when r is sampled from the uniform distribution. Furthermore, we say that the *quantum state* $\hat{F}(\boldsymbol{x})$ is the randomized encoding of the operation F and an input $\boldsymbol{x}$.

One can see that this definition of quantum randomized encoding is syntactically similar to Definition 3.2, with a couple differences. First, the correctness and privacy properties involves the pair $(\boldsymbol{x}, \boldsymbol{q})$. We refer the reader to Section 3.1 for the full explanation of the quantum random variable notation; but in short $(\boldsymbol{x}, \boldsymbol{q})$ represents a bipartite density matrix with an $\boldsymbol{x}$ part, and a $\boldsymbol{q}$ part, and these parts may be entangled. The $\boldsymbol{q}$ part is never acted upon by the decoder or simulator, but distinguishability is measured with respect to the encoding of $\boldsymbol{x}$ as well as $\boldsymbol{q}$, 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 a auxiliary quantum state $\boldsymbol{e}$ (that is independent of the input $\boldsymbol{x}$). The definition allows for any resource state $\boldsymbol{e}$, but in this paper we focus on decomposable QREs where the resource state $\boldsymbol{e}$ is a collection of EPR pairs, which is perhaps the most natural quantum analogue of a randomness string.

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 Definition 3.4.

Definition 4.2 (Universal QRE Schemes for Circuits)**.** Let $\mathcal{C}$ denote a class of general quantum circuits[17] and let $\mathcal{R}$ denote an equivalence relation over $\mathcal{C}$. An (t, ϵ)-*private and efficient* $\mathcal{R}$-*universal QRE scheme for the class* $\mathcal{C}$ is a tuple of polynomial-time quantum algorithms $(\mathsf{Enc}, \mathsf{Dec}, \mathsf{Sim})$ such that given a circuit $F \in \mathcal{C}$ (here we identify the circuit with the function it computes),

- **Efficient Encoding**. For all quantum inputs $\boldsymbol{x}$ and randomness r, $\mathsf{Enc}(F, \boldsymbol{x}; r)$ computes a quantum randomized encoding $\hat{F}(\boldsymbol{x}; r)$.
- **Correctness**. For all quantum states $(\boldsymbol{x}, \boldsymbol{q})$ and randomness r it holds that $(F(\boldsymbol{x}), \boldsymbol{q}) = (\mathsf{Dec}(c, \hat{F}(\boldsymbol{x}); r), \boldsymbol{q})$ where c denotes the equivalence class of F in $\mathcal{R}$.
- (t, ϵ)**-Privacy**. For all quantum states $(\boldsymbol{x}, \boldsymbol{q})$, we have

$$\left(\hat{F}(\boldsymbol{x}; r)\,,\ \boldsymbol{q}\right) \approx_{(t,\epsilon)} \left(\mathsf{Sim}(c, F(\boldsymbol{x}))\,,\ \boldsymbol{q}\right)$$

  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 $(\mathsf{Enc}, \mathsf{Dec}, \mathsf{Sim})$ is *decomposable* if the randomized encoding $\hat{F}(\boldsymbol{x})$ is decomposable and if the input encoding operations $\hat{F}_i$ only depend on the equivalence class of F in $\mathcal{R}$.

[17] See Section 3.2.4 for the definition of general quantum circuits.

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 if they have the same topology (see Section 3.2.4 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 DRE 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 $\kappa(d,s)$. Furthermore, suppose that the encoding of* CRE *can be computed by* $\mathbf{NC}^0$ *circuits, and that the scheme is (t,ϵ)-private with respect to quantum adversaries.*

Recursively define $\kappa_0 = O(1)$, $\kappa_i = \kappa(O(1), O(\kappa_{i-1}^2))$, and define $c_i = c(O(1), O(\kappa_{i-1}^2))$. (All the $O(\cdot)$'s refer to universal constants.)

Then there exists an efficient topologically-universal decomposable QRE scheme $\mathsf{QRE} = (\mathsf{Enc}, \mathsf{Dec}, \mathsf{Sim})$ *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 $\boldsymbol{x}$, the encoding* $\mathsf{Enc}(F, \boldsymbol{x}; r)$ *is computable by a* $\mathbf{QNC}^0_{\mathrm{f}}$ *circuit of size $O(c_d \cdot s)$. The* $\mathbf{QNC}^0_{\mathrm{f}}$ *encoding circuit takes as input a string of random bits r, the quantum input $\boldsymbol{x}$, and a collection of EPR pairs. Furthermore, the input encoding operations $\hat{F}_i$ can be computed by* $\mathbf{QNC}^0_{\mathrm{f}}$ *circuits of size $O(\kappa_d)$. The running time of* Dec *and* Sim *is $O(c_d \cdot s)$.*
- ***Classical Inputs.*** *If an input qubit $\boldsymbol{x}_i$ is classical, then the input encoding operation $\hat{F}_i$ is computable by a classical circuit.*
- ***Privacy.*** *The scheme* QRE *is (t', ϵ')-private where $t' = t - \mathrm{poly}(c_d) \cdot s$ and $s' = \epsilon \cdot s$. Here, s and d refer to the size and depth of the circuit being encoded.*

Remark 4.4. As mentioned in Remark 3.5, 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 and since the description is classical, the encoding of the input F also becomes classical. Furthermore, if the input $\boldsymbol{x}$ is classical as well, then the quantum part of the encoding $\hat{F}(\boldsymbol{x})$ is independent of both $F, \boldsymbol{x}$ and can be generated beforehand as a "resource state" that is given to the encoder.

The proof of Lemma 4.3 is presented in Sections 6 and 7. Specifically, Section 6 describes the encoding, decoding, and simulation procedures of our Quantum Garbled Circuits scheme. The correctness and privacy properties of the scheme are then analyzed in Section 7.

By instantiating CRE in Lemma 4.3 with the classical RE schemes from Theorem 3.6 and Theorem 3.7, respectively, the following theorems immediately follow.

Theorem 4.5 (Information Theoretic DQRE)**.** *There exists an efficient topologically-universal decomposable QRE scheme* $\mathsf{QRE} = (\mathsf{Enc}, \mathsf{Dec}, \mathsf{Sim})$ *with the following properties:*

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

- ***Classical Inputs.*** *If an input qubit $\boldsymbol{x}_i$ is classical, then the input encoding operation $\hat{F}_i$ is computable by a classical circuit.*

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

Proof. Plugging the properties of the DCRE scheme from Theorem 3.6 into the conditions of Lemma 4.3, we get $\kappa_i = \mathrm{poly}(2^{2^i})$, $c_i = \mathrm{poly}(2^{2^i})$ and $(t, \epsilon = 0)$-privacy. The result thus follows. □

Theorem 4.6 (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 QRE scheme $\mathsf{QRE} = (\mathsf{Enc}, \mathsf{Dec}, \mathsf{Sim})$, which implicitly depends on a security parameter λ, 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 $\boldsymbol{x}$, the encoding $\mathsf{Enc}(F, \boldsymbol{x}; r)$ is computable by a $\mathbf{QNC}^0_{\mathrm{f}}$ circuit of size $\mathrm{poly}(\lambda) \cdot s$. The $\mathbf{QNC}^0_{\mathrm{f}}$ encoding circuit takes as input a string of random bits r, the output $\mathsf{G}(r)$ of the PRG, the quantum input $\boldsymbol{x}$, and a collection of EPR pairs.[18] Furthermore, the input encoding operations $\hat{F}_i$ can be computed by $\mathbf{QNC}^0_{\mathrm{f}}$ circuits of size $\mathrm{poly}(\lambda)$. The running time of Dec and Sim is $\mathrm{poly}(\lambda) \cdot s$.*

- ***Classical Inputs.*** *If an input qubit $\boldsymbol{x}_i$ is classical, then the input encoding operation $\hat{F}_i$ 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 $(t'(\lambda, s), \epsilon'(\lambda, s))$-private with respect to quantum adversaries, where $t'(\lambda, s) = t(\lambda) - \mathrm{poly}(\lambda, s)$ and $\epsilon'(\lambda, s) = \epsilon(\lambda) \cdot s$ with s being the size of the circuit being encoded.*

Proof. Plugging the properties of the DCRE scheme from Theorem 3.7 into the conditions of Lemma 4.3, we get $\kappa_i = \mathrm{poly}(\lambda)$, $c_i = \mathrm{poly}(\lambda) \cdot s$. Since $\mathrm{poly}(c_d) \cdot s = \mathrm{poly}(\lambda, s)$, we get that the scheme is (t', ϵ')-private for the functions $t'(\lambda, s)$ and $\epsilon'(\lambda, s)$ specified in the Theorem statement. □

5 A New Zero-Knowledge Σ-Protocol for QMA

We present a 3-message zero-knowledge Σ-protocol for any **QMA** problem. Our construction generically uses QRE with certain properties, as well as quantum-secure classical commitment schemes. Both are instantiable under (flavors of) quantum-secure one-way functions (QRE only requires general one-way functions, and the commitment schemes we use can be achieved with injective one-way functions or from arbitrary one-way functions in the presence of a common random string).

[18] 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 $\mathbf{QNC}^0_{\mathrm{f}}$ that takes as input only $\boldsymbol{x}$, the randomness r, and EPR pairs.

5.1 Building Blocks

We start by going over the building blocks that are used in our protocol.

A QMA Problem with Almost-Perfect Soundness. Let $L = (L_{yes}, L_{no})$ be a promise problem in **QMA**. Let $V_L = \{V_{L,n}\}$ denote the corresponding **QMA** verifier, specified as a (uniform) family of polynomial-size circuits that takes as input $(x, \boldsymbol{w})$ where x is an instance of the language and $\boldsymbol{w}$ is a poly($|x|$)-sized quantum witness. If $x \in L_{yes}$, then $V_L(x, \boldsymbol{w})$ accepts with probability at least $1 - \mu(|x|)$ and if $x \in L_{no}$, $V_L(x, \boldsymbol{w})$ accepts with probability at most $\mu(|x|)$ where $\mu(\cdot)$ is a negligible function. We let $\mathcal{R}_L(x)$ denote the set of witnesses $\boldsymbol{w}$ on which V_L accepts with probability at least $1 - \mu(|x|)$.

A Quantum Randomized Encoding Scheme. Let QRE denote an efficient, computationally-secure quantum randomized encoding scheme satisfying the properties of Theorem 4.6. Since the QRE is decomposable and has classical encoding for classical inputs, we can assume that the scheme has the following structure: given a circuit F and input $(\boldsymbol{q}, c)$ where $\boldsymbol{q}$ is an n_1-qubit quantum state and c is an n_2-bit classical string, the encoding $\hat{F}(\boldsymbol{q}, c)$ can be written as

$$\left(\hat{F}_{\mathsf{off}}, \hat{F}_1, \ldots, \hat{F}_{n_1+n_2}\right)(\boldsymbol{q}, c, r, \boldsymbol{e})$$

where r is a uniformly random string, $\boldsymbol{e}$ is a collection of EPR pairs, $\hat{F}_{\mathsf{off}}$ acts on $(r, \boldsymbol{e})$ only, and $\hat{F}_{n_1+1}(c_1; r), \ldots, \hat{F}_{n_1+n_2}(c_{n_2}; r)$ are classical circuits that encode each bit of c separately. Thus for each $i \in [n_2]$ and r, we can define classical labels for each c_i:

$$\mathsf{lab}_{i,b}(r) = \hat{F}_{n_1+i}(b; r) \ .$$

Furthermore, we assume that for a fixed value of r, the operations $\hat{F}_{\mathsf{off}}$ and $\hat{F}_i$ can be implemented via polynomial-size unitary circuits, possibly using some additional zero ancillas.[19]

We now consider the quantum functionality that, for a fixed value of r, takes as input the quantum input $\boldsymbol{q}$ and a sequence of zero ancilla bits, and outputs the quantum state

$$\left(\hat{\boldsymbol{F}}_0, \{\mathsf{lab}_{i,b}\}_{i\in[n_2], b\in\{0,1\}}, 0\right) ,$$

where $\hat{\boldsymbol{F}}_0$ denotes the part output by $(\hat{F}_{\mathsf{off}}, \hat{F}_1, \ldots, \hat{F}_{n_1})$, and where 0 represents a sequence of zero qubits.

Since the encoding is unitary as we explained above, this functionality can be implemented by a unitary circuit U_r that first creates a number of EPR pairs $\boldsymbol{e}$ from the zero qubits, and then applies the encoding functions to compute $\hat{\boldsymbol{F}}_0$ and $\{\mathsf{lab}_{i,b}\}_{i,b}$. In other words, $U_r(\boldsymbol{q}, 0)$ is almost the same as the QRE encoding of $\boldsymbol{F}(\boldsymbol{q}, c)$, except for the classical input it outputs *all* labels, and uses zero ancillas for scratch space. We note that for a fixed $c \in \{0,1\}^{n_2}$, the state $(\hat{\boldsymbol{F}}_0, \{\mathsf{lab}_{i,c_i}\}_{i\in[n_2]})$ constitutes the encoding of $F(\boldsymbol{q}, c)$.

A Quantum-Secure Classical Commitment Scheme. We require a perfectly-binding non-interactive commitment scheme secure against quantum adversaries. Such a commitment scheme is defined as a polynomial-time function Com that takes as input a security parameter λ, an input message m and randomness s and outputs a string $c = \mathsf{Com}(1^\lambda, x; s)$ with the following properties:

[19] This is true for our QGC scheme and can be assumed without loss of generality. The reason, briefly, is that we can always consider purified versions of the $\hat{F}$ functions, and then instead of tracing out a part of the output, just encrypt it with a quantum one-time pad, using classical bits from a (possibly extended) r.

- (*Perfect Binding*). For all λ there does not exist $x_1 \neq x_2$ and s_1, s_2 such that $\mathsf{Com}(1^\lambda, x_1; s_1) = \mathsf{Com}(1^\lambda, x_2; s_2)$.
- (*Computational Hiding Against Quantum Adversaries*). For any sequence of $\{x_{1,\lambda}, x_{2,\lambda}\}_\lambda$, where $|x_{1,\lambda}| = |x_{2,\lambda}| = \mathrm{poly}(\lambda)$, it holds that the distributions $\mathsf{Com}(1^\lambda, x_{1,\lambda}; s_1)$ and $\mathsf{Com}(1^\lambda, x_{2,\lambda}; s_2)$ where s_1, s_2 are sampled uniformly cannot be distinguished by any polynomial-time quantum circuit with non-negligible advantage.

Such a commitment scheme follows from the existence of quantum-secure injective one-way functions, or in the common-random-string model from any one-way function. There are also explicit constructions from post-quantum assumptions (see, e.g., [JKPT12]).

Commitment schemes like this can be used in the following manner: to commit to a message x, the sender samples a randomness s and computes $c = \mathsf{Com}(1^\lambda, x; s)$, and sends c to the receiver. To reveal the message x, the sender sends (x, s) to the receiver. The receiver can verify that this is a valid commitment by checking that $c = \mathsf{Com}(1^\lambda, x; s)$.

For more background on commitment schemes see, e.g., [Gol06].

Quantum Teleportation. We recall the functionality of quantum teleportation. Let $\boldsymbol{e} = (\boldsymbol{e}_1, \boldsymbol{e}_2)$ denote m EPR pairs, where $\boldsymbol{e}_1$ denotes all "first-halves" of the EPR pairs, and $\boldsymbol{e}_2$ denotes all the "second halves". Suppose that Alice has $\boldsymbol{e}_1$ along with an m-qubit quantum state $\boldsymbol{w}$, and Bob has $\boldsymbol{e}_2$. Alice can teleport $\boldsymbol{w}$ to Bob by performing a measurement on $(\boldsymbol{w}, \boldsymbol{e}_1)$ to obtain measurement outcomes $(u, v) \in (\{0,1\}^m)^2$. The post-measurement outcome on Bob's side is $X^{u_1} Z^{v_1} \otimes \cdots \otimes X^{u_m} Z^{v_m}(\boldsymbol{w})$. Alice can then send (u, v) to Bob so he can undo the Pauli corrections.

5.2 Delayed-Input Zero-Knowledge Σ-Protocols

We now recall the definition of zero-knowledge Σ-protocols with the delayed-input property. We will use the following notation. Let P, V be a pair of interactive machines (interpreted as prover and verifier respectively). We let $\langle P(y_p), V(y_v)\rangle(x)$ denote the output of the verifier V after completing an interaction with P, in which P has private input y_p, V has private input y_v and they also have a common input x (the inputs y_p, y_v can be classical or quantum).

A pair (P, V) of a quantum polynomial-time "honest" prover P and an "honest" verifier V is a *quantum interactive proof system* for L if there exist numbers α (called the *completeness*) and β (called the *soundness*) such that $\alpha > \beta$ such that

- (*Completeness.*) If $x \in L_{yes}$ and $\boldsymbol{w} \in \mathcal{R}_L(x)$, then $\mathbb{P}[\langle P(\boldsymbol{w}), V\rangle(x) \text{ accepts}] \geq \alpha$.
- ((*Statistical) Soundness.*) If $x \in L_{no}$, then for any prover P^* (possibly computationally-unbounded) it holds that $\mathbb{P}[\langle P^*, V\rangle(x) \text{ accepts}] \leq \beta$.

Adaptive Soundness. For public-coin protocols (and in particular in our protocol) we can consider a stronger notion of *adaptive* soundness, in which the instance x is not specified ahead of time, but rather is produced by P^* in the end of the interaction. In this case it will be convenient for us to specify that the output of the verifier includes the accept/reject bit also the instance x produced by P^*. Under this convention, adaptive soundness is the requirement that for any adaptive adversary P^* it holds that

$$\mathbb{P}[\langle P^*, V\rangle = (\text{accept}, x) \wedge x \in L_{no}] \leq \beta \,.$$

Note that the winning event in this case is not necessarily efficiently recognizable.

Zero-Knowledge The proof system is furthermore *(computational) zero-knowledge* if there exists a a quantum polynomial-time simulator ZKSim such that for any malicious verifier V^* and for any asymptotic sequence of instances $x \in L_{yes}$, $\boldsymbol{w} \in \mathcal{R}_L(x)$ and quantum poly($|x|$)-qubit auxiliary input $\boldsymbol{y}$ it holds that the output state of $\mathsf{ZKSim}(V^*, x, \boldsymbol{y})$ and $\langle P(\boldsymbol{w}), V^*(\boldsymbol{y})\rangle(x)$ are computationally indistinguishable. We recall that the latter expression refers to the quantum output produced by V^* at the end of the interaction.

An interactive protocol is a Σ-*protocol* if it consists of a prover sending the first message $\boldsymbol{m}_1$, the verifier sending a uniformly random (classical) challenge m_2, the prover sending a response $\boldsymbol{m}_3$, and the verifier decides whether to accept or reject based on the transcript. We note that for an honest verifier the transcript is well defined even if the messages are quantum, since m_2 is a classical random string independent of $\boldsymbol{m}_1$.

Finally, we say that a zero-knowledge Σ-protocol has the *delayed-input property* if the prover only receives the instance x and a quantum witness $\boldsymbol{w}$ after it has sent the first message. In other words, the prover's first message $\boldsymbol{m}_1$ is computed independently of the instance x and witness $\boldsymbol{w}$.

5.3 Our Proof System

We present a zero-knowledge Σ-protocol $\langle P, V\rangle$ for L in Protocol 1. Note that the prover only gets the instance x and witness $\boldsymbol{w}$ right before generating the third message.

Parameters and Definitions. We assume that at the beginning all parties know the instance length n. To avoid clutter we set the security parameter λ to be equal to n. We let $m = m(n)$ denote the length of the witness required for instances of length n of L. We let F denote the following quantum circuit, that takes $n + 2m$ classical bits (partitioned into strings x, u, v of length n, m, m respectively) and m quantum bits as inputs. On input $(x, u, v, \tilde{\boldsymbol{w}})$ the circuit does the following. It first computes $\boldsymbol{w} = (X^{u_1}Z^{v_1} \otimes \cdots \otimes X^{u_m}Z^{v_m})(\tilde{\boldsymbol{w}})$, i.e. applies a quantum one-time pad, indicated by the vectors u, v to the quantum input. It then executes $V_L(x, \boldsymbol{w})$ and outputs the single-bit outcome of this computation.

Lemma 5.1 (Completeness)**.** *There exists a negligible function μ such that for all $x \in L_{yes}$ and $\boldsymbol{w} \in \mathcal{R}_L(x)$, the honest prover described in Protocol 1 runs in polynomial time given the input $(x, \boldsymbol{w})$, and is accepted by the verifier with probability at least $1 - \mu(|x|)$.*

Proof. If the prover behaves honestly, then with challenge $b = 0$, the verifier will be able to verify that the state $(\hat{\boldsymbol{F}}_0, \{\mathsf{lab}_{i,b}\}_{i\in[n+2m],b\in\{0,1\}})$ is indeed equal to $U(\boldsymbol{e}_2, 0)$; and with challenge $b = 1$, the verifier obtains the output qubit of the circuit $V_L(x, \boldsymbol{w})$ by applying the QRE decoding procedure (which we assume has perfect correctness). Thus if the maximum acceptance probability of $V_L(x, \boldsymbol{w})$ over all choices of $\boldsymbol{w}$ is $1 - \mu(|x|)$, then the acceptance probability of the verifier is $1 - \mu(|x|)$. This establishes the completeness property. □

We now prove statistical adaptive soundness for our protocol as stated in the next Lemma.

Lemma 5.2 (Statistical Adaptive Soundness)**.** *There exists a negligible function μ' such that for any adaptive prover P^*, it holds that*

$$\mathbb{P}[W] \leq 1/2 + \mu'(|x|)\,,$$

where W is the event where $\langle P^*, V\rangle = (accept, x) \wedge x \in L_{no}$.

Global Parameters: Instance length n, witness length $m = m(n)$.

Prover:

Generate m EPR pairs $\boldsymbol{e} = (\boldsymbol{e}_1, \boldsymbol{e}_2)$ for the purpose of quantum teleportation.

Sample a random string r and execute the circuit U_r on input $(\boldsymbol{e}_2, 0)$, where the 0 represents a sufficient number of ancilla zeroes. Let $\left(\hat{\boldsymbol{F}}_0, \{\mathsf{lab}_{i,b}\}_{i\in[n+2m],b\in\{0,1\}}\right)$ denote the outcome of this computation.

Sample random strings $s_{i,b}$ for all $i \in [n + 2m]$, $b \in \{0, 1\}$ and compute the commitment $c_{i,b} = \mathsf{Com}(1^\lambda, \mathsf{lab}_{i,b}; s_{i,b})$. Then, sample a random string s_0 and compute the commitment $c_0 = \mathsf{Com}(1^\lambda, r; s_0)$.

Send $\left(\hat{\boldsymbol{F}}_0, c_0, (c_{i,b})_{i\in[n+2m],b\in\{0,1\}}\right)$ to the verifier.

Verifier:

Send random bit b.

Prover (given $x, \boldsymbol{w}$):

If $b = 0$:

Open all commitments, i.e. send (r, s_0) and $(\mathsf{lab}_{i,b}, s_{i,b})_{i\in[n+2m],b\in\{0,1\}}$ to the verifier.

If $b = 1$:

Teleport the witness $\boldsymbol{w}$ into the "first halves" $\boldsymbol{e}_1$ of the EPR pairs $\boldsymbol{e}$ to obtain classical strings $(u, v) \in \{0, 1\}^m$.

Consider the concatenated string $z = (x, u, v)$. Open the commitments corresponding to z, i.e. send $(z_i, \mathsf{lab}_{i,z_i}, s_{i,z_i})_{i\in[n+2m]}$ to the verifier.

Verifier (given x):

If $b = 0$:

Check that all commitments are valid. If any of them are invalid, then reject.

Apply the inverse circuit U_r^{-1} to $(\hat{\boldsymbol{F}}_0, \{\mathsf{lab}_{i,b}\}_{i\in[n+2m],b\in\{0,1\}})$, and let $(\boldsymbol{e}_2', \boldsymbol{q})$ denote the output. Check that $\boldsymbol{q}$ is the all zeroes state. If so, then accept. Otherwise, reject.

If $b = 1$:

Check that $z = (x, u, v)$ for the instance x and some u, v. If not then reject.

Check that all commitment openings are valid. If any of them are invalid, then reject.

Decode the QRE $(\hat{\boldsymbol{F}}_0, \{\mathsf{lab}_{i,z_i}\}_{i\in[n+2m]})$ to obtain a single-qubit output; return the output of this evaluation.

Protocol 1: A zero-knowledge Σ-protocol for **QMA** problem L

In order to prove the lemma, we require the following two claims.

Claim 5.3. *Let $|\phi\rangle, |\phi_0\rangle, |\phi_1\rangle$ be unit vectors over some Hilbert space representing states of a quantum system. Assume there exist non-negative real values α_1, α_2 so that $|\phi\rangle = \alpha_1|\phi_0\rangle + \alpha_2|\phi_1\rangle$ (note that $|\phi_0\rangle, |\phi_1\rangle$ are not necessarily orthogonal, so $\alpha_1^2 + \alpha_2^2$ is not necessarily 1).*

Let M be some measurement operator defined over this Hilbert space. Let p, p_1, p_2 be the probability that the measurement M succeeds when the system is in state $|\phi\rangle, |\phi_0\rangle, |\phi_1\rangle$ respectively. Then $p \leq (\alpha_1 \sqrt{p_1} + \alpha_2 \sqrt{p_2})^2$.

Proof. The proof follows by triangle inequality. We define $|\tilde{\phi}_i\rangle = M|\phi_i\rangle$ for $i \in \{0, 1\}$ and note that by definition $p_i = \langle\tilde{\phi}_i|\tilde{\phi}_i\rangle$.

$$\begin{aligned} p &= \langle\phi|M^\dagger M|\phi\rangle \\ &= \alpha_1^2\langle\tilde{\phi}_0|\tilde{\phi}_0\rangle + \alpha_2^2\langle\tilde{\phi}_1|\tilde{\phi}_1\rangle + \alpha_1\alpha_2(\langle\tilde{\phi}_0|\tilde{\phi}_1\rangle + \langle\tilde{\phi}_1|\tilde{\phi}_0\rangle) \\ &\leq \alpha_1^2 p_1 + \alpha_2^2 p_2 + 2\alpha_1\alpha_2\sqrt{p_1 p_2} \\ &= \left(\alpha_1\sqrt{p_1} + \alpha_2\sqrt{p_2}\right)^2 . \end{aligned}$$

The claim thus follows. □

The following claim establishes a very weak form of soundness, namely it asserts that soundness $1/2 + \mathrm{negl}(n)$ holds when the $b = 0$ test is guaranteed to pass.

Claim 5.4. *For any adversarial strategy P^* for which $\mathbb{P}[W|b = 0] = 1$ it holds that $\mathbb{P}[W|b = 1] = \mathrm{negl}(n)$.*

Proof. Assume that P^* is such that $\mathbb{P}[W|b = 0] = 1$. Then the message $\boldsymbol{m}_1$ is guaranteed to be a properly generated QRE of the correct functionality F. It follows by the correctness of the QRE that such a circuit cannot accept an instance $x \in L_{no}$ except with negligible probability. The claim thus follows. □

We can now prove the soundness lemma.

Proof of Lemma 5.2. Let P^* be an adaptive possibly-cheating prover. Consider the point in time after P^* sent its first message $\boldsymbol{m}_1$ and let us consider the joint quantum state of P^* and $\boldsymbol{m}_1$, denote this state by $|\phi\rangle$. Assume without loss of generality that this state is pure (we can always add the purification of the state into the internal state of P^*. Assume without loss of generality that if P^* sends valid commitments (i.e. ones that can be opened), then it indeed opens them correctly upon challenge $b = 0$ (this can only increase the advantage of P^* and since it is computationally unbounded it can always find the openings).

We consider the (inefficient) measurement operator defined on this joint state in the following way.

1. Check (via brute force search) that all commitments in $\boldsymbol{m}_1$ are valid. If they are valid, let $r, \mathsf{lab}_{i,b}$ be their openings. If not then reject. Note that this is a projection since it simply accepts a subset of the possible commitments (in the computational basis).

2. Apply U_r^{-1} on $\left(\hat{\boldsymbol{F}}_0, \{\mathsf{lab}_{i,b}\}_{i,b}\right)$, where $\hat{\boldsymbol{F}}_0$ comes from $\boldsymbol{m}_1$ and $r, \mathsf{lab}_{i,b}$ are the openings of the commitments (which at this point are well defined). The outcome is a pair $(\boldsymbol{q}, \boldsymbol{e_2})$. Accept if $\boldsymbol{q} = 0$ and reject otherwise. Note that this part is a projection as well since it only involves applying a unitary followed by accepting a value in the computational basis.

We conclude that this measurement operator is a projection Π, and notice that this measurement corresponds to the event $W|b=0$. Therefore

$$\mathbb{P}[W|b=0] = \langle\phi|\Pi|\phi\rangle \,,$$

and denote this value by ϵ. If $\epsilon = 0$ then the proof is complete because the overall acceptance probability is at most $1/2$. From now on assume $\epsilon > 0$. Note that since Π is a projection it holds that

$$\langle\phi|(I-\Pi)|\phi\rangle = 1-\epsilon \,.$$

Let M denote the (not necessarily projective and inefficient) measurement that acts on the joint state of P^* and $\boldsymbol{m}_1$ as follows. It acts on the state of P^* to generate the third message m_3 and instance x, and then checks whether $x \in L_{no}$ and if so whether the verifier V accepts. Again by definition it holds that

$$\mathbb{P}[W|b=1] = \langle\phi|M^\dagger M|\phi\rangle \,.$$

Let us now define $|\phi_0\rangle = \Pi|\phi\rangle/\sqrt{\epsilon}$ and $|\phi_1\rangle = (I-\Pi)|\phi\rangle/\sqrt{1-\epsilon}$, so we can write $|\phi\rangle = \sqrt{\epsilon}|\phi_0\rangle + \sqrt{1-\epsilon}|\phi_1\rangle$. By Claim 5.3 (triangle inequality) we have

$$\mathbb{P}[W|b=1] \leq \left(\sqrt{\epsilon\langle\phi_0|M^\dagger M|\phi_0\rangle} + \sqrt{(1-\epsilon)\langle\phi_1|M^\dagger M|\phi_1\rangle}\right)^2 \tag{5.1}$$

$$\leq \left(\sqrt{\epsilon\langle\phi_0|M^\dagger M|\phi_0\rangle} + \sqrt{1-\epsilon}\right)^2 \,. \tag{5.2}$$

However, suppose that instead of starting with the joint prover-message state $|\phi\rangle$, we consider a different prover $\tilde{P}^*$ which starts with the state $|\phi_0\rangle$ instead. By construction this prover will pass the challenge $b=0$ with probability 1, because $\Pi|\phi_0\rangle = |\phi_0\rangle$. Therefore, by Claim 5.4 it follows that

$$\mathbb{P}[W_{\tilde{P}^*}|b=1] = \langle\phi_0|M^\dagger M|\phi_0\rangle \leq \mu(n) \,,$$

for some negligible function μ, because M corresponds to the test performed on challenge $b=1$.

Finally we can plug $\langle\phi_0|M^\dagger M|\phi_0\rangle \leq \mu(n)$ into Eq. (5.2) to obtain

$$\begin{aligned}\mathbb{P}[W|b=1] &\leq \left(\sqrt{\epsilon\langle\phi_0|M^\dagger M|\phi_0\rangle} + \sqrt{1-\epsilon}\right)^2 \\ &\leq \left(\sqrt{\epsilon\cdot\mu(n)} + \sqrt{1-\epsilon}\right)^2 \\ &= 1-\epsilon+2\mu'(n) \,,\end{aligned}$$

for some $\mu'(n) = O(\sqrt{\mu(n)})$.

Finally we can conclude that

$$\mathbb{P}[W] \leq \frac{1}{2}\epsilon + \frac{1}{2}\left(1-\epsilon+2\mu'(n)\right) \leq \frac{1}{2} + \mu'(n)$$

which completes the proof of the Lemma. □

Lemma 5.5 (Computational zero-knowledge)**.** *There exists a quantum polynomial-time simulator* ZKSim *satisfying the following: for all cheating verifiers V^*, for all asymptotic sequences of $(x, \boldsymbol{y}, \boldsymbol{w})$ where $x \in L_{yes}$, the state $\boldsymbol{y}$ is an arbitrary quantum state, and $\boldsymbol{w} \in \mathcal{R}_L(x)$ is a witness for x, we have that the output of $\mathsf{ZKSim}(V^*, x, \boldsymbol{y})$ is computationally indistinguishable from $\langle P(\boldsymbol{w}), V^*(\boldsymbol{y})\rangle(x)$.*

Let V^* denote the malicious verifier. In order to prove Lemma 5.5 we first analyze a "conditional" simulator ZKSim^0 that takes as input $(V^*, x, \boldsymbol{y})$, and outputs a quantum state as well as a flag indicating whether the simulator aborted. We show that the output of ZKSim^0 is, conditioned on not aborting, computationally indistinguishable from the view of the interaction $\langle P(\boldsymbol{w}), V^*(\boldsymbol{y})\rangle(x)$. The probability of aborting in the conditional simulator ZKSim^0 is (negligibly close to) $1/2$, but we can use Watrous's Rewinding Lemma [Wat09] to argue the existence of a quantum polynomial-time algorithm ZKSim that satisfies the conclusions of Lemma 5.5. In particular, we use the formulation of the Rewinding Lemma as presented in [BS20, Lemma 2.1]; the resulting algorithm ZKSim queries ZKSim^0 as a blackbox polynomially many times to amplify the success probability. The reason we need to use the Rewinding Lemma instead of simply just repeating ZKSim^0 until it doesn't abort is because of the quantum auxiliary input $\boldsymbol{y}$; running ZKSim^0 once and aborting may alter the state $\boldsymbol{y}$ significantly. Watrous's Rewinding Lemma gets around this issue.

Input: cheating verifier V^*, instance $x \in \{0,1\}^n$, auxiliary quantum state $\boldsymbol{y}$

Sample $t \in \{0,1\}$ uniformly at random.
if $t = 0$ **then**
    Execute the honest prover P to generate the first message $\boldsymbol{m}_1$ (note P needs no input for this).
    Run the cheating verifier $V^*(x, \boldsymbol{y})$ on the first message $\boldsymbol{m}_1$ to generate the challenge bit b. If $b \neq 0$, abort (i.e., output $(a, 0)$ where $a = 1$).
    Otherwise, continue simulating the honest prover P on challenge $b = 0$ to generate the third message $\boldsymbol{m}_3$.
end
else
    Generate m EPR pairs $\boldsymbol{e} = (\boldsymbol{e}_1, \boldsymbol{e}_2)$.
    Sample uniformly random $u, v \in \{0,1\}^m$.
    Run the simulator Sim of the QRE on input $|1\rangle\langle 1|$, with respect to the same circuit topology as F, to obtain an output $\left(\hat{\boldsymbol{F}}_0, \{\mathsf{lab}_i\}_{i\in[n+2m]}\right)$.
    For all $i \in [n+2m]$,$b \in \{0,1\}$, define $\mathsf{lab}_{i,b} = \mathsf{lab}_i$. Set $r = 0$.
    Execute the honest prover P, starting at Step 5 of Protocol 1, where the prover computes the commitments to r and $(\mathsf{lab}_{i,b})_{i,b}$. Let $\boldsymbol{m}_1$ denote the prover's first message.
    Run the cheating verifier $V^*(x, \boldsymbol{y})$ on the first message $\boldsymbol{m}_1$ to generate the challenge bit b. If $b \neq 1$, abort (i.e., output $(a, 0)$ where $a = 1$).
    Otherwise, execute Step 14 of the honest prover P in Protocol 1 to open the relevant commitments, which forms message $\boldsymbol{m}_3$. Note that $\boldsymbol{w}$ is note used by P in this step, therefore the simulator can perform it.
end
Continue simulating the cheating verifier V^* on the third message $\boldsymbol{m}_3$ to obtain a state $\boldsymbol{o}$, and output $(a, \boldsymbol{o})$ where $a = 0$.

Algorithm 2: The simulator ZKSim^0 for the zero-knowledge protocol

Lemma 5.6. *There exists a quantum polynomial-time conditional simulator* ZKSim^0 *satisfying the following: for all cheating verifiers* V^**, for all asymptotic sequences of* $(x, \boldsymbol{y}, \boldsymbol{w})$ *where* $x \in L_{yes}$*, the state* $\boldsymbol{y}$ *is an arbitrary quantum state, and* $\boldsymbol{w} \in \mathcal{R}_L(x)$ *is a witness for* x*, we have that the output of* $\mathsf{ZKSim}^0(V^*, x, \boldsymbol{y})$*, conditioned on not aborting, is computationally indistinguishable from* $\langle P(\boldsymbol{w}), V^*(\boldsymbol{y})\rangle(x)$*.*

Proof. The proof proceeds by a sequence of experiments (or hybrids). We keep track of the output distribution of the simulator and the rejection probability across experiments. Fix V^*, $\boldsymbol{y}$, x, $\boldsymbol{w}$ as in the Lemma statement.

- **Experiment** 0**.** This is the experiment of running the conditional simulator ZKSim^0 on input $(V^*, x, \boldsymbol{y})$ as presented in Protocol 2.
- **Experiment** 1**.** Modify the previous experiment by replacing Step 10 with the following: teleport $\boldsymbol{w}$ into $\boldsymbol{e_1}$ and let u, v be the (classical) outcome of the teleportation. By the properties of the teleportation the strings u, v are uniformly random and therefore this experiment produces an identical distribution to the previous one.
- **Experiment** 2**.** Now, consider Experiment 1 except we modify Step 11 of ZKSim^0. Instead of using the QRE simulator Sim, do the following: sample randomness r^* and evaluate the unitary U_{r^*} on input $(\boldsymbol{e_2}, 0)$ to generate the state $\left(\hat{\boldsymbol{F}}_0, (\mathsf{lab}^*_{i,b})_{i\in[n+2m],b\in\{0,1\}}\right)$. Set lab_i to be lab^*_{i,z_i} for $z = (x, u, v)$. By definition $F(x, u, v, \boldsymbol{e_2})$ computes $V_L(x, \boldsymbol{w})$, which by the completeness of the **QMA** verifier outputs $|1\rangle\langle 1|$ with probability $1 - \mathrm{negl}(n)$.

  By the privacy of QRE, the QRE encoding of circuit $F(x, u, v, \boldsymbol{e_2})$ is computationally indistinguishable from $\mathsf{Sim}(|1\rangle\langle 1|)$ when we marginalize over the randomness r^* and the labels $(\mathsf{lab}_{i,b} : b \neq z_i)$. Since the randomness r^* and the unused labels are not used anywhere else in the experiment, the output distribution of this experiment is computationally indistinguishable from that of the previous one.
- **Experiment** 3**.** Change Step 12 of ZKSim^0 so that $r = r^*$ and $\mathsf{lab}_{i,b} = \mathsf{lab}^*_{i,b}$. Since this only changes locations of the commitment that are never opened by the prover P, the hiding property of the commitment scheme guarantees that the views of V^* between Experiment 2 and Experiment 3 remain computationally indistinguishable.
- **Experiment** 4**.** Move the modified Step 10 (i.e. the teleportation of $\boldsymbol{w}$ into the EPR pairs) to be right before Step 15. This does not change anything in the simulation because none of the steps until Step 15 in Experiment 3 depend on teleportation outcomes (u, v).
- **Experiment** 5**.** Note that in Experiment 4, the steps up to and including receiving the bit b from V^* are identical between $t = 0$ and $t = 1$. Thus we can move these steps outside of the "if" statement, and before the sampling of t. The output of the Experiment is unchanged from Experiment 4.

We see that the output in Experiment 5 is computationally indistinguishable from that of $\mathsf{ZKSim}^0(x)$. Furthermore, in Experiment 5 the bit t is sampled after receiving the bit b and abort occurs if and only if $t \neq b$. It follows that the abort probability is $1/2$ in Experiment 5. Furthermore, conditioned on not aborting, the experiment is identical to that of the execution of V^* with $P(x, \boldsymbol{w})$.

It follows that Experiment 0 (which is to run ZKSim^0 on input $(V^*, x, \boldsymbol{y})$), the probability of abort is negligibly close to $1/2$ and the output conditioned on not aborting is indistinguishable from $\langle P(\boldsymbol{w}), V^*(\boldsymbol{y})\rangle(x)$. □

6 Quantum Garbled Circuits – Construction

In this section we present topologically-universal QRE scheme of Lemma 4.3, called the Quantum Garbled Circuits scheme and denoted by QGC. In particular, given a circuit that computes a quantum operation F, we show how to compute the encoding $\hat{F}$ in Section 6.3, how to decode the encoded value in Section 6.4, and how to simulate the randomized encoding in Section 6.5. We then prove the correctness and security of the scheme in Section 7.

In this paper we assume that quantum circuits F being encoded use the universal gate set $C_2 \cup \{T\}$. The only property of this gate set we use (other that the arity of the gates being bounded by a global constant) is the following: each p-qubit gate U_g of circuit F has the property that for any single-qubit Pauli unitaries $P_1, \ldots, P_p$, there exist single-qubit gates $R_1, \ldots, R_p$ from the PX group such that

$$U_g(P_1 \otimes P_2 \otimes \cdots \otimes P_p) = (R_1 \otimes R_2 \otimes \cdots \otimes R_p)U_g. \tag{6.1}$$

This property indeed holds for the $C_2 \cup \{T\}$ universal set as described in Sections 3.2.1 and 3.2.2.

A Building Block: Topologically-Universal Decomposable RE for Classical Circuits. As stated in Lemma 4.3, we assume the existence of an efficient topologically-universal and label-universal DCRE for classical circuits. We refer to this DCRE scheme as $\mathsf{CRE} = (\mathsf{CEnc}, \mathsf{CDec}, \mathsf{CSim})$. In our construction we use CRE as a generic building block, and different instantiations of CRE will result in quantum encoding scheme with different properties. As in the Lemma statement, we let $\kappa(\cdot,\cdot)$ and $c(\cdot,\cdot)$ be such that for functions f computable by size-s and depth-d classical circuits, the complexity of encoding f is $c(d,s)$ and the length of the labels is $\kappa(d,s)$. We let $\mathsf{CSim}_{\mathfrak{T}}$ and $\mathsf{CDec}_{\mathfrak{T}}$ denote the polynomial-time simulator and decoding procedures of CRE, respectively, for classical circuits with topology $\mathfrak{T}$.

6.1 Gadgets

In this section we introduce various gadgets that are used in our QGC scheme.

6.1.1 Teleportation Gadget

Let $\ell = (\ell_{b,a})_{a\in\{0,1\},b\in\{x,z\}}$ be a vector of strings of length κ, and let $s = (s_z, s_x), t = (t_z, t_x) \in \{0,1\}^2$. Define $\mathsf{TP}_{\ell,s,t}$ to be the unitary computed by the following circuit:

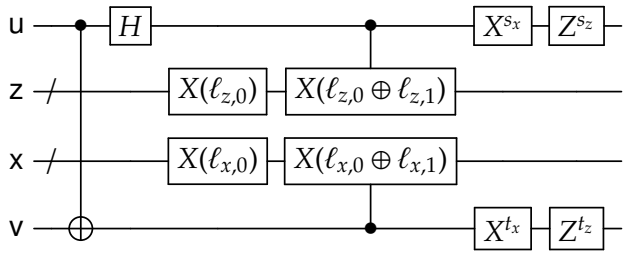


Figure 3: Teleportation gadget

Here, for a string $r \in \{0,1\}^\kappa$ the notation $X(r)$ denotes applying the tensor product of X gates acting on the i-th qubit whenever $r_i = 1$, and identity otherwise. The controlled $X(\ell_{z,0} \oplus \ell_{z,1})$ and $X(\ell_{x,0} \oplus \ell_{x,1})$ gates can also be seen as *fan-out* gates that are applied to the qubits indexed by $\ell_{z,0} \oplus \ell_{z,1}$ and $\ell_{x,0} \oplus \ell_{x,1}$, respectively, because it copies the control qubit into the target qubits simultaneously.

Thus the teleportation gadget is a $\mathbf{QNC}^0_f$ circuit. We refer to ℓ as the *teleportation labels* and s, t as the *randomization bits* of the *teleportation gadget* $\mathsf{TP}_{\ell,s,t}$.

Lemma 6.1. *Let* $\mathsf{u}, \mathsf{v}, \mathsf{u}'$ *denote qubit registers, and let* z, x *denote ancilla registers. For all* ℓ, s, t *and for all qubit states* $|\psi\rangle$ *we have*

$$\mathsf{TP}_{\ell,s,t} \underbrace{|\psi\rangle}_{\mathsf{u}} \otimes \underbrace{|0,0\rangle}_{\mathsf{zx}} \otimes \underbrace{|\mathsf{EPR}\rangle}_{\mathsf{vu'}} = \frac{1}{2} \sum_{d,e\in\{0,1\}} \underbrace{Z^{s_z} X^{s_x} |d\rangle}_{\mathsf{u}} \otimes \underbrace{Z^{t_z} X^{t_x} |e\rangle}_{\mathsf{v}} \otimes \underbrace{|\ell_{z,d}, \ell_{x,e}\rangle}_{\mathsf{zx}} \otimes \underbrace{X^e Z^d |\psi\rangle}_{\mathsf{u'}}$$

Proof. Since $\mathsf{TP}_{\ell,s,t}$ is unitary, it suffices to prove the Lemma when $|\psi\rangle = |c\rangle$ is a standard basis state. After the first CNOT, Hadamard, and $X(\ell_{z,0})$ and $X(\ell_{x,0})$ we have

$$\frac{1}{2} \sum_{d,a\in\{0,1\}} (-1)^{dc} \underbrace{|d\rangle}_{\mathsf{u}} \otimes \underbrace{|\ell_{z,0}\rangle}_{\mathsf{z}} \otimes \underbrace{|\ell_{x,0}\rangle}_{\mathsf{x}} \otimes \underbrace{|a \oplus c, a\rangle}_{\mathsf{vu'}} .$$

After the controlled X's we have

$$\frac{1}{2} \sum_{d,a\in\{0,1\}} (-1)^{dc} \underbrace{|d\rangle}_{\mathsf{u}} \otimes \underbrace{|\ell_{z,d}\rangle}_{\mathsf{z}} \otimes \underbrace{|\ell_{x,a\oplus c}\rangle}_{\mathsf{x}} \otimes \underbrace{|a \oplus c, a\rangle}_{\mathsf{vu'}} .$$

Relabeling $e = a \oplus c$ and re-arranging, we get

$$\frac{1}{2} \sum_{d,e\in\{0,1\}} \underbrace{|d\rangle}_{\mathsf{u}} \otimes \underbrace{|e\rangle}_{\mathsf{v}} \otimes \underbrace{|\ell_{z,d}, \ell_{x,e}\rangle}_{\mathsf{zx}} \otimes \underbrace{X^e Z^d |c\rangle}_{\mathsf{u'}} .$$

Applying the $X^{s_x}, X^{t_x}, Z^{s_z}, Z^{t_z}$ gates, we obtain the desired Lemma statement.

□

6.1.2 Correction Gadget

In our QGC scheme, the encoding consists of a collection of EPR pairs that are connected via gates of the circuit F and teleportation gadgets as described above. However, the teleportation gadget induces a *correction* on the output (as demonstrated by Lemma 6.1) that, ostensibly, needs to be fixed before applying the next gate and teleportation operation – but since the encoding is performed in parallel, it is up to the evaluator to perform the corrections *after* the gates and teleportation gadgets are appled. The next Lemma demonstrates that the desired operation (correction, then teleportation) is equivalent a sequence of circuits $\Lambda_1, \Lambda_2, \Lambda_3$ where the encoder can apply Λ_1, and the decoder can apply Λ_2 and Λ_3. Before stating the Lemma, we first have to describe the *randomization group*.

Randomization Group. In the following Lemma, the circuits $\Lambda_1, \Lambda_2, \Lambda_3$ will act on the same registers that the teleportation gadget acts on (single-qubit registers u, v and κ-qubit registers z, x), as well as ancilla registers b that is $(\kappa+1)^2$ qubits wide. The individual qubits of registers b are indexed by $(i, j) \in \{0, 1, \ldots, \kappa\}^2$. An important conclusion of the Lemma is that the circuit Λ_2 computes a unitary belonging to the *randomization group* $\mathcal{R}_\kappa$, which consists of depth-one circuits that are tensor products of

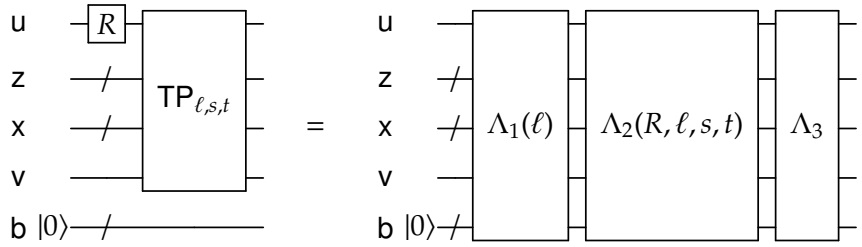
- Two-qubit Clifford gates acting on the pair of qubits $(\mathsf{b}_{ij}, \mathsf{b}_{ji})$ for all $i < j$, and
- Single-qubit Clifford gates acting on all other qubits.

The set $\mathscr{R}_\kappa$ indeed forms a group under the natural gate multiplication operation. It has finite order with $\exp(O(\kappa^2))$ elements, and a uniformly random element of $\mathscr{R}_\kappa$ can be sampled via an $\mathbf{NC}^0$ circuit that is given a uniformly random bitstring as input (essentially, the single- and two-qubit Clifford gates are chosen independently in parallel).

Lemma 6.2. *Let ℓ be κ-bit teleportation labels, and let $s, t \in \{0,1\}^2$ be randomization bits. Let R be an element from the single-qubit PX group. Then there exist*

- $\mathbf{QNC}^0_f$ *circuits $\Lambda_1(\ell)$, Λ_3 and*
- *A depth-one Clifford circuit $\Lambda_2(R, \ell, s, t)$ that computes a unitary in $\mathscr{R}_\kappa$*

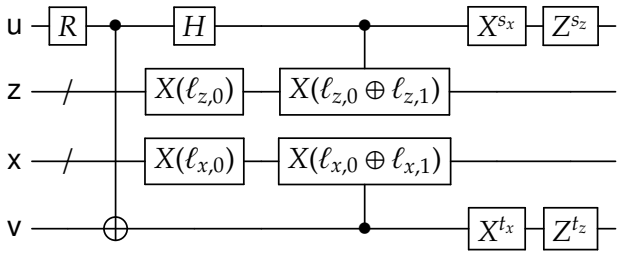
all acting on registers $\mathsf{u}, \mathsf{z}, \mathsf{x}, \mathsf{v}, \mathsf{b}$ *such that the following circuit identity holds:*



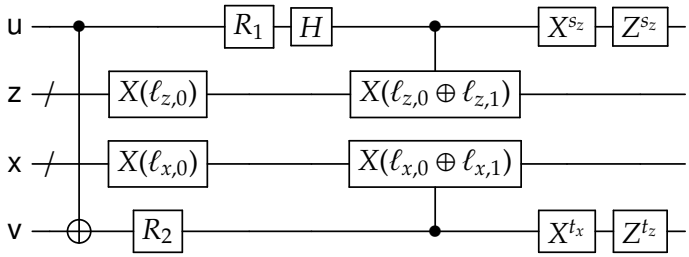
(6.2)

Furthermore, the description of $\Lambda_2(R, \ell, s, t)$ can be computed by a $\mathbf{NC}^0$ circuit of size $O(\kappa^2)$.

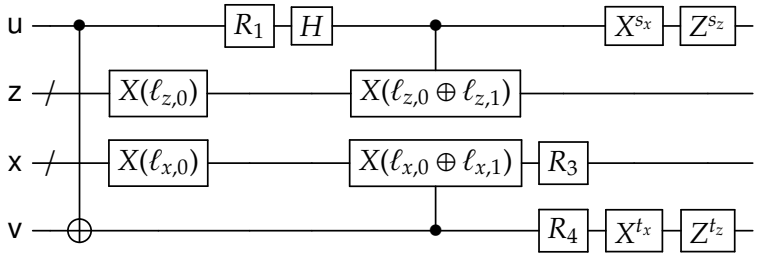
Proof. We can write the left-hand side of (6.2) (omitting the ancilla registers b) as



Since R is a PX group element, we can propagate it past the first CNOT gate to get the equivalent circuit



for some single-qubit gates R_1 and R_2 that are PX group elements. Propagating R_2 past the bottom fan-out operation yields another equivalent circuit

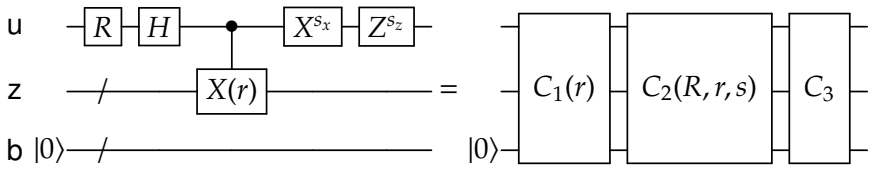


(6.3)

where R_3 is a single-qubit PX group element, and R_4 is a tensor product of single-qubit PX group elements.

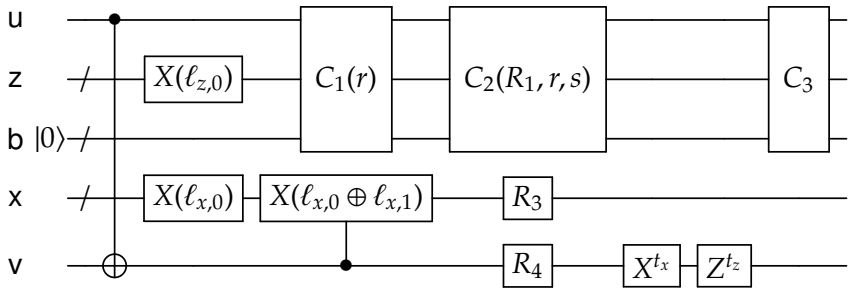
Next, we use the following Proposition in order to "push" the R_1 gate through the circuit as far to the right as possible:

Proposition 6.3. *For all single-qubit PX gates R, $s = (s_z, s_x) \in \{0,1\}^2$, and strings $r \in \{0,1\}^\kappa$ there exist $\mathbf{QNC}^0_f$ circuits $C_1(r)$ and C_3 and a depth-one Clifford circuit $C_2(R, r, s)$ in the randomization group $\mathscr{R}_\kappa$ such that the following circuit identity holds:*



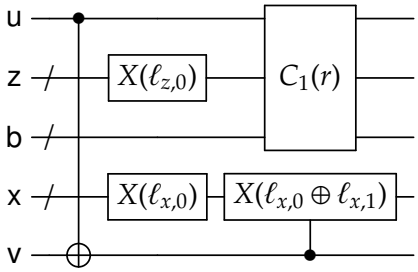
Here, the register b *consists of $(\kappa+1)^2$ qubits. Furthermore, descriptions of the circuits $C_1(r)$ and $C_2(R, r, s)$ can be computed by $\mathbf{NC}^0$ circuits of size $O(\kappa^2)$.*

The proof of this Proposition is deferred to Appendix C. Letting $r = \ell_{z,0} \oplus \ell_{z,1}$, we thus get that Circuit (6.3) is equivalent to



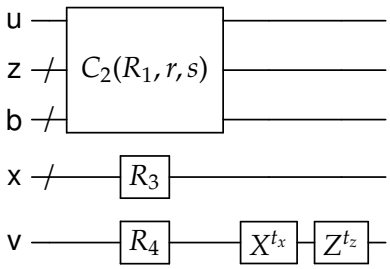
(6.4)

Define $\Lambda_1(\ell)$ to be the circuit



Since $C_1(r)$ is a $\mathbf{QNC}^0_f$ circuit, so is $\Lambda_1(\ell)$.

Define $\Lambda_2(R, \ell, s, t)$ to be the circuit



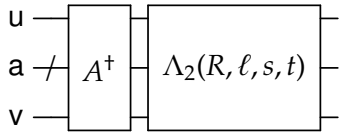
Since $C_2(R_1, r, s)$ is a depth-one Clifford circuit in the randomization group $\mathscr{R}_\kappa$, R_3 is a tensor product of single-qubit Clifford gates, and R_4 is a single-qubit Clifford gate, it follows that $\Lambda_2(R, \ell, s, t)$ is

also a member of $\mathcal{R}_\kappa$. The fact that a description of $\Lambda_2(R, \ell, s, t)$ can be computed by an $\mathbf{NC}^0$ circuit follows from the fact that $C_2(R_1, r, s)$ and R_1, R_2, R_3, R_4 can all be computed via $\mathbf{NC}^0$ circuits (given R, ℓ, s, t as input).

Finally, define Λ_3 to be the circuit C_3, which is a $\mathbf{QNC}^0_f$ circuit. This establishes the Lemma. □

Remark 6.4. Henceforth we will abbreviate the tuple of registers $(\mathsf{z}, \mathsf{x}, \mathsf{b})$ as register a. We call the qubits in this register "ancilla qubits".

The Correction Gadget. Let κ be a positive integer and let A be a unitary from the randomization group $\mathcal{R}_\kappa$, let R be a single-qubit PX unitary, let ℓ be κ-bit teleportation labels, and let $s = (s_x, s_z), t = (t_x, t_z) \in \{0,1\}^2$ be randomization bits. Define the *κ-correction gadget unitary* $\mathsf{Corr}_{A,R,\ell,s,t}$ to be the unitary computed by the following circuit:



Note that since $A \in \mathcal{R}_\kappa$, it follows that the unitary $\mathsf{Corr}_{A,R,\ell,s,t} \in \mathcal{R}_\kappa$. Thus a canonical representation of a correction gadget $\mathsf{Corr}_{A,R,\ell,s,t}$, denoted by $\widehat{\mathsf{Corr}}_{A,R,\ell,s,t}$, is an ordered list of single- and two-qubit Clifford gates on the respective qubits.

Furthermore, then for every κ-bit teleportation labels ℓ, randomization bits $s, t \in \{0,1\}^2$, and single-qubit R from the PX group, for a uniformly random A drawn from $\mathcal{R}_\kappa$, the correction gadget is $\widehat{\mathsf{Corr}}_{A,R,\ell,s,t}$ is uniformly distributed over the set of all κ-correction gadgets.

6.2 Encoding a Single Gate

We now present the *gate encoding* unitary $\mathsf{GateEnc}_{g,r,A,\ell,s,t}$ in Algorithm 3. This is used by QGC to encode each gate of the circuit. In what follows, we let $g \in \mathcal{B}$ denote a "placeholder gate" in the circuit topology $\mathcal{T}$, and let U_g denote a specific unitary for the gate.

The unitary $\mathsf{GateEnc}_{g,r,A,\ell,s,t}$ is a function of a p-qubit gate U_g, a string r (which represents the randomness used by CRE), randomization unitaries $A = (A_j)_{j\in[p]}$ with $A_j \in \mathcal{R}_{\kappa_j}$ for some integer κ_j, strings $\ell = (\ell_{j,b,a})_{j\in[p],b\in\{x,z\},a\in\{0,1\}}$ with $\ell_{j,b,a} \in \{0,1\}^{\kappa_j}$ (which represents the teleportation labels), and randomization bits $s = (s_{j,b})_{j\in[p],b\in\{0,1\}}$, $t = (t_{j,b})_{j\in[p],b\in\{0,1\}}$. As described at the beginning of this section, we assume that the gate U_g satisfes the property described in Equation (6.1).

At a high level, the gate encoding produces a quantum and a classical part. The quantum part is comprised of input qubits in registers $\mathsf{u}_1, \ldots, \mathsf{u}_p$, ancillas in registers $\mathsf{a}_1, \ldots, \mathsf{a}_p$, and target qubits in registers $\mathsf{v}_1, \ldots, \mathsf{v}_p$. These target qubits are each initialized as part of an EPR pair. The gate U_g is first applied to the input registers u. Then the the circuit $\Lambda_1(\ell_j)$ from Lemma 6.2 are applied to registers $(\mathsf{u}_j, \mathsf{a}_j, \mathsf{v}_j)$ for $j \in [p]$, and then the randomization unitaries A_j are applied to registers $(\mathsf{u}_j, \mathsf{a}_j, \mathsf{v}_j)$ for $j \in [p]$.

Since we think of the input qubits as having been previously teleported, the input qubits will have X and Z corrections on them, and applying the gate U_g will incur further corrections. The evaluator will have to fix these corrections; they will use the classical part of the gate encoding to do this, which is a classical randomized encoding of a *correction function*, which we describe next.

6.2.1 Correction Functions and Their Randomized Encoding

Consider a vector $\kappa = (\kappa_j)_{j\in[p]}$ of label lengths. Define the classical *correction function* f_κ, which on input
$(z_1, x_1, \ldots, z_p, x_p, U_g, A, \ell, s, t)$ outputs a tuple of quantum circuits $(\widehat{\mathsf{Corr}}_1, \widehat{\mathsf{Corr}}_2, \ldots, \widehat{\mathsf{Corr}}_p)$ where $\widehat{\mathsf{Corr}}_j = \widehat{\mathsf{Corr}}_{A_j,R_j,\ell_j,s_j,t_j}$ for $j \in [p]$ are the correction gadgets defined in Section 6.1.2. The unitaries $R_1, R_2, \ldots, R_p$ are the single-qubit PX unitaries satisfying

$$U_g(Z^{z_1}X^{x_1} \otimes \cdots \otimes Z^{z_p}X^{x_p}) = (R_1^\dagger \otimes \cdots \otimes R_p^\dagger)U_g \ .$$

The unitaries $R_1, \ldots, R_p$ are well-defined because U_g comes from our universal set of gates (two-qubit Clifford and T gates). The correction gadgets $\widehat{\mathsf{Corr}}_j$ are specified using their canonical representation (i.e., a tensor product of single- and two-qubit Clifford gates).

We now describe a classical circuit to compute f_κ, by first describing the circuit to compute $\widehat{\mathsf{Corr}}_j$ for a single j. As discused in Section 6.1.2, there is a constant-depth circuit of size $O(\kappa_j^2)$ to compute $\widehat{\mathsf{Corr}}_j$, and this circuit is a function of $R_j, A_j, \ell_j, s_j, t_j$. The unitary R_j is itself a function of U_g and the tuple $(z_1, x_1, \ldots, z_p, x_p)$. Since the universal gate set used in this paper has arity bounded by 2 and has a constant number of elements, there is a constant-sized circuit that computes R_j. Composing this circuit with the circuit for $\widehat{\mathsf{Corr}}_j$, we get a constant-depth circuit of size $O(\kappa_j^2)$ that takes input $g, z_1, x_1, \ldots, z_p, x_p, A_j, \ell_j, s_j, t_j$.

Putting everything together, we have a classical $\mathbf{NC}^0$ circuit, whose depth will be denoted a universal constant d_{corr} and whose size is $c_\kappa = O(\sum_j \kappa_j^2)$ that computes f_κ. Note that the topology of this circuit only depends on the vector κ; call this topology $\mathfrak{T}_\kappa$.

Now, consider the encoding $\hat{f}_\kappa$ of f_κ with respect to CRE, the DCRE scheme for classical circuits that we use as a blackbox. Since the scheme is decomposable, we have that $\hat{f}_\kappa(z_1, x_1, \ldots, z_p, x_p, U_g, A, \ell, s, t; r)$ consists of an offline part $\hat{f}_{\kappa,\mathsf{off}}(r)$ that only depends on f_κ, and an online part which are labels for each input $z_1, x_1, \ldots, z_p, x_p, U_g, A, \ell, s, t$. Let $\mathsf{lab}_\kappa(U_g, A, \ell, s, t; r)$ denote the set of labels encoding the inputs U_g, A, ℓ, s, t.

Thus one can consider, for a fixed g, A, ℓ, s, t the correction function $f_{g,A,\ell,s,t}(z_1, x_1, \ldots, z_p, x_p) = f_\kappa(z_1, x_1, \ldots, z_p, x_p, U_g, A, \ell, s, t)$. The randomized encoding $\hat{f}_\kappa$ is also a randomized encoding of $\hat{f}_{g,A,\ell,s,t}$, where now the offline part $\hat{f}_{g,A,\ell,s,t,\mathsf{off}}(r)$ consists of both $\hat{f}_{\kappa,\mathsf{off}}(r)$ and $\mathsf{lab}_\kappa(U_g, A, \ell, s, t; r)$. The online part are labels for $z_1, x_1, \ldots, z_p, x_p$; for each $j \in [p]$, $b \in \{x, z\}$, and $a \in \{0, 1\}$, let $\mathsf{lab}_\kappa(j, b, a; r)$ denote the label of the input variable b_j when it takes value a, when the DCRE randomness is r and the topology of the circuit is $\mathfrak{T}_\kappa$.

6.2.2 The Gate Encoding Unitary

We now present the gate encoding unitary $\mathsf{GateEnc}_{g,r,A,\ell,s,t}$. It acts on registers $\mathsf{u} = (\mathsf{u}_1, \ldots, \mathsf{u}_p)$ (which represents the input qubits to the gate U_g), $\mathsf{a} = (\mathsf{a}_1, \ldots, \mathsf{a}_p)$ (which represent the ancillas qubits for the teleportation and correction gadgets), $\mathsf{v} = (\mathsf{v}_1, \ldots, \mathsf{v}_p)$ (which represents the entrance of connecting EPR pairs), and c (which represents a register to hold the classical randomized encoding of the correction gadget). In the description of the protocol, the unitary $\Lambda_1(\ell_j)$ is given by Lemma 6.2.

// Compute the quantum part of the QRE

1 Apply U_g to registers $(\mathsf{u}_1, \ldots, \mathsf{u}_p)$

2 Apply $\Lambda_1(\ell_j)$ to registers $(\mathsf{u}_j, \mathsf{a}_j, \mathsf{v}_j)$ for all $j \in [p]$

3 Apply A_j to registers $(\mathsf{u}_j, \mathsf{a}_j, \mathsf{v}_j)$ for all $j \in [p]$

// Compute the classical part of the QRE

4 Compute classical randomized encoding of the correction function $f_{g,A,\ell,s,t}$ as defined in Section 6.2.1, and let $\hat{f}_{g,A,\ell,s,t,\mathsf{off}}(r)$ denote the offline part of the randomized encoding using randomness r. Store the string $\hat{f}_{g,A,\ell,s,t,\mathsf{off}}(r)$ in the register c.

Protocol 3: Gate encoding operation $\mathsf{GateEnc}_{g,r,A,\ell,s,t}$

The quantum part of the gate encoding is illustrated in Figure 4. There, Λ_1 denotes the tensor product of $\Lambda_1(\ell_1), \ldots, \Lambda_p(\ell_p)$, and A denotes the tensor product of $A_1, \ldots, A_p$.

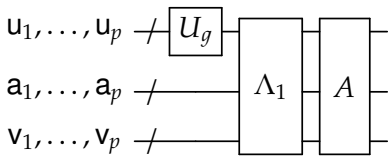


Figure 4: The quantum part of the gate encoding GateEnc

Complexity of the Gate Encoding. The gate encoding consists of a quantum part and a classical part. The quantum part is applying the gate U_g, applying the Λ_1 circuits, and then applying the randomizers A_j. Since the Λ_1 are $\mathbf{QNC}^0_{\mathrm{f}}$ circuits, and the randomizers is just a single layer of single- and two-qubit Clifford gates, the quantum encoding can be computed in $\mathbf{QNC}^0_{\mathrm{f}}$.

The classical part is computing the classical randomized encoding of the correction function $f_{g,A,\ell,s,t}$, which has topology $\mathfrak{T}_{\kappa_g}$. The complexity of encoding $f_{g,A,\ell,s,t}$ is inherited from the classical RE used – if the encoding of CRE can be computed via a $\mathbf{NC}^0$ circuit, then the entire gate encoding procedure can be computed in $\mathbf{QNC}^0_{\mathrm{f}}$.

6.3 Encoding a Circuit and Input

We now describe the encoding algorithm Enc of the QGC scheme. It takes as input a quantum circuit $F = (\mathcal{T}, \mathcal{G})$ and quantum input $\boldsymbol{y}$, and outputs the encoding $\hat{F}(\boldsymbol{y}; J)$ where J is a uniform random string.[20] We let n denote the number of qubits of $\boldsymbol{y}$. The encoding is decomposable, so the encoding consists of

$$\hat{F}(\boldsymbol{y}; J) = (\hat{F}_{\mathsf{off}}, \hat{\boldsymbol{y}}_1, \ldots, \hat{\boldsymbol{y}}_n)$$

where $\hat{F}_{\mathsf{off}}$ is the offline encoding of $\hat{F}$ and $\hat{\boldsymbol{y}}_i = \hat{F}_i(\boldsymbol{y}_i)$ is the encoding of the i-th qubit of $\boldsymbol{y}$.

Let $\mathcal{Z}$ and $\mathcal{T}$ be the zero input qubits and discarded output qubits of topology $\mathcal{T}$, respectively. For simplicity we assume that $\mathcal{Z}$ is empty (the zero inputs can be incorporated into $\boldsymbol{y}$).

The offline encoding is presented in Algorithm 4 and the input encoding is presented in Algorithm 5. First, we discuss details about the label lengths, the ancillary randomness and quantum random variables used in the encoding.

[20] We henceforth use $\boldsymbol{y}$ instead of $\boldsymbol{x}$ to denote the input to the function F being encoded; this is to disambiguate it from the qubits in register x used by the teleportation and correction gadgets.

Label Lengths. Our quantum randomized encoding contains a classical encoding of a correction function for every gate g in the circuit F. To keep track of the lengths of the labels and randomness required, for every wire $w \in \mathcal{W}$ we let κ_w denote the label lengths for encoding each wire w, and for every gate $g \in \mathcal{G}$ we let c_g denote the size of the CRE encoding of the correction function associated with g.

We recursively specify the label lengths κ_w and encoding sizes c_g, starting from the end of the circuit. For all output wires $w \in O$, define $\kappa_w = 1$. Then, for every gate $g \in \mathcal{G}$ whose output wires have label lengths defined so far, let κ_g denote the vector $\kappa_g = (\kappa_{w'} : w' \in \mathtt{outwire}(g))$, which has the label lengths for all the output wires of g. Recall from Section 6.2.1 that classical circuits with topology $\mathfrak{T}_{\kappa_g}$ have depth d_{corr} (which is a universal constant) and size $\sigma_g = O(\sum_j \kappa_j^2)$ where κ_j is the label length of the j-th output wire. Thus the CRE encoding of a correction function $f_{g,A,\ell,s,t}$ has complexity $c_g = c(d_{\text{corr}}, \sigma_g)$ and label lengths $\kappa_g = \kappa(d_{\text{corr}}, \sigma_g)$, where $c(\cdot,\cdot)$ and $\kappa(\cdot,\cdot)$ are given in the statement of Lemma 4.3. For every incoming wire $w \in \mathtt{inwire}(g)$, let $\kappa_w = \kappa_g$. We then recurse on the next layer of gates.

The following observations will be useful for our analysis down the line. First, we note that the encoding complexities c_g and the label lengths κ_w only depend on the topology $\mathcal{T}$ of F, and not on the specific unitaries of the gates. Second, as a recursive argument shows, if d is the depth of the quantum circuit F then it holds that $c_g \le c_d$ and $\kappa_g \le \kappa_d$, where κ_d, c_d are given in the statement of Lemma 4.3.

Quantum Registers. We now specify the registers in the encoding.

- $\mathsf{y} = (\mathsf{y}_1, \ldots, \mathsf{y}_n)$ is an n-qubit register that initially stores the input state $\boldsymbol{y}$.
- $\mathsf{e}^w = (\mathsf{e}^w_1, \mathsf{e}^w_2)$ is a two-qubit register for every wire $w \in \mathcal{W}$, initialized with $|\mathsf{EPR}\rangle$.
- $\mathsf{a}^w = (\mathsf{z}^w, \mathsf{x}^w, \mathsf{b}^w)$ is a $2\kappa_w + (\kappa_w + 1)^2$ qubit register for every wire $w \in \mathcal{W}$, initialized with zeroes. Some of these qubits are used to store teleportation labels, and others are used as part of the correction gadget (see Section 6.1.2 for details).
- c^g is a c_g-qubit register for every gate $g \in \mathcal{G}$, initialized with zeroes. These are used to store the description of the CRE encoding of the correction functions (see Section 6.2 for details).
- d^w is a $4\kappa_w$-qubit register for every non-traced-out wire $w \in O \setminus \mathcal{T}$, initialized with zeroes. These qubits store the "label dictionary" for the output wires (i.e. they store all possible labels for the output wires), so the evaluator can decode the output.

Each part of the encoding $\hat{F}_{\mathsf{off}}, \hat{F}_1, \ldots, \hat{F}_n$ access disjoint subsets of registers.

Classical Randomness. We now specify the classical randomness J that is used by the encoding. It consists of the following random strings:

- For every gate $g \in \mathcal{G}$, the random string r^g is a uniformly random string of length c_g. The randomness is used for the CRE encodings of the correction functions.
- For every gate $g \in \mathcal{G}$, the random string A^g is a sequence $(A^w)_{w \in \mathtt{outwire}(g)}$ where for each output wire w of g, the string A^w is a uniformly random element of the randomization group $\mathcal{R}_{\kappa_w}$. These randomizers are used in the encoding of each gate of the circuit.

- For every wire $w \in \mathcal{W}$, s^w is a random pair of bits $(s_z^w, s_x^w) \in \{0,1\}^2$, and t^w is a random pair of bits $(t_z^w, t_x^w) \in \{0,1\}^2$. These values are used to randomize the teleportation measurements (see Section 6.1.2).
- For every output wire $w \in O$, o^w is a random pair of bits (o_z^w, o_x^w). These values are used as labels for the output wires.

Because the randomness is classical, it can be copied and thus each part of the encoding $\hat{F}_{\mathsf{off}}, \hat{F}_1, \ldots, \hat{F}_n$ has access to the entire randomness J.

```
// Set up the labels
for w ∈ 𝒲 do
    if w ∉ O then
        Let g, j be such that wire w is the j-th input wire of gate g.
        Compute the labels ℓ^w = (ℓ^w_{b,a})_{b∈{x,z},a∈{0,1}} where ℓ^w_{b,a} = lab_{κ_g}(j, b, a; r^g) and lab_{κ_g} is
          the label function corresponding to the CRE encoding of a circuit with topology
          𝔗_{κ_g}.
    end
    else if w ∈ O then
        Let ℓ^w_{b,0} = o^w_b and ℓ^w_{b,1} = o^w_b ⊕ 1 for b ∈ {x, z}.
    end
end
// Store the "label dictionary" for output wires (ones not traced out)
for w ∈ O \ 𝒯 do
    Write the labels (ℓ^w_{z,0}, ℓ^w_{z,1}, ℓ^w_{x,0}, ℓ^w_{x,1}) into the register d^w.
end
// Encode each gate
for g ∈ 𝒢 do
    Let u^g = (e_2^v)_{v∈inwire(g)}, v^g = (e_1^w)_{w∈outwire(g)}, and a^g = (a^w)_{w∈outwire(g)}.
    Let ℓ^g = (ℓ^w)_{w∈outwire(g)}, s^g = (s^w)_{w∈outwire(g)}, and t^g = (t^w)_{w∈outwire(g)}.
    Apply GateEnc_{g,r^g,A^g,ℓ^g,s^g,t^g} to registers (u^g, a^g, v^g, c^g).
end
```

Protocol 4: The encoding of the offline part $\hat{F}_{\mathsf{off}}$ of $\hat{F}$.

```
// Set up the labels
Let w denote the i-th input wire of F, and let y_i denote the i-th input qubit, and let g, j be
  such that wire w is the j-th input wire of gate g.
Compute the labels ℓ^w = (ℓ^w_{b,a})_{b∈{x,z},a∈{0,1}} where ℓ^w_{b,a} = lab_{κ_g}(j, b, a; r^g).
// Teleport the input qubit into the encoding
Apply TP_{ℓ^w,s^w,t^w} to registers (y_i, z^w, x^w, e_1^w).
```

Protocol 5: The encoding $\hat{F}_i(\boldsymbol{y}_i)$ of the i-th input qubit $\boldsymbol{y}_i$.

Complexity and Locality of the Encoding. We now argue that QGC has the claimed complexity properties. First, the QRE is decomposable. The offline part $\hat{F}_{\mathsf{off}}$ only depends on the circuit F, the classical randomness $J = \left(r, A, s, t, o\right)$, and the EPR pairs (e^w) (except for the qubits e_1^w for the input wires w). The online part $\hat{\boldsymbol{y}}_i = \hat{F}_i(\boldsymbol{y}_i)$ only depends on the classical randomness J, the i-th input qubit $\boldsymbol{y}_i$ and the qubit e_1^w for the i-th input wire w. Thus, the offline encoding and input encoding act on disjoint qubits of the EPR pairs (e^w).

The offline encoding $\hat{F}_{\mathsf{off}}$ can be computed using a $\mathbf{QNC}_{\mathrm{f}}^0$ circuit. The label setup procedure can be parallelized over all wires, and its complexity is inherited from the complexity of computing labels of CRE (which we are assuming can be done using an $\mathbf{NC}^0$ circuit). Since the complexity of computing the label function lab_{κ_g} is at most c_d where d is the depth of F, then the time complexity of the label setup is at most $O(c_d \cdot |\mathcal{W}|)$.

The gate encoding can be parallelized over all gates, and as discussed in Section 6.2.2, the complexity of encoding a single gate is $O(c_d)$, which includes the complexity of computing the classical encoding of the correction functions. Therefore encoding all gates has time complexity $O(c_d \cdot |\mathcal{G}|)$. The gate encoding can be implemented by a $\mathbf{QNC}_{\mathrm{f}}^0$ circuit.

Similarly, the encoding of the input qubits can be done in parallel. The complexity of the input encoding comes from setting up the labels and applying the teleportation gadget. This is $O(\kappa_d)$ complexity for encoding each input qubit, and can be done using a $\mathbf{QNC}_{\mathrm{f}}^0$ circuit.

The Case of Classical Inputs. We observe here that when the input $\boldsymbol{y}$ is classical, the input encoding process can also be taken to be entirely classical. Although applying the teleportation gadget TP on the input bits appears to be a fully quantum operation since it involves both a bit of the input $\boldsymbol{y}$ as well as half of an EPR pair, we note that the EPR pair can in this case be "pre-measured" in the standard (computational) basis (so that it collapses to a pair of correlated bits), and then one can apply a "classical" teleportation gadget:

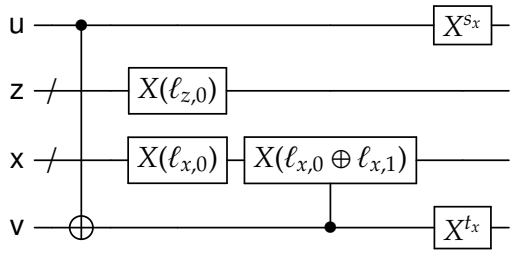


Figure 5: "Classical" teleportation gadget

Note that this circuit simply consists of CNOTs and bit flips, which are classically implementable. If the input state at the beginning of the classical teleportation gagdet is $|y, 0, 0, r, r\rangle$, then the result of applying the gadget is $|y \oplus s_x, \ell_{z,0}, \ell_{x,e}, e \oplus t_x, e \oplus y\rangle$ where $e = r \oplus y$. Thus the effect of this circuit is to transfer the bit y to register u′, and then encrypting it using a random bit e that is encoded in the label $\ell_{x,e}$.

6.4 Circuit Evaluation

We now describe the decoding procedure Dec, which takes a quantum randomized encoding $\hat{F}(\boldsymbol{y})$ and computes $F(\boldsymbol{y})$. The decoding procedure depends on the topology $\mathcal{T}$ of the circuit F, but does not depend on the specific gates themselves. The decoder picks an evaluation order π based on

the topology $\mathcal{T}$, and sequentially evaluates each gate g of the circuit F. Let $\mathcal{B}$ denote the set of "gate placeholders" in the topology $\mathcal{T}$. Let $\mathcal{T} \subseteq \mathcal{O}$ denote the set of output qubits to be traced out.

Compute an evaluation order π for the topology $\mathcal{T}$.
`// Evaluate the gates according to` π
for $g \in \mathcal{B}$ *ordered according to* π **do**
  Apply the unitary $\mathsf{GateEval}(g)$.
end
`// Decode the output`
for $w \in \mathcal{O} \setminus \mathcal{T}$ **do**
  Given the set of labels $\ell^w = (\ell^w_{z,0}, \ell^w_{z,1}, \ell^w_{x,0}, \ell^w_{x,1})$ in the register d^w, coherently decode the labels (z^w, x^w) in registers $(\mathsf{z}^w, \mathsf{x}^w)$ to bits $(d, e) \in \{0,1\}^2$, and apply $Z^d X^e$ to e_2^w.
end

Protocol 6: The decoding procedure Dec of QGC.

For a p-qubit gate g, the gate evaluation unitary $\mathsf{GateEval}(g)$ is defined as follows. Let $(v_1, \ldots, v_p) = \texttt{inwire}(g)$ and $(w_1, \ldots, w_p) = \texttt{outwire}(g)$ denote the input and output wires of g, respectively. Let

1. $(\mathsf{a}^{v_1}, \ldots, \mathsf{a}^{v_p})$ denote the ancilla registers for the input wires. Each register a^{v_i} is composed of subregisters $(\mathsf{z}^{v_i}, \mathsf{x}^{v_i}, \mathsf{b}^{v_i})$.
2. $(\mathsf{a}^{w_1}, \ldots, \mathsf{a}^{w_p})$ denote the ancilla registers for the output wires.
3. $(\mathsf{e}_2^{v_1}, \ldots, \mathsf{e}_2^{v_p})$ denote the input wire registers for g.
4. $(\mathsf{e}_1^{w_1}, \ldots, \mathsf{e}_1^{w_p})$ denote the output wire registers for g.
5. c^g denote the register storing the (offline portion of the) classical randomized encoding of the function $f(z_1, x_1, \ldots, z_p, x_p)$ that computes a tuple of correction gadgets $(\widehat{\mathsf{Corr}}_1, \ldots, \widehat{\mathsf{Corr}}_p)$.

We note that it may seem strange that the *input* registers are denoted by $e_2^{v_j}$, but this is because $e_2^{v_j}$ represents the output end of the EPR pair from a *previous* gate teleportation.

`// Compute the correction circuits`
Controlled on the values $(z^{v_1}, x^{v_1}, \ldots, z^{v_p}, x^{v_p})$ of registers $(\mathsf{z}^{v_1}, \mathsf{x}^{v_1}, \ldots, \mathsf{z}^{v_p}, \mathsf{x}^{v_p})$, and controlled on the value $\hat{f}_{\mathsf{off}}$ in register c^g, coherently run $\mathsf{CDec}_{\mathfrak{T}_g}(\hat{f}_{\mathsf{off}}, z^{v_1}, x^{v_1}, \ldots, z^{v_p}, x^{v_p})$ to obtain descriptions of correction circuits $(\widehat{\mathsf{Corr}}_1, \widehat{\mathsf{Corr}}_2, \ldots, \widehat{\mathsf{Corr}}_p)$.
`// Apply the correction circuits`
for $j \in [p]$ **do**
  Apply Corr_j to registers $(\mathsf{e}_2^{v_j}, \mathsf{a}^{w_j}, \mathsf{e}_1^{w_j})$.
  Apply Λ_3 to registers $(\mathsf{e}_2^{v_j}, \mathsf{a}^{w_j}, \mathsf{e}_1^{w_j})$.
end

Protocol 7: The gate evaluation operation, $\mathsf{GateEval}(g)$, for a p-qubit gate g.

In the gate evaluation, the map Λ_3 is the $\mathbf{QNC}^0_{\mathrm{f}}$ circuit given by Lemma 6.2. Furthermore, the CRE decoding procedure CDec only depends on the topology $\mathfrak{T}_g$ of the correction function

f, which only depends on the label lengths $(\kappa_{w_1}, \ldots, \kappa_{w_p})$ of the output wires. The complexity of gate evaluation is dominated by the complexity of running the decoding procedure CDec, which is polynomial in the size c_g of $\hat{f}_{\mathsf{off}}$. Thus the complexity of the decoding procedure Dec is $O(\sum_{g \in \mathcal{B}} \mathrm{poly}(c_g) + n)$, which is polynomial in the complexity of the encoding procedure.

6.5 The Simulator

We now present the simulator Sim for QGC. It depends on the topology $\mathcal{T}$ of the circuit being simulated, and takes as input a register s that is supposed to store the output of a quantum operation F that has topology $\mathcal{T}$. We assume that $\mathcal{T}$ has n input wires, and that the input register s is $|O \setminus \mathcal{T}|$ qubits wide (i.e. the number of output qubits of $\mathcal{T}$ that aren't traced out).

Intuitively the simulator computes the encoding of the identity circuit E with input s padded with zeroes (so the output should be the quantum state stored in register s). However the topology $\mathcal{T}$ could permute the ordering of qubits, so placing identity gates for all the placeholder gates of $\mathcal{T}$ may still result in a nontrivial operation on the input qubits. Furthermore, the topology $\mathcal{T}$ may trace out some qubits. Thus the simulator pads the input with some zeroes, shuffles the qubits according to the inverse of the permutation effected by the topology $\mathcal{T}$, and then computes the randomized encoding of E and the shuffled input.

1 Let E denote the general quantum circuit with topology $\mathcal{T}$ and gate set $\mathcal{G}$ consisting only of identity gates.
2 Let $n = |\mathcal{I}| = |O|$ denote the number of input and output wires of topology $\mathcal{T}$.
3 Compute the bijection $\xi : \mathcal{I} \to O$ that is the result of the (unitary part of) E.
4 Let P_ξ denote the unitary that swaps n qubits according to the bijection ξ.
5 Let y denote an n-qubit register consisting of s, padded by zeroes.
6 Permute the qubits of y according to the permutation P_ξ^{-1}.
7 Compute the encoding $\hat{E}(\boldsymbol{y})$ of circuit E and input in register y.

Protocol 8: The simulator Sim for QGC.

Clearly the complexity of the simulation procedure is polynomial in the complexity of the encoding procedure.

7 Correctness and Privacy Analysis

We now analyze the correctness and privacy of the QGC scheme. We argue that applying the decoding procedure Dec to a QRE $\hat{F}(\boldsymbol{y})$ of a circuit F and input $\boldsymbol{y}$ yields the state $F(\boldsymbol{y}) \otimes \rho$, where ρ is indistinguishable from a density matrix that only depends on the topology $\mathcal{T}$ of F, and nothing else about F. This clearly implies the correctness of the decoding procedure, but also shows the privacy of the QGC scheme: let E denote the "empty" circuit with the same topology $\mathcal{T}$. As discussed in Section 6.5, the empty circuit E may effect a permutation P on the qubits of $F(\boldsymbol{y})$, along with additional zero qubits. Applying the decoding procedure Dec to QRE $\hat{E}\left(P\left(F(\boldsymbol{y})\right)\right)$ yields a state that is indistinguishable from $F(\boldsymbol{y}) \otimes \rho$. Since the decoding procedure Dec is unitary, this implies that

$$\hat{E}\left(P\left(F(\boldsymbol{y})\right)\right) \approx \mathsf{Dec}^{-1}\Big(F(\boldsymbol{y}) \otimes \rho\Big) \approx \hat{F}(\boldsymbol{y}).$$

where "$\approx$" denotes indistinguishability either in the computational or information-theoretic sense, depending on the security properties of the classical randomized encoding scheme CRE used. This shows that the simulator Sim, on input $F(\boldsymbol{y})$, produces a state indistinguishable from $\hat{F}(\boldsymbol{y})$.

Some Notation. Let QGC denote the quantum randomized encoding scheme described in Section 6, where we use a classical randomized encoding scheme CRE that has perfect correctness and has polynomial-time encoding, decoding and simulation procedures. We assume that CRE has (t, ϵ)-privacy with respect to quantum adversaries. In the setting of computational security, t and ϵ are implicitly functions of a security parameter λ. For a topology $\mathfrak{T}$ of a classical circuit, let $\mathsf{CDec}_{\mathfrak{T}}$ and $\mathsf{CSim}_{\mathfrak{T}}$ denote the decoder and simulator for CRE for classical circuits with topology $\mathfrak{T}$, respfectively.

Fix an n-qubit input $\boldsymbol{y}$, quantum side information $\boldsymbol{q}$ (which may be entangled with $\boldsymbol{y}$), and a general quantum circuit $F = (\mathcal{T}, \mathcal{G})$ with $m = \text{poly}(n)$ gates. Let $\mathcal{B}$ denote the set of "gate placeholders" in the topology $\mathcal{T}$. As described in Section 6.4, the circuit evaluation only depends on the topology $\mathcal{T}$; it sequentially evaluates each gate in $\mathcal{B}$ according to some evaluation order π, and then decodes the output wires using the labels in the set $\mathcal{O} \setminus \mathcal{T}$. Let $g_1, \ldots, g_m \in \mathcal{B}$ denote the placeholder gates of topology $\mathcal{T}$ ordered according to π, and let $U_1, \ldots, U_m$ denote the corresponding unitaries in circuit F. Let $\mathcal{W}$ denote the set of wires in the topology $\mathcal{T}$, with $\mathcal{I}, \mathcal{O}$ denoting the input/output wires respectively.

Let $F_{\leq i}$ denote the part of circuit F up to (and including) gate g_i. (We define $F_{\leq 0}$ to denote the identity circuit with no gates). Thus, $F_{\leq i}$ represents the unitary operation $U_i \cdots U_1$.

Similarly, define $F_{>i}$ to denote the part of circuit F that starts after g_i. (We define $F_{>0}$ to denote F). As a quantum operation it first applies the unitaries $U_{i+1}, \ldots, U_m$, and then traces out the qubits specified by set $\mathcal{T} \subseteq \mathcal{O}$. As we will be considering randomized encodings of the partial circuit $F_{>i}$, we define the randomness used by the encodings. As described in Section 6.3, the randomness used to encode $F = F_{>0}$ is the sequence $J = (r, A, s, t, o)$. The randomness used to encode $F_{>i}$ is the sequence $J_{>i} = (r_{>i}, A_{>i}, s_{>i}, t_{>i}, o)$, where

- $r_{>i} = (r^g : g \text{ comes after } g_i)$
- $A_{>i} = (A^g : g \text{ comes after } g_i)$
- $s_{>i} = (s^w : \text{ wire } w \text{ comes after gates } g_1, \ldots, g_i)$
- $t_{>i} = (t^w : \text{ wire } w \text{ comes after gates } g_1, \ldots, g_i)$
- $o = (o^w)_{w \in \mathcal{O}}$ is the same as before.

We define $J_{>0} = J$. We define $J_{\leq i} = J \setminus J_{>i}$. This is all the randomness that is "used up" in evaluating the first i gates.

Given a classical value c, we write $[\![c]\!]$ to denote the density matrix $|c\rangle\langle c|$.

7.1 Analysis of the Decoding Procedure

The decoding procedure Dec evaluates each of the gates of F in sequence (according to some evaluation order), and then traces out a subset of qubits. The following Lemma gives a characterization of the result of each gate evaluation:

Lemma 7.1. *Let $i \in \{0, 1, \ldots, m-1\}$. Let $\mathsf{GateEval}_{i+1}$ denote the gate evaluation procedure from Section 6.4 corresponding to the gate g_{i+1} of topology $\mathcal{T}$. Let $\boldsymbol{q}$ denote quantum side information that is possibly entangled with $\boldsymbol{y}$, but uncorrelated with the classical randomness used by the encoding $\hat{F}(\boldsymbol{y})$. Then*

$$\mathop{\mathbb{E}}_{J_{>i}} \left(\mathsf{GateEval}_{i+1}\left(\hat{F}_{>i}\left(F_{\leq i}(\boldsymbol{y}\,;\, J_{>i})\right)\right), \boldsymbol{q}\right) = \mathop{\mathbb{E}}_{J_{>i+1}} \left(\hat{F}_{>i+1}(F_{\leq i+1}(\boldsymbol{y}\,;\, J_{>i+1}))\right), \boldsymbol{q}\right) \otimes \boldsymbol{t} \tag{7.1}$$

where for $j \in \{i, i+1\}$,

- *$\hat{F}_{>j}(\hat{F}_{\leq j}(\boldsymbol{y}\,;\, J_{>j}))$ denotes the quantum randomized encoding of the circuit $F_{>j}$ and input $\hat{F}_{\leq j}(\boldsymbol{y})$, where the randomness used for the encoding is $J_{>j}$.*
- *The density matrix $F_{\leq j}(\boldsymbol{y})$ is stored in registers*

$$(\mathsf{e}_2^v : v \text{ is the predecessor wire to some } w \in \mathtt{inwire}(g_{j+1}))$$

  where v is a predecessor wire to w if v, w are the k-th input and output wires respectively of the same gate g, for some k. If w is the k-th input wire of topology $\mathcal{T}$, then there is no predecessor wire, but in that case we define e_2^v to be the register y_k.

Furthermore, the density matrix $\boldsymbol{t}$ satisfies the following:

1. *It is on registers $(\mathsf{e}_2^v : v \text{ is the predecessor wire to } w \in \mathtt{inwire}(g_{i+1}))$ and $(\mathsf{a}^w, \mathsf{e}_1^w)_{w \in \mathtt{inwire}(g_{i+1})}$.*
2. *$\boldsymbol{t}$ is (t, ϵ)-indistinguishable from a density matrix $\tilde{\boldsymbol{t}}$ that only depends on topology $\mathcal{T}$.*
3. *$\boldsymbol{t}$ is unentangled with $\hat{F}_{>i+1}(F_{\leq i+1}(\boldsymbol{y}))$ and $\boldsymbol{q}$.*
4. *The state $\boldsymbol{t}$ is independent of the randomness $J_{>i+1}$.*

We first show how Lemma 7.1 implies the correctness and privacy of QGC. Using Lemma 7.1 repeatedly, we get

$$\mathop{\mathbb{E}}_{J} \left(\mathsf{GateEval}_m \cdot \mathsf{GateEval}_{m-1} \cdots \mathsf{GateEval}_1(\hat{F}(\boldsymbol{y})\,;\, J), \boldsymbol{q}\right) \tag{7.2}$$

is equal to

$$\mathop{\mathbb{E}}_{J_{>m}} \left(\hat{F}_{>m}(F_{\leq m}(\boldsymbol{y}\,;\, J_{>m}))\right), \boldsymbol{q}\right) \otimes \boldsymbol{t}_1 \otimes \cdots \otimes \boldsymbol{t}_m \tag{7.3}$$

where each $\boldsymbol{t}_i$ satisfies the conclusions of the statement of the Lemma. Note that $F_{\leq m}$ is the unitary part of the circuit F (without the tracing out of the output wires $\mathcal{T}$), and $\hat{F}_{>m}$ is the quantum randomized encoding of the partial trace operation $\mathrm{Tr}_{\mathcal{T}}(\cdot)$. Let $\boldsymbol{y}^{(m)} = F_{\leq m}(\boldsymbol{y})$ denote the state of the circuit before tracing out. Since the circuit $F_{>m}$ has no gates, the randomized encoding $\hat{F}_{>m}(\boldsymbol{y}^{(m)})$ is the state $\left(\hat{\boldsymbol{y}}_1^{(m)}, \ldots, \hat{\boldsymbol{y}}_n^{(m)}, (\boldsymbol{d}^w)_{w \in O \setminus \mathcal{T}}\right)$ where $\hat{\boldsymbol{y}}_j^{(m)}$ is the encoding of the j-th qubit of $\boldsymbol{y}^{(m)}$ and $\boldsymbol{d}^w$ is the label dictionary $\ell^w = (\ell^w_{z,0}, \ell^w_{z,1}, \ell^w_{x,0}, \ell^w_{x,1})$ for output wire $w \in O \setminus \mathcal{T}$.

First we note that the randomness $J_{>m}$ used in the encoding $\hat{F}_{>m}(F_{\leq m}(\boldsymbol{y}\,;\, J_{>m}))$ is the collection of random values $(o^w, s^w, t^w)_{w \in O}$, where $o^w = (o^w_x, o^w_z)$ are bits used to generate the labels $(\ell^w)_{w \in O}$ for the output wires, and (s^w, t^w) are randomization bits used for the teleportation gadget (see Section 6.3 for details on the randomness used in the encoding).

Fix an index $j \in [n]$. The state $\hat{\boldsymbol{y}}_j^{(m)}$ is on the following registers. Let $w \in O$ denote the j-th output wire. Let $v \in \mathcal{W}$ denote the predecessor wire to w.

- e_2^v. This register initially stores the j-th qubit of $\boldsymbol{y}_j^{(m)}$.

- $(\mathsf{e}_1^w, \mathsf{e}_2^w)$. These initially store the EPR pair corresponding to wire w.
- $\mathsf{a}^w = (\mathsf{z}^w, \mathsf{x}^w, \mathsf{b}^w)$. This is the ancilla register for wire w.

The joint state of $\hat{\boldsymbol{y}}_j^{(m)}$ and the register e_2^w in (7.3), when averaged over the randomness $(o^w, s^w, t^w)_{w \in \mathcal{O}}$, is equal to

$$\mathop{\mathbb{E}}_{o^w, s^w, t^w} \Big(\mathsf{TP}_{\ell^w, s^w, t^w}(\underbrace{\boldsymbol{y}_j^{(m)}, [\![0]\!], [\![0]\!], \boldsymbol{e}_1^w}_{\mathsf{e}_2^v \mathsf{z}^w \mathsf{x}^w \mathsf{e}_1^w}), \underbrace{\boldsymbol{e}_2^w}_{\mathsf{e}_2^w}, \underbrace{[\![0]\!]}_{\mathsf{b}^w} \Big) = \mathop{\mathbb{E}}_{o^w} \underbrace{\tau \otimes \mathop{\mathbb{E}}_{d,e} \big[\!\big[\ell_{z,d}^w, \ell_{x,e}^w \big]\!\big] \otimes \tau}_{\mathsf{e}_2^v \mathsf{z}^w \mathsf{x}^w \mathsf{e}_1^w} \otimes \underbrace{X^e Z^d(\boldsymbol{y}_j^{(m)})}_{\mathsf{e}_2^w} \otimes \underbrace{[\![0]\!]}_{\mathsf{b}^w} \tag{7.4}$$

where the density matrix τ denotes the maximally mixed qubit, and $[\![0]\!]$ denotes the pure state $|0\cdots0\rangle\langle 0\cdots 0|$ for the appropriate number of zero bits. Equation (7.4) follows from the following lemma.

Lemma 7.2. *Let $\boldsymbol{u}$ denote a qubit density matrix and let $\boldsymbol{q}$ denote quantum side information that is possibly entangled with $\boldsymbol{u}$. Let $s = (s_z, s_x)$ and $t = (t_z, t_x)$ denote randomization bits. Then for all labels $\ell = (\ell_{z,0}, \ell_{z,1}, \ell_{x,0}, \ell_{x,1})$, we have*

$$\mathop{\mathbb{E}}_{s,t} \Big(\mathsf{TP}_{\ell,s,t}\big(\boldsymbol{u}, [\![0\cdots0]\!], \boldsymbol{e}_1\big), \boldsymbol{e}_2, \boldsymbol{q} \Big) = \tau \otimes \mathop{\mathbb{E}}_{d,e} \big[\!\big[\ell_{z,d}, \ell_{x,e} \big]\!\big] \otimes \tau \otimes \Big(X^e Z^d(\boldsymbol{u}), \boldsymbol{q} \Big) \tag{7.5}$$

where $(\boldsymbol{e}_1, \boldsymbol{e}_2)$ is initialized in the state $|\mathsf{EPR}\rangle$.

Proof. We prove this by showing that Equation (7.5) holds when $\boldsymbol{u}$ is the outer product $|a\rangle\langle a'|$ and there is no quantum side information $\boldsymbol{q}$; the Lemma follows by linearity of the TP operation.

By Lemma 6.1, we have that

$$\mathsf{TP}_{\ell,s,t}\Big(|a\rangle \otimes |0\cdots0\rangle \otimes |\mathsf{EPR}\rangle \Big) = \frac{1}{2} \sum_{d,e} Z^{s_z} X^{s_x} |d\rangle \otimes |\ell_{z,d}, \ell_{x,e}\rangle \otimes Z^{t_z} X^{t_x} |e\rangle \otimes X^e Z^d |a\rangle \ .$$

Thus

$$\begin{aligned}
&\mathop{\mathbb{E}}_{s,t} \Big(\mathsf{TP}_{\ell,s,t}\big(|a\rangle\langle a'|, [\![0\cdots0]\!], \boldsymbol{e}_1 \big), \boldsymbol{e}_2 \Big) \\
&= \mathop{\mathbb{E}}_{s,t} \frac{1}{4} \sum_{d,e,d',e'} (-1)^{s_z \cdot (d \oplus d')} \cdot (-1)^{t_z \cdot (e \oplus e')} |d \oplus s_x\rangle\langle d' \oplus s_x| \otimes |\ell_{z,d}, \ell_{x,e}\rangle\langle \ell_{z,d'}, \ell_{x,e'}| \otimes |e \oplus t_x\rangle\langle e' \oplus t_x| \otimes \ X^e Z^d(|a\rangle\langle a'|) \\
&= \mathop{\mathbb{E}}_{s_x, t_x} \frac{1}{4} \sum_{d,e} [\![d \oplus s_x]\!] \otimes \big[\!\big[\ell_{z,d}, \ell_{x,e} \big]\!\big] \otimes [\![e \oplus t_x]\!] \otimes X^e Z^d(|a\rangle\langle a'|) \\
&= \mathop{\mathbb{E}}_{d,e} \tau \otimes \big[\!\big[\ell_{z,d}, \ell_{x,e} \big]\!\big] \otimes \tau \otimes X^e Z^d(|a\rangle\langle a'|).
\end{aligned}$$

□

The first observation is that for each $w \in \mathcal{O}$, the maximally mixed qubits τ in registers $(\mathsf{e}_2^v, \mathsf{e}_1^w)$ in Equation (7.4) are completely unentangled from the rest of the state described in (7.3), because the Pauli twirl bits s^w, t^w are only used to twirl the registers $(\mathsf{e}_2^v, \mathsf{e}_1^w)$, and is uncorrelated with the side information $\boldsymbol{q}$.

The next observation is that for each $w \in \mathcal{T}$ (i.e. the traced-out wires), the registers $(\mathsf{z}^w, \mathsf{x}^w, \mathsf{e}_2^w)$ of (7.3) are in the state

$$\mathop{\mathbb{E}}_{o_z^w, o_x^w} \mathop{\mathbb{E}}_{d,e} [\![o_z^w \oplus d, o_x^w \oplus e]\!] \otimes X^e Z^d(\boldsymbol{y}_j^{(m)}) = \mathop{\mathbb{E}}_{o_z^w, o_x^w} [\![o_z^w, o_x^w]\!] \otimes \mathop{\mathbb{E}}_{d,e} X^e Z^d(\boldsymbol{y}_j^{(m)}) = \tau \otimes \tau \otimes \tau$$

where w is the j-th output wire. Furthermore, the randomness (o_z^w, o_x^w) is uncorrelated with the rest of the state described in (7.3), so the registers $(\mathsf{z}^w, \mathsf{x}^w, \mathsf{e}_2^w)$ are unentangled with the rest of the state.

Finally, for each $w \in O \setminus \mathcal{T}$ (i.e. the non-traced-out wires), the registers $(\mathsf{d}^w, \mathsf{z}^w, \mathsf{x}^w, \mathsf{e}_2^w)$ are in the state

$$\mathop{\mathbb{E}}_{o_z^w, o_x^w} [\![o_z^w, o_z^w \oplus 1, o_x^w, o_x^w \oplus 1]\!] \otimes \mathop{\mathbb{E}}_{d,e} [\![o_z^w \oplus d, o_x^w \oplus e]\!] \otimes X^e Z^d (\boldsymbol{y}_j^{(m)})$$

Now consider the decoding procedure Dec, when applied to the encoding $\hat{F}(\boldsymbol{y})$. It evaluates each gate by applying $\mathsf{GateEval}_1, \mathsf{GateEval}_2, \ldots$, and then finally at the end decodes each wire $w \in O \setminus \mathcal{T}$ by undoing the Pauli twirl. The resulting state in registers $(\mathsf{d}^w, \mathsf{z}^w, \mathsf{x}^w, \mathsf{e}_2^w)$ is then indistinguishable from

$$\mathop{\mathbb{E}}_{o_z^w, o_x^w} [\![o_z^w, o_z^w \oplus 1, o_x^w, o_x^w \oplus 1]\!] \otimes \mathop{\mathbb{E}}_{d,e} [\![o_z^w \oplus d, o_x^w \oplus e]\!] \otimes \boldsymbol{y}_j^{(m)} = \left(\mathop{\mathbb{E}}_{a} [\![a, a \oplus 1]\!] \right)^{\otimes 2} \otimes \tau^{\otimes 2} \otimes \boldsymbol{y}_j^{(m)} .$$

Let σ denote the density matrix $\mathbb{E}_a [\![a, a \oplus 1]\!]$. Putting everything together, the result of applying the decoding procedure Dec to $(\hat{F}(\boldsymbol{y}), \boldsymbol{q})$ results in the following state:

$$\left(\mathrm{Tr}_{\mathcal{T}}(\boldsymbol{y}^{(m)}), \boldsymbol{q} \right) \otimes \underbrace{\left(\sigma^{\otimes 2} \otimes \tau \right)^{\otimes |O \setminus \mathcal{T}|} \otimes \left(\tau^{\otimes 3} \right)^{\otimes |\mathcal{T}|} \otimes \boldsymbol{t}_1 \otimes \cdots \otimes \boldsymbol{t}_m}_{\rho} .$$

Since $F(\boldsymbol{y}) = \mathrm{Tr}_{\mathcal{T}}(\boldsymbol{y}^{(m)})$, this proves the correctness of the decoding procedure. Note that ρ is $(t, \epsilon \cdot m)$-indistinguishable from the density matrix

$$\tilde{\rho} = \left(\sigma^{\otimes 2} \otimes \tau \right)^{\otimes |O \setminus \mathcal{T}|} \otimes \left(\tau^{\otimes 3} \right)^{\otimes |\mathcal{T}|} \otimes \tilde{\boldsymbol{t}}_1 \otimes \cdots \otimes \tilde{\boldsymbol{t}}_m$$

where each $\tilde{\boldsymbol{t}}_i$ only depends on $\mathcal{T}$, as given by Lemma 7.1. Thus $\tilde{\rho}$ only depends on the topology $\mathcal{T}$.

Similarly, since $E(P(F(\boldsymbol{y}))) = F(\boldsymbol{y})$, it must be that

$$\left(\hat{E}(P(F(\boldsymbol{y}))), \boldsymbol{q} \right) = \left(F(\boldsymbol{y}), \boldsymbol{q} \right) \otimes \rho'$$

where ρ' is $(t, \epsilon \cdot m)$-indistinguishable from $\tilde{\rho}$.

This in turn implies that $(\hat{F}(\boldsymbol{y}), \boldsymbol{q})$ is (t', ϵ')-indistinguishable from $(\hat{E}(P(F(\boldsymbol{y}))), \boldsymbol{q}) = (\mathsf{Sim}(F(\boldsymbol{y})), \boldsymbol{q})$, where $t' = t - \mathrm{poly}(s)$ where $\mathrm{poly}(s)$ is the complexity of Dec, and $\epsilon' = \epsilon \cdot m$. This establishes the privacy property of QGC.

7.2 Proof of Lemma 7.1

We prove Lemma 7.1 for the case that $i = 0$; the argument is identical for $i > 0$. In particular, we will show that

$$\mathop{\mathbb{E}}_{J} \left(\mathsf{GateEval}_1(\hat{F}(\boldsymbol{y}); J), \boldsymbol{q} \right) = \mathop{\mathbb{E}}_{J_{>1}} \left(\hat{F}_{>1}(U_1(\boldsymbol{y}); J_{>1}), \boldsymbol{q} \right) \otimes \boldsymbol{t} \tag{7.6}$$

where $\boldsymbol{t}, \boldsymbol{q}$ satisfy the conditions in the statement of Lemma 7.1.

Let g_1 denote a p-qubit gate, with U_1 being the corresponding unitary operator. For simplicity we drop the subscript and just write g, U from now on.

In encoding procedure presented in Section 6.3, each gate of F and each input qubit of $\boldsymbol{y}$ is encoded independently. Thus we can treat $\hat{F}$ as the following process:

1. Sample classical randomness $J = (r, A, s, t, o)$;

2. Compute the labels $(\ell^w)_{w \in \mathcal{W}}$;

3. Apply $M \otimes \mathsf{GateEnc}_1 \otimes \mathsf{TP}_1$ where $\mathsf{GateEnc}_1$ is the gate encoding unitary for gate g; the map TP_1 is the unitary to encode the p input qubits of $\boldsymbol{y}$ that are acted upon by U, and M is the concatenation of all the other gate encoding and input encoding unitaries.

In more detail:

- The map TP_1 is equal to

$$\mathsf{TP}_1 = \bigotimes_{v \in \mathtt{inwire}(g)} \mathsf{TP}_{\ell^v, s^v, t^v}$$

  where $\mathsf{TP}_{\ell^v, s^v, t^v}$ acts on the registers $(\mathsf{y}^v, \mathsf{a}^v, \mathsf{e}_1^v)$. Here, y^v denotes the input qubit that is encoded into input wire v. The register a^v can be decomposed as $(\mathsf{z}^v, \mathsf{x}^v, \mathsf{b}^v)$.

- The map $\mathsf{GateEnc}_1$ is equal to

$$\mathsf{GateEnc}_1 = \mathsf{GateEnc}_{g, r^g, A^g, \ell^g, s^g, t^g}$$

  where $s^g = (s^w)_{w \in \mathtt{outwire}(g)}$, $t^g = (t^w)_{w \in \mathtt{outwire}(g)}$, and $\ell^g = (\ell^w)_{w \in \mathtt{outwire}(g)}$. The map $\mathsf{GateEnc}_1$ acts on registers $(\mathsf{e}_2^v)_{v \in \mathtt{inwire}(g)}$, $(\mathsf{a}^w)_{w \in \mathtt{outwire}(g)}$, $(\mathsf{e}_1^w)_{w \in \mathtt{outwire}(g)}$, and c^g.

- The map M acts on a disjoint set of registers from TP_1 and $\mathsf{GateEnc}_1$.

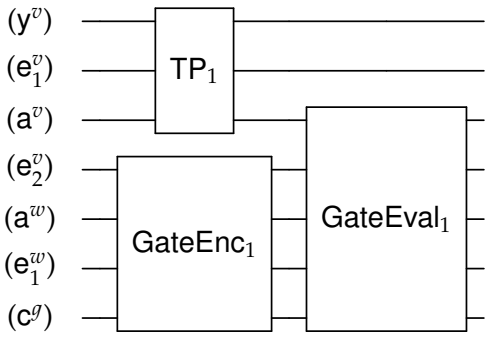


Figure 6: A circuit diagram of the encoding of gate g and its inputs, proceeded by its evaluation. The indices v, w range over $\mathtt{inwire}(g)$ and $\mathtt{outwire}(g)$ respectively.

We compute the state $\left(\mathsf{GateEval}_1(\hat{F}(\boldsymbol{y}; J)), \boldsymbol{q}\right)$, which is correlated with the randomness J and labels $(\ell^w)_{w \in \mathcal{W}}$. We focus specifically on the registers depicted in Figure 6, which are the ones acted upon by TP_1, $\mathsf{GateEnc}_1$, and $\mathsf{GateEval}_1$. In what follows, all randomness J and labels (ℓ^w) are fixed unless we explicitly average over certain random variables. Furthermore, the indices v, w range over $\mathtt{inwire}(g)$ and $\mathtt{outwire}(g)$ respectively.

Computing the Input Teleportation. Fix $v \in \mathtt{inwire}(g)$. We compute the result of applying $\mathsf{TP}_{\ell^v, s^v, t^v}$ to registers $(\mathsf{y}^v, \mathsf{z}^v, \mathsf{x}^v, \mathsf{e}_1^v, \mathsf{e}_2^v)$, after averaging over the randomization bits (s^v, t^v):

$$\mathop{\mathbb{E}}_{s^v, t^v} \left(\mathsf{TP}_{\ell^v, s^v, t^v} \left(\underbrace{\boldsymbol{y}^v, [\![0 \cdots 0]\!], \boldsymbol{e}_1^v}_{\mathsf{y}^v\, \mathsf{z}^v\, \mathsf{x}^v\, \mathsf{b}^v\, \mathsf{e}_1^v}, \underbrace{\boldsymbol{e}_2^v}_{\mathsf{e}_2^v} \right) \right) = \tau \otimes \mathop{\mathbb{E}}_{d_v, e_v} \left[\!\!\left[\ell^v_{z, d_v}, \ell^v_{x, e_v} \right]\!\!\right] \otimes [\![0]\!] \otimes \tau \otimes X^{e_v} Z^{d_v}(\boldsymbol{y}^v)$$

where on the right-hand side, the $[\![0]\!]$ state is on the b^v register (the teleportation gadget acts as the identity on register b^v). This follows from Lemma 7.2.

Computing the Gate Encoding. Now we compute the result of applying $\mathsf{GateEnc}_1$, where we've conditioned on specific values of d_v, e_v:

$$\mathsf{GateEnc}_1\Big(\bigotimes_v \underbrace{X^{e_v}Z^{d_v}(\boldsymbol{y}^v)}_{\mathsf{e}_2^v}, \underbrace{[\![0\cdots0]\!]}_{(\mathsf{a}^w)_w}, \underbrace{(\boldsymbol{e}_1^w)_w}_{(\mathsf{e}_1^w)_w}, \underbrace{[\![0]\!]}_{\mathsf{c}^g}\Big)$$
$$= \Big(A^g \cdot \Lambda_1 \cdot \Big(U\Big(\bigotimes_v X^{e_v}Z^{d_v}(\boldsymbol{y}^v)\Big), [\![0\cdots0]\!], (\boldsymbol{e}_1^w)_w\Big)\Big) \otimes \Big[\!\Big[\hat{f}_{g,\mathsf{off}}\Big]\!\Big] \tag{7.7}$$

The unitary Λ_1 is the tensor product

$$\Lambda_1 = \bigotimes_{w\in\mathtt{outwire}(g)} \Lambda_1(\ell^w)$$

where $\Lambda_1(\ell^w)$ are the $\mathbf{QNC}^0_f$ circuits specified by Lemma 6.2. The value $\hat{f}_{g,\mathsf{off}}$ is the (offline part of the) classical randomized encoding of the correction function $f(d_1, e_1, \ldots, d_p, e_p)$ (see Section 6.2.1 for the definition of correction functions and their encoding).

By our assumption on the gate set of the circuit F, we use Equation (6.1) to argue that

$$U \cdot \bigotimes_{v\in\mathtt{inwire}(g)} X^{e_v}Z^{d_v} = \Big(\bigotimes_{v\in\mathtt{inwire}(g)} R^\dagger_{d_v,e_v}\Big)\cdot U$$

for single-qubit PX group elements (R_{d_v,e_v}). Thus, the right-hand side of Equation (7.7) can be written as

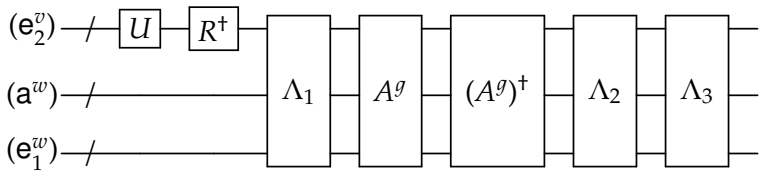
$$\Big(A^g \cdot \Lambda_1 \cdot \Big(\bigotimes_v R^\dagger_{d_v,e_v}\Big)\Big(U(\boldsymbol{y}^v)_v, [\![0\cdots0]\!], (\boldsymbol{e}_1^w)_w\Big)\Big) \otimes \Big[\!\Big[\hat{f}_{g,\mathsf{off}}\Big]\!\Big]$$

Computing the Gate Evaluation. Next, we compute the effect of applying the gate evaluation unitary $\mathsf{GateEval}_1$, which is described in Section 6.4. Controlled on the registers $(\mathsf{z}^v, \mathsf{x}^v)_v$, which store the labels $(\ell^v_{z,d_v}, \ell^v_{x,e_v})$ encoding the Pauli twirl $X^{e_v}Z^{d_v}$, the map $\mathsf{GateEval}_1$ uses the encoding $\hat{f}_{g,\mathsf{off}}$ to compute the correction function $f(d_1, e_1, \ldots, d_p, e_p)$. This yields (descriptions of) correction gadgets $(\widehat{\mathsf{Corr}}_v)_v$ such that

$$\mathsf{Corr}_v = \Lambda_2(R_{d_v,e_v}, \ell^w, s^w, t^w)\cdot(A^w)^\dagger$$

where w is the "successor wire" to v (i.e. v, w are the k-th input and output wires of g, respectively), and $\Lambda_2(R, \ell, s, t)$ are the unitaries specified by Lemma 6.2. Here, we assume that CRE has perfect correctness, which means that the correction gadgets are computed without error. Furthermore, for the rest of the section we will interchangeably index the input wires by either $v \in \mathtt{inwire}(g)$ or integers $1, \ldots, p$.

The unitary $\mathsf{GateEval}_1$ then applies the unitaries Corr_v to registers $(\mathsf{e}_2^v, \mathsf{a}^w, \mathsf{e}_1^w)$, followed by applying the $\mathbf{QNC}^0_f$ circuit Λ_3 (specified again by Lemma 6.2) to the same registers. Put together, the state of registers $(\boldsymbol{e}_2^v, \boldsymbol{a}^w, \boldsymbol{e}_1^w)_{v,w}$ looks like the following:



Here, R denotes $\bigotimes_v R_{d_v,e_v}$, Λ_2 denotes $\bigotimes_v \Lambda_2(R_{d_v,e_v}, \ell^w, s^w, t^w)$, and Λ_3 is applied transversally in the circuit. The A^g unitaries cancel out, and we are left with

$$\Lambda_3 \cdot \Lambda_2 \cdot \Lambda_1 \cdot \Big(\bigotimes_v R^\dagger_{d_v,e_v}\Big) \cdot \Big(U(\boldsymbol{y}^v)_v, [\![0 \cdots 0]\!], (\boldsymbol{e}_1^w)_w\Big)$$

By Lemma 6.2, this is equal to

$$\Big(\bigotimes_w \mathsf{TP}_{\ell^w,s^w,t^w}\Big) \cdot \Big(\bigotimes_v R_{d_v,e_v}\Big) \cdot \Big(\bigotimes_v R^\dagger_{d_v,e_v}\Big)\Big(U(\boldsymbol{y}^v)_v, [\![0 \cdots 0]\!], (\boldsymbol{e}_1^w)_w\Big)$$

The PX unitaries (R_{d_v,e_v}) cancel out, and we are left with

$$\Big(\bigotimes_w \mathsf{TP}_{\ell^w,s^w,t^w}\Big)\Big(U(\boldsymbol{y}^v)_v, [\![0 \cdots 0]\!], (\boldsymbol{e}_1^w)_w\Big)$$

Finishing the Proof. Consider the registers $(\mathsf{y}^v, \mathsf{e}_1^v, \mathsf{a}^v, \mathsf{e}_2^v)_v$, c^g, and $(\mathsf{a}^w, \mathsf{e}_1^w)_w$ of the "global" state $\Big(\mathsf{GateEval}_1(\hat{F}(\boldsymbol{y}\,;\,J)), \boldsymbol{q}\Big)$, where all randomness J is fixed except for the randomization bits $(s^v, t^v)_{v \in \mathtt{inwire}(g)}$, which we've averaged over. The density matrix in this registers can be written as

$$\Big(\bigotimes_v \underbrace{\tau^{\otimes 2}}_{\mathsf{y}^v\mathsf{e}_1^v} \otimes \mathop{\mathbb{E}}_{d_v,e_v} \underbrace{\Big[\!\Big[\ell^v_{z,d_v}, \ell^v_{x,e_v}\Big]\!\Big]}_{\mathsf{z}^v\mathsf{x}^v} \otimes \underbrace{[\![0]\!]}_{\mathsf{b}^v}\Big) \otimes \underbrace{\Big[\!\Big[\hat{f}_{g,\mathsf{off}}\Big]\!\Big]}_{\mathsf{c}^g} \otimes \Big(\bigotimes_w \mathsf{TP}_{\ell^w,s^w,t^w}\Big)\Big(\underbrace{U(\boldsymbol{y}^v)_v}_{(\mathsf{e}_2^v)_v}, \underbrace{[\![0 \cdots 0]\!]}_{(\mathsf{a}^w)_w}, \underbrace{(\boldsymbol{e}_1^w)_w}_{(\mathsf{e}_1^w)_w}\Big) \tag{7.8}$$

where as usual the indices v, w range over $\mathtt{inwire}(g), \mathtt{outwire}(g)$ respectively.

We now average over r^g and A^g. Consider the density matrix on registers $(\mathsf{y}^v, \mathsf{e}_1^v, \mathsf{z}^v, \mathsf{x}^v, \mathsf{b}^v)_v$ and c^g, which can be written as

$$\boldsymbol{t} = \mathop{\mathbb{E}}_{r^g,A^g} \bigotimes_v \underbrace{\tau^{\otimes 2}}_{\mathsf{y}^v\mathsf{e}_1^v} \otimes \mathop{\mathbb{E}}_{d_v,e_v} \underbrace{\Big[\!\Big[\ell^v_{z,d_v}, \ell^v_{x,e_v}\Big]\!\Big]}_{\mathsf{z}^v\mathsf{x}^v} \otimes \underbrace{[\![0]\!]}_{\mathsf{b}^v} \otimes \underbrace{\Big[\!\Big[\hat{f}_{g,\mathsf{off}}\Big]\!\Big]}_{\mathsf{c}^g} \tag{7.9}$$

Observe that the density matrix $\boldsymbol{t}$ does not depend on any other randomness of J. The remainder of the global state can be seen to be equal to the encoding of the partial circuit $F_{>1}$ with input $F_{\le 1}(\boldsymbol{y}) = U(\boldsymbol{y})$, and furthermore does not depend on the randomness $J_{\le 1} = (r^g, A^g, (s^v, t^v)_{v \in \mathtt{inwire}(g)})$.

Thus, averaging the global state over J, we get

$$\mathop{\mathbb{E}}_J \Big(\mathsf{GateEval}_1(\hat{F}(\boldsymbol{y}\,;\,J)), \boldsymbol{q}\Big) = \mathop{\mathbb{E}}_{J_{>1}} (\hat{F}_{>1}(F_{\le 1}(\boldsymbol{y}\,;\,J_{>1}), \boldsymbol{q}) \otimes \boldsymbol{t}\,.$$

We are almost done – we just need to argue that $\boldsymbol{t}$ is indistinguishable from a state $\tilde{\boldsymbol{t}}$ that only depends on the topology $\mathcal{T}$. Note that for fixed $(d_v, e_v)_v$ and A^g, the density matrix

$$\mathop{\mathbb{E}}_{r^g} \Big[\!\Big[\ell^v_{z,d_v}, \ell^v_{x,e_v}\Big]\!\Big] \otimes \Big[\!\Big[\hat{f}_{g,\mathsf{off}}\Big]\!\Big] \tag{7.10}$$

is precisely the output distribution of the CRE encoding of the correction function $f(d_1, e_1, \ldots, d_p, e_p)$. The value $\hat{f}_{g,\mathsf{off}}$ corresponds to the offline part of the encoding, and the $\ell^v_{z,d_v}, \ell^v_{x,e_v}$ correspond to the labels of the input bits d_v, e_v. The privacy guarantee of CRE is that the density matrix in (7.10) is (t, ϵ)-indistinguishable from

$$[\![\mathsf{CSim}_{\mathfrak{T}}(f(d_1, e_1, \ldots, d_p, e_p))]\!] = \Big[\!\Big[\mathsf{CSim}_{\mathfrak{T}}\big(\widehat{\mathsf{Corr}}_1, \ldots, \widehat{\mathsf{Corr}}_p\big)\Big]\!\Big]$$

where $\mathfrak{T}$ is the topology of the circuit computing f, and the topology $\mathfrak{T}$ depends only on the topology $\mathcal{T}$ of F. Thus the density matrix $\boldsymbol{t}$ is (t,ϵ)-indistinguishable from a density matrix $\tilde{\boldsymbol{t}}$, defined as follows:

$$\tilde{\boldsymbol{t}} = \tau^{\otimes 2p} \otimes \mathop{\mathbb{E}}_{A^g,(\tilde{d}_v,e_v)_v} \left[\!\!\left[\mathsf{CSim}_{\mathfrak{T}}\left(\widehat{\mathsf{Corr}}_1,\ldots,\widehat{\mathsf{Corr}}_p\right)\right]\!\!\right] \otimes [\![0]\!] \ .$$

Let $A^g = (A_1,\ldots,A_p)$, and note that $\widehat{\mathsf{Corr}}_j$ only depends on A_j. Since the A_j's are chosen independently from the randomization group $\mathcal{R}_{\kappa_j}$ for some label length κ_j, the distribution of the gadget $\widehat{\mathsf{Corr}}_j$ when averaged over A_j is going to be uniform over all correction gadgets corresponding to label length κ_j (see the discussion at the end of Section 6.1.2 for more details). Thus, $\mathbb{E}_{A^g}\left[\!\!\left[\mathsf{CSim}_{\mathfrak{T}}\left(\widehat{\mathsf{Corr}}_1,\ldots,\widehat{\mathsf{Corr}}_p\right)\right]\!\!\right]$ is a density matrix that only depends on the topology $\mathcal{T}$ of F (and is independent of all other randomness). Thus the density matrix in (7.9) is (t,ϵ)-indistinguishable from $\boldsymbol{t}$, which only depends on the topology $\mathcal{T}$.

This completes the proof of Lemma 7.1.

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.

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., [BOCG+06, Unr10, DNS10, DNS12, DGJ+19]).

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 [App17, Remark 1.4].)

In FHE, a client encrypts an input x and sends $\mathsf{Enc}(x)$ to a remote evaluator. The evaluator can then compute $\mathsf{Enc}(f(x))$ – for *any* function f – from the given ciphertext, without learning

anything about x or $f(x)$. It sends $\mathsf{Enc}(f(x))$ 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 [BJ15, DSS16, Mah18, Bra18].

Program Obfuscation. In program obfuscation, an obfuscator encodes a function f into an *obfuscated program* $\mathsf{Enc}(f)$, 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 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 [App17, 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 [AF16], 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) [BK20]. 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 Potential Applications of QRE

Many of the applications of RE in the classical setting seem to carry over to the quantum setting, possibly with some necessary adjustments. As the variety of applications of RE is so vast, we view it as outside the scope of this paper to go over and attempt to re-prove them. We therefore highlight the most immediate ones here.

PSM and Delegation. Two immediate applications that were mentioned above are private simultaneous messages (PSM) and delegation. Indeed, PSM is almost equivalent to decomposable RE, and therefore our results show how to achieve PSM in the quantum setting, using quantum messages and a quantum shared string (or even a classical shared string in the case where the

input is completely classical). In terms of delegation, QRE implies that any quantum computation can be delegated (in 2 messages) using, essentially, a $\mathbf{QNC}^0_f$ verifier.[21]

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, $\boldsymbol{x}, \boldsymbol{y}$ respectively, and they wish to jointly compute a quantum operation F on their inputs whose output is delivered to A.[22] This can be done as follows. Party A encrypts its input with a classical key using a quantum one-time pad (QOTP, [AMTdW00]), 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 $\boldsymbol{x}$, (unencrypted) $\boldsymbol{y}$, and classical QOTP key. On this input, F' first decrypts $\boldsymbol{x}$ and then applies the original F on $\boldsymbol{x}, \boldsymbol{y}$. Party B creates a decomposable QRE of F', plugging in the encrypted $\boldsymbol{x}$ that was received from A, and its own unencrypted input $\boldsymbol{y}$. 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 [DNS10], which is the mildest model of security in the quantum setting, if the underlying classical OT primitive is also specious secure.[23] 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 [BJ15, DSS16, Mah18, Bra18].

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 [DNS12, DGJ+19].

Functional Encryption. It was noticed in [SS10] that decomposable RE schemes imply a limited form of *functional encryption* (FE). An encryption scheme where there are multiple secret keys, associated with functions, so that when sk_f decrypts $\mathsf{Enc}(x)$ the output is $f(x)$. In the classical setting RE implies FE without "collusion resistance" (i.e. an adversary should not be allowed to obtain more than a single functional key). It was then showed [GVW12] 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 order to formally substantiate the definition and show the connection in the quantum setting.

[21] One should be careful since the final step of verification requires comparing two long classical strings, which can be done in $\mathbf{QNC}^0_f$ with bounded error [HS05].

[22] 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.

[23] At a high level, we believe that a proof goes through by requiring all parties to run all functions of the protocol in a purified manner.

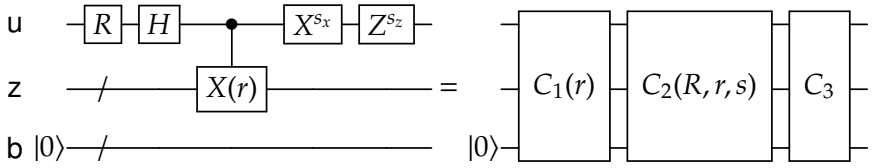
A more ambitious goal is to construct succinct FE schemes (even with bounded collusion) and so-called "reusable" garbled circuits which are function-private symmetric-key FE, analogously to the classical constructions in [GKP+13] (but possibly using different technique). One obstacle that seems to prevent direct application is the absence of a quantum attribute-based encryption schemes that are a central building block in that construction.

Classical Garbling and Quantum Obfuscation. 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 [App14a, BCG+18]. 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.

C Proof of Proposition 6.3

Here we give a proof of Proposition 6.3, restated here for convenience:

Proposition C.1. *For all single-qubit PX gates R, $s = (s_z, s_x) \in \{0,1\}^2$, and strings $r \in \{0,1\}^\kappa$ there exist $\mathbf{QNC}^0_f$ circuits $C_1(r)$ and C_3 and a depth-one Clifford circuit $C_2(R, r, s)$ in the randomization group $\mathcal{R}_\kappa$ such that the following circuit identity holds:*

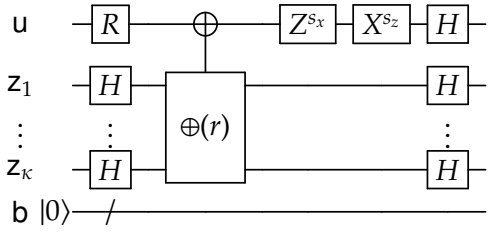


(C.1)

Here, the register b *consists of* $(\kappa+1)^2$ *qubits. Furthermore, descriptions of the circuits* $C_1(r)$ *and* $C_2(R, r, s)$ *can be computed by* $\mathbf{NC}^0$ *circuits of size* $O(\kappa^2)$.

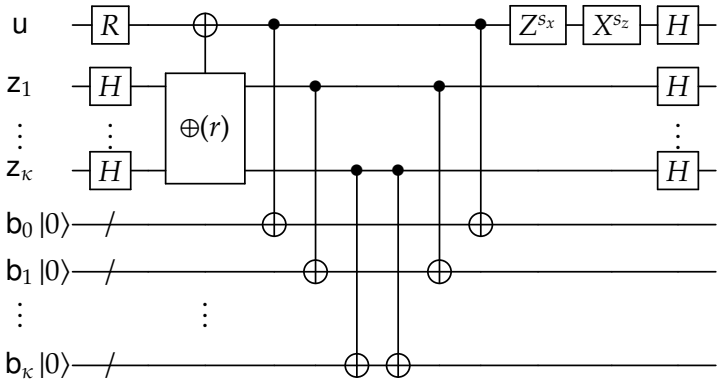
A general PX group element R can be written (up to a global phase) as $X^x Z^z P^p$ for some $x, z, p \in \{0,1\}$. We will perform the analysis for the case of $R = X$, $R = Z$, and $R = P$ separately, and then put everything together to deduce the Proposition.

Before doing so, we will transform the right hand side of (C.1) via sequence of circuit equivalences. First, we observe that the fan-out gate $CX(r)$ which maps $|u, z_1, \ldots, z_\kappa\rangle$ to $|u, z_1 \oplus (u \cdot r_1), \ldots, z_\kappa \oplus (u \cdot r_\kappa)\rangle$, when conjugated by Hadamards, becomes a parity gate $\oplus(r)$ that maps $|u, z_1, \ldots, z_\kappa\rangle$ to $|u \oplus (z \cdot r), z_1, \ldots, z_\kappa\rangle$ where $z \cdot r = \sum_i z_i \cdot r_i$. Next, we observe that we can move the X^{s_x} and Z^{s_z} gates before the Hadamard on register u, which changes them to Z^{s_x} and X^{s_z}. Thus we get that the right hand side of (C.1) is equivalent to



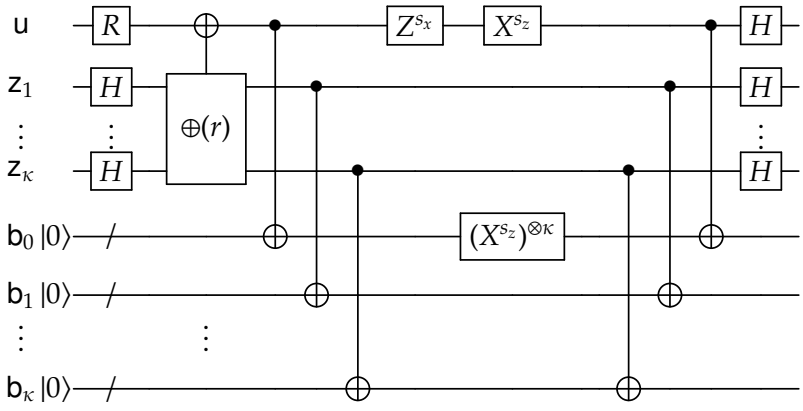
Circuit 7

Next, we apply a series of fan-out gates to copy the contents of the registers $\mathsf{u}, \mathsf{z}_1, \ldots, \mathsf{z}_\kappa$ to the ancilla register b. We label the individual qubits of register b as b_{ij} where $i, j \in \{0, 1, \ldots, \kappa\}$. We first apply a fan-out gate controlled on register u, with targets on $\{\mathsf{b}_{00}, \ldots, \mathsf{b}_{0\kappa}\}$. Then for each $j \in [\kappa]$, we apply a fan-out gate controlled on register z_j with targets $\{\mathsf{b}_{j0}, \ldots, \mathsf{b}_{j\kappa}\}$. We call these the "descending fan-outs". Then, we apply the same series of fan-out gates in reverse, to undo the copying procedure. We call these the "ascending fan-outs". Thus Circuit 7 is equivalent to Circuit 8.



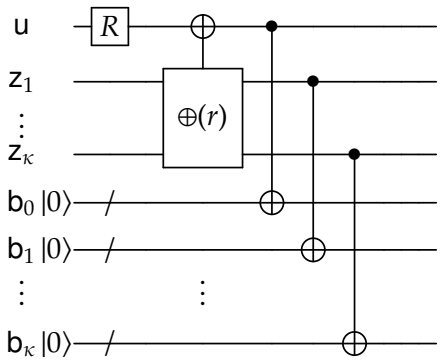
Circuit 8

Next, we move the Z^{s_x} and X^{s_z} gates left, before the ascending fan-outs. When moving the X^{s_z} gate past the fan-out controlled on register u, however, this incurs an X^{s_z} correction on each target of the fan-out, which are the $\{\mathsf{b}_{00}, \ldots, \mathsf{b}_{0\kappa}\}$ registers. Thus Circuit 8 is equivalent to Circuit 9.



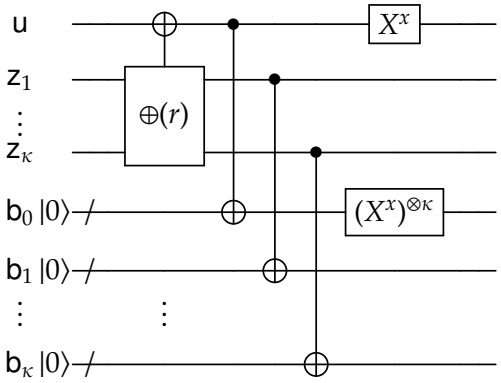
Circuit 9

For now we focus on the part of the circuit depicted in Circuit 10 – we omit the first layer of Hadamard gates, and everything past the descending fan-out gates. We show, for different gates R, how to "push" the R correction past the descending fan-out gates.



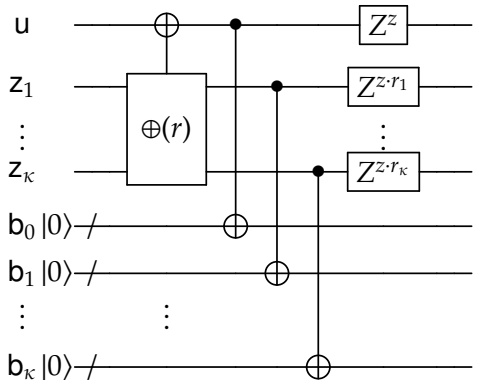
Circuit 10

Case 1. Suppose that $R = X^x$. First, the X^x gate commutes with the parity gate. Moving it past the first fan-out gate that is controlled on the register u incurs a X^x correction on each of the $\{\mathsf{b}_{0j}\}_j$ registers. Thus Circuit 10 is equivalent to Circuit 11.



Circuit 11

Case 2. Suppose now that $R = Z^z$. Moving the Z^z gate past the parity gate incurs a $Z^{z \cdot r_j}$ correction on the register z_j. This can be seen by repeatedly applying the identity $CNOT_{a,b} \cdot (I_a \otimes Z_b) = (Z_a \otimes Z_b) CNOT_{a,b}$, where $CNOT_{a,b}$ denotes a CNOT with control a and target b. Furthermore, the Z gates all commute with the fan-out gates. Thus Circuit 10 is equivalent to Circuit 12.

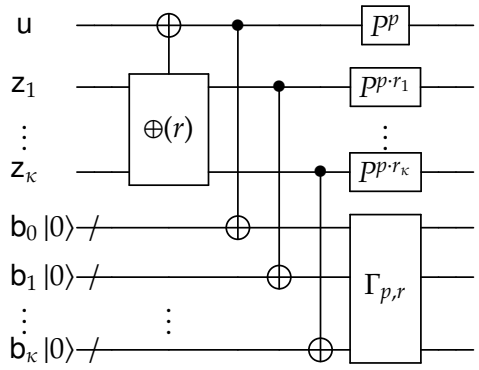


Circuit 12

Case 3. Suppose now that $R = P^p$. Then we claim that Circuit 10 is equivalent to Circuit 13. Here, if $p = 0$, then the gate $\Gamma_{p,r}$ is the identity. Otherwise, the gate $\Gamma_{p,r}$ stands for the following tensor-product of CZ gates:[24]

1. $CZ(\mathsf{b}_{0j}, \mathsf{b}_{j0})$ for all j such that $r_j = 1$, and
2. $CZ(\mathsf{b}_{ij}, \mathsf{b}_{ji})$ for all $i < j$ such that $r_i = r_j = 1$.

Notice that all of these CZ gates are disjoint, and thus can be implemented in a single layer.



Circuit 13

This can be verified by calculating the behavior of both circuits. For simplicity assume that $p = 1$. Circuit 10 on input $|u, z_1, \ldots, z_\kappa\rangle$ produces an output state that has the following structure:

- It is pre-multiplied by a phase factor i^u
- Registers u_0 and b_{0j} are in the state $|u \oplus (z \cdot r)\rangle$ for all j.
- Registers z_i and b_{ij} are in the state $|z_i\rangle$ for all $i > 0$ and all j.

On the other hand, the state of Circuit 13 is exactly the same except the phase factor in front is the product of

- i raised to the power $u + (z \cdot r) \mod 2$ (this comes from the P gate on register u).
- i raised to the power $\sum_i z_i$ (this comes from the P gates on registers $\{\mathsf{z}_i\}$).
- -1 raised to the power $(u+1)\sum_{j:r_j=1} z_j$ (this comes from the CZ gates applied to $(\mathsf{b}_{0j}, \mathsf{b}_{j0})$ for all j such that $r_j = 1$).
- -1 raised to the power $\sum_{i<j:r_i=r_j=1} z_i z_j$ (this comes from the CZ gates applied to $(\mathsf{b}_{ij}, \mathsf{b}_{ji})$ for all $i < j$ such that $r_i = r_j = 1$).

Using the identity that for bits $c_1, \ldots, c_n$,

$$c_1 + \cdots + c_n \mod 2 = \sum_i c_i - 2\sum_{i<j} c_i c_j \mod 4,$$

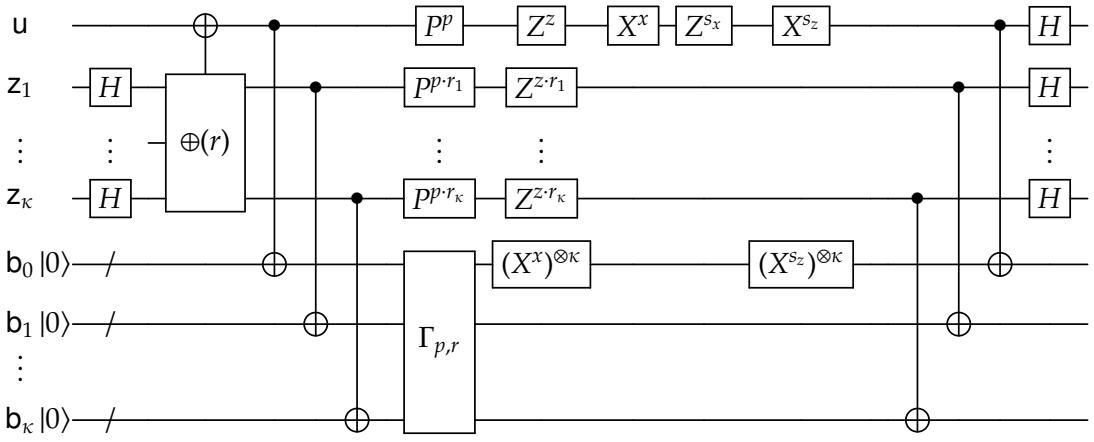
we get that these phase factors are equivalently

[24] Recall that a CZ gate is a controlled-Z gate; it maps $|a, b\rangle$ to $(-1)^{ab}|a, b\rangle$.

- i raised to the power $u + \sum_{i:r_i=1} z_i - 2u \sum_{j:r_j=1} z_j - 2 \sum_{i<j:r_i=r_j=1} z_i z_j$
- i raised to the power $\sum_{i:r_i=1} z_i$
- i raised to the power $2(u+1) \sum_{j:r_j=1} z_j$
- i raised to the power $2(\sum_{i<j:r_i=r_j=1} z_i z_j)$

Summing all of these exponents modulo 4, we get that the phase factor is i^u, as desired.

General case. Suppose that R is a general PX group element, so it can be written (up to a global phase) as $R = X^x Z^z P^p$ for $x, z, p \in \{0, 1\}$. Then by combining Cases 1, 2 and 3 we determine that Circuit 7 is equivalent to Circuit 14.



Circuit 14

Note that Circuit 14 has the desired structure:

- Letting $C_1(r)$ denote the the subcircuit up to and including the descending fan-out gates, we see that C_1 only depends on r and is independent of R and s. Furthermore, $C_1(r)$ can be implemented as a $\mathbf{QNC}^0_f$ circuit.
- Letting $C_2(R, r, s)$ denote the subcircuit in between the descending and ascending fan-out gates, we see that C_2 depends on R, r, s, and can be implemented as a tensor product of single-qubit Clifford unitaries on all registers except for the pairs $(\mathsf{b}_{0j}, \mathsf{b}_{j0})$ for $j > 0$, which have two-qubit Clifford unitaries acting on them (namely, products of X and CZ gates).
- Letting C_3 denote the subcircuit that includes the ascending fan-out gates as well as the final layer of Hadamard gates, we see that C_3 is independent of R, r, s, and can be implemented as a $\mathbf{QNC}^0_f$ circuit.

Furthermore, the description of the circuit $C_2(R, r, s)$ in Lines 7 through 12 shows that each of the single- and two-qubit gates are simple functions of the inputs (R, r, s), so this description can be computed by a classical $\mathbf{NC}^0$ circuit of size $O(\kappa^2)$.

This concludes the proof of the Proposition.

References

[AF16] Gorjan Alagic and Bill Fefferman. On quantum obfuscation. *arXiv preprint arXiv:1602.01771*, 2016.

[AG04] Scott Aaronson and Daniel Gottesman. Improved simulation of stabilizer circuits. *CoRR*, quant-ph/0406196, 2004.

[AIK04] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in NC^0. In *45th Symposium on Foundations of Computer Science (FOCS 2004), 17-19 October 2004, Rome, Italy, Proceedings*, pages 166–175, 2004.

[AIK06] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Computationally private randomizing polynomials and their applications. *Computational Complexity*, 15(2):115–162, 2006.

[AMTdW00] Andris Ambainis, Michele Mosca, Alain Tapp, and Ronald de Wolf. Private quantum channels. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*, pages 547–553, 2000.

[App14a] Benny Applebaum. Bootstrapping obfuscators via fast pseudorandom functions. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II*, volume 8874 of *Lecture Notes in Computer Science*, pages 162–172. Springer, 2014.

[App14b] Benny Applebaum. *Cryptography in Constant Parallel Time*. Information Security and Cryptography. Springer, 2014.

[App17] Benny Applebaum. Garbled circuits as randomized encodings of functions: a primer. In Yehuda Lindell, editor, *Tutorials on the Foundations of Cryptography*, pages 1–44. Springer International Publishing, 2017.

[BCG+18] Nir Bitansky, Ran Canetti, Sanjam Garg, Justin Holmgren, Abhishek Jain, Huijia Lin, Rafael Pass, Sidharth Telang, and Vinod Vaikuntanathan. Indistinguishability obfuscation for RAM programs and succinct randomized encodings. *SIAM J. Comput.*, 47(3):1123–1210, 2018.

[BFK09] Anne Broadbent, Joseph Fitzsimons, and Elham Kashefi. Universal blind quantum computation. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 517–526. IEEE, 2009.

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In Janos Simon, editor, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 103–112. ACM, 1988.

[BG19] Anne Broadbent and Alex B Grilo. Zero-knowledge for qma from locally simulatable proofs. *arXiv preprint arXiv:1911.07782*, 2019.

[BJ15] Anne Broadbent and Stacey Jeffery. Quantum homomorphic encryption for circuits of low t-gate complexity. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, pages 609–629, 2015.

[BJSW16] Anne Broadbent, Zhengfeng Ji, Fang Song, and John Watrous. Zero-knowledge proof systems for qma. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 31–40. IEEE, 2016.

[BK05] Sergey Bravyi and Alexei Kitaev. Universal quantum computation with ideal Clifford gates and noisy ancillas. *Physical Review A*, 71(2):022316, February 2005.

[BK20] Anne Broadbent and Raza Ali Kazmi. Indistinguishability obfuscation for quantum circuits of low *t*-count. *arXiv preprint arXiv:2005.14699*, 2020.

[Blu86] Manuel Blum. How to prove a theorem so no one else can claim it, August 1986. Invited 45 minute address to the International Congress of Mathematicians, 1986. To appear in the Proceedings of ICM 86.

[BMR90] Donald Beaver, Silvio Micali, and Phillip Rogaway. 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*, pages 503–513, 1990.

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

[Bra18] Zvika Brakerski. Quantum FHE (almost) as secure as classical. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part III*, volume 10993 of *Lecture Notes in Computer Science*, pages 67–95. Springer, 2018.

[BS20] Nir Bitansky and Omri Shmueli. Post-quantum zero knowledge in constant rounds. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 269–279, 2020.

[CR20] Lijie Chen and Hanlin Ren. Strong average-case circuit lower bounds from non-trivial derandomization. *Electronic Colloquium on Computational Complexity (ECCC)*, 27:10, 2020. To appear in STOC 2020.

[CVZ20] Andrea Coladangelo, Thomas Vidick, and Tina Zhang. Non-interactive zero-knowledge arguments for qma, with preprocessing. In *Annual International Cryptology Conference*, pages 799–828. Springer, 2020.

[DGJ+19] Yfke Dulek, Alex B. Grilo, Stacey Jeffery, Christian Majenz, and Christian Schaffner. Secure multi-party quantum computation with a dishonest majority. *CoRR*, abs/1909.13770, 2019.

[DNS10] Frédéric Dupuis, Jesper Buus Nielsen, and Louis Salvail. Secure two-party quantum evaluation of unitaries against specious adversaries. In Tal Rabin, editor, *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, volume 6223 of *Lecture Notes in Computer Science*, pages 685–706. Springer, 2010.

[DNS12] Frédéric Dupuis, Jesper Buus Nielsen, and Louis Salvail. Actively secure two-party evaluation of any quantum operation. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, volume 7417 of *Lecture Notes in Computer Science*, pages 794–811. Springer, 2012.

[DSS16] Yfke Dulek, Christian Schaffner, and Florian Speelman. Quantum homomorphic encryption for polynomial-sized circuits. In *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part III*, pages 3–32, 2016.

[FKN94] Uriel Feige, Joe Kilian, and Moni Naor. A minimal model for secure computation (extended abstract). In Frank Thomson Leighton and Michael T. Goodrich, editors, *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada*, pages 554–563. ACM, 1994.

[FLS90] Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple non-interactive zero knowledge proofs based on a single random string (extended abstract). In *31st Annual Symposium on Foundations of Computer Science, St. Louis, Missouri, USA, October 22-24, 1990, Volume I*, pages 308–317. IEEE Computer Society, 1990.

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Conference on the theory and application of cryptographic techniques*, pages 186–194. Springer, 1986.

[GKP+13] Shafi Goldwasser, Yael Kalai, Raluca Ada Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. Reusable garbled circuits and succinct functional encryption. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 555–564, 2013.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on computing*, 18(1):186–208, 1989.

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems. *Journal of the ACM (JACM)*, 38(3):690–728, 1991.

[Gol06] Oded Goldreich. *Foundations of Cryptography: Volume 1*. Cambridge University Press, USA, 2006.

[Got98] Daniel Gottesman. The Heisenberg Representation of Quantum Computers. *arXiv:quant-ph/9807006*, July 1998. arXiv: quant-ph/9807006.

[GVW12] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption with bounded collusions via multi-party computation. In Reihaneh Safavi-Naini and

Ran Canetti, editors, *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, volume 7417 of *Lecture Notes in Computer Science*, pages 162–179. Springer, 2012.

[HS05] Peter Høyer and Robert Spalek. Quantum fan-out is powerful. *Theory of Computing*, 1(1):81–103, 2005.

[HV16] Carmit Hazay and Muthuramakrishnan Venkitasubramaniam. On the power of secure two-party computation. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pages 397–429. Springer, 2016.

[JKPT12] Abhishek Jain, Stephan Krenn, Krzysztof Pietrzak, and Aris Tentes. Commitments and efficient zero-knowledge proofs from learning parity with noise. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 663–680. Springer, 2012.

[Kil88] Joe Kilian. Founding cryptography on oblivious transfer. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 20–31, 1988.

[KW17] Elham Kashefi and Petros Wallden. Garbled quantum computation. *Cryptography*, 1(1):6, 2017.

[Mah18] Urmila Mahadev. Classical homomorphic encryption for quantum circuits. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 332–338, 2018.

[Moo99] Cristopher Moore. Quantum circuits: Fanout, parity, and counting. *Electronic Colloquium on Computational Complexity (ECCC)*, 6(32), 1999.

[Rog91] Philip Rogaway. *The Round-Complexity of Secure Protocols*. PhD thesis, MIT, 1991.

[RT19] Ran Raz and Avishay Tal. Oracle separation of BQP and PH. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 13–23, 2019.

[SS10] Amit Sahai and Hakan Seyalioglu. Worry-free encryption: functional encryption with public keys. In Ehab Al-Shaer, Angelos D. Keromytis, and Vitaly Shmatikov, editors, *Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010, Chicago, Illinois, USA, October 4-8, 2010*, pages 463–472. ACM, 2010.

[Unr10] Dominique Unruh. Universally composable quantum multi-party computation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–505. Springer, 2010.

[VZ20] Thomas Vidick and Tina Zhang. Classical zero-knowledge arguments for quantum computations. *Quantum*, 4:266, 2020.

[Wat09] John Watrous. Zero-knowledge against quantum attacks. *SIAM Journal on Computing*, 39(1):25–58, 2009.

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *FOCS*, pages 162–167, 1986.

[Zha19] Jiayu Zhang. Delegating quantum computation in the quantum random oracle model. In Dennis Hofheinz and Alon Rosen, editors, *Theory of Cryptography - 17th International Conference, TCC 2019, Nuremberg, Germany, December 1-5, 2019, Proceedings, Part II*, volume 11892 of *Lecture Notes in Computer Science*, pages 30–60. Springer, 2019.

[Zha20] Jiayu Zhang. Succinct Blind Quantum Computation Using a Random Oracle, 2020. arXiv:2004.12621v2.