Quantum Homomorphic Encryption for Polynomial-Sized Circuits

MSc Thesis *(Afstudeerscriptie)*

written by

Yfke Dulek
(born May 14, 1992 in Leiden, the Netherlands)

under the supervision of **dr. Christian Schaffner**, and submitted to the Board of Examiners in partial fulfillment of the requirements for the degree of

MSc in Logic

at the *Universiteit van Amsterdam*.

Date of the public defense:	**Members of the Thesis Committee:**
May 13, 2016	prof. dr. Ronald de Wolf (chair)
	prof. dr. Harry Buhrman
	dr. Serge Fehr
	dr. Christian Schaffner
	Florian Speelman, MSc

Institute for Logic, Language and Computation

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", circuit privacy for the evaluator against passive adversaries, and a three-round scheme for blind delegated quantum computation which puts only very limited demands on the quantum abilities of the client.

Keywords: Homomorphic encryption, quantum cryptography, quantum teleportation, gardenhose model, Barrington's theorem, Clifford group

Acknowledgements

First and foremost, I would like to thank my supervisor Christian Schaffner, who invested an enormous amount of time, effort and patience not only into helping me with this project, but also into involving me in academic life by inviting me to talks and meetings, introducing me to lots of people and allowing me to travel to meet even more. He was dedicated to help me get the most out of the experience, and I am grateful that he is happy to keep guiding me through the next four years.

Of course Florian Speelman, who acted as a second supervisor in a lot of ways, also deserves gratitude. When I got confused, he could often pin down exactly what it was that I did not understand yet, and could explain it to me in layman's terms.

Next, I would like to thank the members of the thesis committee, Ronald de Wolf, Harry Buhrman and Serge Fehr, for taking the time to read through my work and come up with valuable remarks and challenging questions.

A huge thanks also goes out to all my friends and family, who have been there over the past years to support me and celebrate with me, and who have had to listen to a lot of quantum talk over the last couple of months. A special thanks to Ruben for devotedly plowing through a draft version of this thesis, and to my parents for never ever doubting me, not even for a second.

Most of all, I want to thank Camiel, who can somehow magically sense what I need even when I don't really know myself, whether it be celebration, consolation, distraction or just some chocolate.

Contents

LIST OF SYMBOLS AND ABBREVIATIONS

$\mathbb{N}_+$	the positive natural numbers	27
$\mathbb{C}$	the complex numbers	5
$\equiv$	equality operator: $(x \equiv y) = 1$ if and only if x and y are equal	40
$\in_R$	a uniformly randomly selected element	12
$\mathbf{NC}^1$	the class functions computable by a logarithmic-depth Boolean circuit of fan-in 2	20
$\eta(\cdot)$	a negligible function	20
$(n_1 n_2 \cdots n_k)$	a k-cycle (permutation)	27
$[k]$	the set $\{1, ..., k\}$	27
k-PBP	k-permutation branching program	27
$x[i]$	i^{th} element of a bitstring $x \in \{0,1\}^n$, defined for $1 \leq i \leq n$	5
$\|c\|$	absolute value of a number c	5
$\|v\|$	norm of a vector v	5
$\|U\|_1$	trace norm of a matrix	5
$\|\cdot\|_\diamond$	diamond norm	44
$.^\dagger$	complex conjugate of a matrix or vector	5
$\mathrm{Tr}(\cdot)$	trace of a matrix	5
$\mathbb{I}_d/\mathbb{I}$	identity matrix of dimension d	5
$U(d)$	set of unitary matrices of dimension d	5
$\mathbf{0}_{n,m}/\mathbf{0}$	null matrix/vector	5
$\otimes$	tensor product operator (for vectors and matrices)	6
$\|\psi\rangle, \|\varphi\rangle$	pure quantum states	6
ρ, σ	mixed quantum states	7
$\{\|0\rangle, \|1\rangle\}$	the computational basis	6
$\{\|+\rangle, \|-\rangle\}$	the Hadamard basis	6
$\{\|\Phi^+\rangle, \|\Phi^-\rangle, \|\Psi^+\rangle, \|\Psi^-\rangle\}$	EPR pairs, Bell pairs	25
$\rho(X)$	mixed quantum state based on a classical random variable X	8
U_i/U_{ij}	single-/two-qubit gate U applied to quantum register i (and j), identity on all other registers	7
$C(Q)_{ij}$	conditional Pauli gate Q, target wire j, conditioned on wire i	16
$\mathsf{X}, \mathsf{Y}, \mathsf{Z}$	the Pauli group generators	8
$\mathsf{H}, \mathsf{P}, \mathsf{CNOT}$	the Clifford group generators	12
T	the single-qubit T gate (non-Clifford)	7
$\mathcal{P}_n$	the n-qubit Pauli group	9
$\mathcal{P}_n^\pm$	the n-qubit Pauli group based on only $\pm\mathsf{I}, \pm\mathsf{X}, \pm\mathsf{Z}, \pm\mathsf{Y}$	11
$\mathcal{C}_n$	the n-qubit Clifford group	12
$\mathcal{C}_n^\star$	subgroup $\{A \in \mathcal{C}_n \mid \pi_A(\mathsf{X}_n) = \mathsf{X}_n, \pi_A(\mathsf{Z}_n) = \mathsf{Z}_n\}$ of $\mathcal{C}_n$	17

(F)HE	classical (fully) homomorphic encryption	1
Q(F)HE	quantum (fully) homomorphic encyption	22
(Q)HE.KeyGen	the key generation procedure of a (Q)HE scheme	19
(Q)HE.Enc	the encryption procedure of a (Q)HE scheme	20
(Q)HE.Eval	the evaluation procedure of a (Q)HE scheme	20
(Q)HE.Dec	the decryption procedure of a (Q)HE scheme	20
$\mathsf{HE.Rec}_{i\to j}$	the recryption procedure of a classical homomorphic encryption scheme from key set i into key set j	21
κ	security parameter	19
pk	classical public key	19
sk	classical secret key	19
evk	classical evaluation key	19
ρ_{evk}	quantum evaluation key	21
$\widetilde{x}/\widetilde{x}^{[i]}$	the encrypted version of a (classical) string x, with respect to the i^{th} key set	21
CL	a scheme for QHE for Clifford circuits	24
EPR	the entanglement-based quantum homomorphic encryption scheme proposed in [BJ15]	24
AUX	the auxiliary-state-based quantum homomorphic encryption scheme proposed in [BJ15]	24
TP	the teleportation-based QFHE scheme defined in this thesis	2
$\mathsf{TP}^{(\ell)}$	the scheme TP, but with all classical information about the last $L-\ell$ gadgets removed	39
GenGadget	the quantum algorithm that generates a conditional computation gadget	32
UseGadget	the quantum algorithm that applies a conditional computation gadget to an input qubit	32
ComputeKeys	the classical algorithm that computes the Pauli error on the state after use of a conditional computation gadget	32
$g_{f,\mathsf{C}}(x)$	classical information accompanying the conditional computation gadget	32
g_i	shorthand for $g_{\mathsf{HE.Dec},\mathsf{P}^\dagger}(sk_i)$	36
$\gamma_r(g_{f,\mathsf{C}}(x))$	quantum state forming the conditional computation gadget	32
$\Gamma_{pk_{i+1}}(sk_i)$	the T gate gadget for the scheme TP	36
q-IND-CPA	quantum indistinguishability w.r.t. chosen plaintext attacks	23
$\mathscr{A}$	adversary in security games	22
$\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{S}}(\kappa)$	the quantum CPA indistinguishability experiment with respect to the scheme S and adversary $\mathscr{A}$	22
$\mathsf{PubK}^{\mathsf{cpa-mult}}_{\mathscr{A},\mathsf{S}}(\kappa)$	the multiple-message quantum CPA indistinguishability experiment with respect to the scheme S and adversary $\mathscr{A}$	23
$\Xi^{cpa,r}_{\mathsf{S}}$	the encryption action of the challenger in $\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{S}}(\kappa)$, based on randomness r	23

CHAPTER 1

INTRODUCTION

Homomorphic encryption enables the processing of data by operating directly on the encryption of the data, without having to decrypt first. Rivest, Adleman and Dertouzous were the first to observe the possibility of manipulating encrypted data in a meaningful way, rather than just storing and retrieving it [RAD78]. A scheme that would allow the evaluation of *any* function on the encrypted data was considered the holy grail of modern cryptography. After some partial progress [GM84, Pai99, BGN05, IP07] over the years, a breakthrough was made in 2009 when Gentry presented the first fully homomorphic encryption (FHE) scheme [Gen09]. Since then, FHE schemes have been simplified [VDGHV10] and based on fewer and more standard assumptions [BV11]. The exciting developments around FHE have sparked a large amount of research in other areas such as functional encryption [GKP+13b, GVW13, GKP+13a, SW14] and obfuscation [GGH+13].

In the 1980s, a new research area emerged as the possibility of a computational model based on quantum mechanics was proposed [Fey82, Ben82]. Developing quantum computers is a formidable technical challenge, so it currently seems likely that quantum computing will not immediately be available for everyone and that quantum computations will have to be outsourced. Given the importance of classical[1] FHE for "computing in the cloud" [NLV11], it is natural to wonder about the existence of encryption schemes which can encrypt *quantum* data in such a way that a server can carry out arbitrary *quantum* computations on the encrypted data (without interacting with the encrypting party[2]). The best currently known schemes for quantum homomorphic encryption only allow for efficient evaluation of a strict subset of all possible quantum operations [RFG12, Lia13, TKO+14, OTF15, BJ15] and are therefore not fully homomorphic. Generally, operations from the so-called *Clifford group* [Got98] tend to be efficiently executable in a homomorphic setting, whereas the evaluation of operations that fall outside of this group tends to be more demanding. In order to perform arbitrary quantum operations, however, it is necessary to be able to evaluate at least one type of non-Clifford

[1] Here and throughout this thesis, "classical" stands for "non-quantum".

[2] In contrast to *blind* or *delegated quantum computation* where some interaction between client and server is usually required, see Chapter 6 for references.

operation, for example the operation called the T gate. In this thesis, we are concerned with the following question:

Is it possible to construct a computationally secure quantum homomorphic scheme that allows for efficient evaluation of arbitrary polynomial-sized quantum circuits?

Building upon previous work by Broadbent and Jeffery [BJ15], we are able to answer the above question in the affirmative.

1.1 Contributions

In this thesis, we present a new scheme TP (as abbreviation for teleportation) for quantum fully homomorphic encryption which is efficient for circuits containing polynomially many T gates. The scheme is leveled in the sense that an upper bound to the number of T gates needs to be known beforehand. The scheme is secure against chosen plaintext attacks from quantum adversaries, as formalized by the security notion *q-IND-CPA security* described by Broadbent and Jeffery [BJ15].

Like the schemes proposed in [BJ15], our scheme extends a known scheme that is homomorphic only for Clifford operations. In this Clifford scheme, the input quantum state is encrypted with a *quantum one-time pad* (see Section 2.2.1), and the keys to this pad are in turn encrypted using a classical FHE scheme. Any computational assumptions required for that classical scheme are inherited by the Clifford scheme.

We extend the Clifford scheme by supplying auxiliary quantum states to the evaluating party which we call quantum *gadgets* and which aid in the evaluation of the T gates. The size of a gadget in TP depends only on (a certain form of) the space complexity of the decryption function of the classical FHE scheme. This relation turns out to be very convenient, as classical FHE schemes are often optimized with respect to the complexity of the decryption operation. As a concrete example, our scheme can be instantiated with the classical FHE scheme by Brakerski and Vaikuntanathan [BV11], resulting in a quantum scheme where each evaluation gadget consists of a number of quantum bits which is polynomial in the security parameter. This is in sharp contrast to the scheme proposed by Broadbent and Jeffery [BJ15], which requires auxiliary states that grow doubly exponentially in size with respect to the number of layers of T gates in the circuit.

After a T gate is evaluated on a quantum state that is encrypted with a quantum one-time pad, the result might contain an error, depending on the key with which the state was encrypted. Since the evaluator is not allowed to know this key, he does not know whether an error is present on the state and whether it needs to be corrected. Instead, he holds a classical fully homomorphic encryption of the key, which he can use in combination with the gadget in order to correct the error if it was present. In the process, the encrypted quantum state is teleported "through the gadget" [GC99], and the quantum part of the gadget is consumed. This quantum part consists of a number of entangled pairs of quantum bits that are prepared in a way that depends on the secret key of the *classical* FHE scheme.

On a high level, the use of an evaluation gadget corresponds to an *instantaneous non-local quantum computation*[3] of the classical decryption function, where one party holds the

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

secret key for the classical FHE scheme and the other party holds a classical encryption of the key to the encrypted quantum state. Together, this information determines whether a correcting operation needs to be performed on the quantum state or not. Recent results by Speelman [Spe15] show how to perform such computation with a bounded amount of entanglement. These techniques are crucial ingredients for our construction and are the reason why the *garden-hose complexity* [BFSS13] of the decryption procedure of the classical FHE scheme is related to the size of our gadgets.

Apart from the definition of the scheme TP and the proof of its security against chosen plaintext attacks, we describe two variations on TP. The first variation provides circuit privacy to the evaluator in the semi-honest setting, allowing him to hide which circuit he evaluated on the data. The second variation shows how the quantum gadgets can be prepared using only very basic quantum operations, with the help of a (possibly malicious) third party. This variation is useful in a setting where a computationally less powerful client wants to delegate a quantum computation to a more powerful server.

The majority of the content of this thesis is based on a recent article by Dulek, Schaffner and Speelman [DSS16]. Additionally, this thesis contains a more in-depth analysis of the structure of the Clifford group, which plays a central role in the presented scheme because of the way it interacts with the quantum one-time pad. This analysis can be seen as a separate contribution that, although it does not contain any new results, may serve as an introductory resource on the Clifford group.

1.2 Organization of this thesis

A very brief introduction to quantum computation is given in Chapter 2. The chapter also provides a more thorough treatment of two important groups of quantum operations: the Pauli group (and how it can be used for the encryption of quantum data), and the Clifford group (and its relation to a universal set of quantum gates). Chapter 3 defines homomorphic encryption in the classical and quantum setting, and discusses the Clifford scheme and the two extensions presented in [BJ15]. Chapter 4 details the construction of the gadget needed for our quantum homomorphic encryption scheme, laying out the results from complexity theory and instantaneous non-local quantum computation on which this gadget is based. Our leveled quantum fully homomorphic encryption scheme TP is defined and proven secure in Chapter 5. This chapter also offers the adaptations needed to provide circuit privacy and to generate the gadgets using limited quantum power. The thesis is concluded in Chapter 6 with a discussion of the results, and some directions for future research are suggested.

CHAPTER 2

QUANTUM COMPUTATION

We start by defining the basic features of a quantum computer: the structure of the most basic data building block, the *qubit*, and the types of operations that can be performed on this data are introduced in Section 2.1. This section should not be regarded as a complete introduction into quantum computation: for a more thorough treatment, the reader is referred to, for example, [NC00]. In Sections 2.2 and 2.3, we will gradually build up the Clifford group of quantum operations via the structurally simpler Pauli group, and explore some of its properties. In particular, we will consider how the unitaries in the single-qubit Clifford group rotate the Bloch sphere. This helps us to better understand in what way the Clifford operations interact with the Paulis and each other. We will see that a non-Clifford gate (such as the T gate) is needed to be able to perform all quantum operations with arbitrary precision.

2.1 Preliminaries

2.1.1 Notation

We assume familiarity with basic notions in linear algebra and use this subsection to establish some notation. Throughout this thesis, new concepts and their notation will be introduced: for an overview, see the list of symbols and abbreviations at the beginning of this document.

If U is a complex matrix, let $U^\dagger$ denote its complex transpose, and let $\mathrm{Tr}(U)$ denote the trace of the matrix. $\|U\|_1 := \mathrm{Tr}\sqrt{UU^\dagger}$ denotes the trace norm of U. Write $\mathbb{I}_d$ for the identity matrix of dimension d, and $\mathbf{0}_{n,m}$ for the null matrix (or vector, if $m = 1$) of dimension $n \times m$. If the dimensions are clear from the context, we might write $\mathbb{I}$ or $\mathbf{0}$. $U(d)$ denotes the set of all unitary matrices of dimension d.

Let $|c|$ denote the absolute value of a complex number $c \in \mathbb{C}$. If v is a vector of complex numbers, let $\|v\|$ denote its norm, i.e. the square root of the sum of its squared entries.

Finally, if $x \in \{0,1\}^n$, let $x[i]$ denote the i^{th} bit of x, for $1 \leq i \leq n$. We use this notation instead of the more conventional x_i in order avoid confusion with the notation for a list of bit strings $(x_i)_{i=1}^m$ where all $x_i \in \{0,1\}^n$.

2.1.2 Quantum states

Quantum data consists of *quantum bits*, or *qubits*, that can be described by vectors $\begin{pmatrix} \alpha \\ \beta \end{pmatrix} \in \mathbb{C}^2$ (or in any two-dimensional complex Hilbert space) which are unit vectors, i.e. $|a|^2 + |b|^2 = 1$. Defining the *computational basis states*

$$|0\rangle := \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad |1\rangle := \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

(written in *Dirac notation*) allows us to describe a qubit using the notation $\alpha|0\rangle + \beta|1\rangle$: this reflects the idea that the basis states $|0\rangle$ and $|1\rangle$ can be thought of as analogues to the classical bits 0 and 1, and the values α and β can be regarded as the *0-amplitude* and *1-amplitude*, respectively. If both amplitudes are non-zero, then we say that the qubit is in a *superposition* between $|0\rangle$ and $|1\rangle$. Two states that are in equal superposition are

$$|+\rangle := \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \quad \text{and} \quad |-\rangle := \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle.$$

These two states form the *Hadamard basis*. In general, we write $|\psi\rangle$ or $|\varphi\rangle$ to denote a (pure) quantum state of one or more qubits.

When *measuring* a qubit in the computational basis, only two outcomes are possible: $|0\rangle$ is observed with probability $|\alpha|^2$, and $|1\rangle$ with probability $|\beta|^2$. For this reason it is important that $|\alpha|^2 + |\beta|^2 = 1$. After a measurement, the qubit collapses to the observed value. It is also possible to measure a qubit in a different basis, for example in the Hadamard basis $\{|+\rangle, |-\rangle\}$. Since any qubit can be written as a linear combination of $|+\rangle$ and $|-\rangle$, both of these basis state can be assigned an amplitude, whose square equals the probability of observing that outcome.

When combining multiple (uncorrelated) qubits, their joint state is the tensor product of the individual states. For example, the tensor product $|\varphi\rangle_1 \otimes |\psi\rangle_2$ (or $|\varphi\rangle_1|\psi\rangle_2$, or $|\varphi\psi\rangle_{12}$) of two qubits $|\varphi\rangle = \alpha_1|0\rangle + \beta_1|1\rangle$ in register 1 and $|\psi\rangle = \alpha_2|0\rangle + \beta_2|1\rangle$ in register 2 is

$$\begin{pmatrix} \alpha_1 \\ \beta_1 \end{pmatrix} \otimes \begin{pmatrix} \alpha_2 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} \alpha_1\alpha_2 \\ \alpha_1\beta_2 \\ \beta_1\alpha_2 \\ \beta_1\beta_2 \end{pmatrix} = \alpha_1\alpha_2|00\rangle_{12} + \alpha_1\beta_2|01\rangle_{12} + \beta_1\alpha_2|10\rangle_{12} + \beta_1\beta_2|11\rangle_{12}.$$

The subscripts specify to which registers or wires the states belong. Whenever this is clear from the context, we leave out those subscripts.

An n-qubit state thus has dimension 2^n. Not all n-qubit states can be written as a tensor product of their component qubits: states that cannot be separated into such a product are called *entangled* (see Section 4.1).

2.1.3 Quantum operations

We have seen that quantum states can be described by vectors of complex numbers. It should not come as a surprise that operations on quantum states are described by *matrices* applied to those vectors. The result of a quantum operation should of course be a valid quantum state, and for this reason we only allow *norm-preserving* matrices as representatives of quantum operations: a matrix U is norm-preserving if for all v, $\|Uv\| = \|v\|$. Equivalently, a quantum

operation should be a *unitary* matrix, meaning that its complex conjugate is also its inverse: $UU^\dagger = U^\dagger U = \mathbb{I}$. (Note that as a consequence, any such matrix must be square.) An additional property of norm-preserving or unitary matrices that we will use in this work is that their rows all have length 1 and are pairwise orthogonal [Wol01]. The same holds for the columns of the matrix.

Similarly to the subscripts for quantum states, we use the notation U_i to denote a single-qubit quantum operation that is applied to the state in the i^{th} register out of a total of n registers. That is, U_i is the n-qubit unitary $\mathsf{I}^{\otimes(i-1)} \otimes U \otimes \mathsf{I}^{\otimes(n-i)}$, where I is the single-qubit identity operation $\mathbb{I}_2$. Similarly, U_{ij} denotes a two-qubit gate U applied to registers i and j only.

In principle, any unitary matrix in $\mathbb{C}^{2^n \times 2^n}$ is a valid quantum operation on an n-qubit quantum state. However, just like classical operations are ultimately built up from **AND**, **OR** and **NOT** gates, we prefer to think of quantum operations as circuits built up from some small set of basic gates. This reflects the way an actual quantum computer could be constructed, and introduces a measure for the complexity of an operation in terms of the size of the circuit required to perform the operation. Note however that with a finite set of basic gates we can only build countably many different quantum circuits, while the number of unitary matrices over complex numbers is uncountable. The trick is to find a set of basic gates that can approximate any unitary matrix arbitrarily well, i.e. up to some arbitrarily small error.

There are different possible choices for such a set of basic quantum gates [Wol01]. In this thesis, we will work with the gates in the 2-qubit Clifford group (see Section 2.3) extended with the single-qubit gate

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

This set of basic gates can approximate any unitary operation using polylogarithmically many gates in the desired error bound by the Solovay-Kitaev theorem [DN05].

2.1.4 Pure versus mixed states

When working with partial information, we cannot always describe a quantum state completely. From the point of view of an observer, the state may be in a number of possible states $|\psi_i\rangle$, each with probability p_i. Such a *mixed state* may be described with the *density operator*

$$\rho := \sum_i p_i |\psi_i\rangle\langle\psi_i|$$

where $\langle\psi_i|$ denotes the conjugate transpose of the vector $|\psi_i\rangle$. If $|\psi_i\rangle$ is a vector of length 2^n, the density operator will be a $2^n \times 2^n$ matrix. In general, a density operator is a positive-definite matrix ρ such that $\mathrm{Tr}(\rho) = 1$. We will denote mixed states with ρ or σ. Note that this state is not a vector, but a matrix: when a unitary transformation U is applied to a mixed state ρ, the result is $U\rho U^\dagger$, obtained through matrix multiplication. The composition of several mixed states can still be described by the tensor product of the individual states. When measuring a mixed state ρ, the probability of observing basis element $|b\rangle$ is given by $\mathrm{Tr}(\langle b|\rho|b\rangle)$.

It is important to note that a mixture of pure states is not the same as a superposition of those states. Consider, for example, the pure quantum state $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ that is in equal superposition between $|0\rangle$ and $|1\rangle$ on the one hand, and the mixed quantum state $\frac{1}{2}(|0\rangle\langle 0| + |1\rangle\langle 1|)$ that is a full mixture of $|0\rangle$ and $|1\rangle$ on the other hand. Each of these states, when

measured in the computational basis, yields $|0\rangle$ or $|1\rangle$ with equal probability. When measured in the Hadamard basis, however, the pure states yields $|+\rangle$ with certainty, whereas the mixed states yields $|+\rangle$ or $|-\rangle$ with equal probability, since $\frac{1}{2}(|0\rangle\langle 0| + |1\rangle\langle 1|) = \frac{1}{2}(|+\rangle\langle +| + |-\rangle\langle -|)$. This example also shows that a single density operator (in this case $\mathbb{I}_2/2$, the *completely mixed state*) can often be interpreted in multiple ways, as several different mixtures of pure states. This observation lies at the heart of the quantum one-time pad (see Section 2.2.1).

Adopting the notation from [BJ15], we sometimes write $\rho(X)$ to denote the mixed quantum state $\sum_x \Pr_X(x) \cdot |x\rangle\langle x|$ for a classical random variable X.

2.1.5 Visualizing single qubits in the Bloch sphere

When trying to visualize a single-qubit pure state $\alpha|0\rangle+\beta|1\rangle$ with complex amplitudes $\alpha = ae^{\varphi i}$ and $\beta = be^{\psi i}$, we seem to run into the problem of having to visualize four degrees of freedom, since a, b, φ, ψ are all real numbers. However, since the global phase of a qubit is unobservable by measurement [NC00], we may choose not to let our visualization display it. By doing so, we may always assume that $\varphi = 0$ since if not, we can multiply the state globally by $e^{-\varphi i}$ to get the state $a|0\rangle + be^{(\psi-\varphi)i}|1\rangle$ and treat $\psi - \varphi$ as a single variable. Similarly, we may assume that $a \geq 0$. On top of these assumptions on the values of a and φ, the constraint $|a|^2+|b|^2 = 1$ ensures the existence of some $\theta \in (-\frac{\pi}{2}, \frac{\pi}{2}]$ such that $a = \cos\theta$ and $b = \sin\theta$. Hence, we can write any single-qubit pure state as $\cos\theta|0\rangle + \sin\theta e^{\psi i}|1\rangle$, which can be visualized on a two-dimensional space in three dimensions, specifically the surface of the *Bloch sphere* or *Poincaré sphere* [NC00], as follows. Define the x, y, and z coordinates as follows:

$$\begin{aligned} x &:= \sin 2\theta \cos\psi \\ y &:= \sin 2\theta \sin\psi \\ z &:= \cos 2\theta. \end{aligned}$$

Since $|x|^2 + |y|^2 + |z|^2 = 1$, these points all lie on the surface of the unit sphere. Figure 2.1 shows examples of pure states visualized in the Bloch sphere.

The Bloch-sphere representation nicely extends to mixed states as well. Let ρ be some state $\sum_i p_i|\psi_i\rangle\langle\psi_i|$, where each of the pure states $|\psi_i\rangle$ is represented on the Bloch sphere by some vector $\vec{r_i}$ of length 1. Then ρ is represented in the open interior of the Bloch sphere, the *Bloch ball*, by the average vector $\sum_i p_i\vec{r_i}$ [OM08]. The completely mixed state $\mathbb{I}_2/2$ is represented by the vector of length 0, exactly at the point of origin of the Bloch ball. This can be seen by (for example) considering a uniform mixture of the pure states $|0\rangle$ and $|1\rangle$, whose vectors (along the positive and the negative z axis, respectively) cancel each other out when averaged.

2.2 The Pauli group

Apart from the identity gate I, some of the most basic operations one can perform on a single qubit are the *bit flip* X, the *phase flip* Z, and a complex combination Y of both flips:

$$\mathsf{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad \mathsf{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad \mathsf{Y} = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}.$$

X flips a bit in the computational basis by mapping $|0\rangle$ to $|1\rangle$ and vice versa. Z flips the phase of a qubit by assigning a negative amplitude only to the $|1\rangle$ part of the qubit: $\mathsf{Z}|0\rangle = |0\rangle$, while

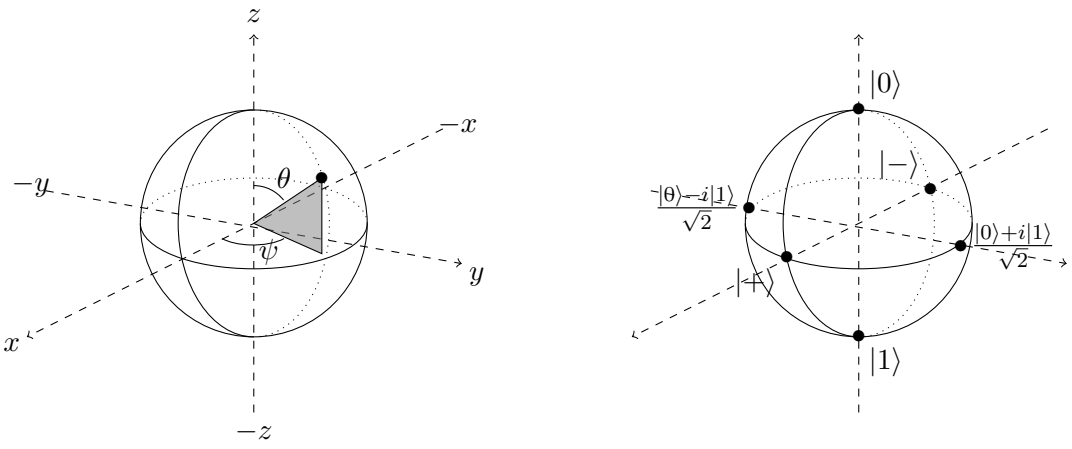


Figure 2.1: Left: the Bloch sphere plus the construction for the vector of an arbitrary single-qubit pure state $\cos\theta|0\rangle + \sin\theta e^{\psi i}|1\rangle$. Right: six examples of common pure states as plotted on the Bloch sphere. Their positions can be verified by checking their individual (θ, ψ) values: $(0,0)$ for $|0\rangle$; $(\frac{\pi}{2}, 0)$ for $|1\rangle$; $(\frac{\pi}{4}, 0)$ for $|+\rangle$; $(\frac{\pi}{4}, \pi)$ for $|-\rangle$; $(\frac{\pi}{4}, \frac{\pi}{2})$ for $\frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$; and $(\frac{\pi}{4}, \frac{3\pi}{2})$ for $\frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle)$.

$\mathsf{Z}|1\rangle = -|1\rangle$. While this phase flip does not seem to make an observable difference because a global phase is not observed in a measurement, it does when working with superpositions of states, for example $\mathsf{Z}|+\rangle = |-\rangle$ and vice versa. One can verify the following useful identities:

$$\begin{aligned} \mathsf{Y} &= i\mathsf{XZ} \\ \mathsf{X}^2 = \mathsf{Y}^2 = \mathsf{Z}^2 &= \mathsf{I} && \text{(self-inverse)} \\ \mathsf{XZ} &= -\mathsf{ZX} && (\mathsf{X} \text{ and } \mathsf{Z} \text{ anti-commute}) \end{aligned}$$

The four operations I, X, Z, and Y are known as the *Pauli matrices*. Together with multiplicative factors ± 1 and $\pm i$, they form a group under multiplication:

Definition 2.1. *The single-qubit* Pauli group $\mathcal{P}_1$ *is the matrix multiplication group over the set*

$$\{\pm\mathsf{I}, \pm i\mathsf{I}, \pm\mathsf{X}, \pm i\mathsf{X}, \pm\mathsf{Z}, \pm i\mathsf{Z}, \pm\mathsf{Y}, \pm i\mathsf{Y}\}.$$

The n-qubit Pauli group $\mathcal{P}_n$ *is the matrix multiplication group over the set*[1]

$$\{Q_{(1)} \otimes \cdots \otimes Q_{(n)} \mid Q_{(i)} \in \mathcal{P}_1\}$$

The Pauli group plays a key role in the stabilizer formalism [NC00] and can be used for basic encryption of quantum states (see Section 2.2.1). We establish some properties of the Pauli group that provide insight into its structure. These properties will mainly be useful for analyzing the Clifford group in the next section.

Lemma 2.2. *The Pauli group* $\mathcal{P}_n$ *is generated by* $\{\mathsf{X}_i, \mathsf{Z}_i, \mathsf{Y}_i \mid 1 \leq i \leq n\}$.

[1]The subscript in $Q_{(i)}$ is simply an index. We use this notation instead of Q_i to in order to distinguish from the application of an operation Q to the i^{th} register.

Proof. For the single-qubit Pauli group $\mathcal{P}_1$, we need to show that it is generated by X, Z and Y. By matrix multiplication, one can check that $\mathsf{I} = \mathsf{XX}$, $i\mathsf{I} = \mathsf{XYZ}$, $-\mathsf{I} = \mathsf{XYXY}$, and $-i\mathsf{I} = \mathsf{XZY}$. All other elements of $\mathcal{P}_1$ are easily generated by combining one of these four identity-like operators with the appropriate generator. For example, $-i\mathsf{X} = (-i\mathsf{I})\mathsf{X}$.

For the general n-qubit Pauli group $\mathcal{P}_n$, clearly it is enough to generate all elements of the set $\{Q_i \mid Q \in \mathcal{P}_1, 1 \leq i \leq n\}$; any other element $Q_{(1)} \otimes \cdots \otimes Q_{(n)}$ is simply the product of $(Q_{(1)})_1$ through $(Q_{(n)})_n$. However, by the same arguments used for the single-qubit Pauli group, any operator Q_i can be generated using only X_i, Z_i and Y_i. □

We now consider how each single-qubit Pauli operation manipulates the points on the Bloch sphere. It turns out that the generators X, Y and Z represent exactly 180° rotations around the x, y and z axes respectively [Gle05]. For example, applying X to some state $\cos\theta|0\rangle + \sin\theta e^{di}|1\rangle$ results in the state $\cos(\frac{\pi}{2} - \theta)|0\rangle + \sin(\frac{\pi}{2} - \theta)e^{-di}|1\rangle$ (up to a global phase): a reflection in the z and y axes, or equivalently a 180° rotation around the x axis. Indeed, the elements that are exactly on the x axis ($|+\rangle$ and $|-\rangle$) are left invariant under the X operation.

Note that using only these 180° rotations, a qubit $|0\rangle$ or $|1\rangle$ can never leave the computational basis, and similarly $|+\rangle$ and $|-\rangle$ will never leave the Hadamard basis. Note furthermore that since the global phase is not visible in the Bloch sphere representation, the rotation around the y axis can be generated by subsequent rotations around the z and x axes, since $\mathsf{Y} = i\mathsf{XZ}$. So if we are not interested in the global phase, we can drop Y as one of the generators of the Pauli group.

The following lemma shows that the only gate that commutes individually with all elements of the Pauli group is the identity gate (up to a global phase).

Lemma 2.3. *The centralizer of the n-qubit Pauli group with respect to the group of all complex-valued unitary $2^n \times 2^n$ matrices $U(2^n)$, defined as*

$$\{\mathsf{U} \in U(2^n) \mid \forall A \in \mathcal{P}_n : \mathsf{U}A = A\mathsf{U}\},$$

is equal to the set $\{c\mathsf{I}^{\otimes n} \mid c \in \mathbb{C}, |c| = 1\}$.

Proof. Trivially, for all $|c| = 1$, $c\mathsf{I}^{\otimes n}$ is a centralizer element. To show that all centralizer elements are of this form, we proceed by induction on n.

For $n = 1$, suppose that $\mathsf{U} \in U(2)$ is an element of the centralizer of $\mathcal{P}_1$. Then in particular, $\mathsf{UZ} = \mathsf{ZU}$, and therefore (writing u_{ij} for the entry in row i and column j of U)

$$\begin{bmatrix} u_{11} & -u_{12} \\ u_{21} & -u_{22} \end{bmatrix} = \mathsf{UZ} = \mathsf{ZU} = \begin{bmatrix} u_{11} & u_{12} \\ -u_{21} & -u_{22} \end{bmatrix}.$$

We observe that $u_{12} = -u_{12}$ and conclude that $u_{12} = 0$. Similarly, $u_{21} = 0$. Using the same technique on the equation $\mathsf{UX} = \mathsf{XU}$, we can deduce that $u_{11} = u_{22}$. Since U is a unitary, and so all its columns must have length 1, U can only be of the form $c\mathsf{I}$ with $|c| = 1$.

For $n > 1$, a similar technique is used. Suppose that $\mathsf{U} \in U(2^n)$ is a centralizer element of $\mathcal{P}_n$. Then for all $Q \in \mathcal{P}_{n-1}$, $\mathsf{U}(\mathsf{Z} \otimes Q) = (\mathsf{Z} \otimes Q)\mathsf{U}$. We shall consider U as a block matrix of four (not necessarily unitary) matrices, namely:

$$\mathsf{U} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}.$$

Then we can deduce, like in the case for $n = 1$, that

$$\begin{bmatrix} AQ & -BQ \\ CQ & -DQ \end{bmatrix} = \mathsf{U}(\mathsf{Z} \otimes Q) = (\mathsf{Z} \otimes Q)\mathsf{U} = \begin{bmatrix} QA & QB \\ -QC & -QD \end{bmatrix}$$

From the above, we see that for all $Q \in \mathcal{P}_{n-1}$, it must be that $-BQ = QB$. In particular, this holds for $Q = \mathsf{I}^{\otimes(n-1)}$, and so we have that $B = -B = \mathbf{0}$. Similarly, we can deduce that $C = \mathbf{0}$. Because U is unitary, its rows and columns must all have norm 1. Since $B = C = \mathbf{0}$, these blocks do not contribute to the norms of the rows and the columns, and so it must be that A and D themselves are unitary. Combining this information with the fact that $QA = AQ$ for all $Q \in \mathcal{P}_{n-1}$, we can conclude that A is a centralizer element of $\mathcal{P}_{n-1}$ with respect to $U(2^{n-1})$, and so (by induction hypothesis) it must be that $A = c\mathsf{I}^{\otimes(n-1)}$ for some $c \in \mathbb{C}$ with $|c| = 1$. Similarly, $D = c'\mathsf{I}^{\otimes(n-1)}$ for some $c' \in \mathbb{C}$ with $|c'| = 1$. Now we know that U is of the form

$$\begin{bmatrix} c\mathsf{I}^{\otimes(n-1)} & \mathbf{0} \\ \mathbf{0} & c'\mathsf{I}^{\otimes(n-1)} \end{bmatrix}$$

It remains to show that $c = c'$. We do so by observing that because $(\mathsf{X} \otimes \mathsf{I}^{\otimes(n-1)})\mathsf{U} = \mathsf{U}(\mathsf{X} \otimes \mathsf{I}^{\otimes(n-1)})$,

$$\begin{bmatrix} \mathbf{0} & c'\mathsf{I}^{\otimes(n-1)} \\ c\mathsf{I}^{\otimes(n-1)} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & c\mathsf{I}^{\otimes(n-1)} \\ c'\mathsf{I}^{\otimes(n-1)} & \mathbf{0} \end{bmatrix}.$$

Hence, $\mathsf{U} = c\mathsf{I}^{\otimes n}$ for some $|c| = 1$, which is what we set out to show. □

So, the identity operator is the only unitary that commutes with all Pauli elements. However, the other Paulis themselves also show interesting commutating behavior with respect to the Pauli group:

Lemma 2.4. *Let $Q \in \mathcal{P}_n$ be such that $Q \notin \{\pm\mathsf{I}^{\otimes n}, \pm i\mathsf{I}^{\otimes n}\}$. Then Q commutes with exactly half of the elements of $\mathcal{P}_n$, and anti-commutes with the other half.*

Proof. We show this by induction on n. For $n = 1$, this can be checked by hand (either by matrix multiplication or by composing rotations of the Bloch sphere): X commutes with I and X, but anti-commutes with Z and Y. Similarly, Z commutes with $\{\mathsf{I}, \mathsf{Z}\}$ but anti-commutes with $\{\mathsf{X}, \mathsf{Y}\}$, and Y commutes with $\{\mathsf{I}, \mathsf{Y}\}$ but anti-commutes with $\{\mathsf{X}, \mathsf{Z}\}$. If some or all of these operators are modified by scalars from $\{\pm 1, \pm i\}$, commutation is not affected.

For $n > 1$, write $Q = Q' \otimes S$ for some $Q' \in \mathcal{P}_{n-1}$ and $S \in \mathcal{P}_1$. By the induction hypothesis, exactly half of the elements of $\mathcal{P}_{n-1}$ commute with Q': call this set $\mathscr{Q}'$. Similarly, by the argument above, exactly half of $\mathcal{P}_1$ commutes with S: call this set $\mathscr{S}$. Then all elements of

$$\mathscr{Q} := \{R' \otimes R \mid R' \in \mathscr{Q}', R \in \mathscr{S}\} \cup \{R' \otimes R \mid R' \in \mathcal{P}_{n-1} - \mathscr{Q}', R \in \mathcal{P}_1 - \mathscr{S}\}$$

commute with Q, since $(Q' \otimes S)(R' \otimes R) = Q'R' \otimes SR$, and the pairs Q', R' and S, R either both commute or both anti-commute. $\mathscr{Q}$ makes up exactly half of the elements of $\mathcal{P}_n$. □

Note that the proof of Lemma 2.4 also shows that any n-qubit Pauli operator $Q \notin \{\pm\mathsf{I}^{\otimes n}, \pm i\mathsf{I}^{\otimes n}\}$ commutes with exactly half of the Paulis in the set $\mathcal{P}_n^{\pm} := \{Q_{(1)} \otimes \cdots \otimes Q_{(n)} \mid Q_{(i)} \in \{\pm\mathsf{I}, \pm\mathsf{X}, \pm\mathsf{Y}, \pm\mathsf{Z}\}\}$.

2.2.1 Quantum one-time pad

Using only Pauli operators, we can build a simple yet effective encryption scheme known as the *quantum one-time pad* [AMTW00]: if a uniformly random Pauli operator is applied to some qubit, the resulting state is fully mixed to anyone that does not have knowledge of which Pauli is applied.

Analogous to the classical one-time pad, where a plaintext bit b is hidden by choosing some $b' \in_R \{0,1\}$ and computing $b \oplus b'$ for the ciphertext, the quantum one-time pad is applied to some single-qubit mixed state ρ by picking *two* random bits $a, b \in_R \{0,1\}$ uniformly at random and computing the state $(\mathsf{X}^a\mathsf{Z}^b)\rho(\mathsf{X}^a\mathsf{Z}^b)^\dagger$ to be the ciphertext. As long as a, b remain secret, this random Pauli operator completely hides the content of any $\rho = \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}$, since

$$\begin{aligned}\sum_{a,b\in 0,1} \frac{1}{4}(\mathsf{X}^a\mathsf{Z}^b)\rho(\mathsf{X}^a\mathsf{Z}^b)^\dagger &= \frac{1}{4}\left(\rho + \mathsf{X}\rho\mathsf{X}^\dagger + \mathsf{Z}\rho\mathsf{Z}^\dagger + (\mathsf{XZ})\rho(\mathsf{XZ})^\dagger\right) \\ &= \frac{1}{4}\left(\begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix} + \begin{bmatrix} \delta & \gamma \\ \beta & \alpha \end{bmatrix} + \begin{bmatrix} \alpha & -\beta \\ -\gamma & \delta \end{bmatrix} + \begin{bmatrix} \delta & -\gamma \\ -\beta & \alpha \end{bmatrix}\right) \\ &= \frac{1}{2}\begin{bmatrix} \alpha+\delta & 0 \\ 0 & \alpha+\delta \end{bmatrix}\end{aligned}$$

which, because $\alpha + \delta = \mathrm{Tr}(\rho) = 1$, equals $\mathbb{I}_d/2$, the completely mixed state.

We can also see that the quantum one-time pad results in the completely mixed state by considering what it does to a (pure or mixed) state represented in the Bloch ball by coordinates (x, y, z). Then X represents a rotation around the x axis, effectively mapping the state to $(x, -y, -z)$. Similarly, Z maps the state to $(-x, -y, z)$, and XZ (which is equal to Y up to a global phase) maps the state to $(-x, y, -z)$. Averaging over these four vectors results in a vector with entries (0,0,0), which represents the completely mixed state.

2.3 The Clifford group

In this section we consider a somewhat larger group of operations known as the Clifford group. It is defined as the set of operations that commute with the Pauli group as a whole:

Definition 2.5. *The n-qubit* Clifford group $\mathcal{C}_n$ *is the normalizer of $\mathcal{P}_n$ with respect to $U(2^n)$. That is,*

$$\mathcal{C}_n := \{\mathsf{U} \in U(2^n) \mid \mathcal{P}_n\mathsf{U} = \mathsf{U}\mathcal{P}_n\}.$$

Clifford operators commute with the entire Pauli group in the sense that left multiplication of the Pauli group results in the same set of operators as right multiplication. This is a more relaxed notion than the centralizer of the Pauli group, since the Clifford operators do not need to commute individually with every Pauli operator. For U to be a Clifford operation, we only require that for every $Q \in \mathcal{P}_n$, there exists *some* $Q' \in \mathcal{P}_n$ such that $Q\mathsf{U} = \mathsf{U}Q'$.

What does the set $\mathcal{C}_n$ look like? One can verify by hand that at least the following operations are included in the single- and two-qubit Clifford groups:

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

The size of $\mathcal{C}_n$ grows quickly with n, but as we will see, the set always remains finite (up to a global phase). We can associate the Clifford elements with permutations on the Paulis in the following way:

Lemma 2.6. *For any $A \in \mathcal{C}_n$, let $\pi_A : \mathcal{P}_n \to \mathcal{P}_n$ be the function defined by $\pi_A(Q) := AQA^\dagger$. π_A is a permutation on $\mathcal{P}_n$. Furthermore, π_A is a linear map that preserves the group structure of $\mathcal{P}_n$.*

Proof. We need to show that π_A is a bijection on $\mathcal{P}_n$. To see that π_A is injective, suppose for $Q, Q' \in \mathcal{P}_n$ that $AQA^\dagger = AQ'A^\dagger$. It follows that $Q = A^\dagger AQA^\dagger A = A^\dagger AQ'A^\dagger A = Q'$. Since the domain and codomain of π_A are finite and of equal size, injectivity immediately implies bijectivity. Hence, π_A is a permutation on $\mathcal{P}_n$.

To verify that π_A preserves addition, simply observe that $A(Q + Q')A^\dagger = AQA^\dagger + AQ'A^\dagger$. Similarly for scalar multiplication: $AcQA^\dagger = cAQA^\dagger$, so π_A is a linear map. π_A preserves the Pauli group structure since $AQQ'A^\dagger = AQA^\dagger AQ'A^\dagger$. □

Because π_A is a permutation for any $A \in \mathcal{C}_n$, we can view every element of the Clifford group as a permutation of the Paulis. It is natural to ask whether (1) different Clifford operations represent different permutations on $\mathcal{P}_n$, and (2) whether all permutations on $\mathcal{P}_n$ are represented by at least one Clifford operation. In the rest of this section, we will first show that all Cliffords do represent unique permutations by showing that two Cliffords that induce the same permutation cannot differ by more than a global phase. Then we will see that only a specific set of permutations is represented by the Clifford group, and that for the single-qubit case this set corresponds to the rotational symmetries of a cube.

Lemma 2.7. *Let $A, B \in \mathcal{C}_n$. If $\pi_A = \pi_B$, then $A = cB$ for some $c \in \mathbb{C}$ with modulus 1.*

Proof. Suppose that for all $P \in \mathcal{P}_n$, $APA^\dagger = BPB^\dagger$. Then for all $P \in \mathcal{P}_n$, $B^\dagger APA^\dagger B = P$, and so $B^\dagger A$ is in the centralizer of the n-qubit Pauli group. From Lemma 2.3, it follows that $B^\dagger A = c\mathsf{I}^{\otimes n}$ for some $c \in \mathbb{C}$ with $|c| = 1$, and so $A = cB$. □

Next, we consider the second question: which permutations on $\mathcal{P}_n$ are represented by the Cliffords? First, note that because π_A preserves the group structure of $\mathcal{C}_n$ (see Lemma 2.6), the way π_A acts on the generators of $\mathcal{P}_n$ completely determines the values of π_A on the rest of the group. For example for a single-qubit Clifford A, $\pi_A(\mathsf{Y}) = i\pi_A(\mathsf{X})\pi_A(\mathsf{Z})$, because $\mathsf{Y} = i\mathsf{XZ}$. The following lemma helps us to reduce the number of permutations we need to consider.

Lemma 2.8. *Let $A \in \mathcal{C}_n$. Then the following restrictions on the values of $\pi_A(\mathsf{X}_n)$ and $\pi_A(\mathsf{Z}_n)$ hold:*

(i) $\pi_A(\mathsf{X}_n), \pi_A(\mathsf{Z}_n) \in \mathcal{P}_n^\pm \backslash \{\pm\mathsf{I}^{\otimes n}\}$.

(ii) $\pi_A(\mathsf{X}_n)$ *and* $\pi_A(\mathsf{Z}_n)$ *anti-commute.*

Proof.

(i) We prove that $\pi_A(\mathsf{X}_n) \in \mathcal{P}_n^\pm \backslash \{\pm\mathsf{I}^{\otimes n}\}$ – the argument for $\pi_A(\mathsf{Z}_n)$ is identical. First, note that for all $c \in \{1, -1, i, -i\}$, $\pi_A(c\mathsf{I}^{\otimes n}) = c\mathsf{I}^{\otimes n}$, so all identity-like matrices are mapped to themselves. Therefore, X_n cannot be mapped to any of them by the permutation π_A.

Next, suppose toward a contradiction that $\pi_A(\mathsf{X}_n) = \pm iQ$ for some $Q \in \mathcal{P}_n^{\pm}$. This would result in

$$\mathsf{I}^{\otimes n} = \pi_A(\mathsf{I}^{\otimes n}) = \pi_A(\mathsf{X}_n\mathsf{X}_n) = \pi_A(\mathsf{X}_n)^2 = (\pm iQ)^2 = (\pm i)^2\mathsf{I}^{\otimes n} = -\mathsf{I}^{\otimes n},$$

a contradiction. Hence, the only possible images for X_n under π_A are the elements from $\mathcal{P}_n^{\pm}\backslash\{\pm\mathsf{I}^{\otimes n}\}$.

(ii) This holds because X_n and Z_n anti-commute, and π_A preserves the group structure (see Lemma 2.6):

$$\pi_A(\mathsf{X}_n)\pi_A(\mathsf{Z}_n) = \pi_A(\mathsf{X}_n\mathsf{Z}_n) = \pi_A(-\mathsf{Z}_n\mathsf{X}_n) = -\pi_A(\mathsf{Z}_n\mathsf{X}_n) = -\pi_A(\mathsf{Z}_n)\pi_A(\mathsf{X}_n).$$

□

It can even be verified that *any* permutation that satisfies the restrictions in Lemma 2.8 determines a (unique) Clifford operator $A \in \mathcal{C}_n$ [Ozo08]. We do so in Section 2.3.2.

2.3.1 The single-qubit Clifford group $\mathcal{C}_1$

We first focus on the single-qubit Clifford group $\mathcal{C}_1$: it is relatively small and we can therefore study its structure explicitly. Combining Lemmas 2.7 and 2.8, we see that the single-qubit Clifford group $\mathcal{C}_1$ contains (at most) $6 \times 4 = 24$ unique elements, up to scalar multiplication. This is because there are six possible values for $\pi_A(\mathsf{X})$ (namely, the elements of $\{\pm\mathsf{X}, \pm\mathsf{Y}, \pm\mathsf{Z}\}$), and four remaining possibilities for $\pi_A(\mathsf{Z})$ (the half of $\mathcal{P}_1^{\pm}$ that anti-commutes with $\pi_A(\mathsf{X})$). We will show that the Clifford group does indeed contain all the operators that induce these 24 possible permutations.

Lemma 2.9. *Suppose for some $A \in \mathcal{C}_1$ that $\pi_A(\mathsf{X}) = Q$, and $\pi_A(\mathsf{Z}) = Q'$. Then*

(i) $\pi_{A\mathsf{X}}(\mathsf{X}) = Q$ *and* $\pi_{A\mathsf{X}}(\mathsf{Z}) = -Q'$.

(ii) $\pi_{A\mathsf{Z}}(\mathsf{X}) = -Q$ *and* $\pi_{A\mathsf{Z}}(\mathsf{Z}) = Q'$.

(iii) $\pi_{A\mathsf{Y}}(\mathsf{X}) = -Q$ *and* $\pi_{A\mathsf{Y}}(\mathsf{Z}) = -Q'$.

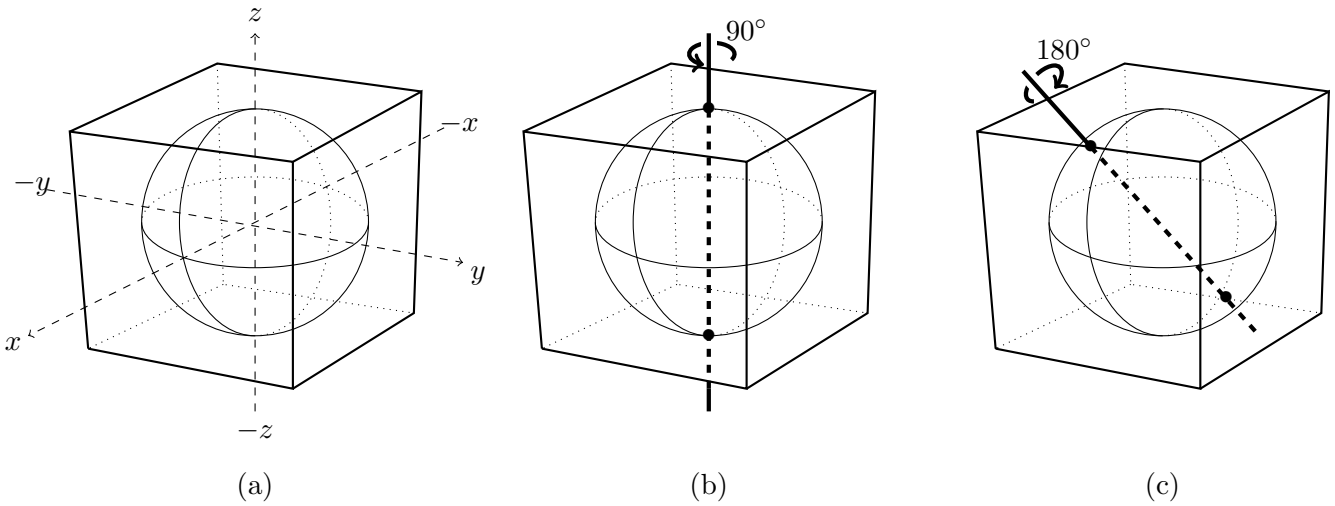
Proof. These identities can be verified by straightforward computation. For example, $\pi_{A\mathsf{X}}(\mathsf{Z}) = A\mathsf{X}\mathsf{Z}\mathsf{X}A^\dagger = \pi_A(\mathsf{X}\mathsf{Z}\mathsf{X}) = -\pi_A(\mathsf{X}\mathsf{X}\mathsf{Z}) = -\pi_A(\mathsf{Z})$. □

Table 2.2 explicitly lists the six Clifford operators that represent the permutations $\pi_A(\mathsf{X}) \in \{\mathsf{X}, \mathsf{Y}, \mathsf{Z}\}$ and $\pi_A(\mathsf{Z}) \in \{\mathsf{X}, \mathsf{Y}, \mathsf{Z}\}\backslash\{\pi_A(\mathsf{X})\}$. We can find all 24 permutations by right-multiplying these operators with the Paulis as described in Lemma 2.9.

Like Pauli operators, each of the resulting 24 elements of the single-qubit Clifford group corresponds to a rotation of the Bloch sphere. These rotations are determined by the permutations listed in Table 2.2. For example, the P operator maps the x axis onto the y axis, maps the y axis onto the $-x$ axis (i.e. the mirrored x axis), and leaves the z axis fixed. This corresponds to the values $\pi_{\mathsf{P}}(\mathsf{X}) = \mathsf{Y}$, $\pi_{\mathsf{P}}(\mathsf{Y}) = -\mathsf{X}$, and $\pi_{\mathsf{P}}(\mathsf{Z}) = \mathsf{Z}$ from Table 2.2, and effectively rotates the Bloch sphere by 45° around the z axis. We can check that the permutations of the axes that arise from the Clifford group elements are exactly those that are orientation-preserving on the Bloch sphere, i.e. realizable when physically rotating the sphere in three dimensions.

A	$\pi_A(\mathsf{X})$	$\pi_A(\mathsf{Z})$
PHP	X	Y
I	X	Z
HPZ	Y	X
P	Y	Z
H	Z	X
PH	Z	Y

Figure 2.2: Six single-qubit Clifford operations and the permutations they induce on the Pauli group, as fixed by the permutations of the generators X and Z. From these six Cliffords, the entire group $\mathcal{C}_1$ can be generated by right-multiplying with Paulis.



(a) (b) (c)

Figure 2.3: (a): the isometry between the Bloch sphere under Clifford rotations and the orientation-preserving rotation group of the cube. (b) and (c): the generators of the Clifford group (P and H, respectively) as corresponding to the generators of the cube rotation group.

In order to establish the generators of $\mathcal{C}_1$, we note that the single-qubit Clifford group also corresponds to the (orientation-preserving) symmetry group of the cube in the following way. Draw the smallest possible cube around the Bloch sphere so that the axes of the sphere run through the centers of the six cube faces (see Figure 2.3(a)). Then every order-preserving rotation of the cube corresponds to a Clifford operation, and vice versa (see Table 2.2). Specifically, Figures 2.3(b) and 2.3(c) show the rotations induced by P and H, respectively. These two rotations are known generate the entire orientation-preserving symmetry group of the cube [Arm88], so P and H together generate the entire single-qubit Clifford group (up to global phase). This isometry also helps to see why there are 24 elements in $\mathcal{C}_1$: the cube face corresponding to the (positive) x-axis can be moved to any of the six cube faces, and then rotated around its center in four possible ways, representing the remaining possible permutations of the z-axis.

2.3.2 The multi-qubit Clifford group $\mathcal{C}_n$

The general n-qubit Clifford group becomes very large very fast. The goal of this subsection is to establish the size of $\mathcal{C}_n$, and along the way we will see that $\mathcal{C}_n$ is generated by just H, P and CNOT. We start by verifying that *any* permutation on the Pauli group that preserves its structure as in Lemma 2.8 can be associated with a Clifford operator:

Lemma 2.10. *Let $f : \mathcal{P}_n \to \mathcal{P}_n$ be a permutation such that $f(\mathsf{X}_n), f(\mathsf{Z}_n) \in \mathcal{P}_n^{\pm} \backslash \{\pm \mathsf{I}\}$, and $f(\mathsf{X}_n)$ anti-commutes with $f(\mathsf{Z}_n)$. Then there exists an $A \in \mathcal{C}_n$ such that $f = \pi_A$. Moreover, A is a product of only H, P and CNOT operations.*

Proof. This is again a proof by induction on n. For $n = 1$, the statement was verified by hand in Section 2.3.1.

For $n > 1$, we will build up the Clifford operation $A^\dagger$ in several steps, and show that $\pi_{A^\dagger} \circ f$ is the identity permutation. It then immediately follows that $f = \pi_A$.

First, let $R \in \mathcal{P}_n$ be the result of $f(\mathsf{X}_n)$. Then $R = R_{(1)} \otimes \cdots \otimes R_{(n-1)} \otimes R_{(n)}$ for $R_{(i)} \in \mathcal{P}_1$. As we have seen in Section 2.3.1, there exists a single-qubit Clifford B such that $\pi_B(R_{(n)}) = \mathsf{X}$, and hence, $\pi_{\mathsf{I}^{\otimes(n-1)} B}(R) = R_{(1)} \otimes \cdots \otimes R_{(n-1)} \otimes \mathsf{X}$. This B can be constructed from just H and P, like all single-qubit Cliffords.

For any single-qubit Pauli Q, define $C(Q)_{ij}$ to be the controlled version of Q with control bit i and target bit j. One can verify that for any Pauli Q,

$$\pi_{C(Q)_{ij}}(\mathsf{X}_i \otimes \mathsf{I}_j) = (\mathsf{X}_i \otimes Q_j) \qquad \text{and} \qquad \pi_{C(Q)_{ij}}(\mathsf{Z}_i \otimes \mathsf{I}_j) = (\mathsf{Z}_i \otimes \mathsf{I}_j).$$

Moreover, such controlled-Q gates are Cliffords and can be easily built using only CNOT, H and single-qubit Pauli operations (which, in turn, can be built from H and P) [NC00]. Let $C_{\mathsf{X}} := \prod_{i=1}^{n-1} C(R_{(i)})_{ni}$. Then clearly, $\pi_{C(R_{(i)})_{ni}}(\mathsf{X}_n) = \pi_{\mathsf{I}^{\otimes(n-1)} B}(f(\mathsf{X}_n))$. Thus, $\pi_{(C(R_{(i)})_{ni})^{-1}} \circ \pi_{\mathsf{I}^{\otimes(n-1)} B} \circ f$ leaves X_n fixed.

Using roughly the same technique, we can find a Clifford that also inverts the effect of f on Z_n, while leaving X_n fixed: let $S_{(1)} \otimes \cdots \otimes S_{(n-1)} \otimes S_{(n)} \in \mathcal{P}_n$ be the result of $\pi_{(C(R_{(i)})_{ni})^{-1}}(\pi_{\mathsf{I}^{\otimes(n-1)} B}(f(\mathsf{Z}_n)))$. Again, there exists a single-qubit Clifford B' such that $\pi_{B'}$ (leaves X fixed and) maps $S_{(n)} \mapsto \mathsf{Z}$. Then, defining $C_{\mathsf{Z}} := \prod_{i=1}^{n-1} C(S_{(i)})_{ni}$, we can check that $\mathsf{H} C_{\mathsf{Z}} \mathsf{H}$ maps $\mathsf{Z}_n \mapsto S_{(1)} \otimes \cdots \otimes S_{(n-1)} \otimes \mathsf{Z}$, while leaving X_n fixed.

From the above, we conclude that the permutation induced by

$$A' := \mathsf{H}(C_{\mathsf{Z}})^{-1} H(\mathsf{I}^{\otimes(n-1)} \otimes B')(C_{\mathsf{X}})^{-1}(\mathsf{I}^{\otimes(n-1)} \otimes B),$$

which can be built entirely from H, P and CNOT, inverts the permutation f simultaneously on the Paulis X_n and Z_n. All other Pauli generators may still be arbitrarily permuted by this permutation, but at least the group structure is respected by the permutation $\pi_{A'} \circ f$ because of Lemma 2.6. Hence, by the induction hypothesis, there exists some Clifford $D \in \mathcal{C}_{n-1}$, a product of H, P and CNOT gates, such that $\pi_{D \otimes \mathsf{I}} \circ \pi_{A'} \circ f$ is the identity permutation. Defining $A := (A')^\dagger (D \otimes \mathsf{I})^\dagger$ completes the proof. □

As a corollary of Lemmas 2.6, 2.7 and 2.10, all Cliffords can be generated by H, P and CNOT. Lemma 2.6 states that all Clifford operations A give rise to a permutation π_A on the Paulis that is group-structure preserving. From Lemma 2.6 we know that all such permutations can be induced by a unitary that is constructed from the set $\{\mathsf{H}, \mathsf{P}, \mathsf{CNOT}\}$. By Lemma 2.7, this unitary that induces the same permutation as A is equal to A (up to a global phase).

We now construct a recursive formula for the size of the n-qubit Clifford group. To do this, we show that a certain subgroup of $\mathcal{C}_n$ is isomorphic to $\mathcal{C}_{n-1}$, and then count the number of cosets of this subgroup in $\mathcal{C}_n$.

Lemma 2.11. *Let $\mathcal{C}_n^\star := \{A \in \mathcal{C}_n \mid \pi_A(\mathsf{X}_n) = \mathsf{X}_n, \pi_A(\mathsf{Z}_n) = \mathsf{Z}_n\}$ be the set of n-qubit Cliffords that leave the generators X_n and Z_n fixed. Then $\mathcal{C}_n^\star$ is isomorphic to $\mathcal{C}_{n-1}$.*

Proof. Define $f : \mathcal{C}_{n-1} \to \mathcal{C}_n^\star$ by $f(A) := A \otimes \mathsf{I}$. Clearly, this is a homomorphism because $f(AB) = AB \otimes \mathsf{I} = (A \otimes \mathsf{I})(B \otimes \mathsf{I}) = f(A)f(B)$. To see that it is an isomorphism, we need to show that it is a bijection as well.

Injectivity of f follows immediately from the observation that if $f(A) = f(B)$, then $A \otimes \mathsf{I} = B \otimes \mathsf{I}$, and hence $A = B$.

For surjectivity, let $A \in \mathcal{C}_n^\star$. Then by definition of $\mathcal{C}_n^\star$, $A\mathsf{X}_n = \mathsf{X}_n A$ and $A\mathsf{Z}_n = \mathsf{X}_n A$. Like in the proof of Lemma 2.3, we can deduce from these equations that A must be of the form $A' \otimes \mathsf{I}$ for some $A' \in U(2^{n-1})$. Because $A \in \mathcal{C}_n$, it must be the case that $A' \in \mathcal{C}_{n-1}$. Hence, there exists an $A' \in \mathcal{C}_{n-1}$ such that $f(A') = A$. □

Lemma 2.12. *The size of the n-qubit Clifford group is $|\mathcal{C}_n| = 2(4^n - 1)4^n|\mathcal{C}_{n-1}|$.*

Proof. For every $A \in \mathcal{C}_n$, $A\mathcal{C}_n^\star$ is a (left) coset of $\mathcal{C}_n^\star$ in $\mathcal{C}_n$. We argue that if two n-qubit Cliffords A and B give rise to permutations that act identically on $\{\mathsf{X}_n, \mathsf{Z}_n\}$, then $A\mathcal{C}_n^\star = B\mathcal{C}_n^\star$. Because $\pi_A(\mathsf{X}_n) = \pi_B(\mathsf{X}_n)$, we have that $\pi_{B^\dagger A}(\mathsf{X}_n) = \mathsf{X}_n$, and similarly that $\pi_{B^\dagger A}(\mathsf{Z}_n) = \mathsf{Z}_n$. By the same argument as in the surjectivity proof of Lemma 2.11, it follows that there exists a $C \in \mathcal{C}_{n-1}$ such that $B^\dagger A = \mathsf{I} \otimes C$. For this C, we have that $B(\mathsf{I} \otimes C) = B(B^\dagger A) = A = A(\mathsf{I} \otimes \mathsf{I}^{\otimes(n-1)})$, and hence $B\mathcal{C}_n^\star \cap A\mathcal{C}_n^\star \neq \emptyset$. Because the (left) cosets of a group always form a partition of that group, this means that $B\mathcal{C}_n^\star = A\mathcal{C}_n^\star$.

From the above argument, we see that every structure-preserving permutation on the Pauli group can be associated with exactly one coset of $\mathcal{C}_{n-1} \cong \mathcal{C}_n^\star$ in $\mathcal{C}_n$. From Lemmas 2.4 and 2.8, it follows that the number of possible images for X_n and Z_n under such a permutation is

$$|\mathcal{P}_n^\pm - \{\pm\mathsf{I}^{\otimes n}\}| \cdot \frac{1}{2}|\mathcal{P}_n^\pm| = (2 \cdot 4^n - 2)\, 4^n = 2(4^n - 1)4^n.$$

The result of the lemma now follows from Lagrange's Theorem. □

From Lemma 2.12, we can derive the closed formula $|\mathcal{C}_n| = 2^{n^2+2n} \prod_{j=1}^n (4^j - 1)$. The size of $\mathcal{C}_n$ thus grows exponentially in n, but always remains finite if we ignore the global phase.

2.4 Universal quantum computation

For universal quantum computation, we want to be able to perform arbitrary quantum operations, or at least approximate them up to some arbitrarily small error. In the single-qubit case, this means approximating arbitrary rotations of the Bloch sphere, or equivalently mapping the $|0\rangle$ state to any point on the Bloch sphere. From the analysis in Section 2.3, it is clear that the Clifford group is not sufficient for this task.

For universal quantum computation, it suffices to add a single-qubit non-Clifford gate, for example the T gate [NC00]. The T gate represents a 45° rotation around the z axis of the Bloch sphere. Any rotation of the Bloch sphere can be approximated by a finite product of this T rotation and the H rotation [BMP+99]. The P gate becomes redundant since it can

be constructed with two consecutive T gates. The Solovay-Kitaev theorem even states that the number of operators needed for this approximation is quite small: polylogarithmic in the desired error bound [NC00]. If we want to do universal quantum computation on n qubits, the operations H, T and CNOT are enough. This set of basic gates is often referred to as the *standard set* or the *Clifford+*T set, because the Cliffords can be generated using only T^2, H, and CNOT (see Section 2.3.2).

We conclude this chapter by stating a number of useful gate equalities (up to a global phase), which can be verified by hand, either by matrix multiplication or by composing the rotations on the Bloch sphere:

$$\begin{aligned} \mathsf{T}^2 &= \mathsf{P} \\ \mathsf{P}^2 &= \mathsf{Z} \\ \mathsf{HXH} &= \mathsf{Z} \\ \mathsf{PZ} &= \mathsf{ZP} \\ \mathsf{PX} &= \mathsf{XZP} \\ \mathsf{TX} &= \mathsf{PXT} \\ \mathsf{TZ} &= \mathsf{ZT} \end{aligned}$$

and some equalties that cannot be visualized in three dimensions:

$$\begin{aligned} \mathsf{CNOT}(\mathsf{I} \otimes \mathsf{X}) &= (\mathsf{I} \otimes \mathsf{X})\mathsf{CNOT} \\ \mathsf{CNOT}(\mathsf{X} \otimes \mathsf{I}) &= (\mathsf{X} \otimes \mathsf{X})\mathsf{CNOT} \\ \mathsf{CNOT}(\mathsf{I} \otimes \mathsf{Z}) &= (\mathsf{Z} \otimes \mathsf{Z})\mathsf{CNOT} \\ \mathsf{CNOT}(\mathsf{Z} \otimes \mathsf{I}) &= (\mathsf{Z} \otimes \mathsf{I})\mathsf{CNOT} \end{aligned}$$

CHAPTER 3

HOMOMORPHIC ENCRYPTION

Usually, encrypted data can only be stored and later retrieved, and in order to manipulate the data one would have to decrypt it first. Homomorphic encryption allows the evaluation of a function on the plaintext by manipulating the ciphertext in some related way, using an evaluation key. By definition, this means that the encryption is malleable by anyone who knows the evaluation key, which can be an undesirable property in some contexts. The homomorphic property is an advantage in other contexts, for example when outsourcing computations to an untrusted party. This advantage is especially evident in a quantum setting, where a possible future scenario is the existence of a limited number of powerful quantum computers that carry out quantum computational tasks on data from clients.

This chapter provides formal 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 a public-key setting. For a more thorough treatment of these concepts, and how they can be transferred to the symmetric-key setting, see [BJ15].

3.1 Classical homomorphic encryption

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 $x_1, \ldots, x_\ell$, and send the resulting ciphertexts $c_1, \ldots, c_\ell$ to an evaluator who evaluates some circuit C on them using the evaluation key. The evaluator sends the result to a party that possesses the secret key, who should be able to decrypt it to $\mathsf{C}(x_1, \ldots, x_\ell)$.

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

$(pk, evk, sk) \leftarrow \mathsf{HE.KeyGen}(1^\kappa)$ where κ 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. The length of the keys is a fixed polynomial in κ. Longer keys generally increase the input size of the computationally hard problem that a potential attacker needs to solve, essentially making the encryption more secure.

$c \leftarrow \mathsf{HE.Enc}_{pk}(x)$ for some message x from a message space M_κ (in this thesis, we will work with message space $M_\kappa = \{0,1\}$). This probabilistic procedure outputs a ciphertext c, using the public key pk.

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

$x' \leftarrow \mathsf{HE.Dec}_{sk}(c')$ outputs a message $x' \in M_\kappa$, using the secret key sk when presented with the output of the evaluation function.

In principle, $\mathsf{HE.Enc}_{pk}$ can only encrypt single bits. When encrypting an n-bit message $x \in \{0,1\}^n$, we encrypt the message bit-by-bit, applying the encryption procedure n times. We sometimes abuse the notation $\mathsf{HE.Enc}_{pk}(x)$ to denote this bitwise encryption of the string x.

For HE to be a homomorphic encryption scheme, we require correctness in the following sense:

Definition 3.1. *A scheme is* correct *for a set of circuits $\mathcal{S}$ if for any circuit $\mathsf{C} \in \mathcal{S}$, there exists a negligible*[1] *function η such that, for any input x,*

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

Strictly speaking, the definition of a homomorphic encryption scheme allows part of the evaluation work to be deferred to the decryption phase, resulting in a trivial scheme where the evaluator only appends instructions for the circuit C, and the decrypting party applies the described circuit to x directly after decrypting it. To discourage the design of such schemes, it is often required that the scheme is compact, meaning that the complexity of the decryption function should not depend on the size of the circuit:

Definition 3.2. *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 $\mathsf{HE.Dec}$ to the result of $\mathsf{HE.Eval}^{\mathsf{C}}(c)$ is at most $p(\kappa)$.*

In practice, all proposed FHE schemes aim to make the decryption function as simple as possible, and usually provide decryption in $\mathbf{NC}^1$, the class of logarithmic-depth Boolean circuits of fan-in 2. This class is contained in the class of log-space computable functions.

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].

Early homomorphic encryption schemes were only somewhat homomorphic: they usually only implemented a single operation (addition or multiplication) [RSA78, GM84, Pai99].

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

Later on it became possible to combine multiple types of operations in a limited way [BGN05, GHV10, SYY99]. Gentry's first fully homomorphic encryption scheme [Gen09] relied on several non-standard computational assumptions, among which assumptions about quantum computational power, needed even for security against *classical* attackers. 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 also believed to be computationally hard 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].

In this thesis, we assume for clarity of exposition that for all classical homomorphic schemes, it is possible to immediately decrypt after encrypting (without doing any evaluation), even though this is not guaranteed by definition. However, it suffices to evaluate the identity operation before decrypting, so any scheme that is homomorphic for a set containing the identity operation can be safely assumed to have this property.

We will use the notation $\widetilde{x}$ to denote the result of running $\mathsf{HE.Enc}_{pk}(x)$: that is, $\mathsf{Dec}_{sk}(\widetilde{x}) = x$ with overwhelming probability. In our construction, we will often deal with multiple classical key sets $(pk_i, sk_i, evk_i)_{i\in I}$ indexed by some set I. In that case, we use the notation $\widetilde{x}^{[i]}$ to denote the result of $\mathsf{HE.Enc}_{pk_i}(x)$, in order to avoid confusion. Recall from Section 2.1.1 that (e.g.) pk_i does *not* refer to the i^{th} bit of the public key: for this, we use the notation $s[i]$.

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

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

Some homomorphic encryption schemes define a different, more efficient recryption procedure [Gen09], but the above definition shows that recryption is possible for any classical homomorphic encryption scheme.

3.2 Quantum homomorphic encryption

A quantum homomorphic encryption scheme QHE, as defined in [BJ15], is a natural extension classical definition, 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 from Section 3.1 carry over quite naturally to the quantum setting (see also [BJ15]):

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

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

$\sigma' \leftarrow \mathsf{QHE.Eval}^{\mathsf{C}}_{\rho_{evk}}(\sigma)$ represents the evaluation of a circuit C. If C requires n input qubits, then σ should be 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 C would output. The evaluation key ρ_{evk} is consumed in the process.

$\rho' \leftarrow \mathsf{QHE.Dec}_{sk}(\sigma')$ maps a single state σ' from the output space to a quantum state ρ' in the message space. Note that if the evaluation procedure $\mathsf{QHE.Eval}$ outputs a product of n' states, then $\mathsf{QHE.Dec}$ 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 divisible schemes, where decryption happens qubit-by-qubit.

Yu, Pérez-Delgado and Fitzsimons [YPDF14] showed that perfectly information-theoretically secure quantum fully homomorphic encryption (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, Tan and Fitzsimons provides information-theoretic security for circuits with at most a constant number of non-Clifford operations [OTF15].

Schemes that are based on computational assumptions have only recently been thoroughly investigated by Broadbent and Jeffery. In [BJ15], they give formal definitions of QFHE and its computational security, and they propose two schemes that achieve homomorphic encryption for nontrivial sets of quantum circuits, which we discuss in Section 3.4. Instead of trying to achieve information-theoretic security, they build their schemes based on a classical FHE scheme and hence any computational assumptions on the classical scheme are also required for the quantum schemes. Such computational assumptions allow bypassing the impossibility result from [YPDF14], and help to work toward a (quantum) fully homomorphic encryption scheme.

3.3 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 3.3. *[BJ15] The* quantum CPA indistinguishability experiment *with respect to a scheme* QHE *and a quantum polynomial-time adversary* $\mathscr{A} = (\mathscr{A}_1, \mathscr{A}_2)$*, denoted by* $\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{QHE}}(\kappa)$*, is defined by the following procedure:*

1. $\mathsf{KeyGen}(1^\kappa)$ *is run to obtain keys* (pk, sk, ρ_{evk}).
2. *Adversary* $\mathscr{A}_1$ *is given* (pk, ρ_{evk}) *and outputs a quantum state on* $\mathcal{M} \otimes \mathcal{E}$.
3. *For* $r \in \{0,1\}$, *let* $\Xi^{cpa,r}_{\mathsf{QHE}} : D(\mathcal{M}) \to D(\mathcal{C})$ *be:* $\Xi^{cpa,0}_{\mathsf{QHE}}(\rho) = \mathsf{QHE.Enc}_{pk}(|0\rangle\langle 0|)$ *and* $\Xi^{cpa,1}_{\mathsf{QHE}}(\rho) = \mathsf{QHE.Enc}_{pk}(\rho)$. *A random bit* $r \in \{0,1\}$ *is chosen and* $\Xi^{cpa,r}_{\mathsf{QHE}}$ *is applied to the state in* $\mathcal{M}$ *(the output being a state in* $\mathcal{C}$*).*
4. *Adversary* $\mathscr{A}_2$ *obtains the system in* $\mathcal{C} \otimes \mathcal{E}$ *and outputs a bit* r'.
5. *The output of the experiment is defined to be 1 if* $r' = r$ *and* 0 *otherwise. In case* $r = r'$, *we say that* $\mathscr{A}$ wins *the experiment.*

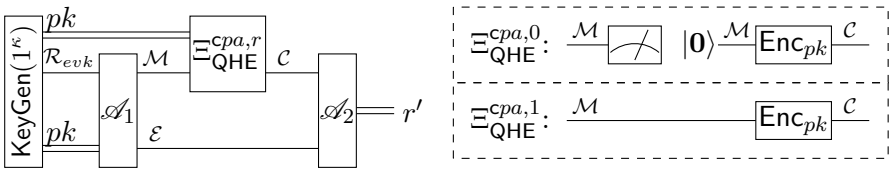


Figure 3.1: [BJ15, Figure 1, reproduced with permission of the authors] The quantum CPA indistinguishability experiment $\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{QHE}}(\kappa)$. Double lines represent classical information flow, and single lines represent quantum information flow. The adversary $\mathscr{A}$ is split up into two separate algorithms $\mathscr{A}_1$ and $\mathscr{A}_2$, which share a working memory represented by the quantum state in register $\mathcal{E}$.

The game $\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{QHE}}(\kappa)$ is depicted in Figure 3.1. 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. This quantum indistinguishability experiment can also straightforwardly be defined for a classical HE scheme: in this case, all wires are classical, and the functionality $\Xi^{cpa,r}_{\mathsf{HE}}$ determines whether or not the chosen message is replaced by a (classical) zero. The adversary still has quantum power.

If an adversary cannot win the indistinguishability game with more than negligible advantage, the encryption procedure is considered to be q-IND-CPA secure:

Definition 3.4. *[BJ15, Definition 3.3] A (classical or quantum) homomorphic encryption scheme* S *is* q-IND-CPA *secure if for any quantum polynomial-time adversary* $\mathscr{A} = (\mathscr{A}_1, \mathscr{A}_2)$ *there exists a negligible function* η *such that:*

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

Analogously to $\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{S}}(\kappa)$, in the game $\mathsf{PubK}^{\mathsf{cpa-mult}}_{\mathscr{A},\mathsf{S}}(\kappa)$, the adversary can give multiple messages to the challenger, which are either all encrypted, or all replaced by zeros. Broadbent and Jeffery [BJ15] show that these notions of security are equivalent.

3.4 QHE for Clifford circuits

In this section, we describe a fairly basic quantum encryption scheme that is homomorphic for all Clifford operations. The idea for this Clifford scheme has been around for quite some time,

but in [BJ15] it was first written down and proven q-IND-CPA secure. Because our scheme TP will be defined as an extension of this scheme, we give an informal overview here. A more formal definition can be found in [BJ15], or extracted as a special case of TP (see Chapter 5) for zero T gates.

The Clifford scheme (CL) is built upon some classical FHE scheme HE. A message qubit is encrypted with a quantum one-time pad (see Section 2.2.1), and the classical keys to this pad are sent along as classical information, encrypted using $\mathsf{HE.Enc}$. Arbitrary Clifford circuits can now be evaluated gate-by-gate by applying each gate to the *encrypted* quantum state and, using the commutation rules for the Clifford group with the Pauli group, update the keys via homomorphic evaluation. For example, to evaluate a phase gate P on the (encrypted) state $\mathsf{X}^a\mathsf{Z}^b|\psi\rangle$, perform the gate to get the resulting state

$$\mathsf{P}\mathsf{X}^a\mathsf{Z}^b|\psi\rangle = \mathsf{X}^a\mathsf{Z}^{a\oplus b}\mathsf{P}|\psi\rangle,$$

which is an encryption of $\mathsf{P}|\psi\rangle$, but with different keys. The evaluator can use $\mathsf{HE.Eval}$ to (homomorphically) compute encryptions of these new keys a and $a \oplus b$ from the encryptions of a and b. For the update rules for all nontrivial Clifford gates, see Appendix A.1. After all gates have been evaluated, decryption is performed by first decrypting the (updated) keys to the quantum one-time pad, and then removing this pad from the output state.

The evaluation procedure of CL works perfectly for all Clifford gates, because these gates commute with the Pauli group. 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. If we try to evaluate a T gate in this way on a state $\mathsf{X}^a\mathsf{Z}^b|\psi\rangle$, we end up with the state

$$\mathsf{T}\mathsf{X}^a\mathsf{Z}^b|\psi\rangle = \mathsf{P}^a\mathsf{X}^a\mathsf{Z}^b\mathsf{T}|\psi\rangle$$

which is not a quantum one-time pad encryption of $\mathsf{T}|\psi\rangle$. After the application of the T gate, an *error* P^a has appeared on the encrypted state. If a is known, this error can easily be corrected by applying $\mathsf{P}^\dagger$ whenever $a = 1$. However, the evaluating party only has access to some encrypted version $\widetilde{a}$ of the key a, and hence is not able to decide whether or not to apply a correction. The value of a is also not known in advance to the key generating party or encrypting party (because it depends on which quantum operations have been evaluated on the input state so far), and so it cannot be supplied to the evaluator in an encrypted form.

In an effort to deal with the evaluation of a T gate, Broadbent and Jeffery [BJ15] have presented two schemes that extend CL, accomplishing homomorphic encryption for circuits with a limited number of T gates. 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, effectively removing all accumulated errors on the state at once. 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.

In Chapter 5, we will present a scheme that also extends CL by supplying auxiliary quantum states (gadgets) in the evaluation key. Using the construction from Theorem 4.5 in the next chapter, it will be possible to create a small gadget that combines the knowledge of the key generating and evaluating parties, and applies an error correction whenever necessary.

CHAPTER 4

COMPUTATION THROUGH TELEPORTATION

Quantum teleportation [NC00] is a powerful tool in quantum computation. At the expense of a Pauli error, we can use pairs of entangled qubits to instantaneously transfer a quantum state to a location that may be very far away. If we so wish, we can set up the entangled resource to perform quantum operations to the state that is teleported 'through' it [GC99]. However, no information about the teleported state is revealed until the Pauli error is removed.

The goal of this chapter is to construct a quantum state, essentially a collection of entangled pairs of qubits, which serves as a *conditional computation gadget*: it can be used to apply a fixed (Clifford) gate on a single-qubit input state, but only if some fixed boolean function evaluates to 1 on an input that is only partially known to the party that is using the gadget. In the construction of our quantum homomorphic encryption scheme, this gadget will be used by the evaluator to correct errors that may be present after the evaluation of a T gate on the encrypted data, even though he does not possess all the information about whether such an error is present.

In Section 4.1, the theory behind quantum teleportation is explained. The other ingredient for our gadget construction, Barrington's Theorem, is briefly discussed in Section 4.2. Using the binary **OR** function as a running example, we show how to combine Barrington's Theorem and quantum teleportation first into a two-party process in Section 4.3, and then into the desired gadget structure in Section 4.4.

4.1 Quantum teleportation

Consider the following two-qubit states:

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

$$\left|\Psi^+\right\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \qquad \left|\Psi^-\right\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$$

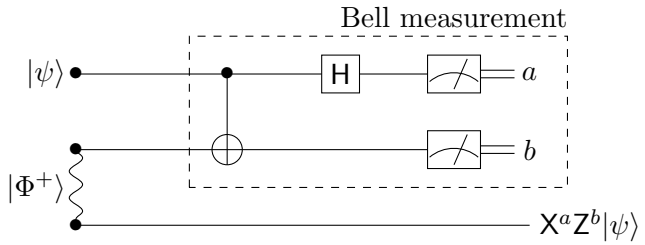


Figure 4.1: A quantum circuit representing the teleportation protocol. The EPR pair $|\Phi^+\rangle$ is represented with a snaky line, and is used up as a resource during the process. Within the Bell measurement subprocedure, the symbols for first a CNOT gate, then an H gate, and finally two computational basis measurements are shown.

These states are called *EPR pairs* or *Bell pairs*, and play an important role in quantum teleportation. Consider what happens if we measure (only) the first qubit of the two-qubit state $|\Phi^+\rangle$. With equal probability, a 0 or a 1 is observed, which makes the entire state collapse to $|00\rangle$ or $|11\rangle$ respectively. The second qubit, which used to be in a superposition of 0 and 1, can be considered to instantly collapse to the same value that was observed in the measurement of the first qubit. This *entanglement* of the two qubits is a crucial ingredient for quantum teleportation.

To see how quantum teleportation works, consider the quantum circuit depicted in Figure 4.1. An arbitrary qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ is connected to the upper qubit of an EPR pair through a *Bell measurement*, a measurement in the basis consisting of the four Bell states. It can be implemented by a CNOT, H and two computational basis measurements. By calculating the effects of the CNOT and H gates, one can verify that the three-qubit state, right before the computational basis measurements, is of the form

$$\frac{1}{2}(\alpha|000\rangle + \beta|001\rangle + \beta|010\rangle + \alpha|011\rangle + \alpha|100\rangle - \beta|101\rangle - \beta|110\rangle + \alpha|111\rangle).$$

When we measure the top two qubits, we will get one of four outcomes $ab \in \{00, 01, 10, 11\}$ with equal probability. Immediately, the bottom qubit of the EPR pair collapses to the state $\mathsf{X}^a\mathsf{Z}^b|\psi\rangle$. For example, if the measurement results in 01, the entire state collapses into $\beta|010\rangle + \alpha|011\rangle = |01\rangle \otimes \mathsf{X}^1\mathsf{Z}^0|\psi\rangle$. This happens instantaneously, even when the two halves of the EPR pair are very far apart. Note, however, that the Pauli error $\mathsf{X}^a\mathsf{Z}^b$ completely hides the content of the qubit to anyone who does not know a and b (see Section 2.2.1), and cannot be removed until the party that has performed the Bell measurement has communicated the measurement outcomes through some classical channel. Thus, quantum teleportation cannot be used to transport information instantaneously, thereby not contradicting special relativity.

4.1.1 Entanglement swapping

Apart from teleporting a pure state through an EPR pair, it is also possible to teleport a mixed state. As an example, let us teleport half of *another* EPR pair $|\Phi^+\rangle$: a Bell measurement is performed on the second and third qubit of the state $|\Phi^+\rangle_{12} \otimes |\Phi^+\rangle_{34}$, with measurement outcomes a and b. Using similar calculations as before, we can verify that this results in the

entanglement of qubits 1 and 4, which will have become one of the four possible EPR pairs (more specifically, they will be in the state $(\mathsf{X}^a\mathsf{Z}^b)_4|\Phi^+\rangle_{14}$. This effect is called *entanglement swapping* [ZZHE93], since after the H and CNOT operations, the second and third qubit are also entangled.

Teleporting a qubit through this new entangled state has the same effect as teleporting it through $|\Phi^+\rangle_{12}$ first and then through $|\Phi^+\rangle_{34}$: in both cases, two independent random Paulis are applied to the state. Therefore, when teleporting a qubit through multiple EPR pairs, the order of measurement is irrelevant.

4.2 Barrington's theorem

A surprising result in classical complexity theory is Barrington's theorem [Bar89]. It states that all functions in $\mathbf{NC}^1$ (i.e. functions computable by a log-depth Boolean circuit of fan-in 2) share a certain structure in terms of so-called *permutation branching programs*. We will start by giving a brief introduction to these types of programs.

In this thesis, we will be concerned with *permutations* on sets $[k] := \{1, ..., k\}$, for some $k \in \mathbb{N}_+$. These permutations on finite sets can be described in a natural way using *cycle notation*: for example, the notation (142)(35) denotes a permutation on the set [5] that consists of a 3-cycle (mapping $1 \mapsto 4$, $4 \mapsto 2$, and $2 \mapsto 1$) and a 2-cycle (mapping $3 \mapsto 5$ and vice versa).

Definition 4.1. *A* k-permutation branching program *(*k*-PBP) of length* L *on an input* $x = x_1x_2\cdots x_n$ *is a list of* L *instructions of the form* $\langle i_\ell, \pi_\ell, \sigma_\ell\rangle$ *(for* $1 \leq \ell \leq L$*) such that:* $i_\ell \in [n]$*, and* π_ℓ *and* σ_ℓ *are permutations on the set* $[k]$*. The program is executed by composing the permutations given by the instructions 1 through* L*, selecting*[1] π_ℓ *if* $x_{i_\ell} = 1$*, or* σ_ℓ *if* $x_{i_\ell} = 0$*.*

The result of executing a k-permutation branching program is thus a permutation on $[k]$. We say that a k-permutation branching program *computes* a Boolean function f on input x if it returns the identity permutation e whenever $f(x) = 0$, and returns some fixed k-cycle μ otherwise.

Example 4.2. *The* **OR** *function on two bits can be computed using a 5-permutation branching program of length 4. It is defined by the following list of instructions:*

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

This program results in the permutation (14235) if $\mathbf{OR}(x_1, x_2) = 1$*, and in the identity permutation* e *otherwise. See Figure 4.2 for an example run on input (0,1).*

Constant-width permutation branching programs turn out to be powerful enough to capture $\mathbf{NC}^1$ circuits efficiently, as formalized in the following theorem [Bar89]:

Theorem 4.3 (Barrington's theorem)**.** *Every boolean circuit* C *with depth* d *and fan-in 2 can be computed by a 5-permutation branching program of length at most* 4^d*.*

[1] This order, where the *second* permutation is selected if $x_{i_\ell} = 0$ and the *first* if $x_{i_\ell} = 1$, may seem reversed, but it consistent with common use and with the way it was first introduced in [Bar89]. One should think of a program instruction as an 'if ␣ then ␣ else ␣ ' statement.

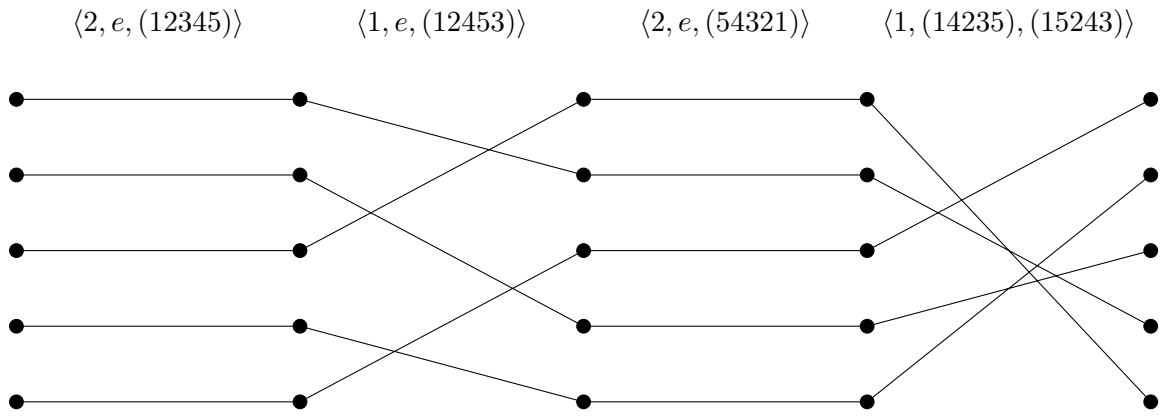


Figure 4.2: Execution of the 5-permutation branching program for the **OR** function on input $(0, 1)$. See also Example 4.2. The program's instructions are displayed above the permutations. The execution for this input results in the permutation $e(12453)e(15243) = (14235)$, as expected.

As a corollary of this theorem, it follows that if C is an $\mathbf{NC}^1$ circuit, then C has depth $O(\log n)$ by definition, and hence its function can be computed using a polynomial-length 5-permutation branching program.

4.3 Instantaneous non-local quantum computation

Because every permutation step only queries a single bit of the input, permutation branching programs are easily adapted to a setting where the input of the function is distributed amongst multiple parties. Suppose that two parties, Alice and Bob, hold the respective inputs x and y, and together want to compute some classical function $f(x, y)$ by permuting a set of k qubits according to some k-cycle π iff $f(x, y)$. If one of the players starts off with k qubits, they can *distributedly* execute a k-permutation branching program for f by letting Alice permute the qubits whenever a program instruction queries a bit from x, and letting Bob do the permutation whenever a bit of y is queried. In between program instructions, they can send the qubits back and forth, or even teleport them (at the expense of a random Pauli error).

For an example computation of the **OR** function according to the program from Example 4.2, see Figure 4.3. For each teleportation action, Alice and Bob need to share k EPR pairs, to which they connect (through Bell measurements) the k qubits according to the program instructions. After all program instructions have been executed in this way, the k qubits end up on the 'output' side of the last set of k EPR pairs, permuted according to the result of the k-permutation branching program. Note, however, that neither Alice nor Bob knows how the qubits are permuted until they communicate about how they connected up the EPR pairs on their respective sides of the protocol. Moreover, there will be a random Pauli error (determined by the outcomes of the Bell measurements) on each of the qubits.

The ideas of the above construction are exploited in [Spe15] to perform *instantaneous*

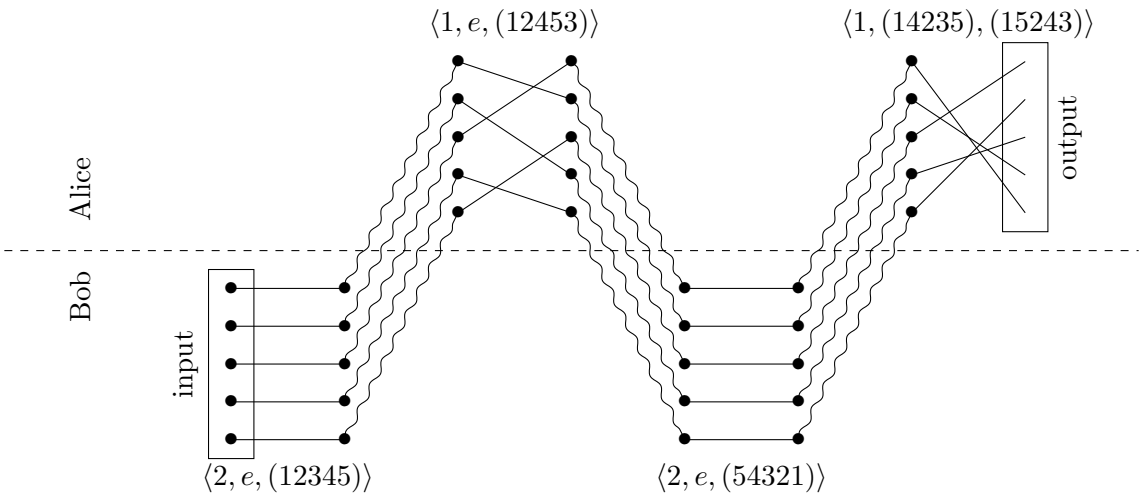


Figure 4.3: A distributed execution of a 5-permutation branching program for the **OR** function on input (0,1), where Alice holds the first input bit (0) and Bob holds the second (1). Bob starts the computation with five input qubits, which end up permuted according to (14235) in the output position. Snaky lines represent EPR pairs, while smooth lines represent Bell measurements on the pairs of qubits they connect.

non-local quantum computation, allowing two parties to jointly perform a computation on a quantum state with only a single round of simultaneous communication (to share their inputs and measurement outcomes). In the spirit of [Spe15], we construct a protocol for two parties to perform a single-qubit Clifford operation conditioned on some (function of) distributed classical information. The amount of EPR pairs needed for the computation depends exponentially on the circuit depth of the function that is computed on the distributed inputs.

Lemma 4.4 (variation on [Spe15, Lemma 8])**.** *Assume Bob holds a single qubit in the state*[2] $|\psi\rangle$*, along with some classical* $y \in \{0,1\}^n$*. Alice holds* $x \in \{0,1\}^n$*. Let* $\mathsf{C} \in \mathcal{C}_1$ *be some single-qubit Clifford operation, and let* $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ *be computable by some Boolean circuit of depth* d*. Then there exists an instantaneous protocol without any communication that uses* $10(4^d + 1)$ *pre-shared EPR pairs, resulting in a known qubit of Bob being in the state* $\mathsf{X}^a\mathsf{Z}^b\mathsf{C}^{f(x,y)}|\psi\rangle$ *for some* $a, b \in \{0,1\}$*. The values* a *and* b *depend on* x*,* y *and the* $20(4^d + 1)$ *measurement outcomes of Alice and Bob, and can be efficiently computed from those input values.*

Proof. The protocol is as follows: Alice and Bob perform a distributed 5-PBP computation for $f(x, y)$ using EPR pairs, as described above. Without loss of generality, assume that the computation starts on Bob's side and and finishes on Alice's (if it does not, an additional $2 \cdot 5$ EPR pairs suffice to ensure that it does). Bob uses $|\psi\rangle$ as the first input qubit, and ancillary $|0\rangle$ qubits for the other four input positions. Let π be the permutation induced by the 5-PBP in case $f(x, y) = 1$. Alice applies the operation C to the output qubit in position $\pi(1)$. Then she uses these 5 output qubits as an input to the *reverse* 5-PBP for f, which is defined by

[2]For notational convenience, we will assume that the state is pure. The lemma also holds for mixed states, and the proof is identical.

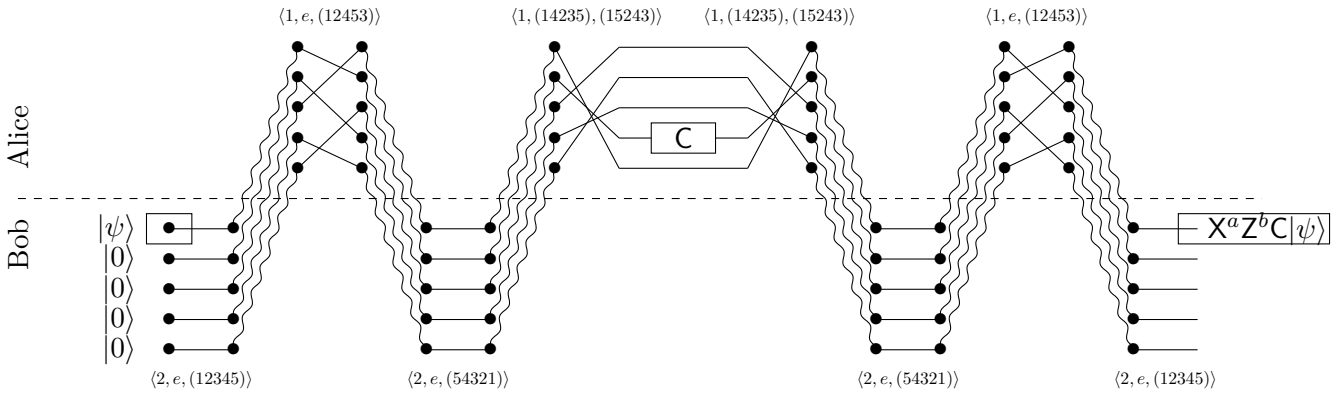


Figure 4.4: An example execution of the instantaneous non-local computation protocol for a Clifford gate C conditioned on $\mathbf{OR}(0,1)$, where Alice holds the first input bit and Bob holds the second input bit. Additionally, Bob has an input qubit $|\psi\rangle$ that he wants to apply the gate C to, plus four ancilla qubits. The output of the protocol is $\mathsf{X}^a\mathsf{Z}^b\mathsf{C}|\psi\rangle$, because $\mathbf{OR}(0,1) = 1$ and so the qubit is routed through the C gate. Again, snaky lines represent EPR pairs, and continuous lines represent Bell measurements (or, on the far right, regular circuit wires), possibly preceded by a C gate.

executing all program instructions in the reverse order, inverting the permutations for each instruction. The ℓ^{th} instruction $\langle i_\ell, \pi_\ell, \sigma_\ell\rangle$ of the original program becomes the $(L-\ell+1)^{\text{th}}$ instruction $\langle i_\ell, \pi_\ell^{-1}, \sigma_\ell^{-1}\rangle$ in the reverse program. The output of this reverse program ends up on Bob's side, and the first output qubit is in a state of the form $\mathsf{X}^a\mathsf{Z}^b\mathsf{C}^{f(x,y)}|\psi\rangle$, as we will argue below. For an example execution of the protocol, see Figure 4.4.

First, we demonstrate that the protocol indeed uses at most[3] $10(4^d+1)$ EPR pairs. Barrington's theorem ensures the existence of a 5-PBP for f of length at most 4^d. To compute this program using teleportation, the qubits have to be teleported at most 4^d-1 times between Alice and Bob, plus possibly an extra two times to make sure the computation starts and ends with Bob and Alice respectively. In total, $5(4^d+1)$ EPR pairs suffice for the distributed computation of the 5-PBP for f. Because Alice and Bob need to compute the entire protocol in reverse, the total amount of EPR pairs needed is doubled to $10(4^d+1)$.

Next, we show that at the end of the computation, a known qubit is in a state of the form $\mathsf{X}^a\mathsf{Z}^b\mathsf{C}^{f(x,y)}|\psi\rangle$, by considering the path that the input qubit takes during the computation. After the execution of the 'forward' direction of the 5-PBP, one of the five qubits will be in the state $\mathsf{X}^{a_1}\mathsf{Z}^{b_1}|\psi\rangle$ for some $a_1, b_1 \in \{0,1\}$. These values a_1, b_1 can be straightforwardly computed by adding, modulo 2, those measurement outcomes from Alice and Bob that correspond to the path the qubit has taken. This path depends directly on the program and the values x and y. If $f(x,y) = 1$, the qubit will be at position $\pi(1)$, while if $f(x,y) = 0$, it will be at position $e(1) = 1$. Because π is a cycle permutation by definition of the 5-PBP, $\pi(1) \neq 1$. Hence, by applying C only to the output qubit at position $\pi(1)$, Alice ensures that the relevant qubit is now in the state $\mathsf{C}^{f(x,y)}\mathsf{X}^{a_1}\mathsf{Z}^{b_1}|\psi\rangle$. However, the location of the qubit is still unknown to Alice and Bob if they cannot communicate. Computing the 5-PBP in reverse ensures that

[3]If multiple consecutive program instructions query input bits from the same player, the qubits do not have to be teleported back and forth in between. Alice or Bob can just apply all consecutive permutations immediately one after another.

the qubit ends up in the first output position independently of the values x, y or $f(x, y)$, since $\pi^{-1}(\pi(1)) = e^{-1}(e(1)) = 1$. Because of the teleportations, additional random Paulis will have been applied to the state, resulting in

$$\mathsf{X}^{a_2}\mathsf{Z}^{b_2}\mathsf{C}^{f(x,y)}\mathsf{X}^{a_1}\mathsf{Z}^{b_2}|\psi\rangle \tag{4.1}$$

for some $a_2, b_2 \in \{0, 1\}$. Again, these values are efficiently computable from the measurement outcomes and the values x, y.

Remember that the Clifford operations commute with the Pauli group (see Section 2.3). Therefore, $\mathsf{CX}^{a_1}\mathsf{Z}^{b_1} = \mathsf{X}^{a_1'}\mathsf{Z}^{b_1'}\mathsf{C} =$ for some $a_1', b_1' \in \{0, 1\}$. These values can be efficiently computed from a_1 and b_1, see Appendix A.1. Therefore, the state in (4.1) can be rewritten to $\mathsf{X}^{a_2 \oplus a_1'}\mathsf{Z}^{b_2 \oplus b_1'}\mathsf{C}|\psi\rangle$ if $f(x, y) = 1$. In general, the state of the first output qubit is of the form $\mathsf{X}^{a}\mathsf{Z}^{b}\mathsf{C}^{f(x,y)}|\psi\rangle$, where $(a, b) = (a_2 \oplus a_1, b_2 \oplus b_1)$ if $f(x, y) = 0$, while $(a, b) = (a_2 \oplus a_1', b_2 \oplus b_1')$ if $f(x, y) = 1$. In either case, the values a and b are efficiently computable from x and y and the measurement outcomes of both Alice and Bob. □

With the construction from Lemma 4.4, Alice and Bob can instantaneously and non-locally perform a single-qubit Clifford operation (plus a random Pauli that can only be removed after a round of simultaneous communication), conditioned on some function f. If $f \in \mathbf{NC}^1$, they can perform the operation using only polynomially many EPR pairs (in the length of x, y). A stronger version of Lemma 4.4 also holds [Spe15, Lemma 8], and is proven using roughly the same construction, but for *garden-hose computations* [BFSS13] instead of distributed 5-PBP computations.

In a garden-hose computation, Alice and Bob start out with a number of EPR pairs which they can connect (through Bell measurements) dependent on their inputs x and y. Bob connects an input qubit to one of the EPR pairs, and the value of $f(x, y)$ will determine where this qubits ends up. Distributed executions of 5-PBP programs are a subset of the garden-hose computations, but garden-hose computations are not limited to connecting only groups of 5 EPR pairs according to some list of program instructions. More general strategies are allowed, which are potentially more efficient in terms of the number of resource EPR pairs. In particular, for any log-space computable f there exists an (efficiently computable) garden-hose strategy using only polynomially many EPR pairs in the length of x, y [BFSS13]. In other words, the *garden-hose complexity* of log-space computable functions is polynomial.

Speelman [Spe15] shows that a single-qubit Clifford $\mathsf{C}^{f(x,y)}$ can be performed instantaneously and non-locally using polynomially many EPR pairs, provided that f has polynomial garden-hose complexity. It is, however, not guaranteed that Alice and Bob can efficiently compute the list of measurements they need to perform given their inputs x and y, because the garden-hose model allows for arbitrary pre-processing of the inputs x and y. Since we need these strategies to be efficiently computable for our construction of a quantum homomorphic encryption scheme in Chapter 5, we will apply the result from [Spe15] only to log-space computable functions, for which it is known that the garden-hose strategies for Alice and Bob can be computed efficiently [BFSS13, Theorem 2.12].

4.4 Conditional computation gadgets

In the previous section, we have seen how two parties Alice and Bob can jointly perform a Clifford operation conditioned on some classical information that is distributed among them,

with the output state affected by a random Pauli operator. Even though we like to think of the qubit in the protocol as being sent back and forth between Alice and Bob step-by-step, with permutations in between, this is not necessarily the case. In fact, the order in which the Bell measurements are performed is irrelevant for the outcome of the protocol because of entanglement swapping (see Section 4.1.1). In this section, we consider a variant on the non-local computation protocol where Alice performs all of her measurements first, effectively creating a 'gadget' for Bob to use depending on his input. This gadget will be an important ingredient for the construction of our QFHE scheme in Chapter 5.

Theorem 4.5. *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be a log-space computable function. Then there exists a number m polynomial in n, two quantum algorithms $\mathsf{GenGadget}_f$ and $\mathsf{UseGadget}_f$ and a classical algorithm $\mathsf{ComputeKeys}_f$ that all run in time poly(n), such that:*

(i) $\mathsf{GenGadget}_f(\mathsf{C}, x)$*, given $\mathsf{C} \in \mathcal{C}_1$ and $x \in \{0,1\}^n$, produces some classical information $g_{f,\mathsf{C}}(x)$ and, using some randomness $r = (r_x, r_z) \in \{0,1\}^{2m}$, creates a* gadget *$\gamma_r(g_{f,\mathsf{C}}(x))$ of size $2m$ from this classical information.*

(ii) $\mathsf{UseGadget}_f(\gamma_r(g_{f,\mathsf{C}}(x)), y, \rho)$*, for $\gamma_r(g_{f,\mathsf{C}}(x)) \leftarrow \mathsf{GenGadget}_f(\mathsf{C}, x)$, $y \in \{0,1\}^n$, and a single-qubit state ρ, produces an output state $\mathsf{X}^a\mathsf{Z}^b\mathsf{C}^{f(x,y)}\rho(\mathsf{C}^\dagger)^{f(x,y)}\mathsf{Z}^b\mathsf{X}^a$ for some $a, b \in \{0,1\}$, and a list of outcomes from m Bell measurements, $r' \in \{0,1\}^{2m}$.*

(iii) $\mathsf{ComputeKeys}_f(g_{f,\mathsf{C}}(x), y, r, r') = (a, b)$.

Proof. For clarity of exposition, we prove the theorem only for functions with $\mathbf{NC}^1$ circuits. The construction can straightforwardly be adapted to log-space computable f if we base the it on [Spe15, Lemma 8] instead of Lemma 4.4, but one has to closely follow the proof of [BFSS13, Theorem 2.12] to ensure that $\mathsf{GenGadget}_f$ and $\mathsf{UseGadget}_f$ run in poly(n) time. The definition of $\mathsf{ComputeKeys}_f$ is also somewhat more complex, but it still runs in time poly(n): see [DSS16, Appendix A.2].

The algorithms $\mathsf{GenGadget}_f$ and $\mathsf{UseGadget}_f$ resemble the tasks of Alice and Bob, respectively, in the protocol in Lemma 4.4. First, let $2m$ be the number of EPR pairs needed for the execution of that protocol: that is, $2m = 10(4^d + 1)$ where d is the smallest depth of a circuit computing f. Clearly, if f has an $\mathbf{NC}^1$ circuit, then m is polynomial in n. We prove the three claims stated in Theorem 4.5 one by one:

(i) Define $\mathsf{GenGadget}_f(\mathsf{C}, x)$ as follows: using the instructions from the 5-PBP for f and the distributed protocol described in Lemma 4.4, create a list $((s_i, t_i))_{i=1}^m$ of Bell measurements that Alice would perform for her part of the protocol: the item (s_i, t_i) describes a Bell measurement between (Alice's halves of) the s_i^{th} and t_i^{th} EPR pair. This list can be generated in time poly(n) simply by going through the instruction list for the 5-PBP and, conditioned on the relevant bits of x, adding the 5 Bell measurements for that instruction to the list. Furthermore, let $c \in \{0,1\}^m$ describe which of these m Bell measurements should be preceded by the execution of C.[4] Define

$$g_{f,\mathsf{C}}(x) := ((s_i, t_i)_{i=1}^m, c, x, \mathsf{C}).$$

where the last item C is just some classical description of the C gate, for example its index in some fixed ordering of the finite set $\mathcal{C}_1$. Using this classical information, the

[4] In the construction of Lemma 4.4, there is only one such measurement, but in the more general construction from [Spe15, Lemma 8], there may be more than one measurement that is preceded by C gate.

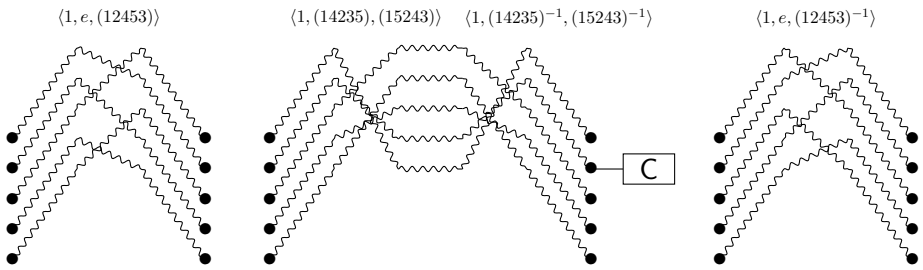


Figure 4.5: Gadget $\gamma_r(g_{\mathbf{OR},\mathsf{C}}(0))$ for the **OR** function for $x = 0$. The snaky lines represent EPR pairs, each of which is independently and uniformly randomly picked from the set $\{|\Phi^+\rangle, |\Phi^-\rangle, |\Psi^+\rangle, |\Psi^-\rangle\}$ by starting with $|\Phi^+\rangle$ applying $\mathsf{X}^{r_x[i]}\mathsf{Z}^{r_z[i]}$ to one of the qubits of the i^{th} pair, for random $r = (r_x, r_z)$. On one EPR pair, the gate C is applied afterwards. The gadget can be used by connecting up the three sets of EPR pairs according to Bob's strategy for the **OR** function, and his input $y \in \{0, 1\}$. For both possible inputs, the correct permutation emerges.

quantum gadget $\gamma_r(g_{f,\mathsf{C}}(x))$ is created by constructing m EPR pairs $|\Phi^+\rangle$, applying random quantum one-time pads $\mathsf{X}^{r_x[i]}\mathsf{Z}^{r_z[i]}$ to the second qubit of each pair, and permuting their qubits according to the list $(s_i, t_i)_{i=1}^m$ so that we end up with a list of $2m$ qubits, where the s_i^{th} qubit is maximally entangled with the t_i^{th} qubit. For every pair (s_i, t_i), the operation C is applied to the t_i^{th} qubit if $c[i] = 1$. The resulting state is

$$\gamma_r(g_{f,\mathsf{C}}(x)) := \prod_{i=1}^{m} \left(\mathsf{C}^{c[i]}\mathsf{X}^{r_x[i]}\mathsf{Z}^{r_z[i]}\right)_{t_i} |\Phi^+\rangle\langle\Phi^+|_{s_i t_i} \left(\mathsf{Z}^{r_z[i]}\mathsf{X}^{r_x[i]}\left(\mathsf{C}^\dagger\right)^{c[i]}\right)_{t_i}$$

of size $2m$. This size is completely independent of x, as is the length of $g_{f,\mathsf{C}}(x)$.

Note that this state is exactly the state that results from first creating $2m$ EPR pairs as a resource for the protocol from Lemma 4.4, then applying C to the relevant qubits, and finally swapping entanglement (see Section 4.1.1) by performing Bell measurements according to Alice's part of the protocol, to connect up the right EPR pairs. See Figure 4.5 for a gadget constructed for the **OR** function.

(ii) Define $\mathsf{UseGadget}_f(\gamma_r(g_{f,\mathsf{C}}(x)), y, \rho)$ as the execution of Bob's strategy from Lemma 4.4, treating the $2m$ qubits from $\gamma_r(g_{f,\mathsf{C}}(x))$ as the resource EPR pairs. By that lemma, there is a known qubit in a state of the form $\mathsf{X}^a\mathsf{Z}^b\mathsf{C}^{f(x,y)}\rho(\mathsf{C}^\dagger)^{f(x,y)}\mathsf{Z}^b\mathsf{X}^a$ (for some $a, b \in \{0, 1\}$) after execution of the protocol. Set this qubit to be the output of $\mathsf{UseGadget}_f$. The algorithm $\mathsf{UseGadget}_f$ can be executed in time $\text{poly}(n)$ by executing the program instructions for the 5-PBP program one by one.

(iii) The path of the input qubit can be determined efficiently from the structure of the gadget, listed in $g_{f,\mathsf{C}}(x)$, and the value of y combined with the program instructions for f. The X and Z keys before and after the conditional application(s) of C can be computed by adding (modulo 2) the corresponding measurement outcomes and one-time pad keys contained in r' and r respectively. Finally, using c to determine whether or not a C gate occurs on the path, combined with the commutation rules listed in Appendix A.1, the values of a and b can be computed. For a more detailed description of the algorithm $\mathsf{ComputeKeys}_f$, see Appendix A.2.

□

We will refer to the gadget created by $\mathsf{GenGadget}_f$ as a *conditional computation gadget*, since Bob (or any other party holding y) can use it to apply some gate C, determined by whoever created the gadget, conditioned on $f(x, y)$. Alice (or any other party holding x) can create the gadget without knowledge of y.

An important property of the conditional computation gadget is that the quantum part is completely mixed to anyone who does not know the randomness r, regardless of the structure of the gadget:

Lemma 4.6. *For all $f : \{0,1\}^n \otimes \{0,1\}^n \to \{0,1\}$, $\mathsf{C} \in \mathcal{C}_1$, and $x \in \{0,1\}^n$:*

$$\frac{1}{2^{2m}} \sum_{r \in \{0,1\}^{2m}} \gamma_r(g_{f,\mathsf{C}}(x)) = \frac{\mathbb{I}_{2^{2m}}}{2^{2m}}.$$

Proof. By distributivity of the matrix product over the direct sum, we have

$$\begin{aligned}
&\sum_{r \in \{0,1\}^{2m}} \gamma_r(g_{f,\mathsf{C}}(x)) \\
&= \prod_{i=1}^{m} \sum_{r_x[i], r_z[i] \in \{0,1\}} \left(\mathsf{C}^{c[i]} \mathsf{X}^{r_x[i]} \mathsf{Z}^{r_z[i]}\right)_{t_i} \left|\Phi^+\right\rangle\!\left\langle\Phi^+\right|_{s_i t_i} \left(\mathsf{Z}^{r_z[i]} \mathsf{X}^{r_x[i]} \left(\mathsf{C}^\dagger\right)^{c[i]}\right)_{t_i} \\
&= \prod_{i=1}^{m} \left(\mathsf{C}^{c[i]}\right)_{t_i} \left(\sum_{r_x[i], r_z[i] \in \{0,1\}} \left(\mathsf{X}^{r_x[i]} \mathsf{Z}^{r_z[i]}\right)_{t_i} \left|\Phi^+\right\rangle\!\left\langle\Phi^+\right|_{s_i t_i} \left(\mathsf{Z}^{r_z[i]} \mathsf{X}^{r_x[i]}\right)_{t_i} \right) \left(\left(\mathsf{C}^\dagger\right)^{c[i]}\right)_{t_i} \\
&= \prod_{i=1}^{m} \left(\mathsf{C}^{c[i]}\right)_{t_i} \left(\left|\Phi^+\right\rangle\!\left\langle\Phi^+\right|_{s_i t_i} + \left|\Phi^-\right\rangle\!\left\langle\Phi^-\right|_{s_i t_i} + \left|\Psi^+\right\rangle\!\left\langle\Psi^+\right|_{s_i t_i} + \left|\Psi^-\right\rangle\!\left\langle\Psi^-\right|_{s_i t_i} \right) \left(\left(\mathsf{C}^\dagger\right)^{c[i]}\right)_{t_i} \\
&= \prod_{i=1}^{m} \left(\mathsf{C}^{c[i]}\right)_{t_i} (\mathbb{I}_4)_{s_i t_i} \left(\left(\mathsf{C}^\dagger\right)^{c[i]}\right)_{t_i} \\
&= \prod_{i=1}^{m} (\mathbb{I}_4)_{s_i t_i} \\
&= \mathbb{I}_{2^{2m}}.
\end{aligned}$$

The step from $|\Phi^+\rangle\langle\Phi^+|_{s_i,t_i} + |\Phi^-\rangle\langle\Phi^-|_{s_i,t_i} + |\Psi^+\rangle\langle\Psi^+|_{s_i,t_i} + |\Psi^-\rangle\langle\Psi^-|_{s_i,t_i}$ to identity can be made by adding the 4×4 matrices for these four states, or by observing that the Bell states form a basis for the two-qubit-state space $\mathbb{C}^4$. □

This property will be important in the security proof of the scheme presented in the next chapter; intuitively it shows that these gadgets do not reveal any information about x whenever the randomness r is unknown or securely encrypted.

CHAPTER 5

A TELEPORTATION-BASED QUANTUM HOMOMORPHIC ENCRYPTION SCHEME

5.1 Definition of the scheme TP

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 [BJ15], 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.

For every T gate, the evaluation key contains one gadget, along with some classical information on how to use that gadget. The number of gadgets in the evaluation key thus grows linearly with the upper bound to the number of T gates in the circuit, and the size of each gadget is polynomial in the security parameter.

The scheme TP is built upon an arbitrary classical fully homomorphic encryption scheme HE. We assume throughout this chapter that HE.Dec runs in space logarithmic in the security parameter κ, and is q-IND-CPA secure.

5.1.1 Key generation

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

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

[1] Our scheme is leveled, so an extra parameter L is supplied to the key generation function, see Section 3.1.

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 $L-1$, create error correction gadgets as follows:

   a) Run the procedure $\mathsf{GenGadget}_{\mathsf{HE.Dec}}(\mathsf{P}^\dagger, sk_i)$ from Theorem 4.5(i) to produce some classical information $g_i := g_{\mathsf{HE.Dec},\mathsf{P}^\dagger}(sk_i)$ and subsequently a random quantum gadget $\gamma_r(g_i)$ consisting of $2m$ qubits (and dependent on some randomness r).

   b) Encrypt the classical information g_i and the randomness r using the classical homomorphic encryption scheme and the *next* public key pk_{i+1} to create the mixed state

$$\Gamma_{pk_{i+1}}(sk_i) := \rho\left(\mathsf{HE.Enc}_{pk_{i+1}}(g_i)\right) \otimes \frac{1}{2^{2m}} \sum_{r\in\{0,1\}^{2m}} \left(\rho\left(\mathsf{HE.Enc}_{pk_{i+1}}(r)\right) \otimes \gamma_r(g_i)\right).$$

   If $\mathsf{HE.Dec}$ is a logspace computable function, then by Theorem 4.5(i) the gadgets will be of size polynomial in the length of sk_i, which is in turn polynomial in κ.

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

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

Note that because the classical information g_i that accompanies each gadget contains information about the secret key sk_i, it might not be secure to encrypt it using the public key pk_i unless the classical homomorphic encryption scheme has circular security (meaning that it can securely encrypt key dependent messages [MTY11]). This is why we need to use $L+1$ different classical key sets. The evaluator will have to do some recrypting during the evaluation phase, but otherwise using independent keys does not complicate the construction much. More details on how the evaluation procedure deals with the different keys is provided in Section 5.1.3.

5.1.2 Encryption

The encryption procedure $\mathsf{TP.Enc}$ is identical to $\mathsf{CL.Enc}$, using the first public key pk_0 for the encryption of the one-time-pad keys. Every single-qubit state σ is encrypted separately with a quantum one-time pad, and the pad key is (classically) encrypted and appended to the quantum encryption of σ, resulting in the classical-quantum state:

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

5.1.3 Circuit evaluation

Consider a circuit C with n wires. The evaluation of the circuit on the encrypted data is carried out one gate at a time, treating gates that are on the same layer separately.

Recall from Section 2.4 that our quantum circuit can be written using only the standard set $\{\mathsf{H}, \mathsf{CNOT}, \mathsf{T}\}$. Although in principle all single-qubit Clifford operations can be constructed using only H and $\mathsf{T} = \sqrt{P}$, we will consider them all as basic gates to build circuits with, because they can be evaluated considerably more efficiently than their equivalent circuits constructed with only H and T.

Before the evaluation of a single gate G, the encryption of an n-qubit state ρ is of the form

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

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

Clifford gates

For the evaluation of a single-qubit Clifford gate or a CNOT, we proceed exactly as in $\mathsf{CL.Eval}$ (see Section 3.4). The gate is simply applied to the encrypted qubit(s), and since it commutes with the Pauli group, the evaluator only needs to update the encrypted keys in a straightforward way (see Appendix A.1). These updates are performed using the procedure $\mathsf{HE.Eval}$ for the update functions of the keys.

T gate

The evaluator starts out by applying a T gate to the appropriate wire w. Afterwards, the qubit at wire w is in the state

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

In order to remove the P error, the evaluator uses one gadget $\Gamma_{pk_{i+1}}(sk_i)$ from the evaluation key; he possesses the classical information $\widetilde{a_w}^{[i]}$ encrypted with the correct key, so by Theorem 4.5(ii), he can run the procedure

$$\mathsf{UseGadget}_{\mathsf{HE.Dec}}\left(\gamma_r(g_i),\quad \widetilde{a_w}^{[i]},\quad \big(\mathsf{P}^{a_w}\mathsf{X}^{a_w}\mathsf{Z}^{b_w}\mathsf{T}\big)\rho_w\big(\mathsf{T}^\dagger\mathsf{X}^{a_w}\mathsf{Z}^{b_w}(\mathsf{P}^\dagger)^{a_w}\big)\right)$$

to obtain a state of the form

$$\left(\mathsf{X}^{a'_w}\mathsf{Z}^{b'_w}(\mathsf{P}^\dagger)^{\mathsf{HE.Dec}(sk_i,\widetilde{a_w}^{[i]})}\mathsf{P}^{a_w}\mathsf{X}^{a_w}\mathsf{Z}^{b_w}\mathsf{T}\right)\rho_w\left(\mathsf{T}\mathsf{Z}^{b_w}\mathsf{X}^{a_w}\mathsf{P}^{a_w}\mathsf{P}^{\mathsf{HE.Dec}(sk_i,\widetilde{a_w}^{[i]})}\mathsf{Z}^{b'_w}\mathsf{X}^{a'_w}\right)$$

for some $a'_w, b'_w \in \{0,1\}$. With very high probability, $a_w \leftarrow \mathsf{HE.Dec}(sk_i, \widetilde{a_w}^{[i]})$, in which case the above state equals

$$\left(\mathsf{X}^{a'_w \oplus a_w}\mathsf{Z}^{b'_w \oplus b_w}\mathsf{T}\right)\rho_w\left(\mathsf{T}\mathsf{Z}^{b'_w \oplus b_w}\mathsf{X}^{a'_w \oplus a_w}\right)$$

which is of the desired form. Let r' describe the Bell measurement outcomes of the evaluator during the execution of $\mathsf{UseGadget}_{\mathsf{HE.Dec}}$. Encryptions (under pk_{i+1}) of g_i and the randomness r from $\gamma_r(g_i)$ are contained in the evaluation key $\Gamma_{pk_{i+1}}(sk_i)$. Using the functionality

$\mathsf{ComputeKeys}_{\mathsf{HE.Dec}}$ from Theorem 4.5(iii), the evaluator can homomorphically compute the values a'_w and b'_w by executing

$$\mathsf{HE.Eval}_{pk_{i+1}}^{\mathsf{ComputeKeys}_{\mathsf{HE.Dec}}}\left(\widetilde{g}_i^{[i+1]},\ \ \mathsf{HE.Rec}_{i\to(i+1)}\left(\widetilde{a_w}^{[i]}\right),\ \ \widetilde{r}^{[i+1]},\ \ \mathsf{HE.Enc}_{pk_{i+1}}(r')\right).$$

Then, recrypting all previous keys $a_1, ..., a_w, b_1, ..., b_w$ into the new key set, the evaluator can homomorphically update the keys to wire w by adding a'_w and b'_w to the old keys (modulo 2).

Note that the gadget is destroyed by the Bell measurements. The evaluation of every new T gate requires the consumption of a new conditional computation gadget, and transfers all classical encryptions to the next key set.

At the end of the evaluation of a circuit C containing k T gates, the evaluator holds a one-time-pad encryption of the state $\mathsf{C}\rho\mathsf{C}^\dagger$, 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.

5.1.4 Decryption

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

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 5.1. *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 $\mathsf{HE.Eval}^f$ 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 gates more for applying X^{a_w} and Z^{b_w}. The total number of steps is polynomial in κ and independent of the circuit, so we conclude that TP is compact. □

5.2 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. that no polynomial-time quantum adversary can tell the difference between an encryption of a real message and an encryption of $|0\rangle\langle 0|$, even if he gets to choose the message himself (recall the definition of q-IND-CPA security from Section 3.3). 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 5.2. *If* HE *is q-IND-CPA secure, then* TP *is q-IND-CPA secure for circuits containing up to polynomially (in κ) many* T *gates.*

In order to prove Theorem 5.2, 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 5.4). 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 $\mathsf{TP}^{(\ell)}$ be identical to TP, except for the key generation procedure: $\mathsf{TP}^{(\ell)}.\mathsf{KeyGen}$ replaces, for every $i \geq \ell$, all classical information $g_i = g_{\mathsf{HE.Dec},\mathsf{P}^\dagger}(sk_i)$ and r accompanying the i^{th} gadget with the all-zero string of length $|g_i| + |r| = O(m)$ before encrypting it.

Note that the length of the classical information does not depend on sk_i itself: the size $2m$ of the gadget and the length of sk_i are completely determined by the choice of the protocol HE and the security parameter κ. Hence, a potential adversary cannot gain any information about sk_i just from this encrypted string of zeros.

In summary,

$$\begin{aligned}\mathsf{TP}^{(\ell)}.\mathsf{KeyGen}(1^\kappa, 1^L) := \bigotimes_{i=0}^{L-1} |evk_i\rangle\langle evk_i| \otimes \bigotimes_{i=0}^{\ell-1} \Gamma_{pk_{i+1}}(sk_i) \otimes \\ \bigotimes_{i=\ell}^{L-1} \Big(\rho(\mathsf{HE.Enc}_{pk_{i+1}}(0^{|g_i|})) \otimes \\ \frac{1}{2^{2m}} \sum_{r \in \{0,1\}^{2m}} \rho(\mathsf{HE.Enc}_{pk_{i+1}}(0^{2m})) \otimes \gamma_r(g_i) \Big) .\end{aligned}$$

Intuitively, one can view $\mathsf{TP}^{(\ell)}$ as the scheme that provides only ℓ usable gadgets in the evaluation key. Note that $\mathsf{TP}^{(L)} = \mathsf{TP}$, and that in $\mathsf{TP}^{(0)}$, only the classical evaluation keys remain, since without the encryptions of the classical r, the quantum part of the gadget is just the completely mixed state. By Lemma 4.6, we can rewrite the final line of the previous equation as

$$\begin{aligned}\frac{1}{2^{2m}} \sum_{r \in \{0,1\}^{2m}} \rho(\mathsf{HE.Enc}_{pk_{i+1}}(0^{2m})) \otimes \gamma_r(g_i) \\ = \rho(\mathsf{HE.Enc}_{pk_{i+1}}(0^{2m})) \otimes \frac{\mathbb{I}_{2^{2m}}}{2^{2m}} . \end{aligned} \tag{5.1}$$

With the definitions of the new schemes, we can lay out the steps to proving Theorem 5.2 in more detail. First, we show that in the quantum CPA indistinguishability experiment, any efficient adversary interacting with $\mathsf{TP}^{(\ell)}$ only has negligible advantage over an adversary interacting with $\mathsf{TP}^{(\ell-1)}$, i.e. the scheme where the classical information $g_{\ell-1}$ is removed (Lemma 5.3). By iteratively applying this argument, we are able to argue that the advantage of an adversary who interacts with $\mathsf{TP}^{(L)}$ over one who interacts with $\mathsf{TP}^{(0)}$ is also negligible (Corollary 5.4). Finally, we conclude the proof by arguing that $\mathsf{TP}^{(0)}$ is q-IND-CPA secure by comparison to the CL scheme.

Lemma 5.3. *Let $0 < \ell \leq L$. If* HE *is q-IND-CPA secure, then for any quantum polynomial-time adversary $\mathscr{A} = (\mathscr{A}_1, \mathscr{A}_2)$, there exists a negligible function η such that*[2]

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

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

Let $\mathscr{A} = (\mathscr{A}_1, \mathscr{A}_2)$ be an adversary for the game $\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{TP}^{(\ell)}}(\kappa)$. We will define an adversary $\mathscr{A}' = (\mathscr{A}'_1, \mathscr{A}'_2)$ for $\mathsf{PubK}^{\mathsf{cpa-mult}}_{\mathscr{A}',\mathsf{HE}}(\kappa)$ that will either simulate the game $\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{TP}^{(\ell)}}(\kappa)$ or $\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{TP}^{(\ell-1)}}(\kappa)$. Which game is simulated will depend on some $s \in_R \{0,1\}$ that is unknown to $\mathscr{A}'$ himself. Using the assumption that HE is q-IND-CPA secure, we are able to argue that $\mathscr{A}'$ is unable to recognize which of the two schemes was simulated. This fact allows us to bound the difference in success probabilities between the security games of $\mathsf{TP}^{(\ell)}$ and $\mathsf{TP}^{(\ell-1)}$. The structure of this proof is very similar to e.g. Lemma 5.3 in [BJ15]. The adversary $\mathscr{A}'$ acts as follows (see also Figure 5.1):

$\mathscr{A}'_1$ takes care of most of the key generation procedure: he generates the classical key sets 0 through $\ell - 1$ himself, generates the random strings $r_0, \ldots, r_{\ell-1}$, and constructs the gadgets $\gamma_{r_0}(g_0), \ldots, \gamma_{r_{\ell-1}}(g_{\ell-1})$ and their classical information $g_0, \ldots, g_{\ell-1}$. He encrypts the classical information using the appropriate public keys. Only $g_{\ell-1}$ and $r_{\ell-1}$ are left unencrypted: instead of encrypting these strings himself using pk_ℓ, $\mathscr{A}'_1$ sends the strings for encryption to the challenger. Whether the challenger really encrypts $g_{\ell-1}$ and $r_{\ell-1}$ or replaces them with a string of zeros, determines which of the two schemes is simulated. $\mathscr{A}'$ is unaware of the random choice of the challenger.

The adversary $\mathscr{A}'_1$ also generates the extra padding inputs that correspond to the already-removed gadgets ℓ up to $L - 1$. Since by definition of $\mathsf{TP}^{(\ell)}$, these gadgets consist of all-zero strings encrypted with independently chosen public keys that are not used anywhere else, together with a completely mixed quantum state (as shown in Equation 5.1), the adversary can generate them without needing any extra information.

$\mathscr{A}'_2$ feeds the evaluation key and public key, just generated by $\mathscr{A}'_1$, to $\mathscr{A}_1$ in order to obtain a chosen message $\mathcal{M}$ (plus the auxiliary state $\mathcal{E}$). He then picks a random $t \in_R \{0,1\}$ and erases $\mathcal{M}$ if and only if $t = 0$. He encrypts the result according to the TP.Enc procedure (using the public key $(pk_i)_{i=0}^{L}$ received from $\mathscr{A}'_1$), and gives the encrypted state, plus $\mathcal{E}$, to $\mathscr{A}_2$, who outputs t' in an attempt to guess t. $\mathscr{A}'_2$ now outputs s' := 1 if and only if the guess by $\mathscr{A}$ was correct, i.e. $s' := t \equiv t'$.

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

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

[2]Note that the difference in probabilities is only bounded in one direction. The other direction can be proven with a slight adaptation of the definition of $\mathscr{A}'$ (by outputting $s' := t \not\equiv t'$, see Figure 5.1). However, for the proof of Theorem 5.2, this bound suffices.

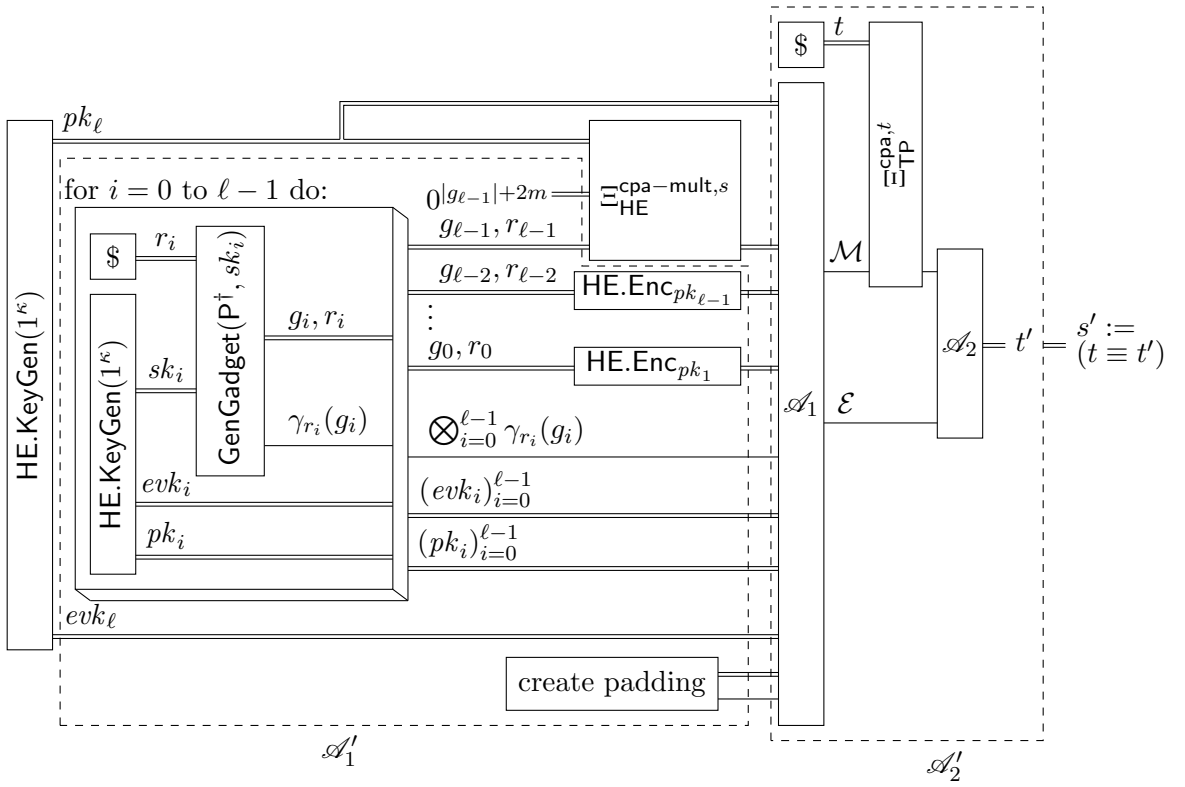


Figure 5.1: A strategy for the game $\mathsf{PubK}^{\mathsf{cpa-mult}}_{\mathscr{A}',\mathsf{HE}}(\kappa)$, using an adversary $\mathscr{A}$ for $\mathsf{PubK}^{\mathsf{cpa}}_{\mathscr{A},\mathsf{TP}^{(\ell)}}(\kappa)$ as a subroutine. All the wires that form an input to $\mathscr{A}_1$ together form the evaluation key and public key for $\mathsf{TP}^{(\ell)}$ or $\mathsf{TP}^{(\ell-1)}$, depending on s. Note that $\Xi^{\mathsf{cpa},t}_{\mathsf{TP}} = \Xi^{\mathsf{cpa},t}_{\mathsf{TP}^{(\ell)}} = \Xi^{\mathsf{cpa},t}_{\mathsf{TP}^{(\ell-1)}}$, so $\mathscr{A}'_2$ can run either one of these independently of s (i.e. without having to query the challenger). The 'create padding' subroutine generates dummy gadgets for ℓ up to $L-1$, as described in the definition of $\mathscr{A}_1$.

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

From the above, we conclude that

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

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

By applying Lemma 5.3 iteratively, L times in total, we can conclude that the difference between $\mathsf{TP}^{(L)}$ and $\mathsf{TP}^{(0)}$ is negligible, because the sum of polynomially many negligible functions is still negligible:

Corollary 5.4. *If L is polynomial in κ, then for any quantum polynomial-time adversary $\mathscr{A} = (\mathscr{A}_1, \mathscr{A}_2)$, there exists a negligible function η such that*

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

Using Corollary 5.4, we can finally prove the q-IND-CPA security of our scheme $\mathsf{TP} = \mathsf{TP}^{(L)}$.

Proof of Theorem 5.2. The scheme $\mathsf{TP}^{(0)}$ is very similar to CL in terms of its key generation and encryption steps. The evaluation key consists of several classical evaluation keys, plus some completely mixed states and encryptions of 0 which we can safely ignore because they do not contain any information about the encrypted message. In both schemes, the encryption of a qubit is a quantum one-time pad together with the encrypted keys. The only difference is that in $\mathsf{TP}^{(0)}$, the public key and evaluation key form a tuple containing a list of public/evaluation keys that are independent of the encryption, in addition to pk_0 and evk_0 which are used for the encryption of the quantum one-time pad. This list of extra keys does not provide any advantage (in fact, the adversary could have generated it himself by repeatedly running $\mathsf{HE.KeyGen}(1^\kappa, 1^L)$). 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 $\mathsf{TP}^{(0)}$, the exact same proof shows that $\mathsf{TP}^{(0)}$ is q-IND-CPA secure as well, that is, for any $\mathscr{A}$ there exists a negligible function η' such that

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

Combining this result with Corollary 5.4, it follows that

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

Since the sum of two negligible functions is itself negligible, we have proved Theorem 5.2. □

5.3 Circuit privacy of TP

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 which circuit C was applied to the input ρ by the evaluator. Obviously, the output value $\mathsf{C}\rho\mathsf{C}^\dagger$ often reveals something about the circuit C, but apart from this necessary leakage of information, one may require a (quantum) homomorphic encryption scheme to ensure *circuit privacy* in the sense that an adversary cannot statistically gain any information about C from the output of the evaluation procedure, other than what it can already gain from $\mathsf{C}\rho\mathsf{C}^\dagger$ itself.

5.3.1 Classical circuit privacy

Classically, circuit privacy is defined by requiring the existence of a simulator $\mathsf{Sim}_{\mathsf{HE}}$ whose inputs are the publicly known parameters (such as the public key and the security parameter) and the value $\mathsf{C}(x)$, and whose output is indistinguishable from the homomorphic evaluation of C on the encryption of x. Formally, circuit privacy is defined as follows.

Definition 5.5 (Classical circuit privacy – semi-honest setting [IP07])**.** *A classical homomorphic encryption scheme* HE *has statistical circuit privacy in the semi-honest ('honest-but-curious') model if there exists a PPT algorithm* $\mathsf{Sim}_{\mathsf{HE}}$ *and a negligible function* η *such that for any security parameter* κ*, input* x*, key set* $(pk, evk, sk) \leftarrow \mathsf{HE.KeyGen}(1^\kappa)$*, and circuit* C*:*

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

Here, $\delta(X, Y) := \frac{1}{2}\sum_{u\in U}\big|\Pr[X = u] - \Pr[Y = u]\big|$ is the *statistical distance* between two random variables over a finite universe U. We will sometimes write $\mathsf{X} \approx_a Y$ to indicate that $\delta(X, Y) \leq a$.

If the recryption functionality $\mathsf{HE.Rec}_{i\to j}$ is defined as the composition of the procedures $\mathsf{HE.Eval}^{\mathsf{HE.Dec}_i}_{evk_j}$ and $\mathsf{HE.Enc}_{pk_j}$, as in Section 3.1, then recryptions do not degrade the privacy of the computation: a homomorphic evaluation of some function with key switching is statistically close to running the simulator directly on the function output using only the *last* key set.

Lemma 5.6. *Suppose* HE *has statistical circuit privacy in the semi-honest setting, and let* $\mathsf{Sim}_{\mathsf{HE}}$ *and* η *be as in Definition 5.5. Then for any security parameter* κ*,* L *polynomial in* κ*, list of circuits* $\mathsf{C}_1, ..., \mathsf{C}_L$ *and list of keysets* $(pk_i, evk_i, sk_i)_{i=1}^L$ *generated by* $\mathsf{HE.KeyGen}(1^\kappa)$*, and input* x*, the statistical distance between*

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

and

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

is negligible in κ*.*

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

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

which, by correctness of HE, is statistically indistinguishable from

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

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

In some classical schemes, more efficient recryption is possible than $\mathsf{HE.Eval}^{\mathsf{HE.Dec}_{sk_i}} \circ \mathsf{HE.Enc}_{pk_j}$. For these schemes, Lemma 5.6 would have to be proven separately for the specific scheme.

5.3.2 Quantum circuit privacy

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

Definition 5.7 (Quantum circuit privacy – semi-honest setting)**.** *A quantum homomorphic encryption scheme* QHE *has statistical circuit privacy in the semi-honest setting if there exists a quantum PPT algorithm* $\mathsf{Sim}_{\mathsf{QHE}}$ *and a negligible function* η *such that for any security parameter* κ*, depth parameter* L*, key set* $(pk, \rho_{evk}, sk) \leftarrow \mathsf{QHE.KeyGen}(1^\kappa, 1^L)$*, state* σ*, and circuit* C *with up to* L T*-gates:*

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

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

We claim that circuit privacy for TP in the semi-honest setting (i.e. against passive adversaries[3]) can be obtained by modifying the scheme only slightly, given that the classical encryption scheme has the circuit privacy property.

Theorem 5.8. *If* HE *has circuit privacy in the semi-honest setting, then* TP *can be adapted to a quantum homomorphic encryption scheme with circuit privacy.*

Informally, if the evaluator randomizes the encrypted 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. We make this informal argument explicit.

Proof. We make the following alteration to the scheme TP: at the end of the evaluation procedure, the evaluator applies a (random) quantum one-time pad to the output of the evaluation, and updates the classical encryptions of the keys accordingly. The rest of the scheme remains exactly the same, and it is clear that this altered version of TP is still compact and correct.

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

To formalize this intuition, we define a quantum algorithm $\mathsf{Sim}_{\mathsf{TP}}$ satisfying the constraints given in Definition 5.7. Let $\mathsf{Sim}_{\mathsf{HE}}$ be the classical simulator guaranteed to exist by the classical circuit privacy of HE (see Definition 5.5). Given some security parameter κ, some keys $pk = (pk_1, ..., pk_L)$ and $evk = (evk_1, ..., evk_L)$, and some quantum state σ, let $\mathsf{Sim}_{\mathsf{TP}}$ apply a uniformly random quantum one-time pad to σ, and apply $\mathsf{Sim}_{\mathsf{HE}}(1^\kappa, pk_L, evk_L, \cdot)$ to the pad keys. The resulting classical-quantum state is the output of $\mathsf{Sim}_{\mathsf{TP}}$. This algorithm resembles TP.Enc, but instead of calling HE.Enc (with pk_1) as a subroutine, it handles the pad key information using the classical simulator $\mathsf{Sim}_{\mathsf{HE}}$ (with pk_L).

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

If we can show that the trace distance

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

is negligible for any quantum state σ of an appropriate dimension, then quantum circuit privacy of TP immediately follows from Defintion 5.7 and the definition of the diamond norm.

Write $\sigma_{sim} := \left((\Phi^{\rho_{evk},pk}_{\mathsf{Sim_{TP}}} \circ \Phi_{\mathsf{C}}) \otimes \mathbb{I} \right) \sigma$, and $\sigma_{eval} := \left(\left(\Phi^{\mathsf{C},\rho_{evk},pk}_{\mathsf{TP.Eval}} \circ \Phi^{pk}_{\mathsf{TP.Enc}} \right) \otimes \mathbb{I} \right) \sigma$. We study the state σ_{sim} in more detail, and show how to transform it into σ_{eval} in a few (negligible) steps. As a result, the trace distance of these two states is negligible.

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

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

During the evaluation procedure of TP, the evaluator updates the keys to the quantum one-time pad for all n qubits in the circuit. These updates depend on the circuit that is being evaluated, some randomness r from the Bell measurement outcomes[4] and of course on the initial one-time pad keys. Let $f_i^{\mathsf{C},r}(a,b)$ denote the X key on the i^{th} qubit after the evaluation of some circuit C with randomness r, with $a, b \in \{0,1\}^n$ the initial pad keys before the evaluation procedure. Similarly, let $g_i^{\mathsf{C},r}(a,b)$ denote the Z key for that qubit.

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

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

This is where the classical circuit privacy property kicks in: for any fixed $i, a, b, \mathsf{C}, r, x$, the result of the probabilistic computation $\mathsf{Sim_{HE}}(f_i^{\mathsf{C},r}(a,b) \oplus x[i])$ is statistically indistinguishable from the evaluation of the function $f_i^{\mathsf{C},r}(\cdot,\cdot) \oplus x[i]$ on the encryptions of a and b. Note however

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

that the evaluation of $f_i^{\mathsf{C},r}$ is performed in several steps, with key switching in between. That is, separate functions h_1 through h_L are evaluated in each key set 1 through L, such that $f_i^{\mathsf{C},r} = h_L \circ \cdots \circ h_1$. We abstract away from the exact way in which the function $f_i^{\mathsf{C},r}$ is broken up into these separate functions $h_1, ..., h_L$, and simply write $\mathsf{HE.Eval}_{1,...,L}^{f_i^{\mathsf{C},r}(\cdot,\cdot)\oplus x[i]}(\mathsf{HE.Enc}_{pk_1}(a,b))$ to denote

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

By Lemma 5.6, it follows that

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

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

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

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

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

is at most $4n \cdot \eta(\kappa)$. Note that this last state is exactly σ_{eval}, the result of putting σ through the channel $(\Phi_{\mathsf{TP.Eval}}^{\mathsf{C},\rho_{evk},pk} \circ \Phi_{\mathsf{TP.Enc}}^{pk}) \otimes \mathbb{I}$. We conclude that for any σ,

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

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

$$\left\| (\Phi_{\mathsf{TP.Eval}}^{\mathsf{C},\rho_{evk},pk} \circ \Phi_{\mathsf{TP.Enc}}^{pk}) - (\Phi_{\mathsf{Sim_{TP}}}^{\rho_{evk},pk} \circ \Phi_{\mathsf{C}}) \right\|_\diamond =$$

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

which is negligible if η is negligible. □

5.4 Quantum power required for TP.KeyGen

In a setting where a less powerful client wants to delegate some computation to a more powerful server, it is important to minimize the amount of quantum work the client needs to do. For example, the client might not have access to a universal quantum computer and might only be able to perform a limited variety of quantum operations. The client could send his homomorphically encrypted quantum data to another party that can perform any quantum computation.

TP.Enc and TP.Dec only require the application of Pauli operators to a quantum state, but TP.KeyGen is more involved. In the current description of the gadget generation (see Section 4.4), the client has to be able to perform a variety of tasks to generate the keys: he has to generate EPR pairs, as well as perform $\mathsf{P}^\dagger$ gates and Bell measurements. We show in this section how the gadgets can be generated securely using only X, Z and swap operations, when the client is supplied with resources by some computationally more powerful (but potentially malicious) party, for example the evaluator. The client can prevent the leakage of information about his input, even when the provided resources are corrupted by the supplier.

As described in Section 4.4, we see from Figure 4.5 that the quantum part of a gadget, $\gamma_r(g_i)$, is effectively a list of $2m$ EPR pairs (some of which have an extra $(\mathsf{I}\otimes\mathsf{P}^\dagger)$ transformation on them), with the qubits ordered in some way that depends on sk_i. If the key generator is supplied with a list of $2m$ EPR pairs $|\Phi^+\rangle$ and as many pairs $(\mathsf{I}\otimes\mathsf{P}^\dagger)|\Phi^+\rangle$, it is clear that he can create the gadget by swapping some of the qubits (e.g. using CNOT gates), and applying random Pauli operations (using X and Z gates) on every pair. Any unused pairs are discarded. For the conditional computation gadgets based on Barrington's theorem, one pair $(\mathsf{I}\otimes\mathsf{P}^\dagger)|\Phi^+\rangle$ suffices, but for the general gadget based on the garden-hose model, more than one may be needed.

If the supplier of these pairs follows the protocol and sends actual EPR pairs to the key generator, this tactic suffices to hide all information about sk_i. However, if the supplier acts maliciously, he may send two qubits to the key generator claiming that they form an EPR pair, while in reality he is keeping some form of entanglement with one or both of the qubits. We need to make sure that even in this case, where the supplier actively tries to gather information about sk_i or the client's input, this information is still secure.

The key generator, upon receiving the (real or fake) EPR pairs, can apply independently selected random Pauli transformations on *every* qubit instead of just on one qubit of every (claimed) EPR pair. By applying this transformation to both qubits, any entanglement that a malicious supplier might hold with any of them becomes completely useless. Since a swap of two qubits consists of three CNOT gates that commute with the Pauli's, the state after swapping the qubits into the correct order is still completely mixed. Hence, no information about sk_i is revealed to the supplier. By updating the classical information r that accompanies the gadget, the client ensures that the gadget can still be used for the homomorphic evaluation of a T gate.

CHAPTER 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 evaluated 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.

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 any 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. Hence, the size of the evaluation key only grows linearly in the upper bound to the number of T gates in the circuit (and polynomially in the security parameter), allowing the scheme to be leveled fully homomorphic. 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, so 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 log-space computable decryption procedure and to be secure against quantum adversaries. In particular, no circular security assumption is required. Since we supply at most a polynomial number of evaluation

gadgets, our scheme TP is *leveled* quantum fully homomorphic by construction, and we simply switch to a new classical key after every evaluation gadget. In fact, under every key set we only need to be able to perform a limited amount of classical computation: 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. If the classical FHE scheme has circuit privacy, then this extra randomization completely hides the quantum 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 round-optimal scheme for *blind delegated quantum computation* [Chi05, BFK09, ABOE10, VFPR14, FBS+14, Bro15, Lia15], using computational assumptions. With only a single round of communication, 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 number and complexity of quantum operations that the client needs to perform. In our scheme, the encryption and decryption only requires the client to apply Pauli operations. We have shown that even the creation of the evaluation key can be performed (with the help of the server, at the expense of an extra communication round) using only swap and Pauli operations.

6.1 Future work

Since Yu, Pérez-Delgado and Fitzsimons [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 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. We also consider it likely that our new techniques will be useful in other contexts such as quantum indistinguishability obfuscation [AF16].

References

[ABF+16] Gorjan Alagic, Anne Broadbent, Bill Fefferman, Tommaso Gagliardoni, Christian Schaffner, and Michael St. Jules. Computational security of quantum encryption. *arXiv preprint arXiv:1602.01441*, 2016.

[ABOE10] Dorit Aharonov, Michael Ben-Or, and Elad Eban. Interactive proofs for quantum computations. *Proceeding of Innovations in Computer Science 2010 (ICS'10)*, pages 453–469, 2010.

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

[AJLA+12] Gilad Asharov, Abhishek Jain, Adriana López-Alt, Eran Tromer, Vinod Vaikuntanathan, and Daniel Wichs. Multiparty computation with low communication, computation and interaction via threshold FHE. In *Advances in Cryptology–EUROCRYPT 2012*, pages 483–501. Springer, 2012.

[AMTW00] Andris Ambainis, Michele Mosca, Alain Tapp, and Ronald de Wolf. Private quantum channels. In *41st IEEE Symposium on Foundations of Computer Science (FOCS 00)*, pages 547–553, 2000.

[Arm88] Mark A Armstrong. *Groups and symmetry*. Springer-Verlag, 1988.

[AS06] Pablo Arrighi and Louis Salvail. Blind quantum computation. *International Journal of Quantum Information*, 4(05):883–898, 2006.

[Bar89] David A. Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in NC1. *Journal of Computer and System Sciences*, 164:150–164, 1989.

[BB14] Ämin Baumeler and Anne Broadbent. Quantum private information retrieval has linear communication complexity. *Journal of Cryptology*, 28(1):161–175, 2014.

[BCG+06] Michael Ben-Or, Claude Crépeau, Daniel Gottesman, Avinatan Hassidim, and Adam Smith. Secure multiparty quantum computation with (only) a strict honest majority. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 249–260. IEEE Computer Society, 2006.

[Ben82] Paul Benioff. Quantum mechanical Hamiltonian models of Turing machines. *Journal of Statistical Physics*, 29(3):515–546, 1982.

[BFK09] Anne Broadbent, Joseph Fitzsimons, and Elham Kashefi. Universal blind quantum computation. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 517–526. IEEE, 2009.

[BFSS13] Harry Buhrman, Serge Fehr, Christian Schaffner, and Florian Speelman. The garden-hose model. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, ITCS '13, pages 145–158, New York, NY, USA, 2013. ACM.

[BGN05] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-DNF formulas on ciphertexts. In *Theory of cryptography*, pages 325–341. Springer, 2005.

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 309–325. ACM, 2012.

[BJ15] Anne Broadbent and Stacey Jeffery. Quantum homomorphic encryption for circuits of low T-gate complexity. In *Advances in Cryptology–CRYPTO 2015*, pages 609–629. Springer, 2015.

[BMP+99] P Oscar Boykin, Tal Mor, Matthew Pulver, Vwani Roychowdhury, and Farrokh Vatan. On universal and fault-tolerant quantum computing: a novel basis and a new constructive proof of universality for Shor's basis. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 486–494. IEEE, 1999.

[Bro15] Anne Broadbent. Delegating private quantum computations. *Canadian Journal of Physics*, 93(9):941–946, 2015.

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 97–106, Oct 2011.

[CDN01] Ronald Cramer, Ivan Damgård, and Jesper B Nielsen. *Multiparty computation from threshold homomorphic encryption*. Springer, 2001.

[Chi05] Andrew M Childs. Secure assisted quantum computation. *Quantum Information & Computation*, 5(6):456–466, 2005.

[CKGS98] Benny Chor, Eyal Kushilevitz, Oded Goldreich, and Madhu Sudan. Private information retrieval. *Journal of the ACM (JACM)*, 45(6):965–981, 1998.

[DN05] Christopher M Dawson and Michael A Nielsen. The Solovay-Kitaev algorithm. *arXiv preprint quant-ph/0505030*, 2005.

[DNS10] Frédéric Dupuis, Jesper Buus Nielsen, and Louis Salvail. Secure two-party quantum evaluation of unitaries against specious adversaries. In *Proceedings of the 30th Annual Conference on Advances in Cryptology*, CRYPTO'10, pages 685–706, Berlin, Heidelberg, 2010. Springer-Verlag.

[DSS16] Yfke Dulek, Christian Schaffner, and Florian Speelman. Quantum homomorphic encryption for polynomial-sized circuits. *arXiv preprint arXiv:1603.09717*, 2016.

[FBS+14] KAG Fisher, A Broadbent, LK Shalm, Z Yan, J Lavoie, R Prevedel, T Jennewein, and KJ Resch. Quantum computing on encrypted data. *Nature communications*, 5, 2014.

[Fey82] Richard P Feynman. Simulating physics with computers. *International journal of theoretical physics*, 21(6):467–488, 1982.

[Fil12] Maximilian Fillinger. Lattice based cryptography and fully homomorphic encryption. Master of Logic Project, 2012. `http://homepages.cwi.nl/~schaffne/courses/reports/MaxFillinger_FHE_2012.pdf`.

[GC99] Daniel Gottesman and Isaac L. Chuang. Quantum Teleportation is a Universal Computational Primitive. *Nature*, 402:390–393, August 1999.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, volume 9, pages 169–178, 2009.

[GGH+13] Shelly Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Anant Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 40–49. IEEE, 2013.

[GHS15] Tommaso Gagliardoni, Andreas Hülsing, and Christian Schaffner. Semantic security and indistinguishability in the quantum world. *arXiv preprint arXiv:1504.05255*, 2015.

[GHV10] Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. A simple BGN-type cryptosystem from LWE. In *Advances in Cryptology–EUROCRYPT 2010*, pages 506–522. Springer, 2010.

[GKP+13a] Shafi Goldwasser, Yael Kalai, Raluca Ada Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. Reusable garbled circuits and succinct functional encryption. In *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing*, STOC '13, pages 555–564, New York, NY, USA, 2013. ACM.

[GKP+13b] Shafi Goldwasser, Yael Tauman Kalai, Raluca Ada Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. How to run Turing machines on encrypted data. In *Advances in Cryptology–CRYPTO 2013*, pages 536–553. Springer, 2013.

[Gle05] Ian Glendinning. The Bloch sphere, 2005. Presentation at Austrian Research Centers Seibersdorf, Tech Gate Vienna, Austria.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of computer and system sciences*, 28(2):270–299, 1984.

[Got98] Daniel Gottesman. The Heisenberg representation of quantum computers. *arXiv preprint arXiv:quant-ph/9807006*, 1998.

[GVW13] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 545–554, New York, NY, USA, 2013. ACM.

[IP07] Yuval Ishai and Anat Paskin. *Theory of Cryptography: 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007. Proceedings*, chapter Evaluating Branching Programs on Encrypted Data, pages 575–594. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[KO97] Eyal Kushilevitz and Rafail Ostrovsky. Replication is not needed: Single database, computationally-private information retrieval. In *FOCS*, page 364. IEEE, 1997.

[Lia13] Min Liang. Symmetric quantum fully homomorphic encryption with perfect security. *Quantum Information Processing*, 12(12):3675–3687, December 2013.

[Lia15] Min Liang. Quantum fully homomorphic encryption scheme based on universal quantum circuit. *Quantum Information Processing*, 14(8):2749–2759, 2015.

[MTY11] Tal Malkin, Isamu Teranishi, and Moti Yung. Key dependent message security: recent results and applications. In *Proceedings of the first ACM conference on Data and application security and privacy*, pages 3–12. ACM, 2011.

[NC00] Michael Nielsen and Isaac Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[NLV11] Michael Naehrig, Kristin Lauter, and Vinod Vaikuntanathan. Can homomorphic encryption be practical? In *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*, pages 113–124. ACM, 2011.

[OM08] Maris Ozols and Laura Mancinska. Generalized Bloch vector and the eigenvalues of a density matrix, 2008. `http://home.lu.lv/~sd20008/papers/essays.html`.

[OTF15] Yingkai Ouyang, Si-Hui Tan, and Joseph Fitzsimons. Quantum homomorphic encryption from quantum codes. *arXiv preprint arXiv:1508.00938*, 2015.

[Ozo08] Maris Ozols. Clifford group, 2008. `http://home.lu.lv/~sd20008/papers/essays.html`.

[Pai99] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in cryptology – EUROCRYPT99*, pages 223–238. Springer, 1999.

[RAD78] Ronald L Rivest, Len Adleman, and Michael L Dertouzos. On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180, 1978.

[RFG12] Peter P Rohde, Joseph F Fitzsimons, and Alexei Gilchrist. Quantum walks with encrypted data. *Physical review letters*, 109(15):150501, 2012.

[RSA78] Ronald L Rivest, Adi Shamir, and Len Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[SB08] Dan Shepherd and Michael J Bremner. Instantaneous quantum computation. *arXiv preprint arXiv:0809.0847*, 2008.

[Spe15] Florian Speelman. Instantaneous non-local computation of low T-depth quantum circuits. *arXiv preprint arXiv:1511.02839*, 2015.

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: Deniable encryption, and more. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14, pages 475–484, New York, NY, USA, 2014. ACM.

[SYY99] Tomas Sander, Adam Young, and Moti Yung. Non-interactive cryptocomputing for NC1. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 554–566. IEEE, 1999.

[TKO+14] Si-Hui Tan, Joshua A Kettlewell, Yingkai Ouyang, Lin Chen, and Joseph F Fitzsimons. A quantum approach to fully homomorphic encryption. *arXiv preprint arXiv:1411.5254*, 2014.

[Vai11] Vinod Vaikuntanathan. Computing blindfolded: New developments in fully homomorphic encryption. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 5–16. IEEE, 2011.

[VDGHV10] Marten Van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In *Advances in cryptology–EUROCRYPT 2010*, pages 24–43. Springer, 2010.

[VFPR14] Dunjko Vedran, Joseph F Fitzsimons, Christopher Portmann, and Renato Renner. Composable security of delegated quantum computation. In *Advances in Cryptology–ASIACRYPT 2014*, pages 406–425. Springer, 2014.

[Wol01] Ronald de Wolf. *Quantum Computing and Communication Complexity.* PhD thesis, University of Amsterdam, 2001.

[YPDF14] Li Yu, Carlos A. Pérez-Delgado, and Joseph F. Fitzsimons. Limitations on information-theoretically-secure quantum homomorphic encryption. *Phys. Rev. A*, 90:050303, Nov 2014.

[ZZHE93] M Zukowski, A Zeilinger, MA Horne, and AK Ekert. " event-ready-detectors" Bell experiment via entanglement swapping. *Physical Review Letters*, 71(26):4287–4290, 1993.

APPENDIX A

KEY-UPDATE RULES

A.1 Applying Clifford-group gates

When applying a Clifford gate C to a state of the form $\mathsf{X}^a\mathsf{Z}^b|\psi\rangle$, we can rewrite the resulting state as $\mathsf{X}^{a'}\mathsf{Z}^{b'}\mathsf{C}|\psi\rangle$ for some updated key values $a', b' \in \{0, 1\}$. In this appendix, we list the key-update rules when applying the generators of the Clifford group to a state that is encrypted with the quantum one-time pad. These rules can be found in many places in the literature (or can be easily calculated by hand using the equalities in Section 2.4). See also e.g. [BJ15, Appendix C].

The X and Z gates do not affect the keys to the quantum one-time pad. For the other gates, after applying them to the ith wire of a quantum state that has one-time pad keys a_i and b_i, we update the keys as

$$\mathsf{P}_i : (a_i, b_i) \to (a_i, a_i \oplus b_i)$$

and

$$\mathsf{H}_i : (a_i, b_i) \to (b_i, a_i)\,.$$

For the two-qubit CNOT gate applied on control wire i, with target j, we update the corresponding keys as

$$\mathsf{CNOT}_{ij} : (a_i, b_i; a_j, b_j) \to (a_i, b_i \oplus b_j; a_i \oplus a_j, b_j)\,.$$

A.2 Using a conditional computation gadget

This appendix specifies the algorithm $\mathsf{ComputeKeys}_f(g_{f,\mathsf{C}}(x), y, r, r')$ from Theorem 4.5(iii), where $g_{f,\mathsf{C}}$ is the classical information accompanying the gadget. This algorithm is called in $\mathsf{TP.Eval}$ after using a gadget from the evaluation key to remove a phase error.

After performing $\mathsf{UseGadget}_f(\gamma_r(g_{\mathsf{C},f}(x)), y, |\psi\rangle)$ for some input state[1] $|\psi\rangle$, the output state is of the form $\mathsf{X}^{a_2}\mathsf{Z}^{b_2}\mathsf{C}^{f(x,y)}\mathsf{X}^{a_1}\mathsf{Z}^{b_1}|\psi\rangle$ (see Equation 4.1), where the values $a_1, a_2, b_1, b_2 \in \{0, 1\}$ depend on the path the qubit has taken through the gadget. We sketch how these values can be computed from the gate C, the values x and y, and the measurement outcomes from Alice (contained in r) and Bob (who performs $\mathsf{UseGadget}_f$).

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

We first sketch the algorithm to find a_1, b_1, the keys that are applied to the state $|\psi\rangle$ *before* the Clifford gate C is applied. The algorithm to find a_2, b_2 is similar. Let $\mathbf{a}, \mathbf{b}$ be variables that hold the current X and Z keys respectively, at every step of the algorithm. Let **loc** be the variable that contains the current location of the qubit, with possible locations 0 to $2m$. That is, we view the current state as being $\mathsf{X}^{\mathbf{a}}\mathsf{Z}^{\mathbf{b}}|\psi\rangle$ at location **loc**. For every step we update the location depending on which qubits Bob measures and on the list $\{(s_1, t_1), (s_2, t_2), \ldots, (s_m, t_m)\}$ contained in $g_{f,\mathsf{C}}(x)$, and update the keys depending on the corresponding measurement outcomes from both Alice and Bob, as follows:

Initialize: $\mathbf{a} \leftarrow 0$, $\mathbf{b} \leftarrow 0$, $\mathbf{loc} \leftarrow 0$.
While $\mathbf{loc} \leq m$, do:

1. Let $\mathbf{loc}'$ be the position of the qubit that Bob has connected the qubit at position **loc** to via a Bell measurement. Let the outcome of this Bell measurement be (a', b'). Effectively, these outcomes change the current state to

$$\mathsf{X}^{a'}\mathsf{Z}^{b'}\mathsf{X}^{\mathbf{a}}\mathsf{Z}^{\mathbf{b}}|\psi\rangle,$$

   therefore we update $\mathbf{a} \leftarrow \mathbf{a} \oplus a'$ and $\mathbf{b} \leftarrow \mathbf{b} \oplus b'$.

2. Find the pair $(s_i, t_i) \in \{(s_1, t_1), (s_2, t_2), \ldots, (s_m, t_m)\}$ such that $s_i \equiv \mathbf{loc}'$. The teleportation of the qubit through this pair effectively applies $\mathsf{X}^{r_x[i]}\mathsf{Z}^{r_z[i]}$ to the state. Update $\mathbf{a} \leftarrow \mathbf{a} \oplus r_x[i]$ and $\mathbf{b} \leftarrow \mathbf{b} \oplus r_z[i]$. Update $\mathbf{loc} \leftarrow t_i$.

Output **a**, **b**, **loc**.

The output of this algorithm corresponds to the values a_1, b_1 and the location *loc* where (possibly) the Clifford gate C is applied. To compute a_2, b_2, initialize **loc** to *loc*, and continue until **loc** holds the known output position for the conditional computation gadget.

After the computation of a_1, b_1, a_2, b_2, it is straightforward to compute the output of $\mathsf{ComputeKeys}_f$ by looking up in the list c (contained inside $g_{f,\mathsf{C}}(x)$) whether or not the gate C was applied at *loc*, and commute the Paulis according to the rules in Appendix A.1. For example, if $\mathsf{C} = \mathsf{P}^\dagger$ (as in the scheme TP), then the output (a, b) of $\mathsf{ComputeKeys}_f$ is $(a_1 \oplus a_2, a_1 \cdot c[i] \oplus b_1 \oplus b_2)$, where i is the index such that $t_i = loc$. Note that at this point, multiplication is needed to update the keys.

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

For the key updates during the evaluation phase of TP, all the above computations are carried out homomorphically. There is already a quantum one-time pad present on the input state to the conditional computation gadget, which has to be commuted with the (possible) $\mathsf{P}^\dagger$ gate like we did with the keys a_2, b_2 in the previous paragraph. After the key updates, all temporary variables can be discarded, and only the new keys are needed for continuing the protocol.

The $\mathsf{ComputeKeys}_f$ algorithm for updating the one-time-pad keys after the use of a single conditional computation gadget runs in time polynomial in m, which is polynomial in the security security parameter κ. The time complexity of $\mathsf{ComputeKeys}_f$ can be established by observing that computing the values a_1 and b_1 (and similarly computing a_2 and b_2) requires at most $m/5$ iterations of the while loop, and each iteration requires $O(m \log m)$ time (it involves addition modulo 2, looking up values in lists of length $2m$, and comparing these values to the value stored in **loc**). Commuting the resulting keys with the possible Clifford operation at location *loc* can also be done in time linear in m. Overall, the time complexity of $\mathsf{ComputeKeys}_f$ is a low-degree polynomial in m.