Heinrich-Heine-University Düsseldorf
Faculty of Mathematics and Natural Sciences

BACHELOR THESIS

Juan Manuel Henning*

Blind quantum computation including noise

Institute for Theoretical Physics III
Working Group of Prof. Bruß†

Supervisor: Ph.D. student Sarnava Datta‡
Study programm: Physics

November 2020

*Juan.Henning@uni-duesseldorf.de
†Dagmar.Bruss@uni-duesseldorf.de
‡dattas@uni-duesseldorf.de

Acknowledgements

This bachelor thesis would not have been possible without the help and advice from various people.

First, I would like to thank Prof. Dr. Dagmar Bruß and Dr. Hermann Kampermann for providing instrumental literature references and discussions that laid the foundation of the present work. I would also like to thank my supervisor, Ph.D. student Sarnava Datta, for his useful suggestions and advice.

I want to thank the members of the Institute of Theoretical Physics III at the Heinrich-Heine-University in Düsseldorf for the support and advice throughout this bachelor thesis' writing. I would particularly like to thank Ph.D. student Lennart Bittel, Ph.D. student Lucas Tendick, Ph.D. student Daniel Miller, and Dr. Federico Grasselli for their help, support, and for the engaging conversations.

Moreover, I would like to thank all my family for always being there for me. I would not be here without my parents' love and constant support.

Abstract

As with the development of classical computers, it is reasonable to believe that quantum computers will not be widely available once they become a reality. Thus it may be the case that only selected private and public parties will have the luxury of having one on-site. Blind quantum computation (BQC) protocols aim to make quantum computation more accessible by making it possible to delegate quantum computations to a remote untrusted quantum computer, i.e., the server. BQC protocols make the delegation of quantum computations possible while preventing the server's side from learning any meaningful information about the computation, input, and output. Thus the server's side is said to be "blind".

The delegated computation's fidelity and the overall "blindness" of the protocol will be affected by quantum noise in any realistic scenario. Noisy qubit state preparation, noisy transmission, and noisy quantum gate implementation are not perfect and must be considered.

We introduced a white-noise model to the efficient universal blind quantum computation protocol by Giovannetti et al. [15] and investigated how noise in the client's and the server's quantum system affects the final computation. More specifically, we numerically evaluate the von Neumann entropy of the server's memory for repeating instructions and the computation fidelity of a specific circuit, the Bell circuit, in the context of the protocol.

Contents

1 Basic Definitions

We will make some basic but useful definitions in this section, as well as present some notation conventions.

If and only if:	iff.
Equivalent, identical:	$\equiv$.
The set of complex numbers:	$\mathbb{C}$.
The set of real numbers:	$\mathbb{R}$.
The set of real, positive numbers:	$\mathbb{R}^+ \equiv \{x \vert x \in \mathbb{R}, x > 0\}$.
The set of real, non-negative numbers:	$\mathbb{R}_{\geq 0} \equiv \{x \vert x \in \mathbb{R}, x \geq 0\}$.
The set of natural numbers:	$\mathbb{N}$, $\mathbb{N}_0$, where $0 \notin \mathbb{N}$, $0 \in \mathbb{N}_0$.
The interval set [a,b) in K:	$[a, b) \equiv \{x \in K \vert a \leq x < b\}$,
analogously for [a,b], (a,b] and (a,b).	
The Cartesian product "$\times$" of sets X, Y:	$X \times Y \equiv \{(x, y) \vert x \in X, y \in Y\}$.
Function f maps from set X to set Y:	$f : X \to Y$.
Function f maps $x \in X$ to $f(x) \in Y$:	$f : x \mapsto f(x)$.
The inner product in Dirac-Notation:	$\langle \psi \vert \psi \rangle$
The outer product in Dirac-Notation:	$\vert \psi \rangle\langle \psi \vert$
The *dagger* operation:	$A^\dagger \equiv \overline{A^T}$
The 2×2 identity:	$\mathbb{1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

2 Introduction to quantum computation

Here the necessary concepts will be reviewed so that the reader may follow the discussion in sections 3, 4, and 5. References [1], [2], [3], [4], and [5] provide further insight into this section's topics. A refresher on complex numbers and vector spaces is included in appendices A and B.

2.1 The qubit

The indivisible unit of classical information is called the *bit*, which can take values 0 or 1. The corresponding unit of quantum information is called the quantum bit or *qubit*. The qubit is a two-dimensional quantum system, e.g., photon polarization or atomic spin, described by a two-dimensional complex Hilbert space $\mathcal{H}$, i.e., a complex vector space with an inner product. See appendix C for a more detailed description of Hilbert spaces. We choose two orthonormal qubit state vectors to span this space using Dirac-Notation: $|0\rangle$, $|1\rangle$. We call it the *computational basis*.

We represent the Ket-Vectors, $|0\rangle$ and $|1\rangle$, as column vectors. One possible choice of computational basis vectors would be

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

An arbitrary qubit state can be written as the superposition

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{2}$$

with complex numbers α and β. The state $|\psi\rangle$ must be normalized, i.e., $\langle\psi|\psi\rangle = |\alpha|^2 + |\beta|^2 = 1$. A Bra-Vector $\langle\psi|$ is defined as the Hermitian conjugate of a Ket-Vector $|\psi\rangle$, i.e., $\langle\psi| = |\psi\rangle^\dagger$. Consequently, Bra-Vectors will be represented by row vectors. Note that changing the state's global phase has no physical significance. See Eq. (13). For this reason, a general qubit state (2) can also be represented as

$$|\psi\rangle = \cos\frac{\theta}{2}\,|0\rangle + e^{i\phi}\sin\frac{\theta}{2}\,|1\rangle \tag{3}$$

by choosing $\alpha \geq 0$ and real, with $\pi \geq \theta \geq 0$ and $2\pi \geq \phi > 0$.

It is convenient to visualize qubit states through a unit sphere, the *Bloch Sphere*.

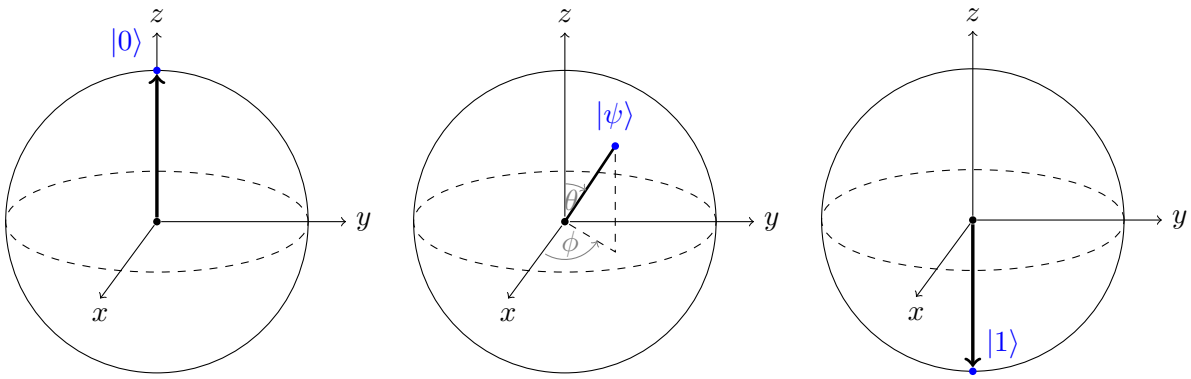

Figure 1: Graphical visualization of the states $|0\rangle$, $|\psi\rangle$ and $|1\rangle$ on the Bloch Sphere.

2.2 The density matrix

The density matrix ρ provides a more general way of representing quantum states. It is defined as:

$$\rho \equiv \sum_i p_i \, |\psi_i\rangle\langle\psi_i| \,, \tag{4}$$

where the probabilities $p_i \geq 0$ satisfy the condition $\sum_i p_i = 1$ and the vectors are elements of an n-dimensional Hilbert space, i.e., $|\psi_i\rangle \in \mathcal{H}^n$. The resulting $n \times n$ density matrix satisfies the following conditions:

Hermiticity:	$\rho = \rho^\dagger$,
Non-Negativity:	$\langle\psi| \, \rho \, |\psi\rangle \geq 0, \quad \forall \, |\psi\rangle \in \mathcal{H}^n$,
Unit Trace:	$Tr(\rho) = 1$.

Using spherical coordinates and Eqs. (3) and (4), we get the density matrix representation of a general qubit state

$$\rho = \frac{1}{2} \left(\mathbb{1} + \vec{\mathbf{r}} \cdot \vec{\sigma} \right), \tag{5}$$

where $\vec{\mathbf{r}} \in \mathbb{R}^3$ is the so-called *Bloch vector* and $\vec{\sigma} = (X, Y, Z)^T$ is the vector of Pauli matrices

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{6}$$

2.3 Pure states & mixed states

A *pure state* is a state or a superposition of states of the form (2). The density matrix can be written as $\rho = |\psi\rangle\langle\psi|$; therefore, it satisfies the projector property $\rho^2 = \rho$. Every point on the Bloch sphere's surface corresponds to a pure state since Eq. (5) and the projector property of pure state density matrices imply $|\vec{\mathbf{r}}| = 1$.

A *mixed state* is a statistical "mixture" of pure states. Therefore it cannot be represented by a single ket vector, only by the corresponding density matrix, as shown in (4). The system is in a state taken from the ensemble of k different states $\{|\psi_1\rangle, |\psi_2\rangle, ..., |\psi_k\rangle\}$ with probabilities $\{p_1, p_2, ..., p_k\}$. We write $\rho = \sum_k p_k \rho_k$, where ρ_k are density matrices of pure states and $\sum_k p_k = 1$. Using Eq. (5), the corresponding Bloch vector is a linear combination of Pure-State Bloch vectors $\vec{\mathbf{r}} = \sum_k p_k \vec{\mathbf{r_k}}$, which implies that the inequality $|\vec{\mathbf{r}}| < 1$ must hold for mixed states.

The density matrix described by $\vec{\mathbf{r}} = 0$, i.e., the normalized identity, is referred to as the maximally mixed state.

2.4 The depolarizing channel

Quantum systems are often subject to quantum noise. Even if the errors are small, they can build up and ultimately corrupt quantum computation output. *Error correction* techniques will be necessary in order to scale the size of reliable quantum computation. A good error model used as a benchmark for error correction is the so-called *depolarizing channel*. [5] [6]

The depolarizing channel, also referred to as depolarizing *noise map* or depolarizing *noise channel*, is a function that maps a density matrix ρ to $\Lambda(\rho)$ as

$$\Lambda(\rho) = (1-p)\rho + \frac{p}{d}\mathbb{1}\,. \tag{7}$$

The result is again a density matrix $\Lambda(\rho)$, p is called the noise parameter and d normalizes the identity. A quantum system ρ going through the depolarizing channel will remain in that state with probability $(1-p)$ and changes to the maximally mixed state $\mathbb{1}$ with a probability given by the noise parameter p.

2.5 Entanglement (pure states)

The Hilbert space of a bipartite quantum system is the tensor product of the Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$.

$$\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2 \tag{8}$$

A pure state $|\psi\rangle \in \mathcal{H}$ is *entangled* or *non-separable* if it is impossible to write it as a tensor product of states $|\nu\rangle_1 \in \mathcal{H}_1$ and $|\phi\rangle_2 \in \mathcal{H}_2$. On the other hand, if it is possible to write it as $|\psi\rangle = |\nu\rangle_1 \otimes |\phi\rangle_2$, then $|\psi\rangle$ is called *separable*.

Similarly, for an n-partite quantum system

$$\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \cdots \otimes \mathcal{H}_n. \tag{9}$$

We call the pure composite state $|\psi\rangle \in \mathcal{H}$ entangled [7] if it is not possible to write it as a tensor product of local states $|\phi_i\rangle$:

$$|\psi\rangle \neq \bigotimes_{i=1}^{k} |\phi_i\rangle \,, \tag{10}$$

with $n \geq k \geq 2$. The state is called *k-separable* if it is possible to write it as a tensor product of k local states.

2.6 Partial trace

Given a composite quantum system $\rho_{AB} \in \mathcal{H}_A \otimes \mathcal{H}_B$, it is possible to describe its local systems on $\mathcal{H}_A$ or $\mathcal{H}_B$ by taking the *partial trace*.

Tracing over subsystem B results on the local state described by $\rho_A \in \mathcal{H}_A$, given by

$$\rho_A = \mathrm{Tr}_B(\rho_{AB}) = \sum_{i=0}^{\dim(\mathcal{H}_B)-1} \mathbb{1}_A \otimes \langle i|_B (\rho_{AB}) \mathbb{1}_A \otimes |i\rangle_B \,, \tag{11}$$

where $\{|i\rangle_B\}$ forms a basis in $\mathcal{H}_B$. Similarly, tracing over subsystem A yields $\rho_B \in \mathcal{H}_B$.

2.7 Measurement formalism

The measurement process is the only way of extracting information from any given quantum state. Nielsen and Chuang [5] provide a good introduction to the measurement process in its general form, the projective measurement, and the positive operator-valued measure (POVM).

2.7.1 General quantum measurement

General quantum measurement is described by a set of measurement operators $\{M_m\}$, that act on the quantum system's space being measured and satisfy the equation $\sum_m M_m^\dagger M_m = \mathbb{1}$. The index m represents the possible measurement outcomes. If a quantum system in the state $|\psi\rangle$ is measured, then the probability of getting outcome m after the measurement is given by $p(m)$. The state of the system transforms to $|\psi'\rangle$ after the measurement:

$$p(m) = \langle\psi| M_m^\dagger M_m |\psi\rangle \,, \qquad |\psi'\rangle = \frac{M_m |\psi\rangle}{\sqrt{p(m)}}. \tag{12}$$

If the quantum system is described by a density operator ρ we can calculate the outcome probabilities by $p(m) = tr(M_m^\dagger M_m \rho)$. The density matrix transforms to $\rho' = \frac{M_m \rho M_m^\dagger}{p(m)}$ after the measurement.

We can see that the measurement statistics remain the same after a global phase transformation, since for the states $|\psi\rangle$ and $e^{i\phi}|\psi\rangle$ it holds

$$p(m) = \langle\psi| M_m^\dagger M_m |\psi\rangle = e^{i\phi} \cdot e^{-i\phi} \langle\psi| M_m^\dagger M_m |\psi\rangle = \left\langle e^{i\phi}\psi \right| M_m^\dagger M_m \left| e^{i\phi}\psi \right\rangle . \tag{13}$$

The global phase has no physical meaning for the statistics of the measurement.

2.7.2 Projective measurement

A projective measurement is described by an *observable* M, i.e., a measurable physical quantity such as energy, spin, and position. Observables are described by Hermitian operators, i.e., $M = M^\dagger$, that act on the space of the quantum system being measured. The observable's eigenvalues are real and correspond to the possible measurement outcomes. It is always possible to rewrite the observable in the diagonal form

$$M = \sum_m m P_m, \tag{14}$$

where P_m projects onto the eigenspace associated with eigenvalue m. If we consider the state $|\psi\rangle$, the probability of getting outcome m is given by $p(m)$. The state transforms to $|\psi'\rangle$ after the measurement:

$$p(m) = \langle\psi| P_m |\psi\rangle , \qquad |\psi'\rangle = \frac{P_m |\psi\rangle}{\sqrt{p(m)}}. \tag{15}$$

The projectors P_m are orthogonal $P_n P_m = \delta_{nm} P_n$ and satisfy $\sum_m P_m = \mathbb{1}$. We see that the projective measurement is the same as the general quantum measurement, for the special case where M_m are Hermitian and orthogonal projectors, i.e., $M_n M_m = \delta_{nm} M_n$.

2.7.3 POVM measurement

Consider a measurement described by measurement operators M_m on a quantum system in state $|\psi\rangle$. The probability of getting outcome m will be $p(m) = \langle\psi| M_m^\dagger M_m |\psi\rangle$. The set $\{E_m\}$ is called a POVM, with POVM elements $E_m = M_m^\dagger M_m$.

E_m is a positive operator and satisfies $\sum_m E_m = \mathbb{1}$ and $p(m) = \langle\psi| E_m |\psi\rangle$. Note that, in general, the POVM $\{E_m\}$ is not enough to determine the quantum system's state after the measurement. However, it is sufficient to determine the probabilities of the different outcomes m. Therefore, POVMs are useful for situations where only the measurement statistics matter.

2.8 The quantum circuit model and quantum gates

The circuit model of quantum computation is, like its classical counterpart, the idea of applying multiple gates, one after another, in order to get any desired operation. The time-evolution of a closed quantum system, i.e., a quantum system with no interaction with an external quantum system, is *unitary*. Consequently, we represent these gates by unitary matrices, i.e., matrices U that satisfy $UU^\dagger = U^\dagger U = \mathbb{1}$, that act on the quantum system. It is possible to find a finite set of operations from which any other possible operation can be constructed. We call it a *universal* set of gates.

Some of the most important gates are the three *Pauli* gates X, Y, Z, the *Hadamard* gate H, and the T gate. The *Controlled-U* gate C_U, with $U = \left(\begin{smallmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{smallmatrix}\right) \in \{X, Y, Z\}$, is also essential in quantum computing.

We represent these gates as 2×2 matrices in the $|0\rangle$, $|1\rangle$ basis and a 4×4 matrix in the $|00\rangle$, $|01\rangle$,$|10\rangle$, $|11\rangle$ basis:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad T = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{pmatrix}, \quad C_U = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{11} & u_{12} \\ 0 & 0 & u_{21} & u_{22} \end{pmatrix}. \tag{16}$$

Any single-qubit unitary operation can be constructed using only the Hadamard and the T gate. A universal set of gates for multiple qubits requires the C_X gate's addition. Thus the set $\{H, T, C_X\}$ is universal.

For example, we can build the so-called *Bell circuit*, which generates the entangled two-qubit states of the *Bell-basis* $\{|\phi^+\rangle, |\phi^-\rangle, |\psi^+\rangle, |\psi^-\rangle\}$.

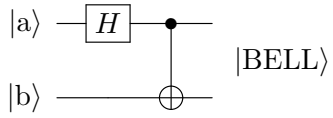

Figure 2: Quantum circuit representation of the Bell circuit acting on input qubits in the states $|a\rangle, |b\rangle\}$. A Hadamard gate acts on the first qubit, i.e., on $|\mathrm{a}\rangle$, and a C_X gate is applied with the first qubit as control. The circuit yields $|BELL\rangle \in \{|\phi^+\rangle, |\phi^-\rangle, |\psi^+\rangle, |\psi^-\rangle\}$ for $|a\rangle, |b\rangle \in \{|0\rangle, |1\rangle\}$.

We read the quantum circuit from left to right, following the qubit's wires. C_X operations are represented by a line that vertically connects two wires, with "$\bullet$" on the control qubit's wire and "$\oplus$" on the target qubit's wire. The C_X operation acts on the joint quantum system of the two qubits, i.e., $|a\rangle \otimes |b\rangle$. The unitary matrix operation on the first qubit is represented by a box, labeled by the Hadamard transformation H, on the qubit's wire. Note that H also acts on the joint quantum system as $H \otimes \mathbb{1}$.

The circuit maps the input qubits to the Bell-basis as follows.

$$\begin{aligned} |00\rangle \mapsto |\phi^+\rangle = \tfrac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \qquad |01\rangle \mapsto |\psi^+\rangle = \tfrac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \\ |10\rangle \mapsto |\phi^-\rangle = \tfrac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \qquad |11\rangle \mapsto |\psi^-\rangle = \tfrac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \end{aligned} \tag{17}$$

3 Blind quantum computation

Blind quantum computation (BQC) protocols can be one of the early widespread applications of theoretical quantum computation and communication. The main idea is to allow someone whose quantum processing capabilities are limited to privately delegate quantum computations to a remote universal quantum computer, i.e., the quantum server. The privacy and the correctness of the delegated computation may be compromised if the server is not trusted.

The delegated quantum computation is private if the server cannot obtain any meaningful information about the computation, input, or output. The server is, therefore, "blind" to the client's input. BQC protocols mainly aim to achieve privacy; however, many protocols also allow for verifiable computation.

The client may be purely classical or with limited quantum capabilities while delegating the computation to multiple quantum servers.

Client	**Quantum Server (QS)**
Classical	Single QS
Limited Quantum Capabilities	Multiple QS

Table 1: The table shows possible client and server types for a BQC protocol. A specific BQC protocol can be designed after choosing a client type (first column) and a Server type (second column).

The setting where a classical client communicates with a single quantum server is the most desirable. Anyone with a classical computer would be able to perform quantum computations on a quantum server in a way that does not reveal any relevant information to the server. However, the quantum server could store a transcript of the client's classical communication during the protocol and rerun it multiple times on their side, possibly revealing information about the client's computation. A solution to this problem

is to allow the client to communicate with multiple non-communicating quantum servers. However, it may be difficult to guarantee non-communicating servers in a realistic scenario [8].

Instead, we will focus on the setting where the client has limited quantum capabilities and communicates with a single quantum server in section 3.1. It may be reasonable to allow the client to have some quantum capabilities, like preparing or measuring single-qubit states, given the current state of quantum technologies [12].

3.1 The efficient universal blind quantum computation protocol

3.1.1 Quantum computation

As with all BQC protocols, the server is not trusted. The server is equipped with a quantum memory $\mathcal{M}$ in this particular protocol, initially in state $|\Psi\rangle_{\mathcal{M}} = |0\rangle^{\otimes N}$, and is capable of universal quantum computation by implementing a universal set of gates $\mathcal{G}$; we define G as the number of gates in $\mathcal{G}$.

The client has fewer quantum capabilities than the server. Her objective is to communicate with the server following the protocol to get the result of an arbitrary quantum computation.

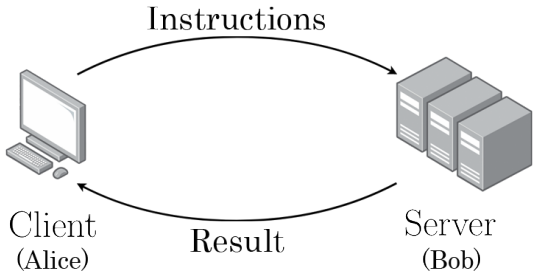

Figure 3: Sketch of the main idea. The client communicates with the server sending him a series of instructions to get the result of the quantum computation she would otherwise not be able to compute by herself.

The client can prepare qubits in the states $|0\rangle$, $|1\rangle$, $|+\rangle$ and $|-\rangle$ and is capable of measuring single qubits in the computational basis $\{|0\rangle, |1\rangle\}$ and in the tilted basis $\{|+\rangle, |-\rangle\}$. Note that the preparation and measurement of qubits in the $\{|+\rangle, |-\rangle\}$ basis is needed to guarantee the server's "blindness" and not for the computation itself.

We will now go step by step, through the computation process of the protocol.

Step 1:

Figure 4: The server initializes the memory $\mathcal{M}$ to the state $|\Psi\rangle_{\mathcal{M}} = |0\rangle^{\otimes N}$ and waits for incoming communication from the client.

Step 2:

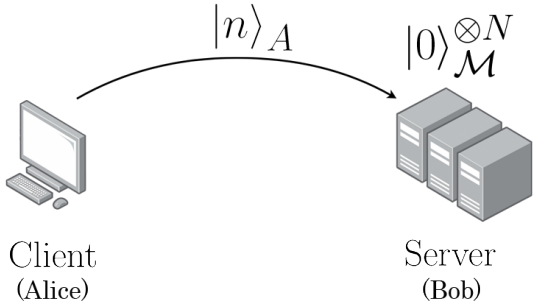

Figure 5: The client sends one computation instruction encoded in a quantum register A. The encoding, i.e., the mapping $n \mapsto U_n \in \mathcal{G}$, is known to both the client and the server. The instruction $|n\rangle$ means *"Apply U_n to $|0\rangle_{\mathcal{M}}^{\otimes N}$"*. Each qubit is prepared either in the $|0\rangle$ or in the $|1\rangle$ state. The state of the quantum register has the form $|\phi\rangle_A = |n\rangle_A = |1011\cdots\rangle_A$. Note that the quantum register needs to have at least $\log_2(G)$ qubits in order to represent any gate in $\mathcal{G}$.

Step 3:

The server uses the register $|n\rangle_A$without measuring it in order to apply U_n to his quantum memory $\mathcal{M}$. He applies U_{Bob} on the global state $|n\rangle_A \otimes |\Psi\rangle_{\mathcal{M}}$,

$$U_{Bob} = \sum_n |n\rangle\langle n| \otimes U_n. \tag{18}$$

This yields the mapping

$$|n\rangle_A \otimes |\Psi\rangle_{\mathcal{M}} \to |n\rangle_A \otimes U_n |\Psi\rangle_{\mathcal{M}}. \tag{19}$$

Step 4:

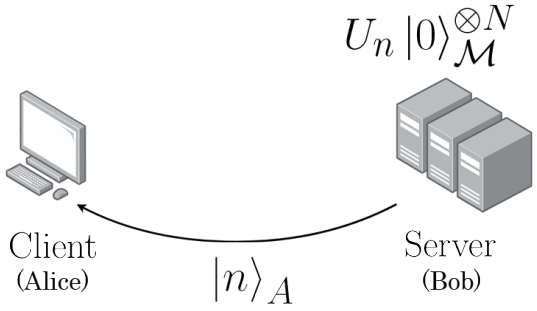

Figure 6: The server then sends the quantum register $|n\rangle_A$ back to the client. The client can then prepare the quantum register for the next instruction, i.e., for the next gate in the quantum computation.

Alice can instruct Bob to perform an arbitrary computation on the quantum memory $\mathcal{M}$ by going back to step two and repeating this process multiple times.

3.1.2 Server's blindness

The previous section explains how a quantum computation might take place. But, what about the server's blindness? How can Alice (the client) make sure that Bob (the server) cannot get any meaningful information about the computation by measuring the quantum register? The solution is to send the quantum register in a state of superposition. If the server then measures the quantum register, the state will collapse with measurable effects on the client's side. It is equivalent to a special case of the unbalanced BB84 protocol. The curious reader is referred to [13], [14] , and to appendix E.

We will refer to these superposition states as *lures*. Each qubit of the quantum register is prepared either in the $|0\rangle$, $|1\rangle$, $|+\rangle$ or in the $|-\rangle$ state. Consequently, lures have the form $|\phi\rangle_A = |0 + 11 - \cdots\rangle_A = \sum_n \eta_n |n\rangle_A$, with $\eta_n \in \{-1, 0, 1\}$. Note that the state's normalization has been omitted for ease of notation.

When the client sends the quantum register to the server, it will either be an instruction or, at random times, a lure. If the client sends a lure, she will measure it when she gets it back from the server. The client will measure each of the quantum register's qubits in the same basis as they were prepared to detect if the server has attempted to extract information from the register. She stops the protocol if her measurement results are not consistent with the lure she prepared.

Note that the lures have to be carefully chosen so that the quantum computation is not disrupted. Lures can be chosen to act on a memory section, unknown to the server, separated from where the actual quantum computation occurs.

U_{Bob} transforms the global state as

$$|\phi\rangle_A \otimes |\Psi\rangle_{\mathcal{M}} \to \sum_n \eta_n |n\rangle_A \otimes U_n |\Psi\rangle_{\mathcal{M}} \tag{20}$$

if Alice sends a lure to Bob. It is convenient to have a factorized state between the subsystems of the client and the server. The client might prematurely stop the protocol if she measures a lure that is entangled to Bob's memory, as the measurement results will not match the prepared lure. Therefore, it is preferable to have a situation where

$$U_n |\Psi\rangle_{\mathcal{M}} = U_{n'} |\Psi\rangle_{\mathcal{M}} = U_{n''} |\Psi\rangle_{\mathcal{M}} = \cdots . \tag{21}$$

We will get a factorized state if the unitaries act trivially on the server's memory.

The probability that Bob can measure Alice's quantum register, also called *cheating*, j times without being detected decreases exponentially as $p^{\gamma j}$. Here, p is the probability that the server can cheat without being detected on a single lure, and γ is the average fraction of registers sent by Alice that are lures.

The server will only get information on an exponentially small fraction of Alice's instructions before she detects him and stops the protocol.

3.1.3 Overview

We can summarize the efficient universal blind quantum computation (EUBQC) protocol [15] in the following steps:

1. The server initializes his memory $\mathcal{M}$ to the $|0\rangle_{\mathcal{M}}^{\otimes N}$ state.
2. The client sends the server a quantum register A. It is randomly either an instruction or a lure.
3. The server applies U_{Bob} on the joint system and sends the quantum register back to the client.
4. If the quantum register was a lure and is unentangled from the quantum memory $\mathcal{M}$, the client measures it. If the measurement result does not match the state she had prepared, she stops the protocol. Otherwise she proceeds to step 2.
5. The server measures each qubit of his quantum memory in the computational basis and reveals the results.

As mentioned in [15], in a classical cloud computation setting, an N-bit classical universal programmable computer (CUPC) would require at least $O(\mathrm{J} \log_2 \mathrm{N})$ bits to be exchanged between the client and the server for a J-step computation. The EUBQC protocol

requires qubits instead of bits, as instructions are encoded in a quantum register instead of in a classical register. It also requires a small overhead in the number of exchanged qubits due to lures. There is no overhead in terms of gates compared to what a CUPC would need. Therefore, the protocol is efficient in terms of the number of exchanged qubits and the number of gates needed for universal quantum computation.

4 Noise in the UBQC protocol

So far, we have discussed the protocol for the ideal situation where there is no quantum noise present. We must consider the fidelity of the prepared qubits, how well we can transmit them, and how well we can perform quantum gates in any realistic scenario. Fault-tolerant scalability, i.e., increasing the number of logical qubits, see appendix D, while having a reasonably high computation fidelity, will be necessary to reach the quantum advantage domain of quantum computing. It will require that the error probability associated with qubit initialization, single- and two-qubit gate implementation, and readout all remain below the $\sim 1\%$ threshold [16].

4.1 Noise on the client's side

We will mainly talk about two processes that introduce noise to the client's quantum system: the quantum register's state preparation and transmission.

Even though it is possible to initialize single qubits, e.g., initialization to the $|0\rangle$ state, with the high fidelity of approximately 99.93% [17], small preparation errors can become significant in the context of the EUBQC protocol. The ability to send quantum information over long distances is especially important in any BCQ protocol. It seems that the best way of transmitting a qubit's state is by sending photons through optical fibers or satellite links. However, photon loss and decoherence are induced due to photon absorption and the communications channel's imperfections. The implementation of so-called *quantum repeaters* will play an essential role in making long-distance quantum communication possible [19].

The best way to describe the client's noisy quantum register would be with a noise model that acts locally on each qubit, as errors would occur at the single-qubit scale. However, we will limit ourselves to model the client's noisy quantum register by the global depolarizing channel

$$\Lambda(\rho_A) = (1-p)\rho_A + \frac{p}{G}\mathbb{1}, \quad p \in [0,1], \tag{22}$$

where G is, as before, the number of gates in $\mathcal{G}$ and $\rho_A = |\phi\rangle\langle\phi|_A$ is the density matrix of the client's quantum register. The quantum register may be noisy due to imperfect qubit state preparation and their imperfect transmission. Note that we are making a strong simplification since the channel is not local.

4.2 Noise on the server's side

Assuming the server's quantum system is well isolated, applying single- and two-qubit quantum gates will be the only relevant process on his side that introduces noise to his quantum memory. We will ignore noise coming from quantum memory initialization, as it is a very high fidelity process that only happens once at the beginning of the protocol.

It has been shown that it is possible to achieve 99.9(1)% fidelity for single-qubit gates, and 99.9934(3)% fidelity for two-qubit gates on trapped ions [16].

The best way to describe the noise coming from the server is, as before, with a noise model that acts locally on the memory at the single- or two-qubit scale, as errors would occur whenever a single- or a two-qubit gate is applied. We will again limit ourselves to using a global depolarizing channel on the server's quantum system

$$\Lambda_{\mathcal{M}}(\sigma_{\mathcal{M}}) = (1-q)U_n\sigma_{\mathcal{M}}U_n^\dagger + \frac{q}{2^N}\mathbb{1}, \quad q \in [0,1]. \tag{23}$$

N is the number of qubits in the quantum memory, q is the noise parameter, and $\sigma_{\mathcal{M}} = |\Psi\rangle\langle\Psi|_{\mathcal{M}}$ is it's noiseless density matrix. The map is a simplified description of the noisy memory caused by faulty unitary operations on the server's side.

4.3 Noise in the quantum computation

We will focus on the effects that noise has on the compu tation aspect of the protocol. An analysis of how noise affects the server's blindness is an interesting topic for future work.

As we know from the protocol, the computation takes place on the server's quantum memory. The end goal is to find a noise model for his quantum memory, including noise coming from the client's system and the server's system. The map $\Lambda_{\mathcal{M}}$ acts on the server's system, but the one described in section 4.1 acts only on the client's quantum system. However, the client's noisy quantum register will induce a noise map on the quantum memory after the server applies U_{Bob}.

Let us see what happens if the client sends an arbitrary quantum register $|\phi\rangle_A = \sum_n \eta_n |n\rangle$ with $\eta_n \in \{-1, 0, 1\}$ to the server. We will ignore the state's normalization for simplicity. First, let the depolarizing channel act on $|\phi\rangle\langle\phi|_A$.

$$
\begin{aligned}
\Lambda(|\phi\rangle\langle\phi|_A) &= (1-p)\,|\phi\rangle\langle\phi|_A + \frac{p}{G}\,\mathbb{1} \\
&= (1-p)\sum_{n,k}\eta_n\,\eta_k\;|n\rangle\langle k| + \frac{p}{G}\,\mathbb{1} \\
&= (1-p)\sum_{n}\eta_n^2\;|n\rangle\langle n| + (1-p)\sum_{n\neq k}\eta_n\,\eta_k\;|n\rangle\langle k| + \frac{p}{G}\,\mathbb{1} \\
&= (1-p)\sum_{n}\eta_n^2\;|n\rangle\langle n| + (1-p)\sum_{n\neq k}\eta_n\,\eta_k\;|n\rangle\langle k| + \frac{p}{G}\sum_{n}|n\rangle\langle n| \\
&= \sum_{n}\Big[(1-p)\eta_n^2 + \frac{p}{G}\Big]\,|n\rangle\langle n| + (1-p)\sum_{n\neq k}\eta_n\,\eta_k\;|n\rangle\langle k|
\end{aligned}
\tag{24}
$$

Now consider the action of U_{Bob} on the global state $\Lambda(|\phi\rangle\langle\phi|_A)\otimes\sigma_{\mathcal{M}}$, where $\sigma_{\mathcal{M}}$ is the density matrix of the server's quantum memory.

$$
\begin{aligned}
&U_{Bob}(\Lambda(|\phi\rangle\langle\phi|_A)\otimes\sigma_{\mathcal{M}})U_{Bob}^{\dagger} = \\
&= U_{Bob}\left(\sum_{n}\Big[(1-p)\eta_n^2 + \frac{p}{G}\Big]\,|n\rangle\langle n|\otimes\sigma_{\mathcal{M}} + (1-p)\sum_{n\neq k}\eta_n\,\eta_k\;|n\rangle\langle k|\otimes\sigma_{\mathcal{M}}\right)U_{Bob}^{\dagger} \\
&= \sum_{l}|l\rangle\langle l|\otimes U_l\left(\sum_{n}\Big[(1-p)\eta_n^2 + \frac{p}{G}\Big]\,|n\rangle\langle n|\otimes\sigma_{\mathcal{M}} + (1-p)\sum_{n\neq k}\eta_n\,\eta_k\;|n\rangle\langle k|\otimes\sigma_{\mathcal{M}}\right)\sum_{j}|j\rangle\langle j|\otimes U_j^{\dagger} \\
&= \sum_{n}\Big[(1-p)\eta_n^2 + \frac{p}{G}\Big]\,|n\rangle\langle n|\otimes U_n\sigma_{\mathcal{M}}U_n^{\dagger} + (1-p)\sum_{n\neq k}\eta_n\,\eta_k\;|n\rangle\langle k|\otimes U_n\sigma_{\mathcal{M}}U_k^{\dagger}
\end{aligned}
\tag{25}
$$

We get a relatively complicated expression in Eq. (25). However, we will simplify it by taking the partial trace over the client's system, as we are interested in the state of the server's memory, where the quantum computation occurs.

$$
\begin{aligned}
&\mathrm{Tr}_A\Big[U_{Bob}(\Lambda(|\phi\rangle\langle\phi|_A)\otimes\sigma_{\mathcal{M}})U_{Bob}^{\dagger}\Big] = \\
&= \sum_{j}\langle j|\otimes\mathbb{1}\left[\sum_{n}\Big[(1-p)\eta_n^2 + \frac{p}{G}\Big]\,|n\rangle\langle n|\otimes U_n\sigma_{\mathcal{M}}U_n^{\dagger} + (1-p)\sum_{n\neq k}\eta_n\,\eta_k\;|n\rangle\langle k|\otimes U_n\sigma_{\mathcal{M}}U_k^{\dagger}\right]|j\rangle\otimes\mathbb{1} \\
&= \sum_{n}\Big[(1-p)\eta_n^2 + \frac{p}{G}\Big]U_n\sigma_{\mathcal{M}}U_n^{\dagger} \equiv \sum_{n}\lambda_n U_n\sigma_{\mathcal{M}}U_n^{\dagger} \quad ,
\end{aligned}
\tag{26}
$$

where $\lambda_n \equiv \left[(1-p)\eta_n^2 + \frac{p}{G}\right]$. The equation $\sum_n \lambda_n = 1$ holds if $|\phi\rangle_A$ is normalized. Equation (26) is the induced noise map on the server's system that results after applying U_{Bob} on the global system. We effectively have two different noise maps that can act on the server's quantum memory Λ_A and $\Lambda_{\mathcal{M}}$. See Eq. (27). The Λ_A noise map is relevant if the client's quantum register is noisy. In contrast, the $\Lambda_{\mathcal{M}}$ noise map is relevant if the quantum operations on the server's side are noisy.

$$\begin{aligned} \Lambda_A(|\phi\rangle_A, \sigma_{\mathcal{M}}) &= \sum_{n=0}^{G-1} \lambda_n U_n \sigma_{\mathcal{M}} U_n^\dagger, \\ \Lambda_{\mathcal{M}}(\sigma_{\mathcal{M}}) &= (1-q)\sigma_{\mathcal{M}} + \frac{q}{2^N}\mathbb{1}. \end{aligned} \tag{27}$$

Note that Λ_A depends on the specific values of λ_n, which come from the specific state of the quantum register $|\phi\rangle_A$. We see that

$$\begin{aligned} \Lambda_A(\Lambda_{\mathcal{M}}(\sigma)) &= (1-p)\sum_n \lambda_n U_n \sigma U_n^\dagger + \frac{p}{2^N}\sum_n \lambda_n \mathbb{1} \\ &= (1-p)\sum_n \lambda_n U_n \sigma U_n^\dagger + \frac{p}{2^N}\mathbb{1} \\ &= \Lambda_{\mathcal{M}}(\Lambda_A(\sigma)). \end{aligned} \tag{28}$$

The maps commute, which means we can apply the noise maps in any order when calculating the effects of both noise maps numerically.

4.4 Noisy instructions: a special case

We want to analyze the special case, where the client's quantum register is an instruction of the form $|\phi\rangle_A = |k\rangle\langle k|$, i.e., the instruction *"Apply U_k"*. Setting $\eta_k = 1$ and $\eta_{n\neq k} = 0$ reduces the noise map Λ_A to

$$\begin{aligned} \Lambda_A(|k\rangle, \sigma_{\mathcal{M}}) &= \left[(1-p) + \frac{p}{G}\right] U_k \sigma_{\mathcal{M}} U_k^\dagger + \frac{p}{G}\sum_{n\neq k} U_n \sigma_{\mathcal{M}} U_n^\dagger \\ &= \sum_n \lambda_n U_n \sigma_{\mathcal{M}} U_n^\dagger, \end{aligned} \tag{29}$$

where n runs over all instructions in $\mathcal{G}$, i.e., $n \in [0, G-1]$, and $\lambda_k = (1-p) + \frac{p}{G}$, $\lambda_{n\neq k} = \frac{p}{G}$. Equation (29) is what results after sending only one computational instruction. How does the server's memory look like after multiple computational instructions?

If we follow the same steps for instructions $|x\rangle$, $|y\rangle$, $|z\rangle\ldots$ (in that order) we get

$$\sigma'_{\mathcal{M}} = \sum_{n,k,l,\ldots} (\cdots \lambda_l \beta_k \alpha_n) \cdots U_l U_k U_n \sigma U_n^\dagger U_k^\dagger U_l^\dagger \cdots , \tag{30}$$

with prefactors

$$\begin{aligned} \alpha_x &= (1-p) + \tfrac{p}{G}, \qquad \alpha_{n\neq x} = \tfrac{p}{G}; \\ \beta_y &= (1-p) + \tfrac{p}{G}, \qquad \beta_{k\neq y} = \tfrac{p}{G}; \\ \lambda_z &= (1-p) + \tfrac{p}{G}, \qquad \lambda_{l\neq z} = \tfrac{p}{G}. \end{aligned} \tag{31}$$

We can see that the resulting noisy density matrix $\sigma'_{\mathcal{M}}$ will be a mixed state containing all possible operations that can be done in a limited number of steps on the server's quantum memory $\sigma_{\mathcal{M}}$. For example, after j computational instructions, the noisy memory will be in a mixed state represented by the sum of G^j matrices with different weights depending on those j instructions. We can write

$$\sigma'_{\mathcal{M}} = \sum_i \tilde{\lambda}_i \tilde{\sigma}_i \tag{32}$$

for simplicity, but now the index runs from 0 to $G^j - 1$.

4.5 Fidelity and entropy

We can learn a lot about the server's noisy quantum memory by looking at its von Neumann entropy and at the computation's fidelity. The von Neumann entropy of the noisy memory's state $S(\sigma'_{\mathcal{M}})$ will give us information about how mixed it has become.

$$S(\sigma'_{\mathcal{M}}) = -\mathrm{Tr}(\sigma'_{\mathcal{M}} \log_2 \sigma'_{\mathcal{M}}) = -\sum_i \lambda_i \log_2 \lambda_i, \quad S(\sigma'_{\mathcal{M}}) \in [0, N] \tag{33}$$

The values for λ_i are the eigenvalues of $\sigma'_{\mathcal{M}}$, and N is the number of qubits. Pure states like in section 2.3 have zero entropy, while the maximally mixed state has maximal entropy, i.e., $S(\mathbb{1}/2^N) = N$.

The fidelity can measure how different the noisy state $\sigma'_{\mathcal{M}}$ is, compared to the pure noiseless state $|\Psi\rangle_{\mathcal{M}}$ of the quantum memory.

$$F\big(|\Psi\rangle_{\mathcal{M}}, \sigma'_{\mathcal{M}}\big) = \langle\Psi|\, \sigma'_{\mathcal{M}}\, |\Psi\rangle_{\mathcal{M}}, \quad F\big(|\Psi\rangle, \sigma'_{\mathcal{M}}\big) \in [0, 1] \tag{34}$$

A fidelity of zero means that the states are maximally "different", i.e., no overlap, while a fidelity of 1 means the states are the same, i.e., maximal overlap. Note that this is the fidelity for the complete memory. It may be necessary to reduce the state if the computation uses less qubits than there are qubits in the server's memory. For example, if Alice wanted to compute the Bell circuit (two logical qubits), the state would be reduced to the system of these two qubits before computing the fidelity. See appendix F.

5 Numerical results

5.1 Noisy instructions

We will assume that $\mathcal{G} = \{H_i, T_i, C_X(j,k)\}$, where H_i and T_i act on the ith qubit of the server's memory, and $C_X(j,k)$ takes the jth and kth qubit as control and target. Thus, the gate set $\mathcal{G}$ contains $G = N(N+1)$ gates, and the client's quantum register A consists of $O(\log_2(N))$ qubits. See [25] for the numerical result's code.

Let us analyze the von Neumann entropy of the memory's mixed state, i.e., the noisy memory, after we repeat one specific gate instruction j times, i.e., $(H_1)^j$, $(T_1)^j$ and $(C_X(1,2))^j$. The memory is starting from the initial state $|\Psi\rangle_{\mathcal{M}} = |0\rangle^{\otimes N}$. The corresponding density matrix is given by $\sigma_{\mathcal{M}} = |\Psi\rangle\langle\Psi|_{\mathcal{M}}$. We will only use the noise map mentioned in section 4.4, i.e., $\Lambda_A(|k\rangle, \sigma_{\mathcal{M}})$.

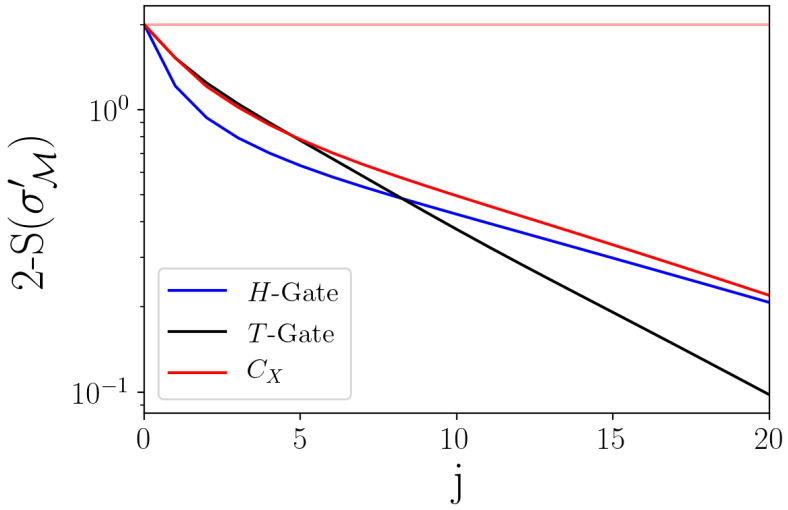

Figure 7: Logarithmic plot of $N - S(\sigma'_{\mathcal{M}})$ vs. j (computational steps) for a memory of $N = 2$ qubits and a noise map Λ_A with noise parameter $p = 0.5$. The computational instructions for the "H-Gate" (blue) plot are given by $(H_1)^j$ for every j. The instructions for the "C_X" (red) and the "T-Gate" (black) plots are analogous . Note that Alice's quantum register has to be sent j times. We see that the entropy behaves exponentially after some steps.

We can extend the range to $j \in [0, 100]$ to better appreciate the exponential behavior.

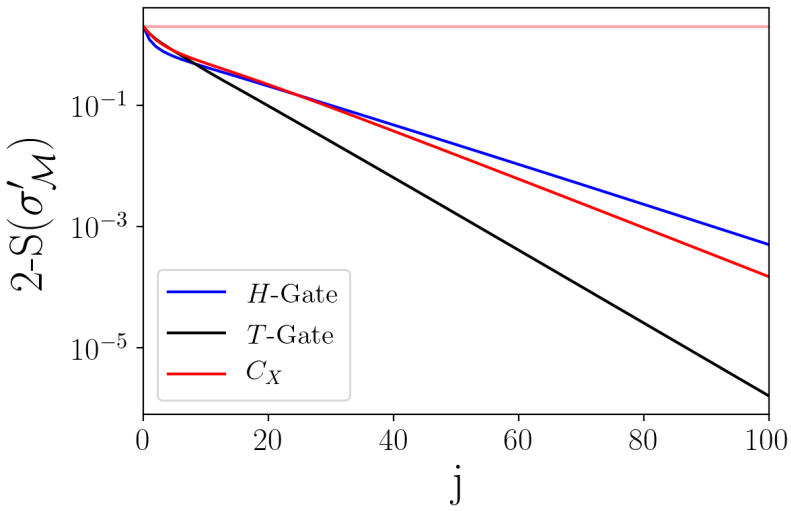

Figure 8: Logarithmic plot of $N - S(\sigma'_{\mathcal{M}})$ vs. j (computational steps) for a memory of $N = 2$ qubits and a noise map Λ_A with noise parameter $p = 0.5$.

We see in Fig. 7 and Fig. 8 that sending either H-Gate, T-Gate, or C_X instructions will increase the von Neumann entropy of the server's noisy quantum memory at different rates over the number of steps j. The entropies plotted in those figures can be approximated in the exponential regime as

$$S(k, j) \propto 1 - e^{-kj}, \quad k \geq 0, \tag{35}$$

where k is the slope in the logarithmic plot. We can analyze the evolution of the von Neumann entropy as a function of the computational steps j for different different noise parameter values p. Note that we will focus mainly on the noisy memory state's entropy that corresponds to sending the H-Gate instruction j times, since the discussion for the T-Gate and C_X j-dependent entropy plots is analogous. We set $N = 2$, but the discussion for other values of N is analogous.

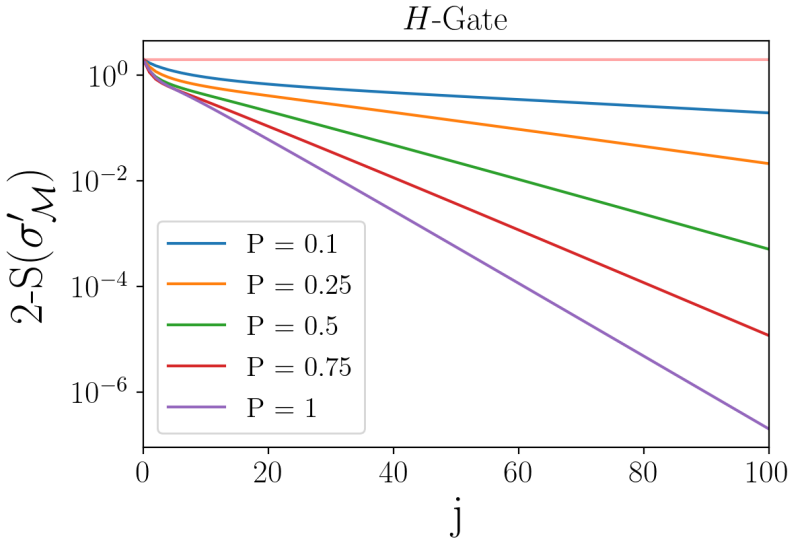

Figure 9: Logarithmic plot of $N - S(\sigma'_{\mathcal{M}})$ for the H-Gate in the range of $j \in [0, 100]$ computational steps for a memory of $N = 2$ qubits and a noise map Λ_A with noise parameters $p \in \{0.1, 0.25, 0.5, 0.75, 1\}$.

We see in Fig. 9 that the slope in the logarithmic plot of $N - S(\sigma'_{\mathcal{M}})$ changes depending on the noise parameter p of the noise map. We may write the slope as $k = k(p)$, and so the entropy becomes p-dependent and can be written as

$$S(j, p) \propto 1 - e^{-k(p)j}, \quad k \geq 0, \tag{36}$$

in the exponential regime.

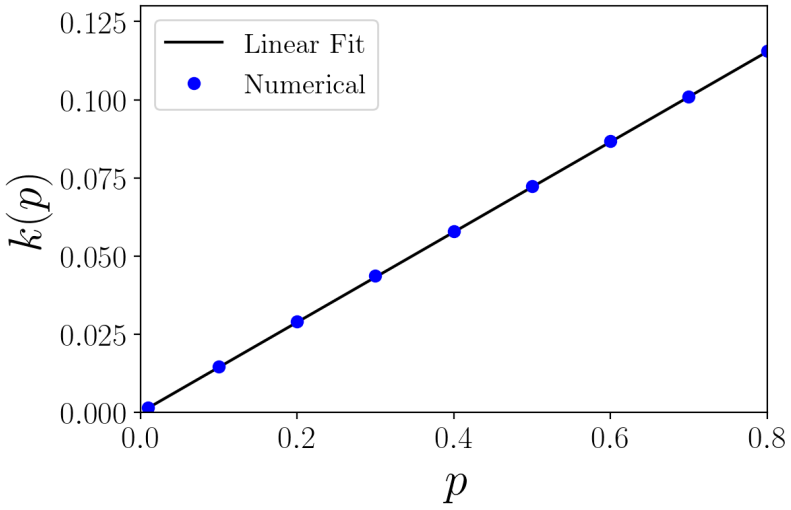

Figure 10: A plot of the p-dependent parameter $k(p)$ mentioned in Eq. (36) for noise-parameter values $p \in \{0.01, 0.1, 0.2, 0.3, \cdots, 0.8\}$. We see that $k(p)$ can be approximated by a linear function given by $k(p) \approx 0.144 \cdot p$, with a mean squared error of MSE $\approx 10^{-8}$.

Figure 10 shows that the parameter $k(p)$ mentioned in Eq. (36) depends linearly on the noise parameter p. We may now write

$$S(j, p) \sim 1 - e^{-0.144\, p\, j}. \tag{37}$$

It is worth noting that for the values p_1, j_1, p_2 and j_2 with $\frac{j_1}{j_2} = \alpha$, we can write

$$\begin{aligned} & 1 - e^{-k(p_1)\, j_1} = 1 - e^{-k(p_2)\, j_2} \\ \Leftrightarrow \quad & p_1\, j_1 = p_2\, j_2 \\ \Leftrightarrow \quad & \frac{j_1}{j_2} = \frac{p_2}{p_1} = \alpha. \end{aligned} \tag{38}$$

We see that changing the noise parameter's value effectively scales the entropy function, i.e., given the values $p_1 = 0.5$ and $p_2 = 0.25$, the entropy is scaled as $S(j/2, 0.5) = S(j, 0.25)$.

5.2 Bell circuit fidelity

Let us consider the situation where the client wants to compute the Bell circuit, see Fig. 2, on the server's memory.

We will get numerical results for the Bell circuit's fidelity on a quantum memory with $N = 2$ qubits, for the noise maps Λ_A and $\Lambda_{\mathcal{M}}$ individually, and together, see Eq. (27). The noise map's parameters are set to $p = q = 0.01$ in Eqs. (22) and (23) since that is the value we are interested in.

We want to compute the fidelity of computing the four Bell states $|\phi^+\rangle, |\psi^+\rangle, |\phi^-\rangle$, and $|\psi^-\rangle$ when noise is present. The noiseless quantum computations are represented by the following quantum circuits, see Fig. 2.

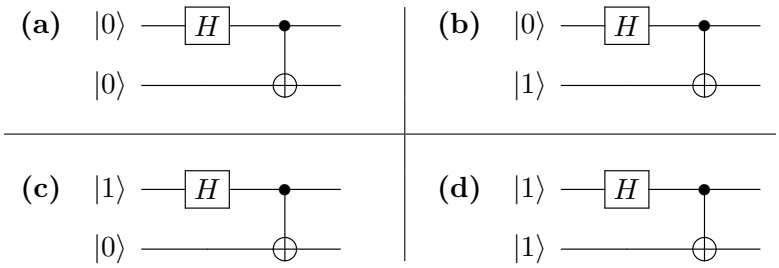

Figure 11: Quantum circuits representing the Bell circuit with inputs **(a)** $|00\rangle$, **(b)** $|01\rangle$, **(c)** $|10\rangle$, and **(d)** $|11\rangle$. The resulting noiseless states are $|\phi^+\rangle, |\psi^+\rangle, |\phi^-\rangle$ and $|\psi^-\rangle$, respectively. See Eq. (17).

Note that, since the memory is initially in the $|00\rangle_{\mathcal{M}}$ state, an operation that flips $|0\rangle$ to $|1\rangle$ is needed, i.e., $X|0\rangle = |1\rangle$, see Eq. (6). We can find such a transformation by combining unitary operations from our universal set of gates $\mathcal{G} = \{H_i, T_i, C_{X_{i,j}}\}$. We find that $X = H\,T^4\,H$, and we write $X \otimes \mathbb{1} \equiv X_0 = H_0\,T_0^4\,H_0$ for the flip operation acting on the first qubit and $\mathbb{1} \otimes X \equiv X_1 = H_1\,T_1^4\,H_1$ for the flip operation acting on the second qubit.

We can rewrite the four quantum circuits in Fig. 11 as follows.

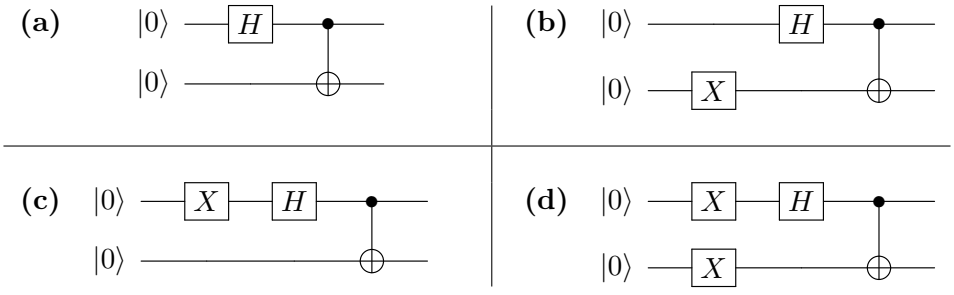

Figure 12: Quantum circuits representing the Bell circuit with inputs **(a)** $|00\rangle$, **(b)** $|01\rangle$, **(c)** $|10\rangle$, and **(d)** $|11\rangle$. The server's quantum memory starts in the $|00\rangle_{\mathcal{M}}$ state, and the $X = H\,T^4\,H$ operation flipps the qubits. The resulting noiseless states are $|\phi^+\rangle, |\psi^+\rangle, |\phi^-\rangle$, and $|\psi^-\rangle$, respectively. See Eq. (17).

We expect the fidelity to decrease with the number of noisy instructions. Sending more instructions will introduce more noise in the server's quantum memory. Therefore, the quantum computations represented by the quantum circuits in Fig. 12 will all have different fidelities. As two instructions are needed for Fig. 12 (a), eight for Fig. 12 (b), six for Fig. 12 (c), and twelve for Fig. 12 (d).

Note that in Figs. 12 (c) and 12 (d), we use the identity $H^2 = \mathbb{1}$, and so $H\,X = H\,(H\,T^4\,H) = T^4\,H$.

We will use the following notation for the different fidelities

$$\mathrm{F}_{S,M} = \langle S|\,\sigma'_{S,M}\,|S\rangle\,, \tag{39}$$

where index S represents one of the Bell states' computation, and index M represents the noise map being considered. The density matrix $\sigma'_{S,M}$ represents the noisy state of the quantum memory, given indices S and M. See appendix F for a description of how to compute the Bell circuit fidelity for a quantum memory with $N > 2$.

Index S can be either $S = \phi^+$, $S = \psi^+$, $S = \phi^-$, or $S = \psi^-$, depending on the computation. Index M can be either $M = A$, i.e., only the noise map Λ_A is considered, $M = \mathcal{M}$, i.e., only the noise map $\Lambda_{\mathcal{M}}$ is considered, or $M = \mathcal{M}A$, i.e., both noise maps are considered. For example, the fidelity of the computation represented by Fig. 12 (a), while only considering the Λ_A noise map, would be written as

$$\mathrm{F}_{\phi^+,A} = \left\langle \phi^+ \right|\,\sigma'_{\phi^+,A}\,\left|\phi^+\right\rangle\,. \tag{40}$$

Numerical results for the different fidelities yield the following plot.

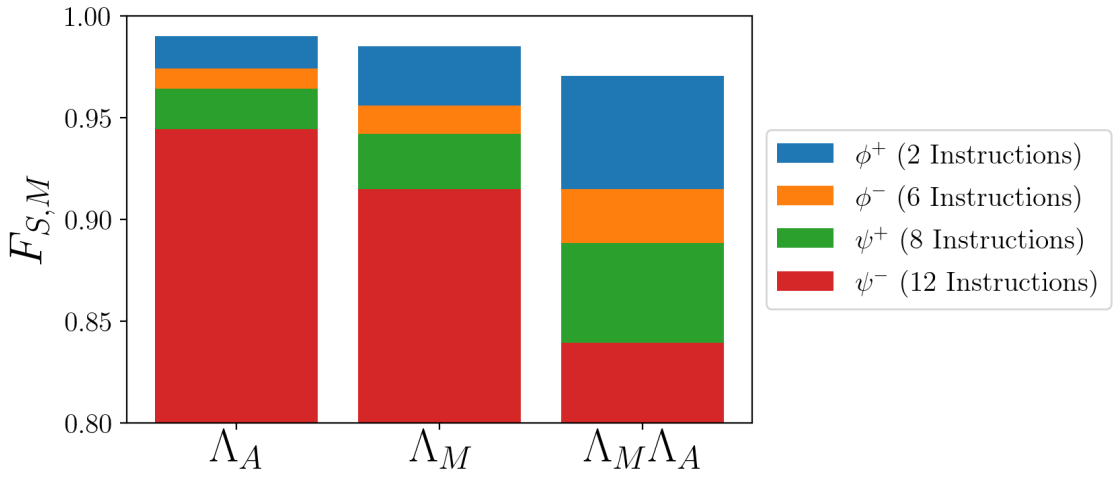

Figure 13: Bar-plot of the fidelities $\mathrm{F}_{S,M}$, see Eq. (39), for different noise maps. The Λ_A noise map is considered on the far left ($M = A$), see Eq. (29), the $\Lambda_{\mathcal{M}}$ noise map in the middle ($M = \mathcal{M}$), see Eq. (23), and both on the far right ($M = \mathcal{M}A$). The computation of each Bell state requires a different number of instructions. Two instructions are required for $|\phi^+\rangle$, eight for $|\psi^+\rangle$, six for $|\phi^-\rangle$, and twelve for $|\psi^-\rangle$. See Eq. (17) and Fig. 12.

The specific fidelity values plotted in Fig. 13 are

$$\begin{aligned}
&\mathrm{F}_{\phi^+,A} \approx 0.990, \quad &&\mathrm{F}_{\phi^+,\mathcal{M}} \approx 0.985, \quad &&\mathrm{F}_{\phi^+,\mathcal{M}A} \approx 0.970,\\
&\mathrm{F}_{\phi^-,A} \approx 0.974, \quad &&\mathrm{F}_{\phi^-,\mathcal{M}} \approx 0.956, \quad &&\mathrm{F}_{\phi^-,\mathcal{M}A} \approx 0.915,\\
&\mathrm{F}_{\psi^+,A} \approx 0.964, \quad &&\mathrm{F}_{\psi^+,\mathcal{M}} \approx 0.942, \quad &&\mathrm{F}_{\psi^+,\mathcal{M}A} \approx 0.889,\\
&\mathrm{F}_{\psi^-,A} \approx 0.945, \quad &&\mathrm{F}_{\psi^-,\mathcal{M}} \approx 0.915, \quad &&\mathrm{F}_{\psi^-,\mathcal{M}A} \approx 0.839.
\end{aligned} \tag{41}$$

As we expected, the fidelity drops with the number of instructions needed for the computation. It would be interesting to quantify how the fidelity drops with the number of instructions. However, it remains an open question since this behavior will depend on the specific quantum computation being considered. The four quantum computations we have considered, see Fig. 12, are not sufficient to quantify the fidelity's dependency on the number of instructions.

6 Conclusions and outlook

Due to today's adoption of classical cloud computing to delegate computationally demanding tasks, it is only natural to expect the same development in the field of quantum computation and quantum communication. BQC protocols allow for the delegation of quantum computations to an untrusted quantum server while not allowing the server to learn any meaningful information about the client's computation. BQC protocols are, arguably, of great importance for the future development of quantum computation and communication, as it facilitates access to quantum computation's state-of-the-art technology.

The fidelity of the delegated computation will be affected by quantum noise in any realistic scenario. Noisy qubit state preparation, noisy transmission, and noisy quantum gate implementation are not perfect and must be considered.

We introduced the depolarizing channel as a noise model to the efficient universal blind quantum computing protocol by Giovannetti et al. [15] and investigated how noise in the client's and in the server's quantum system affect the state of the server's quantum memory. It is important to point out that using a local noise model would be a more realistic description in contrast to what has been done in this bachelor thesis as errors would occur at the single- and two-qubit scale. We considered two noise maps, Λ_A and $\Lambda_{\mathcal{M}}$. The Λ_A map describes the noise acting on the server's quantum memory coming from the client's noisy quantum register. In contrast, the $\Lambda_{\mathcal{M}}$ map describes the noise acting on the server's quantum memory, coming from noisy unitary operations.

We evaluated the the von Neumann entropy's behavior of the server's noisy memory by repeating the same instruction multiple times. We found that the von Neumann entropy depends on the number of instructions, given by j, and on the noise parameter, given by p. It can be approximated by an exponential function

$$S(j,p) \propto 1 - e^{-k\,p\,j}, \quad k \geq 0,$$

where we get different values of k if we repeat a different instruction from the universal set $\mathcal{G}$.

We evaluated the computation fidelity of a specific circuit, the Bell circuit, in the context of the protocol. We found numerical results for the fidelity of preparing the $|\phi^+\rangle$, $|\psi^+\rangle$, $|\phi^-\rangle$, and $|\psi^-\rangle$ states for noise maps Λ_A, $\Lambda_{\mathcal{M}}$, and $\Lambda_{\mathcal{M}}\Lambda_A$, given a fixed universal set of gates $\mathcal{G}$. As we expected, the computation fidelity decreases with the number of instructions. The numerical results for the Bell circuit's fidelities are not enough to quantify the fidelity's dependency on the number of instructions. One may give a quantitative description of this behavior by taking the average over a large set of different computations.

Future work must be done to quantify the protocol's "blindness" when noise is present. Since the protocol's blindness can be reformulated as a special case of the unbalanced BB84 protocol, one could use quantum cryptography's security methods when noise is included to quantify the protocol's blindness [13] [14].

We can also expect a *Blindness vs. Fidelity* trade-off. An untrusted server will know less about the computation if more lures are sent. However, more lures will also introduce more noise to the server's quantum memory, which will decrease the computation's fidelity.

References

[1] Bruß and Kampermann, Quanteninformationstheorie, Heinrich-Heine-Universität, (2004).

[2] Bruß, Theoretical Quantum Optics and Quantum Information, Heinrich-Heine-Universität, (2013).

[3] Benenti, Casati and Strini, Principles of Quantum Computation and Information, Volume I: Basic Concepts .

[4] Benenti, Casati and Strini, Principles of Quantum Computation and Information, Volume II: Basic Tools and Special Topics.

[5] Nielsen and Chuang, Quantum Computation and Quantum Information, 10th Anniversary Edition (2010).

[6] Lidar and Brun, Quantum Error Correction, Cambridge University Press.

[7] Kampermann, Gühne, Wilmott and Bruß, Algorithm for characterizing stochastic local operations and classical communication classes of multiparticle entanglement, Phys. Rev. A 86, 032307 (2012).

[8] Fitzsimons, Private quantum computation: an introduction to blind quantum computing and related protocols, npj Quantum Information 3, 23 (2017).

[9] Childs, Secure assisted quantum computation, Quantum Information & Computation 5, 6, 456 (2005).

[10] Broadbent, Fitzsimons and Kashefi, Universal blind quantum computation, arXiv:0807.4154v3 [quant-ph] (2009).

[11] Vernam, J. American Inst. Elec. Eng. 45, 109 (1926).

[12] Acín Et al., The quantum technologies roadmap: a European community view, New J. Phys. 20 080201 (2018).

[13] Bennett and Brassard, Quantum cryptography: Public key distribution and coin tossing, Theoretical Computer Science, 560 ,7-11, (2014).

[14] H.-K. Lo, H.F. Chau, M. Ardehali, J. of Cryptology, 18, 133 (2005).

[15] Giovannetti Et al., Efficient Universal Blind Quantum Computation, PRL 111, 230501 (2013).

[16] Ballance Et al., High-fidelity quantum logic gates using trapped-ion hyperfine qubits, Phys. Rev. Lett. 117, 060504 (2016).

[17] Harty Et al., High-fidelity preparation, gates, memory and readout of a trapped-ion quantum bit, Phys. Rev. Lett. 113, 220501 (2014).

[18] John Preskill, Quantum Computing in the NISQ era and beyond, Quantum 2, 79 (2018).

[19] Muralidharan Et al., Efficient long distance quantum communication, Scientific Reports 6, 20463 (2016).

[20] Bogopolski, Lineare Algebra I, Heinrich-Heine-Universität, (2018).

[21] Xandri, Notes on metric spaces, https://scholar.princeton.edu/

[22] Hunter and Nachtergaele, Applied Analysis, Chapter 6, University of California at Davis, (2000).

[23] Hunter and Nachtergaele, Applied Analysis, Chapter 5, University of California at Davis, (2000).

[24] Wolf, Functional Analysis, Lecture 3 (Notes by M. Mosonyi), Technische Universität München, (2015).

[25] Henning, EUBQC including noise (Numerical results), GitHub repository, https://github.com/jmsett/EUBQC-including-noise-Numerical-results-, (2020)

Appendices

A Complex numbers

Any complex number $z \in \mathbb{C}$ may be written as

$$z = \mathrm{Re}\{z\} + i\,\mathrm{Im}\{z\} \equiv x + iy, \tag{42}$$

where $i^2 = -1$, $x = \mathrm{Re}\{z\} \in \mathbb{R}$ is called the *real* part, and $y = \mathrm{Im}\{z\} \in \mathbb{R}$ is called the *imaginary* part of z. We call $\overline{z}$ the complex conjugate of z,

$$\overline{z} = \mathrm{Re}\{z\} - i\,\mathrm{Im}\{z\} \equiv x - iy. \tag{43}$$

We say $\overline{z}$ and z are each other's complex conjugate since $\overline{(\overline{z})} \equiv z$. The absolute value of a complex number is defined as $|z| = \sqrt{\overline{z} \cdot z}$. A different but useful notation is given by the real parameters $r \in \mathbb{R}_{\geq 0}$ and $\phi \in [0, 2\pi)$,

$$r = |z| = \sqrt{x^2 + y^2}, \qquad \begin{array}{l} x = r\cos(\phi), \\ y = r\sin(\phi). \end{array} \tag{44}$$

Inserting Eq. (44) into Eqs. (42) and (43) leads to

$$\begin{aligned} z &= x + iy = r\cos(\phi) + ir\sin(\phi) = re^{i\phi}, \\ \overline{z} &= x - iy = r\cos(\phi) - ir\sin(\phi) = re^{-i\phi}. \end{aligned} \tag{45}$$

B Vector spaces & $\mathbb{C}^n$

We will define vector spaces following the definition used in [20] and [21]. Given a field $(F, +, \cdot)$, a non-empty set V together with two operations

$$\begin{aligned} + &: V \times V \to V \quad \text{(Vector Addition)}, \\ * &: F \times V \to V \quad \text{(Scalar Multiplication)}, \end{aligned} \tag{46}$$

is called a vector space over F if the following axioms are satisfied.
Regarding vector addition "$+$" for all $u, v, w \in V$:

$$\begin{aligned} \textbf{(Associativity)} \quad & (u + v) + w = u + (v + w), \\ \textbf{(Neutral element)} \quad & 0_V \in V \quad \text{exists, so that} \quad 0_V + v = v + 0_V = v, \\ \textbf{(Inverse elements)} \quad & -v \in V \quad \text{exists, so that} \quad v + (-v) = 0_V, \\ \textbf{(Commutativity)} \quad & u + v = v + u. \end{aligned} \tag{47}$$

Regarding scalar multiplication "$*$" for $1_F \in F$, and all $\lambda, \mu \in F$; $u, v \in V$:

$$\begin{aligned} \textbf{(Compatibility)} \quad & (\lambda \cdot \mu) * v = \lambda * (\mu * v), \\ \textbf{(Neutral element)} \quad & 1_F * v = v, \\ \textbf{(Distributivity I)} \quad & (\lambda + \mu) * v = \lambda * v + \mu * v, \\ \textbf{(Distributivity II)} \quad & \lambda * (u + v) = \lambda * u + \lambda * v. \end{aligned} \tag{48}$$

We can, for example, define an n-dimensional complex vector space V over the field of complex numbers $\mathbb{C}$ as the set

$$V = \mathbb{C}^n = \left\{ \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} | u_1, u_2, \ldots, u_n \in \mathbb{C} \right\} \tag{49}$$

together with the operations

$$u + v = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} + \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} = \begin{pmatrix} u_1 + v_1 \\ u_2 + v_2 \\ \vdots \\ u_n + v_n \end{pmatrix} \quad \text{and} \quad \lambda * u = \lambda * \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} = \begin{pmatrix} \lambda \cdot u_1 \\ \lambda \cdot u_2 \\ \vdots \\ \lambda \cdot u_n \end{pmatrix}, \tag{50}$$

where $u, v \in \mathbb{C}^n$ and $\lambda \in \mathbb{C}$.

The vector space V can be equipped with an inner product, which allows us to define the length of any vector and the angle between two vectors. Such a space is often called an inner product space. The inner product over the complex vector space $\mathbb{C}^n$ is a function

$$\langle \cdot, \cdot \rangle : \mathbb{C}^n \times \mathbb{C}^n \to \mathbb{C} \tag{51}$$

that satisfies the following conditions for all $u, v, w \in \mathbb{C}^n$ and $\lambda, \mu \in \mathbb{C}$:

$$\begin{aligned} \textbf{(Linear in second argument)} \quad & \langle u, \lambda v + \mu w \rangle = \lambda \langle u, v \rangle + \mu \langle u, w \rangle, \\ \textbf{(Hermitian symmetric)} \quad & \langle u, v \rangle = \overline{\langle v, u \rangle}, \\ \textbf{(Positive definite)} \quad & \langle u, u \rangle \geq 0 \quad \text{and} \quad \langle u, u \rangle = 0 \text{ iff } u = 0. \end{aligned} \tag{52}$$

Now we can define the norm on $\mathbb{C}^n$ for any given vector by the function

$$\|\cdot\| : \mathbb{C}^n \to \mathbb{R}, \quad \|\cdot\| : u \mapsto \sqrt{\langle u, u \rangle}, \tag{53}$$

which satisfies the following conditions for all $u, v \in \mathbb{C}^n$ and $\lambda \in \mathbb{C}$:

$$\begin{aligned} \textbf{(Positive definite)} \quad & \|u\| \geq 0 \quad \text{and} \quad \|u\| = 0 \text{ iff } u = 0, \\ \textbf{(Homogeneity)} \quad & \|\lambda u\| = |\lambda| \|u\|, \\ \textbf{(Triangle inequality)} \quad & \|u + v\| \leq \|u\| + \|v\|. \end{aligned} \tag{54}$$

The norm $\|\cdot\|$ induces a notion of distance, i.e., a metric, between any two elements of $\mathbb{C}^n$. The metric is given by the function

$$d : \mathbb{C}^n \times \mathbb{C}^n \to \mathbb{R}, \quad d : (u, v) \mapsto \|u - v\|, \tag{55}$$

which satisfies the following conditions for all $u, v, w \in \mathbb{C}^n$:

$$\begin{aligned} \textbf{(Positive definite)} \quad & d(u, v) \geq 0 \quad \text{and} \quad d(u, v) = 0 \text{ iff } u = v, \\ \textbf{(Symmetry)} \quad & d(u, v) = d(v, u), \\ \textbf{(Triangle inequality)} \quad & d(u, v) \leq d(u, w) + d(w, v). \end{aligned} \tag{56}$$

C Hilbert spaces

The concept of a Hilbert space is used in quantum mechanics to describe the quantum system at hand. We may look at a given quantum state as a vector in the Hilbert space $\mathcal{H}$, where the dimension of $\mathcal{H}$ depends on the physical system being studied. We will follow the definition of a Hilbert space, as shown in [22]. Other useful definitions can be found in [21], [23] and [24].

The first step is to define a complete metric space. Given a set X and a function $d : X \times X \to \mathbb{R}$ that satisfies the conditions mentioned in (56), we say that (X, d) is a metric space. We can now define a Cauchy sequence in the metric space as $(x_n)_{n\in\mathbb{N}} \subseteq X$, for which there exists an $N_\varepsilon \in \mathbb{N}$ for all $\varepsilon \in \mathbb{R}^+$ so that

$$d(x_n, x_m) < \varepsilon, \quad \forall n, m \geq N_\varepsilon. \tag{57}$$

The metric space is called complete if every Cauchy sequence $(x_n)_{n\in\mathbb{N}}$ converges in X.

A Hilbert space $\mathcal{H}$ is an inner product space $(V, \langle\cdot,\cdot\rangle)$, which is also a complete metric space (V, d) regarding the metric induced by the norm.

In other words, $\mathcal{H}$ is a vector space equipped with an inner product that defines the norm $\|\cdot\|$, which induces the metric $d(\cdot,\cdot)$, and $\mathcal{H}$ is a complete metric space with regards to that metric.

For example, a finite-dimensional Hilbert space is given by the vector space $\mathbb{C}^n$ together with the standard definition of the inner product, given by

$$\langle u, v\rangle = \sum_{j=1}^{n} \overline{u_j} v_j, \tag{58}$$

with $u, v \in \mathbb{C}^n$. Note that the necessity for $\mathcal{H}$ to be a complete metric space becomes particularly relevant once we deal with infinite sums or integrals.

D Physical and logical qubits

We have seen in section 2.1 that the so-called qubit is the unit of quantum information. We often find some qubit systems referred to as *physical qubits*, others as *logical qubits* depending on their role.

A physical qubit is any physical system that fits the definition of section 2.1, i.e., a two-dimensional quantum system described by a two-dimensional complex Hilbert space $\mathcal{H}$. We sometimes find that qubits, on which the logical gates of a quantum computation act, are encoded in many physical qubits. These logical qubits can be encoded in a variety of ways. The simplest example is the following encoding

$$|0\rangle_L \equiv |000\rangle_P \quad |1\rangle_L \equiv |111\rangle_P \,, \tag{59}$$

where L denotes the logical qubits and P the physical qubits.

E The BB84 protocol

In the BB84 protocol [13], two parties, Alice and Bob, want to establish a shared key so that an eavesdropper (Eve) cannot figure it out by intercepting their communication. They can restart the protocol if they suspect that Eve has interfered. Alice initially generates a random register that contains elements from $\{|0\rangle, |1\rangle |+\rangle, |-\rangle\}$ with equal probability and sends it to Bob.

In our case, lures are superposition states of the form described in section 3.1.2, that have to be carefully chosen such that they do not interfere with the computation. Thus, the distribution of $\{|0\rangle, |1\rangle |+\rangle, |-\rangle\}$ will not be uniform in the quantum register and will be *unbalanced*. The lure plays the role of the so-called *raw key*, i.e., the "train of photons" that Alice sends to Bob mentioned in [13], .

It is a special case of the BB84 protocol because we have the following "role mapping" from the BB84 protocol to the EUBQC protocol.

$$\begin{aligned} \text{Alice}_{\text{ (BB84)}} &\mapsto \text{The Client} \\ \text{Bob}_{\text{ (BB84)}} &\mapsto \text{The Client} \\ \text{Eve}_{\text{ (BB84)}} &\mapsto \text{The Server} \end{aligned} \tag{60}$$

To detect an eavesdropper, the client needs to compare the lure she generated with the lure after traveling to and back from the server.

F Bell circuit fidelity for $N > 2$

We will assume that without loss of generality, the circuit is applied on the first and the second qubit of the quantum memory $\mathcal{M}$, starting in the $|0\rangle_{\mathcal{M}}^{\otimes N}$ state.

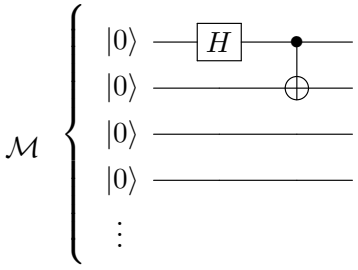

Figure 14: Quantum circuit representation of the noiseless computation on the server's N-qubit quantum memory $\mathcal{M}$, starting in the initial state $|\Psi\rangle_{\mathcal{M}} = |0\rangle^{\otimes N}$. The Bell circuit acts on the first two qubits of the quantum memory, resulting in the global memory state $|\Psi\rangle_{Bell,\mathcal{M}}$. We will write $|\Psi\rangle_{B,\mathcal{M}}$ for simplicity.

There are two distinct sections in the server's quantum memory: the section where the computation occurs, i.e., the first two qubits, and a section that remains idle. We will label the computation's section by $\mathcal{H}_C$, the remaining space, i.e., the idle section, by $\mathcal{H}_I$.

Note that $|\Psi\rangle_{B,\mathcal{M}} \in \mathcal{H}_C \otimes \mathcal{H}_I$, but since we are interested only in the computation, we will trace over subsystem $\mathcal{H}_I$. The resulting reduced state of the quantum memory in $\mathcal{H}_C$ after the computation is given by

$$\mathrm{Tr}_I(|\Psi\rangle\langle\Psi|_{B,\mathcal{M}}) = \left|\phi^+\right\rangle\left\langle\phi^+\right|, \tag{61}$$

see equation (17).

We will compute the fidelity using the noisy reduced state, i.e., the first two qubits' noisy state. If we consider the Λ_A noise map, the computation fidelity is given by

$$\mathrm{F}_{\phi^+,A} = \left\langle\phi^+\right| \sigma'_{\phi^+,A} \left|\phi^+\right\rangle, \tag{62}$$

where $\sigma'_{\phi^+,A}$ is the noisy reduced state of the memory. The state is given by

$$\sigma'_{\phi^+,A} = \mathrm{Tr}_I\Big(\quad \Lambda_A\Big(C_X(0,1), \Lambda_A(H_0, |0\rangle\langle 0|_{\mathcal{M}}^{\otimes N})\Big) \quad \Big), \tag{63}$$

using the more intuitive notation $\Lambda_A(|k\rangle, \sigma_{\mathcal{M}}) \equiv \Lambda_A(U_k, \sigma_{\mathcal{M}})$, see Eq. (29). We can analogously compute the fidelities for $M = \mathcal{M}$ and $M = \mathcal{M}A$, see Eq. (39).