

# Quantum Homomorphic Encryption for Polynomial-Sized Circuits

Yfke Dulek<sup>1,2,3(✉)</sup>, Christian Schaffner<sup>1,2,3(✉)</sup>, and Florian Speelman<sup>2,3(✉)</sup>

<sup>1</sup> University of Amsterdam, Amsterdam, The Netherlands  
C.Schaffner@uva.nl

<sup>2</sup> CWI, Amsterdam, The Netherlands

<sup>3</sup> QuSoft, Amsterdam, The Netherlands  
{Y.M.Dulek,F.Speelman}@cwi.nl

**Abstract.** We present a new scheme for quantum homomorphic encryption which is compact and allows for efficient evaluation of arbitrary polynomial-sized quantum circuits. Building on the framework of Broadbent and Jeffery [BJ15] and recent results in the area of instantaneous non-local quantum computation [Spe15], 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”, and circuit privacy for the evaluator against passive adversaries.

**Keywords:** Homomorphic encryption · Quantum cryptography · Quantum teleportation · Garden-hose model

## 1 Introduction

Fully homomorphic encryption (FHE) is the holy grail of modern cryptography. Rivest et al. were the first to observe the possibility of manipulating encrypted data in a meaningful way, rather than just storing and retrieving it [RAD78]. After some partial progress [GM84, Pai99, BGN05, IP07] over the years, a breakthrough happened in 2009 when Gentry presented a fully-homomorphic encryption (FHE) scheme [Gen09]. Since then, FHE schemes have been simplified [VDGHV10] and based on more standard assumptions [BV11]. The exciting developments around FHE have sparked a large amount of research in other areas such as functional encryption [GKP+13a, GVW13, GKP+13b, SW14] and obfuscation [GGH+13].

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<sup>1</sup> 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<sup>2</sup>). While previous work on *quantum homomorphic encryption* has mostly focused on information-theoretic security (see Sect. 1.2 below for details), schemes that are based on computational assumptions have only recently been thoroughly investigated by Broadbent and Jeffery. In [BJ15], 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 [BJ15]. 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 quantum homomorphic scheme that allows evaluation of polynomial-sized quantum circuits?*

## 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 for circuits with polynomially many T gates. The scheme is secure against chosen plaintext attacks from quantum adversaries, as formalized by the security notion *q-IND-CPA security* defined by Broadbent and Jeffery [BJ15].

Like the schemes proposed in [BJ15], 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

<sup>1</sup> Here and throughout the article, we use “classical” to mean “non-quantum”.

<sup>2</sup> In contrast to *blind* or *delegated quantum computation* where some interaction between client and server is usually required, see Sect. 1.2 for references.

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 [BV11], each evaluation gadget of our scheme consists of a number of qubits which is polynomial in the security parameter.

In TP, we require exactly one evaluation gadget for every T gate that we would like to evaluate homomorphically. Intuitively, after a T gate is performed on a one-time-pad encrypted qubit  $X^a Z^b |\psi\rangle$ , the result might contain an unwanted phase  $P^a$  depending on the key  $a$  with which the qubit was encrypted, since  $T X^a Z^b |\psi\rangle = P^a X^a Z^b T |\psi\rangle$ . 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” [GC99] 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*<sup>3</sup> 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  $P^\dagger$  needs to be performed on the qubit or not. Very recent results by Speelman [Spe15] 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* [BFSS13] of the decryption procedure of the classical FHE is related to the size of our gadgets.

The quantum part of our evaluation gadget is strikingly simple, which provides a number of advantages. To start with, the evaluation of a T gate requires only one gadget, and 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 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.

---

<sup>3</sup> This term is not related to the term ‘instantaneous quantum computation’ [SB08], and instead first was used as a specific form of non-local quantum computation, one where all parties have to act simultaneously.

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.

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 [AJLA+12]. In that case, it is likely that our techniques can be adapted to perform *multiparty quantum computation* [BCG+06] 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) [RSA78, GM84, Pai99]; later on it became possible to combine several operations in a limited way [BGN05, GHV10, SYY99]. Gentry’s first fully homomorphic encryption scheme [Gen09] relied on several non-standard computational assumptions. Subsequent work [BGV12, BV11] 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) [KO97] that

breaks the lower bound on the amount of communication required for that task [CKGS98, Fil12].

While quantum fully homomorphic encryption (QFHE) is closely related to the task of blind or delegated quantum computation [Chi05, BFK09, ABOE10, VFPR14, FBS+14, Bro15a, Lia15], 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 et al. [YPDF14] showed that perfectly information-theoretically secure QFHE is not possible 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 [AS06, RFG12] or can only be used to evaluate a subset of all unitary transformations on the input [RFG12, Lia13, TKO+14]. 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 et al. provides information-theoretic security for circuits with at most a constant number of non-Clifford operations [OTF15].

Broadbent and Jeffery [BJ15] 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 [YPDF14] and work toward a (quantum) fully homomorphic encryption scheme.

Both of the schemes presented in [BJ15] are extensions of the scheme CL described in Sect. 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 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 polynomially many T gates can be efficiently evaluated.

### 1.3 Structure of the Paper

We start by introducing some notation in Sect. 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 Sect. 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 Sect. 4, along with an informal description of a circuit-private variant of the scheme. In Sect. 5, the rationale behind the quantum gadgets is explained, and some examples are discussed to clarify the construction. We conclude our work in Sect. 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 [NC00]). 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,

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

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  $X^a Z^b$  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  $X^{a_1} Z^{b_1} \otimes \dots \otimes X^{a_n} Z^{b_n}$ , this means that  $C$  is a Clifford operator if for any  $a_1, b_1, \dots, a_n, b_n \in \{0, 1\}$  there exist  $a'_1, b'_1, \dots, a'_n, b'_n \in \{0, 1\}$  such that (ignoring global phase):

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

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

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

(See for example [Got98].) 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 [BJ15], we choose this non-Clifford gate to be the T gate,

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

Note that the  $T$  gate, because it is non-Clifford, does not commute with the Pauli group. More specifically, we have  $TX^aZ^b = P^aX^aZ^bT$ . 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  $\rho$  or  $\sigma$ . Let  $\mathbb{I}_d$  denote the identity matrix of dimension  $d$ : this allows us to write the *completely mixed state* as  $\mathbb{I}_d/d$ .

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

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

This property is used in the construction of the *quantum one-time pad*: applying a random Pauli  $X^a Z^b$  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  $X^a Z^b$  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 [BJ15].

**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  $C(x_1, \dots, x_\ell)$ .

More formally, HE consists of the following four algorithms which run in probabilistic polynomial time in terms of their input and parameters [BV11]:

$(pk, evk, sk) \leftarrow \text{HE.KeyGen}(1^\kappa)$  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.

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

$c' \leftarrow \text{HE.Eval}_{evk}^C(c_1, \dots, c_\ell)$  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  $\text{Eval}$  as an evaluation of a function  $f$  instead of some canonical circuit for  $f$ , and write  $\text{HE.Eval}_{evk}^f(c_1, \dots, c_\ell)$  in this case.

$x' \leftarrow \text{HE.Dec}_{sk}(c)$  outputs a message  $x' \in \{0, 1\}$ , using the secret key  $sk$ .

In principle,  $\text{HE.Enc}_{pk}$  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  $\text{HE.Enc}_{pk}(x)$  to denote this bitwise encryption of the string  $x$ .

For HE to be a homomorphic encryption scheme, we require *correctness* in the sense that for any circuit  $C$ , there exists a negligible<sup>4</sup> function  $\eta$  such that, for any input  $x$ ,

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

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: a scheme is compact if there exists a polynomial  $p(\kappa)$  such that for any circuit  $C$  and any ciphertext  $c$ , the complexity of applying  $\text{HE.Dec}$  to the result of  $\text{HE.Eval}^C(c)$  is at most  $p(\kappa)$ .

A scheme that is both correct for all circuits and compact, is called *fully* homomorphic. If it is only correct for a subset of all possible circuits (e.g. all circuits with no multiplication gates) or if it 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 argument to the key generation function [Vai11].

We will use the notation  $\tilde{x}$  to denote the result of running  $\text{HE.Enc}_{pk}(x)$ : that is,  $\text{Dec}_{sk}(\tilde{x}) = x$  with overwhelming probability. In our construction, we will often deal with multiple classical key sets  $(pk_i, sk_i, evk_i)_{i \in I}$  indexed by some set  $I$ . In that case, we use the notation  $\tilde{x}^{[i]}$  to denote the result of  $\text{HE.Enc}_{pk_i}(x)$ , in order to avoid confusion. Here,  $pk_i$  does *not* refer to the  $i$ th bit of the public key: in case we want to refer to the  $i$ th 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  $\tilde{x}^{[i]}$  into an encryption  $\tilde{x}^{[j]}$  using a different key

<sup>4</sup> A *negligible function*  $\eta$  is a function such that for every positive integer  $d$ ,  $\eta(n) < 1/n^d$  for big enough  $n$ .



set  $j \neq i$ . To achieve this transformation, we define the procedure  $\text{HE.Rec}_{i \rightarrow j}$  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,  $\text{HE.Rec}_{i \rightarrow j}$  homomorphically evaluates the decryption of  $\widetilde{x}^{[i]}$ :

$$\text{HE.Rec}_{i \rightarrow j}(\widetilde{x}^{[i]}) := \text{HE.Eval}_{evk_j}^{\text{HE.Dec}}\left(\widetilde{sk}_i^{[j]}, \text{HE.Enc}_{pk_j}(\widetilde{x}^{[i]})\right).$$

**The Quantum Setting.** A quantum homomorphic encryption scheme QHE, as defined in [BJ15], 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 [BJ15]):

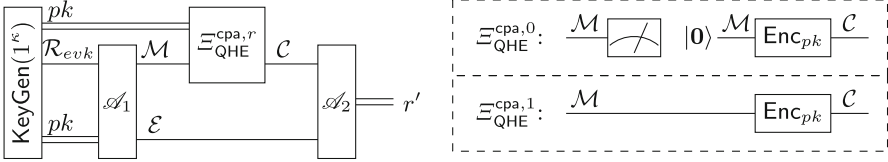
- $(pk, \rho_{evk}, sk) \leftarrow \text{QHE.KeyGen}(1^\kappa)$  where  $\kappa \in \mathbb{N}$  is the security parameter. In contrast to the classical case, the evaluation key is a quantum state.
- $\sigma \leftarrow \text{QHE.Enc}_{pk}(\rho)$  produces, for every valid public key  $pk$  and input state  $\rho$  from some message space, to a quantum cipherstate  $\sigma$  in some cipherspace.
- $\sigma' \leftarrow \text{QHE.Eval}_{\rho_{evk}}^{\text{C}}(\sigma)$  represents the evaluation of a circuit  $\text{C}$ . If  $\text{C}$  requires  $n$  input qubits, then  $\sigma$  should be a product of  $n$  cipherstates. The evaluation function maps it to a product of  $n'$  states in some output space, where  $n'$  is the number of qubits that  $\text{C}$  would output. The evaluation key  $\rho_{evk}$  is consumed in the process.
- $\rho' \leftarrow \text{QHE.Dec}_{sk}(\sigma')$  maps a single state  $\sigma'$  from the output space to a quantum state  $\rho'$  in the message space. Note that if the evaluation procedure  $\text{QHE.Eval}$  outputs a product of  $n'$  states, then  $\text{QHE.Dec}$  needs to be run  $n'$  times.

The decryption procedure differs from the classical definition in that we require the decryption to happen subsystem-by-subsystem: this is fundamentally different from the more relaxed notion of *indivisible schemes* [BJ15] where an auxiliary quantum register may be built up for the entire state, and the state can only be decrypted as a whole. In this work, we only consider the divisible definition.

**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 [BJ15, Definition 3.3] (see [GHS15] for a similar notion of the security of classical schemes against quantum attackers) and ensures semantic security [ABF+16]. We restate it here for completeness.

**Definition 1** [BJ15]. *The quantum CPA indistinguishability experiment with respect to a scheme QHE and a quantum polynomial-time adversary  $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ , denoted by  $\text{PubK}_{\mathcal{A}, \text{QHE}}^{\text{cpa}}(\kappa)$ , is defined by the following procedure:*

1.  $\text{KeyGen}(1^\kappa)$  is run to obtain keys  $(pk, sk, \rho_{evk})$ .
2. Adversary  $\mathcal{A}_1$  is given  $(pk, \rho_{evk})$  and outputs a quantum state on  $\mathcal{M} \otimes \mathcal{E}$ .
3. For  $r \in \{0, 1\}$ , let  $\Xi_{\text{QHE}}^{\text{cpa},r} : D(\mathcal{M}) \rightarrow D(\mathcal{C})$  be:  $\Xi_{\text{QHE}}^{\text{cpa},0}(\rho) = \text{QHE.Enc}_{pk}(|0\rangle\langle 0|)$  and  $\Xi_{\text{QHE}}^{\text{cpa},1}(\rho) = \text{QHE.Enc}_{pk}(\rho)$ . A random bit  $r \in \{0, 1\}$  is chosen and  $\Xi_{\text{QHE}}^{\text{cpa},r}$  is applied to the state in  $\mathcal{M}$  (the output being a state in  $\mathcal{C}$ ).
4. Adversary  $\mathcal{A}_2$  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  $\mathcal{A}$  wins the experiment.



**Fig. 1.** The quantum CPA indistinguishability experiment  $\text{PubK}_{\mathcal{A},\text{QHE}}^{\text{cpa}}(\kappa)$ . Double lines represent classical information flow, and single lines represent quantum information flow. The adversary  $\mathcal{A}$  is split up into two separate algorithms  $\mathcal{A}_1$  and  $\mathcal{A}_2$ , which share a working memory represented by the quantum state in register  $\mathcal{E}$ . [BJ15, reproduced with permission of the authors]

The game  $\text{PubK}_{\mathcal{A},\text{QHE}}^{\text{cpa}}(\kappa)$  is depicted in Fig. 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** [BJ15, Definition 3.3]. *A (classical or quantum) homomorphic encryption scheme  $\mathcal{S}$  is q-IND-CPA secure if for any quantum polynomial-time adversary  $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$  there exists a negligible function  $\eta$  such that:*

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

Analogously to  $\text{PubK}_{\mathcal{A},\mathcal{S}}^{\text{cpa}}(\kappa)$ , in the game  $\text{PubK}_{\mathcal{A},\mathcal{S}}^{\text{cpa-mult}}(\kappa)$ , the adversary can give multiple messages to the challenger, which are either all encrypted, or all replaced by zeros. Broadbent and Jeffery [BJ15] 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 et al. [BFSS13] to study a protocol for position-based quantum cryptography. The model recently saw new use, when Speelman [Spe15] 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 [Bro15b].)

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 Sect. 5.2 will depend on the number of pipes needed to compute the decryption function 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 [BJ15]: 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 Sect. 5.

Recall that when a T gate is applied to the state  $X^a Z^b |\psi\rangle$ , an unwanted P error may occur since  $TX^a Z^b = P^a X^a Z^b T$ . If  $a$  is known, this error can easily be corrected by applying  $P^\dagger$  whenever  $a = 1$ . However, as we will see, the evaluating party only has access to some encrypted version  $\tilde{a}$  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  $\mathsf{P}$  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  $\tilde{a}$ .

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  $\mathsf{HE}$ . If the decryption function  $\mathsf{HE.Dec}$  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,  $(\mathsf{P}^\dagger \otimes \mathbb{1})|\Phi^+\rangle$ , 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  $\tilde{a}$ . 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  $\mathsf{HE.Dec}_{sk}(\tilde{a})$  equals 1. The Pauli corrections can then be updated using the homomorphically-encrypted classical information and the measurement outcomes.

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

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

The quantum part of the gadget consists of  $2m$  qubits, with  $m$  some number which is polynomial in the security parameter  $\kappa$ . Let  $\{(s_1, t_1), (s_2, t_2), \dots, (s_m, t_m)\}$  be disjoint pairs in  $\{1, 2, \dots, 2m\}$ , 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  $sk$ ;

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

The tuple  $g(sk)$  is the classical information that determines the structure of the gadget as a function of the secret key  $sk$ . The length of  $g(sk)$  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  $\mathsf{HE}$  and the security parameter  $\kappa$ .

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

$$\gamma_{x,z}(g(sk)) := \prod_{i=1}^m \mathsf{X}^{x[i]} \mathsf{Z}^{z[i]} (\mathsf{P}^\dagger)^{p[i]} |\Phi^+\rangle \langle \Phi^+|_{s_i t_i} \mathsf{P}^{p[i]} \mathsf{Z}^{z[i]} \mathsf{X}^{x[i]}.$$

(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  $\{(s_1, t_1), (s_2, t_2), \dots, (s_m, t_m)\}$  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,

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

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

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

To summarize, the gadget consists of a quantum state  $\gamma_{x,z}(g(sk))$ , instantiated with randomly chosen  $x, z$ , the classical information denoting the random choice of  $x, z$ , and the other classical information  $g(sk)$  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  $\text{HE.Dec}$  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 reencrypting 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 Sect. 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 can be generated by an efficient (classical) algorithm  $\text{TP.GenMeasurement}(\tilde{a})$  which produces a list  $M$  containing  $m$  disjoint pairs of elements in  $\{0, 1, 2, \dots, 2m\}$ . 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  $\text{P}$  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  $\text{P}^a|\psi\rangle$ , i.e. the qubit has an extra phase gate if  $a = 1$ . Let  $\tilde{a}$  be an encrypted version of  $a$ , such that  $a = \text{HE.Dec}_{sk}(\tilde{a})$ . Then the evaluator performs Bell measurements on  $\Gamma_{pk'}(sk)$  and the input qubit, according to  $M \leftarrow \text{TP.GenMeasurement}(\tilde{a})$ . By construction, one out the  $2m + 1$  qubits is still unmeasured. This qubit will be in the state

$X^{a'}Z^{b'}|\psi\rangle$ , for some  $a'$  and  $b'$ , both of which are functions of the specification of the gadget, the measurement choices which depend on  $\tilde{a}$ , and the outcomes of the teleportation measurements. Also see Sect. 3.4 (and the full version of this paper) 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  $\text{HE.Dec}_{sk}(\tilde{a}) = 1$ , and avoids all such pairs whenever  $\text{HE.Dec}_{sk}(\tilde{a}) = 0$ .

### 3.2 Key Generation

Using the classical  $\text{HE.KeyGen}$  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  $\text{TP.KeyGen}(1^\kappa, 1^L)$  is defined as follows:

1. For  $i = 0$  to  $L$ : execute  $(pk_i, sk_i, evk_i) \leftarrow \text{HE.KeyGen}(1^\kappa)$  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  $\text{TP.GenGadget}_{pk_{i+1}}(sk_i)$  to create the gadget  $\Gamma_{pk_{i+1}}(sk_i)$  as described in Sect. 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

$$\bigotimes_{i=0}^{L-1} \left( \Gamma_{pk_{i+1}}(sk_i) \otimes |evk_i\rangle\langle evk_i| \right).$$

### 3.3 Encryption

The encryption procedure  $\text{TP.Enc}$  is identical to  $\text{CL.Enc}$ , 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  $\sigma$  is encrypted separately with a quantum one-time pad, and the pad key is (classically) encrypted and appended to the quantum encryption of  $\sigma$ , resulting in the classical-quantum state:

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

### 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  $\{\text{H}, \text{P}, \text{CNOT}\}$  and the  $\text{T}$  gate. A Clifford gate may affect multiple wires at the same time, while  $\text{T}$  gates can only affect a single qubit. Before the evaluation of a single gate  $U$ , the encryption of an  $n$ -qubit state  $\rho$  is of the form

$$(\mathbf{X}^{a_1} \mathbf{Z}^{b_1} \otimes \dots \otimes \mathbf{X}^{a_n} \mathbf{Z}^{b_n}) \rho (\mathbf{X}^{a_1} \mathbf{Z}^{b_1} \otimes \dots \otimes \mathbf{X}^{a_n} \mathbf{Z}^{b_n}).$$

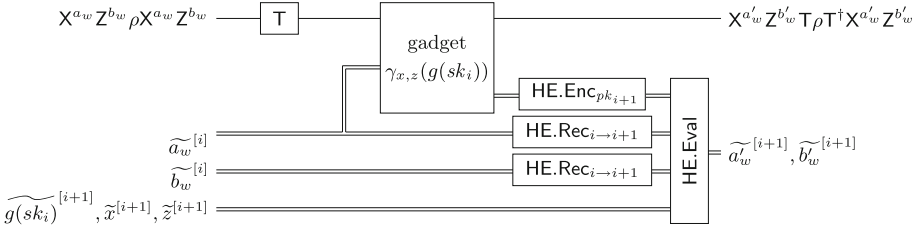
The evaluating party holds the encrypted versions  $\widetilde{a}_1^{[i]}, \dots, \widetilde{a}_n^{[i]}$  and  $\widetilde{b}_1^{[i]}, \dots, \widetilde{b}_n^{[i]}$ , with respect to the  $i$ th key set for some  $i$  (initially,  $i = 0$ ). The goal is to obtain a quantum encryption of the state  $U\rho U^\dagger$ , 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  $i$ th key. If  $U$  is a  $\text{T}$  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 `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. For a detailed description of this computation, also see the full version of this paper, or e.g. [BJ15, Appendix C].
- If  $U = \text{T}$ , the evaluator should start out by applying a  $\text{T}$  gate to the appropriate wire  $w$ . Afterwards, the qubit at wire  $w$  is in the state

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

In order to remove the  $\text{P}$  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 correct key to be able to compute measurements  $M \leftarrow \text{TP.GenMeasurement}(\widetilde{a}_w^{[i]})$  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 Fig. 2 and see the full version of this paper for a detailed description of the update algorithm. After these computations, the evaluator also recrypts the keys of all other wires into the  $(i + 1)$ th key set.

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



**Fig. 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. 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 decryption procedures, is contained in the evaluation key.

### 3.5 Decryption

The decryption procedure is identical to CL.Dec. For each qubit, HE.Dec<sub>sk<sub>L</sub></sub> 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  $C|\psi\rangle$ .

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 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  $f$ , the complexity of applying HE.Dec to the output of HE.Eval <sup>$f$</sup>  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  $\kappa$  and independent of  $C$ , so we conclude that TP is compact.  $\square$

## 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 Sect. 2.2). Like in the security proofs in [BJ15], 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 already know to be small. The aim of this section is to prove the following theorem:

**Theorem 2.** *If HE is  $q$ -IND-CPA secure, then TP is  $q$ -IND-CPA secure for circuits containing up to polynomially (in  $\kappa$ ) many T gates.*

In order to prove Theorem 2, we first prove that an efficient adversary's performance in the indistinguishability game is only negligibly different whether or not he receives a real evaluation key with real gadgets, or just a completely mixed quantum state with encryptions of 0's accompanying them (Corollary 1). 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  $\ell \in \{0, \dots, L\}$ , let  $\text{TP}^{(\ell)}$  be identical to TP, except for the key generation procedure:  $\text{TP}^{(\ell)}.\text{KeyGen}$  replaces, for every  $i \geq \ell$ , all classical information accompanying the  $i$ th gadget with the all-zero string before encrypting it. For any number  $i$ , define the shorthand

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

As seen in Sect. 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,

$$\begin{aligned} \text{TP}^{(\ell)}.\text{KeyGen}(1^\kappa, 1^L) &:= \bigotimes_{i=0}^{L-1} |evk_i\rangle\langle evk_i| \otimes \bigotimes_{i=0}^{\ell-1} \Gamma_{pk_{i+1}}(sk_i) \otimes \\ &\quad \bigotimes_{i=\ell}^{L-1} \left( \rho(\text{HE.Enc}_{pk_{i+1}}(0^{|g_i|})) \right) \otimes \\ &\quad \frac{1}{2^{2m}} \sum_{x,z \in \{0,1\}^m} \rho(\text{HE.Enc}_{pk_{i+1}}(0^m, 0^m)) \otimes \gamma_{x,z}(g_i). \end{aligned}$$

Intuitively, one can view  $\text{TP}^{(\ell)}$  as the scheme that provides only  $\ell$  usable gadgets in the evaluation key. Note that  $\text{TP}^{(L)} = \text{TP}$ , and that in  $\text{TP}^{(0)}$ , 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

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

With the definitions of the new schemes, we can lay out the steps to prove Theorem 2 in more detail. First, we show that in the quantum CPA indistinguishability experiment, any efficient adversary interacting with  $\text{TP}^{(\ell)}$  only has negligible advantage over an adversary interacting with  $\text{TP}^{(\ell-1)}$ , i.e. the scheme where the classical information  $g_{\ell-1}$  is removed (Lemma 1). By iteratively applying this argument, we are able to argue that the advantage of an adversary who interacts with  $\text{TP}^{(L)}$  over one who interacts with  $\text{TP}^{(0)}$  is also negligible (Corollary 1). Finally, we conclude the proof by arguing that  $\text{TP}^{(0)}$  is q-IND-CPA secure by comparison to the CL scheme.

**Lemma 1.** *Let  $0 < \ell \leq L$ . If HE is q-IND-CPA secure, then for any quantum polynomial-time adversary  $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ , there exists a negligible function  $\eta$  such that*

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

*Proof.* The difference between schemes  $\text{TP}^{(\ell)}$  and  $\text{TP}^{(\ell-1)}$  lies in whether the gadget state  $\gamma_{x_{\ell-1}, z_{\ell-1}}(g_{\ell-1})$  is supplemented with its classical information  $\widetilde{g_{\ell-1}}, \widetilde{x_{\ell-1}}, \widetilde{z_{\ell-1}}$ , or just with an encryption of  $0^{|g_{\ell-1}|+2m}$ .

Let  $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$  be an adversary for the game  $\text{PubK}_{\mathcal{A}, \text{TP}^{(\ell)}}^{\text{cpa}}(\kappa)$ . We will define an adversary  $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$  for  $\text{PubK}_{\mathcal{A}', \text{HE}}^{\text{cpa-mult}}(\kappa)$  that will either simulate the game  $\text{PubK}_{\mathcal{A}, \text{TP}^{(\ell)}}^{\text{cpa}}(\kappa)$  or  $\text{PubK}_{\mathcal{A}, \text{TP}^{(\ell-1)}}^{\text{cpa}}(\kappa)$ . Which game is simulated will depend on some  $s \in_R \{0, 1\}$  that is unknown to  $\mathcal{A}'$  himself. Using the assumption that HE is q-IND-CPA secure, we are able to argue that  $\mathcal{A}'$  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  $\text{TP}^{(\ell)}$  and  $\text{TP}^{(\ell-1)}$ . The structure of this proof is very similar to e.g. Lemma 5.3 in [BJ15]. The adversary  $\mathcal{A}'$  acts as follows (see also Fig. 3):

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

The adversary  $\mathcal{A}'_1$  also generates the extra padding inputs that correspond to the already-removed gadgets  $\ell$  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 Eq. 1), the adversary can generate them without needing any extra information.

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

Because HE is q-IND-CPA secure, the probability that  $\mathcal{A}'$  wins  $\text{PubK}_{\mathcal{A}', \text{HE}}^{\text{cpa-mult}}(\kappa)$ , i.e. that  $s' \equiv s$ , is at most  $\frac{1}{2} + \eta'(\kappa)$  for some negligible function  $\eta'$ . There are two scenarios in which  $\mathcal{A}'$  wins the game:

- $s = 1$  and  $\mathcal{A}$  guesses  $r$  correctly: If  $s = 1$ , the game that is being simulated is  $\text{PubK}_{\mathcal{A}, \text{TP}(\ell)}^{\text{cpa}}(\kappa)$ . If  $\mathcal{A}$  wins the simulated game ( $r \equiv r'$ ), then  $\mathcal{A}'$  will correctly output  $s' = 1$ . (If  $\mathcal{A}$  loses, then  $\mathcal{A}'$  outputs 0, and loses as well).
- $s = 0$  and  $\mathcal{A}$  does not guess  $r$  correctly: If  $s = 0$ , the game that is being simulated is  $\text{PubK}_{\mathcal{A}, \text{TP}(\ell-1)}^{\text{cpa}}(\kappa)$ . If  $\mathcal{A}$  loses the game ( $r \not\equiv r'$ ), then  $\mathcal{A}'$  will correctly output  $s' = 0$ . (If  $\mathcal{A}$  wins, then  $\mathcal{A}'$  outputs 1 and loses).

From the above, we conclude that

$$\begin{aligned} & \Pr[s = 1] \cdot \Pr[\text{PubK}_{\mathcal{A}, \text{TP}(\ell)}^{\text{cpa}}(\kappa) = 1] + \Pr[s = 0] \cdot \Pr[\text{PubK}_{\mathcal{A}, \text{TP}(\ell-1)}^{\text{cpa}}(\kappa) = 0] && \leq \frac{1}{2} + \eta'(\kappa) \\ \Leftrightarrow & \frac{1}{2} \Pr[\text{PubK}_{\mathcal{A}, \text{TP}(\ell)}^{\text{cpa}}(\kappa) = 1] + \frac{1}{2} \left( 1 - \Pr[\text{PubK}_{\mathcal{A}, \text{TP}(\ell-1)}^{\text{cpa}}(\kappa) = 1] \right) && \leq \frac{1}{2} + \eta'(\kappa) \\ \Leftrightarrow & \Pr[\text{PubK}_{\mathcal{A}, \text{TP}(\ell)}^{\text{cpa}}(\kappa) = 1] - \Pr[\text{PubK}_{\mathcal{A}, \text{TP}(\ell-1)}^{\text{cpa}}(\kappa) = 1] && \leq 2\eta'(\kappa) \end{aligned}$$

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

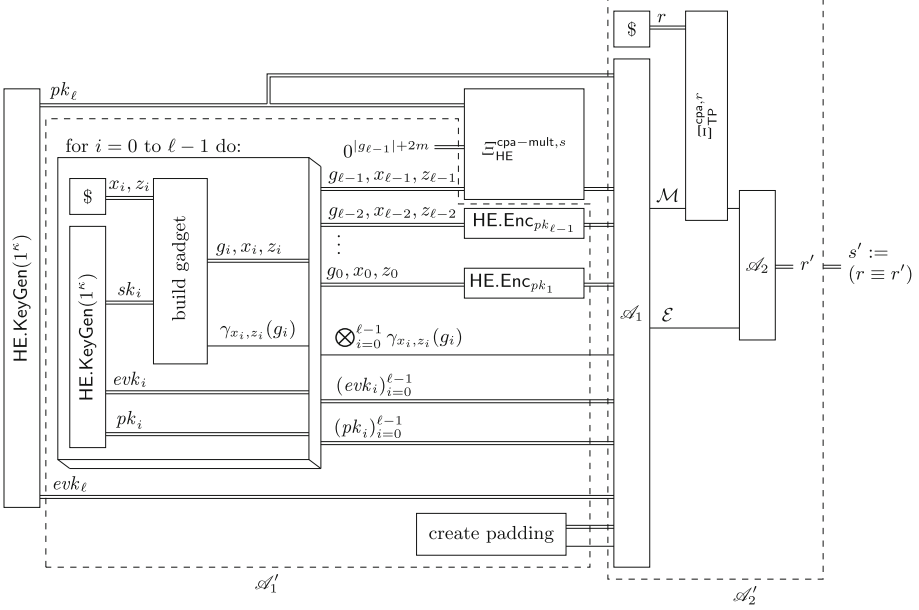
By applying Lemma 1 iteratively,  $L$  times in total, we can conclude that the difference between  $\text{TP}^{(L)}$  and  $\text{TP}^{(0)}$  is negligible, because the sum of polynomially many negligible functions is still negligible:

**Corollary 1.** *If  $L$  is polynomial in  $\kappa$ , then for any quantum polynomial-time adversary  $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ , there exists a negligible function  $\eta$  such that*

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

Using Corollary 1, we can finally prove the q-IND-CPA security of our scheme  $\text{TP} = \text{TP}^{(L)}$ .

*Proof of Theorem 2.* The scheme  $\text{TP}^{(0)}$  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  $\text{TP}^{(0)}$ , 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



**Fig. 3.** A strategy for the game  $\text{PubK}_{\mathcal{A}, \text{TP}^{(\ell)}}^{\text{cpa-mult}}(\kappa)$ , using an adversary  $\mathcal{A}$  for  $\text{PubK}_{\mathcal{A}, \text{TP}^{(\ell)}}^{\text{cpa}}(\kappa)$  as a subroutine. All the wires that form an input to  $\mathcal{A}'_1$  together form the evaluation key and public key for  $\text{TP}^{(\ell)}$  or  $\text{TP}^{(\ell-1)}$ , depending on  $s$ . Note that  $\Xi_{\text{TP}^{(\ell)}}^{\text{cpa}, r} = \Xi_{\text{TP}^{(\ell-1)}}^{\text{cpa}, r} = \Xi_{\text{TP}^{(\ell-1)}}^{\text{cpa}, r}$ , so  $\mathcal{A}'_2$  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  $\ell$  up to  $L - 1$ , as described in the definition of  $\mathcal{A}'_1$ .

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  $\text{HE.KeyGen}(1^\kappa, 1^L)$ ). Therefore, we can safely ignore these keys as well.

In [BJ15, Lemma 5.3], it is shown that CL is q-IND-CPA secure. Because of the similarity between CL and  $\text{TP}^{(0)}$ , the exact same proof shows that  $\text{TP}^{(0)}$  is q-IND-CPA secure as well, that is, for any  $\mathcal{A}$  there exists a negligible function  $\eta'$  such that

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

Combining this result with Corollary 1, it follows that

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

Since the sum of two negligible functions is itself negligible, we have proved Theorem 2.  $\square$

## 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  $\rho$  by the evaluator. Obviously, the output value  $C\rho C^\dagger$  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  $C\rho C^\dagger$  itself.

We claim that circuit privacy for TP in the semi-honest setting (i.e. against passive adversaries<sup>5</sup>) can be obtained by modifying the scheme only slightly, given that the classical encryption scheme has the circuit privacy property.

**Theorem 3.** *If HE has circuit privacy in the semi-honest setting, then TP can be adapted to a quantum homomorphic encryption scheme with circuit privacy.*

*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. Because of space constraints, the full detailed proof is given in the full version of this paper.  $\square$

## 5 Constructing the Gadgets

In this section we will first show how 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  $\text{NC}^1$ ). 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. Afterwards, in Sect. 5.2, we will present a larger toolkit to construct gadgets, which is efficient for a larger class of possible decryption functions. To illustrate these techniques, we apply these tools to create gadgets for schemes that are based on Learning With Errors (LWE). Finally, we will reflect on the possibility of constructing these gadgets in scenarios where quantum power is limited.

### 5.1 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,

<sup>5</sup> Note that there various ways to define passive adversaries in the quantum setting [DNS10, BB14]. Here, we are considering adversaries that follow all protocol instructions exactly.

which states that all boolean circuits of logarithmic depth can be encoded as polynomial-sized width-5 permutation branching programs. Every instruction of such a branching program will be encoded as connections between five Bell pairs.

**Definition 3.** A width- $k$  permutation branching program of length  $L$  on an input  $x \in \{0, 1\}^n$  is a list of  $L$  instructions of the form  $\langle i_\ell, \sigma_\ell^1, \sigma_\ell^0 \rangle$ , for  $1 \leq \ell \leq L$ , such that  $i_\ell \in [n]$ , and  $\sigma_\ell^1$  and  $\sigma_\ell^0$  are elements of  $S_k$ , i.e., permutations of  $[k]$ . The program is executed by composing the permutations given by the instructions 1 through  $L$ , selecting  $\sigma_\ell^1$  if  $x_{i_\ell} = 1$  and selecting  $\sigma_\ell^0$  if  $x_{i_\ell} = 0$ . 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 [Bar89].

**Theorem 4** (Barrington [Bar89]). *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 TP.GenGadget $_{pk'}(sk)$ , and usage instructions TP.GenMeasurement( $\tilde{a}$ ).

**Theorem 5.** *Let HE.Dec $_{sk}(\tilde{a})$  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  $\kappa$  is the security parameter, then there exist gadgets for TP of size polynomial in  $\kappa$ .*

*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  $\tilde{a}$ . 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  $\tilde{a}$ . Since by assumption the circuit is of depth  $O(\log \kappa)$ , there exists width-5 permutation branching program  $\mathcal{P}$  of length  $L = \kappa^{O(1)}$ , with the following properties. We write

$$\mathcal{P} = (\langle i_1, \sigma_1^1, \sigma_1^0 \rangle, \langle i_2, \sigma_2^1, \sigma_2^0 \rangle, \dots, \langle i_L, \sigma_L^1, \sigma_L^0 \rangle)$$

as the list of instructions of the width-5 permutation branching program. Without loss of generality<sup>6</sup>, we can assume that the instructions alternately depend on bits

<sup>6</sup> 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.

of  $\tilde{a}$  and bits of  $sk$ . That is, the index  $i_\ell$  refers to a bit of  $\tilde{a}$  if  $\ell$  is odd, and to a bit of  $sk$  if  $\ell$  is even. There are  $L$  instructions in total, of which  $L/2$  are odd-numbered and  $L/2$  are even.

The output of  $\text{TP.GenGadget}_{pk'}(sk)$ , i.e., the list of pairs 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  $\ell \leq L$ , we connect ten qubits in the following way. Suppose the  $\ell^{\text{th}}$  instruction evaluates to some permutation  $\sigma_\ell := \sigma_\ell^{sk_{i_\ell}}$ . Label the 10 qubits of this part of the gadget by  $1_{\ell,\text{in}}, 2_{\ell,\text{in}}, \dots, 5_{\ell,\text{in}}$  and  $1_{\ell,\text{out}}, 2_{\ell,\text{out}}, \dots, 5_{\ell,\text{out}}$ . These will correspond to 5 EPR pairs, connected according to the permutation:  $(1_{\ell,\text{in}}, \sigma_\ell(1)_{\ell,\text{out}}), (2_{\ell,\text{in}}, \sigma_\ell(2)_{\ell,\text{out}}), \dots$ , up to  $(5_{\ell,\text{in}}, \sigma_\ell(5)_{\ell,\text{out}})$ .

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

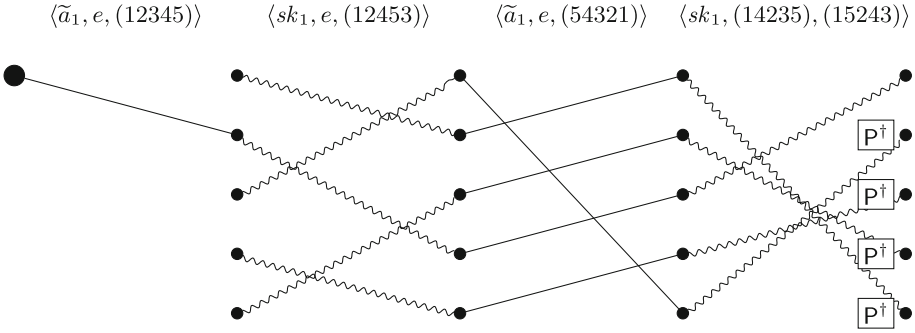
For this construction,  $\text{TP.GenMeasurement}(\tilde{a})$  will be given by the odd instructions, which depend on the bits of  $\tilde{a}$ . Again, for all odd  $\ell \leq L$ , let  $\sigma_\ell := \sigma_\ell^{\tilde{a}_{i_\ell}}$  be the permutation given by the evaluation of instruction  $\ell$  on  $\tilde{a}$ . For all  $\ell$  strictly greater than one, the measurement instructions will be: perform a Bell measurement according to the permutation  $\sigma_\ell$  between the ‘out’ qubits of the previous set, and the ‘in’ qubits of the next. The measurement pairs will then be  $(1_{\ell-1,\text{out}}, \sigma(1)_{\ell,\text{in}}), (2_{\ell-1,\text{out}}, \sigma(2)_{\ell,\text{in}}), \dots$ , up to  $(5_{\ell-1,\text{out}}, \sigma(5)_{\ell,\text{in}})$ .

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  $\text{HE.Dec}_{sk}(\tilde{a}) = 0$  then the product, say  $\tau$ , 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,  $\tau$  is a  $k$ -cycle. After teleporting a qubit through these EPR pairs, with teleportation measurements chosen accordingly, the input qubit will be present at  $\tau(1)_{L,\text{out}}$ , with an inverse phase gate if  $\tau(1)$  is unequal to 1.

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

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  $\langle i_L, (\sigma_L^1)^{-1}, (\sigma_L^0)^{-1} \rangle$ , then  $\langle i_{L-1}, (\sigma_{L-1}^1)^{-1}, (\sigma_{L-1}^0)^{-1} \rangle$ , with final instruction  $\langle i_1, (\sigma_1^1)^{-1}, (\sigma_1^0)^{-1} \rangle$ . One small detail is that  $i_L$  is used twice in a row, breaking the alternation; the user of the gadget can simply perform the



**Fig. 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.

measurements that correspond to the identity permutation  $e$  in between, since  $(\sigma_L^0)(\sigma_L^0)^{-1} = (\sigma_L^1)(\sigma_L^1)^{-1} = e$ .

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  $L$ , of which the instructions alternate between depending on  $\tilde{a}$  and depending on  $sk$ , is  $2 \cdot (5L) = 10L$ .  $\square$

*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  $\text{HE.Dec}_{sk}(\tilde{a})$  is  $sk_1 \text{ OR } \tilde{a}_1$ . Then, for one possible example set of values of  $\tilde{a}$  and  $sk$ , half of the gadget and measurements will be as given in Fig. 4. To complete this gadget, the same construction is appended, reflected horizontally.

## 5.2 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  $\text{NC}^1$ . Secondly, for some specific decryption functions, executing the construction of Sect. 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  $\kappa^2$  — a direct approach can often easily improve on the exponent of the polynomial. See also the garden-hose protocols for equality [Mar14, CSWX14] and the majority function [KP14] for examples of non-trivial protocols that are much more efficient than applying Barrington’s theorem as a black box.

**Theorem 6.** *Let  $\text{HE.Dec}_{sk}(\tilde{a})$  be the decryption function of the classical homomorphic encryption scheme HE. If  $\text{HE.Dec}$  is computable by a Turing machine that uses space  $O(\log \kappa)$ , where  $\kappa$  is the security parameter, then there exist gadgets for TP of size polynomial in  $\kappa$ .*

A detailed explanation of how to construct a gadget for a log-space computation is given in the full version of this paper. All more-complicated constructions use a different language than the direct encoding of the previous section: 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*. 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.

Besides clarifying the log-space construction, viewing the gadget construction as an instantiation of the garden-hose mode 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 [Spe11, BFSS13, KP14], 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 or 6.

The scheme by Brakerski and Vaikuntanathan [BV11] 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 in the full version of our paper.

### 5.3 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. TP.Enc and TP.Dec only require local application of Pauli operators to a quantum state, but TP.KeyGen is more involved because of the gadget construction. However, when supplied with a set of EPR pairs from the server (or any other untrusted source), the client can generate the quantum evaluation key for TP using only Pauli and swap operations. Even if the server produces some other state than the claimed list of EPR pairs, the client can prevent the leakage of information about her input by encrypting the input with random Pauli operations. More details are supplied in the appendix of the full version of this paper.

Alternatively, TP can be regarded as a two-round delegated quantum computation protocol in a setting where the client can perform arbitrary Clifford operations, but is limited to a constant-sized quantum memory, given that HE.Dec is in  $\text{NC}^1$ . In that case, the gadgets can be constructed ten qubits at a time, by constructing the sets of five EPR pairs as specified in Sect. 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.

## 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-sized 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 [Spe15], based on the garden-hose model [BFSS13], 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 et al. [YPDF14] 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 [CDN01]. We consider it likely that our new techniques will also be useful in other contexts such as quantum indistinguishability obfuscation [AF16].

**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. This work was supported by the 7th framework EU SIQS and QALGO, and a NWO VIDI grant.

## References

- [ABF+16] Alagic, G., Broadbent, A., Fefferman, B., Gagliardoni, T., Schaffner, C., St. Jules, M.: Computational security of quantum encryption (2016). arXiv preprint [arXiv:1602.01441](https://arxiv.org/abs/1602.01441)
- [ABOE10] Aharonov, D., Ben-Or, M., Eban, E.: Interactive proofs for quantum computations. In: *Proceeding of Innovations in Computer Science (ICS 2010)*, pp. 453–469 (2010)
- [AF16] Alagic, G., Fefferman, B.: On quantum obfuscation (2016). arXiv preprint [arXiv:1602.01771](https://arxiv.org/abs/1602.01771)
- [AJLA+12] Asharov, G., Jain, A., López-Alt, A., Tromer, E., Vaikuntanathan, V., Wichs, D.: Multiparty computation with low communication, computation and interaction via threshold FHE. In: Pointcheval, D., Johansson, T. (eds.) *EUROCRYPT 2012*. LNCS, vol. 7237, pp. 483–501. Springer, Heidelberg (2012)
- [AS06] Arrighi, P., Salvail, L.: Blind quantum computation. *Int. J. Quantum Inf.* **4**(05), 883–898 (2006)
- [Bar89] Barrington, D.A.: Bounded-width polynomial-size branching programs recognize exactly those languages in NC1. *J. Comput. Syst. Sci.* **164**, 150–164 (1989)
- [BB14] Baumeler, Ä., Broadbent, A.: Quantum private information retrieval has linear communication complexity. *J. Cryptol.* **28**(1), 161–175 (2014)
- [BCG+06] Ben-Or, M., Crépeau, C., Gottesman, D., Hassidim, A., Smith, A.: Secure multiparty quantum computation with (only) a strict honest majority. In: *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006)*, pp. 249–260 (2006)
- [BFK09] Broadbent, A., Fitzsimons, J., Kashefi, E.: Universal blind quantum computation. In: *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009*, pp. 517–526. IEEE (2009)
- [BFSS13] Buhrman, H., Fehr, S., Schaffner, C., Speelman, F.: The garden-hose model. In: *Proceedings of the 4th Innovations in Theoretical Computer Science Conference*, pp. 145–158. ACM (2013)

- [BGN05] Boneh, D., Goh, E.-J., Nissim, K.: Evaluating 2-DNF formulas on ciphertexts. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 325–341. Springer, Heidelberg (2005)
- [BGV12] Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (Leveled) fully homomorphic encryption without bootstrapping. In: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, pp. 309–325. ACM (2012)
- [BJ15] Broadbent, A., Jeffery, S.: Quantum homomorphic encryption for circuits of low T-gate complexity. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9216, pp. 609–629. Springer, Heidelberg (2015)
- [Bro15a] Broadbent, A.: Delegating private quantum computations. *Can. J. Phys.* **93**(9), 941–946 (2015)
- [Bro15b] Broadbent, A.: Popescu-Rohrlich correlations imply efficient instantaneous nonlocal quantum computation (2015). arXiv preprint [arXiv:1512.04930](https://arxiv.org/abs/1512.04930)
- [BV11] Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. In: 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS), pp. 97–106, October 2011
- [CDN01] Cramer, R., Damgård, I.B., Nielsen, J.B.: Multiparty computation from threshold homomorphic encryption. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 280–300. Springer, Heidelberg (2001)
- [Chi05] Childs, A.M.: Secure assisted quantum computation. *Quantum Inf. Comput.* **5**(6), 456–466 (2005)
- [CKGS98] Chor, B., Kushilevitz, E., Goldreich, O., Sudan, M.: Private information retrieval. *J. ACM (JACM)* **45**(6), 965–981 (1998)
- [CSWX14] Chiu, W.Y., Szegedy, M., Wang, C., Xu, Y.: The garden hose complexity for the equality function. In: Gu, Q., Hell, P., Yang, B. (eds.) AAIM 2014. LNCS, vol. 8546, pp. 112–123. Springer, Heidelberg (2014)
- [DNS10] Dupuis, F., Nielsen, J.B., Salvail, L.: Secure two-party quantum evaluation of unitaries against specious adversaries. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 685–706. Springer, Heidelberg (2010)
- [FBS+14] Fisher, K.A.G., Broadbent, A., Shalm, L.K., Yan, Z., Lavoie, J., Prevedel, R., Jennewein, T., Resch, K.J.: Quantum computing on encrypted data. *Nat. Commun.* **5** (2014). Article number: 3074
- [Fil12] Fillinger, M.: Lattice based cryptography and fully homomorphic encryption. Master of Logic Project (2012). [http://homepages.cwi.nl/schaffne/courses/reports/MaxFillinger\\_FHE.2012.pdf](http://homepages.cwi.nl/schaffne/courses/reports/MaxFillinger_FHE.2012.pdf)
- [GC99] Gottesman, D., Chuang, I.L.: Quantum teleportation is a universal computational primitive. *Nature* **402**, 390–393 (1999)
- [Gen09] Gentry, C.: Fully homomorphic encryption using ideal lattices. In: STOC, vol. 9, pp. 169–178 (2009)
- [GGH+13] Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. In: 2013 IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS), pp. 40–49. IEEE (2013)
- [GHS15] Gagliardoni, T., Hülsing, A., Schaffner, C.: Semantic security, indistinguishability in the quantum world (2015). arXiv preprint [arXiv:1504.05255](https://arxiv.org/abs/1504.05255)

- [GHV10] Gentry, C., Halevi, S., Vaikuntanathan, V.: A simple BGN-type cryptosystem from LWE. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 506–522. Springer, Heidelberg (2010)
- [GKP+13a] Goldwasser, S., Kalai, Y.T., Popa, R.A., Vaikuntanathan, V., Zeldovich, N.: How to run turing machines on encrypted data. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 536–553. Springer, Heidelberg (2013)
- [GKP+13b] Goldwasser, S., Kalai, Y., Popa, R.A., Vaikuntanathan, V., Zeldovich, N.: Reusable garbled circuits and succinct functional encryption. In: Proceedings of the 45th Annual ACM Symposium on Theory of Computing, STOC 2013, pp. 555–564 (2013)
- [GM84] Goldwasser, S., Micali, S.: Probabilistic encryption. *J. Comput. Syst. Sci.* **28**(2), 270–299 (1984)
- [Got98] Gottesman, D.: Theory of fault-tolerant quantum computation. *Phys. Rev. A* **57**, 127–137 (1998)
- [GVW13] Gorbunov, S., Vaikuntanathan, V., Wee, H.: Attribute-based encryption for circuits. In: Proceedings of the 45th Annual ACM Symposium on Theory of Computing, STOC 2013, pp. 545–554 (2013)
- [IP07] Ishai, Y., Paskin, A.: Evaluating branching programs on encrypted data. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 575–594. Springer, Heidelberg (2007)
- [KO97] Kushilevitz, E., Ostrovsky, R.: Replication is not needed: single database, computationally-private information retrieval. In: FOCS, p. 364. IEEE (1997)
- [KP14] Klauck, H., Podder, S.: New bounds for the garden-hose model. In: 34th International Conference on Foundation of Software Technology and Theoretical Computer Science, pp. 481–492 (2014)
- [Lia13] Liang, M.: Symmetric quantum fully homomorphic encryption with perfect security. *Quantum Inf. Process.* **12**(12), 3675–3687 (2013)
- [Lia15] Liang, M.: Quantum fully homomorphic encryption scheme based on universal quantum circuit. *Quantum Inf. Process.* **14**(8), 2749–2759 (2015)
- [Mar14] Margalit, O.: On the riddle of coding equality function in the garden hose model. In: Information Theory and Applications Workshop (ITA), pp. 1–5. IEEE (2014)
- [NC00] Nielsen, M., Chuang, I.: *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge (2000)
- [OTF15] Ouyang, Y., Tan, S.-H., Fitzsimons, J.: Quantum homomorphic encryption from quantum codes (2015). arXiv preprint [arXiv:1508.00938](https://arxiv.org/abs/1508.00938)
- [Pai99] Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)
- [RAD78] Rivest, R.L., Adleman, L., Dertouzos, M.L.: On data banks, privacy homomorphisms. *Found. Secur. Comput.* **4**(11), 169–180 (1978)
- [RFG12] Rohde, P.P., Fitzsimons, J.F., Gilchrist, A.: Quantum walks with encrypted data. *Phys. Rev. Lett.* **109**(15), 150501 (2012)
- [RSA78] Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* **21**(2), 120–126 (1978)
- [SB08] Shepherd, D., Bremner, M.J.: Instantaneous quantum computation (2008). arXiv preprint [arXiv:0809.0847](https://arxiv.org/abs/0809.0847)

- [Spe11] Speelman, F.: Position-based quantum cryptography, the garden-hose game. Master's thesis, University of Amsterdam. [arXiv:1210.4353](https://arxiv.org/abs/1210.4353)
- [Spe15] Speelman, F.: Instantaneous non-local computation of low T-depth quantum circuits (2015). arXiv preprint [arXiv:1505.02695](https://arxiv.org/abs/1505.02695)
- [SW14] Sahai, A., Waters, B.: How to use indistinguishability obfuscation: deniable encryption, and more. In: Proceedings of the 46th Annual ACM Symposium on Theory of Computing, STOC 2014, pp. 475–484 (2014)
- [SYY99] Sander, T., Young, A., Yung, M.: Non-interactive cryptocomputing for NC1. In: 40th Annual Symposium on Foundations of Computer Science, pp. 554–566. IEEE (1999)
- [TKO+14] Tan, S.-H., Kettlewell, J.A., Ouyang, Y., Chen, L., Fitzsimons, J.F.: A quantum approach to fully homomorphic encryption (2014). arXiv preprint [arXiv:1411.5254](https://arxiv.org/abs/1411.5254)
- [Vai11] Vaikuntanathan, V.: Computing blindfolded: new developments in fully homomorphic encryption. In: 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS), pp. 5–16. IEEE (2011)
- [VDGHV10] van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully homomorphic encryption over the integers. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 24–43. Springer, Heidelberg (2010)
- [VFPR14] Dunjko, V., Fitzsimons, J.F., Portmann, C., Renner, R.: Composable security of delegated quantum computation. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Part II. LNCS, vol. 8874, pp. 406–425. Springer, Heidelberg (2014)
- [YPDF14] Li, Y., Pérez-Delgado, C.A., Fitzsimons, J.F.: Limitations on information-theoretically-secure quantum homomorphic encryption. *Phys. Rev. A* **90**, 050303 (2014)