Télécom Paris
Perimeter Institute for Theoretical Physics

Master's thesis

Higher-Order Delegated Quantum Computation

Thomas VINET
Supervisor: Hlér Kristjánsson

15th August 2024

Abstract

In the near future, where only a small number of companies and institutions will have access to large-scale quantum computers, it is essential that clients are able to delegate their computations in a secure way, without their data being accessible by the server. The field of blind quantum computation has emerged in recent years to address this issue, however, the majority of work on this topic has so far been restricted to the secure computation of sequences of quantum gates acting on a quantum state. Yet, a client capable of performing quantum subroutines may want to conceal not only their quantum states but also the subroutines they perform themselves. In this work, we aim to introduce a framework of higher-order delegated quantum computation, where a client performs a quantum subroutine (for example a unitary gate), which is transformed in a functional way by a server with more powerful quantum capabilities (described by a higher-order transformation), without the server learning about the details of the subroutine performed. The major contribution of this thesis is to introduce a protocol for when the server has the knowledge of the implementation of the higher-order transformation. We show that this requires information leakage for both parties and that we cannot achieve security in the same manner as for other blind quantum computation protocols. However, we try to find ways to mitigate the importance of those leaks and how much an adversary is able to learn through multiple rounds of communication.

Acknowledgments and context of work

I would first like to thank Hlér Kristjánsson for doing this thesis under his supervision, for his availability and his comments on this thesis. I would also like to thank Anne Broadbent for her help on the cryptographic aspects and her time to discuss the project, and Isaac David Smith for his help on the construction of several protocols and the security definitions. I also thank Robert Spekkens for providing funding from the PI Quantum Causal Inference Initiative. Finally, I would like to thank my parents, my sister and my girlfriend for their support during this internship.

This work was made during a 6-month internship at Perimeter Institute, under the supervision of Hlér Kristjánsson. The internship was the end of study internship part of the cursus of the engineering diploma of Telecom Paris.

Contents

Chapter I

Introduction

Building quantum computers is a hard task, and although the technology and the results obtained progress more and more, they still come with constraints. Therefore, one could consider that when quantum computers will become commercially available, there is going to be a disparity in the computing power distribution between the different parties that have access to quantum computers. A vaste majority of quantum computers will come with low computational power (restriction on the number of qubits, allowed gates, etc.), while some computers will be able to do much more powerful computations. However, by allowing communication between these parties, smaller quantum computers could take advantage of those with bigger computational power. This is called **delegated quantum computation**, which is a computation scheme that allows a client *Alice* to do some computation, with the help of a server *Bob*; if Alice wishes to keep her data private, we call it **blind quantum computation**. This was first introduced in Childs' paper *Secure assisted quantum computation* [1], which allows a client with very limited quantum capabilities to compute any gate by communicating with a server, with each communication round being encrypted with a one-time pad. A more developed version, *Universal blind quantum computation (UBQC)* was introduced in Ref. [2], which uses measurement-based quantum computation (MBQC) [3], and also allows to hide the computation itself from the server. Alternative algorithms have also been developed, like the ABE protocol that modifies Child's protocol so that the client needs less quantum memory [4], or the alternative to the use of MBQC for blind quantum computation by Morimae and Fuji, where the client now does measurement only instead of state preparation [5]. Ref. [6] contains a summary of the main existing BQC protocols.

Such protocols have also evolved in terms of security definition. Indeed, the first protocols were defined to be secure as "blind while leaking some quantity $L(X)$". Namely, most BQC protocols always leak some information $L(X)$ (which can be for example some bound on the depth of the computation) on input X, but the classical and quantum information that the adversary gets is independent of X given $L(X)$ [2]. However, this does not fully define the security criteria when in presence of a larger system, or the composability of protocols. In particular, it has been proven for some protocols satisfying this criteria that they can still leak extra information to the adversary [7]. Ref. [8] has introduced a stronger definition for composability of blind quantum computation protocols using the Abstract Cryptography framework [9]. The security of the UBQC protocol has been reproven since using this framework [8, 10].

Although these protocols differ in various ways (mainly in the requirements on what the client is able to do), they all delegate quantum computation of a unitary transformation circuit. However, we could also consider a situation where Alice would like to do higher-order computation, i.e. transformations of quantum gates. Such transformations are described by quantum supermaps, which transform quantum channels to quantum channels [1] [19–21]. In particular, there has been a lot of development in this area recently to find supermaps that achieve functional transformations given multiple calls to a black-box unitary channel. An example is the inverse function, i.e. that

[1] In this thesis we only consider supermaps that can be written as a quantum circuit with open slots that call the input quantum channels in a well-defined order. Some other works have considered more exotic supermaps which cannot be writtten in this way, and query the input channels in an indefinite causal order [11–14], however it is still a matter of debate whether indefinite causal order supermaps can be implemented in standard physics experiments [15–18].

maps any unitary to its inverse; this is an important topic, both from a foundational aspect and a practical viewpoint, as it represents negative time-evolution [22–32]. Other work has been done on finding supermaps to achieve the transposition, conjugation and the controlled version of a unitary channel [28, 33–38]. However, as there is no way of assuring the existence of supermaps for any function on unitaries, and some no-go theorems have been proven for some functional transformation [32, 37, 39], having the knowledge of how to implement such a supermap could be powerful, and companies could want to keep the knowledge to themselves. No framework on this type of two-way blind-quantum computation has been considered as of today.

We therefore aim to develop a framework for this type of computation, which we call higher-order delegated quantum computation. The situation is the following: two parties communicate with each other. The client, Alice, wishes to compute a function of a black-box unitary channel that she has, i.e. she does not know anything about it. The server, Bob, is here to communicate with Alice to help her achieve this computation. One of the two parties has the knowledge on how to achieve this function, written as a quantum supermap. If Alice has the knowledge of the supermap, we show that she can easily do the computation securely. The most interesting case is when Bob has the knowledge. In this case, both parties have some hidden data that they want to hide from the other party. We show that we cannot define an ideal resource without leaks for both parties, and that it puts a constraint on the security of our protocol. However, we show that we can mitigate the importance of those leaks, by adding some verification rounds or some randomization.

The structure of this thesis is the following. In Chapter II, we introduce all quantum notations, and the different topics addressed. Section II.3 introduces the cryptographic framework used in BQC protocols, and Section II.4, Section II.5 respectively present the state of the art in blind quantum computation and higher-order quantum computation. Chapter III presents the results of this thesis, when either party has the knowledge. Finally, Chapter IV summarizes the results and discusses potential future improvements of the higher-order transformation.

Chapter II

Preliminaries

II.1 Notations

For any quantum system A, the associated space is a Hilbert space denoted $\mathcal{H}_A$. All Hilbert spaces considered here will be of finite dimension $d_A \coloneqq \dim \mathcal{H}_A$. We denote a matrix M that acts on system A and maps it to system B as $M \in \mathcal{L}(\mathcal{H}_A, \mathcal{H}_B)$. If $A = B$, then we use the notation $M \in \mathcal{L}(\mathcal{H}_A)$. For $0 \leq i \leq d_A - 1$, we use the braket notation $|i\rangle$ which denotes a vector of size d_A in the computational basis, with 0 everywhere except for the i-th position where there is 1. The identity matrix on a quantum system A is written $\mathbb{1}_A$. When only the dimension d is precised, we can write it as $\mathbb{1}_d$; we can omit the dimension when it is clear, and write only $\mathbb{1}$. For a vector or matrix X, we denote $X^\dagger$ as the Hermitian adjoint of X. A unitary matrix $U \in \mathcal{L}(\mathcal{H}_A)$ is a matrix that satisfies $U^\dagger U = UU^\dagger = \mathbb{1}_A$; we denote $\mathbb{U}_d$ as the unitary matrices of dimension d, and $\mathbb{U}(\mathcal{H}_A)$ the unitary matrices in $\mathcal{L}(\mathcal{H}_A)$, with the special case of the unit group $\mathbb{U}_1 \coloneqq \mathbb{U}$. For any vector $|\psi\rangle \in \mathcal{H}_A$, we denote $\langle\psi| \coloneqq (|\psi\rangle)^\dagger$. The inner product is then denoted $\langle\psi|\phi\rangle$.

II.2 Quantum computing

We describe here the behaviour of quantum computing by comparing it with classical computing in three main subjects: allowed states, allowed transformations and measurements.

Classical computing consists of manipulating bits on a wire. Each bit represents the smallest amount of information possible, and is either 0 or 1. Any wire can store this representation by either having some current passing through it or not. One can put n wires side-by-side and have an n-bit state $m \in \{0,1\}^n$. Any transformation is allowed from an n-bit state to an n'-bit state, even with $n \neq n'$. One can always look up the state to measure its value at any given point of an algorithm. Quantum computing is described with quantum mechanics, and therefore has different paradigms that change the way computing works.

II.2.1 Quantum states

In quantum mechanics, the usual states 0 and 1 are written respectively as $|0\rangle$ and $|1\rangle$ and are called qubits. However, one of the fundamental principle of quantum mechanics is quantum superposition, which allows linear combinations of states. Therefore, we call pure states any state written as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where $|\psi\rangle$ is a unit vector of $\mathbb{C}^2$, and thus must satisfy $|\alpha|^2 + |\beta|^2 = 1$. The space containing pure qubit states is $\mathcal{H}_A \cong \mathbb{C}^2$, and we denote $|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}$ and $|1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$. Typical states in a superposition are $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$, $|-\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}}$. More generally, let consider a quantum system A, with a Hilbert space $\mathcal{H}_A$. Then any pure quantum state can be written as $|\psi\rangle_A = \sum_{i=0}^{d-1} c_i |i\rangle$, with $\sum_{i=0}^{d-1} |c_i|^2 = 1$, and $d = \dim \mathcal{H}_A$, $\mathcal{H}_A \cong \mathbb{C}^d$, and are called qudits.

However, these are not the most general states that exist. We can also consider a probabilistic mixture of states prepared. For example, we can consider a situation where one prepares the state $|\psi\rangle$ with probability $\frac{1}{4}$, and the state $|\phi\rangle$ with probability $\frac{3}{4}$. Then a way of representing this state

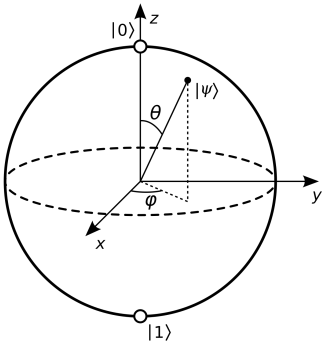

Figure II.1: The Bloch sphere, which represent any qubit quantum state. Pure quantum states lie on the sphere, while mixed state are inside the ball. (Credits: Wikipedia)

is by a density matrix $\rho = \frac{1}{4}|\psi\rangle\langle\psi| + \frac{3}{4}|\phi\rangle\langle\phi| \in \mathcal{L}(\mathcal{H}_A)$. The pure quantum states defined above can also be written in this fashion; if we have the pure state $|\psi\rangle$, then it means we have it prepared with probability 1, so the density matrix representing this pure state is $\rho = |\psi\rangle\langle\psi|$. This matrix is hermitian, positive semi-definite, and of trace 1. All states satisfy this condition.

Definition II.1. *Given a quantum system A with Hilbert space $\mathcal{H}_A$, any quantum state on this system is a matrix $\rho \in \mathcal{L}(\mathcal{H}_A)$ which satisfies $\rho^\dagger = \rho$, $\rho \geq 0$, $Tr[\rho] = 1$. Any matrix can be written as $\rho = \sum_{i=1}^m p_i |\psi_i\rangle\langle\psi_i|$, where $|\psi_i\rangle \in \mathcal{H}_A$ and where $\sum_{i=1}^m p_i = 1$. If $\rho = |\psi\rangle\langle\psi|$, then it is a pure state, else we call it a mixed state.*

As we work with finite dimensional Hilbert space, then any space can be decomposed into a given basis. There are two common basis that are used in quantum computing: the computational basis $\{|0\rangle, |1\rangle\}$, and the Hadamard basis $\{|+\rangle, |-\rangle\}$.

An easy way to represent qubit states is through the Bloch sphere. One can verify that any pure qubit can be written as $|\psi\rangle = \cos(\theta/2)|0\rangle + e^{i\phi}\sin(\theta/2)|1\rangle$, with $\theta \in [0, \pi]$, $\phi \in [0, 2\pi)$. These variables describe a polar representation, and can thus be shown on a sphere, as shown in Figure II.1. For mixed states, any mixed state can be written as $\rho = \frac{\mathbb{1} + r_x X + r_y Y + r_z Z}{2}$, where $r_x, r_y, r_z \in \mathbb{R}^3, r_x^2 + r_y^2 + r_z^2 \leq 1$. Then ρ can be placed on the sphere with coordinates (r_x, r_y, r_z). If we take $r_x = r_y = r_z = 0$, then $\rho = \mathbb{1}$ is placed at the center of the sphere, and called the maximally mixed state.

Given two quantum systems A, B, how do we represent their joint system ? Given their respective Hilbert space $\mathcal{H}_A, \mathcal{H}_B$, then the composite system is represented by the Hilbert space $\mathcal{H}_A \otimes \mathcal{H}_B$. Given $\{|i\rangle_A\}, \{|j\rangle_B\}$ the basis of $\mathcal{H}_A, \mathcal{H}_B$, then any pure state on this composite space can be decomposed in the basis $\{|i\rangle_A \otimes |j\rangle_B\}$. A particular type of state is called a product state when it can be written as $|\psi\rangle_A \otimes |\phi\rangle_B$; if not, we call it an entangled state. Not all states are product states: one example is $|\Psi^+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$, which is one of the four Bell states. One can then define mixed states as before on this composite space $\mathcal{H}_A \otimes \mathcal{H}_B$. We can also define a partial trace on this composite space $\mathrm{Tr}_{\mathcal{H}_B} : \mathcal{H}_A \otimes \mathcal{H}_B \to \mathcal{H}_A$ as $\mathrm{Tr}_{\mathcal{H}_B}[\rho_A \otimes \sigma_B] = Tr[\sigma_B]\rho_A$.

II.2.2 Unitary transformations

Transforming a quantum state is more restrictive that in classical computing. Indeed, the evolution of quantum states is given by the Schrödinger equation, in which the evolution operator U is a unitary operator, $UU^\dagger = U^\dagger U = \mathbb{1}$. This is required to preserve the norm of the state. This means that the input and output spaces must be of same dimension, and classical gates (like AND, OR, XOR, etc.) cannot be achieved directly. One set of unitary operators is called the Pauli matrices, which we denote $G = \{\sigma_i\}_{0 \leq i \leq 3}$, with the four matrices $\sigma_0 = \mathbb{1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $\sigma_1 = X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $\sigma_2 = Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$, $\sigma_3 = Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$. We can generalize this to n-qubit unitary matrices as $G_n = \{\otimes_{i=1}^n g_i : g_i \in G\} \subseteq \mathbb{U}_{2^n}$, and this is called the n-qubit Pauli group. Another matrix acting

on qubits is called the Hadamard gate $H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$. Finally, there are two matrices that will be useful later: $S = \sqrt{Z} = \begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix}$ and $T = \sqrt{S} = \begin{pmatrix}1 & 0\\ 0 & e^{i\pi/4}\end{pmatrix}$.

All of the matrices described above do not create entanglement, as they act qubit per qubit. One way to create entanglement is through a gate called CNOT, shown in Figure II.2. The gate is described by the following equation:

$$\text{CNOT} = |0\rangle\langle 0| \otimes \mathbb{1} + |1\rangle\langle 1| \otimes X \tag{II.1}$$

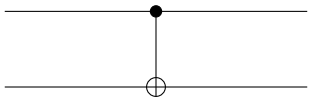

Figure II.2: The representation of the CNOT gate.

Given a pure state $|\psi\rangle$, the evolution of the state by applying U is represented as $U|\psi\rangle$. For a mixed state ρ, the evolution is given by $U\rho U^\dagger$.

II.2.3 Quantum measurement

Finally, how do we measure a quantum state ? Again, this is more complicated than in the classical case. In quantum mechanics, a measurement is probabilistic and modifies the output. A first approach to measurement is by using projective measurement. Given a quantum system A with an orthonormal basis $\{|i\rangle\}$, and a quantum state $|\psi\rangle$, then measuring this state will give as result one of the states of the basis, and will modify the state, i.e. all information about $|\psi\rangle$ will be destroyed. Each state is obtained with probability $\mathbb{P}(i) = |\langle i|\psi\rangle|^2$. Therefore, unless $|\psi\rangle$ is one state in the basis, the outcome will be probabilistic. For density matrices, the probability is described as $\mathbb{P}(i) = \text{Tr}[\rho|i\rangle\langle i|]$. The most general definition of measurements is called positive-operator-valued-measure (POVM).

Definition II.2. *Let $\mathcal{H}_A$ be an Hilbert space. We call a POVM a finite set $\{M_i\}$, where $M_i \in \mathcal{L}(\mathcal{H}_A)$, $M_i \geq 0$ and $\sum_i M_i = \mathbb{1}_A$.*

Given each measurement M_i associated with outcome i, the probability to obtain this outcome is still $\mathbb{P}(i) = \text{Tr}[\rho M_i]$. The update rule is more complicated. Given each M_i, which can be rewritten as $M_i = K_i^\dagger K_i$, the update rule is:

$$\rho \to \rho' = \frac{K_i \rho K_i^\dagger}{\text{Tr}[\rho M_i]} \tag{II.2}$$

II.2.4 Quantum circuits

We now have every part needed to represent a circuit. How are they represented, and how do they work? First, the three parts of quantum computing that we described above are written in the circuit form in Figure II.3. A wire represents a quantum register, and the computation direction is from left to right. To indicate the initialization of a quantum register in a certain quantum state (pure or mixed), we write it left of the register. Any unitary transformation is represented in the circuit form by a box with its name in it. Finally, the way to represent measurement is given by this special box. Most of the time, the measurement basis is precised in the description of the algorithm rather than in the circuit itself.

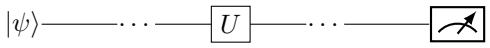

Figure II.3: The architecture of a quantum circuit and its main components. The three parts are respectively the initialization of a quantum register in the state $|\psi\rangle$, the computation of a quantum gate U on the current register, and the measurement of the register.

We will now present how to compute the result of a quantum circuit by showing a well known quantum algorithm: quantum teleportation, represented in Figure II.4. This protocol is a communication between Alice and Bob, and the states here denote which party has the qubit (A for Alice, B for Bob). Unless specified, a wire represents a qubit.

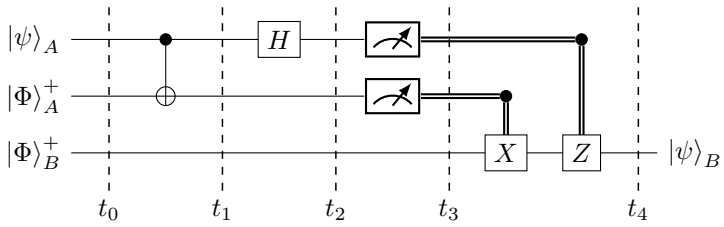

Figure II.4: The quantum circuit that implements quantum teleportation. Classical wires take the output of the measurement (0 or 1) and apply the gate if the result is 1, and apply $\mathbb{1}$ otherwise.

Alice possesses a qubit $|\psi\rangle$, and they both share half a pair of an entangled state $|\Phi\rangle_{AB}^+ = \frac{|00\rangle_{AB}+|11\rangle_{AB}}{\sqrt{2}}$. When wires are put aside, they represent a larger tensor system; so here the total system is $(\mathbb{C}^2)^{\otimes 3}$. Then, we can split the circuit into multiple parts to compute the evolution at each time, as if the information was propagating, as shown above. For example, at $t = t_1$, Alice applies a CNOT on her qubits, and Bob does nothing on his qubit (applies $\mathbb{1}$). The state obtained at t_1 is therefore, if we denote $|\psi\rangle_A = \alpha\,|0\rangle + \beta\,|1\rangle$:

$$(\mathrm{CNOT} \otimes \mathbb{1})\,|\psi\rangle_A \otimes |\Phi\rangle_{AB}^+ = \frac{1}{\sqrt{2}}(\alpha\,|0\rangle_A\,[|00\rangle + |11\rangle]_{AB} + \beta\,|1\rangle\,[|10\rangle + |01\rangle]_{AB}) \tag{II.3}$$

We can do the computation like this to proceed. At t_3, the measurement is done in the computational basis, which gives as a result either 0 or 1. Alice sends this result to Bob, that applies X and Z conditionally to the measurement outputs. Then one can check that the state obtained at the end is $|\psi\rangle$. This is known as quantum teleportation, because Bob has received Alice's input qubit by a quantum algorithm, without Alice giving him directly the state or the coefficients; and as long as they share this entangled state, they theoretically can be far from one another.

II.2.5 Quantum channels

The unitary transformations considered earlier are not the most general ones. Indeed, the most general considered are called quantum channels, which map quantum states to quanum states.

Definition II.3. *Let $\mathcal{C} \in \mathcal{L}(\mathcal{H}_A) \to \mathcal{L}(\mathcal{H}_B)$ be a linear map. Then it is a quantum channel if it is trace preserving and completely positive (CPTP):*

$$\forall \rho \in \mathcal{L}(\mathcal{H}_A),\ \mathrm{Tr}[\mathcal{C}(\rho)] = \mathrm{Tr}[\rho]$$

$$\forall \mathcal{H}_C,\ \rho_{AC} \in \mathcal{L}(\mathcal{H}_A \otimes \mathcal{H}_C) \text{ where } \rho_{\mathrm{AC}} \geq 0,\ (\mathcal{C} \otimes \mathcal{I}_{\mathcal{C}})(\rho_{\mathrm{AC}}) \geq 0,$$

where $\mathcal{I}_{\mathcal{C}}$ is the identity quantum channel, and $\mathcal{H}_C$ is any Hilbert space. We denote the set of quantum channels from $\mathcal{H}_A$ to $\mathcal{H}_B$ as $\mathrm{QChan}(\mathcal{H}_A \to \mathcal{H}_B)$.

Any unitary transformation falls back into this definition, as $\mathcal{C}_U(\rho) = U\rho U^\dagger$. However, we can also allow a channel that applies a probabilistic mixture of unitaries, as $\mathcal{C}(X) = \sum_{i=1}^m p_i U_i X U_i^\dagger$, where U_i are unitary matrices and $\sum_{i=1}^m p_i = 1$. Another channel that we will use later is a depolarizing channel, given by $\mathcal{C}_p(X) = (1-p)X + p\,\mathrm{Tr}[X]\mathbb{1}/d$, where $0 \leq p \leq 1$ and $d = \dim \mathcal{H}_A$. When $p = 1$, we obtain a fully depolarizing channel, which is a channel that sends every state to the maximally mixed state; this is the state with zero entropy, so a state with no information.

Any quantum channel $\mathcal{C} \in \mathcal{L}(\mathcal{H}_A) \to \mathcal{L}(\mathcal{H}_B)$ can be written as $\mathcal{C}(X) = \sum_{i=1}^m K_i X K_i^\dagger$, where $K_i \in \mathcal{L}(\mathcal{H}_A, \mathcal{H}_B)$ are called Kraus operators and satisfy the condition $\sum_{i=1}^m K_i^\dagger K_i = \mathbb{1}$. Conversely, any map with this decomposition and satisfying the condition are quantum channels. The decomposition is not unique, however there exists relations between two sets of Kraus operators achieving the same quantum channel; and one can find a set of Kraus operators from the quantum channel. Another way of seeing a quantum channel is through Choi matrices. For a quantum channel, the Choi matrix $C = \sum_{i,j} |i\rangle\langle j| \otimes \mathcal{C}(|i\rangle\langle j|) \in \mathcal{L}(\mathcal{H}_A \otimes \mathcal{H}_B)$ satisfies $C \geq 0$ and $\mathrm{Tr}_{\mathcal{H}_B}[C] = I_{\mathcal{H}_A}$. Any matrix C satisfying this condition describes a quantum channel, which can be recovered as $\mathcal{C}(X) = \mathrm{Tr}_{\mathcal{H}_A}[(X^T \otimes \mathbb{1}_{\mathcal{H}_B})C]$.

Let assume that we have a quantum channels $\mathcal{C}$ with two sets of Kraus operators, i.e. $\mathcal{C}(\rho) = \sum_{i=1}^n E_i \rho E_i^\dagger = \sum_{j=1}^m F_j \rho F_j^\dagger$. What can we say about those finite sets ? Ref. [40] gives us the following lemma:

Lemma II.1. *Let $\mathcal{E}, \mathcal{F}$ be two quantum channels, with their respective Kraus representations $\{E_i\}_{1\leq i\leq n}$, $\{F_j\}_{1\leq j\leq m}$. By adding some zero operators ($F_i = 0$ for example), we can take $m = n$. Then $\mathcal{E} = \mathcal{F}$ if and only if we have a $n \times n$ unitary matrix $U = (u_{i,j})_{1\leq i,j\leq n}$ such that:*

$$\forall i,\ E_i = \sum_{j=1}^n u_{i,j} F_j$$

We also prove here one result about the composition of quantum channels.

Lemma II.2. *Let $\mathcal{E}_i \in \mathrm{QChan}(\mathcal{H}_A \to \mathcal{H}_A)$ be a collection of quantum channels. Then if the composition $\circ_{i=1}^n \mathcal{E}_i$ is a unitary channel, then each $\mathcal{E}_i$ is a unitary channel.*

Proof. We prove this by induction. For $n = 2$, we write the channels with their Kraus operators:

$$\mathcal{E}(\rho) = \sum_i E_i \rho E_i^\dagger,\ \mathcal{F}(\rho) = \sum_j F_j \rho F_j^\dagger,\ (\mathcal{F}\circ\mathcal{E})(\rho) = \sum_{i,j} F_j E_i \rho (F_j E_i)^\dagger = A\rho A^\dagger \tag{II.4}$$

As both channels $\mathcal{F}\circ\mathcal{E}$ and $\mathcal{A}$ are equal by hypothesis, then we can use Lemma II.1:

$$\forall i,j,\ F_j E_i = \alpha_{i,j} A,\ \sum_{i,j} |\alpha_{i,j}|^2 = 1 \tag{II.5}$$

There exists i_0, j_0 such that $\alpha_{i_0,j_0} \neq 0$, and thus E_{i_0}, F_{j_0} are invertible. Therefore, if we use Equation II.5 with $i = i_0$, as E_{i_0} is invertible, and A too (as it is a unitary matrix):

$$\begin{aligned} &\forall j,\ F_j = \beta_j A^{-1} E_{i_0}, \beta_j = \alpha_{i_0,j} \\ &\mathcal{F}(\rho) = \sum_j |\beta_j|^2 A E_{i_0} \rho (A E_{i_0})^{-1} \coloneqq F\rho F^\dagger, F = \sqrt{\sum_j |\beta_j|^2} A E_{i_0} \end{aligned} \tag{II.6}$$

Therefore $\mathcal{F}$ is a unitary channel. The same reasoning can be made with $\mathcal{E}$ by taking $j = j_0$, which concludes the base case.

The induction case comes from the associativity of the induction: if the hypothesis holds for $n = k$, then we can write $\circ_{i=1}^{k+1}\mathcal{E}_i = (\circ_{i=1}^k \mathcal{E}_i)\circ\mathcal{E}_{k+1}$, and take $\mathcal{E} = \circ_{i=1}^k \mathcal{E}_i$, $\mathcal{F} = \mathcal{E}_{k+1}$. Then we proved earlier that $\mathcal{E}$ and $\mathcal{F}$ are unitary channels; and $\mathcal{E}$ is a composition of k channels, which are all unitary channels by the induction hypothesis, which concludes the proof.

□

II.3 Cryptography notations

II.3.1 Quantum one-time pad

In classical cryptography, given a message $m \in \{0,1\}^n$, we can achieve security by using a one-time pad. This means that we choose a key $k \in \{0,1\}^n$ to produce the cypher text $c = m \oplus k$, $\oplus$ representing the bitwise XOR operation. If the key is changed at each time and chosen randomly, the attacker has no way to decrypt the message from the cypher text. Moreover, applying once again the key gives back the message, so if the key is shared between two parties, the message can be shared securely.

A quantum version of this one-time pad has been introduced in Ref. [41]. Starting with a qubit ρ, Alice picks two random bits $j, k \in \{0,1\}$, then applies $X^j Z^k$. From anyone that has no information about j, k, the state looks like this:

$$\rho' = \frac{1}{4} \sum_{j,k=0}^{1} X^j Z^k \rho X^j Z^k = \frac{1}{2} \mathbb{1}_2 \tag{II.7}$$

This is a maximally mixed state, and therefore no information can be extracted from it, unless you know the key (j, k). This is widely used in order to share a quantum state securely with a non-trusted party. This can be extended to multiple qubits and qudits.

II.3.2 Cryptographic framework

To consider all the privacy settings, we need a clearer definition on what is security. We will use the Abstract Cryptography framework from Maurer and Renner, which is defined to allow composition of protocols [9]. Ref. [8] provides a more explicit use of this framework, especially in the case of two-party protocols and delegated quantum computation; it also provides another security proof fo the quantum one-time pad with this framework. Ref. [10] comes back to the UBQC protocol and uses this framework definition to provide security.

In this framework, three notions are used: resources, converters and distinguishers. A resource $\mathcal{R}$ is a system, which may be abstract, made with interfaces, expressed as a set $\mathcal{I} = \{A, B, E\}$. Each interface symbolizes an access for a party. A converter π_i is a system made with two interfaces. One is connected to the outside world, where the party can interact; the other is connected to the resource interface. A protocol $\pi = (\pi_i)$ is made of a collection of converters for each party. An example is shown in Figure II.5, where $\mathcal{I} = \{A, B\}$ (i.e. two parties, Alice and Bob). This protocol is represented by the concatenation of a resource and two converters, which can be written as $\pi_A \mathcal{R} \pi_B$. For any subset of interfaces $\mathcal{P} \subseteq \mathcal{I}$, we define $\pi_\mathcal{P}$ as the protocols for each party in $\mathcal{P}$, i.e. $\pi_\mathcal{P} = \{\pi_i\}_{i \in \mathcal{P}}$. Same can be done for any set of converters.

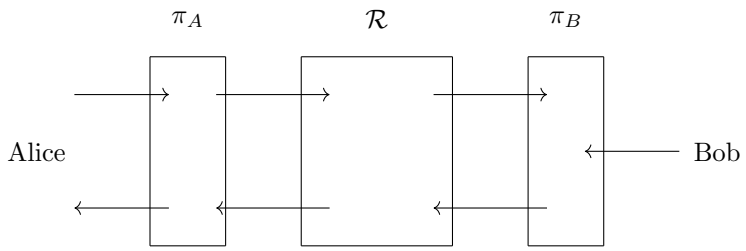

Figure II.5: An example of a protocol $\pi = (\pi_A, \pi_B)$ with an resource $\mathcal{R}$. Arrows represent communications between systems and parties.

Security is defined by distinguishability. Given an n-interface resource, $\mathcal{R}$, a distinguisher $\mathcal{D}$ is a converter with $n+1$ interfaces, where n are connected to the resource $\mathcal{R}$, and one outputs a bit. The distinguishing advantage, between two resources $\mathcal{R}, \mathcal{S}$, on a class of distinguishers $\mathfrak{D}$, represents a pseudo-metric, and is given by:

$$d(\mathcal{R},\mathcal{S}) = \max_{\mathcal{D}\in\mathfrak{D}}[\mathbb{P}(\mathcal{D}\mathcal{R}=1) - \mathbb{P}(\mathcal{D}\mathcal{S}=1)] \tag{II.8}$$

The bit output by the distinguisher is which resource the distinguisher thinks he is currently plugged onto. Therefore, the distinguisher advantage represents how much an adversary can distinguish between two resources. If this distance is small, this means that any party having access to one resource being either $\mathcal{R}$ or $\mathcal{S}$ can only guess with small probability which resource they are using. If $d(\mathcal{R},\mathcal{S}) \leq \epsilon$, we say that they are ϵ-close and write $\mathcal{R} \approx_\epsilon \mathcal{S}$. We can then define security, using this pseudo-metric d. A more general definition with filters is given in Ref. [8].

Definition II.4. *Let $\mathcal{R},\mathcal{S}$ be two resources with interface $\mathcal{I}$. We say that a protocol π securely constructs $\mathcal{S}$ out of $\mathcal{R}$ within ϵ, and write $\mathcal{R} \to_{\pi,\epsilon} \mathcal{S}$, if there exists simulators $\{\sigma_i\}_{i\in\mathcal{I}}$, such that:*

$$\forall \mathcal{P} \subseteq \mathcal{I}, d(\pi_\mathcal{P}\mathcal{R}, \sigma_{\mathcal{I}\setminus\mathcal{P}}\mathcal{S}) \leq \epsilon,$$

where $\pi_\mathcal{P} = \{\pi_i\}_{i\in\mathcal{P}}$ for any subset of interfaces $\mathcal{P} \subseteq \mathcal{I}$, and same for the simulators.

How do we use this definition ? The resource $\mathcal{R}$ defines the real world resource (also called concrete setting), while $\mathcal{S}$ defines the ideal world resource. While the inputs are directly fed into the ideal resource, there is a mapping for each party's interface, which is the protocol $\pi = \{\pi_i\}_{i\in\mathcal{I}}$. The ideal resource has the wanted behaviour, i.e. builds the good output depending on the inputs. This is shown in the definition above: if we take $\mathcal{P} = \mathcal{I}$, the inequality becomes $d(\pi_\mathcal{I}\mathcal{R},\mathcal{S}) \leq \epsilon$, which is the distance between the ideal resource and the concrete setting, and thus shows the correctness of the protocol. Now this also defines secureness. If a party i was to cheat, then we have $d(\pi_{\mathcal{I}\setminus\{i\}}\mathcal{R}, \sigma_i\mathcal{S}) \leq \epsilon$. This means that a party cannot distinguish, up to ϵ, between the concrete setting when it does not uses its protocol but cheats, and the ideal resource where there is a simulator at its interface. Therefore, any cheating behaviour can be simulated with the ideal resource, and the party learns nothing more than in the ideal scenario, up to ϵ. If we really want to have perfect security, the ideal resource must therefore leak nothing. This is the case when the ideal resource gives no output, except for the actual result of the computation to an honest party [8, 10]. Therefore, the simulator depends only on the party's input, and this explains this notion of security. Note that this also allows for multiple parties cheating at the same time and sharing their information.

This definition also satisfies sequential composability: if we have $\mathcal{R} \to_{\pi,\epsilon} \mathcal{S}$ and $\mathcal{S} \to_{\pi',\epsilon'} \mathcal{T}$, then $\mathcal{R} \to_{\pi'\circ\pi,\epsilon+\epsilon'} \mathcal{T}$. This can also be done for parallel composability.

II.4 Blind quantum computation

Blind quantum computation (BQC) consists of a secure delegated computation. It is designed to help a client with a low computational power to gain advantage from a server with a bigger computational power via communication. A client Alice has a quantum state $|\psi\rangle \in \mathcal{H}_A$, and wishes to do a computation $U \in \mathcal{L}(\mathcal{H}_A)$. Sadly, she does not have this gate ready to apply locally. But a server Bob with a bigger power could be able to implement any gate if asked to, and Alice could communicate with him to achieve this computation. However, Alice wishes to keep her input data private and hidden from the server; in some cases, she also wishes to hide the computation she is doing from the server. We present here the first BQC protocol, and another BQC protocol that hides the computation and which will be useful later. Ref. [6] contains a good introduction to BQC and a wider list of protocols.

II.4.1 Secure assisted quantum computation

The first BQC protocol was introduced in a paper by Andrew Childs [1]. The idea is to use the universal gate set Clifford + T, which consists of the gates $\{H, \texttt{CNOT}, S\} + T$ [42]. A universal gate set is a finite set built such that we can approximate any gate as a sequence of gates picked into this set. Therefore, if Alice gives U to Bob, then he can decompose it as a sequence of gates in this set, and he only has to be able to implement those 4 gates. However, Alice still needs to encode her state before sending it to Bob to achieve security. We showed in Section II.3.1 that a

way of encoding a qubit is to apply a one time-pad X^iZ^j, with $i, j \in \{0, 1\}$ two randomly chosen bits. The idea behind this choice of set is that the behaviour of the one-time pad with those gates is well defined. For example, we have the relation $XHZ = ZHX = H$, so if Alice encodes with X^iZ^j, then she needs to decode with X^jZ^i. This can be represented in Figure II.6. We can also build an encoding and decoding scheme for each other gate. Therefore, with many rounds, Alice and Bob can communicate to build U, while keeping Alice's input fully secure.

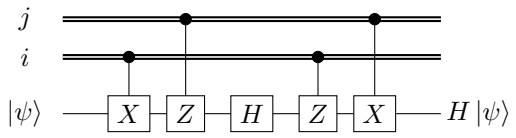

Figure II.6: The encryption and decryption scheme for the H gate.

II.4.2 Universal blind quantum computation

We now present another BQC protocol that also hides the computation achieved U [2]. This uses Measurement-Based Quantum Computation [3]. This computing scheme resides in three steps:

- State preparation: The user prepares multiples qubits in the $|+\rangle$ state, then entangles them, following a certain pattern. This creates a brickwork state [2]; the choice of the graph state is fixed beforehand.
- Measurement: The user measures each qubit in the X-Y plane with an angle θ, so measures in the basis $\{\frac{|0\rangle \pm e^{i\theta}|1\rangle}{\sqrt{2}}\}$.
- Correcting the outputs: After all the measurements have been done, the state is reduced to a smaller number of qubits, which is the state that is supposed to be prepared. However, due to the randomness of the measurement, the user needs to correct the outputs, by applying X or Z, depending on all measurement outcomes.

How are the measurement angles chosen during the protocol? As there exists an equivalence between MBQC and the ciruit model, each circuit (and therefore each gate) can be represented in the MBQC fashion, and so we can find angles and a number of qubits that will build this computation. However, the measurement angles also depend on the previous results. Figure II.7 provides an example on how one measurement propagates to the next state, and how the next measurement angle can be updated, with $R_z(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}$ the rotation around Z-axis. In particular, creating entanglement between two states and measuring the first one in a certain basis propagates the state to the second one, while adding some randomness that comes from the measurement angle and output.

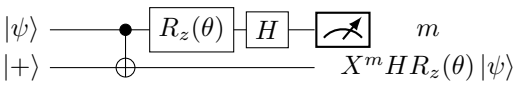

Figure II.7: How the measurement propagates to the next states. The measurement is done in the computational basis.

This BQC protocol uses the above computation scheme to do the computation. The difference comes from the fact that the input states are randomized from Alice's side, and also the measurement angles; therefore Bob cannot learn anything by doing the computation. More precisely, the qubits are prepared in $\{\frac{|0\rangle + e^{i\frac{k\pi}{4}}|1\rangle}{\sqrt{2}}\}_{0 \leq k \leq 7}$, and then the measurement angle depends on this angles, the angle from the MBQC parameter, and some randomized parameter, such that the classical and quantum information is completely randomized and independent. Therefore, every classical data sent to Bob (in particular the measurement angles) is always one-time padded, and at every time

during the protocol, each qubit that Bob has a one-time padded, and therefore Bob cannot learn anything from the protocol. As the protocol is written in the MBQC model, and Bob does not learn the measurement angles, then this also hides the current computation that Alice is doing.

II.5 Higher-order computation

All the transformations that we considered earlier (either unitary transformations or quantum channels) were transformations that map quantum states to quantum states. A branch of quantum computing called higher-order computation consists of considering transformations of higher-order objects, for example quantum channels. We call this type of transformation supermaps. As quantum channels need to preserve the property of quantum states, i.e. being CPTP, supermaps need to preserve the properties of quantum channels. Quantum supermaps with one slot were first introduced in Ref. [43], with a decomposition as a concatenation of isometries. Generalizations to quantum supermaps have also been introduced by using their Choi operator which satisfy some properties [21, 44]; we here use another way of defining them as we did with quantum channels, by defining how they keep the properties of the objects they map [20].

Definition II.5. *A supermap $\mathcal{S}$ with M slots is a M-linear map $\mathcal{S} \in (\otimes_{i=1}^{M}[\mathcal{L}(\mathcal{H}_{A_i}) \to \mathcal{L}(\mathcal{H}_{B_i})]) \to [\mathcal{L}(\mathcal{H}_{A_0}) \to \mathcal{L}(\mathcal{H}_{B_0})]$ such that for every CPTP input $Q \in (\otimes_{i=1}^{M}[\mathcal{L}(\mathcal{H}_{A_i}) \to \mathcal{L}(\mathcal{H}_{B_i})])$, then $\mathcal{S}(Q)$ is a CPTP map.*

There also exists a link between the map definition and its Choi operator [20].

II.5.1 Functional supermaps

Supermaps are widely used to try to implement functions acting on unitaries. For example, Ref. [22–32] introduced supermaps that are capable of reversing unknown unitaries of various sizes; other work has been done on supermaps for the conjugation, transposition and the controlled version of a given unitary [28, 33–38]. The supermaps that we consider here will be of this type, thus can be decomposed into a concatenation of unitary channels and the input unitary channels, with a memory channel, such that it achieves a given function on unitary channels. Definition II.6 describes the set of functions that we will consider here, and Figure II.8 shows how this type of decomposition can be described in a quantum circuits.

Definition II.6. *Let A be a quantum system with Hilbert space $\mathcal{H}_A$, $f : \mathbb{U}_d \to \mathbb{U}_d$ be a map, where $d = d_A$. We say that f is achievable through a supermap if there exists a supermap $\mathcal{S} : (\mathbb{U}_d)^{\otimes m} \to \mathbb{U}_d$ such that $S(U^{\otimes m}) = f(U)$. We denote the decomposition as $\mathcal{D}(f) := (V_0, \ldots, V_m)$.*

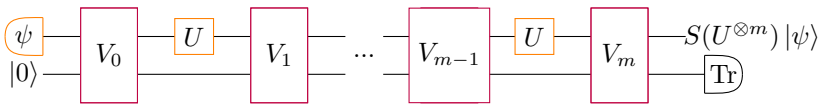

Figure II.8: A decomposition of a supermap such that $S(U^{\otimes m}) = f(U)$. The blocks in orange are the inputs, while the blocks in purple are unitaries that are fixed by the supermap. $|0\rangle$ represents the initialization of the ancillary qubits.

Chapter III

Results

We now try to develop a framework that groups both higher-order computation and BQC. We consider two parties, Alice and Bob. Alice has a quantum state $|\psi\rangle \in \mathcal{H}_A$ and black-box access to a unitary quantum gate $U \in \mathbb{U}(\mathcal{H}_A)$. The quantum gate U is unknown for Alice (and Bob); it may be a gate produced by some physical experimentation, or just given to Alice as a black box. We consider that Alice can use this gate as much as she wants, i.e. has unlimited accesses to it. Although she could try to learn the whole gate by doing quantum process tomography [45, 46], maybe this is too costly and she just wants to learn some property of the gate, or she wants to transform the unitary into another unitary by some function. The situation is therefore the following: Alice wants to apply a function $f : \mathbb{U}(\mathcal{H}_A) \to \mathbb{U}(\mathcal{H}_A)$ to her unknown quantum gate U. We see here why we require that U is unknown; if Alice knew it, then she could classically compute $f(U)$, and then use any BQC protocol to compute $f(U)$. As explained earlier, we will only consider functions achievable through supermaps as defined in Definition II.6, and we therefore have a decomposition $\mathcal{D}(f) = (V_0, \dots, V_m)$ that exists. However, finding it is not something easy to do. Therefore, Alice may not have the decomposition herself. In this case, privacy needs to be considered for both parties. We will consider this case in the following sections.

Definition III.1. *We call a higher-order delegated quantum computation (HODQC) protocol any communication protocol between two parties, named Alice and Bob, where:*

- *Alice has a quantum state $|\psi\rangle \in \mathcal{L}(\mathcal{H}_A)$, which may be known to her, and a quantum gate $U \in \mathbb{U}_{d_A}$ that she can physically apply, but U is unknown.*
- *Bob is a server with whom Alice can discuss.*
- *Alice wishes to compute $f(U)|\psi\rangle$, with $f \in \mathbb{U}_{d_A} \to \mathbb{U}_{d_A}$ without Bob learning about either $|\psi\rangle$ or U.*
- *One of the two parties has a decomposition $\mathcal{D}(f) \in (\mathcal{L}(\mathcal{H}_A \otimes \mathcal{H}_E) \to \mathcal{L}(\mathcal{H}_A \otimes \mathcal{H}_E))^{\times(m+1)}$, with $m = |\mathcal{D}(f)|$ which achieves f through a supermap, where $\mathcal{H}_E$ is the ancillary space, and wishes to keep $\mathcal{D}(f)$ private.*

III.1 Simple case when only the client has sensitive information

For now, let's focus on the case when Alice has $\mathcal{D}(f)$, which is always the case when f is linear. We still consider that she has a limited computational power, so she might not be able to implement the V_i by herself. However, she can communicate with a server Bob to achieve each V_i. If we look back at the decomposition described in Figure II.8, we can see an easy way to do HODQC: Alice communicates with Bob to achieve each V_i, and then when she gets back the state, she applies her unitary U on the upper qubits. This protocol is described in Protocol 0.

The correctness of this protocol comes directly as a concatenation of correct UBQC protocols, which results in a correct protocol [8].

Protocol 0 HODQC with Alice having $\mathcal{D}(f)$

1. **Alice's preparation**
   (a) Alice has an input state $|\psi_0\rangle = |\psi\rangle$ and a unitary U acting on the same space; she also has a supermap decomposition $\mathcal{D}(f) = (V_0, \dots, V_m)$.
   (b) She prepares the ancillary input $|\phi_0\rangle = |0\rangle \in \mathcal{L}(\mathcal{H}_E)$, then encrypts it as in the UBQC protocol before sending it to Bob.
2. **Communication rounds**
   For $i \in [0 \dots m]$
   (a) Alice communicates with Bob using the UBQC protocol to compute the unitary V_i, which acts on the state $|\psi_i\rangle$ sent by Alice and the ancillary state $|\phi_i\rangle$, taking into account the encryption of the ancillary state into the protocol
   (b) Bob keeps the ancillary state and sends the data state to Alice.
   (c) If $i < m$, Alice decrypts the data state, then applies U on it; we denote this state $|\phi_{i+1}\rangle$
3. **Final correction**
   (a) After the last communication round, Bob traces out the ancillary state $|\phi_m\rangle$
   (b) Alice decrypts her data state, resulting in the state $|\psi_m\rangle = f(U)\,|\psi\rangle$

Theorem III.1. *Protocol 0 is blind while leaking at most a bound on* $\dim \mathcal{H}_E \times \dim \mathcal{H}_A, |\mathcal{D}(f)|$, *and on the implementation size of each* V_i *in the UBQC protocol.*

Proof. Ref. [8, 10] have shown that a UBQC protocol was perfectly secure. Namely, the protocol constructs an ideal resource $\mathcal{S}^{\texttt{UBQC}}$ which has only inputs and no output on Bob's interface, apart from leaking a bound on the computation size, and cannot be distinguished from the concrete setting $\mathcal{R}^{\texttt{UBQC}}$, thus $\mathcal{R}^{\texttt{UBQC}} \to_{\pi,0} \mathcal{S}^{\texttt{UBQC}}$. As explained above, the Abstract Cryptography framework is designed to work well on sequential composition of protocols. Denoting π_i the UBQC protocol to construct V_i, the total protocol described above is $\pi = \pi_m \circ \pi_U \dots \pi_U \circ \pi_0$, where π_U is the protocol that applies $U \otimes \mathbb{1}_E$. It is clear that as π_U is a one-party protocol (Bob does not intervene) and is correct, then it provides perfect blindness. Therefore, the concatenation of all those protocols satisfy total blindness, i.e. no distinguishable advantage exists. The classical data shared (and thus the leak), are from each sub-protocol; thus they are a bound on the implementation size of V_i. Bob also gets a bound on the dimension of the computation, i.e. $\dim \mathcal{H}_E \times \dim \mathcal{H}_A$, and on the number of slots of the supermap, thus $|\mathcal{D}(f)|$. □

Note that if Bob has the knowledge of the decomposition $\mathcal{D}(f)$, but does not care about his security, or trusts Alice, then he could just send the decomposition to Alice through a secure classical channel, and then she could do the same as above.

III.2 Ideal resource and concrete setting

We now consider that Bob possesses the knowledge of the decomposition $\mathcal{D}(f)$. This is now a much harder task, because we are trying to achieve a two-party secure protocol, where each party has some information that they want to hide. Ref. [47] shows that in a special case of two-party protocols, it is impossible to achieve security. Although this is not the same situation as considered here, this gives a good insight about the difficulty of this task. We will first define, using the formalism from Ref. [9], our ideal resource and the general concrete setting used. Then we show the limitations to such a definition, and that there must exist some leakage for both parties, through two different protocols. We also consider the importance of that leakage in multiple consecutive

computations; if Bob has multiple accesses to the same resource, he may be able to recover the classical description of this quantum resource through quantum process tomography [45, 46].

In order to model correctness and security of a HODQC protocol, we need first to define the ideal resource. Note that this is different of the case where Bob had no knowledge, because now both parties have some inputs, and the ideal resource thus will change. Any HODQC protocol will leak some classical data, namely an upper bound on the dimension of U for Alice, and an upper bound on $|\mathcal{D}(f)|$ for Bob. As there is no way of preventing this leak, and it could be some data that is shared between the two parties before they do the actual computation, we can omit it in the description of the ideal resource. The ideal resource $\mathcal{S}^{\texttt{HODQC}}$ takes as input from Alice an input state $|\psi\rangle$, and m copies of U, and takes from Bob the supermap decomposition $V_0, \ldots, V_m$. Note that as Bob knows each V_i, depending on the protocol, this input may be classical. Alice gets as output the result of the computation, i.e. $f(U)|\psi\rangle$. This is shown in Figure III.1.

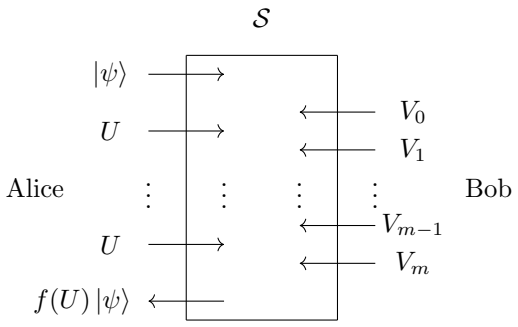

Figure III.1: The ideal resource wished for in a HODQC protocol.

Although we don't know how $\mathcal{S}^{\texttt{HODQC}}$ is defined, the concrete setting (the real resource $\mathcal{R}$) seems a clear choice: Alice and Bob only need to communicate, through classical and quantum channels. Therefore, $\mathcal{R}$ is a two party communication channel.

Definition III.2. *An ϵ-correct and secure HODQC protocol π is a protocol such that it constructs securely $\mathcal{S}^{HODQC}$ out of the concrete setting $\mathcal{R}$. Explicitely, this means we have simulators σ_A, σ_B such that:*

$$d(\pi_A\pi_B\mathcal{R}, \mathcal{S}^{HODQC}) \leq \epsilon$$
$$d(\pi_A\mathcal{R}, \sigma_B\mathcal{S}^{HODQC}) \leq \epsilon$$
$$d(\pi_B\mathcal{R}, \sigma_A\mathcal{S}^{HODQC}) \leq \epsilon$$

This definition (and the three distance inequalities) mean the following: the protocol is ϵ-correct, and if Alice or Bob were to cheat, they cannot distinguish between the ideal and concrete setting up to ϵ. If both parties were to cheat, then this is a dull case, and we do not need to consider it. Moreover, as the ideal resource $\mathcal{S}^{\texttt{HODQC}}$ has no input from both parties during the protocol, this would mean that no party would learn anything more than they already know.

However, this definition is sadly too strong. We show in the following theorem that Bob cannot encrypt his matrices V_i, and needs to send them uncrypted to Alice.

Theorem III.2. *No protocol can achieve securely the ideal resource $\mathcal{S}^{HODQC}$. Namely, it leaks at least uncrypted one-shot acesses to each V_i to Alice.*

We first need to prove the following lemma.

Lemma III.1. *Let $A \in \mathcal{L}(\mathcal{H}_E \otimes \mathcal{H}_F)$. Then: $(\forall U \in \mathbb{U}(\mathcal{H}_E), [A, U \otimes \mathbb{1}_F] = 0) \implies A = \mathbb{1}_E \otimes B$, where $B \in \mathbb{U}(\mathcal{H}_F)$.*

Proof. Let $A = \sum_{ijkl} a_{ijkl} |i\rangle\langle j|_E \otimes |k\rangle\langle l|_F$. As the commutation relation is true for any U, let us

consider $U = \sum_m u_m |m\rangle\langle m|$. We can write the two matrices as the following:

$$\begin{aligned} A(U \otimes \mathbb{1}_F) &= \sum_{ijklm} a_{ijkl} u_m (|i\rangle\langle j|_E \otimes |k\rangle\langle l|_F) \times (|m\rangle\langle m| \otimes \mathbb{1}_F) = \sum_{ijkl} a_{ijkl} u_j \, |i\rangle\langle j|_E \otimes |k\rangle\langle l| \\ (U \otimes \mathbb{1}_F)A &= \sum_{ijklm} a_{ijkl} u_m (|m\rangle\langle m| \otimes \mathbb{1}_F) \times (|i\rangle\langle j|_E \otimes |k\rangle\langle l|_F) = \sum_{ijkl} a_{ijkl} u_i \, |i\rangle\langle j|_E \otimes |k\rangle\langle l| \end{aligned} \tag{III.1}$$

The commutation relation tells us that those two matrices must be equal, and therefore all of their coefficients. In particular, for $i \neq j$, we have:

$$a_{ijkl} u_i = a_{ijkl} u_j \implies a_{ijkl} = 0 \text{ or } u_i = u_j \tag{III.2}$$

In the special case where U contains all differents coefficients (for example, $u_m = e^{2i\pi \frac{m}{d_E}}$, where $i^2 = 1$ and $d_E = \dim \mathcal{H}_E$), then this means that $\forall i \neq j, a_{ijkl} = 0$, and we can rewrite $A = \sum_{ikl} a_{ikl} \, |i\rangle\langle i|_E \otimes |k\rangle\langle l|_F$.

We can do the same trick when taking $U = \sum_m |m\rangle\langle m + r|$, with $r \in \mathbb{N}$, and we consider $|m + r\rangle = |(m + r) \mod d_E\rangle$, same for $\langle m + r|$. Again, we obtain:

$$\begin{aligned} A(U \otimes \mathbb{1}_F) &= \sum_{iklm} a_{ikl} (|i\rangle\langle i|_E \otimes |k\rangle\langle l|_F) \times (|m\rangle\langle m + r| \otimes \mathbb{1}_F) = \sum_{ikl} a_{ikl} \, |i\rangle\langle i + r|_E \otimes |k\rangle\langle l| \\ (U \otimes \mathbb{1}_F)A &= \sum_{iklm} a_{ikl} (|m\rangle\langle m + r| \otimes \mathbb{1}_F) \times (|i\rangle\langle i|_E \otimes |k\rangle\langle l|_F) = \sum_{ikl} a_{i+r,kl} \, |i\rangle\langle i + r|_E \otimes |k\rangle\langle l| \, , \end{aligned} \tag{III.3}$$

with the index change $i = i + r$ for the second line. Taking the equality of the coefficients, we get $a_{i,k,l} = a_{i+r,k,l}$, for any r. Therefore the coefficient does not depend on i, and we can write it as $a_{k,l}$. This means that we get:

$$A = \sum_{ikl} a_{kl} \, |i\rangle\langle i|_E \otimes |k\rangle\langle l|_F = (\sum_i |i\rangle\langle i|_E) \otimes (\sum_{kl} a_{kl} \, |k\rangle\langle l|_F) = \mathbb{1}_E \otimes B_F, B_F = \sum_{kl} a_{kl} \, |k\rangle\langle l| \tag{III.4}$$

□

Proof of Theorem III.2. Let us suppose that there exists i such that Alice does not have an uncrypted access to V_i. This means that if we look at the shape of the global computation in Figure II.8, then Bob possesses two maps $\mathcal{E}, \mathcal{D} \in \mathrm{QChan}(\mathcal{H}_A \otimes \mathcal{H}_F \to \mathcal{H}_A \otimes \mathcal{H}_F)$ such that:

$$\mathcal{D} \circ (\mathcal{U} \otimes \mathcal{I}) \circ \mathcal{E} = (\mathcal{U} \otimes \mathcal{I}) \tag{III.5}$$

namely he can encode and decode through Alice applying her unitary. We make here a small note to explain the right handside term. Although we could allow for a custom channel on the ancillary part and obtain $\mathcal{U} \otimes \mathcal{C}$, if Bob wants to constructs a correct protocol, the channel needs to be invertible and thus be a unitary channel. One could show that in that case, the unitary channel can be taken into account by the encryption and decryption maps, and thus this case is the most general one.

We also make a distinction here about why we consider this scheme. If Bob does not make an encryption scheme that looks like Equation III.5, then the total computation that Bob is doing can be seen as another supermap, namely the decomposition is different. And still, if Bob is making the correct computation $f(U)$ at the end, Alice would still get access to matrices that achieve $f(U)$, even though they could be different each time. Bob also cannot make his gates private using gate teleportation, which is a way to delay the gate applied using a similar principle as in quantum teleportation [48]. Indeed, as Alice is the one getting the final result, then in order to have a correct protocol, Bob would need to teleport the gates first, and therefore Alice would still get unencrypted access to those gates.

We now prove that no non-trivial maps $\mathcal{E}, \mathcal{D}$ exist that satisfy Equation III.5, namely they must satisfy the following equation:

$$\exists A \in \mathbb{U}(\mathcal{H}_F), \mathcal{E} = \mathcal{I}_A \otimes \mathcal{A}, \mathcal{D} = \mathcal{I}_A \otimes \mathcal{A}^\dagger, \tag{III.6}$$

where $\mathcal{A}$ represents the unitary channel corresponding to A. Lemma II.2 tells us that the encryption and decryption map must be unitary channels, as $\mathcal{U}\otimes\mathcal{I}$ is a unitary channel corresponding to $U\otimes\mathbb{1}_F$. We can thus rewrite $\mathcal{E}(\rho) = E\rho E^\dagger, \mathcal{D}(\rho) = D\rho D^\dagger$, with $D, E \in \mathbb{U}(\mathcal{H}_E \otimes \mathcal{H}_F)$. Therefore, up to a global phase, we have the equality:

$$\forall U \in \mathbb{U}(\mathcal{H}_A), D(U \otimes \mathbb{1}_F)E = U \otimes \mathbb{1}_F \tag{III.7}$$

In particular, with $U = \mathbb{1}_A$, we get $ED = \mathbb{1}$, and thus $D = E^\dagger$. Equation III.7 becomes:

$$\forall U \in \mathbb{U}(\mathcal{H}_A), E^\dagger(U \otimes \mathbb{1}_F)E = U \otimes \mathbb{1}_F \implies \forall U \in \mathbb{U}(\mathcal{H}_A), [E, U \otimes \mathbb{1}_F] = 0 \tag{III.8}$$

Lemma III.1 tells us that $E = \mathbb{1}_E \otimes B, B \in \mathbb{U}(\mathcal{H}_F)$, and therefore $D = \mathbb{1}_E \otimes B^\dagger$, which concludes the proof.

□

This result is quite strong, because it tells us the following: Bob cannot directly encrypt and decrypt each of his matrices V_i. Indeed, if he does, then as U is unknown, then any encryption or decryption scheme that is non-trivial (e.g. not the one from Equation III.6) cannot yield back the unitary for Alice. Therefore, if he encrypts his data with a non-trivial scheme, then another matrix than U will be applied, and this will be transformed into another supermap (with maybe different matrices used in each slot). This would need higher-order results about how a supermap would react to this type of transformation, and we will talk about it in Section III.5. In the meantime, we can consider that Bob has no encryption scheme for his data, and thus Alice always has unencrypted one-shot access to each V_i, and could decide to cheat and to learn each V_i through tomography.

Does this means that the same reasoning could be done for Alice, and no security could be achieved for both parties? Fortunately, this is not the case. The result from above comes from the fact that U is unknown; and here, although Bob wants to keep V_i secret, he still has the control over which matrix he is applying, and could share classical data with Alice to achieve her security. How could we build a protocol like this? Let us consider one round. Alice has some state $|\psi\rangle \in \mathcal{H}_A$ that she wishes to keep private. She communicates with Bob such that at the end of the computation, Alice has the state $V|\psi\rangle$, with $V \in \mathcal{L}(\mathcal{H}_A)$ being known by Bob. The most general encryption Alice could do is apply a unitary matrix based on a key, which can be representend on a quantum register $|x_i\rangle \in \mathcal{H}_K$. Therefore, we denote the encryption map from Alice $\mathcal{E}(\rho) = \sum_i |x_i\rangle\langle x_i| \otimes U_i \rho U_i^\dagger \in \mathrm{QChan}(\mathcal{H}_K \otimes \mathcal{H}_A \to \mathcal{H}_K \otimes \mathcal{H}_A)$, where the encryption on the effective qubits is $\tilde{\mathcal{E}} = \mathrm{Tr}_K[\mathcal{E}]$.

Bob then applies a unitary matrix W, which may or may not be the computation matrix V. There are two main possibilities for the construction of the protocol:

- Alice wants a perfect encryption scheme, namely $\tilde{\mathcal{E}}(\rho) = \frac{\mathbb{1}_A}{d_A}$. Then in order to decode, Alice and Bob will need to share to each other some classical data. In this case, the encryption scheme does not commute with the computation from Bob, which will bring some problems for security, but the encryption itself is perfectly secure. This is explained in detail in Section III.4.1.

- We try to build an encoding scheme that commutes with Bob's computation. Namely, if V commutes with each U_i up to a global phase, then Bob can just apply V, and Alice decrypts it by applying $U_i^\dagger$. Note that in this case, Alice could delay the choice of the key until the end of the global computation. This focuses on them communicating before the communication, and not sending anything during it. This is explained in detail in Section III.4.2.

- Finally, we use the commuting framework where Alice chooses the basis in which she does the encryption. This requires Bob to send some correction. This is a trade-off that increases Alice's security but makes Bob's classical leaks more important. This is explained in detail in Section III.4.3.

We consider these three situations in the following sections. Although it seems to be the only ways to do an actual encryption over an unknown unitary channel, a way to continue this project would be to show that these are the three main ways an encryption over an unknown computation happens.

One term that we will use in proofs is a one-shot access to a matrix M. This means that if a party was to cheat and send some known state $|\phi\rangle$ to another party, then when it gets it back,

there is some matrix applied onto the original state that depends on M, and such that Bob knows this relation. This may be for example getting back $MN|\phi\rangle$, where Bob knows N. Bob thus can measure in any basis he wants, but basically gets an access of any state he knows applied on M. This is to indicate that Bob will be able to learn this matrix through tomography if multiple communication rounds take place.

III.3 Verifiability

We make here a short remark on the verifiability of such protocol. Both parties cannot do any verification during the protocol, because the other party applies a unitary channel that is unknown. The only verification possible is at the end; and as Bob does not know U, he can't do this verification. The only verification possible is Alice checking the result at the end. Of course, she can only check it if she knows the input state and unitary. Unless the function f satisfies some property (for example, if $f(A \otimes B) = f(A) \otimes f(B)$, Alice could input $U \otimes A$, with A a chosen matrix, and verify only on these qubits), the verification is destructive, and she cannot verify and get the result at the same time. She can insert n computation rounds for 1 actual round, where she uses a known state and unitary and measures at the end to verify if Bob cheated. Sadly, if Bob can cheat without disturbing the final result, then this verification does not detect this malicious behaviour. Moreover, if he was to cheat on the actual round, this would not be detected.

However, this still has some advantages. First, Bob is only able to learn some actual data with probabibility $\sim \frac{1}{n}$. Furthermore, if they communicate through multiple rounds, even if Bob was to cheat each time and does not get detected, as he starts with no information from Alice's inputs, then it will be hard for him to learn data through the rounds. Indeed, he has no idea about if it is an actual round or a test round, and therefore it is hard for him to do tomography on the states he gets to try to recover the inputs from Alice.

III.4 HODQC protocols

III.4.1 First protocol - keysharing

In this protocol, we focus on the second situation described above. Our starting point is that we want to have a secure encryption. This means that our encryption map is $\tilde{\mathcal{E}}(\rho) = \sum_i U_i \rho U_i^\dagger = \frac{\mathbb{1}_A}{d_A}$, i.e. a fully depolarizing channel. We know that another set of Kraus operators is given by the Pauli matrices $\{P_j\}$ of size d_A. Therefore, Lemma II.1 tells us that we have a unitary matrix $A = (a_{i,j})$ such that:

$$\forall j, P_j = \sum_i a_{i,j} U_j \tag{III.9}$$

If V commutes with each U_i, then it commutes with any Pauli matrix. However, we show that any matrix commuting (up to a phase) with the Pauli group is one of them. Why do we consider it up to a phase? Given A, B two matrices, if $AB = \alpha BA, \alpha \in \mathbb{U}$, then the quantum operation resulting is, since $|\alpha| = 1$:

$$AB\rho(AB)^\dagger = \alpha BA\rho(\alpha BA)^\dagger = \alpha\bar{\alpha} BA\rho(BA)^\dagger = BA\rho(BA)^\dagger \tag{III.10}$$

Therefore, AB and BA give the same quantum operation, and we can consider commuting up to a phase.

Lemma III.2. *Let $n \in \mathbb{N}^*, M \in \mathbb{U}_{2^n}$. Then if M commutes up to a phase with the Pauli group of size n, i.e. $\forall P \in G_n, \exists \alpha \in \mathbb{U}$ such that $MP = \alpha PM$, then $M \in G_n$ (up to a phase factor).*

We need first this lemma.

Lemma III.3. *Let $n \in \mathbb{N}^*, M \in \mathbb{U}_{2^n}$. Then there exists $P \in G_1, A \in \mathbb{U}_{2^{n-1}}$ such that $M = P \otimes A$.*

Proof. Let M be a unitary matrix of size n. We can rewrite M as the following: $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$, where A, B, C, D are square matrices of size 2^{n-1}. In particular, it commutes up to a phase with $Z \otimes \mathbb{1}_{2^{n-1}}$, where this matrix can be rewritten as $\begin{pmatrix} \mathbb{1}_2 & 0 \\ 0 & \mathbb{1}_2 \end{pmatrix}$. This means that we have $\alpha \in \mathbb{U}$ such that $M(Z \otimes \mathbb{1}_{2^{n-1}}) = \alpha(Z \otimes \mathbb{1}_{2^{n-1}})M$. Writing the product as blocks, we get:

$$\begin{pmatrix} A & -B \\ C & -D \end{pmatrix} = \alpha \begin{pmatrix} A & B \\ -C & -D \end{pmatrix} \tag{III.11}$$

We can rewrite this as block equalities, that lead to two possible cases: either $\alpha = 1$, and then $B = C = 0$; either $\alpha = -1$, and then $A = D = 0$. This comes simply by plugging $\alpha = 1,\ -1$ in the block equalities. If α was something else, then all the blocks would be 0, and thus M would not be a unitary matrix.

The same reasoning can be done with the Pauli $X \otimes \mathbb{1}_{2^{n-1}} = \begin{pmatrix} 0 & \mathbb{1}_2 \\ \mathbb{1}_2 & 0 \end{pmatrix}$. Therefore, there exists $\beta \in \mathbb{U}$ such that:

$$\begin{pmatrix} B & A \\ D & C \end{pmatrix} = \beta \begin{pmatrix} C & D \\ A & B \end{pmatrix} \tag{III.12}$$

We see that each matrix satisfies $A = \beta^2 A$ (same for B, C, D). Therefore if $\beta \notin \{1, -1\}$, then $A = B = C = D = 0$, and M is not a unitary matrix anymore. If $\beta = 1$, this gives $A = D$ and $B = C$; if $\beta = -1$, then we have $A = -D$ and $B = -C$.

Those two cases can be summed up in Table III.1, and each matrix can be rewritten as defined above, which concludes the proof.

α \ β	1	-1
1	$\begin{pmatrix} A & 0 \\ 0 & A \end{pmatrix} = \mathbb{1}_2 \otimes A$	$\begin{pmatrix} A & 0 \\ 0 & -A \end{pmatrix} = Z \otimes A$
-1	$\begin{pmatrix} 0 & A \\ A & 0 \end{pmatrix} = X \otimes A$	$\begin{pmatrix} 0 & A \\ -A & 0 \end{pmatrix} = iY \otimes A$

Table III.1: Table representing M depending on the commuting phase. $A \in \mathbb{U}_{2^{n-1}}$ is not identical at each case.

□

We can now prove the first lemma.

Proof of Lemma III.2. We will prove this lemma by induction on $n \in \mathbb{N}^*$. For the base case $n = 1, M \in \mathbb{U}_2$, Lemma III.3 tells us that we have $P \in G_1, A \in \mathbb{U}_1 = \mathbb{U}$ such that $M = P \otimes A$. As A is a matrix of size 1, it can be assimilated to a scalar, and therefore we can rewrite it as $M = AP$, with $A \in \mathbb{U}$, thus satisfying the base case.

For the induction step, let $n = k, M \in \mathbb{U}_{2^k}$. Again, Lemma III.3 tells us that we have $P \in G_1, A \in \mathbb{U}_{2k-1}, M = P \otimes A$. We know that as M commutes with all the Pauli in G_k, then in particular it commutes, up to a phase $\alpha_Q \in \mathbb{U}$ with $\mathbb{1}_2 \otimes Q$, for any $Q \in G_{k-1}$. We can write it with the specific form of M:

$$\begin{aligned} &\forall Q \in G_{k-1},\ \exists \alpha_Q \in \mathbb{U},\ M(\mathbb{1}_2 \otimes Q) = (\mathbb{1}_2 \otimes Q)M \\ &\forall Q \in G_{k-1},\ \exists \alpha_Q \in \mathbb{U},\ P \otimes AQ = P \otimes QA \end{aligned} \tag{III.13}$$

Therefore, this means that A commutes up to a phase with every Pauli in G_{k-1}. Therefore, the induction hypothesis tells us that A is a Pauli up to a phase, i.e. $A = \alpha Q, Q \in G_{k-1}, \alpha \in \mathbb{U}$; and we have $M = \alpha P \otimes Q = \alpha P', P' = P \otimes Q \in G_k$, and M is also a Pauli up to a phase (as $G_1 \otimes G_{k-1} \subseteq G_k$).

□

Therefore, Bob can only apply a Pauli matrix, which is too restrictive. Even if he did so, then he would need to send a matrix classically to Alice in the case where V_i is not a Pauli, and Alice could find back V_i from this communication (by testing all the possibilites for each Pauli). This means that the decoding must depend on the matrix that Bob applied, and they must communicate classically; Bob sends a matrix to Alice for her to apply. If Alice keeps her key private, then Bob has no choice but to send classically V to Alice, which we would like to avoid. Also, if Bob applies V, then the decryption part will consist on applying $V^\dagger U_i^\dagger V$, and then Alice can fully recover V from this matrix. We propose a way to avoid this in Protocol 1, and the communication scheme of one round is drawn in Figure III.2, where we represent how Alice and Bob communicate to achieve V_i.

Protocol 1 HODQC with Bob having $\mathcal{D}(f)$ - keysharing protocol

1. **Alice's preparation**
   (a) Alice has a state $|\psi\rangle$ and a unitary U acting on the same space $\mathcal{L}(\mathcal{H}_A)$.
   (b) She prepares the state on the data and ancillary state $|\gamma_0\rangle = |\psi\rangle \otimes |0\rangle \in \mathcal{L}(\mathcal{H}_A \otimes \mathcal{H}_E)$
2. **Communication rounds**
   For $i \in [0 \dots m]$
   (a) Alice choose a random n-qubit Pauli $K_j \in G_d$, with $d = \dim \mathcal{H}_A \times \dim \mathcal{H}_E$, and applies it to $|\gamma_i\rangle$, then sends her state to Bob
   (b) Bob chooses a random unitary $W_i \in \mathbb{U}_d$, and applies to the state, then sends it back to Alice
   (c) After receiving the state, Alice sends through a classical channel the used one-time pad K_j
   (d) Bob sends through a classical channel $C_{i,j} = V_i K_j^\dagger W_i^\dagger$
   (e) Alice applies $C_{i,j}$ using BQC protocols with a server (it may be Bob)
   (f) If $i < m$, Alice applies $U \otimes \mathbb{1}_E$
   (g) The state after this round is denoted $|\gamma\rangle_{i+1}$

The idea of this algorithm is that the classical information shared by Bob is independent from V_i (as W_i is chosen randomly), and thus Alice does not learn V_i classically. However, this requires Alice to share her key to Bob, by assuming that Bob has not the state and the key at the same time. Therefore, if Bob possesses a quantum memory, he could keep the state, and decrypt it; in this case, this would not be coherent with our ideal resource model. Another problem is that although applying a random unitary on a state gives an informationally secure state the same way as a quantum one-time pad, it is not possible to apply the unitary later and to delay the choice. Therefore, in this framework, there is going be to be leakage of such unitary. This will be more useful later, as here we do not write formally a proof using this framework, as it seems complicated to define formally the absence of quantum memory for Bob.

Theorem III.3 (Security results of Protocol 1 - No quantum memory)**.** *Let Alice and Bob communicate to achieve HODQC over Protocol 1. We suppose that Bob has no quantum memory. Then:*
- Alice does not leak anything.
- In one communication, Bob leaks one-shot accesses to each V_i. Alice can learn those quantities as close as she wants over multiple computation rounds through tomography.

Proof. We will do this proof using the informationally secure definition. Namely, any encryption is informationally secure if this yields for any quantum state the depolarizing map, i.e. $\mathcal{E}(\rho) = \frac{\mathbb{1}}{d}$. For a given state $\rho \in \mathcal{L}(\mathcal{H}_P)$, with $d_P = \dim \mathcal{H}_P = 2^n, n \in \mathbb{N}$, then we have the following equalities,

with Ω being the Haar measure on $\mathcal{U}_P$ [49]:

$$\frac{1}{d_P^2}\sum_{G\in G_{2^n}} G\rho G^\dagger = \frac{\mathbb{1}_P}{d_P} \tag{III.14}$$

$$\int_{G\in\mathcal{U}_P} G\rho G^\dagger d\Omega = \frac{\mathbb{1}_P}{d_P} \tag{III.15}$$

Therefore we can use this as the following: the state that Alice sends to Bob is informationally secure, because she applies one randomly chosen Pauli. Bob applies a randomly chosen unitary, so it is still informationally secure. As we assumed that Bob had no quantum memory, if he wants to cheat, he must do it while he has the state, and thus while he does not know the key the state is still secure. When Alice sends her key, he is not in possession of the state anymore, and thus the key itself does not allow Bob to learn anything. Also, as W_i is chosen randomly, then $C_{i,j}$ is independent from V_i, and thus Alice does not learn anything from the classical communication. The only leak appears after the decryption: Alice gets an uncrypted access to V_i. □

Protocol 1b Cheating protocol for Bob when he has $\mathcal{D}(f)$ and Alice runs Protocol 1

1. **Alice's preparation**
   (a) Alice has a state $|\psi\rangle$ and a unitary U acting on the same space $\mathcal{L}(\mathcal{H}_A)$.
   (b) She prepares the state on the data and ancillary state $|\gamma_0\rangle = |\psi\rangle \otimes |0\rangle \in \mathcal{L}(\mathcal{H}_A \otimes \mathcal{H}_E)$
2. **Communication rounds**
   For $i \in [0 \dots m]$
   (a) Bob acts honestly as in Protocol 1, and communicates with Alice to compute V_i
   (b) When $i = m$, he sends to Alice another state prepared by himself, and keeps Alice's state
   (c) He remembers $C_{m,j}$ for later, and sends classically a random matrix $C'_{n,j}$ to Alice.

   For $i \in [m+1 \dots m+m']$
   (a) When Bob receives a state, he keeps it, and sends a state that he prepares to Alice.
   (b) When he receives K_j, he can decrypt the state and has a one-shot access to Alice's input state, which is $U|\psi'\rangle$, with $|\psi'\rangle$ the state sent by Bob on the previous round.
   (c) He chooses $C_{i,j}$ randomly.
   (d) For the final round $i = m + m'$, he sends back the state from the m-th round, with the correction $C_{m,j}$

Theorem III.4 (Security results of Protocol 1 - Quantum memory)**.** - *Alice leaks an unencrypted access to $|\psi\rangle$ and m unencrypted accesses to U. Bob can learn those quantities as close as he wants on multiple computations, through tomography.*

- In one communication, Bob leaks one-shot accesses to each V_i. Alice can learn those quantities as close as she wants over multiple computations rounds, through tomography.

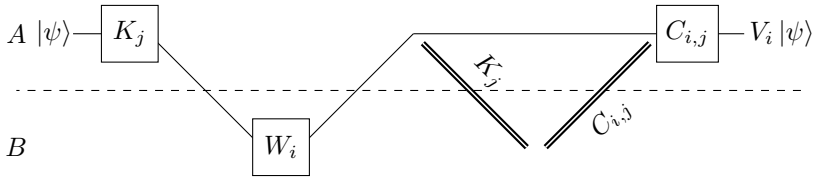

Figure III.2: Circuit Description of one round of Protocol 1. Lines represent quantum communication, and doubled lines classical communication.

A simple way of seeing how Bob can cheat in this case is described in Protocol 1b, without him altering the final computation. Here, we see that Bob does not deviate from the computation, as Alice gets the good result; and therefore Bob has multiple accesses to U. Assuming Alice has no result on the number of copies required, Bob can ask for more and get good results on U, even with only one computation of $f(U)$, and Alice cannot detect this cheating.

This problem occurs because we assumed that Bob would not have access to both the state and the key, but letting Bob know the key at a given point is a problem, because Bob can therefore delay the end of the computation, and undo the privacy from Alice. We therefore need to find another way of doing the protocol. To avoid this, we could let Bob send to Alice all $C_{i,j}$ for all j, and Alice would just choose the one that correspond to her. But if Alice has access to two different copies, then she could just take the product $C_{i,j}C_{i,j'}^\dagger = V_i K_j^\dagger K_{j'} V_i^\dagger$, and as she knows the expression of K_i, this gives her the expression of V_i.

As long as Alice's operation does not commute with W_i, then Bob's correction will depend on the operation that Alice did, because the set of possible one-time pad do not yield the same matrix before correction. So we could try to build a protocol where W_i commute with the one-time pads. This gives an advantage, which is that Bob cannot keep the state anymore without perturbing the computation. But this comes with a disadvantage, which is that Alice learn a part of Bob's information, but which is constant; however, Alice's information is not perfectly secure anymore. This is a trade-off between the two parties.

III.4.2 Second protocol - commuting matrices

Now we consider the second situation. We want to find a way for Alice to build an encryption scheme that commutes with Bob's computation, and such that she can choose her keys independently and without sharing them with Bob; and thus this would be more resistent against all attacks from Bob. Proposition III.1 shows the conditions to commute with a matrix, given its diagonalization basis and the number of eigenvalues.

Definition III.3. *Let $n \in \mathbb{N}$. We denote $T_n(k)$ the set of k-tuple integers that sum up to n. For any permutation $\sigma \in \mathcal{S}_n$, we write the given permutation matrix as S_σ. We then call R_n the set of block randomized matrices defined as $R_n = \{S_\sigma^\dagger(\oplus_{i=1}^k M_i)S_\sigma : \sigma \in S_n, (m_1, \dots, m_k) \in T_n(k) : M_i \in \mathcal{U}_{m_i}\}$. For a given $\sigma \in S_n, (m_1, \dots, m_k) \in T_n(k)$, we denote $R_n(\sigma, m_1, \dots, m_k) = \{S_\sigma^\dagger(\oplus_{i=1}^k M_i)S_\sigma : M_i \in \mathcal{U}_{m_i}\}$.*

Proposition III.1. *Let B be a unitary matrix of size $n \times n$, with the diagonalization given by $B = P^\dagger D P$. We denote $\{(\lambda_i, m_i)\}_{1 \le i \le k}$ the eigenvalues of B with their respective multiplicities. Then there exists $\sigma \in S_n$ such that the set of matrices that commute with B is $\{P^\dagger R P, R \in R_n(\sigma, m_1, \dots, m_k)\}$.*

In order to prove this proposition, we need two lemmas.

Lemma III.4. *Let A, B be two unitary matrices, with the diagonalization of B given by $B = P^\dagger D P$. Then $[A, B] = 0 \iff [D, PAP^\dagger] = 0$*

Proof. The following relations hold because $PP^\dagger = P^\dagger P = \mathbb{1}$, and therefore P is invertible and preserve the equivalence.
$[A, B] = 0 \iff BA = AB \iff P^\dagger DPA = AP^\dagger DP \iff DPAP^\dagger = PAP^\dagger D \iff [D, PAP^\dagger] = 0$ □

Lemma III.5. *Let D be a diagonal matrix. We denote its eigenvalues $\{\lambda_i\}_{1\leq i\leq K}$ with respective multiplicity $\{m_i\}_{1\leq i\leq K}$, so that $D = \oplus_{i=1}^{K}\lambda_i I_{m_i}$, with I_k the identity of dimension k. Then all commutating matrices can be written in the form $\oplus_{i=1}^{K} M_i$, with M_i a $m_i \times m_i$ unitary matrix.*

Proof. Let $M = (m_{ij})_{i,j}$ be a matrix commuting with $D = (\lambda_i\delta_{ij})_{i,j}$. We can compute the products: $MD = (\sum_k m_{ik}d_{kj})_{i,j} = (\lambda_j m_{ij})_{i,j}$, and $DM = (\sum_k d_{ik}m_{kj})_{i,j} = (\lambda_i m_{ij})_{i,j}$. As we work with unitaries, then $\lambda_j \neq 0$, and we have the following condition: $\lambda_i \neq \lambda_j \implies m_{ij} = 0$. Therefore, if the eigenvalues are grouped together and $D = \oplus_{i=1}^{K}\lambda_i I_{m_i}$, then this condition implies the block form of M. We can check that every matrix in this block form commute with D. □

Proof of Proposition III.1. The result comes from the concatenation of the two previous lemmas. Let A be a matrix commuting with B. By Lemma III.4, we know that $PAP^\dagger$ commutes with D. Then we can rewrite $D = S_\sigma^\dagger D' S_\sigma$, by grouping the eigenvalues by multiplicity in D'. Then Lemma III.5, this means that $S_\sigma PA(S_\sigma P)^\dagger = \oplus_{i=1}^{K} M_i$, with M_i a $m_i \times m_i$ unitary matrix. Therefore, we can write $A = (S_\sigma P)^\dagger \oplus_{i=1}^{K} M_i S_\sigma P$, which concludes the proof. □

Now, suppose that we are given a unitary matrix with the diagonalization $B = P^\dagger DP$, where $D = \mathrm{diag}(1, i, -i, -1)$. Proposition III.1 tells that any matrix A commuting with B needs to be of the form $A = P^\dagger D' P$, with D' also diagonal. Therefore, if Bob wants to tell to Alice how she can build a matrix commuting with B, then Alice would have to learn P, and that B has 4 eigenvalues. But if $D = \mathrm{diag}(1, 1, -1, -1)$, then A is allowed to have a more general shape, which is $A = P^\dagger(E \oplus F)P$, with E, F being two unitaries of size 2. In this case, instead of sending P to Alice, Bob could send $Q = (N \oplus M)P$, and it would still commute if Alice builds $A = Q^\dagger(E \oplus F)Q$. However, this is still dependent on the multiplicities of the eigenvalues of B. Proposition III.2 exhibits a decomposition for all unitaries of size $4m$ where $B = B^1B^2B^3$, with B^i being a unitary of same size, with the same diagonalization basis as B, such that the number and structure of the eigenvalues is fixed and the same for each B, and each eigenvalue is of multiplicity at least 2. Using this, Alice can build commuting matrices without learning everything back from B.

Proposition III.2. *Let B be a unitary matrix of size $4n$, and its diagonal decomposition $B = P^\dagger DP$. Then we can find 3 matrices B^i such that $B = B^1B^2B^3$, and B^i is a matrix with at most $2n$ eigenvalues of multiplicity at least 2, and is diagonal in the same basis as B.*

Lemma III.6. *Let D be a 4×4 diagonal matrix with non-zero determinant (i.e. all eigenvalues are non-zero). We denote diagonal matrices as $D = diag(\alpha, \beta, \gamma, \delta)$. Then we can find $a, b, c, d, e, f \in \mathbb{C}$ such that:*

$$D = \mathrm{diag}(a, a, b, b)\mathrm{diag}(c, d, c, d)\mathrm{diag}(e, f, f, e).$$

We denote those three matrices as $\Lambda^{(i)}(D), i \in 1, 2, 3$, so $D = \Lambda^{(1)}(D)\Lambda^{(2)}(D)\Lambda^{(3)}(D)$.

Proof. We can take the coefficients given by the following equations and verify the equalities.

$$\begin{cases} a = (\frac{\alpha\beta}{\sqrt{\gamma\delta}})^{1/3} \\ b = (\frac{\gamma\delta}{\sqrt{\alpha\beta}})^{1/3} \\ c = (\frac{\alpha\gamma}{\sqrt{\beta\delta}})^{1/3} \\ d = (\frac{\beta\delta}{\sqrt{\alpha\gamma}})^{1/3} \\ e = (\frac{\alpha\delta}{\sqrt{\beta\gamma}})^{1/3} \\ f = (\frac{\beta\gamma}{\sqrt{\alpha\delta}})^{1/3} \end{cases} \tag{III.16}$$

□

Proof of Proposition III.2. This decomposition comes directly from Lemma III.6. Each diagonal matrix of size $4n$ can be written as n blocks of size 4. Then we can use the previous result for each block, and we have a decomposition of at most $2n$ eigenvalues of multiplicity at least 2. More formally, we can write $D = \oplus_{j=1}^{n} D^j$, with D^j diagonal matrices of size 4. Then Lemma III.6 tells

us that for each j, we can find $\Lambda^{(1)}(D^j), \Lambda^{(2)}(D^j), \Lambda^{(3)}(D^j)$ such that $\Lambda^{(1)}(D^j)\Lambda^{(2)}(D^j)\Lambda^{(3)}(D^j) = D^j$. Then:

$$\begin{aligned} D &= \oplus_{j=1}^{n} \Lambda^{(1)}(D^j)\Lambda^{(2)}(D^j)\Lambda^{(3)}(D^j) \\ &= \prod_{i=1}^{3} \oplus_{j=1}^{n} \Lambda^{(i)}(D^j) \end{aligned} \tag{III.17}$$

Then we can take $B_i = P^\dagger \oplus_{j=1}^{n} \Lambda^{(i)}(D^j)P$, and one can check that $B = B^1B^2B^3$. □

Now that we have the general scheme of commuting matrices, how can we bring security using this structure for Alice ?

Proposition III.3. *Let B be a unitary matrix of size $4n$, and its decomposition $B = P^\dagger DP$, with the decomposition $B = B^1B^2B^3$. Then for each j, we can find a set of matrices $K_{i,j}$ that commute with B^j such that if one party applies randomly $\{K_{i,j}\}_i$, the state sent to the other party is a diagonal matrix, which contains $2n$ distinct diagonal coefficients. Each of those coefficients are given by the mean of two distinct diagonal coefficients of the input state ρ in the basis P, all $4n$ diagonal coefficients being used.*

Proof. Let one party have a set of size κ of matrices K_i in which she picks randomly one to apply to her state, and that commute with B^1; the same reasoning can also be done for B^2 and B^3. Then Lemma III.6 states that they need to be of the shape $K_i = P^\dagger \oplus_{j=1}^{2n} K_{i,j}P$, with $K_{i,j}$ unitary matrices of size 2. After applying randomly a matrix on a given state ρ, the output state is $\frac{1}{K}\sum_{i=1}^{\kappa} K_i^\dagger \rho K_i$, where we can factorize the basis to rewrite as $P^\dagger EP$.

The objective is to create a fully mixed state, so to have $P^\dagger EP = \frac{1}{4n}\mathbb{1}$, therefore we want $E = \frac{1}{4n}\mathbb{1}$. We use the following block notation $P\rho P^\dagger = (\rho_{i,j})_{1\leq i,j\leq 2n}$, where $\rho_{i,j}$ are blocks of size 2×2. Then we can also decompose E into $4n^2$ blocks as $E = (E_{i,j})_{1\leq i,j\leq 2n}$. By definition of E, we have $E_{i,j} = \frac{1}{\kappa}\sum_{l=1}^{\kappa} K_{l,i}^\dagger \rho_{i,j} K_{l,j}$. If we consider the diagonal blocks of E, we get the equation:

$$E_{j,j} = \frac{1}{\kappa}\sum_{i=1}^{\kappa} K_{i,j}^\dagger \rho_{j,j} K_{i,j} \tag{III.18}$$

If we want a secure state, then we need to bring this close to identity. We can take $K_{i,j}$ being a set of Kraus operators of the depolarizing channel acting on qubits for a fixed j. Sadly, $\rho_{j,j}$ is not of trace 1, so we can't remove that. This means that we get:

$$E_{j,j} = \frac{Tr(\rho_{j,j})}{2}\mathbb{1} \tag{III.19}$$

For the off-diagonal blocks, we have:

$$E_{j,j'} = \frac{1}{K}\sum_{i=1}^{K} K_{i,j}^\dagger \rho_{j,j'} K_{i,j'} \tag{III.20}$$

To minimize this quantity, we can choose $K_{i,j} = \sigma_k e^{i\alpha_{l,j}Z}$, where $i = (k,l) \in \{0,1,2,3\} \times [0\dots n-1]$, and $\alpha_{l,j} = \frac{\pm 2\pi l j}{n}$. Those are still forming a depolarizing channel, and we can separate the two sums on k and l to get:

$$E_{j,j'} = \frac{Tr(\rho_{j,j'})}{2n}\sum_{l=1}^{n} (e^{i\alpha_{l,j}Z})^\dagger e^{i\alpha_{l,j'}Z} \tag{III.21}$$

The two diagonal coefficients of this matrix are $c_{j,j'}^{\pm}$:

$$c_{j,j'}^{\pm} = \frac{Tr(\rho_{j,j'})}{2n}\sum_{l=1}^{n} e^{\frac{\pm 2i\pi l(j'-j)}{n}} = 0 \tag{III.22}$$

As $j \neq j'$ for off diagonal blocks, we recognize a geometric sum, and we have $c_{j,j'}^{\pm} = 0$. This concludes the proof.

□

Appendix A contains a more general proof when we apply a matrix like B^i to a known state; in that case, we can have perfect security for Alice, i.e. we can build a maximally mixed state.

Definition III.4. *Let P be a unitary matrix of size n. We say that P is row-hidden if for each row, if Alice learns one coefficient in the row, then she learns the whole row.*

Proposition III.4. *Let B be a unitary matrix of size $4n$, and its decomposition $B = \prod_{i=1}^{3} B^i$, with its diagonalization basis P. Then if Bob sends to Alice matrices that commute with each B^i, then P is row-hidden.*

Proof. First we need to consider the structure of B^i. We denote S_i^0 the permutation matrix of size 4 associated with the permutation $\sigma = (2, i+1)$, and $S_i = \oplus_{j=1}^{n} S_i^0$. Then for each B_i, we can find a matrix $\tilde{B^i} = \oplus_{j=1}^{2n} \alpha_j I_2$, such that $B^i = (S_i P)^\dagger \tilde{B^i} S_i P$. Basically, the coefficients from $\tilde{B_i}$ are the eigenvalues of B_i, but we have swapped them to have a better structure. Therefore, if Alice has any matrix K_i that commutes with B_i, then we know that it needs to be block diagonal in the basis $S_i P$. Taking random basis P_j for those block diagonal matrices, then this means that $K_i = Q_i^\dagger D_i Q_i$, with D_i a diagonal matrix, and $Q_i = (\oplus_{j=1}^{2n} P_j) S_i P$. From Q_i, although Alice does not know the P_j, she could take try to learn some blocks of Q_i. For any matrix of size $4n$, we can write it as $4n^2$ blocks of size 2×2. By denoting $S_i P = (C_{a,b}^i)_{1 \leq a,b \leq 2n}$, then by definition $Q_i = (P_a C_{a,b}^i)_{1 \leq a,b \leq 2n}$. Therefore for each block line, Alice can "erase" P_a by taking the product of the two blocks, the first one being inversed. This means that if Alice has access, to $\mathcal{B}_i$, then she has access to $(C_{a,b}^i)^\dagger C_{a,c}^i$ for $1 \leq a, b, c \leq 2n$. Taking $i = 1, b \neq c$, and denoting $P = (p_{i,j})_{1 \leq i,j \leq 4n}$, the product will be:

$$(C_{a,b}^1)^\dagger C_{a,c}^1 = P_{a,b}^\dagger P_{a,c} = \begin{pmatrix} p_{2a-1,2b-1}^* p_{2a-1,2c-1} + p_{2a,2b-1}^* p_{2a,2c-1} & p_{2a-1,2b-1}^* p_{2a-1,2c} + p_{2a,2b-1}^* p_{2a,2c} \\ p_{2a-1,2b}^* p_{2a-1,2c-1} + p_{2a,2b}^* p_{2a,2c-1} & p_{2a-1,2b}^* p_{2a-1,2c} + p_{2a,2b}^* p_{2a,2c} \end{pmatrix}$$

If we write all those equations for a fixed, and each $b \neq c$, we see that this is a linear system of equations, with the variables being product of the form $p_{a,j}^* p_{a,l}$, for each line a. Therefore, for each line, Alice learns one coefficient $p_{a,b}$, then with the products she learns all others $p_{a,b'}$. Due to the fact that this is conjugate products, then each line is defined like this, with α being undefined: $(\alpha, \alpha^* p_{i,2}, \ldots, \alpha^* p_{i,4n})$. Therefore even if she defines the lines up to a global phase, she still has an unknown. Therefore P is line-hidden (one variable for each line). □

All of the work above can be summed up in Theorem III.5, and detailed in Protocol 2.

Theorem III.5 (Security results for Protocol 2)**.** *Let Alice and Bob communicate to achieve HODQC over Protocol 2. We denote the current round $i \in [0 \ldots m]$ that achieves V_i which has as diagonalization basis P_i. Then:*

- In one computation, Alice leaks a one-shot accesses to the quasi-depolarization of $P_0 |\psi\rangle\langle\psi| (P_0)^\dagger$ and to $P_i U P_i^\dagger$ for $1 \leq i \leq m$, which means that it looks like a diagonal matrix with $2n$ diagonal coefficients, and each of them is given by the mean of two distinct diagonal coefficients of the original matrix. On multiple computations, Bob can recover all diagonal coefficients, through tomography, of $P_0 |\psi\rangle\langle\psi| (P_0)^\dagger$ and $P_i U P_i^\dagger$.

- In one computation, Bob leaks a one-shot accesss to V_i^j (from the decomposition of V_i), and leaks classically a row-hidden version of P_i, which means that if Alice learns one coefficient on one row of P_i, then she is able to learn the whole row. On multiple computations, Alice can recover through tomography V_i^j, but she does not learn more with multiple accesses to a row-hidden version of P_i.

What are the consequences of those leaks ? Bob leaks a row-hidden version of P_i. Therefore, as Alice just needs to find one coefficient for each row, she could use some search algorithm to try to find those coefficient, although this would be quite costly, i.e. she needs to learn $(m+1) \times d$ coefficients, where d is the dimension of the total space, and thus is exponential in the number of qubits. Bob could also increase the size of his ancillary system to make it more difficult for Alice.

For Alice, the leaks are made such that Bob learns at most the diagonal coefficients into a certain basis. However, these local views into different bases may be sufficient for Bob to get the full view of the unitary U, or he may cheat and provide a different basis to achieve this. Therefore, Alice's encryption may not be enough to guarantee her security.

Protocol 2 HODQC where Bob has $\mathcal{D}(f)$ - commuting matrices protocol

1. **Alice's preparation**
   (a) Alice has a state $|\psi\rangle$ and a unitary U acting on the same space $\mathcal{L}(\mathcal{H}_A)$.
   (b) She prepares the state on the data and ancillary state $|\gamma_{0,0}\rangle = |\psi\rangle \otimes |0\rangle \in \mathcal{L}(\mathcal{H}_A \otimes \mathcal{H}_E)$
   (c) We consider working with qubits, so we can write $d = \dim \mathcal{H}_A \times \dim \mathcal{H}_E = 4n$. Thus the total number of qubits in the data and ancillary space is $2 + \log_2(n)$.
2. **Communication rounds**
   For $i \in [0 \dots m]$, Bob has $V_i = P_i^\dagger D_i P_i$, expressed as $V_i^1 V_i^2 V_i^3$ from the decomposition of Proposition III.2. Then for each $j \in [1, 2, 3]$:
   (a) Bob picks randomly $2n$ unitaries $M_{i,1}^j, \dots, M_{i,2n}^j \in \mathbb{U}_2$, and sends to Alice through a classical channel $Q_i^j = \oplus_{k=1}^{2n} M_{i,k}^j S_{\sigma_j} P_i$
   (b) Alice builds a random one-time pad block, K_i^j diagonal in the basis Q_i^j, then applies it to $|\gamma_{i,j}\rangle$ and sends the state to Bob using BQC protocols with a server (it may be Bob)
   (c) Bob applies V_i^j, sends back the state
   (d) Alice applies $(K_i^j)^\dagger$ using BQC protocols with a server (it may be Bob)
   (e) If $i < m, j = 3$, Alice applies $U \otimes \mathbb{1}_E$.
   (f) After this round we have the state $|\gamma_{i,j+1}\rangle$ ($|\gamma_{i+1,j}\rangle$ if $j = 3$).

III.4.3 Third protocol - commuting matrices with correction

As explained above, we showed that as Bob has control on the basis over which Alice encrypts her communications, then this set could be sufficiently big enough such that Bob has access to all settings of Alice's data (or he could just cheat and choose one that achieves this condition). The problem here is that we give the choice of the encryption basis to Bob. However, if we let Alice choose one basis, Q, such that she encrypts her data with a key on this basis, then if she keeps the same basis for every computation round that uses the same input, Bob could at most learn the diagonal coefficients of $QUQ^\dagger$. This is represented in Protocol 3.

On the other hand, this requires Bob to apply matrices only in this basis, and therefore he cannot do his computation as before. This means that a correction needs to be done at the end of each computation round. We can still use above framework to allow Bob to apply a more complex computation. The idea is the following: for each round i, Bob will choose a random matrix whose diagonalization basis is Q, denoted W_i. Then they communicate as in the protocol defined above to compute W_i. Finally, at the end, Bob sends classically $V_i W_i^\dagger$ to Alice, which she applies using any DQC scheme. However, this means that Alice will learn information from V_i and not from the diagonalization basis.

Theorem III.6 (Security results for Protocol 3)**.** *Let Alice and Bob communicate to achieve HODQC over Protocol 3, with a chosen basis Q. Then:*

- In one computation, Alice leaks a one-shot accesses to the quasi-depolarization of $Q|\psi\rangle\langle\psi|Q^\dagger$ and m accesses to to QUQ for $1 \leq i < m$, which means that it looks like a diagonal matrix with $2n$ diagonal coefficients, and each of them is given by the mean of two distinct diagonal coefficients of the original matrix. On multiple computations, Bob can recover all diagonal coefficients, through tomography, of $Q|\psi\rangle\langle\psi|Q^\dagger$ and $QUQ^\dagger$.

- In one computation, Bob leaks quantum one-shot accesses to each V_i and to each W_i^j, and leaks classically a $\tilde{V}_i$, where each column of V_i is multiplied by a unit scalar. On multiple computations, Alice is able to learn via tomography each V_i, but the amount of information that she gets from the classical leak does not increase.

Protocol 3 HODQC where Bob has $\mathcal{D}(f)$ - commuting matrices with correction protocol

1. **Alice's preparation**
   (a) Alice has a state $|\psi\rangle \in \mathcal{H}_A$ and a unitary $U \in \mathcal{L}(\mathcal{H}_A)$.
   (b) She prepares the state on the data and ancillary state $|\gamma_{0,0}\rangle = |\psi\rangle \otimes |0\rangle \in \mathcal{L}(\mathcal{H}_A \otimes \mathcal{H}_E)$
   (c) We consider working with qubits, so we can write $d = \dim \mathcal{H}_A \times \dim \mathcal{H}_E = 4n$. Thus the total number of qubits in the data and ancillary space is $2 + \log_2(n)$.
   (d) She has a matrix $Q \in \mathcal{L}(\mathcal{H}_A \otimes \mathcal{H}_E)$, which is picked randomly beforehand and stays the same for multiple uses of the protocol with the same inputs.
2. **Communication rounds**
   For $i \in [0 \dots m]$, Bob has V_i, and picks randomly $W_i = QD_iQ^\dagger$, expressed as $W_i^1 W_i^2 W_i^3$ from the decomposition of Proposition III.2. Then for each $j \in [1, 2, 3]$:
   (a) The basis over which they communicate is $Q^j = S_{\sigma_j} Q$
   (b) Alice builds a random one-time pad, K_i^j block diagonal in the basis Q^j, then applies it to $|\gamma_{i,j}\rangle$ using DQC protocols with a server (it may be Bob), and sends the state to Bob
   (c) Bob applies W_i^j, sends back the state
   (d) Alice applies $(K_i^j)^\dagger$ using DQC protocols with a server (it may be Bob)
   (e) If $i < m, j = 3$, Bob sends to Alice classically $V_i W_i^\dagger$, which she applies through a DQC protocol with a server, and then she applies $U \otimes \mathbb{1}_E$.
   (f) After this round we have the state $|\gamma_{i,j+1}\rangle$ ($|\gamma_{i+1,j}\rangle$ if $j = 3$).

III.5 Enhanced security in restricted settings

As shown in Theorem III.2, Bob cannot encrypt his data. Therefore, Alice could always cheat and try to learn those matrices through tomography. Moreover, Bob has no way of checking the correctness of the computation like Alice, because he does not get the output, and Alice could input whatever she wants.Therefore, in order to make Bob's information secure, we need to find a way to randomize it each time Alice calls it, so that she cannot learn anything by tomography.

Let's consider the special case where f is either homomorphic or antihomomorphic, i.e. respectively $f(AB) = f(A)f(B)$ or $f(AB) = f(B)f(A)$. Some work has already been done to find supermaps for a functional transformation f that satisfies this condition [28]; moreover, the inverse, conjugation and transposition functions for which supermaps have been found satisfy this property. The idea is to add on the data qubits a random unitary before and after applying U, and to use the property of f to cancel it out. For the ancillary qubit, Bob can always apply a random unitary before sending it to Alice, and then to apply its inverse in order to keep the computation correct. More formally, at the beginning of the protocol, Bob chooses two random unitaries C, D, and a set of random unitaries $\{Q_i\}_{1 \le i \le n}$. Then, the matrices that Bob will take for his decomposition are given by Equation III.23:

$$\begin{cases} \tilde{V}_i = (D \otimes Q_i^\dagger) V_i (C \otimes Q_{i+1}) \\ \tilde{V}_0 = (E \otimes \mathbb{1}) V_0 (C \otimes Q_1) \\ \tilde{V}_n = (D \otimes Q_n) V_0 (F \otimes \mathbb{1}) \end{cases}, \text{ with } (E, F) = \begin{cases} (f(C)^\dagger, f(D)^\dagger) & \text{if } f \text{ is homomorphic} \\ (f(D)^\dagger, f(C)^\dagger) & \text{if } f \text{ is antihomomorphic} \end{cases} \tag{III.23}$$

This adds some randomness, and Alice won't be able to do tomography to recover the matrices V_i. This can therefore be used with the previous protocols to build a modified protocol where Alice is cheating, and she cannot learn anything via tomography. This is shown in Figure III.3.

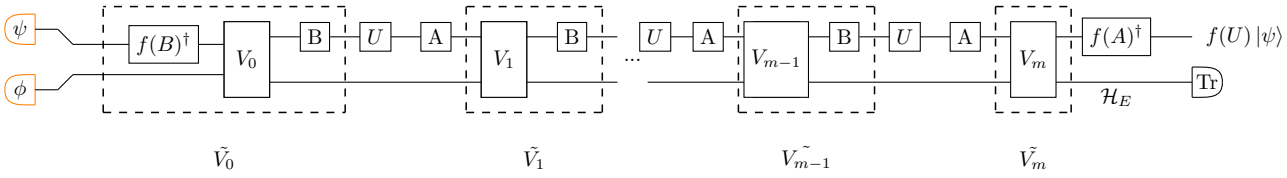

Figure III.4: A protocol when both parties are trying to cheat, in the special case where f is homomorphic. In this protocol, Bob keeps the ancillary wire.

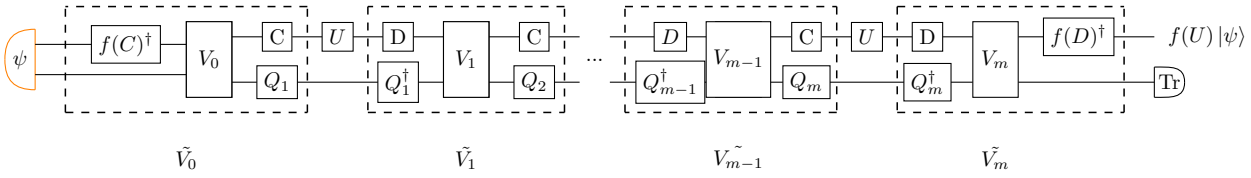

Figure III.3: A protocol when Alice is trying to steal Bob's information.

Moreover, Alice could also do the same. She chooses a random matrix to apply before / after U. She therefore applies the same one-time pad through the whole computation to her state. Up to tomography within this computation, Bob cannot learn the state that he receives, because it will change for each computation. This is described in Figure III.4. A and B are matrices randomly chosen respectively by Alice and Bob, and Alice applies $\tilde{U} = UA$ ($\tilde{U} = AU$ if f was antihomomorphic). This protocol has some side effects, which is that Alice's input state is not private, and V_0 or V_n (depending on f) could be learned by Alice. But this also has a big positive effect, which is that now Alice only needs access to the data qubits, and Bob can keep the ancillary state.

Theorem III.7 (Security results for Protocol 3)**.** *Let Alice and Bob communicate to achieve HODQC over Protocol 3, in the specific case when f is homomorphic. Then:*

- In one computation, Alice leaks one-shot accesses, namely one to her input state $|\psi\rangle$, and m accessses to a unitary channel UA, with U her input black-box unitary and A a randomly chosen unitary.

- In one computation, Bob leaks one-shot accesses, namely one to $\mathrm{Tr}_{\mathcal{H}_E}[V_i(B \otimes \mathbb{1}_E)]$ for $1 \leq i < m$, one to $\mathrm{Tr}_{\mathcal{H}_E}[V_m]$ and one to $\mathrm{Tr}_{\mathcal{H}_E}[(f(B)^\dagger \otimes \mathbb{1}_E)V_0(B \otimes \mathbb{1}_E)]$, where B is a randomly chosen unitary.

The partial trace for Bob's leaks comes from the fact that he is now not obligated to send back the ancillary qubits, which is both easier, because less qubits are sent, and safer, because Alice can learn less information. Although some data will inevitably leak, namely $|\psi\rangle$ and $\mathrm{Tr}_{\mathcal{H}_E}[V_m]$ through tomography, as both parties choose their unitaries A, B randomly, doing tomography on this rest of this type of data (i.e. UA, $V_1(B \otimes \mathbb{1}_E)$, etc.) will not be useful as long as they do not know the key used. Those leaks happen because of the structure of the encoding: Alice does not apply A on $|\psi\rangle$ and thus Bob has an unencrypted access to it, same for V_m for Bob. Especially, even if Bob was to cheat during the whole round, and had enough accessess to UA to learn it closely enough, A is unknown, and thus this gives him nothing about U. However, we still see that this cannot be written with the abstract cryptography framework, because their choice of key A, B is from a non finite set, and thus cannot be delayed in the same fashion as a quantum one-time pad, and we cannot write a simulator that does not get the leakage of the chosen key. Still, even if it is only in the special case of homomorphic and anti-homomorphic unitaries, it gives an idea of how to continue this project to mitigate the leaks, namely bringing higher-order results to improve the security.

Chapter IV

Conclusions and Outlook

In this thesis, we have tried to define the foundations of HODQC, by using the existing frameworks in the field of delegated quantum computation. We have shown that in this framework definition, it seems impossible to have an ideal resource with no leakage for both parties, although we saw that we could try to enhance the security of the leakages in some special cases. In particular, Bob's data cannot be encrypted over one use of Alice's unitary. Moreover, Alice gets the final result, and Bob has no way of verifying that Alice is following the protocol honestly. Alice also has some leakage; even if we try to limit it, it seems hard for Alice to ensure security of her unitary, although she is able to insert some test computations to trick Bob and try to detect when he cheats. While Protocol 1 was more of a toy protocol that show its limits very rapidly, i.e. when Bob has a quantum memory, Protocol 2 seemed more promising. However, depending on Bob's decomposition, he could be able to reconstruct Alice's state for different bases. To paliate this problem, we tried to provide a protocol where Alice chooses the basis in which she encodes her state for the whole time she communicates with Bob, Protocol 3, and Bob would share classically a correction matrix to Alice. Although Alice's security is now more controlled, the classical information that Bob has to leak is now much bigger.

We also tried to build a protocol where each party does an encryption separately, and at the end of the protocol they reconstruct the correct output together. However, if we do not have any higher-order result for the current supermap, this means that they are supposed to cancel the other party's encryption, which means that they ultimately learn the key from the other party, and the security is cancelled near the end of the protocol. Note that we have not considered the action of a third-party in this thesis, because most of the protocols that we considered with a secure third-party can be rewritten as a two-party protocol with one or two honest parties. We also did not consider the intervention of a malicious Eve, because as Bob and Alice do not trust each other, Eve's action of trying to learn or disrupt the computation of one party can be considered part of the action of the other party.

All the protocols considered in this thesis (except Protocol 1, where there is a security result when Bob has no quantum memory) are considering that both adversaries are completely malicious, i.e. they are able to cheat as much as they want. Although considering honest parties is not really interesting (Bob can just share his decomposition with an honest Alice, Alice can send her states unencrypted to an honest Bob), we also tried to consider a subclass of malicious adversaries called specious adversaries [50]. A specious adversary has a quantum memory, and is able to not follow the protocol; however, at any time during the protocol, he must be able, through a transformation, to produce back a state which is arbitrary close to the honest state. While our protocols offer the same " resistance " to specious and malicious adversaries (either none for Protocol 1, or the same coefficients in the basis for Protocol 2), and we have not found a protocol that would benefit from this definition, maybe we could find some adversaries that exist in the litterature, or define adversaries that are coherent with the current context, to provide security in this context.

Another thing worth mentioning is that in all the protocols considered (except from Section III.5), the quantum states that were sent back and forth by each party was on the whole space, i.e. including the unitary. This seemed to be required to be able to achieve an encryption scheme for Alice that was relevant, and would need to be proven more rigorously. However, this

is not ideal, because not only does it increase the size of the computation, but also it could be considered as a big leak for Bob. Even though he could always apply some matrix on the ancillary before sending it to Alice, and then apply it back to cancel it, such that it perturbes Alice, it does not seem a good practice to do, and further protocols could benefit from Bob keeping the ancillary state.

In this thesis, we have tried to explore the space of all possible protocols that could be built, such that they have no leakage for both parties. Even with the explanation about how we built and chose the protocols, there would need to be a theorem to fundamentally prove that there must be leakage for both parties. Nevertheless, is this leakage definition the best possible in our case : consider Section III.5. Indeed, we have built a protocol that does not do any proper encryption, and all accesses to U or V_i are unencrypted (up to Alice's input state and one matrix from the decomposition). However, as Bob and Alice each time choose a random matrix to apply to their gates, it seems impossible to recover for any of them any relevant data through tomography. Therefore, maybe the cryptographic framework that we considered in this thesis is too strong, and we could try to consider or build a weaker one, in which the leaks themselves are less important than how much an adversary learns in multiple rounds.

In any case, it seems that we would need higher-order results to mitigate the leaks. In the example of a homomorphic function f (same can be done for anti-homomorphic), we found an encoding scheme that utilises the properties of f. For other functions, finding ways to do a similar procedure would be a good way to continue this project, either by exhibiting such functions on a case-by-case basis, or by doing it for all functions of unitaries (even if it is not constructive). Bob could also try to decompose his function f into a composition or concatenation of functions that can be written into homomorphic / anti-homomorphic supermaps, even if it seems hard to do and would increase a lot the depth of the computation. Another way of adding this extra randomness could be done if Bob finds multiple decompositions for one function f, i.e. S_1, S_2 with the same number of slots such that they are equal on unitaries: $\forall U,\ S_1(U^{\otimes m}) = S_2(U^{\otimes m}) = f(U)$. This is already the case for the unitary inversion, where two supermaps have been found by different teams, and they do not seem to have anything in common [23, 24]. As Alice would not be able to distinguish between both decompositions, it would seem hard for her to do tomography to recover the supermaps.

Bibliography

[1] Andrew M. Childs. In: *Quantum Information and Computation* 5.6 (Sept. 2005). ISSN: 1533-7146. DOI: 10.26421/qic5.6. URL: http://dx.doi.org/10.26421/QIC5.6.

[2] Anne Broadbent, Joseph Fitzsimons, and Elham Kashefi. "Universal Blind Quantum Computation". In: *2009 50th Annual IEEE Symposium on Foundations of Computer Science*. IEEE, Oct. 2009. DOI: 10.1109/focs.2009.36. URL: http://dx.doi.org/10.1109/FOCS.2009.36.

[3] Robert Raussendorf and Hans J. Briegel. "A One-Way Quantum Computer". In: *Phys. Rev. Lett.* 86 (22 May 2001), pp. 5188–5191. DOI: 10.1103/PhysRevLett.86.5188. URL: https://link.aps.org/doi/10.1103/PhysRevLett.86.5188.

[4] Dorit Aharonov, Michael Ben-Or, and Elad Eban. *Interactive Proofs For Quantum Computations*. 2008. arXiv: 0810.5375 [quant-ph]. URL: https://arxiv.org/abs/0810.5375.

[5] Tomoyuki Morimae and Keisuke Fujii. "Blind quantum computation protocol in which Alice only makes measurements". In: *Phys. Rev. A* 87 (5 May 2013), p. 050301. DOI: 10.1103/PhysRevA.87.050301. URL: https://link.aps.org/doi/10.1103/PhysRevA.87.050301.

[6] Joseph F. Fitzsimons. "Private quantum computation: an introduction to blind quantum computing and related protocols". In: *npj Quantum Information* 3.1 (June 2017), p. 23. ISSN: 2056-6387. DOI: 10.1038/s41534-017-0025-3. URL: https://doi.org/10.1038/s41534-017-0025-3.

[7] Ueli Maurer and Björn Tackmann. "On the Soundness of Authenticate-then-Encrypt: Formalizing the Malleability of Symmetric Encryption". In: *Proceedings of the 17th ACM Conference on Computer and Communication Security*. Ed. by Angelia D. Keromytis and Vitaly Shmatikov. ACM. ACM, Oct. 2010, pp. 505–515.

[8] Vedran Dunjko et al. "Composable Security of Delegated Quantum Computation". In: *Advances in Cryptology – ASIACRYPT 2014*. Springer Berlin Heidelberg, 2014, pp. 406–425. ISBN: 9783662456088. DOI: 10.1007/978-3-662-45608-8_22. URL: http://dx.doi.org/10.1007/978-3-662-45608-8_22.

[9] Ueli Maurer and Renato Renner. "Abstract Cryptography". In: *Innovations in Computer Science - ICS 2011, Tsinghua University, Beijing, China, January 7-9, 2011. Proceedings*. Ed. by Bernard Chazelle. Tsinghua University Press, 2011, pp. 1–21. URL: http://conference.iiis.tsinghua.edu.cn/ICS2011/content/papers/14.html.

[10] Anne Broadbent. "Delegating private quantum computations". In: *Canadian Journal of Physics* 93.9 (Sept. 2015), pp. 941–946. ISSN: 1208-6045. DOI: 10.1139/cjp-2015-0030. URL: http://dx.doi.org/10.1139/cjp-2015-0030.

[11] Lucien Hardy. "Probability theories with dynamic causal structure: a new framework for quantum gravity". 2005. arXiv: gr-qc/0509120.

[12] Lucien Hardy. "Quantum gravity computers: On the theory of computation with indefinite causal structure". In: *Quantum reality, relativistic causality, and closing the epistemic circle*. Springer, 2009, pp. 379–401. DOI: 10.1007/978-1-4020-9107-0_21. arXiv: quant-ph/0701019.

[13] Giulio Chiribella et al. "Quantum computations without definite causal structure". In: *Phys. Rev. A* 88 (2013), p. 022318. DOI: 10.1103/PhysRevA.88.022318. arXiv: 0912.0195. URL: https://doi.org/10.1103/PhysRevA.88.022318.

[14] Ognyan Oreshkov, Fabio Costa, and Časlav Brukner. "Quantum correlations with no causal order". In: *Nat. Commun.* 3.1 (2012), p. 1092. DOI: 10.1038/ncomms2076. arXiv: 1105.4464.

[15] Nikola Paunkovi'c and Marko Vojinovi'c. "Causal orders, quantum circuits and spacetime: distinguishing between definite and superposed causal orders". In: *Quantum* 4 (2020), p. 275. DOI: 10.22331/q-2020-05-28-275. arXiv: 1905.09682.

[16] Ognyan Oreshkov. "Time-delocalized quantum subsystems and operations: on the existence of processes with indefinite causal structure in quantum mechanics". In: *Quantum* 3 (2019), p. 206. DOI: 10.22331/q-2019-12-02-206. arXiv: 1801.07594.

[17] Venkatesh Vilasini and Renato Renner. "Embedding cyclic causal structures in acyclic spacetimes: no-go results for process matrices". 2022. arXiv: 2203.11245.

[18] Nick Ormrod, Augustin Vanrietvelde, and Jonathan Barrett. "Causal structure in the presence of sectorial constraints, with application to the quantum switch". In: *Quantum* 7 (2023), p. 1028. DOI: 10.22331/q-2023-06-01-1028. arXiv: 2204.10273.

[19] Giulio Chiribella et al. "Quantum computations without definite causal structure". In: *Physical Review A* 88.2 (Aug. 2013). DOI: 10.1103/physreva.88.022318. URL: https://doi.org/10.1103%2Fphysreva.88.022318.

[20] Gilad Gour. "Comparison of Quantum Channels by Superchannels". In: *IEEE Transactions on Information Theory* 65.9 (2019), pp. 5880–5904. DOI: 10.1109/TIT.2019.2907989.

[21] Marco Túlio Quintino et al. "Probabilistic exact universal quantum circuits for transforming unitary operations". In: *Phys. Rev. A* 100 (6 Dec. 2019), p. 062339. DOI: 10.1103/PhysRevA.100.062339. URL: https://link.aps.org/doi/10.1103/PhysRevA.100.062339.

[22] Satoshi Yoshida, Akihito Soeda, and Mio Murao. "Reversing Unknown Qubit-Unitary Operation, Deterministically and Exactly". In: *Physical Review Letters* 131.12 (Sept. 2023). ISSN: 1079-7114. DOI: 10.1103/physrevlett.131.120602. URL: http://dx.doi.org/10.1103/PhysRevLett.131.120602.

[23] Yin Mo et al. *Parameterized quantum comb and simpler circuits for reversing unknown qubit-unitary operations*. 2024. arXiv: 2403.03761 [quant-ph]. URL: https://arxiv.org/abs/2403.03761.

[24] Yu-Ao Chen et al. *Quantum Advantage in Reversing Unknown Unitary Evolutions*. 2024. arXiv: 2403.04704 [quant-ph]. URL: https://arxiv.org/abs/2403.04704.

[25] Miguel Navascués. "Resetting uncontrolled quantum systems". In: *Phys. Rev. X* 8.3 (2018), p. 031008. DOI: 10.1103/PhysRevX.8.031008. arXiv: 1710.02470.

[26] Marco Túlio Quintino et al. "Reversing Unknown Quantum Transformations: Universal Quantum Circuit for Inverting General Unitary Operations". In: *Phys. Rev. Lett.* 123 (21 2019), p. 210502. DOI: 10.1103/PhysRevLett.123.210502. arXiv: 1810.06944. URL: https://link.aps.org/doi/10.1103/PhysRevLett.123.210502.

[27] David Trillo, Benjamin Dive, and Miguel Navascués. "Translating uncontrolled systems in time". In: *Quantum* 4 (2020), p. 374. DOI: 10.22331/q-2020-12-15-374. arXiv: 1903.10568.

[28] Marco Túlio Quintino and Daniel Ebler. "Deterministic transformations between unitary operations: Exponential advantage with adaptive quantum circuits and the power of indefinite causality". In: *Quantum* 6 (Mar. 2022), p. 679. ISSN: 2521-327X. DOI: 10.22331/q-2022-03-31-679. URL: http://dx.doi.org/10.22331/q-2022-03-31-679.

[29] D. Trillo, B. Dive, and M. Navascués. "Universal Quantum Rewinding Protocol with an Arbitrarily High Probability of Success". In: *Phys. Rev. Lett.* 130 (11 Mar. 2023), p. 110201. DOI: 10.1103/PhysRevLett.130.110201. arXiv: 2205.01131. URL: https://link.aps.org/doi/10.1103/PhysRevLett.130.110201.

[30] Peter Schiansky et al. "Demonstration of universal time-reversal for qubit processes". In: *Optica* 10.2 (2023), p. 200. DOI: 10.1364/OPTICA.469109. arXiv: 2205.01122.

[31] Yin Mo et al. "Parameterized quantum comb and simpler circuits for reversing unknown qubit-unitary operations". 2024. arXiv: 2403.03761.

[32] Tatsuki Odake, Satoshi Yoshida, and Mio Murao. "Analytical lower bound on the number of queries to a black-box unitary operation in deterministic exact transformations of unknown unitary operations". 2024. arXiv: 2405.07625.

[33] Jisho Miyazaki, Akihito Soeda, and Mio Murao. "Complex conjugation supermap of unitary quantum maps and its universal implementation protocol". In: *Physical Review Research* 1.1 (Aug. 2019). ISSN: 2643-1564. DOI: 10.1103/physrevresearch.1.013007. URL: http://dx.doi.org/10.1103/PhysRevResearch.1.013007.

[34] Daniel Ebler et al. "Optimal Universal Quantum Circuits for Unitary Complex Conjugation". In: *IEEE Trans. Inf. Theory* 69.8 (2023), pp. 5069–5082. DOI: 10.1109/TIT.2023.3263771. arXiv: 2206.00107.

[35] Giulio Chiribella and Hlér Kristjánsson. "Quantum Shannon theory with superpositions of trajectories". In: *Proceedings of the Royal Society A* 475.2225 (2019), p. 20180903.

[36] Qingxiuxiong Dong et al. "Controlled quantum operations and combs, and their applications to universal controllization of divisible unitary operations". 2019. arXiv: 1911.01645.

[37] Mateus Araújo et al. "Quantum circuits cannot control unknown operations". In: *New J. Phys.* 16.9 (2014), p. 093026. DOI: 10.1088/1367-2630/16/9/093026. arXiv: 1309.7976.

[38] Alessandro Bisio, Michele Dall'Arno, and Paolo Perinotti. "Quantum conditional operations". In: *Phys. Rev. A* 94.2 (2016), p. 022340. DOI: 10.1103/PhysRevA.94.022340. arXiv: 1509.01062.

[39] Mehdi Soleimanifar and Vahid Karimipour. "No-go theorem for iterations of unknown quantum gates". In: *Phys. Rev. A* 93.1 (2016), p. 012344. DOI: 10.1103/PhysRevA.93.012344. arXiv: 1510.06888.

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

[41] Michele Mosca, Alain Tapp, and Ronald de Wolf. *Private Quantum Channels and the Cost of Randomizing Quantum Information*. 2000. arXiv: quant-ph/0003101 [quant-ph].

[42] P. Oscar Boykin et al. *On Universal and Fault-Tolerant Quantum Computing*. 1999. arXiv: quant-ph/9906054 [quant-ph]. URL: https://arxiv.org/abs/quant-ph/9906054.

[43] G. Chiribella, G. M. D'Ariano, and P. Perinotti. "Transforming quantum operations: Quantum supermaps". In: *EPL (Europhysics Letters)* 83.3 (July 2008), p. 30004. ISSN: 1286-4854. DOI: 10.1209/0295-5075/83/30004. URL: http://dx.doi.org/10.1209/0295-5075/83/30004.

[44] Giulio Chiribella, Giacomo Mauro D'Ariano, and Paolo Perinotti. "Theoretical framework for quantum networks". In: *Physical Review A* 80.2 (Aug. 2009). ISSN: 1094-1622. DOI: 10.1103/physreva.80.022339. URL: http://dx.doi.org/10.1103/PhysRevA.80.022339.

[45] J. F. Poyatos, J. I. Cirac, and P. Zoller. "Complete Characterization of a Quantum Process: The Two-Bit Quantum Gate". In: *Phys. Rev. Lett.* 78 (2 Jan. 1997), pp. 390–393. DOI: 10.1103/PhysRevLett.78.390. URL: https://link.aps.org/doi/10.1103/PhysRevLett.78.390.

[46] Isaac L. Chuang and Michael A. Nielsen. "Prescription for experimental determination of the dynamics of a quantum black box". In: *Journal of Modern Optics* 44 (1996), pp. 2455–2467. URL: https://api.semanticscholar.org/CorpusID:119497365.

[47] Hoi-Kwong Lo. "Insecurity of quantum secure computations". In: *Physical Review A* 56.2 (Aug. 1997), pp. 1154–1162. ISSN: 1094-1622. DOI: 10.1103/physreva.56.1154. URL: http://dx.doi.org/10.1103/PhysRevA.56.1154.

[48] Daniel Gottesman and Isaac L. Chuang. "Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations". In: *Nature* 402.6760 (Nov. 1999), pp. 390–393. ISSN: 1476-4687. DOI: 10.1038/46503. URL: http://dx.doi.org/10.1038/46503.

[49] Alfred Haar. "Der Massbegriff in der Theorie der Kontinuierlichen Gruppen". In: *Annals of Mathematics* 34 (1933), p. 147. URL: https://api.semanticscholar.org/CorpusID:124917543.

[50] Frédéric Dupuis, Jesper Buus Nielsen, and Louis Salvail. "Secure Two-Party Quantum Evaluation of Unitaries against Specious Adversaries". In: *Advances in Cryptology – CRYPTO 2010*. Springer Berlin Heidelberg, 2010, pp. 685–706. ISBN: 9783642146237. DOI: 10.1007/978-3-642-14623-7_37. URL: http://dx.doi.org/10.1007/978-3-642-14623-7_37.

Appendices

Appendix A

Proofs

We have shown above that we can't get rid of some trace of blocks of the input state. However, we here show how to do it when there is only one communication round, and the input state is known by Alice. This requires a perfect knowledge of the state Alice is in possession of, and as she does not know what computation Bob is doing, she loses the knowledge of her state after one round.

Proposition A.1. *Let A be a unitary matrix of size $4n \times 4n$, with diagonalization $A = P^\dagger DP$, with D being a matrix with $2n$ eigenvalues of multiplicity 2, grouped together. Then for each U_i, we can find a set of matrices $\{K_i\}$ that commute with A, hide information about P, and form a one-time pad.*

Lemma A.1. *Let $n \geq 1, p \in \mathbb{Z}^*$. Let $(\alpha_k)_{1\leq k\leq n} \in (\mathbb{Z}^*)^n$ such that they are distinct pairwise. We denote the n-dimensional simplex as $D_n = \{(x_i)_{1\leq i\leq n} \in \mathbb{R}^n : \sum_{i=1}^n x_i \leq 1 and\ \forall i, x_i \geq 0, \}$. Then $I_n = \int_{D_n} e^{2i\pi \sum_{k=1}^n \alpha_k x_k} dx = 0$.*

Proof. We can prove this by induction on n. The base case $n = 1$ comes directly from the calculus of the integral.

$$\forall \alpha \in \mathbb{Z}^*, p \in \mathbb{Z}^*, I_1 = \int_{x=0}^{1} e^{2i\pi p\alpha x} dx = \frac{1}{2i\pi p\alpha}(e^{2i\pi p\alpha} - 1) = 0$$

Assume it holds for $n = m$. Let $p \in \mathbb{Z}\backslash\{0\}, (\alpha_k)_{1\leq k\leq m+1} \in (\mathbb{Z}^*)^{m+1}$. Then we can split this integral in 2, knowing that if we integrate on D_{m+1}, it is the same as integrating on D_m and integrating x_{m+1} on $[0, 1-\sum_{k=1}^m x_k]$. We denote $I_m(a_1, ..., a_m)$ the integral I_m in the case $\alpha_i = a_i$. Then we have the following relation:

$$\begin{aligned} I_{m+1} &= \int_{D_{m+1}} e^{2i\pi \sum_{k=1}^{m+1} \alpha_k x_k} dx_1 \dots dx_{m+1} = \int_{D_m} \int_0^{1-\sum_{k=1}^m x_k} e^{2i\pi \sum_{k=1}^{m+1} \alpha_k x_k} dx_{m+1} dx_1 \dots dx_m \\ &= \frac{1}{2i\pi p\alpha_{m+1}} \int_{D_m} e^{2i\pi p \sum_{k=1}^m (\alpha_k - \alpha_{m+1}) x_k} e^{2i\pi p\alpha_{m+1}} - e^{2i\pi p \sum_{k=1}^m \alpha_k x_k} dx_1 \dots dx_m \\ &= \frac{1}{2i\pi p\alpha_{m+1}} (e^{2i\pi p\alpha_{m+1}} I_m(\alpha_1 - \alpha_{m+1}, \dots, \alpha_m - \alpha_{m+1}) - I_m) \end{aligned}$$

We see that I_{m+1} is split into two parts of I_m. By hypothesis, the second one is 0, and the first one is 0 as well, because $\alpha_i \neq \alpha_j \implies \alpha_i - \alpha_{m+1} \neq \alpha_j - \alpha_{m+1}$, so it verifies the condition for the induction. Therefore, $I_{m+1} = 0$.

□

Now we can prove the proposition from above.

Proof from Proposition A.1. We denote $K_i \in \{\mathbb{1}, X, Y, Z\}$ one of the four Pauli matrices. For any matrix B of size 2×2, $\frac{1}{4}\sum_{i=1}^4 K_i A K_i^\dagger = Tr(B)\frac{I_2}{2}$. We know that any matrix that commutes with A can be written as $P^\dagger \oplus_{i=1}^{2n} M_i P$. We saw that this means that Alice can only learn P

up to a matrix, so we can act as if Bob sends to Alice $\mathcal{B} = P \oplus_{i=1}^{2n} P_i$, with P_i being random unitaries. As $\mathcal{B}\rho\mathcal{B}^\dagger$ is a state of size $4n$, we can write it as a matrix with $4n^2$ blocks of size 2×2, i.e. $\mathcal{B}\rho\mathcal{B}^\dagger = (\rho_{i,j})_{1 \leq i,j \leq 2n}$. Alice picks randomly one K_i, and she computes $x_k = Tr(\rho_{k,k})$ for $1 \leq k \leq 2n-1$. Then the matrix that she applies is $M_{i,x_1,\dots,x_{2n-1}} = \mathcal{B}^\dagger \oplus_{j=1}^{2n} U_j K_i \mathcal{B}$, with U_j being a diagonal matrix with eigenvalues $e^{2i\pi j(\sum_{k=1}^{2n-1} k x_k)}, e^{-2i\pi j(\sum_{k=1}^{2n-1} k x_k)}$. This matrix is the right form to commute with $A1$, and from Bob's point of view, K_i and randomly picked, and the set of x_i is uniformly picked in the simplex D_{2n-1}. We denote $V_n(a)$ the volume of the n-dimensional simplex bounded by $a, a \in \mathbb{R}^+$. So by ponderating by the number of K_i and the volume of the simplex $V_{2n-1}(1) = \frac{1}{(2n-1)!}$, we get:

$$\begin{aligned}\rho' &= \int_{D_{2n-1}} \frac{(2n-1)!}{4} \sum_{i=1}^{4} \mathcal{M}_{i,x_1,\dots,x_{2n-1}} \rho M^\dagger_{i,x_1,\dots,x_{2n-1}} dx_1 \dots dx_{2n-1} \\ &= \int_{D_{2n-1}} \frac{(2n-1)!}{4} \sum_{i=1}^{4} \mathcal{B}^\dagger(\oplus_{j=1}^{2n} U_j K_i)\mathcal{B}\rho\mathcal{B}^\dagger(\oplus_{j=1}^{2n} U_j K_i)^\dagger \mathcal{B} dx_1 \dots dx_{2n-1}\end{aligned}$$

We want $\rho' = \frac{I_{4n}}{4n}$. Using the form of $\mathcal{B}\rho\mathcal{B}^\dagger$ defined above, we obtain $4n^2$ equations to check:

$$\forall j,l \in \{1,\dots,2n\}, \mathcal{B}^\dagger \int_{D_{2n-1}} \frac{(2n-1)!}{4} \sum_{i=1}^{4} U_j K_i \rho_{j,l} K_i^\dagger U_l^\dagger dx_1 \dots dx_{2n-1} \mathcal{B} = \frac{I_2}{4n}\delta_{i,j}$$

As $\mathcal{B} I_2 \mathcal{B}^\dagger = I_2$, then we can simplify the equation and remove the basis. For the diagonal terms, this yields, for all $j \in \{1,\dots,2n\}$, by recognizing the depolarizing channel:

$$\begin{aligned}\rho'_{j,j} &= \int_{D_{2n-1}} U_j \frac{(2n-1)!}{4} \sum_{i=1}^{4} K_i \rho_{j,j} K_i^\dagger U_j^\dagger dx_1 \dots dx_{2n-1} \\ &= \frac{(2n-1)!}{2} I_2 \int_{D_{2n-1}} Tr(\rho_{j,j}) U_j U_j^\dagger dx_1 \dots dx_{2n-1} \\ &= \frac{(2n-1)!}{2} I_2 \int_{D_{2n-1}} x_j dx_1 \dots dx_{2n-1} = \frac{(2n-1)!}{2} \int_{x_j=0}^{1} x_j V_{2n-2}(1-x_j) dx_j \\ &= \frac{1}{2} I_2 \int_{x_j=0}^{1} \frac{(2n-1)!}{(2n-2)!} x_j (1-x_j)^{2n-2} dx_j \underset{u=1-x}{=} \frac{2n-1}{2} \int_{u=0}^{1} u^{2n-2}(1-u) du \\ &= \frac{2n-1}{2} I_2 \int_{u=0}^{1} u^{2n-2} - u^{2n-1} du = \frac{2n-1}{2}\left[\frac{1}{2n-1} - \frac{1}{2n}\right] = \frac{1}{4n} I_2\end{aligned}$$

Now we need to check the off-diagonal blocks. For all $j,l \in \{1,\dots,2n\}, i \neq j$:

$$\begin{aligned}\rho'_{j,l} &= \int_{D_{2n-1}} U_j \frac{(2n-1)!}{4} \sum_{i=1}^{4} K_i \rho_{j,j} K_i^\dagger U_l^\dagger dx_1 \dots dx_{2n-1} \\ &= \frac{(2n-1)! Tr(\rho_{j,l})}{2} I_2 \int_{D_{2n-1}} U_j U_l^\dagger dx_1 \dots dx_{2n-1}\end{aligned}$$

As $\rho_{j,l}$ does not depend on x_i, then we can move it out the integral. Now we integrate over two diagonal matrix, so we need to check that the two diagonal coefficients yield 0. This means we need $\int_{D_{2n-1}} e^{\pm 2i\pi(j-l)(\sum_{k=1}^{2n-1} k x_k)} dx_1 \dots dx_{2n-1}$. By definition, this is an integral of the form of Lemma A.1, with $p = \pm(j-l) \in \mathbb{Z}^*$, and $\alpha_k = k$ that are distinct pairwise. Therefore, this integrates over 0, and we have $\rho'_{j,l} = 0$, which concludes the proof.

□