THEORY OF COMPUTING, Volume 14 (7), 2018, pp. 1–45
www.theoryofcomputing.org

Quantum Homomorphic Encryption for Polynomial-Size Circuits

Yfke Dulek Christian Schaffner* Florian Speelman†

Received December 24, 2016; Revised July 5, 2017; Published May 17, 2018

Abstract: We present a new scheme for quantum homomorphic encryption which is compact and allows for efficient evaluation of arbitrary quantum circuits, as long as those circuits are of size, polynomial in the security parameter. Building on the framework of Broadbent and Jeffery (CRYPTO'15) and recent results in the area of instantaneous non-local quantum computation by Speelman (TQC'16), we show how to construct quantum gadgets that allow perfect correction of the errors which occur during the homomorphic evaluation of T gates on encrypted quantum data. Our scheme can be based on any classical (leveled) fully homomorphic encryption (FHE) scheme and requires no computational assumptions besides those already used by the classical scheme. The size of our quantum gadget depends on the space complexity of the classical decryption function—which aligns well with the current efforts to minimize the complexity of the decryption function.

Our scheme (or slight variants of it) offers a number of additional advantages such as ideal compactness, the ability to supply gadgets "on demand," circuit privacy for the evaluator against passive adversaries, and a three-round scheme for blind delegated quantum computation which puts only very limited demands on the quantum abilities of the client.

ACM Classification: E.3, F.1.2

AMS Classification: 68P25, 69Q12, 81P68, 81P94, 94A60

Key words and phrases: quantum computing, quantum cryptography, homomorphic encryption, quantum teleportation, garden-hose model

A conference version of this paper appeared in Advances in Cryptology: Proceedings of the 36th International Cryptology Conference (CRYPTO 2016) [22].

*7th framework EU SIQS and QALGO, and a NWO VIDI grant.

†ERC Grant Agreement no 337603, Danish Council for Independent Research, and VILLUM FONDEN Grant no 10059.

DOI: 10.4086/toc.2018.v014a007

1 Introduction

Fully homomorphic encryption (FHE) is the holy grail of modern cryptography. Rivest, Adleman and Dertouzous were the first to observe the possibility of manipulating encrypted data in a meaningful way, rather than just storing and retrieving it [46]. After some partial progress [33, 45, 10, 37] over the years, a breakthrough happened in 2009 when Gentry presented a fully-homomorphic encryption (FHE) scheme [29]. Since then, FHE schemes have been simplified [56] and based on more standard assumptions [12]. The exciting developments around FHE have sparked a large amount of research in other areas such as functional encryption [31, 34, 32, 49] and obfuscation [28].

Developing quantum computers is a formidable technical challenge, so it currently seems likely that quantum computing will not be available immediately to everyone and hence quantum computations have to be outsourced. Given the importance of classical[1] FHE for "computing in the cloud," it is natural to wonder about the existence of encryption schemes which can encrypt *quantum data* in such a way that a server can carry out arbitrary *quantum computations* on the encrypted data (without interacting with the encrypting party[2]). While previous work on *quantum homomorphic encryption* has mostly focused on information-theoretic security (see Section 1.2 below for details), schemes that are based on computational assumptions have only recently been thoroughly investigated by Broadbent and Jeffery. In [16], they give formal definitions of quantum fully homomorphic encryption (QFHE) and its security and they propose three schemes for quantum homomorphic encryption assuming the existence of classical FHE.

A natural idea is to encrypt a message qubit with the quantum one-time pad (i. e., by applying a random Pauli operation), and send the classical keys for the quantum one-time pad along as classical information, encrypted by the classical FHE scheme. This basic scheme is called CL in [16]. It is easy to see that CL allows an evaluator to compute arbitrary Clifford operations on encrypted qubits, simply by performing the actual Clifford circuit, followed by homomorphically updating the quantum one-time pad keys according to the commutation rules between the performed Clifford gates and the Pauli encryptions. The CL scheme can be regarded as analogous to additively homomorphic encryption schemes in the classical setting. The challenge, like multiplication in the classical case, is to perform non-Clifford gates such as the T gate. Broadbent and Jeffery propose two different approaches for doing so, accomplishing homomorphic encryption for circuits with a limited number of T gates. These results lead to the following main open problem:

Is it possible to construct a computationally secure quantum homomorphic scheme that allows evaluation of arbitrary quantum circuits of polynomial size?

1.1 Our contributions

We answer the above question in the affirmative by presenting a new scheme TP (as abbreviation for teleportation) for quantum homomorphic encryption which is both compact and efficient. The scheme is secure against chosen-plaintext attacks from quantum adversaries, as formalized by the security notion

[1] Here and throughout the article, we use "classical" to mean "non-quantum."

[2] In contrast to *blind* or *delegated quantum computation* where some interaction between client and server is usually required, see Section 1.2 for references.

q-IND-CPA security defined by Broadbent and Jeffery [16], as long as the number of T gates in the evaluated circuit (and thus the total size of the evaluated circuit) is polynomial[3] in the security parameter.

Like the schemes proposed in [16], our scheme is an extension of the Clifford scheme CL. We add auxiliary quantum states to the evaluation key which we call quantum *gadgets* and which aid in the evaluation of the T gates. The size of a gadget depends only on (a certain form of) the space complexity of the decryption function of the classical FHE scheme. This relation turns out to be very convenient, as classical FHE schemes are often optimized with respect to the complexity of the decryption operation (in order to make them bootstrappable). As a concrete example, if we instantiate our scheme with the classical FHE scheme by Brakerski and Vaikuntanathan [12], each evaluation gadget of our scheme consists of a number of qubits which is polynomial in the security parameter.

In TP, the quantum gadgets are used as follows. After a T gate is performed on a one-time-pad encrypted qubit $\mathsf{X}^a\mathsf{Z}^b|\psi\rangle$, the result might contain an unwanted phase P^a depending on the key a with which the qubit was encrypted, since $\mathsf{T}\mathsf{X}^a\mathsf{Z}^b|\psi\rangle = \mathsf{P}^a\mathsf{X}^a\mathsf{Z}^b\mathsf{T}|\psi\rangle$. Obviously, the evaluator is not allowed to know the key a. Instead, he holds an encryption $\tilde{a}$ of the key, produced by a classical FHE scheme. The evaluator can teleport the encrypted qubit "through the gadget" [36] in a way that depends on $\tilde{a}$, in order to remove the unwanted phase. In more detail, the quantum part of the gadget consists of a number of EPR pairs which are prepared in a way that depends on the secret key of the classical FHE scheme. Some classical information is provided with the gadget that allows the evaluator to homomorphically update the encryption keys after the teleportation steps. On a high level, the use of an evaluation gadget corresponds to a *instantaneous non-local quantum computation*[4] where one party holds the secret key of the classical FHE scheme, and the other party holds the input qubit and a classical encryption of the key to the quantum one-time pad. Together, this information determines whether an inverse phase gate $\mathsf{P}^\dagger$ needs to be performed on the qubit or not. Very recent results by Speelman [53] show how to perform such computations with a bounded amount of entanglement. These techniques are the crucial ingredients for our construction and are the reason why the *garden-hose complexity* [17] of the decryption procedure of the classical FHE is related to the size of our gadgets.

We require exactly one evaluation gadget for every T gate that is to be homomorphically evaluated. For this reason, TP is leveled: in order to produce enough gadgets, the key generation algorithm requires an upper bound to the number of T gates as an argument. Note that this also means that key generation scales (linearly) with this upper bound. However, since the work that is done at key generation time is completely independent of the input to the circuit, we can view key generation as an offline preparation phase.

The quantum part of our evaluation gadget consists only of a number of Bell pairs arranged in a specific way. This structural simplicity provides a number of advantages in the construction and analysis of TP. To start with, the evaluation of a T gate does not cause errors to accumulate on the quantum state. The scheme is very compact in the sense that the state of the system after the evaluation of a T gate has the same form as after the initial encryption, except for any classical changes caused by the classical FHE evaluation. This kind of compactness also implies that individual evaluation gadgets can be supplied "on

[3] It is important for compactness that the size of the circuit can be chosen as an *arbitrary* polynomial of the security parameter, and in particular doesn't influence the complexity of decryption. We give a precise specification of the quantifiers involved in Section 2.2.

[4] This term is not related to the term "instantaneous quantum computation" [51], and instead first was used as a specific form of non-local quantum computation, one where all parties have to act simultaneously.

demand" by the holder of the secret key. Once an evaluator runs out of gadgets, the secret key holder can simply supply more of them.

Furthermore, TP does not depend on a specific classical FHE scheme, hence any advances in classical FHE can directly improve our scheme. Our requirements for the classical FHE scheme are quite modest: we only require the classical scheme to have a space-efficient decryption procedure and to be secure against quantum adversaries. In particular, no circular-security assumption is required. Since we supply at most a polynomial number of evaluation gadgets, our scheme TP is leveled homomorphic by construction, and we can simply switch to a new classical key after every evaluation gadget. In fact, the Clifford gates in the quantum evaluation circuit only require additive operations from the classical homomorphic scheme, while each T gate needs a fixed (polynomial) number of multiplications. Hence, we do not actually require fully homomorphic classical encryption, but leveled fully homomorphic schemes suffice.

Finally, circuit privacy in the passive setting almost comes for free. When wanting to hide which circuit was evaluated on the data, the evaluating party can add an extra randomization layer to the output state by applying his own one-time pad. We show that if the classical FHE scheme has the circuit-privacy property, then this extra randomization completely hides the circuit from the decrypting party. This is not unique to our specific scheme: the same is true for CL.

In terms of applications, our construction can be appreciated as a constant-round scheme for *blind delegated quantum computation*, using computational assumptions. The server can evaluate a universal quantum circuit on the encrypted input, consisting of the client's quantum input and a (classical) description of the client's circuit. In this context, it is desirable to minimize the quantum resources needed by the client. We argue that our scheme can still be used for constant-round blind delegated quantum computation if we limit either the client's quantum memory or the types of quantum operations the client can perform. In both cases, the client still needs some quantum computational power and a reliable quantum channel to the server, but it seems unlikely that one can get rid of these requirements completely [1].

As another application, we can instantiate our construction with a classical FHE scheme that allows for *distributed* key generation and decryption amongst different parties that all hold a share of the secret key [6]. In that case, it is likely that our techniques can be adapted to perform *multiparty quantum computation* [9] in the semi-honest case. However, the focus of this article lies on the description and security proof of the new construction, and more concrete applications are the subject of upcoming work.

1.2 Related work

Early classical FHE schemes were limited in the sense that they could not facilitate arbitrary operations on the encrypted data: some early schemes only implemented a single operation (addition or multiplication) [47, 33, 45]; later on it became possible to combine several operations in a limited way [10, 30, 50]. Gentry's first fully homomorphic encryption scheme [29] relied on several non-standard computational assumptions. Subsequent work [11, 12] has relaxed these assumptions or replaced them with more conventional assumptions such as the hardness of learning with errors (LWE), which is believed to be hard also for quantum attackers. It is impossible to completely get rid of computational assumptions for a classical FHE scheme, since the existence of such a scheme would imply the existence of an information-theoretically secure protocol for private information retrieval (PIR) [39] that breaks the lower bound on the amount of communication required for that task [20, 24].

While quantum fully homomorphic encryption (QFHE) is closely related to the task of blind or delegated quantum computation [18, 15, 2, 57, 25, 13, 41], QFHE does not allow interaction between the client and the server during the computation. Additionally, in QFHE, the server is allowed to choose which unitary it wants to apply to the (encrypted) data.

Yu, Pérez-Delgado and Fitzsimons [58] showed that perfectly information-theoretically secure QFHE is not possible for arbitrary unitary transformations, unless the size of the encryption grows exponentially in the input size. Thus, any scheme that attempts to achieve information-theoretically secure QFHE has to leak some proportion of the input to the server [5, 48] or can only be used to evaluate a subset of all unitary transformations on the input [48, 40, 54]. Information-theoretically secure QFHE is not known to be achievable for the subset of our interest, that of all unitary transformations for which efficient quantum circuits exist. Like the multiplication operation is hard in the classical case, the hurdle in the quantum case seems to be the evaluation of non-Clifford gates. A recent result by Ouyang, Tan and Fitzsimons provides information-theoretic security for circuits with at most a constant number of non-Clifford operations [44].

Broadbent and Jeffery [16] proposed two schemes that achieve homomorphic encryption for nontrivial sets of quantum circuits. Instead of trying to achieve information-theoretic security, they built their schemes based on a classical FHE scheme and hence any computational assumptions on the classical scheme are also required for the quantum schemes. Computational assumptions allow bypassing the impossibility result from [58] and work toward a (quantum) fully homomorphic encryption scheme.

Both of the schemes presented in [16] are extensions of the scheme CL described in Section 1.1. These two schemes use different methods to implement the evaluation of a T gate, which we briefly discuss here. In the EPR scheme, some entanglement is accumulated in a special register during every evaluation of a T gate, and stored there until it can be resolved in the decryption phase. Because of this accumulation, the complexity of the decryption function scales (quadratically) with the number of T gates in the evaluated circuit, thereby violating the compactness requirement of QFHE. The scheme AUX also extends CL, but handles T gates in a different manner. The evaluator is supplied with auxiliary quantum states, stored in the evaluation key, that allow him to evaluate T gates and immediately remove any error that may have occurred. In this way, the decryption procedure remains very efficient and the scheme is compact. Unfortunately, the required auxiliary states grow doubly exponentially in size with respect to the T depth of the circuit, rendering AUX useful only for circuits with constant T depth. Our scheme TP is related to AUX in that extra quantum resources for removing errors are stored in the evaluation key. In sharp contrast to AUX, the size of the evaluation key in TP only grows linearly in the number of T gates in the circuit (and polynomially in the security parameter), allowing the scheme to be leveled fully homomorphic. Since the evaluation of the other gates causes no errors on the quantum state, no gadgets are needed for those; any circuit containing a number of T gates that is polynomial in the security parameter can be efficiently and securely evaluated.

1.3 Structure of this paper

We start by introducing some notation in Section 2 and presenting the necessary preliminaries on quantum computation, (classical and quantum) homomorphic encryption, and the garden-hose model which is essential to the most-general construction of the gadgets. In Section 3, we describe the scheme TP and show that it is compact. The security proof of TP is somewhat more involved, and is presented in several steps in Section 4, along with an informal description of a circuit-private variant of the scheme. In Section 5, the rationale behind the quantum gadgets is explained, and some examples are discussed to clarify the construction. We conclude our work in Section 6 and propose directions for future research.

2 Preliminaries

2.1 Quantum computation

We assume the reader is familiar with the standard notions in the field of quantum computation. (For an introduction, see [43].) In this subsection, we only mention the concepts that are essential to our construction.

The single-qubit *Pauli group* is, up to global phase, generated by the bit flip and phase flip operations,

$$\mathsf{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathsf{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

A *Pauli operator* on n qubits is simply any tensor product of n independent single-qubit Pauli operators. All four single-qubit Pauli operators are of the form $\mathsf{X}^a\mathsf{Z}^b$ with $a,b \in \{0,1\}$. Here, and in the rest of the paper, we ignore the global phase of a quantum state, as it is not observable by measurement.

The *Clifford group* on n qubits consists of all unitaries C that commute with the Pauli group, i. e., the Clifford group is the normalizer of the Pauli group. Since all Pauli operators are of the form $\mathsf{X}^{a_1}\mathsf{Z}^{b_1} \otimes \cdots \otimes \mathsf{X}^{a_n}\mathsf{Z}^{b_n}$, this means that C is a Clifford operator if for any $a_1, b_1, \ldots, a_n, b_n \in \{0,1\}$ there exist $a'_1, b'_1, \ldots, a'_n, b'_n \in \{0,1\}$ such that (ignoring global phase):

$$\mathsf{C}(\mathsf{X}^{a_1}\mathsf{Z}^{b_1} \otimes \cdots \otimes \mathsf{X}^{a_n}\mathsf{Z}^{b_n}) = (\mathsf{X}^{a'_1}\mathsf{Z}^{b'_1} \otimes \cdots \otimes \mathsf{X}^{a'_n}\mathsf{Z}^{b'_n})\mathsf{C}.$$

All Pauli operators are easily verified to be elements of the Clifford group, and the entire Clifford group is generated by

$$\mathsf{P} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \quad \mathsf{H} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad \text{and } \mathsf{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

(See for example [35].) The commutation rules for single-qubit Clifford gates on wire i, or two-qubit Clifford gates on wires i and j, are given by

$$\begin{aligned} \mathsf{P}\mathsf{X}^{a_i}\mathsf{Z}^{b_i} &= \mathsf{X}^{a_i}\mathsf{Z}^{a_i \oplus b_i}\mathsf{P}, \\ \mathsf{H}\mathsf{X}^{a_i}\mathsf{Z}^{b_i} &= \mathsf{X}^{b_i}\mathsf{Z}^{a_i}\mathsf{H}, \\ \mathsf{CNOT}(\mathsf{X}^{a_i}\mathsf{Z}^{b_i} \otimes \mathsf{X}^{a_j}\mathsf{Z}^{b_j}) &= (\mathsf{X}^{a_i}\mathsf{Z}^{a_i \oplus b_j} \otimes \mathsf{X}^{a_j \oplus a_i}\mathsf{Z}^{b_j})\mathsf{CNOT}. \end{aligned}$$

These commutation rules are important to our construction, because they describe how the encryption key should be updated after the application of a Clifford gate. They can be found in many places in the literature (or can be easily calculated by hand), see also, e. g., [16, Appendix C].

The Clifford group does not suffice to simulate arbitrary quantum circuits, but by adding any single non-Clifford gate, any quantum circuit can be efficiently simulated with only a small error. As in [16], we choose this non-Clifford gate to be the T gate,

$$\mathsf{T} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}.$$

Note that the T gate, because it is non-Clifford, does not commute with the Pauli group. More specifically, we have $\mathsf{T}\mathsf{X}^a\mathsf{Z}^b = \mathsf{P}^a\mathsf{X}^a\mathsf{Z}^b\mathsf{T}$. It is exactly the formation of this P gate that has proven to be an obstacle to the design of an efficient quantum homomorphic encryption scheme.

We use $|\psi\rangle$ or $|\varphi\rangle$ to denote pure quantum states. Mixed states are denoted with ρ or σ. Let $\mathbb{I}_d$ denote the identity matrix of dimension d, this allows us to write the d-dimensional *completely mixed state* as $\mathbb{I}_d/d$.

Define

$$\left|\Phi^+\right\rangle := \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

to be an EPR pair.

If X is a random variable ranging over the possible basis states B for a quantum system, then let $\rho(X)$ be the density matrix corresponding to X, i. e., $\rho(X) := \sum_{b\in B} \Pr[X = b]|b\rangle\langle b|$.

Applying a Pauli operator that is chosen uniformly at random results in a single-qubit completely mixed state, since

$$\forall \rho \ : \ \sum_{a,b\in\{0,1\}} \left(\frac{1}{4}\mathsf{X}^a\mathsf{Z}^b\rho\mathsf{Z}^b\mathsf{X}^a\right) = \frac{\mathbb{I}_2}{2}\ .$$

This property is used in the construction of the *quantum one-time pad*: applying a random Pauli $\mathsf{X}^a\mathsf{Z}^b$ to a qubit completely hides the content of that qubit to anyone who does not know the key (a,b) to the pad. Anyone in possession of the key can decrypt simply by applying $\mathsf{X}^a\mathsf{Z}^b$ again.

2.2 Homomorphic encryption

This subsection provides the definitions of (classical and quantum) homomorphic encryption schemes, and the security conditions for such schemes. In the current work, we only consider homomorphic encryption in the public-key setting. For a more thorough treatment of these concepts, and how they can be transferred to the symmetric-key setting, see [16].

2.2.1 The classical setting

A classical homomorphic encryption scheme HE consists of four algorithms: key generation, encryption, evaluation, and decryption. The key generator produces three keys: a public key and evaluation key, both of which are publicly available to everyone, and a secret key which is only revealed to the decrypting party. Anyone in possession of the public key can encrypt the inputs $x_1,\dots,x_\ell$, and send the resulting ciphertexts $c_1,\dots,c_\ell$ to an evaluator who evaluates some circuit C on them. The evaluator sends the result to a party that possesses the secret key, who should be able to decrypt it to $\mathsf{C}(x_1,\dots,x_\ell)$.

More formally, HE consists of the following four algorithms which run in probabilistic polynomial time in terms of their input and parameters [12].

$(pk, evk, sk) \leftarrow \mathsf{HE.KeyGen}(1^\kappa)$ where $\kappa \in \mathbb{N}$ is the *security parameter*. Three keys are generated: a public key pk, which can be used for the encryption of messages; a secret key sk used for decryption; and an evaluation key evk that may aid in evaluating the circuit on the encrypted state. The keys pk and evk are announced publicly, while sk is kept secret.

$c \leftarrow \mathsf{HE.Enc}_{pk}(x)$ for some one-bit message $x \in \{0,1\}$. This probabilistic procedure outputs a ciphertext c, using the public key pk.

$c' \leftarrow \mathsf{HE.Eval}^{\mathsf{C}}_{evk}(c_1,\dots,c_\ell)$ uses the evaluation key to output some ciphertext c' which decrypts to the evaluation of circuit C on the decryptions of $c_1,\dots,c_\ell$. We will often think of Eval as an evaluation of a function f instead of some canonical circuit for f, and write $\mathsf{HE.Eval}^{f}_{evk}(c_1,\dots,c_\ell)$ in this case.

$x' \leftarrow \mathsf{HE.Dec}_{sk}(c')$ outputs a message $x' \in \{0,1\}$, using the secret key sk. (In general, the decryption functionality could be defined to also sometimes output $x' = \perp$, representing an undefined output. For the purpose of defining a scheme that is correct for an honest evaluator and that maintains secrecy against a dishonest evaluator, it suffices to only consider $x' \in \{0,1\}$.)

In principle, $\mathsf{HE.Enc}_{pk}$ can only encrypt single bits. When encrypting an n-bit message $x \in \{0,1\}^n$, we encrypt the message bit-by-bit, applying the encryption procedure n times. We sometimes abuse the notation $\mathsf{HE.Enc}_{pk}(x)$ to denote this bitwise encryption of the string x. Similarly, we define $\mathsf{HE.Dec}_{sk}$ only for ciphertexts that decrypt to a single bit. If we want to evaluate a more general circuit that outputs m bits, then we have to apply $\mathsf{HE.Dec}_{sk}$ bit-by-bit (m times in total) to all ciphertext output bits separately. Note that this assumes that the output of $\mathsf{HE.Eval}^{\mathsf{C}}$ consists of m distinct parts: schemes with this property are called *divisible*, because they allow bit-by-bit decryption.

For HE to be a $\mathcal{C}$-homomorphic encryption scheme for some class of circuits $\mathcal{C}$, we require *correctness* in the sense that for any circuit $\mathsf{C} \in \mathcal{C}$, there exists a negligible[5] function η such that, for any input x,

$$\Pr[\mathsf{HE.Dec}_{sk}(\mathsf{HE.Eval}^{\mathsf{C}}_{evk}(\mathsf{HE.Enc}_{pk}(x))) \neq \mathsf{C}(x)] \leq \eta(\kappa)\,.$$

In this article, we assume for clarity of exposition that classical schemes HE are perfectly correct, and that it is possible to immediately decrypt after encrypting (without doing any evaluation).

Another desirable property is *compactness*, which states that the complexity of the decryption function should not depend on the size of the circuit. More formally, a $\mathcal{C}$-homomorphic encryption scheme is compact if there exists a polynomial p such that for any security parameter κ, any circuit $\mathsf{C} \in \mathcal{C}$, and any ciphertext c, the time complexity of running $\mathsf{HE.Dec}_{sk}$ on the output of $\mathsf{HE.Eval}^{\mathsf{C}}(c)$ is at most $p(\kappa)$.

A compact $\mathcal{C}$-homomorphic encryption scheme where $\mathcal{C}$ is the class of all circuits is called *fully* homomorphic. If $\mathcal{C}$ is only a subset of all possible circuits (e. g., all circuits with no multiplication gates) or if the scheme is not compact, it is considered to be a *somewhat* homomorphic scheme. Finally, a *leveled* fully homomorphic scheme is (compact and) homomorphic for all circuits up to a variable depth L, which is supplied as an extra argument to the key generation function [55]. For leveled FHE, the definition of compactness is slightly more subtle: there should exist a polynomial p such that for any set of parameters (κ, L), any polynomial q, any circuit C with at most $q(L)$ gates, and any ciphertext c, the output size of

[5] A *negligible function* η is a function such that for every positive integer d, $\eta(n) < 1/n^d$ for big enough n.

$\mathsf{HE.Eval}^{\mathsf{C}}(c)$ is at most $p(\kappa)$. Note that the polynomial p should be independent of the number of gates in the circuit (given by $q(L)$). Our quantum scheme TP will be leveled fully homomorphic.

We will use the notation $\widetilde{x}$ to denote the result of running $\mathsf{HE.Enc}_{pk}(x)$. That is, $\mathsf{Dec}_{sk}(\widetilde{x}) = x$ with overwhelming probability. In our construction, we will often deal with multiple classical key sets $(pk_i, evk_i, sk_i)_{i \in I}$ indexed by some set I. In that case, we use the notation $\widetilde{x}^{[i]}$ to denote the result of $\mathsf{HE.Enc}_{pk_i}(x)$, in order to avoid confusion. Also note that, e. g., pk_i does *not* refer to the ith bit of the public key. In case we want to refer to the ith bit of some string s, we will use the notation $s[i]$.

When working with multiple key sets, it will often be necessary to transform an already encrypted message $\widetilde{x}^{[i]}$ into an encryption $\widetilde{x}^{[j]}$ using a different key set $j \neq i$. To achieve this transformation, we define the procedure $\mathsf{HE.Rec}_{i \to j}$ that can always be used for this *recryption* task as long as we have access to an encrypted version $\widetilde{sk_i}^{[j]}$ of the old secret key sk_i. Effectively, $\mathsf{HE.Rec}_{i \to j}$ homomorphically evaluates the decryption of $\widetilde{x}^{[i]}$:

$$\mathsf{HE.Rec}_{i \to j}(\widetilde{x}^{[i]}) := \mathsf{HE.Eval}^{\mathsf{HE.Dec}}_{evk_j}\left(\widetilde{sk_i}^{[j]}, \mathsf{HE.Enc}_{pk_j}(\widetilde{x}^{[i]})\right).$$

2.2.2 The quantum setting

A quantum homomorphic encryption scheme QHE, as defined in [16], is a natural extension of the classical case, and differs from it in only a few aspects. The secret and public keys are still classical, but the evaluation key is allowed to be a quantum state. This means that the evaluation key is not necessarily reusable, and can be consumed during the evaluation procedure. The messages to be encrypted are qubits instead of bits, and the evaluator should be able to evaluate quantum circuits on them.

All definitions given above carry over quite naturally to the quantum setting (see also [16]):

$(pk, \rho_{evk}, sk) \leftarrow \mathsf{QHE.KeyGen}(1^\kappa)$ where $\kappa \in \mathbb{N}$ is the security parameter. In contrast to the classical case, the evaluation key is a quantum state.

$\sigma \leftarrow \mathsf{QHE.Enc}_{pk}(\rho)$ produces, for every valid public key pk and input state ρ from some single-qubit message space, to a quantum cipherstate σ in some cipherspace.

$\sigma' \leftarrow \mathsf{QHE.Eval}^{\mathsf{C}}_{\rho_{evk}}(\sigma)$ represents the evaluation of a circuit C. If C requires n input qubits, then σ should be the result of applying $\mathsf{QHE.Enc}^{\otimes n}$ to the plaintext input. The evaluation function maps it to a state in the n'-fold tensor product of the ciphertext output space, where n' is the number of qubits that C would output. The evaluation key ρ_{evk} is consumed in the process.

$\rho' \leftarrow \mathsf{QHE.Dec}_{sk}(\sigma')$ maps a single state σ' from the output space to a quantum state ρ' in the message space. Note that if the evaluation procedure $\mathsf{QHE.Eval}$ evaluated a circuit with n' output wires, then $\mathsf{QHE.Dec}$ needs to be run n' times.

Again, we require the encryption and decryption procedure to act on single qubits, and hence we only consider divisible schemes. Divisible schemes form a subclass to the more general notion of *indivisible schemes* [16], where an auxiliary quantum register may be built up for the entire state, and the output cipherstate is decrypted as a whole. For divisible schemes, the auxiliary quantum register is empty.

2.2.3 Quantum security

The notion of security that we aim for is that of *indistinguishability under chosen-plaintext attacks*, where the attacker may have quantum computational powers (q-IND-CPA). This security notion was introduced in [16, Definition 3.3] (see [27] for a similar notion of the security of classical schemes against quantum attackers) and ensures semantic security [3]. We restate it here for completeness.

Definition 2.1. [16] The *quantum CPA indistinguishability experiment* with respect to a scheme QHE and a quantum polynomial-time adversary $\mathscr{A} = (\mathscr{A}_1, \mathscr{A}_2)$, denoted by $\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{QHE}}(\kappa)$, is defined by the following procedure:

1. $\mathsf{KeyGen}(1^\kappa)$ is run to obtain keys (pk, ρ_{evk}, sk).

2. Adversary $\mathscr{A}_1$ is given (pk, ρ_{evk}) and outputs a quantum state on $\mathcal{M} \otimes \mathcal{E}$.

3. For $r \in \{0,1\}$, let $\Xi^{\mathsf{cpa},r}_{\mathsf{QHE}} : D(\mathcal{M}) \to D(\mathcal{C})$ be:

$$\Xi^{\mathsf{cpa},0}_{\mathsf{QHE}}(\rho) = \mathsf{QHE.Enc}_{pk}(|0\rangle\langle 0|) \qquad \text{and} \qquad \Xi^{\mathsf{cpa},1}_{\mathsf{QHE}}(\rho) = \mathsf{QHE.Enc}_{pk}(\rho)\,.$$

   A random bit $r \in \{0,1\}$ is chosen and $\Xi^{\mathsf{cpa},r}_{\mathsf{QHE}}$ is applied to the state in $\mathcal{M}$ (the output being a state in $\mathcal{C}$).

4. Adversary $\mathscr{A}_2$ obtains the system in $\mathcal{C} \otimes \mathcal{E}$ and outputs a bit r'.

5. The output of the experiment is defined to be 1 if $r' = r$ and 0 otherwise. In case $r = r'$, we say that $\mathscr{A}$ *wins* the experiment.

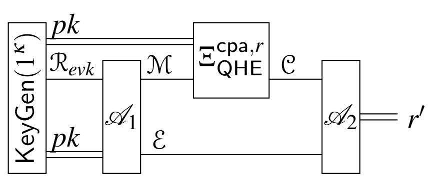

Figure 1: [16, reproduced with permission of the authors] The quantum CPA indistinguishability experiment $\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{QHE}}(\kappa)$. Double lines represent classical information flow, and single lines represent quantum information flow. The adversary $\mathscr{A}$ is split up into two separate algorithms $\mathscr{A}_1$ and $\mathscr{A}_2$, which share a working memory represented by the quantum state in register $\mathcal{E}$. For the definition of $\Xi^{\mathsf{cpa},r}_{\mathsf{QHE}}$, see step 3 in Definition 2.1.

The game $\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{QHE}}(\kappa)$ is depicted in Figure 1. Informally, the challenger randomly chooses whether to encrypt some message, chosen by the adversary, or instead to encrypt the state $|0\rangle\langle 0|$. The adversary has to guess which of the two happened. If he cannot do so with more than negligible advantage, the encryption procedure is considered to be q-IND-CPA secure:

Definition 2.2. [16, Definition 3.3] A (classical or quantum) homomorphic encryption scheme S is *q-IND-CPA* secure if for any quantum polynomial-time adversary $\mathscr{A} = (\mathscr{A}_1, \mathscr{A}_2)$ there exists a negligible function η such that:

$$\Pr[\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{S}}(\kappa) = 1] \leq \frac{1}{2} + \eta(\kappa).$$

Analogously to $\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{S}}(\kappa)$, in the game $\mathsf{PubK}^{\mathsf{cpa-mult}}_{\mathscr{A},\mathsf{S}}(\kappa)$, the adversary can give multiple messages to the challenger, which are either all encrypted, or all replaced by zeros. Broadbent and Jeffery [16] show that these notions of security are equivalent.

2.3 Garden-hose complexity

The *garden-hose model* is a model of communication complexity which was introduced by Buhrman, Fehr, Schaffner and Speelman [17] to study a protocol for position-based quantum cryptography. The model recently saw new use, when Speelman [53] used it to construct new protocols for the task of instantaneous non-local quantum computation, thereby breaking a wider class of schemes for position-based quantum cryptography. (Besides the garden-hose model, this construction used tools from secure delegated computation. These techniques were first used in the setting of instantaneous non-local quantum computation by Broadbent [14].)

We will not explain the garden-hose model thoroughly, but instead give a short overview. The garden-hose model involves two parties, Alice with input x and Bob with input y, that jointly want to compute a function f. To do this computation, they are allowed to use garden hoses to link up pipes that run between them, one-to-one, in a way which depends on their local inputs. Alice also has a water tap, which she connects to one of the pipes. Whenever $f(x,y) = 0$, the water has to exit at an open pipe on Alice's side, and whenever $f(x,y) = 1$ the water should exit on Bob's side.

The applicability of the garden-hose model to our setting stems from a close correspondence between protocols in the garden-hose model and teleporting a qubit back-and-forth; the "pipes" correspond to EPR pairs and the "garden hoses" can be translated into Bell measurements. Our construction of the gadgets in Section 5.3 will depend on the number of pipes needed to compute the decryption function $\mathsf{HE.Dec}$ of a classical fully homomorphic encryption scheme. It will turn out that any log-space computable decryption function allows for efficiently constructable polynomial-size gadgets.

3 The TP scheme

Our scheme TP (for teleportation) is an extension of the scheme CL presented in [16]: the quantum state is encrypted using a quantum one-time pad, and Clifford gates are evaluated simply by performing the gate on the encrypted state and then homomorphically updating the encrypted keys to the pad. The new scheme TP, like AUX, includes additional resource states (gadgets) in the evaluation key. These gadgets can be used to immediately correct any P errors that might be present after the application of a T gate. The size of the evaluation key thus grows linearly with the upper bound to the number of T gates in the circuit: for every T gate the evaluation key contains one gadget, along with some classical information on how to use that gadget.

3.1 Gadget

In this section we only give the general form of the gadget, which suffices to prove security. The explanation on how to construct these gadgets, which depend on the decryption function of the classical homomorphic scheme HE.Dec, is deferred to Section 5.

Recall that when a T gate is applied to the state $\mathsf{X}^a\mathsf{Z}^b|\psi\rangle$, an unwanted P error may occur since $\mathsf{T}\mathsf{X}^a\mathsf{Z}^b = \mathsf{P}^a\mathsf{X}^a\mathsf{Z}^b\mathsf{T}$. If a is known, this error can easily be corrected by applying $\mathsf{P}^\dagger$ whenever $a = 1$. However, as we will see, the evaluating party only has access to some encrypted version $\widetilde{a}$ of the key a, and hence is not able to decide whether or not to correct the state.

We show how the key generator can create a gadget ahead of time that corrects the state, conditioned on a, when the qubit $\mathsf{P}^a\mathsf{X}^a\mathsf{Z}^b\mathsf{T}|\psi\rangle$ is teleported through it. The gadget will not reveal any information about whether or not a P gate was present before the correction. Note that the value of a is completely unknown to the key generator, so the gadget cannot depend on it. Instead, the gadget will depend on the secret key sk, and the evaluator will use it in a way that depends on $\widetilde{a}$.

The intuition behind our construction is as follows. A gadget consists of a set of fully entangled pairs that are crosswise linked up in a way that depends on the secret key sk and the decryption function of the classical homomorphic scheme HE. If the decryption function HE.Dec is simple enough, i. e., computable in logarithmic space or by low-depth binary circuit, the size of this state is polynomial in the security parameter.

Some of these entangled pairs have an extra inverse phase gate applied to them. Note that teleporting any qubit $\mathsf{X}^a\mathsf{Z}^b|\psi\rangle$ through, for example, $(\mathsf{P}^\dagger \otimes \mathsf{I})|\Phi^+\rangle$, effectively applies an inverse phase gate to the qubit, which ends up in the state $\mathsf{X}^{a'}\mathsf{Z}^{b'}\mathsf{P}^\dagger|\psi\rangle$, where the new Pauli corrections a',b' depend on a,b and the outcome of the Bell measurement.

When wanting to remove an unwanted phase gate, the evaluator of the circuit teleports a qubit through this gadget state in a way which is specified by $\widetilde{a}$. The gadget state is constructed so that the qubit follows a path through this gadget which passes an inverse phase gate if and only if $\mathsf{HE.Dec}_{sk}(\widetilde{a})$ equals 1. The Pauli corrections can then be updated using the homomorphically-encrypted classical information and the measurement outcomes.

Specification of gadget

Assume HE.Dec is computable in space logarithmic in the security parameter κ. In Section 5 we will show that there exists an efficient algorithm $\mathsf{TP.GenGadget}_{pk'}(sk)$ which produces a gadget, i. e., a quantum state $\Gamma_{pk'}(sk)$ of the form specified in this section.

The gadget will be able to remove a single phase gate P^a, using only knowledge of $\widetilde{a}$, where $\widetilde{a}$ decrypts to a under the secret key sk. The public key pk' is used to encrypt all classical information associated with the gadget.

The quantum part of the gadget consists of $2m$ qubits, with m some number which is polynomial in the security parameter κ. Let $\{(s_1,t_1),(s_2,t_2),\ldots,(s_m,t_m)\}$ be disjoint pairs in $\{1,2,\ldots,2m\}$, and let $p \in \{0,1\}^m$ be a string of m bits. Let $g(sk)$ be a shorthand for the tuple of both of these, together with the secret key sk;

$$g(sk) := (\{(s_1,t_1),(s_2,t_2),\ldots,(s_m,t_m)\}, p, sk)\,.$$

The tuple $g(sk)$ is the classical information that determines the structure of the gadget as a function of the secret key sk. It will be the case that the variables $(s_1,t_1),(s_2,t_2),\ldots,(s_m,t_m)$ and p that describe the gadget structure are themselves determined by sk, and therefore the variable s_i, for example, should always be read in context as $s_i(sk)$.

For any bitstring $x,z \in \{0,1\}^m$, define the quantum state

$$\gamma_{x,z}\big(g(sk)\big) := \bigotimes_{i=1}^{m} \mathsf{X}^{x[i]}\mathsf{Z}^{z[i]}\left(\mathsf{P}^\dagger\right)^{p[i]}\left|\Phi^+\right\rangle\!\left\langle\Phi^+\right|_{s_it_i}\mathsf{P}^{p[i]}\mathsf{Z}^{z[i]}\mathsf{X}^{x[i]}\,.$$

(Here the single-qubit gates are applied to s_i, the first qubit of the entangled pair.) This quantum state is a collection of maximally-entangled pairs of qubits, some with an extra inverse phase gate applied, where the pairs are determined by the disjoint pairs $\{(s_1,t_1),(s_2,t_2),\ldots,(s_m,t_m)\}$ chosen earlier. The entangled pairs have arbitrary Pauli operators applied to them, described by the bitstrings x and z.

Note that, no matter the choice of gadget structure, averaging over all possible x,z gives the completely mixed state on $2m$ qubits,

$$\frac{1}{2^{2m}}\sum_{x,z\in\{0,1\}^m}\gamma_{x,z}\big(g(sk)\big) = \frac{\mathbb{I}_{2^{2m}}}{2^{2m}}\,.$$

This property will be important in the security proof; intuitively it shows that these gadgets do not reveal any information about sk whenever x and z are encrypted with a secure classical encryption scheme.

The entire gadget then is given by

$$\Gamma_{pk'}(sk) = \rho\big(\mathsf{HE.Enc}_{pk'}\big(\mathsf{g}(sk)\big)\big)\otimes\frac{1}{2^{2m}}\sum_{x,z\in\{0,1\}^m}\rho\big(\mathsf{HE.Enc}_{pk'}(x,z)\big)\otimes\gamma_{x,z}\big(g(sk)\big)\,,$$

where $\mathsf{g}(sk)$ denotes a canonical encoding of $g(sk)$ into a bit string. The length of $g(sk)$ (and thus of $\mathsf{g}(sk)$) is not dependent on the secret key: the number of qubits m and the size of sk itself are completely determined by the choice of protocol HE and the security parameter κ.

To summarize, the gadget consists of a quantum state $\gamma_{x,z}\big(g(sk)\big)$, instantiated with randomly chosen x,z, the classical information denoting the random choice of x,z, and the other classical information $\mathsf{g}(sk)$ which specifies the gadget. All classical information is homomorphically encrypted with a public key pk'.

Since this gadget depends on the secret key sk, simply encrypting this information using the public key corresponding to sk would not be secure, unless we assume that $\mathsf{HE.Dec}$ is circularly secure. In order to avoid the requirement of circular security, we will always use a fresh, independent key pk' to encrypt this information. The evaluator will have to do some recrypting before he is able to use this information, but otherwise using independent keys does not complicate the construction much. More details on how the evaluation procedure deals with the different keys is provided in Section 3.4.

Usage of gadget

The gadget is used by performing Bell measurements between pairs of its qubits, together with an input qubit that needs a correction, without having knowledge of the structure of the gadget.

The choice of measurements is generated by an efficient classical algorithm $\mathsf{TP.GenMeasurement}(\widetilde{a})$ which produces a list M containing m disjoint pairs of elements in $\{0,1,2,\ldots,2m\}$ that only depend on ciphertext $\widetilde{a}$. Here the labels 1 to $2m$ refer to the qubits that make up a gadget and 0 is the label of the qubit with the possible P error. The pairs represent which qubits will be connected through Bell measurements; note that all but a single qubit will be measured according to M.

Consider an input qubit, in some arbitrary state $\mathsf{P}^a|\psi\rangle$, i. e., the qubit has an extra phase gate if $a = 1$. Let $\widetilde{a}$ be an encrypted version of a, such that $a = \mathsf{HE.Dec}_{sk}(\widetilde{a})$. Then the evaluator performs Bell measurements on $\Gamma_{pk'}(sk)$ and the input qubit, according to $M \leftarrow \mathsf{TP.GenMeasurement}(\widetilde{a})$. By construction, one out the $2m+1$ qubits is still unmeasured. This qubit will be in the state $\mathsf{X}^{a'}\mathsf{Z}^{b'}|\psi\rangle$, for some a' and b', both of which are functions of the specification of the gadget, the measurement choices which depend on $\widetilde{a}$, and the outcomes of the teleportation measurements. Also see Section 3.4 and Appendix A for a more in-depth explanation of how the accompanying classical information is updated.

Intuitively, the "path" the qubit takes through the gadget state, goes through one of the fully entangled pairs with an inverse phase gate whenever $\mathsf{HE.Dec}_{sk}(\widetilde{a}) = 1$, and avoids all such pairs whenever $\mathsf{HE.Dec}_{sk}(\widetilde{a}) = 0$.

3.2 Key generation

Using the classical $\mathsf{HE.KeyGen}$ as a subroutine to create multiple classical homomorphic keysets, we generate a classical secret and public key, and a classical-quantum evaluation key that contains L gadgets, allowing evaluation of a circuit containing up to L T gates. Every gadget depends on a different secret key, and its classical information is always encrypted using the next public key. The key generation procedure $\mathsf{TP.KeyGen}(1^\kappa, 1^L)$ is defined as follows:

1. For $i = 0$ to L: execute $(pk_i, evk_i, sk_i) \leftarrow \mathsf{HE.KeyGen}(1^\kappa)$ to obtain $L+1$ independent classical homomorphic key sets.

2. Set the public key to be the tuple $(pk_i)_{i=0}^L$.

3. Set the secret key to be the tuple $(sk_i)_{i=0}^L$.

4. For $i = 0$ to $L-1$: Run the procedure $\mathsf{TP.GenGadget}_{pk_{i+1}}(sk_i)$ to create the gadget $\Gamma_{pk_{i+1}}(sk_i)$ as described in Section 3.1.

5. Set the evaluation key to be the set of all gadgets created in the previous step (including their encrypted classical information), plus the tuple $(evk_i)_{i=0}^L$. The resulting evaluation key is the quantum state
$$\bigotimes_{i=0}^{L-1} \big(\Gamma_{pk_{i+1}}(sk_i) \otimes |evk_i\rangle\langle evk_i|\big) \otimes |evk_L\rangle\langle evk_L|.$$

3.3 Encryption

The encryption procedure $\mathsf{TP.Enc}$ is identical to $\mathsf{CL.Enc}$, using the first public key pk_0 for the encryption of the one-time-pad keys. We restate it here for completeness.

Every single-qubit state σ is encrypted separately with a quantum one-time pad, and the pad key is (classically) encrypted and appended to the quantum encryption of σ, resulting in the classical-quantum state

$$\sum_{a,b\in\{0,1\}} \frac{1}{4}\rho(\mathsf{HE.Enc}_{pk_0}(a), \mathsf{HE.Enc}_{pk_0}(b)) \otimes \mathsf{X}^a\mathsf{Z}^b\sigma\mathsf{Z}^b\mathsf{X}^a.$$

3.4 Circuit evaluation

Consider a circuit with n wires. The evaluation of the circuit on the encrypted data is carried out one gate at a time.

Recall that our quantum circuit is written using a gate set that consists of the Clifford group generators $\{\mathsf{H}, \mathsf{P}, \mathsf{CNOT}\}$ and the T gate. A Clifford gate may affect multiple wires at the same time, while T gates can only affect a single qubit. Before the evaluation of a single gate U, the encryption of an n-qubit state ρ is of the form

$$\left(\mathsf{X}^{a_1}\mathsf{Z}^{b_1} \otimes \cdots \otimes \mathsf{X}^{a_n}\mathsf{Z}^{b_n}\right)\rho\left(\mathsf{Z}^{b_1}\mathsf{X}^{a_1} \otimes \cdots \otimes \mathsf{Z}^{b_n}\mathsf{X}^{a_n}\right).$$

The evaluating party holds the encrypted versions $\widetilde{a_1}^{[i]}, \ldots, \widetilde{a_n}^{[i]}$ and $\widetilde{b_1}^{[i]}, \ldots, \widetilde{b_n}^{[i]}$, with respect to the ith key set for some i (initially, $i = 0$). The goal is to obtain a quantum encryption of the state $U\rho U^\dagger$, such that the evaluator can homomorphically compute the encryptions of the new keys to the quantum one-time pad. If U is a Clifford gate, these encryptions will still be in the ith key. If U is a T gate, then all encryptions are transferred to the $(i+1)$th key during the process.

- If U is a Clifford gate, we proceed exactly as in $\mathsf{CL.Eval}$. The gate U is simply applied to the encrypted qubit, and since U commutes with the Pauli group, the evaluator only needs to update the encrypted keys in a straightforward way, given by the commutation rules from Section 2.1.

- If $U = \mathsf{T}$, the evaluator should start out by applying a T gate to the appropriate wire w. Afterwards, the qubit at wire w is in the state

  $$\left(\mathsf{P}^{a_w}\mathsf{X}^{a_w}\mathsf{Z}^{b_w}\mathsf{T}\right)\rho_w\left(\mathsf{T}^\dagger\mathsf{Z}^{b_w}\mathsf{X}^{a_w}(\mathsf{P}^\dagger)^{a_w}\right).$$

  In order to remove the P error, the evaluator uses one gadget $\Gamma_{pk_{i+1}}(sk_i)$ from the evaluation key; he possesses the classical information $\widetilde{a_w}^{[i]}$, encrypted with the ith key and therefore matching this gadget, to be able to compute measurements $M \leftarrow \mathsf{TP.GenMeasurement}(\widetilde{a_w}^{[i]})$ and performs the measurements on the pairs given by M. Afterwards, using his own measurement outcomes, the classical information accompanying the gadget (encrypted using pk_{i+1}), and the recryptions of $\widetilde{a_w}^{[i]}$ and $\widetilde{b_w}^{[i]}$ into $\widetilde{a_w}^{[i+1]}$ and $\widetilde{b_w}^{[i+1]}$, the evaluator homomorphically computes the new keys $\widetilde{a'_w}^{[i+1]}$ and $\widetilde{b'_w}^{[i+1]}$. See also Figure 2 and Appendix A. After these computations, he should also recrypt the keys of all other wires into the $(i+1)$th key set.

At the end of the evaluation of some circuit C containing k T gates, the evaluator holds a one-time-pad encryption of the state $\mathsf{C}|\psi\rangle$, together with the keys to the pad, classically encrypted in the kth key. The last step is to recrypt (in $L-k$ steps) this classical information into the Lth (final) key. Afterwards, the quantum state and the key encryptions are sent to the decrypting party. Any unused gadgets are discarded.

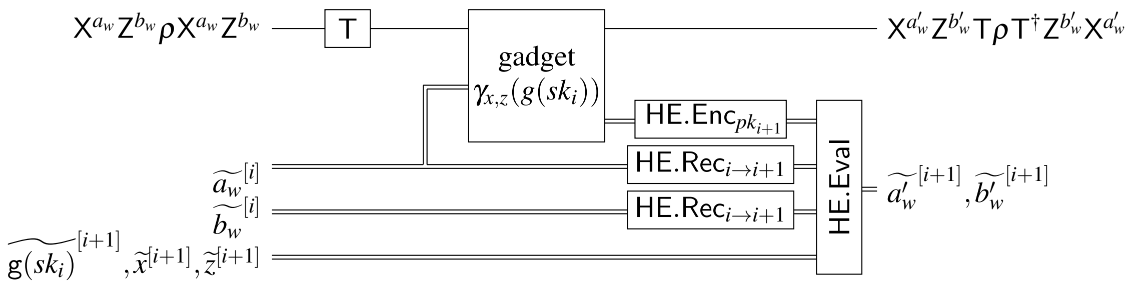

Figure 2: The homomorphic evaluation of the $(i+1)$th T gate of the circuit. The gadget is consumed during the process. After the use of the gadget, the evaluator encrypts his own classical information (including measurement outcomes) in order to use it in the homomorphic computation of the new keys. $\mathsf{HE.Eval}$ evaluates this fairly straightforward computation that consists mainly of looking up values in a list and adding them modulo 2. Note that $\widetilde{sk_i}^{[i+1]}$, needed for the recryption procedures, is contained in the evaluation key and that the keys of the wires that are not shown are also recrypted.

3.5 Decryption

The decryption procedure is identical to $\mathsf{CL.Dec}$. For each qubit, $\mathsf{HE.Dec}_{sk_L}$ is run twice in order to retrieve the keys to the quantum pad. The correct Pauli operator can then be applied to the quantum state in order to obtain the desired state $\mathsf{C}|\psi\rangle$.

The decryption procedure is fairly straightforward, and its complexity does not depend on the circuit that was evaluated. This is formalized in a compactness theorem for the TP scheme:

Theorem 3.1. *If* HE *is compact, then* TP *is compact.*

Proof. Note that because the decryption only involves removing a one-time pad from the quantum ciphertext produced by the circuit evaluation, this decryption can be carried out a single qubit at a time. By compactness of HE, there exists a polynomial $p(\kappa)$ such that for any function f and any ciphertext c, the output size of $\mathsf{HE.Eval}^f(c)$ is at most $p(\kappa)$. Since the keys to the quantum one-time pad of any wire w are two single bits encrypted with the classical HE scheme, decrypting the keys for one wire requires at most $2p(\kappa)$ steps. Obtaining the qubit then takes at most two steps more for (conditionally) applying X^{a_w} and Z^{b_w}. The total number of steps is polynomial in κ and independent of C (and in particular, independent of L), so we conclude that TP is compact. □

4 Security of TP

In order to guarantee the privacy of the input data, we need to argue that an adversary that does not possess the secret key cannot learn anything about the data with more than negligible probability. To this end, we show that TP is q-IND-CPA secure, i. e., no polynomial-time quantum adversary can tell the difference between an encryption of a real message and an encryption of $|0\rangle\langle 0|$, even if he gets to choose the message himself (recall the definition of q-IND-CPA security from Section 2.2.3). Like in the security proofs in [16], we use a reduction argument to relate the probability of being able to distinguish between the two encryptions to the probability of winning an indistinguishability experiment for the classical HE, which we assume to be close to $1/2$. The aim of this section is to prove the following theorem:

Theorem 4.1. *If* HE *is q-IND-CPA secure, then* TP *is q-IND-CPA secure for circuits containing up to polynomially (in κ) many* T *gates.*

In order to prove Theorem 4.1, we first prove that an efficient adversary's performance in the indistinguishability game is only negligibly different when he receives a real evaluation key with real gadgets, versus just a completely mixed quantum state with encryptions of 0's accompanying them (Corollary 4.3). Then we argue that without the evaluation key, an adversary does not receive more information than in the indistinguishability game for the scheme CL, which has already been shown to be q-IND-CPA secure whenever HE is.

We start with defining a sequence of variations on the TP scheme. For $\ell \in \{0,\ldots,L\}$, let $\mathsf{TP}^{(\ell)}$ be identical to TP, except for the key generation procedure: $\mathsf{TP}^{(\ell)}.\mathsf{KeyGen}$ replaces, for every $i \geq \ell$, all classical information accompanying the ith gadget with the all-zero string before encrypting it. For any number i, define the shorthand

$$g_i := g(sk_i).$$

Similarly, $\mathsf{g}_i := \mathsf{g}(sk_i)$, the canonical bit string representation of g_i. As seen in Section 3.1, the length of the classical information does not depend on sk_i itself, so a potential adversary cannot gain any information about sk_i just from this encrypted string. In summary, the evaluation key produced by $\mathsf{TP}^{(\ell)}.\mathsf{KeyGen}(1^\kappa, 1^L)$ is

$$\bigotimes_{i=0}^{\ell-1}\Big(|evk_i\rangle\langle evk_i| \otimes \Gamma_{pk_{i+1}}(sk_i)\Big)\otimes$$
$$\bigotimes_{i=\ell}^{L-1}\Big(|evk_i\rangle\langle evk_i| \otimes \rho(\mathsf{HE.Enc}_{pk_{i+1}}(0^{|\mathsf{g}_i|}))\otimes \frac{1}{2^{2m}}\sum_{x,z\in\{0,1\}^m}\rho(\mathsf{HE.Enc}_{pk_{i+1}}(0^m,0^m))\otimes \gamma_{x,z}(g_i)\Big)\,.$$

The secret key and public key remain the same as in TP. Intuitively, one can view $\mathsf{TP}^{(\ell)}$ as the scheme that provides only ℓ usable gadgets in the evaluation key. Note that $\mathsf{TP}^{(L)} = \mathsf{TP}$, and that in $\mathsf{TP}^{(0)}$, only the classical evaluation keys remain, since without the encryptions of the classical x and z, the quantum part of the gadget is just the completely mixed state. That is, we can rewrite the final line of the previous equation as

$$\begin{aligned}&\frac{1}{2^{2m}}\sum_{x,z\in\{0,1\}^m}\rho(\mathsf{HE.Enc}_{pk_{i+1}}(0^m,0^m))\otimes \gamma_{x,z}(g_i)\\ &\qquad = \rho(\mathsf{HE.Enc}_{pk_{i+1}}(0^m,0^m))\otimes \frac{\mathbb{I}_{2^{2m}}}{2^{2m}}\,.\end{aligned} \tag{4.1}$$

With the definitions of the new schemes, we can lay out the steps to prove Theorem 4.1 in more detail. First, we show that in the quantum CPA indistinguishability experiment, any efficient adversary interacting with $\mathsf{TP}^{(\ell)}$ only has negligible advantage over an adversary interacting with $\mathsf{TP}^{(\ell-1)}$, i. e. the scheme where the classical information $\mathsf{g}_{\ell-1}$ is removed (Lemma 4.2). By iteratively applying this argument, we are able to argue that the advantage of an adversary who interacts with $\mathsf{TP}^{(L)}$ over one who interacts with $\mathsf{TP}^{(0)}$ is also negligible (Corollary 4.3). Finally, we conclude the proof by arguing that $\mathsf{TP}^{(0)}$ is q-IND-CPA secure by comparison to the CL scheme.

Lemma 4.2. *Let $0 < \ell \leq L$. If HE is q-IND-CPA secure, then for any quantum polynomial-time adversary $\mathscr{A} = (\mathscr{A}_1, \mathscr{A}_2)$, there exists a negligible function η such that*

$$\Pr[\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{TP}^{(\ell)}}(\kappa) = 1] - \Pr[\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{TP}^{(\ell-1)}}(\kappa) = 1] \leq \eta(\kappa).$$

Proof. The difference between schemes $\mathsf{TP}^{(\ell)}$ and $\mathsf{TP}^{(\ell-1)}$ lies in whether the gadget state

$$\gamma_{x_{\ell-1},z_{\ell-1}}(g_{\ell-1})$$

is supplemented with its classical information $\widetilde{\mathsf{g}_{\ell-1}}, \widetilde{x_{\ell-1}}, \widetilde{z_{\ell-1}}$, or just with an encryption of $0^{|\mathsf{g}_{\ell-1}|+2m}$.

Let $\mathscr{A} = (\mathscr{A}_1, \mathscr{A}_2)$ be an adversary for the game $\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{TP}^{(\ell)}}(\kappa)$. We will define an adversary $\mathscr{A}' = (\mathscr{A}'_1, \mathscr{A}'_2)$ for $\mathsf{PubK}^{\mathsf{cpa-mult}}_{\mathscr{A}',\mathsf{HE}}(\kappa)$ that will either simulate the game

$$\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{TP}^{(\ell)}}(\kappa) \qquad \text{or} \qquad \mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{TP}^{(\ell-1)}}(\kappa).$$

Which game is simulated will depend on some $s \in_R \{0,1\}$ that is unknown to $\mathscr{A}'$ himself. Using the assumption that HE is q-IND-CPA secure, we are able to argue that $\mathscr{A}'$ is unable to recognize which of the two schemes was simulated, and this fact allows us to bound the difference in success probabilities between the security games of $\mathsf{TP}^{(\ell)}$ and $\mathsf{TP}^{(\ell-1)}$. The structure of this proof is very similar to, e. g., Lemma 5.3 in [16]. The adversary $\mathscr{A}'$ acts as follows (see also Figure 3):

$\mathscr{A}'_1$ takes care of most of the key generation procedure: he generates the classical key sets 0 through $\ell-1$ himself, generates the random strings $x_0, z_0, \ldots, x_{\ell-1}, z_{\ell-1}$, and constructs the gadgets $\gamma_{x_0,z_0}(g_0), \ldots, \gamma_{x_{\ell-1},z_{\ell-1}}(g_{\ell-1})$ and their classical information $\mathsf{g}_0, \ldots, \mathsf{g}_{\ell-1}$. He encrypts the classical information using the appropriate public keys. Only $\mathsf{g}_{\ell-1}$, $x_{\ell-1}$ and $z_{\ell-1}$ are left unencrypted. Instead of encrypting these strings himself using pk_ℓ, $\mathscr{A}'_1$ sends the strings for encryption to the challenger. Whether the challenger really encrypts $\mathsf{g}_{\ell-1}$, $x_{\ell-1}$ and $z_{\ell-1}$ or replaces the strings with a string of zeros, determines which of the two schemes is simulated. $\mathscr{A}'$ is unaware of the random choice of the challenger.

The adversary $\mathscr{A}'_1$ also generates the extra padding inputs that correspond to the already-removed gadgets ℓ up to $L-1$. Since these gadgets consist of all-zero strings encrypted with independently chosen public keys that are not used anywhere else, together with a completely mixed quantum state (as shown in equation (4.1)), the adversary can generate them without needing any extra information.

$\mathscr{A}'_2$ feeds the evaluation key and public key, just generated by $\mathscr{A}'_1$, to $\mathscr{A}_1$ in order to obtain a chosen message $\mathcal{M}$ (plus the auxiliary state $\mathcal{E}$). He then picks a random $r \in_R \{0,1\}$ and replaces $\mathcal{M}$ by 0 if and only if $r = 0$. He encrypts the result according to the $\mathsf{TP.Enc}$ procedure (using the public key $(pk_i)_{i=0}^{L}$ received from $\mathscr{A}'_1$), and gives the encrypted state, plus $\mathcal{E}$, to $\mathscr{A}_2$, who outputs r' in an attempt to guess r. $\mathscr{A}'_2$ now outputs 1 if and only if the guess by $\mathscr{A}$ was correct, i. e., $r \equiv r'$.

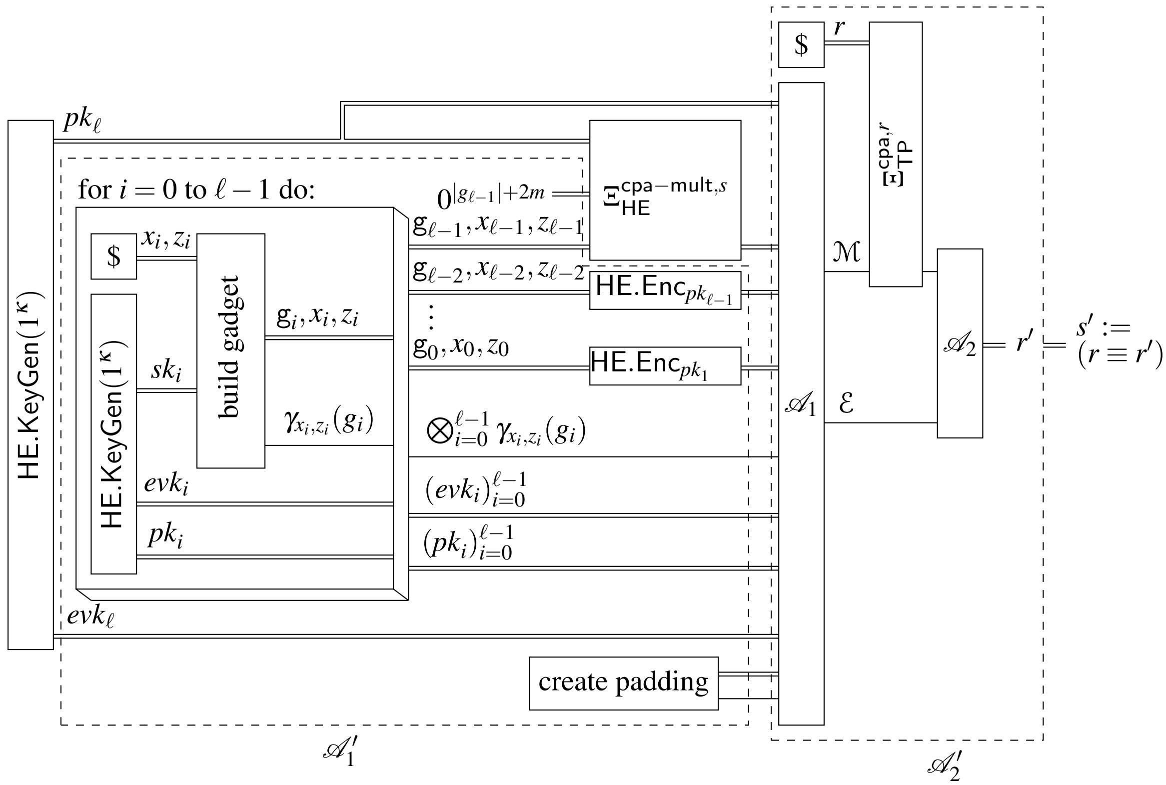

Figure 3: A strategy for the game $\mathsf{PubK}^{\mathsf{cpa-mult}}_{\mathscr{A}',\mathsf{HE}}(\kappa)$, using an adversary $\mathscr{A}$ for $\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{TP}^{(\ell)}}(\kappa)$ as a subroutine. All the wires that form an input to $\mathscr{A}_1$ together form the evaluation key and public key for $\mathsf{TP}^{(\ell)}$ or $\mathsf{TP}^{(\ell-1)}$, depending on s. Note that $\Xi^{\mathsf{cpa},r}_{\mathsf{TP}} = \Xi^{\mathsf{cpa},r}_{\mathsf{TP}^{(\ell)}} = \Xi^{\mathsf{cpa},r}_{\mathsf{TP}^{(\ell-1)}}$, so $\mathscr{A}'_2$ can run either one of these independently of s (i. e., without having to query the challenger). The "create padding" subroutine generates dummy gadgets for ℓ up to $L-1$, as described in the definition of $\mathscr{A}_1$.

Because HE is q-IND-CPA secure, the probability that $\mathscr{A}'$ wins

$$\mathsf{PubK}^{\mathsf{cpa-mult}}_{\mathscr{A}',\mathsf{HE}}(\kappa),$$

i. e., that $s' \equiv s$, is at most $1/2 + \eta'(\kappa)$ for some negligible function η'. There are two scenarios in which $\mathscr{A}'$ wins the game:

- $s = 1$ *and* $\mathscr{A}$ guesses r correctly: If $s = 1$, the game that is being simulated is

  $$\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{TP}^{(\ell)}}(\kappa).$$

  If $\mathscr{A}$ wins the simulated game ($r \equiv r'$), then $\mathscr{A}'$ will correctly output $s' = 1$. (If $\mathscr{A}$ loses, then $\mathscr{A}'$ outputs 0, and loses as well.)

- $s = 0$ *and* $\mathscr{A}$ does not guess r correctly: If $s = 0$, the game that is being simulated is

  $$\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{TP}^{(\ell-1)}}(\kappa).$$

If $\mathscr{A}$ loses the game ($r \not\equiv r'$), then $\mathscr{A}'$ will correctly output $s' = 0$. (If $\mathscr{A}$ wins, then $\mathscr{A}'$ outputs 1 and loses.)

From the above, we conclude that

$$\begin{aligned}
&\Pr[s=1]\cdot\Pr[\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{TP}^{(\ell)}}(\kappa)=1]+\Pr[s=0]\cdot\Pr[\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{TP}^{(\ell-1)}}(\kappa)=0] \le \frac{1}{2}+\eta'(\kappa)\\
\Leftrightarrow\quad &\frac{1}{2}\Pr[\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{TP}^{(\ell)}}(\kappa)=1]+\frac{1}{2}\Big(1-\Pr[\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{TP}^{(\ell-1)}}(\kappa)=1]\Big) \le \frac{1}{2}+\eta'(\kappa)\\
\Leftrightarrow\quad &\Pr[\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{TP}^{(\ell)}}(\kappa)=1]-\Pr[\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{TP}^{(\ell-1)}}(\kappa)=1] \le 2\eta'(\kappa).
\end{aligned}$$

Set $\eta(\kappa) := 2\eta'(\kappa)$, and the proof is complete. □

By applying Lemma 4.2 iteratively, L times in total, we can conclude that the difference between $\mathsf{TP}^{(L)}$ and $\mathsf{TP}^{(0)}$ is negligible, because the sum of polynomially many negligible functions is still negligible:

Corollary 4.3. *If L is polynomial in κ, then for any quantum polynomial-time adversary $\mathscr{A} = (\mathscr{A}_1, \mathscr{A}_2)$, there exists a negligible function η such that*

$$\Pr[\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{TP}^{(L)}}(\kappa)=1]-\Pr[\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{TP}^{(0)}}(\kappa)=1] \le \eta(\kappa).$$

Using Corollary 4.3, we can finally prove the q-IND-CPA security of our scheme $\mathsf{TP} = \mathsf{TP}^{(L)}$.

Proof of Theorem 4.1. The scheme $\mathsf{TP}^{(0)}$ is very similar to CL in terms of its key generation and encryption steps. The evaluation key consists of several classical evaluation keys, plus some completely mixed states and encryptions of 0 which we can safely ignore because they do not contain any information about the encrypted message. In both schemes, the encryption of a qubit is a quantum one-time pad together with the encrypted keys. The only difference is that in $\mathsf{TP}^{(0)}$, the public key and evaluation key form a tuple containing, in addition to pk_0 and evk_0 which are used for the encryption of the quantum one-time pad, a list of public/evaluation keys that are independent of the encryption. These keys do not provide any advantage (in fact, the adversary could have generated them himself by repeatedly running $\mathsf{HE.KeyGen}(1^\kappa, 1^L)$). Therefore, we can safely ignore these keys as well.

In [16, Lemma 5.3], it is shown that CL is q-IND-CPA secure. Because of the similarity between CL and $\mathsf{TP}^{(0)}$, the exact same proof shows that $\mathsf{TP}^{(0)}$ is q-IND-CPA secure as well, that is, for any $\mathscr{A}$ there exists a negligible function η' such that

$$\Pr[\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{TP}^{(0)}}(\kappa)=1] \le \frac{1}{2}+\eta'(\kappa).$$

Combining this result with Corollary 4.3, it follows that

$$\begin{aligned}
\Pr[\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{TP}}(\kappa)=1] &\le \Pr[\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{TP}^{(0)}}(\kappa)=1]+\eta(\kappa)\\
&\le \frac{1}{2}+\eta'(\kappa)+\eta(\kappa).
\end{aligned}$$

Since the sum of two negligible functions is itself negligible, we have proved Theorem 4.1. □

4.1 Circuit privacy

The scheme TP as presented above ensures the privacy of the input data. It does not guarantee, however, that whoever generates the keys, encrypts, and decrypts cannot gain information about the circuit C that was applied to some input ρ by the evaluator. Obviously, the output value $\mathsf{C}\rho\mathsf{C}^\dagger$ often reveals something about the circuit C, but apart from this necessary leakage of information, one may require a (quantum) homomorphic encryption scheme to ensure *circuit privacy* in the sense that an adversary cannot statistically gain any information about C from the output of the evaluation procedure that it could not already gain from $\mathsf{C}\rho\mathsf{C}^\dagger$ itself and the value of L.

We claim that circuit privacy for TP in the semi-honest setting (i. e., against passive adversaries[6]) can be obtained by modifying the scheme only slightly, given that the classical encryption scheme has the circuit privacy property.

Theorem 4.4. *If* HE *has circuit privacy in the semi-honest setting, then* TP *can be adapted to a quantum homomorphic encryption scheme with circuit privacy against honest-but-curious adversaries.*

Proof sketch. If the evaluator randomizes the encryption of the output data by applying a quantum one-time pad to the (already encrypted) result of the evaluation, the keys themselves are uniformly random and therefore do not reveal any information about what circuit was evaluated. The evaluator can then proceed to update the classical encryptions of those keys accordingly, and by the circuit privacy of the classical scheme, the resulting encrypted keys will also contain no information about the computations performed. A more thorough proof is given in Appendix B. □

5 Constructing the gadgets

In this section we will first show how apply Barrington's theorem to construct gadgets that have polynomial size whenever the scheme HE has a decryption circuit with logarithmic depth, i. e., the decryption function is in NC^1. To illustrate these techniques, we apply them to create gadgets for schemes that are based on Learning With Errors (LWE).

This construction will already be powerful enough to instantiate TP with current classical schemes for homomorphic encryption, since these commonly have low-depth decryption circuits. In Section 5.3 and Appendix C, we will present a larger toolkit to construct gadgets, which is efficient for a larger class of possible decryption functions. This construction is based on the garden-hose model [17], and may be read as an alternative to Section 5.1 and 5.2 by readers already familiar with that model.

Finally, we will reflect on the possibility of constructing both types of gadgets in scenarios where quantum power is limited.

5.1 Gadget construction for log-depth decryption circuits

The main tool for creating gadgets that encode log-depth decryption circuits comes from Barrington's theorem, a classic result in complexity theory, which states that all Boolean circuits of logarithmic depth

[6]Note that there various ways to define passive adversaries in the quantum setting [23, 8]. Here, we are considering adversaries that follow all protocol instructions exactly.

can be encoded as polynomial-size width-5 permutation branching programs. Every instruction of such a branching program will be encoded as connections between five Bell pairs.

Definition 5.1. A *width-k permutation branching program* of length M on an input $x \in \{0,1\}^n$ is a list of M instructions of the form $\langle i_m, \sigma_m^1, \sigma_m^0 \rangle$, for $1 \le m \le M$, such that $i_m \in [n]$, and σ_m^1 and σ_m^0 are elements of S_k, i. e., permutations of $[k]$. The program is executed by composing the permutations given by the instructions 1 through M, selecting σ_m^1 if $x_{i_m} = 1$ and selecting σ_m^0 if $x_{i_m} = 0$. The program *rejects* if this product equals the identity permutation and *accepts* if it equals a fixed k-cycle.

Since these programs have a very simple form, it came as a surprise when they were proven to be quite powerful [7].

Theorem 5.2 (Barrington [7])**.** *Every fan-in 2 Boolean circuit C of depth d can be simulated by a width-5 permutation branching program of length at most* 4^d.

Our gadget construction will consist of first transforming the decryption function HE.Dec into a permutation branching program, and then encoding this permutation branching program as a specification of a gadget, as produced by $\mathsf{TP.GenGadget}_{pk'}(sk)$, and usage instructions $\mathsf{TP.GenMeasurement}(\widetilde{a})$.

Theorem 5.3. *Let* $\mathsf{HE.Dec}_{sk}(\widetilde{a})$ *be the decryption function of the classical homomorphic encryption scheme* HE. *If* HE.Dec *is computable by a Boolean fan-in 2 circuit of depth* $O(\log(\kappa))$*, where* κ *is the security parameter, then there exist gadgets for* TP *of size polynomial in* κ.

Proof. Our description will consist of three steps. First, we write HE.Dec as a width-5 permutation branching program, of which the instructions alternately depend on the secret key sk and on the ciphertext $\widetilde{a}$. Secondly, we specify how to transform these instructions into a gadget which almost works correctly, but for which the qubit ends up at an unknown location. Finally, we complete the construction by executing the inverse program, so that the qubit ends up at a known location.

The first part follows directly from Barrington's theorem. The effective input of HE.Dec can be seen as the concatenation of the secret key sk and the ciphertext $\widetilde{a}$. Since by assumption the circuit is of depth $O(\log \kappa)$, there exists width-5 permutation branching program $\mathcal{P}$ of length $M = \kappa^{O(1)}$, with the following properties. We write

$$\mathcal{P} = \left(\langle i_1, \sigma_1^1, \sigma_1^0 \rangle, \langle i_2, \sigma_2^1, \sigma_2^0 \rangle, \ldots, \langle i_M, \sigma_M^1, \sigma_M^0 \rangle\right)$$

as the list of instructions of the width-5 permutation branching program. Without loss of generality,[7] we can assume that the instructions alternately depend on bits of $\widetilde{a}$ and bits of sk. That is, the index i_m refers to a bit of $\widetilde{a}$ if m is odd, and to a bit of sk if m is even. There are M instructions in total, of which $M/2$ are odd-numbered and $M/2$ are even.

The output of $\mathsf{TP.GenGadget}_{pk'}(sk)$, i. e., the list of pairs $\{(s_i, t_i)\}_i$ in $g(sk)$ that defines the structure of the gadget, will be created from the even-numbered instructions, evaluated using the secret key sk. For every even-numbered $m \le M$, we connect ten qubits in the following way. Suppose the m^{th} instruction evaluates to some permutation $\sigma_m := \sigma_m^{sk_{i_m}}$. Label the 10 qubits of this part of the gadget

[7] This can be seen by inserting dummy instructions that always perform the identity permutation between any two consecutive instructions that depend on the same variable. Alternatively, it would be possible to improve the construction by "multiplying out" consecutive instructions whenever they depend on the same variable.

by $1_{m,\text{in}}, 2_{m,\text{in}}, \ldots, 5_{m,\text{in}}$ and $1_{m,\text{out}}, 2_{m,\text{out}}, \ldots, 5_{m,\text{out}}$. These will correspond to 5 EPR pairs, connected according to the permutation:

$$(1_{m,\text{in}}, \sigma_m(1)_{m,\text{out}}),\ (2_{m,\text{in}}, \sigma_m(2)_{m,\text{out}}), \quad \text{etc., up to} \quad (5_{m,\text{in}}, \sigma_m(5)_{m,\text{out}}).$$

After the final instruction of the branching program, σ_M, also perform an inverse phase gate $\mathsf{P}^\dagger$ on the qubits labeled as $2_{M,\text{out}}$, $3_{M,\text{out}}$, $4_{M,\text{out}}$, $5_{M,\text{out}}$. Execution of the gadget will teleport the qubit through one of these whenever $\mathsf{HE.Dec}(\widetilde{a}) = 1$.

For this construction, $\mathsf{TP.GenMeasurement}(\widetilde{a})$ will be given by the odd instructions, which depend only on the bits of $\widetilde{a}$ (and the function $\mathsf{HE.Dec}$, but not on sk). Again, for all odd $m \leq M$, let $\sigma_m := \sigma_m^{\widetilde{a}_{i_m}}$ be the permutation given by the evaluation of instruction m on $\widetilde{a}$. For all m strictly greater than one, the measurement instructions will be: perform a Bell measurement according to the permutation σ_m between the "out" qubits of the previous set, and the "in" qubits of the next. The measurement pairs will then be

$$(1_{m-1,\text{out}}, \sigma(1)_{m,\text{in}}),\ (2_{m-1,\text{out}}, \sigma(2)_{m,\text{in}}), \quad \text{up to} \quad (5_{m-1,\text{out}}, \sigma(5)_{m,\text{in}}).$$

For $\ell = 1$, there is no previous layer to connect to, only the input qubit. For that, we add the measurement instruction $(0, \sigma(1)_{1,\text{in}})$, where 0 is the label of the input qubit.

By Barrington's theorem, if $\mathsf{HE.Dec}_{sk}(\widetilde{a}) = 0$ then the product, say τ, of the permutations coming from the evaluated instructions equals the identity. In that case, consecutively applying these permutations on "1," results in the unchanged starting value, "1." If instead the decryption would output 1, the consecutive application results in another value in $\{2,3,4,5\}$, because in that case, τ is a k-cycle. After teleporting a qubit through these EPR pairs, with teleportation measurements chosen accordingly, the input qubit will be present at $\tau(1)_{M,\text{out}}$, with an inverse phase gate if $\tau(1)$ is unequal to 1. Thus, if an inverse phase $P^\dagger$ is applied to the output qubits at position 2 through 5 (but not position 1), then a phase gate is applied whenever $\mathsf{HE.Dec}_{sk}(\widetilde{a}) = 1$. The positions where an inverse phase is applied define the bitstring p in $g(sk)$.

The gadget constructed so far would correctly apply the phase gate, conditioned on $\mathsf{HE.Dec}_{sk}(\widetilde{a})$, with one problem: afterward, the qubit is at a location unknown to the user of the gadget, because the user cannot compute τ.

We fix this problem in the following way: execute the inverse branching program afterwards. The entire construction is continued in the same way, but the instructions of the inverse program are used. The inverse program can be made from the original program by reversing the order of instructions, and then for each permutation using its inverse permutation instead. The first inverse instruction is

$$\langle i_M, (\sigma_M^1)^{-1}, (\sigma_M^0)^{-1} \rangle, \quad \text{then} \quad \langle i_{M-1}, (\sigma_{M-1}^1)^{-1}, (\sigma_{M-1}^0)^{-1} \rangle,$$

with final instruction

$$\langle i_1, (\sigma_1^1)^{-1}, (\sigma_1^0)^{-1} \rangle.$$

One small detail is that i_M is used twice in a row, breaking the alternation; the user of the gadget can simply perform the measurements that correspond to the identity permutation in between, since

$$(\sigma_M^0)(\sigma_M^0)^{-1} = (\sigma_M^1)(\sigma_M^1)^{-1} = e \quad \text{(the identity permutation)}.$$

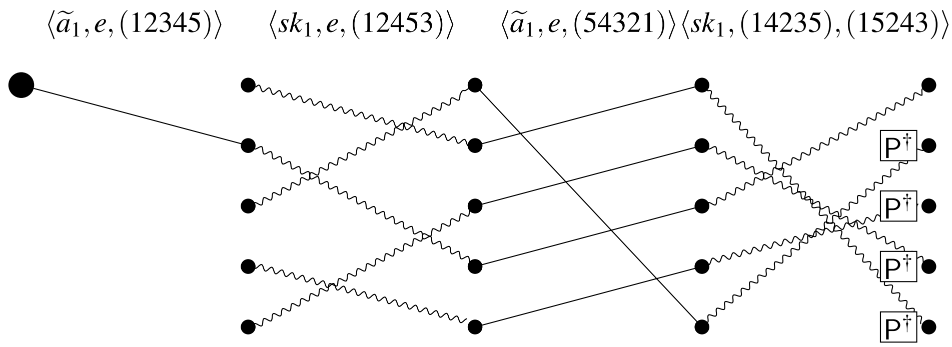

Figure 4: Structure of the (first half of the) gadget, with measurements, coming from the 5-permutation branching program for the OR function on the input $(0,0)$. The example program's instructions are displayed above the permutations. The solid lines correspond to Bell measurements, while the wavy lines represent EPR pairs.

After having repeated the construction with the inverse permutation branching program, the qubit is guaranteed to be at the location where it originally started: $\sigma_1(1)$ of the final layer of five qubits—that will then be the corrected qubit which is the output of the gadget.

The total number of qubits which form the gadget, created from a width-5 permutation branching program of length M, of which the instructions alternate between depending on $\widetilde{a}$ and depending on sk, is $2 \cdot (5M) = 10M$. $\square$

Example. The OR function on two bits can be computed using a width-5 permutation branching program of length 4, consisting of the following list of instructions:

1. $\langle 1, e, (12345) \rangle$
2. $\langle 2, e, (12453) \rangle$
3. $\langle 1, e, (54321) \rangle$
4. $\langle 2, (14235), (15243) \rangle$

As a simplified example, suppose the decryption function $\mathsf{HE.Dec}_{sk}(\widetilde{a})$ is $sk_1 \,\mathsf{OR}\, \widetilde{a}_1$. Then, for one possible example set of values of $\widetilde{a}$ and sk, half of the gadget and measurements will be as given in Figure 4. To complete this gadget, the same construction is appended, reflected horizontally.

5.2 Specific case: Learning With Errors

The scheme by Brakerski and Vaikuntanathan [12] is well-suited for our construction, and its decryption function is representative for a much wider class of schemes which are based on the hardness of Learning With Errors (LWE). As an example, we construct gadgets that enable quantum homomorphic encryption based on the BV11 scheme. Let κ be the security parameter, and let p be the modulus of the integer ring over which the scheme operates.

The ciphertext c is given by a pair $(\mathbf{v}, w)$, with $\mathbf{v} \in \mathbb{Z}_p^\kappa$ and $w \in \mathbb{Z}_p$. The secret key $\mathbf{s}$ is an element of $\mathbb{Z}_p^\kappa$. The decryption to a message x involves computation of an inner product over the ring $\mathbb{Z}_p$,

$$x = \mathsf{Dec}_{\mathbf{s}}((\mathbf{v}, w)) := (w - \langle \mathbf{v}, \mathbf{s} \rangle) \pmod{p} \pmod{2}. \tag{5.1}$$

The BV11 scheme is able to make the modulus small, i. e., polynomial in κ, before encryption. We present an explicit construction for the case of small modulus p, which can be illustrative to read as example of our construction, and an implicit construction for more complicated gadgets for the case of superpolynomial p. Both constructions lead to gadgets that are polynomial in κ.

Small modulus

Take the modulus p to be polynomial in κ. We describe a series of small "permutation gadgets" that move an arbitrary qubit to a location, depending on whether $x = 0$ or $x = 1$. By doubling the construction as seen before, it is easy to turn these into a gadget which applies an inverse phase gate whenever $x = 1$. Note that we could just apply Theorem C.1 in order to construct a gadget directly from a log-space Turing machine of the decryption function. In this example, however, we choose to exhibit a more efficient gadget that exploits the structure of the BV11 scheme.

We follow [12, Section 4.5] in rewriting equation (5.1) in terms of binary arithmetic. Let $\mathbf{s}[i](j)$ denote the jth bit of the ith entry of $\mathbf{s}$, then the inner product can be written as

$$\begin{aligned} w - \langle \mathbf{v}, \mathbf{s} \rangle \pmod{p} &= w - \sum_{i=1}^{k} \mathbf{v}[i]\mathbf{s}[i] \pmod{p} \\ &= w - \sum_{i=1}^{k} \sum_{j=0}^{\log p} \mathbf{v}[i](j) \cdot 2^j \cdot \mathbf{s}[i] \pmod{p}. \end{aligned} \tag{5.2}$$

Let a *permutation gadget* be a subgadget of size $2p$, parametrized by a number $q \in \mathbb{Z}_p$. Label the first p qubits by 0_{in} to $(p-1)_{\text{in}}$, and the second p qubits by 0_{out} to $(p-1)_{\text{out}}$. The gadget simply creates EPR pairs between y_{in} and $(y + q \pmod{p})_{\text{out}}$, for all $y \in \mathbb{Z}_p$. Such a gadget can effectively simulate addition with q over $\mathbb{Z}_p$.

For each element of the vector $\mathbf{s}$ we will create $\log p$ permutation gadgets. The intuition behind the construction is as follows. The inner product which computed the decryption of the ciphertext is written as a sum of $\kappa \log p$ numbers, that either contribute to the sum or not, depending on a bit of the ciphertext $\mathbf{v}$.

For each i from 1 to κ, and each j from 0 to $\log p$, we create a permutation gadget, labeled by (i, j), for the number $2^j \cdot \mathbf{s}[i]$.

The evaluator uses this gadget in the following way. He performs a Bell measurement between the input qubit and the 0_{in} qubit of the first gadget such that $\mathbf{v}[i](j) = 1$. Then, he connects all output qubits 0_{out} to $(p-1)_{\text{out}}$ of this gadget to all the input qubits of the next gadget for which $\mathbf{v}[i](j) = 1$.

After teleporting his qubit through all gadgets, the qubit will be exactly at the location z_{out} of the final gadget the evaluator used, where

$$z = \sum_{i=1}^{\kappa} \mathbf{v}[i]\mathbf{s}[i] \pmod{p},$$

although of course the evaluator has no way of knowing which of the p locations this is. He can then, by simple permutation, apply an inverse phase gate whenever $w - z \pmod{2} = 1$.

Finally, as in the construction from the previous section, we double the entire construction to route the unknown qubit back to a known location. The size of the total gadget is then bounded by $4\kappa p \log p$.

Large modulus

In case the modulus p is superpolynomially large, constructing the gadget explicitly appears to be much harder, and a log-space algorithm for this inner product is not immediately obvious. For completeness, we sketch a proof strategy to reiterate that such a polynomial-size gadget does still exist in this case.

The decryption function of equation (5.1) has depth $O(\log \kappa + \log \log p)$, see for example [12, Lemma 4.5]. This can be proved by writing the decomposition of equation (5.2) as a Wallace tree.

Given a low-depth circuit, we could now apply Theorem 5.3 to convert this circuit into a garden-hose protocol. In contrast to the small-modulus case, we do not exploit the structure of the decryption function to construct a more efficient gadget.

5.3 Gadget construction for log-space computable decryption functions

Even though the construction based on Barrington's theorem has enough power for current classical homomorphic schemes, it is possible to improve on this construction in two directions. Firstly, we extend our result to be able to handle a larger class of decryption functions: those that can be computed in logarithmic space, instead of only NC^1. Secondly, for some specific decryption functions, executing the construction of Section 5.1 might produce significantly larger gadgets than necessary. For instance, even for very simple circuits of depth $\log \kappa$, Barrington's theorem produces programs of length κ^2—a direct approach can often easily improve on the exponent of the polynomial. See also the garden-hose protocols for equality [42, 19] and the majority function [38] for examples of non-trivial protocols that are much more efficient than applying Barrington's theorem as a black box.

In Appendix C we describe a construction for log-space computation in depth. The explanation in the appendix uses a different language than the direct encoding of the previous section: Namely, there is a natural way of writing the requirements on the gadgets as a two-player task, and then writing strategies for this task in the *garden-hose model*.

Theorem 5.4. *Let* $\mathsf{HE.Dec}_{sk}(\widetilde{a})$ *be the decryption function of the classical homomorphic encryption scheme* HE. *If* $\mathsf{HE.Dec}$ *is computable by a Turing machine that uses space* $O(\log \kappa)$, *where* κ *is the security parameter, then there exist gadgets for* TP *of size polynomial in* κ.

Writing these gadgets in terms of the garden-hose model, even though it adds a layer of complexity to the construction, gives more insight into the structure of the gadgets, and forms its original inspiration. We therefore sketch the link between log-space computation and gadget construction within this framework.

Viewing the gadget construction as instance of the garden-hose model, besides clarifying the log-space construction, also makes it easier to construct gadgets for specific cases. Earlier work developed protocols in the garden-hose model for several functions, see for instance [52, 17, 38], and connections to other models of computation. These results on the garden-hose model might serve as building blocks to create more efficient gadgets for specific decoding functions of classical homomorphic schemes, that are potentially much smaller than those created as a result of following the general constructions of Theorem 5.3 or 5.4.

5.4 Constructing gadgets using limited quantum resources

In a setting where a less powerful client wants to delegate some quantum computation to a more powerful server, it is important to minimize the amount of effort required from the client. In delegated quantum computation, the complexity of a protocol can be measured by, among other things, the total amount of communication between client and server, the number of rounds of communication, and the quantum resources available to the client, such as possible quantum operations and memory size.

We claim that TP gives rise to a three-round delegated quantum computation protocol in a setting where the client can perform only Pauli and swap operations. $\mathsf{TP.Enc}$ and $\mathsf{TP.Dec}$ only require local application of Pauli operators to a quantum state, but $\mathsf{TP.KeyGen}$ is more involved: in the current description of the gadget generation, the key generator has to generate EPR-pairs, as well as perform $\mathsf{P}^\dagger$ gates and Bell measurements. We show in this section how the gadgets can be generated securely using only X, Z and swaps, when the key generator is given resources by some computationally more powerful (but potentially malicious) party, for example the evaluator.

As described in Section 3.1, the gadget $\Gamma_{pk'}(sk)$ is effectively a list of $2m$ EPR-pairs, some of which have an extra $(\mathsf{P}^\dagger \otimes I)$ transformation on them, with the qubits ordered in some way that depends on sk. See also Figure 7 in Appendix C. If the key generator is supplied with a list of $2m$ EPR-pairs $|\Phi^+\rangle$ and as many pairs $(I \otimes \mathsf{P}^\dagger)|\Phi^+\rangle$, it is clear that he can create the gadget by swapping some of the qubits (e. g., using CNOT gates), and applying a single random Pauli operation (using X and Z gates) on every pair. Any unused pairs are discarded. If the supplier of these pairs follows the protocol and sends actual EPR-pairs to the key generator, this tactic suffices to hide all information about sk. However, we need to make sure that even if the supplier acts maliciously and actively tries to gather information about sk, this information is still secure. The key generator, upon receiving the (real or fake) EPR-pairs, will instead apply an independently-selected random Pauli transformation on each of the qubits individually, i. e., the key generator applies a quantum one-time pad. From the perspective of a supplier that has no knowledge of the quantum one-time pad key, this is equivalent to tracing out these qubits and replacing them with a completely mixed state—even if the malicious supplier was entangled with these qubits before sending them to the client. Since it doesn't make a difference whether or not qubits are swapped amongst themselves before they are replaced by a completely mixed state, the supplier can not learn whether this happened and therefore no information about sk will be revealed to the supplier. Of course, the correctness of the outcome is not guaranteed in the case of a malicious supplier.

We just discussed a three-round delegated protocol in a setting where the client is limited in the types of operations he can perform. Alternatively, TP can be regarded as a two-round delegated protocol in a setting where the client can perform arbitrary Clifford operations, but is limited to a constant-size quantum memory, given that $\mathsf{HE.Dec}$ is in NC^1. In that case, the gadgets can be constructed ten qubits at a time, by constructing the sets of five EPR pairs as specified in Section 5.1. By decomposing the 5-cycles into products of 2-cycles, the quantum memory can even be reduced to only four qubits. The client sends these small parts of the gadgets to the server as they are completed. Because communication remains one-way until all gadgets have been sent, this can be regarded as a single round of communication.

These two delegated-computation protocols exhibit a trade-off between the extensiveness of the client's quantum storage and his quantum computing power. Other known protocols for delegated quantum computation often require even less from the client both in terms of computing power and storage, but these generally sacrifice on a third parameter: the number of communication rounds. For example, the universal blind quantum computing protocol introduced by Broadbent et al. [15] only requires the client to generate single-qubit states from a fixed, small set of possible states, but client and server need to communicate separately for every quantum gate in the circuit. In our scheme, we minimize communication during the evaluation phase using the gadgets, the generation of which involves more quantum computational power and/or quantum storage from the client. An instructive overview of the current state of the art in delegated quantum computing can be found in a recent survey paper by Fitzsimons [26].

6 Conclusion

We have presented the first quantum homomorphic encryption scheme TP that is compact and allows evaluation of circuits with polynomially many T gates in the security parameter, i. e., arbitrary polynomial-size circuits. Assuming that the number of wires involved in the evaluation circuit is also polynomially related to the security parameter, we may consider TP to be leveled fully homomorphic. The scheme is based on an arbitrary classical FHE scheme, and any computational assumptions needed for the classical scheme are also required for security of TP. However, since TP uses the classical FHE scheme as a black box, any FHE scheme can be plugged in to change the set of computational assumptions.

Our constructions are based on a new and interesting connection between the area of instantaneous non-local quantum computation and quantum homomorphic encryption. Recent techniques developed by Speelman [53], based on the garden-hose model [17], have turned out to be crucial for our construction of quantum gadgets which allow homomorphic evaluation of T gates on encrypted quantum data.

6.1 Future work

Since Yu, Pérez-Delgado and Fitzsimons [58] showed that information-theoretically secure QFHE is impossible (at least in the exact case), it is natural to wonder whether it is possible to construct a non-leveled QFHE scheme based on computational assumptions. If such a scheme is not possible, can one find lower bounds on the size of the evaluation key of a compact scheme? Other than the development of more efficient QFHE schemes, one can consider the construction of QFHE schemes with extra properties, such as circuit privacy against active adversaries. It is also interesting to look at other cryptographic tasks that might be executed using QFHE. In the classical world for example, multiparty computation protocols can be constructed from fully homomorphic encryption [21]. We consider it likely that our new techniques will also be useful in other contexts such as quantum indistinguishability obfuscation [4].

Acknowledgements

We acknowledge useful discussions with Anne Broadbent, Harry Buhrman, and Leo Ducas. We thank Stacey Jeffery for providing the inspiration for a crucial step in the security proof, and Gorjan Alagic and Anne Broadbent for helpful comments on a draft of this article.

A Using the gadget

After using the gadget, but before updating any classical information, the evaluator has the following pieces of classical information. The encrypted one-time pad keys $\widetilde{a}$, $\widetilde{b}$, a list of m pairs for Bell measurements $M \leftarrow \mathsf{TP.GenMeasurement}(\widetilde{a})$ and a list of outcomes for each of these m measurements, say $c, d \in \{0,1\}^m$.

The evaluator also has encrypted versions of $\{(s_1,t_1),(s_2,t_2),\ldots,(s_m,t_m)\}$, $p \in \{0,1\}^m$, and $x,z \in \{0,1\}^m$ that specify the structure of the gadget.

Say an arbitrary qubit[8] was teleported through the gadget, so that the qubit started in some state $\mathsf{P}^a\mathsf{X}^a\mathsf{Z}^b|\psi\rangle$ and is currently in state $\mathsf{X}^{a'}\mathsf{Z}^{b'}|\psi\rangle$. We sketch the algorithm an evaluator would execute on this encrypted state, to compute (encrypted versions of) the updated keys a' and b'. Updating the keys is not complicated, it mostly involves bookkeeping to keep track of the current location of the qubit, and its current X-correction, Z-correction and phase.

We explain the calculation as if performed with the unencrypted versions; in the actual execution, only the encrypted versions of all variables are used, and this entire calculation is performed homomorphically. Since all the mentioned classical information either is or can be encrypted with the same public key, this calculation can be handled by the classical homomorphic scheme HE.

The algorithm tracks the path the qubit takes through the gadget, by resolving the teleportations that involve the qubit one by one. Even though the measurements were all performed at the same time, we will describe them as if ordered in this manner. All additions of the keys of the one-time pad will be performed modulo 2, since $\mathsf{X}^2 = \mathsf{Z}^2 = \mathbb{I}$.

Let $\mathbf{a},\mathbf{b}$ be variables that hold the current key to the one-time pad at every step of the algorithm. We initialize these as $\mathbf{a} \leftarrow a$ and $\mathbf{b} \leftarrow b$. Let $\mathbf{q}$ be a variable that stores whether or not the qubit currently has an extra phase gate, initialized as $\mathbf{q} \leftarrow a$. Let $\mathbf{r}$ be the variable that contains the current location of the qubit, with possible locations 0 to $2m$, initialized to 0. That is, we view the current state as being $\mathsf{P}^{\mathbf{q}}\mathsf{X}^{\mathbf{a}}\mathsf{Z}^{\mathbf{b}}|\psi\rangle$ at location $\mathbf{r}$. For every step we update the location depending on M and $\{(s_1,t_1),(s_2,t_2),\ldots,(s_m,t_m)\}$, and update the keys depending on the corresponding measurement outcomes.

First, find the pair in M that contains the current location $\mathbf{r}$, say pair i which consists of (r,s) for some other location s. The outcome of this measurement is given by $c[i]$ and $d[i]$. Effectively, these outcomes

[8] The input qubit is not necessarily a pure state, but we write an arbitrary pure state without loss of generality, to simplify notation.

change the current state to

$$\mathsf{X}^{c[i]}\mathsf{Z}^{d[i]}\mathsf{P}^{\mathbf{q}}\mathsf{X}^{\mathbf{a}}\mathsf{Z}^{\mathbf{b}}|\psi\rangle = \mathsf{P}^{\mathbf{q}}\mathsf{X}^{\mathbf{a}+c[i]}\mathsf{Z}^{\mathbf{b}+d[i]+\mathbf{q}\cdot c[i]}|\psi\rangle\,,$$

therefore we update $\mathbf{a} \leftarrow \mathbf{a}+c[i]$ and $\mathbf{b} \leftarrow \mathbf{b}+d[i]+q\cdot c[i]$. Note that the key update rules also involve multiplication—an extra Z gate is added if the phase gate was present, $\mathbf{q}=1$, and the teleportation measurement required an X correction, $c[i]=1$.

Next, find the pair in $\{(s_1,t_1),(s_2,t_2),\ldots,(s_m,t_m)\}$ that contains the new location s, say pair j containing (s,t). The teleportation of the qubit through this pair effectively applies $\mathsf{X}^{x[j]}\mathsf{Z}^{z[j]}(\mathsf{P}^\dagger)^{p[j]}$ to the state. Then, if we already use the updated $\mathbf{a}$ and $\mathbf{b}$, the quantum state at this step equals

$$\begin{aligned}\mathsf{X}^{x[j]}\mathsf{Z}^{z[j]}(\mathsf{P}^\dagger)^{p[j]}\mathsf{P}^{\mathbf{q}}\mathsf{X}^{\mathbf{a}}\mathsf{Z}^{\mathbf{b}}|\psi\rangle &= \mathsf{X}^{x[j]}\mathsf{Z}^{z[j]}\mathsf{P}^{p[j]}\mathsf{P}^{\mathbf{q}}\mathsf{X}^{\mathbf{a}}\mathsf{Z}^{\mathbf{b}+p[j]}|\psi\rangle\\ &= \mathsf{P}^{p[j]+\mathbf{q} \pmod 2}\mathsf{X}^{\mathbf{a}+x[j]}\mathsf{Z}^{\mathbf{b}+z[j]+p[j]\cdot(1+\mathbf{q})+x[j]\cdot(p[j]+\mathbf{q})}|\psi\rangle\,.\end{aligned}$$

For rewriting, we used the fact that $\mathsf{P}^2=\mathsf{Z}$ and that $\mathsf{P}^\dagger=\mathsf{PZ}$, together with the commutation relations from the previous section. We therefore update the phase $\mathbf{q} \leftarrow p[j]+\mathbf{q} \pmod 2$, and the components of the quantum one-time pad to $\mathbf{a} \leftarrow \mathbf{a}+x[j]$ and $\mathbf{b} \leftarrow \mathbf{b}+z[j]+p[j]\cdot(1+q[j])+x[j]\cdot(p[j]+q[j])$. Finally, set the new location of the qubit $\mathbf{r} \leftarrow t$.

The previous two steps are then repeated m times, where $2m$ is the size of the gadget, to eventually (homomorphically) compute the new updated keys a',b' to the quantum one-time pad. Afterwards, all temporary variables can be discarded, and only the updated keys will be needed for continuing the protocol.

B Circuit privacy

In this appendix, we demonstrate that with only a slight modification of TP, the scheme has circuit privacy in the semi-honest setting, i. e., against passive adversaries. Classically, circuit privacy is defined by requiring the existence of a simulator $\mathsf{Sim}_{\mathsf{HE}}$ whose inputs are the public parameters and $\mathsf{C}(x)$ and which produces an output which is indistinguishable from the homomorphic evaluation of C on the encryption of x. Formally, circuit privacy is defined as follows.

Definition B.1 (Classical circuit privacy—semi-honest setting [37])**.** A classical homomorphic encryption scheme HE has statistical circuit privacy in the semi-honest ("honest-but-curious") model if there exists a *PPT* algorithm $\mathsf{Sim}_{\mathsf{HE}}$ and a negligible function η such that for any security parameter κ, input x, key set $(pk,evk,sk) \leftarrow \mathsf{HE.KeyGen}(1^\kappa)$, and circuit C:

$$\delta(\mathsf{HE.Eval}^{\mathsf{C}}_{evk}(\mathsf{HE.Enc}_{pk}(x)),\mathsf{Sim}_{\mathsf{HE}}(1^\kappa,pk,evk,\mathsf{C}(x))) \leq \eta(\kappa)\,.$$

Here,

$$\delta(X,Y) := \frac{1}{2}\sum_{u\in U}\big|\Pr[X=u]-\Pr[Y=u]\big|$$

is the *statistical distance* between two random variables over a finite universe U. For notational convenience, we will often write $\mathsf{Sim}_{\mathsf{HE}}(\mathsf{C}(x))$ if the rest of the arguments are clear from the context. Also we will sometimes write $\mathsf{X} \approx_a Y$ to denote that $\delta(X,Y) \leq a$.

If the recryption functionality $\mathsf{Rec}_{i\to j}$ is defined as the composition of the procedures

$$\mathsf{HE.Eval}_{evk_j}^{\mathsf{HE.Dec}_i} \qquad \text{and} \qquad \mathsf{HE.Enc}_{pk_j},$$

as in Section 2.2, then recryptions do not degrade the privacy of the computation: a homomorphic evaluation of some function with key switching is statistically close to running the simulator directly on the function output using only the *last* key set.

Lemma B.2. *Suppose* HE *has statistical circuit privacy in the semi-honest setting, and let* $\mathsf{Sim}_{\mathsf{HE}}$ *and* η *be as in Definition B.1. Then for any security parameter* κ*,* L *polynomial in* κ*, list of circuits* $\mathsf{C}_1,\dots,\mathsf{C}_L$ *and list of keysets* $(pk_i, evk_i, sk_i)_{i=1}^{L}$ *generated by* $\mathsf{HE.KeyGen}(1^\kappa)$*, and input* x*, the statistical distance between*

$$\mathsf{HE.Eval}_{evk_L}^{\mathsf{C}_L}(\mathsf{HE.Rec}_{(L-1)\to L}(\mathsf{HE.Eval}_{evk_{L-1}}^{\mathsf{C}_{L-1}}(\cdots \mathsf{HE.Eval}_{evk_1}^{\mathsf{C}_1}(\mathsf{HE.Enc}_{pk_1}(x))))$$

and

$$\mathsf{Sim}_{\mathsf{HE}}(1^\kappa, pk_L, evk_L, \mathsf{C}_L(\mathsf{C}_{L-1}(\cdots C_1(x))))$$

is negligible in κ*.*

Proof. Since $\mathsf{HE.Rec}_{(L-1)\to L} = \mathsf{HE.Eval}_{evk_L}^{\mathsf{HE.Dec}_{sk_{L-1}}} \circ \mathsf{HE.Enc}_{pk_L}$ by definition, we have that

$$\begin{aligned}
& \mathsf{HE.Eval}_{evk_L}^{\mathsf{C}_L}(\mathsf{HE.Rec}_{(L-1)\to L}(\mathsf{HE.Eval}_{evk_{L-1}}^{\mathsf{C}_{L-1}}(\cdots \mathsf{HE.Eval}_{evk_1}^{\mathsf{C}_1}(\mathsf{HE.Enc}_{pk_1}(x))))\\
= \quad & \mathsf{HE.Eval}_{evk_L}^{\mathsf{C}_L\circ \mathsf{HE.Dec}_{sk_{L-1}}}(\mathsf{HE.Enc}_{pk_L}(\mathsf{HE.Eval}_{evk_{L-1}}^{\mathsf{C}_{L-1}}(\cdots \mathsf{HE.Eval}_{evk_1}^{\mathsf{C}_1}(\mathsf{HE.Enc}_{pk_1}(x))))\\
\approx_{\eta(\kappa)} \quad & \mathsf{Sim}_{\mathsf{HE}}(1^\kappa, pk_L, evk_L, \mathsf{C}_L(\mathsf{HE.Dec}_{sk_{L-1}}(\mathsf{HE.Eval}_{evk_{L-1}}^{\mathsf{C}_{L-1}}(\cdots \mathsf{HE.Eval}_{evk_1}^{\mathsf{C}_1}(\mathsf{HE.Enc}_{pk_1}(x))))))
\end{aligned}$$

which, by correctness of HE, is statistically indistinguishable from

$$\mathsf{Sim}_{\mathsf{HE}}(1^\kappa, pk_L, evk_L, \mathsf{C}_L(\mathsf{C}_{L-1}(\mathsf{C}_{L-2}(\cdots \mathsf{C}_1(x)))))$$

as long as L is polynomial in κ. By triangle inequality, the statement of the lemma follows. □

In the quantum setting, we need to take into account the fact that the input state may be part of some larger (possibly entangled) system. This leads to the following definition of *quantum circuit privacy* in the semi-honest setting:

Definition B.3 (Quantum circuit privacy—semi-honest setting)**.** A quantum homomorphic encryption scheme QHE has statistical circuit privacy in the semi-honest setting if there exists a quantum *PPT* algorithm $\mathsf{Sim}_{\mathsf{QHE}}$ and a negligible function η such that for any security parameter κ, depth parameter L, key set $(pk, \rho_{evk}, sk) \leftarrow \mathsf{QHE.KeyGen}(1^\kappa, 1^L)$, state σ, and circuit C with up to L T-gates:

$$\left\| \left(\Phi_{\mathsf{QHE.Eval}}^{\mathsf{C},\rho_{evk},pk} \circ \Phi_{\mathsf{QHE.Enc}}^{pk} \right) - \left(\Phi_{\mathsf{Sim}_{\mathsf{QHE}}}^{\rho_{evk},pk} \circ \Phi_{\mathsf{C}} \right) \right\|_{\diamond} \leq \eta(\kappa).$$

In this definition, Φ_{U} denotes the quantum channel induced by the circuit or functionality U. The diamond norm $\|\Phi_U\|_\diamond$ is defined in terms of the trace norm: $\|\Phi_U\|_\diamond := \max_\sigma \|(\Phi_U \otimes \mathbb{I})\sigma\|_1$ where the maximization is over input states σ.

We now show that the scheme TP can, with very little overhead, be modified to provide circuit privacy, as stated in Theorem 4.4 from Section 4.1:

Theorem 4.4. If HE has circuit privacy in the semi-honest setting, then TP can be adapted to a quantum homomorphic encryption scheme with circuit privacy against honest-but-curious adversaries.

Proof. We make the following alteration to the scheme TP: at the end of the evaluation procedure, the evaluator applies a (random) quantum one-time pad to the output of the evaluation, and updates the classical encryptions of the keys accordingly. The rest of the scheme remains exactly the same, and it is clear that this altered version of TP is still compact and correct. Recall from Section 3.4 that, at the end of the evaluation procedure, the data is recrypted into the Lth key set, and all unused gadgets are discarded. Thus, the output ciphertext is encrypted under the same key set, regardless of the number of T gates in the circuit C. (In a setting where gadgets are supplied on-demand, one has to be more careful not to reveal the number of T gates in the circuit.)

Intuitively, the randomization step at the end of the evaluation phase completely hides the circuit. Namely, the keys to the quantum one-time pads themselves are now entirely independent of the circuit, and circuit privacy of HE will ensure that even the classical encryption of these keys does not reveal any information about the computations performed on them.

To formalize this intuition, we define a quantum algorithm $\mathsf{Sim}_{\mathsf{TP}}$ satisfying the constraints given in Definition B.3. Let $\mathsf{Sim}_{\mathsf{HE}}$ be the classical simulator guaranteed to exist by the classical circuit privacy of HE (see Definition B.1). Given some security parameter κ, some keys $pk = (pk_1, \dots, pk_L)$ and $evk = (evk_1, \dots, evk_L)$, and some quantum state σ, let $\mathsf{Sim}_{\mathsf{TP}}$ apply a uniformly random quantum one-time pad to σ, and apply $\mathsf{Sim}_{\mathsf{HE}}(1^\kappa, pk_L, evk_L, \cdot)$ to the pad keys. The resulting classical-quantum state is the output of $\mathsf{Sim}_{\mathsf{TP}}$. This algorithm resembles $\mathsf{TP.Enc}$, but instead of calling $\mathsf{HE.Enc}$ (with pk_1) as a subroutine, it handles the pad key information using the classical simulator $\mathsf{Sim}_{\mathsf{HE}}$ (with pk_L).

If we can show that the trace distance

$$\left\| \left(\left(\Phi^{\mathsf{C}, \rho_{evk}, pk}_{\mathsf{TP.Eval}} \circ \Phi^{pk}_{\mathsf{TP.Enc}} \right) \otimes \mathbb{I} \right) \sigma - \left((\Phi^{\rho_{evk}, pk}_{\mathsf{Sim}_{\mathsf{TP}}} \circ \Phi_{\mathsf{C}}) \otimes \mathbb{I} \right) \sigma \right\|_1$$

is negligible for any quantum state σ of an appropriate dimension, then quantum circuit privacy of TP immediately follows from Definition B.3 and the definition of the diamond norm.

Write

$$\sigma_{sim} := \left((\Phi^{\rho_{evk}, pk}_{\mathsf{Sim}_{\mathsf{TP}}} \circ \Phi_{\mathsf{C}}) \otimes \mathbb{I} \right) \sigma, \qquad \text{and} \qquad \sigma_{eval} := \left(\left(\Phi^{\mathsf{C}, \rho_{evk}, pk}_{\mathsf{TP.Eval}} \circ \Phi^{pk}_{\mathsf{TP.Enc}} \right) \otimes \mathbb{I} \right) \sigma .$$

We study the state σ_{sim} in more detail, and show how to transform it into σ_{eval} in only a few (negligible) steps. As a result, the trace distance of these two states will be negligible.

By definition of the algorithm $\mathsf{Sim}_{\mathsf{TP}}$, the state σ_{sim} is equal to

$$\frac{1}{2^{2n}} \sum_{x,z \in \{0,1\}^n} \Bigg(\bigotimes_{i=1}^{n} \rho\left(\mathsf{Sim}_{\mathsf{HE}}(1^\kappa, pk_L, evk_L, x[i])\right) \otimes \bigotimes_{i=1}^{n} \rho\left(\mathsf{Sim}_{\mathsf{HE}}(1^\kappa, pk_L, evk_L, z[i])\right) \otimes \\ \left(\left(\bigotimes_{i=1}^{n} \mathsf{X}^{x[i]} \mathsf{Z}^{z[i]} \mathsf{C} \otimes \mathbb{I} \right) \sigma \left(\mathsf{C}^\dagger \bigotimes_{i=1}^{n} \mathsf{Z}^{z[i]} \mathsf{X}^{x[i]} \otimes \mathbb{I} \right) \right) \Bigg).$$

During the evaluation procedure of TP, the evaluator updates the keys to the quantum one-time pad for all n qubits in the circuit. These updates depend on the circuit that is being evaluated, some randomness r

from the Bell measurement outcomes[9] and of course on the initial one-time pad keys. Let $f_i^{\mathsf{C},r}(a,b)$ denote the X key on the ith qubit after the evaluation of some circuit C with randomness r, with $a,b \in \{0,1\}^n$ the initial pad keys before the evaluation procedure. Similarly, let $g_i^{\mathsf{C},r}(a,b)$ denote the Z key for that qubit.

At the end of the evaluation phase, the evaluator chooses bit strings x and z uniformly at random, so the final keys $f_i^{\mathsf{C},r}(a,b) \oplus x[i]$ and $g_i^{\mathsf{C},r}(a,b) \oplus z[i]$ are themselves completely uniform for any a,b. Therefore, the state σ_{sim} is actually equal to

$$\frac{1}{2^{4n}} \sum_{\substack{a,b,x,z \in \{0,1\}^n \\ r \in \{0,1\}^*}} \Pr_R(r) \Bigg(\bigotimes_{i=1}^{n} \rho\big(\mathsf{Sim}_{\mathsf{HE}}(1^\kappa, pk_L, evk_L, f_i^{\mathsf{C},r}(a,b) \oplus x[i])\big) \otimes$$
$$\bigotimes_{i=1}^{n} \rho\big(\mathsf{Sim}_{\mathsf{HE}}(1^\kappa, pk_L, evk_L, g_i^{\mathsf{C},r}(a,b) \oplus z[i])\big) \otimes$$
$$\Bigg(\Bigg(\bigotimes_{i=1}^{n} \mathsf{X}^{f_i^{\mathsf{C},r}(a,b) \oplus x[i]} \mathsf{Z}^{g_i^{\mathsf{C},r}(a,b) \oplus z[i]} \mathsf{C} \otimes \mathbb{I} \Bigg) \sigma \Bigg(\mathsf{C}^\dagger \bigotimes_{i=1}^{n} \mathsf{Z}^{g_i^{\mathsf{C},r}(a,b) \oplus z[i]} \mathsf{X}^{f_i^{\mathsf{C},r}(a,b) \oplus x[i]} \otimes \mathbb{I} \Bigg) \Bigg) \Bigg).$$

This is where the classical circuit privacy property kicks in: for any fixed i,a,b,C,r,x, the result of the probabilistic computation $\mathsf{Sim}_{\mathsf{HE}}(f_i^{\mathsf{C},r}(a,b) \oplus x[i])$ is statistically indistinguishable from the evaluation of the function $f_i^{\mathsf{C},r}(\cdot,\cdot) \oplus x[i]$ on the encryptions of a and b. Note however that the evaluation of $f_i^{\mathsf{C},r}$ is performed in several steps, with key switching in between. That is, separate functions h_1 through h_L are evaluated in each key set 1 through L, such that $f_i^{\mathsf{C},r} = h_L \circ \cdots \circ h_1$. We abstract away from the exact way that the function $f_i^{\mathsf{C},r}$ is broken up into these separate functions $h_1, ..., h_L$, and simply write

$$\mathsf{HE.Eval}_{1,\ldots,L}^{f_i^{\mathsf{C},r}(\cdot,\cdot) \oplus x[i]}(\mathsf{HE.Enc}_{pk_1}(a,b))$$

to denote

$$\mathsf{HE.Eval}_{evk_L}^{(\cdot \oplus x[i]) \circ h_L}(\mathsf{HE.Rec}_{(L-1)\to L}(\mathsf{HE.Eval}_{evk_{L-1}}^{h_{L-1}}(\cdots \mathsf{HE.Eval}_{evk_1}^{h_1}(\mathsf{HE.Enc}_{pk_1}(a,b))))).$$

By Lemma B.2, it follows that

$$\delta\Big(\mathsf{HE.Eval}_{1,\ldots,L}^{f_i^{\mathsf{C},r}(\cdot,\cdot) \oplus x[i]}(\mathsf{HE.Enc}_{pk_1}(a,b)), \mathsf{Sim}_{\mathsf{HE}}(1^\kappa, pk_L, evk_L, f_i^{\mathsf{C},r}(a,b) \oplus x[i])\Big) \leq \eta(\kappa)$$

for some negligible function η. We can rewrite this equation in terms of the trace distance to get

$$\Big\| \Big(\mathsf{HE.Eval}_{1,\ldots,L}^{f_i^{\mathsf{C},r}(\cdot,\cdot) \oplus x[i]}(\mathsf{HE.Enc}_{pk_1}(a,b)) - \mathsf{Sim}_{\mathsf{HE}}(1^\kappa, pk_L, evk_L, f_i^{\mathsf{C},r}(a,b) \oplus x[i])\Big\|_1 \leq 2\eta(\kappa)$$

[9]Although for the scheme TP, the measurement outcomes will in principle be uniformly distributed, we will not make this assumption here. In case of a malicious key generator, measurement outcomes might be correlated in some way. Therefore, we will simply assume that r is distributed according to some distribution R.

A similar result holds for $g_i^{\mathsf{C},r}(\cdot,\cdot)\oplus z[i]$. Using subadditivity of the trace norm with respect to the tensor product, it follows that the trace distance between σ_{sim} and

$$\frac{1}{2^{4n}}\sum_{\substack{a,b,x,z\in\{0,1\}^n\\ r\in\{0,1\}^*}}\Pr_R(r)\Bigg(\bigotimes_{i=1}^{n}\rho\left(\mathsf{HE.Eval}_{1,\ldots,L}^{f_i^{\mathsf{C},r}\oplus x[i]}(\mathsf{HE.Enc}_{pk_1}(a,b))\right)\otimes$$
$$\bigotimes_{i=1}^{n}\rho\left(\mathsf{HE.Eval}_{1,\ldots,L}^{g_i^{\mathsf{C},r}\oplus z[i]}(\mathsf{HE.Enc}_{pk_1}(a,b))\right)\otimes$$
$$\left(\left(\bigotimes_{i=1}^{n}\mathsf{X}^{f_i^{\mathsf{C},r}(a,b)\oplus x[i]}\mathsf{Z}^{g_i^{\mathsf{C},r}(a,b)\oplus z[i]}\mathsf{C}\otimes\mathbb{I}\right)\sigma\left(\mathsf{C}^{\dagger}\bigotimes_{i=1}^{n}\mathsf{Z}^{g_i^{\mathsf{C},r}(a,b)\oplus z[i]}\mathsf{X}^{f_i^{\mathsf{C},r}(a,b)\oplus x[i]}\otimes\mathbb{I}\right)\right)\Bigg)$$

is at most $4n\cdot\eta(\kappa)$. Note that this last state is exactly σ_{eval}, the result of putting σ through the channel

$$\left(\Phi_{\mathsf{TP.Eval}}^{\mathsf{C},\rho_{evk},pk}\circ\Phi_{\mathsf{TP.Enc}}^{pk}\right)\otimes\mathbb{I}.$$

We conclude that for any σ,

$$\|\sigma_{eval}-\sigma_{sim}\|_1\leq 4n\cdot\eta(\kappa)$$

for some negligible function η that does not depend on σ. Hence,

$$\left\|(\Phi_{\mathsf{TP.Eval}}^{\mathsf{C},\rho_{evk},pk}\circ\Phi_{\mathsf{TP.Enc}}^{pk})-(\Phi_{\mathsf{Sim_{TP}}}^{\rho_{evk},pk}\circ\Phi_{\mathsf{C}})\right\|_{\diamond}=$$
$$\max_{\sigma}\left\|\left((\Phi_{\mathsf{TP.Eval}}^{\mathsf{C},\rho_{evk},pk}\circ\Phi_{\mathsf{TP.Enc}}^{pk})\otimes\mathbb{I}\right)\sigma-\left((\Phi_{\mathsf{Sim_{TP}}}^{\rho_{evk},pk}\circ\Phi_{\mathsf{C}})\otimes\mathbb{I}\right)\sigma\right\|_1\leq 4n\cdot\eta(\kappa)$$

which is negligible if η is negligible. □

C Gadget construction using the garden-hose model

To construct the gadgets for specific decryption functions $\mathsf{HE.Dec}$, we will consider a purified version of the construction of the gadget state.

Consider the following task among two parties Alice and Bob. Alice corresponds to the party which creates the gadget, so she has knowledge of the secret key sk. Bob corresponds to the party which uses the gadget, therefore Bob has some input $\widetilde{a}$ and a state $\mathsf{P}^a|\psi\rangle$, where $a=\mathsf{HE.Dec}_{sk}(\widetilde{a})$. The end goal of the task is for Bob to possess the state $\mathsf{X}^{a'}\mathsf{Z}^{b'}|\psi\rangle$ for some a',b' that are computable from classical information known to Alice and Bob. The players pre-share a number of EPR pairs between them, and are only allowed to perform their actions without receiving any communication from the other player. We only consider strategies where the players perform Bell measurements and inverse phase gates on their local halves of the given EPR pairs (and in Bob's case also on the input qubit).

Before presenting strategies for this task, we first describe how this task translates to the creation of a gadget. Say Alice and Bob share $2m$ EPR pairs between them (i. e., they start with $4m$ qubits in total). Alice will perform Bell measurements between the halves of EPR pairs on her side, where the choices she makes depend on sk. Since both players act before receiving any information from the other player, the actions of Alice and Bob are not ordered—we first consider how to describe the state when Alice has acted on her local half. If Alice measures between, say, qubits s and t, with a two-bit outcome that describes the X and Z corrections, then we can instantly describe the qubits s and t on Bob's side as forming a fully entangled state—this teleportation of EPR halves is sometimes called *entanglement swapping*. Which out of the four Bell states is formed depends on the outcomes of Alice's measurement.

We also allow Alice to perform a $\mathsf{P}^\dagger$ gate on some qubits before teleportations. Note that if Alice measures on the qubits given by $\{(s_1,t_1),(s_2,t_2),\dots,(s_m,t_m)\}$, after applying an inverse phase gate when specified by the bit-string p, the state on Bob's side will exactly have the form of $\gamma_{x,z}\big(g(sk)\big)$, for some random binary strings x,z that correspond to the outcomes of Alice's Bell measurements. The quantum part of the gadget will be given by the reduced state on Bob's side, while the measurement choices and outcomes of Alice will form the accompanying classical information. The pairs that Bob chooses to perform Bell measurements on, which only depend on the encrypted information, are exactly the output of the function $\mathsf{TP.GenMeasurement}$.

An upper bound to the hardness of this task is given by the *garden-hose complexity* $\mathsf{HE.Dec}$, written $GH(\mathsf{HE.Dec})$, which is the least number of pipes needed for the players to compute it in the garden-hose model described in Section 2.3. This complexity measure is the main measure of hardness in the garden-hose model, and is relevant for the size of the gadgets in our construction.

The amount of space a Turing machine needs to compute any function f is closely related to it garden-hose complexity $GH(f)$. The following theorem, proved in [17], provides us with a general way of transforming space-efficient algorithms into garden-hose protocols.

Theorem C.1. *[17, Theorem 2.12] If $f:\{0,1\}^n\times\{0,1\}^n\to\{0,1\}$ is log-space computable, then $GH(f)$ is polynomial in n.*

Since the garden-hose complexity is defined in a non-uniform way, the strategies of the players are not necessarily easily computable. However, by inspection of the original proof, we see that the players effectively have to list all configurations for the Turing machine for f, and connect them according to the machine's transition function. For a log-space decryption function $\mathsf{HE.Dec}$, a player therefore only has to perform a polynomial-time computation to determine the strategy for a specific input.

The general construction is a direct consequence of the following lemma,[10] which was recently derived in the context of instantaneous non-local quantum computation.

Lemma C.2. *[53, Lemma 8, paraphrased] Assume Bob has a single qubit with state $\mathsf{P}^{f(x,y)}|\psi\rangle$, for binary strings $x,y\in\{0,1\}^n$, where Alice knows the string x and Bob knows y. Let $GH(f)$ be the garden-hose complexity of the function f. Then the following holds:*

1. *There exists an instantaneous protocol without any communication which uses $2GH(f)$ pre-shared EPR pairs after which a known qubit of Bob is in the state $\mathsf{X}^{g(\hat{x},\hat{y})}\mathsf{Y}^{h(\hat{x},\hat{y})}|\psi\rangle$. Here $\hat{x}$ depends only on x and the measurement outcomes of Alice, and $\hat{y}$ depends on y and the measurement outcomes of Bob.*

2. *The garden-hose complexities of the functions g and h are at most linear in the complexity of the function f.*

[10] The names of Alice and Bob have been swapped in order to fit the framework of this paper.

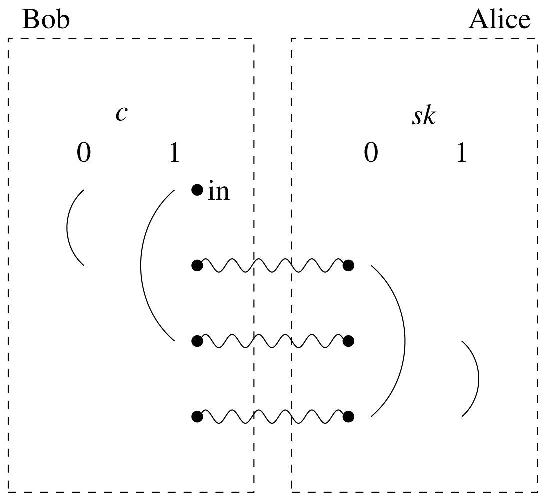

Figure 5: Garden-hose protocol for TOY.Dec. The snaky lines represent the EPR-pairs that form the resources to the protocol. The bended lines represent Bell measurements that Bob and Alice perform dependent on their inputs. For example, if $c = 0$ and $sk = 0$, the qubit starting at "in" is teleported through the first EPR-pair by Bob, then back through the third EPR-pair by Alice. It comes out on the bottom location on Bob's side.

For our purposes, only the first part of this result is required. The construction used in this lemma is a direct application of the garden-hose model [17], together with a simplifying step which was inspired by results on the garden-hose model by Klauck and Podder [38]. In our case, the function f will be the decryption function HE.Dec where Alice holds sk, and Bob holds $\widetilde{a}$.

C.1 Toy example

The garden-hose protocols that correspond to actual decryption functions will quickly become complicated in their description. Therefore, as an illustrative example, we will explicitly show how to convert a garden-hose protocol for the decryption function of a toy classical scheme TOY to a gadget. We do not claim TOY to be homomorphic at all; we only define its very simple decryption function and leave the rest of the scheme undefined.

Consider the following definition of TOY.Dec on ciphertext c and key sk of a single bit:

$$\mathsf{TOY.Dec}_{sk}(c) = sk \oplus c\,.$$

In Figure 5, a garden-hose protocol [17] for TOY.Dec is shown. For the protocol, Alice and Bob share three EPR-pairs which they use to teleport some qubit through, in a way that depends only on their own inputs c (for Bob) and sk (for Alice). The qubit always starts in the location marked "in." After the execution of the protocol, the qubit $|\psi\rangle$ should end up on Bob's side whenever $\mathsf{TOY.Dec}_{sk}(c) = 0$, and on Alice's side otherwise. For this small function, correctness is easily verified to hold for all possible inputs.

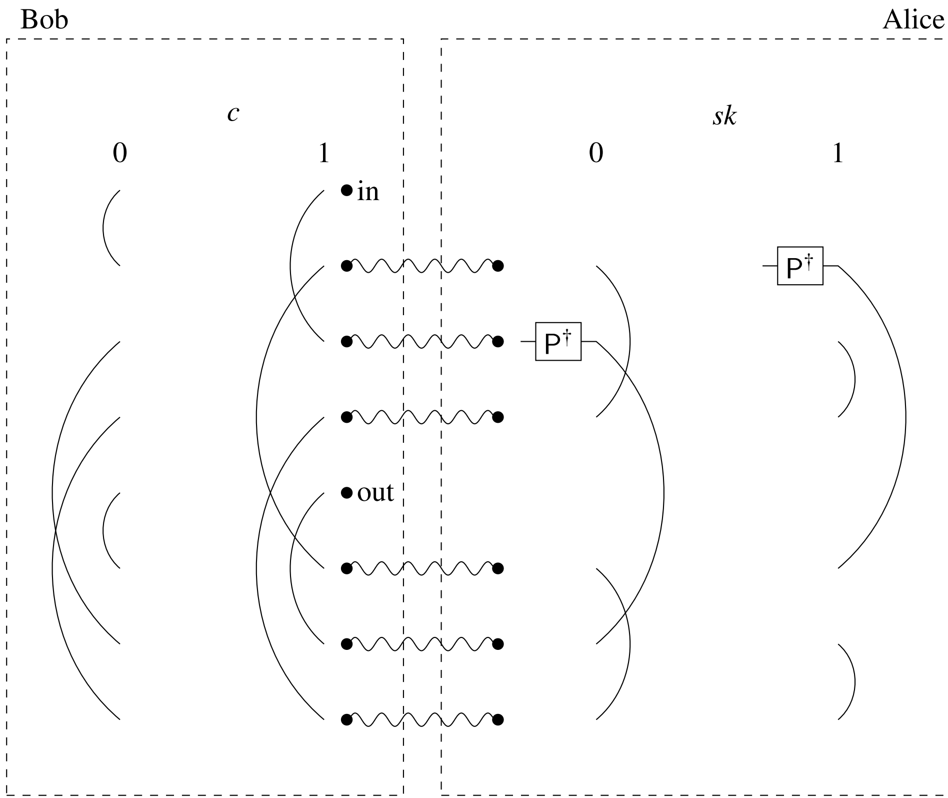

Figure 6: Removal of a possible P error using two copies of the garden-hose protocol for $\mathsf{TOY.Dec}$. For example, if $c=0$ and $sk=1$, the qubit is teleported through EPR pairs 1 and 4, with a $\mathsf{P}^\dagger$ applied to it by Alice in between. The input qubit always ends up on position "out."

The *garden-hose complexity* $GH(\mathsf{TOY.Dec})$ is the minimum amount of EPR-pairs needed for the computation of $\mathsf{TOY.Dec}$ in this way. See also [17].

Suppose that Bob teleports some qubit $\mathsf{P}^a\mathsf{X}^a\mathsf{Z}^b|\psi\rangle$ through the protocol, and sets his input c to be $\widetilde{a}$. Then whenever $a=\mathsf{TOY.Dec}_{sk}(\widetilde{a})=1$, the qubit will come out on Alice's side, and we will want to apply the correction $\mathsf{P}^\dagger$. To make sure that the correction is applied to $\mathsf{P}^a\mathsf{X}^a\mathsf{Z}^b|\psi\rangle$, Alice can apply a $\mathsf{P}^\dagger$ gate on all possible locations of the qubit. However, after this step, Alice and Bob do not know the location of the qubit, unless they share their inputs with one another non-homomorphically. The construction from [53, Lemma 8] solves this problem by applying the entire garden-hose protocol again in reverse: every EPR-half on which no measurement is performed, is connected through measurement with the EPR-half at the same position in the second copy of the protocol. That way, the (corrected) qubit follows the same path backwards, and always ends up on Bob's side at the "in" position of the second protocol (marked "out" in Figure 6).

After the execution of the protocol, the potential P error on the qubit $\mathsf{P}^a\mathsf{X}^a\mathsf{Z}^b|\psi\rangle$ has been removed, but additional Pauli transformations also have occurred as a result of the teleportations. The exact transformations depend on both the path the qubit has taken and the measurement outcomes of Alice and

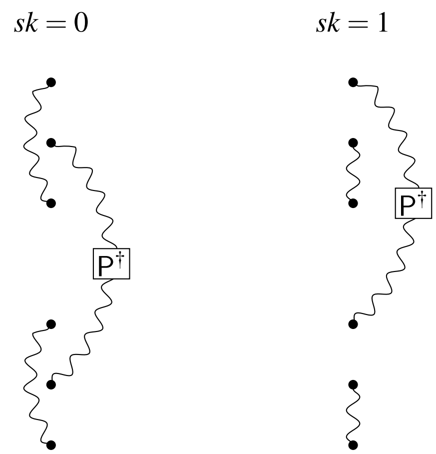

Figure 7: The two possible gadgets $\Gamma(sk)$ that $\mathsf{TP.GenGadget}$ might generate for $\mathsf{TOY.Dec}$. Effectively, a gadget consists of $2GH(\mathsf{TOY.Dec})$ EPR-pairs, ordered in a way that depends on sk. Some EPR-pairs have an additional $(\mathsf{P}^\dagger \otimes I)$ transformation applied to them. The evaluator's input c determines whether or not the input qubit is teleported through such a transformation, but an evaluator is unable to tell whether it is.

Bob. Therefore, Alice has to send all of this information (homomorphically encrypted) to Bob, so that he can update his keys to reflect the new state $\mathsf{X}^{a'}\mathsf{Z}^{b'}|\psi\rangle$ of the qubit.

Since the order of the Bell measurements does not influence the outcome of the protocol, Alice can perform her part of the protocol already during the key-generation phase. She starts by generating enough EPR-pairs for the gadget (six in this example), and performs the measurement on her own halves of the EPR-pairs. Effectively, this action generates six qubits that are entangled in some way that depends on sk (see Figure 7). Because of the random Pauli's that arise from the Bell measurements, Bob is not able to tell which pairs are connected without knowing Alice's measurement outcomes. To him, the state of the gadget is completely mixed (see equation (4.1)).

References

[1] SCOTT AARONSON, ALEXANDRU COJOCARU, ALEXANDRU GHEORGHIU, AND ELHAM KASHEFI: On the implausibility of classical client blind quantum computing, 2017. [arXiv:1704.08482] 4

[2] DORIT AHARONOV, MICHAEL BEN-OR, AND ELAD EBAN: Interactive proofs for quantum computations. In *Proc. 1st Innovations in Computer Science Conf. (ICS'10)*, pp. 453–469, 2010. [arXiv:1704.04487] 5

[3] GORJAN ALAGIC, ANNE BROADBENT, BILL FEFFERMAN, TOMMASO GAGLIARDONI, CHRISTIAN SCHAFFNER, AND MICHAEL ST. JULES: Computational security of quantum encryption. In

Proc. 9th Internat. Conf. on Information Theoretic Security (ICITS'16), pp. 47–71. Springer, 2016. [doi:10.1007/978-3-319-49175-2_3, arXiv:1602.01441] 10

[4] GORJAN ALAGIC AND BILL FEFFERMAN: On quantum obfuscation, 2016. [arXiv:1602.01771] 29

[5] PABLO ARRIGHI AND LOUIS SALVAIL: Blind quantum computation. *Internat. J. Quantum Information*, 4(5):883–898, 2006. [doi:10.1142/S0219749906002171, arXiv:quant-ph/0309152] 5

[6] GILAD ASHAROV, ABHISHEK JAIN, ADRIANA LÓPEZ-ALT, ERAN TROMER, VINOD VAIKUNTANATHAN, AND DANIEL WICHS: Multiparty computation with low communication, computation and interaction via threshold FHE. In *Proc. 31st Internat. Conf. on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'12)*, pp. 483–501. Springer, 2012. Cryptology Archive. [doi:10.1007/978-3-642-29011-4_29] 4

[7] DAVID A. BARRINGTON: Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. *J. Comput. System Sci.*, 38(1):150–164, 1989. Preliminary version in STOC'86. [doi:10.1016/0022-0000(89)90037-8] 22

[8] ÄMIN BAUMELER AND ANNE BROADBENT: Quantum private information retrieval has linear communication complexity. *Journal of Cryptology*, 28(1):161–175, 2015. [doi:10.1007/s00145-014-9180-2, arXiv:1304.5490] 21

[9] MICHAEL BEN-OR, CLAUDE CRÉPEAU, DANIEL GOTTESMAN, AVINATAN HASSIDIM, AND ADAM SMITH: Secure multiparty quantum computation with (only) a strict honest majority. In *Proc. 47th FOCS*, pp. 249–260. IEEE Comp. Soc. Press, 2006. [doi:10.1109/FOCS.2006.68, arXiv:0801.1544] 4

[10] DAN BONEH, EU-JIN GOH, AND KOBBI NISSIM: Evaluating 2-DNF formulas on ciphertexts. In *Proc. 2nd Theory of Cryptography Conf. (TCC'05)*, pp. 325–341. Springer, 2005. [doi:10.1007/978-3-540-30576-7_18] 2, 4

[11] ZVIKA BRAKERSKI, CRAIG GENTRY, AND VINOD VAIKUNTANATHAN: (Leveled) fully homomorphic encryption without bootstrapping. *ACM Trans. Comput. Theory*, 6(3):13:1–13:36, 2014. Preliminary version in ITCS'12. Cryptology Archive. [doi:10.1145/2633600] 4

[12] ZVIKA BRAKERSKI AND VINOD VAIKUNTANATHAN: Efficient fully homomorphic encryption from (standard) LWE. *SIAM J. Comput.*, 43(2):831–871, 2014. Preliminary version in FOCS'11. Cryptology Archive. [doi:10.1137/120868669] 2, 3, 4, 7, 25, 26

[13] ANNE BROADBENT: Delegating private quantum computations. *Canadian Journal of Physics*, 93(9):941–946, 2015. [doi:10.1139/cjp-2015-0030, arXiv:1506.01328] 5

[14] ANNE BROADBENT: Popescu-Rohrlich correlations imply efficient instantaneous nonlocal quantum computation. *Phys. Rev. A*, 94(2):022318, 2016. [doi:10.1103/PhysRevA.94.022318, arXiv:1512.04930] 11

[15] ANNE BROADBENT, JOSEPH FITZSIMONS, AND ELHAM KASHEFI: Universal blind quantum computation. In *Proc. 50th FOCS*, pp. 517–526. IEEE Comp. Soc. Press, 2009. [doi:10.1109/FOCS.2009.36, arXiv:0807.4154] 5, 28

[16] ANNE BROADBENT AND STACEY JEFFERY: Quantum homomorphic encryption for circuits of low T-gate complexity. In *Proc. 35th Ann. Internat. Cryptology Conf. (CRYPTO'15)*, pp. 609–629. Springer, 2015. [doi:10.1007/978-3-662-48000-7_30, arXiv:1412.8766] 2, 3, 5, 6, 7, 9, 10, 11, 16, 18, 20

[17] HARRY BUHRMAN, SERGE FEHR, CHRISTIAN SCHAFFNER, AND FLORIAN SPEELMAN: The garden-hose model. In *Proc. 4th Innovations in Theoretical Computer Science Conf. (ITCS'13)*, pp. 145–158. ACM Press, 2013. [doi:10.1145/2422436.2422455, arXiv:1109.2563] 3, 11, 21, 27, 28, 35, 36, 37

[18] ANDREW M. CHILDS: Secure assisted quantum computation. *Quantum Inf. Comput.*, 5(6):456–466, 2005. ACM DL link. [arXiv:quant-ph/0111046] 5

[19] WELL Y. CHIU, MARIO SZEGEDY, CHENGU WANG, AND YIXIN XU: The garden hose complexity for the equality function. In *Proc. 10th Internat. Conf. on Algorithmic Aspects in Information and Management (AAIM'14)*, pp. 112–123. Springer, 2014. [doi:10.1007/978-3-319-07956-1_11, arXiv:1312.7222] 26

[20] BENNY CHOR, EYAL KUSHILEVITZ, ODED GOLDREICH, AND MADHU SUDAN: Private information retrieval. *J. ACM*, 45(6):965–981, 1998. Preliminary version in FOCS'95. [doi:10.1145/293347.293350] 5

[21] RONALD CRAMER, IVAN DAMGÅRD, AND JESPER BUUS NIELSEN: Multiparty computation from threshold homomorphic encryption. In *Proc. 20th Internat. Conf. on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'01)*, pp. 280–299. Springer, 2001. Cryptology Archive. [doi:10.1007/3-540-44987-6_18] 29

[22] YFKE DULEK, CHRISTIAN SCHAFFNER, AND FLORIAN SPEELMAN: Quantum homomorphic encryption for polynomial-sized circuits. In *Proc. 36th Ann. Internat. Cryptology Conf. (CRYPTO'16)*, pp. 3–32. Springer, 2016. [doi:10.1007/978-3-662-53015-3_1, arXiv:1603.09717] 1

[23] FRÉDÉRIC DUPUIS, JESPER BUUS NIELSEN, AND LOUIS SALVAIL: Secure two-party quantum evaluation of unitaries against specious adversaries. In *Proc. 30th Ann. Internat. Cryptology Conf. (CRYPTO'10)*, pp. 685–706. Springer, 2010. [doi:10.1007/978-3-642-14623-7_37, arXiv:1009.2096] 21

[24] MAXIMILIAN FILLINGER: Lattice based cryptography and fully homomorphic encryption, 2012. Available here. 5

[25] KENT A. G. FISHER, ANNE BROADBENT, LYNDEN K. SHALM, ZHENYA YAN, JONATHAN LAVOIE, ROBERT PREVEDEL, THOMAS JENNEWEIN, AND KEVIN J. RESCH: Quantum computing on encrypted data. *Nature Communications*, 5(3074), 2014. [doi:10.1038/ncomms4074, arXiv:1309.2586] 5

[26] JOSEPH F. FITZSIMONS: Private quantum computation: An introduction to blind quantum computing and related protocols. *npj Quantum Information*, 3(23), 2017. [doi:10.1038/s41534-017-0025-3, arXiv:1611.10107] 28

[27] TOMMASO GAGLIARDONI, ANDREAS HÜLSING, AND CHRISTIAN SCHAFFNER: Semantic security and indistinguishability in the quantum world. In *Proc. 36th Ann. Internat. Cryptology Conf. (CRYPTO'16)*, pp. 60–89. Springer, 2016. [doi:10.1007/978-3-662-53015-3_3, arXiv:1504.05255] 10

[28] SHELLY GARG, CRAIG GENTRY, SHAI HALEVI, MARIANA RAYKOVA, ANANT SAHAI, AND BRENT WATERS: Candidate indistinguishability obfuscation and functional encryption for all circuits. *SIAM J. Comput.*, 45(3):882–929, 2016. Preliminary version in FOCS'13. Cryptology Archive. [doi:10.1137/14095772X] 2

[29] CRAIG GENTRY: Fully homomorphic encryption using ideal lattices. In *Proc. 41st STOC*, pp. 169–178. ACM Press, 2009. [doi:10.1145/1536414.1536440] 2, 4

[30] CRAIG GENTRY, SHAI HALEVI, AND VINOD VAIKUNTANATHAN: A simple BGN-type cryptosystem from LWE. In *Proc. 29th Internat. Conf. on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'10)*, pp. 506–522. Springer, 2010. Cryptology Archive. [doi:10.1007/978-3-642-13190-5_26] 4

[31] SHAFI GOLDWASSER, YAEL KALAI, RALUCA ADA POPA, VINOD VAIKUNTANATHAN, AND NICKOLAI ZELDOVICH: How to run Turing machines on encrypted data. In *Proc. 33rd Ann. Internat. Cryptology Conf. (CRYPTO'13)*, pp. 536–553. Springer, 2013. [doi:10.1007/978-3-642-40084-1_30] 2

[32] SHAFI GOLDWASSER, YAEL KALAI, RALUCA ADA POPA, VINOD VAIKUNTANATHAN, AND NICKOLAI ZELDOVICH: Reusable garbled circuits and succinct functional encryption. In *Proc. 45th STOC*, pp. 555–564. ACM Press, 2013. Cryptology Archive. [doi:10.1145/2488608.2488678] 2

[33] SHAFI GOLDWASSER AND SILVIO MICALI: Probabilistic encryption. *J. Comput. System Sci.*, 28(2):270–299, 1984. [doi:10.1016/0022-0000(84)90070-9] 2, 4

[34] SERGEY GORBUNOV, VINOD VAIKUNTANATHAN, AND HOETECK WEE: Attribute-based encryption for circuits. *J. ACM*, 62(6):45:1–45:33, 2015. Preliminary version in STOC'13. Cryptology Archive. [doi:10.1145/2824233] 2

[35] DANIEL GOTTESMAN: Theory of fault-tolerant quantum computation. *Phys. Rev. A*, 57(1):127–137, 1998. [doi:10.1103/PhysRevA.57.127, arXiv:quant-ph/9702029] 6

[36] DANIEL GOTTESMAN AND ISAAC L. CHUANG: Quantum Teleportation is a Universal Computational Primitive. *Nature*, 402:390–393, 1999. [doi:10.1038/46503, arXiv:quant-ph/9908010] 3

[37] YUVAL ISHAI AND ANAT PASKIN: Evaluating branching programs on encrypted data. In *Proc. 4th Theory of Cryptography Conf. (TCC'07)*, pp. 575–594. Springer, 2007. Cryptology Archive. [doi:10.1007/978-3-540-70936-7_31] 2, 30

[38] HARTMUT KLAUCK AND SUPARTHA PODDER: New bounds for the garden-hose model. In *Proc. 34th IARCS Ann. Conf. on Foundations of Software Tech. and Theor. Comput. Sci. (FSTTCS'14)*, pp. 481–492. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2014. [doi:10.4230/LIPIcs.FSTTCS.2014.481, arXiv:1412.4904] 26, 27, 36

[39] EYAL KUSHILEVITZ AND RAFAIL OSTROVSKY: Replication is not needed: Single database, computationally-private information retrieval. In *Proc. 38th FOCS*, pp. 364–373. IEEE Comp. Soc. Press, 1997. [doi:10.1109/SFCS.1997.646125] 4

[40] MIN LIANG: Symmetric quantum fully homomorphic encryption with perfect security. *Quantum Information Processing*, 12(12):3675–3687, 2013. [doi:10.1007/s11128-013-0626-5, arXiv:1304.5087] 5

[41] MIN LIANG: Quantum fully homomorphic encryption scheme based on universal quantum circuit. *Quantum Information Processing*, 14(8):2749–2759, 2015. [doi:10.1007/s11128-015-1034-9, arXiv:1410.2435] 5

[42] ODED MARGALIT: On the riddle of coding equality function in the garden hose model. In *Proc. 2014Information Theory and Applications Workshop (ITA'14)*, pp. 1–5. IEEE Comp. Soc. Press, 2014. [doi:10.1109/ITA.2014.6804262] 26

[43] MICHAEL NIELSEN AND ISAAC CHUANG: *Quantum Computation and Quantum Information*. Cambridge University Press, 2000. 6

[44] YINGKAI OUYANG, SI-HUI TAN, AND JOSEPH FITZSIMONS: Quantum homomorphic encryption from quantum codes, 2015. [arXiv:1508.00938] 5

[45] PASCAL PAILLIER: Public-key cryptosystems based on composite degree residuosity classes. In *Proc. 18th Internat. Conf. on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'99)*, pp. 223–238. Springer, 1999. [doi:10.1007/3-540-48910-X_16] 2, 4

[46] RONALD L. RIVEST, LEN ADLEMAN, AND MICHAEL L. DERTOUZOS: On data banks and privacy homomorphisms. *Foundations of Secure Computation*, 4(11):169–180, 1978. LINK. 2

[47] RONALD L. RIVEST, ADI SHAMIR, AND LEN ADLEMAN: A method for obtaining digital signatures and public-key cryptosystems. *Comm. ACM*, 21(2):120–126, 1978. [doi:10.1145/359340.359342] 4

[48] PETER P. ROHDE, JOSEPH F. FITZSIMONS, AND ALEXEI GILCHRIST: Quantum walks with encrypted data. *Phys. Rev. Lett.*, 109(15):150501, 2012. [doi:10.1103/PhysRevLett.109.150501, arXiv:1204.3370] 5

[49] AMIT SAHAI AND BRENT WATERS: How to use indistinguishability obfuscation: Deniable encryption, and more. In *Proc. 46th STOC*, pp. 475–484. ACM Press, 2014. Cryptology Archive. [doi:10.1145/2591796.2591825] 2

[50] TOMAS SANDER, ADAM YOUNG, AND MOTI YUNG: Non-interactive cryptocomputing for NC^1. In *Proc. 40th FOCS*, pp. 554–566. IEEE Comp. Soc. Press, 1999. [doi:10.1109/SFFCS.1999.814630] 4

[51] DAN SHEPHERD AND MICHAEL J. BREMNER: Temporally unstructured quantum computation. *Proc. Royal Soc. A*, 465(2105):1413–1439, 2009. [doi:10.1098/rspa.2008.0443] 3

[52] FLORIAN SPEELMAN: Position-Based Quantum Cryptography and the Garden-Hose Game. Master's thesis, University of Amsterdam, 2011. [arXiv:1210.4353] 27

[53] FLORIAN SPEELMAN: Instantaneous non-local computation of low T-depth quantum circuits. In *Proc. 11th Conf. Theory of Quantum Comput., Communic. and Cryptography (TQC'16)*, pp. 9:1–9:24. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016. [doi:10.4230/LIPIcs.TQC.2016.9, arXiv:1511.02839] 3, 11, 28, 35, 37

[54] SI-HUI TAN, JOSHUA A. KETTLEWELL, YINGKAI OUYANG, LIN CHEN, AND JOSEPH F. FITZSIMONS: A quantum approach to homomorphic encryption. *Scientific Reports*, 6, 2016. [doi:10.1038/srep33467, arXiv:1411.5254] 5

[55] VINOD VAIKUNTANATHAN: Computing blindfolded: New developments in fully homomorphic encryption. In *Proc. 52nd FOCS*, pp. 5–16. IEEE Comp. Soc. Press, 2011. [doi:10.1109/FOCS.2011.98] 8

[56] MARTEN VAN DIJK, CRAIG GENTRY, SHAI HALEVI, AND VINOD VAIKUNTANATHAN: Fully homomorphic encryption over the integers. In *Proc. 29th Internat. Conf. on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'10)*, volume 6110, pp. 24–43. Springer, 2010. Cryptology Archive. [doi:10.1007/978-3-642-13190-5_2] 2

[57] DUNJKO VEDRAN, JOSEPH F. FITZSIMONS, CHRISTOPHER PORTMANN, AND RENATO RENNER: Composable security of delegated quantum computation. In *Proc. 20th Internat. Conf. on the Theory and Application of Cryptology and Information Security (ASIACRYPT'14)*, pp. 406–425. Springer, 2014. [doi:10.1007/978-3-662-45608-8_22, arXiv:1301.3662] 5

[58] LI YU, CARLOS A. PÉREZ-DELGADO, AND JOSEPH F. FITZSIMONS: Limitations on information-theoretically secure quantum homomorphic encryption. *Phys. Rev. A*, 90(5), 2014. [doi:10.1103/PhysRevA.90.050303, arXiv:1406.2456] 5, 28

AUTHORS

Yfke Dulek
Ph. D. student
Centrum voor Wiskunde en Informatica, and
University of Amsterdam
The Netherlands
dulek@cwi.nl
https://www.cwi.nl/people/yfke-dulek

Christian Schaffner
Assistant professor
Centrum voor Wiskunde en Informatica, and
University of Amsterdam
The Netherlands
c.schaffner@uva.nl
http://homepages.cwi.nl/~schaffne

Florian Speelman
Postdoc
University of Copenhagen, Denmark
speelman@math.ku.dk
http://www.florianspeelman.nl

ABOUT THE AUTHORS

YFKE DULEK is a Ph. D. student at the Centrum voor Wiskunde en Informatica and the Institute for Logic, Language and Computation (University of Amsterdam), under the supervision of Christian Schaffner. There, she is also part of the research center QuSoft, where she works on quantum cryptography under computational assumptions, and in particular, quantum computing on encrypted data. She holds a B. Sc. in Artificial Intelligence from Utrecht University (2013) and a M. Sc. in Logic from the University of Amsterdam (2016).

CHRISTIAN SCHAFFNER received a diploma in mathematics from ETH Zurich (Switzerland) in 2003 and a Ph. D. in computer science from Aarhus University (Denmark) in 2007 under the supervision of Louis Salvail and Ivan Damgård. He is currently an Assistant Professor of Theoretical Computer Science at the Institute for Logic, Language and Computation (ILLC) at the University of Amsterdam. He is also a senior researcher at QuSoft, the Dutch research center for quantum software. Christian's research interests include quantum cryptography, cryptographic protocols, and (quantum) information theory.

FLORIAN SPEELMAN performed his Ph. D. research at the Centrum voor Wiskunde en Informatica in Amsterdam, under the supervision of Harry Buhrman, receiving his degree in 2016. He also holds a B. Sc. in Physics and a M. Sc. in Computational Science from the University of Amsterdam. Florian's research interests include quantum cryptography, computational complexity theory, and communication complexity. He is currently a postdoctoral researcher at QMATH, the Centre for the Mathematics of Quantum Theory, at the University of Copenhagen.