EFFICIENT SIMULATION FOR QUANTUM MESSAGE AUTHENTICATION

Evelyn Wainewright

Thesis submitted to the Faculty of Graduate and Postgraduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Mathematics[1]

Department of Mathematics and Statistics
Faculty of Science
University of Ottawa

[1]The M.Sc. Program is a joint program with Carleton University, administered by the Ottawa-Carleton Institute of Mathematics and Statistics

Abstract

A mix of physics, mathematics, and computer science, the study of quantum information seeks to understand and utilize the information that can be held in the state of a quantum system. Quantum cryptography is then the study of various cryptographic protocols on the information in a quantum system. One of the goals we may have is to verify the integrity of quantum data, a process called quantum message authentication. In this thesis, we consider two quantum message authentication schemes, the *Clifford code* and the *trap code*. While both of these codes have been previously proven secure, they have not been proven secure in the simulator model, with an *efficient simulation*. We offer a new class of simulator that is efficient, so long as the adversary is efficient, and show that both of these codes can be proven secure using the efficient simulator. The efficiency of the simulator is typically a crucial requirement for a composable notion of security. The main results of this thesis have been accepted to appear in the Proceedings of the 9^{th} International Conference on Information Theoretic Security (ICITS 2016).

Acknowledgements

I would like to thank my supervisor, Dr. Anne Broadbent, for her continued support and mentorship through not only the writing of this thesis but also my journey into quantum information and quantum cryptography. She always knew the right question to ask and her guidance in finding the answers made me a better researcher and mathematician.

This process has reiterated to me how grateful I am for the continuous support, and patience, that I have received from my family. I could not have done this without them.

Finally, I would like to thank the University of Ottawa for their financial support of this research, and the faculty and staff in the Mathematics and Statistics department who have always been available and welcoming.

Contents

List of Figures

Chapter 1

Introduction

Quantum cryptography was first popularized by the publication of a *quantum key distribution* protocol, presented by Bennett and Brassard, called BB84 [BB84]. The BB84 protocol sends quantum states between two parties in order to establish a shared classical key, from an initial short key. BB84 was the first instance which used quantum particles in the context of securing information, and it spurred an interest in the ways in which we can use quantum information to secure classical data. As the field progressed, some researchers asked if we could achieve the same, or better, results on quantum data.

Since the publication of [BB84], the field of quantum cryptography has come a long way. While key distribution is still often thought of as the most successful quantum cryptographic technology, [BEM+07, Feh10], there is actually a depth of research that has been presented achieving various cryptographic goals on quantum data. We have seen both proofs of protocols and impossibility proofs for many different cryptographic protocols. For example, bit commitment, the idea that you can choose a bit and not be able to alter it, but also not have to reveal it until a certain time, was shown to be impossible to prove unconditional security in the quantum setting [BCMS97]. On the other hand, universal blind computations, where a client can have a server perform computations without the server knowing the client's input, output, or computations, have been shown to be possible in the quantum setting [BFK09]. We have seen protocols that offer quantum fully homomorphic encryption, that is, encryption

that would allow for computations to be performed on the data without having to decrypt it, for an increasingly large family of quantum circuits [BJ15,DSS16]. We have seen the development of quantum secret sharing [CGL99], quantum multiparty computation [BCG$^+$06], and quantum oblivious transfer [BBCS01], to name just a few [BS16]. Quantum message authentication is another fundamental cryptographic goal that we have achieved [BCG$^+$02, BCG$^+$06, ABE10, BGS13, DNS12].

Quantum message authentication schemes are families of keyed encoding and decoding maps, designed to detect tampering on encoded quantum data. They were first introduced in [BCG$^+$02] where they were given in a very efficient form based on *purity testing* and were shown to satisfy a composable notion of security [HLM11].

Since then, additional quantum message authentication codes have been introduced including the *signed polynomial code* [BCG$^+$06, ABE10], the *Clifford code* [ABE10, DNS12], and the *trap code* [BGS13]. In this thesis, we focus on the Clifford and trap codes since they have a nice structure that makes them not only easy to study, but also useful within other protocols. Since the Cliffords and Paulis, applied to the messages in the Clifford and trap codes respectively, are highly structured, they allow for certain types gates to be performed through the authentication, called *quantum computation on authenticated data (QCAD)* [BGS13].

Part of the challenge in this field is how new it still is. We talk about each of the protocols above being proven to be secure or shown to be impossible to prove security, but they do not all follow the same proof model and in some cases do not prove security under the same assumptions. Even if we look specifically at the five different papers mentioned above with quantum message authentication schemes of some sort, they have different security definitions.

One of the ways that we can address this issue is to introduce the concept of a composable security notion [HLM11]. A composable notion of security is the idea that protocols proven secure within this definition can be composed and still be secure. Clearly this is a desirable property, especially when working with protocols that perform basic tasks which are likely to be implemented within larger protocols.

More formally, as described in [BW16], based on the work of [DNS12], the security of quantum message authentication schemes is typically defined in terms of the

existence of an ideal channel that consists of a *simulator* and the *ideal functionality*. The simulator only acts on the reference system of the message along and has access to the attack. Based on this information it outputs to the ideal functionality whether to act as the identity on the input message and accept, or to reject the message and replace it with a fixed state. The protocol is secure if the the real-world protocol (involving the adversary) is statistically indistinguishable from the ideal-world protocol (involving the simulator). This type of definition fits in the quantum Universal Composability (UC) [Can01, Unr10] framework, as long as we add a further condition: if the adversary runs in polynomial time, so must the simulator (an *efficient* simulation). Until now, direct efficient simulations were known only for the purity-testing based codes [BCG$^+$02].

The work in this thesis focuses on defining a simulator that is efficient and then using that simulator to prove the security of two quantum message authentication schemes, the *Clifford code* and the *trap code*.

1.1 Structure

The thesis is structured as follows. Chapter 2 gives an overview of the quantum information background required. Chapter 3 gives an overview of the work in quantum message authentication prior to this research, and then provides the necessary definitions to prove security. Chapter 4 describes the two codes that are considered in this thesis, the Clifford code (Section 4.1) and the trap code (Section 4.2). Chapter 5 offers the security proof for the two codes, in addition to defining the simulators required in each case. Finally, Chapter 6 discusses some the problems that further work in this area could explore and Appendix A offers a summary of the notation used. We note that the main results of this thesis have been accepted for publication and will appear in the Proceedings of the 9^{th} International Conference on Information Theoretic Security (ICITS 2016). As such, Chapters 4 and 5 follow [BW16] closely.

Chapter 2

Quantum Information

Quantum information is a fascinating combination of quantum mechanics, computer science, and information theory. In contrast to classical computing which uses the bit as its fundamental concept, quantum information and quantum computing uses the quantum bit or *qubit*. Physically, a qubit is given as a two level quantum system. A common physical interpretation would be a light particle, called a *photon*, that is polarized in one of two ways, either vertically or horizontally. Mathematically, we use two dimensional complex vectors to represent the state of a *pure* qubit. The pure qubit is the fundamental unit within quantum information but we will introduce a more abstract version, a *mixed state* qubit, later. The concept of a qubit can also be further generalized into a d-level system, called a *qudit*. We will not need this generalization, but simply note that quantum information is not limited to two level systems.

Throughout this thesis, we will use Dirac's *"bra-ket"* notation to express row and column vectors. A *ket* is used to denote a column vector, for example $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, and a *bra* is used to denote a row vector, for example $\langle 0| = \begin{bmatrix} 1 & 0 \end{bmatrix}$.

Using the bra-ket notation we can simplify our notation significantly. In addition to representing row and column vectors easily, we can also simplify the notation for certain vector operations. The inner product, or dot product, of two vectors, ϕ and ψ is given by $\langle \phi \,|\, \psi \rangle$. The outer product is given by $|\phi\rangle \langle \psi|$.

In order to introduce all of the characteristics of qubits, we will use the four postulates of quantum mechanics. We will follow the presentation of the postulates of Nielson and Chuang [NC00].

2.1 State Space

Postulate 1. Associated to any isolated physical system is a complex vector space with inner product (that is, a Hilbert space) known as the *state space* of the system. The system is completely described by its *state vector*, which is a unit vector in the system's state space.

This postulate allows us to represent any qubit by its state vector. In quantum information we have two unit vectors that are ubiquitous, $|0\rangle$ and $|1\rangle$. They form what we call the *computational basis*, which is defined as:

Definition 2.1.1. The computational basis is given by two perpendicular vectors, represented using the Dirac bra-ket notation as:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{and} \, |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

We will also often use what we call the *Hadamard Basis*.

Definition 2.1.2. The Hadamard basis is given by two perpendicular vectors:

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

As we can see in the definition of the Hadamard basis, the states of qubits are not limited to simply $|0\rangle$ or $|1\rangle$. Quantum states can also be in what is called a *superposition* of states. This is a fundamental difference between classical and quantum information. We can denote an arbitrary qubit by $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$, where $\alpha, \beta \in \mathbb{C}$ and $|\alpha|^2 + |\beta|^2 = 1$. This state is in a superposition of the two computational basis states where the squared norms of the complex coefficients, α and β, give a probability distribution. While the state should be seen to be in both $|0\rangle$ and $|1\rangle$ at the same time, the squared norms of the coefficients give the probability that, when

measured in the computational basis, the result will be 0 or 1. In other words, the qubit, when measured, will be in the state $|0\rangle$ with probability $|\alpha|^2$ and similarly, the probability that the qubit will be measured in the state $|1\rangle$ is given by $|\beta|^2$. This of course begs the question of what it means to measure a qubit. This will be addressed in Postulate 3.

We note that we often use the computational basis and the Hadamard basis because the two are mutually unbiased. Specifically, the squared norm of the inner product of any basis state in the computational basis with any basis state in the Hadamard basis will be $\frac{1}{2}$. Physically, this can be interpreted as a basis state from the computational basis, when measured in the Hadamard basis, has an equal probability of being in either of the basis states of the Hadamard basis and vice versa.

2.2 Evolution

Postulate 2. The evolution of a *closed* quantum system is described by a *unitary transformation*. That is, the state $|\psi\rangle$ of the system at t_1 is related to the state $|\psi'\rangle$ of the system at time t_2 by a unitary operator U which depends only on the times t_1 and t_2,

$$|\psi'\rangle = U|\psi\rangle . \tag{1}$$

We use unitary operators to describe quantum gates or the intentional, controlled evolution of qubits. For this reason, an attack on a system is typically represented as an arbitrary unitary operation. There is an important class of unitary operators that are used frequently in quantum cryptography, the Pauli matrices.

The Pauli Matrices for single-qubit gates are given by:

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \; X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \; Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \text{ and } Y = iXZ = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}. \tag{2}$$

Where the X Pauli is analogous to the classical bit flip (swapping the $|0\rangle$ and $|1\rangle$), and the Z Pauli is considered the phase flip (flipping the sign of β in the arbitrary qubit $\alpha|0\rangle + \beta|1\rangle$). The I Pauli is, of course, the identity which does nothing to the qubit. It is not hard to see that these matrices form a basis for all 2×2 complex

matrices, when we allow complex coefficients. We can therefore decompose any single qubit gate into a linear combination, with complex coefficients, of single qubit Pauli matrices. We will discuss the Paulis in more detail in Section 2.11.

Another important unitary operator is the Hadamard gate, which takes basis states from the computational basis to the Hadamard basis and vice versa. The Hadamard gate is given by:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{3}$$

The Hadamard matrix applied to each of the basis states results in the following transformations:

$$H|0\rangle = |+\rangle\,,\ H|1\rangle = |-\rangle\,,\ H|+\rangle = |0\rangle\,,\ H|-\rangle = |1\rangle\,. \tag{4}$$

2.3 Quantum Measurement

Postulate 3. Quantum measurements are described by a collection $\{M_m\}$ of *measurement operators*. These are the operators acting on the state space of the system being measured. The index m refers to the measurement outcomes that may occur in the experiment. If the state of the quantum system is $|\psi\rangle$ immediately before the measurement then the probability that the result m occurs is given by

$$p(m) = \langle\psi| M_m^\dagger M_m |\psi\rangle\,, \tag{5}$$

and the state of the system after the measurement, given that m was observed, is

$$\frac{M_m|\psi\rangle}{\sqrt{\langle\psi| M_m^\dagger M_m |\psi\rangle}}. \tag{6}$$

The measurement operators satisfy the *completeness equation*,

$$\sum_m M_m^\dagger M_m = I. \tag{7}$$

The completeness equation expresses the fact that probabilities sum to one:

$$1 = \sum_m p(m) = \sum_m \langle\psi| M_m^\dagger M_m |\psi\rangle\,. \tag{8}$$

Now when we look at our arbitrary qubit, $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, we can define the measurement projectors $M_0 = |0\rangle\langle 0|$ and $M_1 = |1\rangle\langle 1|$. We can see that $p(0) = \langle\psi| M_0^\dagger M_0 |\psi\rangle = |\alpha|^2$, as previously claimed, and the state after measurement is given by $\frac{M_0|\psi\rangle}{\sqrt{\langle\psi|M_0^\dagger M_0|\psi\rangle}} = \frac{\alpha}{\sqrt{\alpha^2}}|0\rangle = |0\rangle$. Importantly, we also note that once measured the state is changed. In this case, if we measure and find the state to be $|0\rangle$, we lose any information about what the coefficients α and β were.

We note that we can use this construction to distinguish between orthonormal states, however, we cannot reliably distinguish between non-orthogonal states. A proof of this can be found in [NC00].

2.4 Composite Systems

Postulate 4. The state space of a composite physical system is the tensor product of the state spaces of the component physical systems. Moreover, if we have systems numbered 1 through n, and the system number i is prepared in the state $|\psi_i\rangle$, then the joint state of the total system is $|\psi_1\rangle \otimes |\psi_2\rangle \otimes \ldots \otimes |\psi_n\rangle$.

Similarly, if we want to talk about the operators acting on a composite system, we refer to the tensor product, defined in Section 2.5, of the operators acting on each component system.

2.5 Tensor Products

We will follow the presentation of [NC00] in defining the tensor product. The *tensor product* is a way of combining two vector spaces to form a larger vector space.

If we let V and W be vectors spaces of dimension m and n respectively, and further assume that V and W are Hilbert spaces, since this is the context in which we will use the definition, then $V \otimes W$ is an mn dimensional vector space. For $|v\rangle$ of V and $|w\rangle$ of W, the elements of $V \otimes W$ are linear combinations of tensor products $|v\rangle \otimes |w\rangle$. In particular, if $|i\rangle$ and $|j\rangle$ are orthonormal bases for the spaces V and W then $|i\rangle \otimes |j\rangle$ is a basis for $V \otimes W$.

By definition the tensor product satisfies the following basic properties:

1. For an arbitrary scalar z and elements $|v\rangle$ of V and $|w\rangle$ of W,

$$z(|v\rangle \otimes |w\rangle) = (z\,|v\rangle) \otimes |w\rangle = |v\rangle \otimes (z\,|w\rangle).$$

2. For arbitrary $|v_1\rangle$ and $|v_2\rangle$ of V and $|w\rangle$ of W,

$$(|v_1\rangle + |v_2\rangle) \otimes |w\rangle = |v_1\rangle \otimes |w\rangle + |v_2\rangle \otimes |w\rangle\,.$$

3. For arbitrary $|v\rangle$ of V and $|w_1\rangle$ and $|w_2\rangle$ of W,

$$|v\rangle \otimes (|w_1\rangle + |w_2\rangle) = |v\rangle \otimes |w_1\rangle + |v\rangle \otimes |w_2\rangle\,.$$

Finally, we note that if we have two linear operators, A and B acting on V and W, respectively, we can define a linear operator $(A \otimes B)(V \otimes W) \equiv A\,|v\rangle + B\,|w\rangle$.

2.6 Entanglement

Another key difference between classical and quantum information is that quantum states can be entangled. Entanglement expresses the fact that the states of two qubits can be codependent. If you measure one qubit and force it to be in a certain state that can also influence the second qubit, even if they are physically separated. Mathematically, this is a fairly simple concept to express. When we talk about the state of a composite physical system being the tensor product of the component physical systems, a 2-qubit entangled state, $|\psi\rangle$ is entangled if there are no single qubit states $|\gamma\rangle$ and $|\delta\rangle$ such that $|\psi\rangle = |\gamma\rangle \otimes |\delta\rangle$. This notion can be extended to have entanglement between more than 2 qubits, as well. In simple terms, an entangled state is the smallest component system of a composite system, even though it contains more than one qubit.

We denote a two-qubit maximally entangled pure state as $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. This is one of four Bell states. The other three Bell states are also maximally entangled pure states, $|\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$, $|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$, and $|\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$. The four Bell states are orthogonal and therefore perfectly distinguishable and so we can perform a projective measurement into the Bell basis and determine which of the four Bell states we have. This is called a *Bell basis measurement*.

2.7 Density Operators and Mixed States

Now that we are comfortable with the state vector formalism, we reveal that it is not a complete description of possible quantum states. The state vector formalism refers only to what we call *pure states*. In addition to pure states, we can have a more general notion, *mixed states*. Mixed states require a different formalism, the *density operator* or *density matrix*. The density operator is the most general notion of quantum information. The density operator allows us to express states that are in some sort of probabilistic mixture of pure states. If the state is in $|\psi_i\rangle$ with probability p_i, (where i belongs to some finite set $\mathcal{I}$ that indexes pure states), then we denote an *ensemble of pure states* with $\{p_i, |\psi_i\rangle\}$. The density operator for the system is defined by the equation:

$$\rho \equiv \sum_{i \in \mathcal{I}} p_i \left|\psi_i\right\rangle \left\langle\psi_i\right| . \tag{9}$$

Clearly, given ρ, it is not possible to uniquely determine an ensemble that gives ρ since this form is not, in general, unique. For example, the ensemble of $|0\rangle$ and $|1\rangle$ with equal probability and the ensemble that contains $|+\rangle$ and $|-\rangle$ with equal probability gives the same density operator, $\rho = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}$.

There are two additional conditions that the density operator must satisfy; ρ must have trace (sum of the diagonal entries of the matrix, denoted tr) equal to one, and ρ must be a positive semidefinite matrix, which means that if ρ is an $n \times n$ complex matrix, then for every non-zero column vector of n complex numbers, m, the scalar $m^\dagger \rho m$ is greater than or equal to 0.

Clearly, pure states can be represented in this same formalism, where instead of a sum, we have just one term, $|\psi\rangle \langle\psi|$.

Now that we have expanded the states we can represent, we can reformulate the postulates using the density operator notation:

Postulate 1. Associated to any isolated physical system is a complex vector space with inner product (that is, a Hilbert space) known as the *state space* of the system. The system is completely described by its *density operator*, which is a positive operator ρ

with trace one, acting on the state space of the system. Given a finite set of indexes of density operators, $\mathcal{I}$, if a quantum system is in the state ρ_i with probability p_i, for $i \in \mathcal{I}$ then the density operator for the system is $\sum_{i\in\mathcal{I}} p_i\rho_i$.

Given the conditions on the density operators, we know that density operators can equivalently be seen as compact self-adjoint linear operators in the Hilbert space [GHW09].

Postulate 2. The evolution of a *closed* quantum system is described by a *unitary transformation*. That is, the state ρ of the system at time t_1 is related to the state of ρ' of the system at time t_2 by a unitary operator U which depends only on the times t_1 and t_2,

$$\rho' = U\rho U^\dagger. \tag{10}$$

In the algebraic formalism, we can see a unitary transformation as a bounded linear operator, $U : \mathcal{H} \to \mathcal{H}$, where $\mathcal{H}$ is a Hilbert space, I is the identity, and where $U^\dagger U = UU^\dagger = I$, or U is self-adjoint. Further to that, we observe that any B in the space of bounded linear operators acting on a Hilbert space, $\mathcal{H}$, is a linear combination of self-adjoint bounded linear operators, or unitary operators.

Postulate 3. Quantum measurements are described by a collection $\{M_m\}$ of *measurement operators*. These are operators acting on the state space of the system being measured. The index m refers to the measurement outcomes that may occur in the experiment. If the state of the quantum system is ρ immediately before the measurement then the probability that result m occurs is given by

$$p(m) = tr(M_m^\dagger M_m \rho), \tag{11}$$

and the state of the system after the measurement, given the outcome m was observed, is

$$\frac{M_m \rho M_m^\dagger}{tr(M_m^\dagger M_m \rho)}. \tag{12}$$

The measurement operators satisfy the *completeness equation*,

$$\sum_m M_m^\dagger M_m = I. \tag{13}$$

Postulate 4. The state space of a composite physical system is the tensor product of the state spaces of the component physical systems. Moreover, if we have systems numbered 1 through n, and system number i is prepared in the state ρ_i, then the joint state of the total system is $\rho_1 \otimes \rho_2 \otimes \ldots \otimes \rho_n$.

2.8 Quantum Channels

Here we continue to generalize to quantum channels, the most general quantum operation. Intuitively, quantum channels are operations that take quantum states to quantum states. Formally, if ρ is a valid density operator, then a quantum channel, T, applied to ρ, given by $T(\rho)$, must also be a valid density operator. In order for this to be the case, T must be *trace preserving*, that is, $tr(\rho) = tr(T(\rho))$. Furthermore, since ρ is a positive matrix, $T(\rho)$ must be as well. However, we also want to be able to apply a quantum channel to only a part of a system and still be valid, so we require that $(T \otimes \mathbb{I}_n)(\rho)$ is a positive matrix, for all values of $n \in \mathbb{N}$. This requirement is equivalent to being *completely positive*. Together, these two requirements give us the description of a quantum channel, a *completely positive trace preserving (CPTP)* map.

2.9 Trace Norm and Trace Distance

We will often need a measure of distinguishability between quantum states. To do this, we use the *trace distance*. The trace distance is a value between zero and one that represents the probability that given two states, ρ and σ, an observer would be able to distinguish between the two states using a single measurement. It is equal to zero if and only if ρ and σ are the same state, and therefore perfectly indistinguishable, and it is equal to one if and only if ρ and σ have orthogonal supports, in which case they are perfectly distinguishable. The trace distance is denoted $D(\rho, \sigma)$ and is defined in terms of the *trace norm* as $D(\rho, \sigma) = \frac{1}{2} \|\rho - \sigma\|_1$. The trace norm of a state, $\|\rho\|_1$, is defined as $\|\rho\|_1 = tr[\sqrt{\rho^\dagger \rho}] = \sum_i \sqrt{\lambda_i}$, where the λ_i are the eigenvalues of the

matrix $\rho^\dagger\rho$. The trace norm, and therefore the trace distance, satisfies the triangle inequality: $\|\rho+\sigma\|_1 \leq \|\rho\|_1 + \|\sigma\|_1$.

2.10 Permutations

When we want to change the order of the qubits in a system, we will apply a permutation map. We use Π_n to denote the set of all permutation maps on n qubits. A permutation map, denoted throughout by π, is a unitary operation that acts on n qubits and permutes the order of the n qubits. This can equivalently be seen as a permutation, σ, of the indices of the qubits, where π would take the i^{th} qubit to the $\sigma(i)^{th}$ position. Permutation maps are orthogonal, real valued matrices so $\pi^{-1} = \pi^\dagger$.

2.11 Pauli Matrices

Recall that the four single qubit *Pauli matrices* are given by:

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \text{ and } Y = iXZ = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \tag{14}$$

and any single qubit quantum gate can be written as a complex linear combination of the four Pauli matrices.

Then an n-qubit Pauli matrix is given by the n-fold tensor product of potentially different single-qubit Pauli matrices. We denote the set of all n-qubit Pauli matrices by $\mathbb{P}_n$, where $|\mathbb{P}_n| = 4^n$. We can similarly decompose any n-qubit operator into a linear sum of Paulis. We write any n-qubit unitary as $U = \sum_{P\in\mathbb{P}_n} \alpha_P P$, with $\sum_{P\in\mathbb{P}_n} |\alpha_P| = 1$. This is called the *Pauli decomposition* of a unitary quantum operation. When the number of qubits is clear from context, or unimportant, we will often simply refer to them as "Pauli matrices" or "Paulis".

The *Pauli weight* of an n-qubit Pauli, denoted $\omega(P)$, is the number of non-identity Paulis in the n-fold tensor product. We will also define sets of Paulis composed only of specific Pauli matrices, such as $\{I, X\}^{\otimes n}$ which is the set of all n-qubit Paulis composed of only I and X Paulis, or $\{I, Z\}^{\otimes n}$ which is the set of all n-qubit Paulis

composed of only I and Z Paulis. Additionally, we note that since we do not distinguish Pauli matrices by a coefficient of ± 1 or $\pm i$, an arbitrary Pauli can be defined by whether or not there is an X and/or a Z Pauli at each index. Therefore, given two n-bit strings, a and b, then we can write any $P \in \mathbb{P}_n$ as $P = X^a Z^b$, where X^a indicates the n-fold tensor product of I in every index that a has a 0, and X in every index that a has a 1 and Z^b is defined the same way, as the n-fold tensor product of I in every index that b has a 0, and Z in every index that b has a 1. This means that we can uniquely, up to a multiplicative factor of $\{\pm 1, \pm i\}$, identify a Pauli by giving two n-bit strings. We also note that since X and Z anti-commute, for a single qubit Pauli, $X^a Z^b = (-1)^{ab} Z^b X^a$. For an n-qubit Pauli, we have that for each index i from 1 to n, $X^{a_i} Z^{b_i} = (-1)^{a_i b_i} Z^{b_i} X^{a_i}$. Finally, Paulis are self-inverses, so $P = P^{-1} = P^\dagger$.

The following lemma, called the *Pauli Twirl* [DCEL09], shows how we can greatly simplify expressions that involve the twirling of an operation by the Pauli matrices. The proof of this lemma closely follows the technique in [Bro15, ABE10], but extends to the general case for n-qubit Paulis. We also note that an alternate proof of this theorem appeared in [DCEL09].

Lemma 2.11.1 (Pauli Twirl)**.** *Let P, P' be Pauli operators in $\mathbb{P}_n$. Then for any density operator, ρ, it holds that:*

$$\frac{1}{|\mathbb{P}_n|} \sum_{Q \in \mathbb{P}_n} Q^\dagger P Q \rho Q^\dagger P'^\dagger Q = \begin{cases} 0, & P \neq P' \\ P \rho P^\dagger, & \textit{otherwise}. \end{cases}$$

Proof. Suppose $P = X^a Z^b$ and $P' = X^{a'} Z^{b'}$, where a, b, a', b' are binary strings of length n. We will first consider the case where $P \neq P'$. Then, in this case, there exists at least one index where at least one bit of a disagrees with one bit of a' or one bit of b disagrees with one bit of b'. We will call this index j, and remark $a_j \neq a'_j$ and/or $b_j \neq b'_j$.

Next, we will consider how the term $Q^\dagger P Q$ behaves. We can similarly write Q as $X^c Z^d$, where c and d are binary strings of length n. Then:

$$\begin{aligned} Q^\dagger P Q &= (X^c Z^d)^\dagger X^a Z^b X^c Z^d \\ &= Z^d X^c X^a Z^b X^c Z^d. \end{aligned} \tag{15}$$

Now we can rearrange this expression if we know how the Paulis commute or anti-commute. We know that for the i^{th} single qubit Pauli in the n-fold tensor product, we can describe whether or not the pairs of Paulis X^a and Z^d, and Z^b and X^c commute at that index by $(-1)^{c_i b_i}(-1)^{d_i a_i}$, where a_i, b_i, c_i, d_i denotes the i^{th} bit in each of the n-bit strings.

Then if we rewrite $Z^d X^c X^a Z^b X^c Z^d$ in terms of $Z^d X^c X^c Z^d X^a Z^b$, the necessary coefficient is given by:

$$\prod_{i=1}^{n}(-1)^{c_i b_i}(-1)^{d_i a_i} = (-1)^{\bigoplus_{i=0}^{n} c_i b_i \oplus d_i a_i}. \tag{16}$$

Therefore,

$$\begin{aligned} Q^\dagger P Q &= Z^d X^c X^a Z^b X^c Z^d \\ &= (-1)^{\bigoplus_{i=0}^{n} c_i b_i \oplus d_i a_i} Z^d X^c X^c Z^d X^a Z^b \\ &= (-1)^{\bigoplus_{i=0}^{n} c_i b_i \oplus d_i a_i} X^a Z^b. \end{aligned} \tag{17}$$

We can do the same for the term with P':

$$\begin{aligned} Q^\dagger P' Q &= Z^d X^c X^{a'} Z^{b'} X^c Z^d \\ &= (-1)^{\bigoplus_{i=0}^{n} c_i b'_i \oplus d_i a'_i} Z^d X^c X^c Z^d X^{a'} Z^{b'} \\ &= (-1)^{\bigoplus_{i=0}^{n} c_i b'_i \oplus d_i a'_i} X^{a'} Z^{b'}. \end{aligned} \tag{18}$$

Now let us consider the twirled state, using the above expressions:

$$
\begin{aligned}
&\frac{1}{|\mathbb{P}_n|}\sum_{Q\in\mathbb{P}_n} Q^\dagger PQ\rho Q^\dagger P'^\dagger Q \\
&= \frac{1}{|\mathbb{P}_n|}\sum_{c,d} Z^d X^c X^a Z^b X^c Z^d \rho Z^d X^c X^{a'} Z^{b'} X^c Z^d \\
&= \frac{1}{|\mathbb{P}_n|}\sum_{c,d} (-1)^{\bigoplus_{i=0}^{n} c_i b_i \oplus d_i a_i} Z^d X^c X^c Z^d X^a Z^b \rho (-1)^{\bigoplus_{i=0}^{n} c_i b'_i \oplus d_i a'_i} Z^d X^c X^c Z^d X^{a'} Z^{b'} \\
&= \frac{1}{|\mathbb{P}_n|}\sum_{c,d} (-1)^{\bigoplus_{i=0}^{n} c_i b_i \oplus d_i a_i} X^a Z^b \rho (-1)^{\bigoplus_{i=0}^{n} c_i b'_i \oplus d_i a'_i} X^{a'} Z^{b'} \\
&= \frac{1}{|\mathbb{P}_n|}\sum_{c,d} (-1)^{\bigoplus_{i=0}^{n} c_i b_i \oplus d_i a_i \oplus c_i b'_i \oplus d_i a'_i} X^a Z^b \rho X^{a'} Z^{b'} \\
&= \frac{1}{|\mathbb{P}_n|}\sum_{c,d} (-1)^{\bigoplus_{i=0}^{n} c_i (b_i \oplus b'_i) \oplus d_i (a_i \oplus a'_i)} X^a Z^b \rho X^{a'} Z^{b'}. \qquad (19)
\end{aligned}
$$

From here we can pull out the j^{th} index from the rest (recall that the j^{th} index is where at least one of a, a' or b, b' disagree):

$$
= \frac{1}{|\mathbb{P}_n|}\sum_{c,d} (-1)^{c_j (b_j \oplus b'_j) \oplus d_j (a_j \oplus a'_j)} (-1)^{\bigoplus_{i=0, i\neq j}^{n} c_i (b_i \oplus b'_i) \oplus d_i (a_i \oplus a'_i)} X^a Z^b \rho X^{a'} Z^{b'}. \qquad (20)
$$

Since we have at least one of $a_j \neq a'_j$ or $b_j \neq b'_j$, then we have three possible scenarios:

1. $a_j = a'_j$, $b_j \neq b'_j$, then:

$$
= \frac{1}{|\mathbb{P}_n|}\sum_{c,d} (-1)^{c_j (1) \oplus d_j (0)} (-1)^{\bigoplus_{i=0, i\neq j}^{n} c_i (b_i \oplus b'_i) \oplus d_i (a_i \oplus a'_i)} X^a Z^b \rho X^{a'} Z^{b'}. \qquad (21)
$$

And since half of all c terms will have $c_j = 0$ and half will have $c_j = 1$, then we have:

$$
\begin{aligned}
&= \frac{1}{|\mathbb{P}_n|}\sum_{c,d} \left(\frac{1}{2}\right) (-1)^{\bigoplus_{i=0, i\neq j}^{n} c_i (b_i \oplus b'_i) \oplus d_i (a_i \oplus a'_i)} X^a Z^b \rho X^{a'} Z^{b'} \\
&\quad - \frac{1}{2} (-1)^{\bigoplus_{i=0, i\neq j}^{n} c_i (b_i \oplus b'_i) \oplus d_i (a_i \oplus a'_i)} X^a Z^b \rho X^{a'} Z^{b'} \\
&= 0. \qquad (22)
\end{aligned}
$$

We can use the same logic for d in the next case:

2. $a_j \neq a'_j$, $b_j = b'_j$, then:

$$\begin{aligned}
&= \frac{1}{|\mathbb{P}_n|} \sum_{c,d} (-1)^{c_j(0) \oplus d_j(1)} (-1)^{\bigoplus_{i=0, i\neq j}^{n} c_i(b_i \oplus b'_i) \oplus d_i(a_i \oplus a'_i)} X^a Z^b \rho X^{a'} Z^{b'} \\
&= \frac{1}{|\mathbb{P}_n|} \sum_{c,d} \left(\frac{1}{2}\right) (-1)^{\bigoplus_{i=0, i\neq j}^{n} c_i(b_i \oplus b'_i) \oplus d_i(a_i \oplus a'_i)} X^a Z^b \rho X^{a'} Z^{b'} \\
&\quad - \frac{1}{2} (-1)^{\bigoplus_{i=0, i\neq j}^{n} c_i(b_i \oplus b'_i) \oplus d_i(a_i \oplus a'_i)} X^a Z^b \rho X^{a'} Z^{b'} \\
&= 0. \qquad (23)
\end{aligned}$$

3. And finally, $a_j \neq a'_j$, $b_j \neq b'_j$, then:

$$= \frac{1}{|\mathbb{P}_n|} \sum_{c,d} (-1)^{c_j(1) \oplus d_j(1)} (-1)^{\bigoplus_{i=0, i\neq j}^{n} c_i(b_i \oplus b'_i) \oplus d_i(a_i \oplus a'_i)} X^a Z^b \rho X^{a'} Z^{b'}. \qquad (24)$$

Here we have that again, of the possible c and d terms, $c_j \oplus d_j$ will be 0 in half of the cases (when c_j and d_j agree) and $c_j \oplus d_j$ will be 1 in half of the cases (when c_j and d_j disagree). Therefore:

$$\begin{aligned}
&= \frac{1}{|\mathbb{P}_n|} \sum_{c,d} \left(\frac{1}{2}\right) (-1)^{\bigoplus_{i=0, i\neq j}^{n} c_i(b_i \oplus b'_i) \oplus d_i(a_i \oplus a'_i)} X^a Z^b \rho X^{a'} Z^{b'} \\
&\quad - \frac{1}{2} (-1)^{\bigoplus_{i=0, i\neq j}^{n} c_i(b_i \oplus b'_i) \oplus d_i(a_i \oplus a'_i)} X^a Z^b \rho X^{a'} Z^{b'} \\
&= 0. \qquad (25)
\end{aligned}$$

As we can see, in each case, the sum when $P \neq P'$ is 0. The last thing to do is consider the case when $P = P'$. In this case, since $a_i = a'_i$ and $b_i = b'_i$ for all i, we

can write our sum as:

$$
\begin{aligned}
&= \frac{1}{|\mathbb{P}_n|} \sum_{c,d} (-1)^{\bigoplus_{i=0}^{n} c_i(b_i \oplus b_i') \oplus d_i(a_i \oplus a_i')} X^a Z^b \rho X^{a'} Z^{b'} \\
&= \frac{1}{|\mathbb{P}_n|} \sum_{c,d} (-1)^{\bigoplus_{i=0}^{n} c_i(0) \oplus d_i(0)} X^a Z^b \rho X^{a'} Z^{b'} \\
&= \frac{1}{|\mathbb{P}_n|} \sum_{c,d} X^a Z^b \rho X^a Z^b \\
&= P \rho P. \qquad \square
\end{aligned}
$$

2.12 Clifford Group

The *Clifford group*, $\mathcal{C}_n$, on n qubits are unitaries that map Pauli matrices to Pauli matrices (up to a phase of ± 1 or $\pm i$). Specifically, if $P \in \mathbb{P}_n$, then for all $C \in \mathcal{C}_n$, there exists an $\alpha \in \{\pm 1, \pm i\}$ such that $\alpha CPC^\dagger \in \mathbb{P}_n$. An alternate way to describe the Cliffords is that they are generated by the following single qubit gates:

$$P = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

along with the two qubit gate,

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

As can be seen in [Got97], the size of the Clifford group, while unwieldy, is given by $|\mathcal{C}_n| = 2^{n^2+2n} \prod_{j=i}^{n} (4^j - 1)$.

An important trait of the Cliffords is that they not only map Paulis to Paulis, but they do so with a uniform distribution [ABE10]. If we let P be a non-identity n-qubit Pauli operator, then applying an average over all n-qubit Cliffords to P by conjugation maps P to a uniformly distributed mixture of all non-identity n-qubit Pauli operators. More formally:

Lemma 2.12.1 (Clifford Randomization)**.** *For every P, $Q \in \mathbb{P}_n \setminus \{\mathbb{I}\}$, it holds that:*

$$\left|\{C \in \mathcal{C}_n | \exists \alpha \in \{\pm 1, \pm i\}, \alpha C^\dagger P C = Q\}\right| = \frac{|\mathcal{C}_n|}{|\mathbb{P}_n| - 1}. \tag{26}$$

The proof of this lemma can be found in [ABE10].

Additionally, there is a lemma that is analogous to the Pauli twirl [DCEL09]:

Lemma 2.12.2 (Clifford Twirl)**.** *Let $P \neq P'$ be Pauli operators. For any ρ it holds that:*

$$\sum_{C \in \mathcal{C}_n} C^\dagger P C \rho C^\dagger P' C = 0. \tag{27}$$

Proof. We will follow the structure of [DCEL09], but simplify for our purposes. Since $\mathbb{P}_n$ is a subgroup of $\mathcal{C}_n$, then we know that the number of left cosets of $\mathbb{P}_n$ in $\mathcal{C}_n$, or the index of $\mathbb{P}_n$ in $\mathcal{C}_n$, is given by:

$$[\mathcal{C}_n : \mathbb{P}_n] = \frac{|\mathcal{C}_n|}{|\mathbb{P}_n|} \tag{28}$$

Then given a representative from each of the cosets, $\{C_1, C_2, \ldots, C_{\frac{|\mathcal{C}_n|}{|\mathbb{P}_n|}}\}$, we can rewrite the sum over all Cliffords as a double sum of the Paulis and the coset representatives as below. We note that it does not matter which representative we choose, only that we have one from each of the cosets, and that the indices on the C terms give which coset they came from:

$$\begin{aligned}\sum_{C \in \mathcal{C}_n} C^\dagger P C \rho C^\dagger P' C &= \sum_{i=1}^{\frac{|\mathcal{C}_n|}{|\mathbb{P}_n|}} \sum_{R \in \mathbb{P}_n} (C_i R)^\dagger P C_i R \rho (C_i R)^\dagger P' C_i R \\ &= \sum_{i=1}^{\frac{|\mathcal{C}_n|}{|\mathbb{P}_n|}} \sum_{R \in \mathbb{P}_n} R^\dagger C_i^\dagger P C_i R \rho R^\dagger C_i^\dagger P' C_i R \end{aligned} \tag{29}$$

Now since $C_i^\dagger P C_i = Q_i$ for some $Q_i \in \mathbb{P}_n$, and since if $P \neq P'$ then $Q_i \neq Q_i'$, we can

simplify our expression to one that only involves Paulis:

$$\begin{aligned} &= \sum_{i=1}^{\frac{|\mathcal{C}_n|}{|\mathbb{P}_n|}} \sum_{R \in \mathbb{P}_n} R^\dagger Q_i R \rho R^\dagger Q_i' R \\ &= \sum_{i=1}^{\frac{|\mathcal{C}_n|}{|\mathbb{P}_n|}} 0 \\ &= 0 \end{aligned} \tag{30}$$

by the Pauli Twirl (Theorem 2.11.1). □

Finally, we note that sampling a uniformly random Clifford can be done efficiently [Got97].

2.13 Quantum Error Correction Theory

Intuitively, a quantum error correcting code is a mapping of k qubits into n qubits, with $n > k$. The original k qubits are the *logical qubits*, or encoded qubits, and the additional $n-k$ qubits allow us to store the k logical qubits in a way that will protect against errors. We know from [Got97] that if a quantum error correcting code can correct errors A and B, then it can correct any linear combination of A and B. For this reason, if a code corrects all weight t Paulis, then it corrects all t-qubit errors.

We will use the notation of an $[[n, 1, d]]$-code to represent a quantum error correcting code that encodes one logical qubit into n qubits and has distance d; if $d = 2t+1$, the code can correct up to t bit or phase flips. We assume that the error correcting code always decodes, even if there are more than t bit or phase flips (in this case, however, the code is not guaranteed to decode to the original input).

Chapter 3

Quantum Message Authentication

This chapter is split into two parts: a discussion and summary of the prior research in quantum message authentication schemes and then the cryptographic definitions required to mathematically represent quantum message authentication schemes and their security.

3.1 Background

Informally, a quantum message authentication (QMA) scheme is a pair of keyed encoding and decoding channels that is used to send a message and check that it has not been altered in transmission. While it is easiest to talk about sending messages from one party to another, in practice this technique would also be useful for verifying the integrity of data that has been stored. Therefore, in addition to sending and receiving, we could look at storing and retrieving. For this reason it is important that this protocol is not an interactive protocol, that is, there should be no communication required between the two parties, classical or quantum, once the protocol has started except for the actual message that is sent.

When we look to prove security of these protocols we will use a comparison of the actual protocol to that of an ideal protocol. The ideal protocol will not analyze the message but instead it will only analyze a reference system through what we call a simulator, which has access to the attack and the reference system, as well

as a polynomial number of additional qubits, as needed. It is important that the simulator, and later the comparison between the ideal and real protocol, are defined for all messages, not for specific messages, since we do not want our security to depend on the message that we are trying to send, only the attack that is being applied. This, of course, means that our simulator can, and in this case will, depend on the attack.

One of the major contributions in this thesis is that we give proofs that follow this ideal/simulator proof structure and that our simulations are *efficient*, so long as the attack is efficient. Here we are taking efficient to mean that the number of quantum gates that need to be applied to implement the given circuit scales at most polynomially with the size of the input register. We will give a more formal definition in Section 3.2. Having efficient simulations is one of the key requirements for *Universally Composable (UC) security*, [Can01, Unr10] which is the notion that any protocol that is UC secure can be composed with any other UC secure protocol and the resulting composition is also secure. Clearly this is a very desirable quality as it greatly reduces the amount of time spent proving security when we are using well-studied subprotocols.

The first QMA scheme was presented in [BCG$^+$02]. In their paper, Barnum, Crépeau, Gottesman, Smith and Tapp define a QMA scheme and show that it can be reduced for a security proof to the case of an interactive protocol that consists of establishing shared EPR pairs and then sending the message using these EPR pairs with a technique called *teleporting*, after using a technique called *purity testing* to verify that the EPR pairs themselves have not been altered. This work also showed that a good authentication scheme must also be a good encryption scheme. This differs significantly from the classical case since classically authentication can be completely separated from encryption [BCG$^+$02].

Further work in the field produced two additional codes with similar flavours. Aharonov, Ben-Or, and Eban presented the Clifford code [ABE10] and Broadbent, Gutoski, and Stebila presented the trap code [BGS13]. The Clifford and trap codes follow the same general format of [BCG$^+$02] by taking a message and then adding extra *trap* qubits. These trap qubits are in a fixed state and are measured during decoding to check for any tampering. In the case of the Clifford code, this is done

by applying a random Clifford to both the message and the traps, and in the case of the trap code, this is done by applying a random Pauli and permutation to both the message and traps. These two codes were proven secure when they were published, but in the case of the Clifford code it was not proven using a simulator definition and therefore finding an efficient simulator was not obvious. In the case of the trap code it was said to be UC secure as a special case of a larger family of protocols that was shown to be UC secure. In this case, however, it was not obvious how the simulator would be defined as the simulator was written for the larger family of protocols, called *Quantum One Time Programs*, and not specifically for the trap code.

Following [ABE10], Dupuis, Nielsen and Salvail [DNS12] provided a more cryptographically rigourous proof for the Clifford code, including the use of a simulator. There was, however, still room for improvement. For one, the simulator that was given was in a form that could not be efficiently constructed. Their simulator relied on the Pauli decomposition of the attack unitary, a square matrix with dimension 2^n, if applied to an n-qubit register. Since the matrix itself scales exponentially with the size of the register, the decomposition of that matrix cannot be efficient, in terms of the size of the register. In addition, their proof relied on the Pauli twirl from [ABE10] which proved the Pauli twirl for the single qubit case but did not offer any details on the n-qubit extension, as well as the proof of the Clifford Twirl from [ABE10] which only held for certain, but not all, cases. Specifically, in [ABE10], when proving the Clifford twirl, they find an index where P and P' disagree, and write that index as $X^a Z^b$ and $X^{a'} Z^{b'}$, and then state that $(a, b) \neq (a', b')$, which is correct. Following that, however, the proof in [ABE10] fails to account for the case where both $a \neq a'$ *and* $b \neq b'$. Finally, the security proof in [DNS12] had minor typos that occasionally made the proof hard to follow, for example switching between n and a for the size of one of the registers, and writing the coefficients in technically correct, but unintuitive or hard to follow ways.

In this thesis, we seek to correct these issues. We offer a proof for the Clifford Twirl lemma that is correct, (Theorem 2.12.2), and we prove the Pauli Twirl for arbitrary dimensions (Theorem 2.11.1). Finally, and most importantly, we offer a proof for the Clifford code that uses many of the same ideas as in [DNS12], however

we present an efficient simulator and we believe it to be explained more fully.

What makes these two codes particularly interesting to study is their structure. Both the Clifford and the trap codes are notable for their ability to allow computation through authentication. Quantum computing on authenticated data (QCAD) allows for gates to be applied through the authentication, without having to decode the data first. When we combine this with the result from [BCG$^+$02] that authentication also encrypts, this means that we can compute on encrypted and authenticated data without having to decrypt our message first. Unfortunately, while these codes are excellent with many different quantum gates, they do not allow for efficient computation with a universal gate set; there is always at least one gate in a universal gate set that will require an exponential complexity to compute through the encryption. However, recent advances [BJ15] have shown that if we can limit the gates that are not efficient, we can still achieve computation through encryption in a fairly satisfying way. For these reasons, it is important that we continue to study these codes and show in our proofs that they have efficient simulators so that we can continue to develop protocols around them that can achieve more than just authentication.

In order to formalize our study of these codes, we will need to define not only what the codes are, but how we can show that they are secure.

3.2 Definitions

In Section 3.1, we talked about needing efficient simulators and for certain aspects of the protocols to be polynomial-time. We will formally define these concepts here.

A family of quantum maps is *polynomial-time* if they can be written as a polynomial-time uniform family of quantum circuits. A quantum state is *polynomial-time generated* if it given as the output of a polynomial-time quantum map (which takes as input the all-zeros state) [Wat11].

At this point, we have all of the tools we need to start talking about the formal definition of quantum message authentication schemes.

Formally, we will follow [DNS12, BCG$^+$02] and define a *quantum message authentication scheme* as a pair of encoding and decoding maps that satisfy the following:

Definition 1 (Quantum message authentication scheme). A *quantum message authentication scheme* is a polynomial-time set of encoding and decoding channels $\{(\mathcal{E}_k^{M\to C}, \mathcal{D}_k^{C\to MF}) \mid k \in \mathcal{K}\}$, where $\mathcal{K}$ is the set of possible keys, M is the input system, C is the encoded system, and F is a flag system that is spanned by two orthogonal states: $|\text{acc}\rangle$ and $|\text{rej}\rangle$, such that for all ρ_M, $(\mathcal{D}_k \circ \mathcal{E}_k)(\rho_M) = \rho_M \otimes |\text{acc}\rangle\langle\text{acc}|$.

Visually, this can be seen as the following circuit:

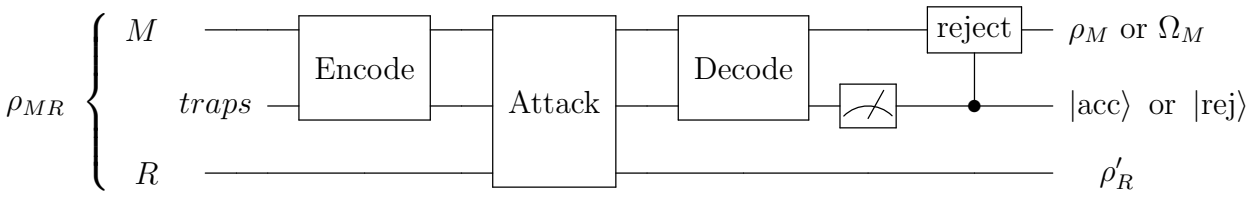

Figure 1: Quantum message authentication general scheme

Here we see that the input is a message system, M, and a reference system, R, in a potentially entangled state (and without any specifics on their dimension), given by ρ_{MR}. To this system we add the traps, then we encode, we allow for an attack, and then we decode and measure the traps. So long as the measurement of the traps indicates accept, we will have our original message register back at the end, and if our measurement of the traps gives reject, then we will replace our M system with a fixed state, Ω_M. We note that the reference system may be altered by the attack, so it is denoted ρ'_R. Finally, the M and R system may still be entangled; they are shown separately simply for ease of reading.

We will closely follow [BW16] for the remainder of this section. We first consider a reference system, R, so that the input can be described as ρ_{MR} and we can furthermore assume that the system consisting of the encoded message, together with the reference system, undergoes a unitary adversarial attack U_{CR}. For a fixed key, k, we thus define the *real-world* channel as:

$$\mathscr{E}_k^{MR\to MRF} : \rho_{MR} \mapsto (\mathcal{D}_k \otimes \mathbb{I}_R)(U_{CR}(\mathcal{E}_k \otimes \mathbb{I}_R)(\rho_{MR})U_{CR}^\dagger), \tag{31}$$

where $\mathbb{I}_R$ is the identity map on the reference system, R. From now on we will not include the identity maps since it will be clear from context which system undergoes

a linear map and which one does not. It is understood that $\mathscr{E}_k$ can take as an input ρ_{MR} but both $\mathcal{E}_k$ and $\mathcal{D}_k$ act only on the M and C systems, respectively.

To "measure" how secure our protocol is, we construct an ideal-world process that applies the protocol ideally given access to a simulator that in turn has access to the attack. We call this the ideal channel or ideal protocol. The ideal protocol consists of the ideal functionality and the simulator. The ideal functionality only acts on the M register and, based on the simulator, either accepts the message or rejects it and replaces the message with a fixed state, Ω_M, but does not interact with the message register in any other way. The simulator has access to both the R register and the attack, given in the form of a circuit. The simulator outputs to the ideal functionality only whether it should accept or reject. The simulator can also alter the R register through this process. Essentially, this can be seen as the adversary only being able to select "accept" or "reject" through the simulator.

Visually, this can be seen through the circuit diagram in Fig. 2.

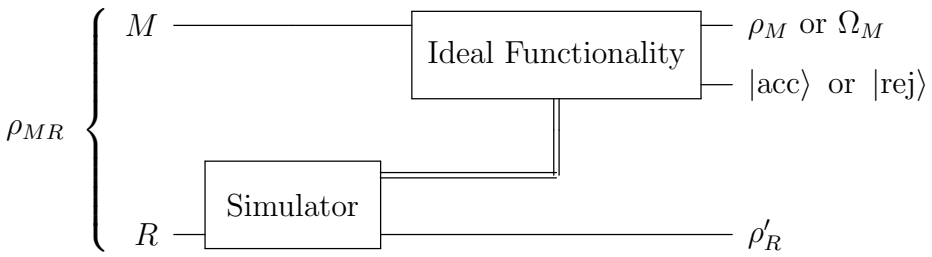

Figure 2: Ideal Channel

When we display quantum circuit diagrams we will be casual with a few of the formalities in order to present a picture that is easy to understand. In these diagrams single lines represent quantum wires (and in this case we assume each wire represents the system it is labeled as, including its dimension), double lines represent classical data, typically a single bit value for either accepting or rejecting, and joint systems (i.e. ρ_{MR}) are shown using a curly bracket and then their subsystems are shown on individual wires. We note that we do not assume that these systems are not entangled (i.e. we do not assume we can write ρ_{MR} as $\rho_M \otimes \rho_R$), but merely use the separation of the wires to indicate which part of the system is being acted on in each case. These diagrams should be read from left to right to understand the order in which

we perform operations.

Precisely, we model the ideal-world process by the quantum channel $\mathscr{F}$, where for each attack, U_{CR}, there exists two CP maps $\mathscr{U}^{acc}$ and $\mathscr{U}^{rej}$ acting only on the reference system R such that $\mathscr{U}^{acc} + \mathscr{U}^{rej} = \mathbb{1}$:

$$\begin{aligned}\mathscr{F}^{MR\to MRF} : \rho_{MR} \to &(\mathbb{1}_M \otimes \mathscr{U}_R^{acc})\rho_{MR} \otimes |\text{acc}\rangle \langle \text{acc}| \\ &+ tr_M((\mathbb{1}_M \otimes \mathscr{U}_R^{rej})\rho_{MR})\Omega_M \otimes |\text{rej}\rangle \langle \text{rej}| . \end{aligned} \tag{32}$$

We then define the protocol to be secure if the real-world protocol is *close* to the ideal-world protocol. We define *close* to be a trace distance between the output of the two channels, acting on the same input, that is exponentially small for all possible inputs.

Definition 2 (Security of quantum message authentication schemes). Let the pair $\{(\mathcal{E}_k^{M\to C}, \mathcal{D}_k^{C\to MF}) \mid k \in \mathcal{K}\}$ be a quantum message authentication scheme, with keys k chosen from $\mathcal{K}$. Then the scheme is ϵ-secure if for all attacks, U_{CR}, there exists an ideal channel, $\mathscr{F}$, such that for all ρ_{MR}:

$$D\left(\frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}} \mathscr{E}_k(\rho_{MR}), \mathscr{F}(\rho_{MR})\right) \leq \epsilon. \tag{33}$$

Furthermore, we require that if $\mathscr{E}_k$ is polynomial-time in the size of the input register, M, then $\mathscr{F}$ is also polynomial-time in the size of the input register, M.

It is important to note that any CPTP map can be written as a unitary on a larger system, and this can be done efficiently [MPZ01]. Since we make no assumptions on the reference system R, we can represent any CPTP map that the adversary applies as a unitary on the message register, and the reference system.

Finally, we note that this definition is similar to the definition in [DNS12], however we require a *polynomial-time simulation*. This requirement does not limit the attacker to a polynomial time attack; we make no computational assumptions about the attack. We only limit the simulator to being no more complex than the attack. As such, if the attack is polynomial-time, then the simulator must be as well.

Chapter 4

QMA Schemes

Here we present two quantum message authentication schemes, the *Clifford* code (Section 4.1) and the *trap* code (Section 4.2). The two encoding procedures are quite similar with a few key differences. Both will take a message and add trap qubits to the message, and then encode the message and traps before sending them. Once recieved, the message and traps will be decoded and then the traps will be measured to check for any tampering. If the traps are all still in their original state the protocol will accept and if any of the traps have been altered the protocol will reject. The biggest difference is that the trap code requires that the input message is first encoded in an error correcting code. The rest of the differences come from what type of encoding is used and how the traps are added. The Clifford code applies a Clifford to the message and traps in the state $|0\rangle\langle 0|$, whereas the trap code applies a permutation and a Pauli to the encoded message along with traps in the state $|0\rangle\langle 0|$ and $|+\rangle\langle +|$. In both cases decoding is simply applying the inverse operations, the inverse permutation, Pauli, and the error decoding, or the inverse Clifford, and finally the traps are measured in their respective bases to check for any tampering.

As explained in [BW16], in the case of the Clifford code only one set of traps is needed because the Clifford twirl breaks any Pauli attack into a uniform mixture of Paulis which is detected on the traps with high probability. The trap code, however, relies on two sets of traps with both a Pauli twirl and a permutation of the message and trap qubits. Furthermore, the trap scheme requires that we first *encode* the input

message into an error correcting code (essentially, this is because the Pauli twirl is not as powerful as the Clifford twirl and will detect only high-weight Pauli attacks with the error correcting code simply correcting the low-weight ones). It may seem as though the Clifford code is preferable to the trap code at this point, since it is more powerful and requires fewer steps and only one type of trap. While these are clear advantages, message authentication is typically only one of the protocols that is likely to be implemented. If, for example, we wanted to perform a computation through authentication then the structure of the Paulis in the trap code offers an advantage to the Clifford code since most gates are easier to perform through Paulis. An example of this can be found in [BGS13] which is where the trap code first appeared.

The description of the two codes comes from [BW16]. The only significant difference is that here we have updated the trap code to allow for a flexible number of traps of each type to be added, as opposed to adding n of each as in [BW16]. The ability to allow a flexible number of trap qubits can be an advantage in a setting where someone is willing to sacrifice some security in order to reduce the number of qubits needed to implement the protocol. This type of resource management can, in theory, go in both directions. While we are currently quite limited in how many qubits we can prepare and control at once, it is also possible that once the technology improves it could be valuable to be able to add more than n qubits and improve the security further than the bound in [BW16].

4.1 The Clifford Code

We define a message authentication scheme by applying a random Clifford to a message and traps as given below. Note that a circuit diagram for this code is given in Fig. 3.

1. The encoding, $\mathcal{E}_k^{M\to C}$, takes as input an n-qubit message in the M system. It appends an additional d-qubit trap register in the state $|0\rangle\langle 0|^{\otimes d}$. A Clifford twirl is then applied to the resulting $n+d$-qubit register, according to the key, k. The output register is called C. While there is not necessarily a standard way

to identify a Clifford by a classical key, one way to do so efficiently is given in [KS14].

Mathematically, the encoding, $\mathcal{E}_k^{M\to C}$, indexed by a secret key, k, on input ρ_M (where C_k the k^{th} Clifford) is given by:

$$\mathcal{E}_k : \rho_M \mapsto C_k(\rho_M \otimes |0\rangle\langle 0|^{\otimes d})C_k^\dagger. \tag{34}$$

2. The decoding, $\mathcal{D}_k^{C\to MF}$, takes the C register and applies the inverse Clifford, according to the key, k. The last d qubits are then measured in the computational basis. If this measurement returns $|0\rangle\langle 0|^{\otimes d}$ then an additional qubit $|\text{acc}\rangle\langle\text{acc}|$ is appended in the flag system, F. If the measurements return anything else then the remaining system, M, is traced out and replaced with a fixed n-qubit state, Ω_M, and an additional qubit, $|\text{rej}\rangle\langle\text{rej}|$, is appended in the flag system.

   Mathematically, the decoding, $\mathcal{D}_k^{C\to MF}$, also indexed by the secret key, k, is given by:

$$\begin{aligned}\mathcal{D}_k : \rho_C \mapsto\ & tr_0(\mathcal{P}_{acc}C_k^\dagger(\rho_C)C_k\mathcal{P}_{acc}^\dagger)\otimes|\text{acc}\rangle\langle\text{acc}| \\ & + tr_{M,0}(\mathcal{P}_{rej}C_k^\dagger(\rho_C)C_k\mathcal{P}_{rej}^\dagger)\Omega_M\otimes|\text{rej}\rangle\langle\text{rej}|\,,\end{aligned}$$

   where $\mathcal{P}_{acc} = \mathbb{1}^{\otimes n}\otimes|0\rangle\langle 0|^{\otimes d}$ and $\mathcal{P}_{rej} = \mathbb{1}^{\otimes(n+d)} - \mathcal{P}_{acc}$ are measurement projectors representing the trap qubits being in their initial states or altered, respectively. Finally, tr_0 refers to the trace over the d trap qubits.

Recalling the general form of a message authentication scheme in Fig. 1, we can therefore give a circuit diagram that describes how the Clifford code would work, given an attack U_{CR}, and an input ρ_{MR}, Fig. 3.

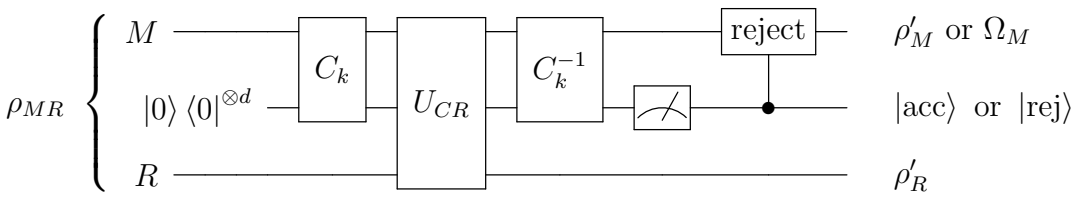

Figure 3: Clifford Code

Note that in Fig. 3 the meter icon represents a measurement and we use ρ'_M to denote the fact that in theory the adversary could (with exponentially small probability) alter the message without altering the traps and therefore we would have an altered message in the output of the protocol.

4.2 The Trap Code

We define a trap code message authentication scheme in a very similar way to [BW16] but here we allow for the number of traps of each kind to be chosen, as opposed to fixed:

1. The encoding, $\mathcal{E}_k^{M\to C}$, takes as input ρ_M and applies an $[[n,1,d]]$-error correcting code to the single-qubit M register, which will correct up to t errors (where $d = 2t+1$). It then appends two additional trap registers, the first in the state $|0\rangle\langle 0|^{\otimes c}$ and the second in the state $|+\rangle\langle +|^{\otimes h}$. The resulting $n+c+h$-qubit register is then permuted and a Pauli encryption is applied, according to the key, k. The resulting register is called C.

   Mathematically the encoding, $\mathcal{E}_k^{M\to C}$, indexed by a two-part secret key $k = (k_1, k_2)$ is given by:

   $$\mathcal{E}_k : \rho_M \mapsto P_{k_2}\pi_{k_1}(Enc_M(\rho_M) \otimes |0\rangle\langle 0|^{\otimes c} \otimes |+\rangle\langle +|^{\otimes h})\pi_{k_1}^\dagger P_{k_2}, \tag{35}$$

   where $Enc_M(\rho_M)$ represents the input state after the error correcting code has been applied to the M system, π_{k_1} is the k_1^{th} permutation and P_{k_2} is the k_2^{th} Pauli matrix.

2. The decoding, $\mathcal{D}_k^{C\to MF}$, takes the C register and applies the inverse Pauli and then the inverse permutation according to the key, k. The last h qubits are then measured in the Hadamard basis and the c qubits before those are measured in the computational basis. If these two measurements return $|+\rangle\langle +|^{\otimes h}$ and $|0\rangle\langle 0|^{\otimes c}$ respectively, then an additional qubit $|\text{acc}\rangle\langle\text{acc}|$ is appended in the flag system F and the resulting M register is decoded (according to the error correcting code applied in the encoding). If the measurements return anything

else, then the remaining system M is traced out and replaced with a fixed single-qubit state Ω_M and an additional qubit, $|\text{rej}\rangle\langle\text{rej}|$, is appended in the flag system.

Define $\mathbb{P}_{\mathscr{E}} = \{P \otimes R \otimes Q | P \in \mathbb{P}_n, R \in \{I, Z\}^{\otimes c}, Q \in \{I, X\}^{\otimes h}\}$. Then define the measurement projector corresponding to the protocol accepting as $\mathcal{P}_{acc} = \mathbb{1}^{\otimes n} \otimes |0\rangle\langle 0|^{\otimes c} \otimes |+\rangle\langle +|^{\otimes h}$. The accepted states are then the states that can be achieved by applying any $P \in \mathbb{P}_{\mathscr{E}}$ to $\rho_M \otimes |0\rangle\langle 0|^{\otimes c} \otimes |+\rangle\langle +|^{\otimes h}$. We define $\mathcal{P}_{rej} = \mathbb{1}^{\otimes n+c+h} - \mathcal{P}_{acc}$, the measurement projector corresponding to the protocol rejecting, where the states achieved by applying any $P \in \mathbb{P}_{n+c+h} \setminus \mathbb{P}_{\mathscr{E}}$ to $Enc_M(\rho_M) \otimes |0\rangle\langle 0|^{\otimes c} \otimes |+\rangle\langle +|^{\otimes h}$ are rejected.

Mathematically, the decoding, $\mathcal{D}_k^{C \to MF}$, also indexed by the two-part secret key, k, is given by:

$$\mathcal{D}_k : \rho_C \mapsto Dec_M tr_{0,+}(\mathcal{P}_{acc}\pi_{k_1}^\dagger P_{k_2}(\rho_C) P_{k_2}\pi_{k_1}\mathcal{P}_{acc}^\dagger) \otimes |\text{acc}\rangle\langle\text{acc}| + tr_{M,0,+}(\mathcal{P}_{rej}\pi_{k_1}^\dagger P_{k_2}(\rho_C) P_{k_2}\pi_{k_1}\mathcal{P}_{acc}^\dagger)\Omega_M \otimes |\text{rej}\rangle\langle\text{rej}|, \quad (36)$$

where Dec_M is the decoding of the error correcting code applied in the encoding and $tr_{0,+}$ refers to the trace over the last two sets of c and h trap qubits.

Recalling the general form for a message authentication scheme given in Fig. 1, we can describe the trap code using a circuit diagram, given an attack U_{CR}, and an input ρ_{MR}, as in Fig. 4. We note that the key differences between the Clifford code in Fig. 3 and the trap code in Fig. 4 are the fact that we use two sets of traps and our encoding and decoding happens in three steps: the error correcting code, the keyed permutation and the keyed Pauli. Otherwise the trap code follows the same form as the Clifford code, and mirrors the form in Fig. 1.

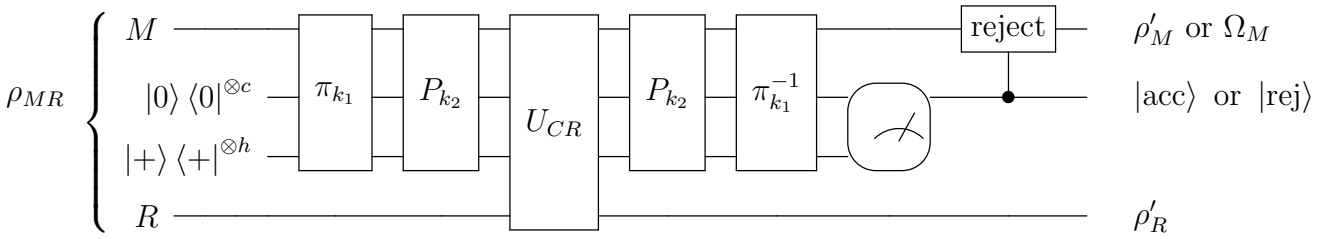

Figure 4: Trap Code

Again, we note that in Fig. 4 the operation given by the meter symbol in a rounded box represents a measurement and we use ρ'_M to denote the fact that in theory the adversary could (with exponentially small probability) alter the message without altering the traps and therefore we would have an altered message in the output of the protocol.

Chapter 5

Security of QMA Schemes

In this section, which follows [BW16] very closely, we present simulation-based proofs for the Clifford (Section 5.1) and the trap (Section 5.2) codes. At a high level the security of the two codes is analyzed in very similar ways (see the discussion in Chapter 3). The main idea (in both cases) is to use a simulator that replaces the encoded message in C with half EPR pairs, without encryption in the Clifford code, and with only a permutation in the trap code; the attack is then applied to these half EPR pairs, as well as any reference system, R. From there we are able to compare the accepted and rejected states between the real world and ideal protocols in order to find the upper bound for the trace distance between them. We will notice that these differences are the cases where the real world protocol accepts something that the simulator rejects. Specifically, this is where an attack gets through and changes a logical qubit but is not detected in the traps. Of course, these same states are not rejected by the real world protocol but they are rejected by the simulator. Because the Clifford twirl maps any non-identity Pauli attack to a uniform mixture of non-identity Paulis, the bound for this distance is simple to compute in the case of the Clifford code. In the case of the trap code a more complicated argument is needed based on permuting the attack and a combinatorial argument that bounds the undetected attacks that can alter the logical data.

5.1 Security of the Clifford Code

As our security model requires, we will first define a simulator (Section 5.1.1) that will act on the reference system, with access to the attack, and then we will use that simulator in our security proof (Section 5.1.2).

5.1.1 Simulator

Recall (Section 3.2) that the simulator interacts with the input by only altering the reference system and selecting either *accept* or *reject*. Given the attack, U_{CR}, to which the simulator has access, the simulator will apply the attack to half EPR pairs in place of the C system and then perform a Bell basis measurement on the EPR pairs. It will select *accept* if the EPR pairs are still in their original state, and *reject* otherwise. When we define the half EPR pairs as $|\Phi^+\rangle\langle\Phi^+|_{C_1C_2}^{\otimes(n+d)}$, and allow the attack to be applied to C_2 instead of C, the simulator can be described in the following circuit, as everything within the dotted lines. Recall that this circuit diagram should be seen as what is in the simulator in Fig. 2.

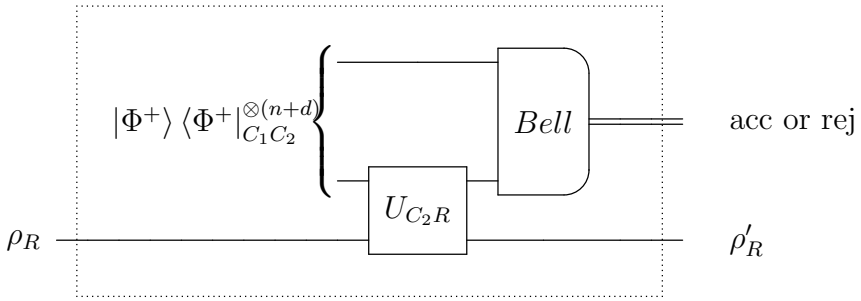

Figure 5: Simulator for Clifford Code

Mathematically, if we let $\mathcal{P}_{acc}^{\mathscr{U}} = \mathbb{1}_{MR} \otimes |\Phi^+\rangle\langle\Phi^+|_{C_1C_2}^{\otimes(n+d)}$ and $\mathcal{P}_{rej}^{\mathscr{U}} = \mathbb{1} - \mathcal{P}_{acc}^{\mathscr{U}}$. The ideal channel runs the simulator and then either acts as the identity on the M register or replaces it with Ω_M depending on whether the simulator outputs accept or reject, respectively. The ideal channel is given by:

$$\mathscr{F}^{MR\to MRF} : \rho_{MR} \to tr_{C_1C_2}(\mathcal{P}_{acc}^{\mathscr{U}} U_{C_2R}(\rho_{MR}\otimes|\Phi^+\rangle\langle\Phi^+|_{C_1C_2}^{\otimes(n+d)})U_{C_2R}^\dagger \mathcal{P}_{acc}^{\mathscr{U}\dagger})\otimes|\text{acc}\rangle\langle\text{acc}|$$
$$+ tr_M(tr_{C_1C_2}(\mathcal{P}_{rej}^{\mathscr{U}} U_{C_2R}(\rho_{MR}\otimes|\Phi^+\rangle\langle\Phi^+|_{C_1C_2}^{\otimes(n+d)})U_{C_2R}^\dagger \mathcal{P}_{rej}^{\mathscr{U}\dagger}))\Omega_M \otimes |\text{rej}\rangle\langle\text{rej}| . \quad (37)$$

According to the above, we define $\mathscr{U}^{acc}$ and $\mathscr{U}^{rej}$ that satisfy Eq. (32) as:

$$\mathscr{U}^{acc} : \rho_{MR} \to tr_{C_1C_2}(\mathcal{P}^{\mathscr{U}}_{acc} U_{C_2R}(\rho_{MR} \otimes |\Phi^+\rangle \langle\Phi^+|^{\otimes(n+d)}_{C_1C_2}) U^\dagger_{C_2R} \mathcal{P}^{\mathscr{U}\dagger}_{acc}), \tag{38}$$

and

$$\mathscr{U}^{rej} : \rho_{MR} \to tr_{C_1C_2}(\mathcal{P}^{\mathscr{U}}_{rej} U_{C_2R}(\rho_{MR} \otimes |\Phi^+\rangle \langle\Phi^+|^{\otimes(n+d)}_{C_1C_2}) U^\dagger_{C_2R} \mathcal{P}^{\mathscr{U}\dagger}_{rej}). \tag{39}$$

For a fixed attack $U_{C_2R} = \sum\limits_{P\in\mathbb{P}_{n+d}} \alpha_P P_{C_2} \otimes U^P_R$, with $\sum\limits_{P\in\mathbb{P}_{n+d}} |\alpha_P|^2 = 1$, we note the effects of $\mathscr{U}^{acc}$ and $\mathscr{U}^{rej}$:

$$\begin{aligned}
\mathscr{U}^{acc}(\rho_{MR}) &= tr_{C_1C_2}(\mathcal{P}^{\mathscr{U}}_{acc} U_{C_2R}(\rho_{MR} \otimes |\Phi^+\rangle \langle\Phi^+|^{\otimes(n+d)}_{C_1C_2}) U^\dagger_{C_2R} \mathcal{P}^{\mathscr{U}\dagger}_{acc}) \\
&= |\alpha_{\mathbb{1}}|^2 (\mathbb{1}_M \otimes U^{\mathbb{1}}_R) \rho_{MR} (\mathbb{1}_M \otimes U^{\mathbb{1}\dagger}_R) \qquad (40) \\
\mathscr{U}^{rej}(\rho_{MR}) &= tr_{C_1C_2}\Big(\mathcal{P}^{\mathscr{U}}_{rej}\Big(\sum_{P\neq\mathbb{1}} |\alpha_P|^2 P_{C_2} \otimes U^P_R\Big)(\rho_{MR} \otimes |\Phi^+\rangle \langle\Phi^+|^{\otimes(n+d)}_{C_1C_2}) \\
&\qquad\qquad \Big(\sum_{P\neq\mathbb{1}} |\alpha_P|^2 P_{C_2} \otimes U^{P\dagger}_R\Big) \mathcal{P}^{\mathscr{U}\dagger}_{rej}\Big) \\
&= \sum_{P\neq\mathbb{1}} |\alpha_P|^2 (\mathbb{1}_M \otimes U^P_R)(\rho_{MR})(\mathbb{1}_M \otimes U^{P\dagger}_R). \qquad (41)
\end{aligned}$$

We are now ready to state and prove our main theorem on the security of the Clifford message authentication scheme.

5.1.2 Security

Theorem 5.1.1. *Let $\{(\mathcal{E}^{M\to C}_k, \mathcal{D}^{C\to MF}_k) \mid k \in \mathcal{K}\}$ be the Clifford quantum message authentication scheme, with parameter d. Then the Clifford code is an ϵ-secure quantum authentication scheme, for $\epsilon = \frac{3}{2^d}$.*

Proof. We will follow the proof structure used in [DNS12, ABE10]. Using the simulator described above, we wish to show that:

$$D\left(\frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}} \mathscr{E}_k(\rho_{MR}), \mathscr{F}(\rho_{MR})\right) \leq \epsilon, \forall \rho_{MR}. \tag{42}$$

Consider a general attack U_{CR}, written as $U_{CR} = \sum_{P\in\mathbb{P}_{n+d}} \alpha_P P_C \otimes U_R^P$ where $\sum_{P\in\mathbb{P}_{n+d}} |\alpha_P|^2 = 1$. The real-world channel is then represented as:

$$\mathscr{E}_k^{MR\to MRF} : \rho_{MR} \mapsto \mathcal{D}_k\Big(\Big(\sum_{P\in\mathbb{P}_{n+d}} \alpha_P P_C\otimes U_R^P\Big)\mathcal{E}_k(\rho_{MR})\Big(\sum_{P\in\mathbb{P}_{n+d}} \overline{\alpha_P} P_C\otimes U_R^{P\dagger}\Big)\Big). \quad (43)$$

We will use $\psi = \rho_{MR}\otimes|0\rangle\langle 0|^{\otimes d}$ to simplify the following expressions. Consider the effect of the real protocol on input ρ_{MR} with attack $\sum_{P\in\mathbb{P}_{n+d}} \alpha_P P_C \otimes U_R^P$, conditioned on acceptance:

$$\frac{1}{|\mathcal{K}|}\sum_{k\epsilon\mathcal{K}} tr_0\Big(\mathcal{P}_{acc}C_k^\dagger\Big(\sum_{P\in\mathbb{P}_{n+d}} \alpha_P P_C\otimes U_R^P\Big)(C_k\psi C_k^\dagger) \Big(\sum_{P\in\mathbb{P}_{n+d}} \overline{\alpha_P} P_C^\dagger\otimes U_R^{P\dagger}\Big)C_k\mathcal{P}_{acc}^\dagger\Big)\otimes|\mathrm{acc}\rangle\langle\mathrm{acc}|. \quad (44)$$

Now we can apply the Clifford Twirl (Theorem 2.12.2), since the sum over all keys is, of course, the sum over all Cliffords (since the keys index all $n+d$-qubit Cliffords) and then break up the expression into the identity Pauli from the attack, and all other Paulis. What we are left with is:

$$\begin{aligned}
&\frac{1}{|\mathcal{K}|}\sum_{k\epsilon\mathcal{K}} tr_0\Big(\sum_{P\in\mathbb{P}_{n+d}} |\alpha_P|^2\,\mathcal{P}_{acc}C_k^\dagger(P_C\otimes U_R^P)(C_k\psi C_k^\dagger)(P_C^\dagger\otimes U_R^{P\dagger})C_k\mathcal{P}_{acc}^\dagger\Big)\otimes|\mathrm{acc}\rangle\langle\mathrm{acc}| \\
&= \frac{1}{|\mathcal{K}|}\sum_{k\epsilon\mathcal{K}} tr_0\Big(|\alpha_{\mathbb{1}}|^2\,\mathcal{P}_{acc}C_k^\dagger(\mathbb{1}_C\otimes U_R^{\mathbb{1}})(C_k\psi C_k^\dagger)(\mathbb{1}_C\otimes U_R^{\mathbb{1}\dagger})C_k\mathcal{P}_{acc}^\dagger\Big)\otimes|\mathrm{acc}\rangle\langle\mathrm{acc}| \\
&+ \frac{1}{|\mathcal{K}|}\sum_{k\epsilon\mathcal{K}} tr_0\Big(\sum_{P\neq\mathbb{1}} |\alpha_P|^2\,\mathcal{P}_{acc}C_k^\dagger(P_C\otimes U_R^P)(C_k\psi C_k^\dagger)(P_C^\dagger\otimes U_R^{P\dagger})C_k\mathcal{P}_{acc}^\dagger\Big)\otimes|\mathrm{acc}\rangle\langle\mathrm{acc}|.
\end{aligned} \quad (45)$$

Clearly the first term is exactly what the simulator will accept, and the second term is in exactly the right form to use a Clifford Randomization (Theorem 2.12.1), resulting in:

$$\begin{aligned}
&= \mathscr{U}^{acc}(\rho_{MR})\otimes|\mathrm{acc}\rangle\langle\mathrm{acc}| \\
&+ \frac{1}{|\mathcal{C}_{n+d}|}tr_0\Big(\sum_{\tilde{P}\neq\mathbb{1}}\sum_{P\neq\mathbb{1}} |\alpha_P|^2\,\frac{|\mathcal{C}_{n+d}|}{|\mathbb{P}_{n+d}|-1}\mathcal{P}_{acc}(\tilde{P}_C\otimes U_R^P)\psi(\tilde{P}_C^\dagger\otimes U_R^{P\dagger})\mathcal{P}_{acc}^\dagger\Big)\otimes|\mathrm{acc}\rangle\langle\mathrm{acc}|.
\end{aligned} \quad (46)$$

The $\tilde{P}$s are the results of the Clifford Randomization applied to a Pauli, P. The randomization is not applied to the reference system, so the U_R^P terms are not changed by the randomization. We can use the properties of the trace to move the trace inside the first sum, and we can move the $\frac{|\mathcal{C}_{n+d}|}{|\mathbb{P}_{n+d}|-1}$ coefficient out of both of the sums:

$$\begin{aligned}&= \mathscr{U}^{acc}(\rho_{MR}) \otimes |\text{acc}\rangle \langle \text{acc}| \\ &+ \frac{1}{|\mathcal{C}_{n+d}|} \frac{|\mathcal{C}_{n+d}|}{|\mathbb{P}_{n+d}|-1} \Big(\sum_{\tilde{P} \neq \mathbb{1}} tr_0 \sum_{P \neq \mathbb{1}} |\alpha_P|^2 \, \mathcal{P}_{acc}(\tilde{P}_C \otimes U_R^P) \psi (\tilde{P}_C^\dagger \otimes U_R^{P\dagger}) \mathcal{P}_{acc}^\dagger \Big) \otimes |\text{acc}\rangle \langle \text{acc}| \, .\end{aligned} \tag{47}$$

We recognize the R register in the second sum as the states that the simulator will reject. Recall that the simulator is in terms of the sum over all non-identity Paulis and includes the α_P coefficients. We can therefore write the previous line in terms of the simulator as:

$$\begin{aligned}&= \mathscr{U}^{acc}(\rho_{MR}) \otimes |\text{acc}\rangle \langle \text{acc}| \\ &+ \frac{1}{|\mathbb{P}_{n+d}|-1} \Big(\sum_{\tilde{P} \neq \mathbb{1}} tr_0 \mathcal{P}_{acc}(\tilde{P}_C(\mathscr{U}^{rej}(\rho_{MR}) \otimes |0\rangle \langle 0|^{\otimes d}) \tilde{P}_C^\dagger) \mathcal{P}_{acc}^\dagger \Big) \otimes |\text{acc}\rangle \langle \text{acc}| \, .\end{aligned} \tag{48}$$

If we let $\mathbb{P}_t$ be the set of all Paulis that do not alter the trap qubits, then when we apply $\mathcal{P}_{acc}$ to the above, we end up with the sum over the $\tilde{P} \in \mathbb{P}_t \setminus \{\mathbb{1}\}$. Therefore the previous line can be simplified to:

$$\begin{aligned}&= \mathscr{U}^{acc}(\rho_{MR}) \otimes |\text{acc}\rangle \langle \text{acc}| \\ &\quad + \frac{1}{|\mathbb{P}_{n+d}|-1} \sum_{\tilde{P} \in \mathbb{P}_t \setminus \{\mathbb{1}\}} tr_0(\tilde{P}_C(\mathscr{U}^{rej}(\rho_{MR}) \otimes |0\rangle \langle 0|^{\otimes d}) \tilde{P}_C^\dagger) \otimes |\text{acc}\rangle \langle \text{acc}| \, .\end{aligned} \tag{49}$$

The effect of the real protocol on input ρ_{MR} with attack $\sum_{P \in \mathbb{P}_{n+d}} \alpha_P P_C \otimes U_R^P$,

conditioned on rejection, can be manipulated in the same way to arrive at:

$$\begin{aligned}
&\frac{1}{|\mathcal{K}|}\sum_{k\epsilon\mathcal{K}}\Big(tr_{M,0}\Big(\mathcal{P}_{rej}C_k^\dagger\Big(\sum_{P\in\mathbb{P}_{n+d}}\alpha_P P_C\otimes U_R^P\Big)(C_k(\psi)C_k^\dagger)\\
&\qquad\qquad\Big(\sum_{P\in\mathbb{P}_{n+d}}\overline{\alpha_P}P_C^\dagger\otimes U_R^{P\dagger}\Big)C_k\mathcal{P}_{rej}^\dagger\Big)\Big)\Omega_M\otimes|\text{rej}\rangle\langle\text{rej}|\\
&=\frac{1}{|\mathcal{K}|}\sum_{k\epsilon\mathcal{K}}\Big(tr_{M,0}(|\alpha_{\mathbb{1}}|^2\,\mathcal{P}_{rej}C_k^\dagger(\mathbb{1}_C\otimes U_R^{\mathbb{1}})(C_k(\psi)C_k^\dagger)(\mathbb{1}_C\otimes U_R^{\mathbb{1}\dagger})C_k\mathcal{P}_{rej}^\dagger)\Big)\\
&\qquad\qquad\Omega_M\otimes|\text{rej}\rangle\langle\text{rej}|\\
&\quad+\frac{1}{|\mathcal{K}|}\sum_{k\epsilon\mathcal{K}}\Big(tr_{M,0}\Big(\sum_{P\neq\mathbb{1}}|\alpha_P|^2\,\mathcal{P}_{rej}C_k^\dagger(P_C\otimes U_R^P)(C_k(\psi)C_k^\dagger)(P_C^\dagger\otimes U_R^{P\dagger})C_k\mathcal{P}_{rej}^\dagger\Big)\Big)\\
&\qquad\qquad\Omega_M\otimes|\text{rej}\rangle\langle\text{rej}|\\
&=\frac{1}{|\mathbb{P}_{n+d}|-1}\sum_{\tilde{P}\neq\mathbb{1}}\sum_{P\neq\mathbb{1}}|\alpha|^2\Big(tr_{M,0}(\mathcal{P}_{acc}(\tilde{P}_C\otimes U_R^P)(\psi)(\tilde{P}_C^\dagger\otimes U_R^{P\dagger})\mathcal{P}_{acc}^\dagger)\Big)\\
&\qquad\qquad\Omega_M\otimes|\text{rej}\rangle\langle\text{rej}|\\
&=tr_M(\mathscr{U}^{rej}(\rho_{MR}))\Omega_M\otimes|\text{rej}\rangle\langle\text{rej}|\\
&\qquad\qquad-\frac{1}{|\mathbb{P}_{n+d}|-1}tr_M\Big(\sum_{P\in\mathbb{P}_t\setminus\{\mathbb{1}\}}\mathscr{U}^{rej}(\rho_{MR})\Big)\Omega_M\otimes|\text{rej}\rangle\langle\text{rej}|\\
&=tr_M(\mathscr{U}^{rej}(\rho_{MR}))\Omega_M\otimes|\text{rej}\rangle\langle\text{rej}|\\
&\qquad\qquad-\frac{4^n2^d-1}{|\mathbb{P}_{n+d}|-1}tr_M(\mathscr{U}^{rej}(\rho_{MR}))\Omega_M\otimes|\text{rej}\rangle\langle\text{rej}|\,. \qquad (50)
\end{aligned}$$

When we combine the accepted states and the rejected states into the real world protocol given by Eq. (43), we can write it in terms of the simulator as:

$$\begin{aligned}
&\mathcal{D}_k(U_{CR}\mathcal{E}_k(\rho_{MR})U_{CR}^\dagger)\\
&=\mathscr{U}^{acc}(\rho_{MR})\otimes|\text{acc}\rangle\langle\text{acc}|\\
&\quad+\frac{1}{|\mathbb{P}_{n+d}|-1}\sum_{\tilde{P}\in\mathbb{P}_t\setminus\{\mathbb{1}\}}tr_0(\tilde{P}_C(\mathscr{U}^{rej}(\rho_{MR})\otimes|0\rangle\langle 0|^{\otimes d})\tilde{P}_C^\dagger)\otimes|\text{acc}\rangle\langle\text{acc}|\\
&\quad+tr_M(\mathscr{U}^{rej}(\rho_{MR}))\Omega_M\otimes|\text{rej}\rangle\langle\text{rej}|\\
&\quad-\frac{4^n2^d-1}{|\mathbb{P}_{n+d}|-1}tr_M(\mathscr{U}^{rej}(\rho_{MR}))\Omega_M\otimes|\text{rej}\rangle\langle\text{rej}|\,. \qquad (51)
\end{aligned}$$

We can therefore write Eq. (42) as:

$$\begin{aligned}
&\frac{1}{2}\Big\| \mathscr{U}^{acc}(\rho_{MR}) \otimes |\text{acc}\rangle\langle\text{acc}| \\
&\qquad + \frac{1}{|\mathbb{P}_{n+d}|-1} \sum_{\tilde{P}\in\mathbb{P}_t\setminus\{\mathbb{1}\}} tr_0(\tilde{P}_C(\mathscr{U}^{rej}(\rho_{MR})\otimes|0\rangle\langle 0|^{\otimes d})\tilde{P}_C^\dagger)\otimes|\text{acc}\rangle\langle\text{acc}| \\
&+ tr_M(\mathscr{U}^{rej}(\rho_{MR}))\Omega_M\otimes|\text{rej}\rangle\langle\text{rej}| - \frac{4^n2^d-1}{|\mathbb{P}_{n+d}|-1} tr_M(\mathscr{U}^{rej}(\rho_{MR}))\Omega_M\otimes|\text{rej}\rangle\langle\text{rej}| \\
&\qquad - (\mathscr{U}^{acc}(\rho_{MR})\otimes|\text{acc}\rangle\langle\text{acc}| + tr_M(\mathscr{U}^{rej}(\rho_{MR}))\Omega_M\otimes|\text{rej}\rangle\langle\text{rej}|)\Big\|_1 \\
&= \frac{1}{2}\Big\| \frac{1}{|\mathbb{P}_{n+d}|-1} \sum_{\tilde{P}\in\mathbb{P}_t\setminus\{\mathbb{1}\}} tr_0(\tilde{P}_C(\mathscr{U}^{rej}(\rho_{MR})\otimes|0\rangle\langle 0|^{\otimes d})\tilde{P}_C^\dagger)\otimes|\text{acc}\rangle\langle\text{acc}| \\
&\qquad - \frac{4^n2^d-1}{|\mathbb{P}_{n+d}|-1} tr_M(\mathscr{U}^{rej}(\rho_{MR}))\Omega_M\otimes|\text{rej}\rangle\langle\text{rej}| \Big\|_1 \qquad (52)
\end{aligned}$$

Since $|\mathbb{P}_t \setminus \{\mathbb{1}\}| = 4^n2^d - 1$, and the maximum trace distance between two states is 1, we can see that by the triangle inequality, the above is bounded by:

$$\begin{aligned}
&\leq \Big(\frac{1}{2}\Big)\frac{4^n2^d-1}{|\mathbb{P}_{n+d}|-1} \\
&= \Big(\frac{1}{2}\Big)\frac{4^n2^d-1}{4^{n+d}-1} \\
&= \Big(\frac{1}{2}\Big)\frac{1-\frac{1}{4^n2^d}}{2^d-\frac{1}{4^n2^d}} \\
&\leq 3\times\frac{1}{2^d}. \qquad (53)
\end{aligned}$$

This concludes the proof, showing that the Clifford code is $\frac{3}{2^d}$-secure. □

While the bound of $\frac{3}{2^d}$ is not tight, it is identical to the bound of $\frac{6}{2^d}$ achieved in [DNS12] when we consider that we use the trace distance in our definition of security, and [DNS12] uses the trace norm, which differs from the trace distance by a factor of 2.

5.2 Security of the Trap Code

The analysis of the security of the trap code proceeds in a very similar manner to that of the Clifford code. The simulator is structured in the same way, but we allow

for some potential non-identity attacks to get through the traps. In addition, while we use the same structure as [BW16], the protocol, and therefore the proof, has been changed to allow for a flexible number of traps of each kind that are added. In [BW16] the number of traps that are added are fixed at n of each kind.

5.2.1 Simulator

Recall (Section 3.2) that the simulator interacts with the input by only altering the reference system and selecting either *accept* or *reject*. Given the attack, U_{CR}, to which the simulator has access, the simulator applies the attack to randomly permuted half EPR pairs in place of the C system and then de-permutes the EPR pairs and performs a Bell basis measurement. It selects *accept* if the first n of the EPR pairs have $\leq t$ errors, the next c of the EPR pairs are either unchanged or have phase flip errors, and the last h of the EPR pairs are either unchanged or have bit flip errors. It selects *reject* otherwise. The function of the simulator can be seen in the circuit diagram in Fig. 6, where the simulator is everything in the dotted lines. Note that this diagram should be interpreted as explaining the functionality of the simulator in Fig. 2.

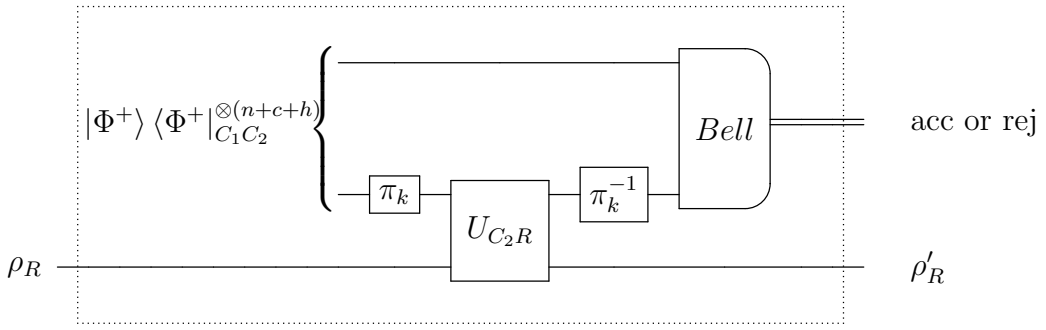

Figure 6: Simulator for Trap Code

Mathematically, we let $\mathbb{P}_{\mathscr{F}} = \{P \otimes R \otimes Q | P \in \mathbb{P}_n, \omega(P) \leq t, R \in \{I, Z\}^{\otimes c}, Q \in \{I, X\}^{\otimes h}\}$. Specifically, $\mathbb{P}_{\mathscr{F}}$ is the set of all Paulis that the ideal protocol will accept being applied to the half EPR pairs—Paulis that would apply at most t non-identity Paulis on the message space and would not alter the $|0\rangle\langle 0|^{\otimes c}$ or the $|+\rangle\langle +|^{\otimes h}$ traps in the real world protocol. Finally, define the measurement projector corresponding

to the simulator selecting *accept* as:

$$\mathcal{P}^{\mathscr{U}}_{acc} = \sum_{P\in\mathbb{P}_{\mathscr{F}}} \mathbb{1}_{MR} \otimes (P_{C_2} \left|\Phi^+\right\rangle \left\langle\Phi^+\right|^{\otimes(n+c+h)}_{C_1C_2} P^\dagger_{C_2}), \tag{54}$$

and the measurement projector corresponding to the simulator selecting *reject* as:

$$\mathcal{P}^{\mathscr{U}}_{rej} = \mathbb{1} - \mathcal{P}^{\mathscr{U}}_{acc}. \tag{55}$$

The ideal functionality with attack U_{C_2R} is therefore:

$$\begin{aligned}
&\mathscr{F}^{MR\to MRF} : \rho_{MR} \to \\
&\quad \frac{1}{\left|\Pi_{(n+c+h)}\right|} tr_{C_1C_2}\Big(\sum_{\pi\in\Pi_{(n+c+h)}} \Big(\mathcal{P}^{\mathscr{U}}_{acc}\pi^\dagger_{C_2} U_{C_2R}\pi_{C_2}(\rho_{MR}\otimes\left|\Phi^+\right\rangle\left\langle\Phi^+\right|^{\otimes(n+c+h)}_{C_1C_2}) \\
&\qquad\qquad \pi^\dagger_{C_2}U^\dagger_{C_2R}\pi_{C_2}\mathcal{P}^{\mathscr{U}\dagger}_{acc}\Big)\Big) \otimes \left|\text{acc}\right\rangle\left\langle\text{acc}\right| \\
&\quad + tr_M\Big(\frac{1}{\left|\Pi_{(n+c+h)}\right|} tr_{C_1C_2}\Big(\sum_{\pi\in\Pi_{(n+c+h)}} \Big(\mathcal{P}^{\mathscr{U}}_{rej}\pi^\dagger_{C_2} U_{C_2R}\pi_{C_2}(\rho_{MR}\otimes\left|\Phi^+\right\rangle\left\langle\Phi^+\right|^{\otimes(n+c+h)}_{C_1C_2}) \\
&\qquad\qquad \pi^\dagger_{C_2}U^\dagger_{C_2R}\pi_{C_2}\mathcal{P}^{\mathscr{U}\dagger}_{rej}\Big)\Big)\Big)\Omega_M \otimes \left|\text{rej}\right\rangle\left\langle\text{rej}\right|.
\end{aligned} \tag{56}$$

For a fixed attack $U_{C_2R} = \sum_{P\in\mathbb{P}_{(n+c+h)}} \alpha_P P_{C_2}\otimes U^P_R$, with $\sum_{P\in\mathbb{P}_{(n+c+h)}} \left|\alpha_P\right|^2 = 1$ and where for the sake of brevity we will represent $\rho_{MR}\otimes\left|\Phi^+\right\rangle\left\langle\Phi^+\right|^{\otimes(n+c+h)}_{C_1C_2}$ with $\phi_{MRC_1C_2}$, the ideal functionality becomes:

$$\begin{aligned}
&\mathscr{F}^{MR\to MRF} : \rho_{MR} \to \\
&\quad \frac{1}{\left|\Pi_{(n+c+h)}\right|} tr_{C_1C_2}\Bigg(\sum_{\pi\in\Pi_{(n+c+h)}} \mathcal{P}^{\mathscr{U}}_{acc}\pi^\dagger_{C_2}\Big(\sum_{P\in\mathbb{P}_{(n+c+h)}} \alpha_P P_{C_2}\otimes U^P_R\Big)\pi_{C_2}\phi_{MRC_1C_2}\pi^\dagger_{C_2} \\
&\qquad\qquad \Big(\sum_{P\in\mathbb{P}_{(n+c+h)}} \overline{\alpha_P} P_{C_2}\otimes U^{P\dagger}_R\Big)\pi_{C_2}\mathcal{P}^{\mathscr{U}\dagger}_{acc} \otimes \left|\text{acc}\right\rangle\left\langle\text{acc}\right| \\
&\quad + tr_M\Big(\mathcal{P}^{\mathscr{U}}_{rej}\pi^\dagger_{C_2}\Big(\sum_{P\in\mathbb{P}_{(n+c+h)}} \alpha_P P_{C_2}\otimes U^P_R\Big)\pi_{C_2}\phi_{MRC_1C_2}\pi^\dagger_{C_2} \\
&\qquad\qquad \Big(\sum_{P\in\mathbb{P}_{(n+c+h)}} \overline{\alpha_P} P_{C_2}\otimes U^{P\dagger}_R\Big)\pi_{C_2}\mathcal{P}^{\mathscr{U}\dagger}_{rej}\Big)\Omega_M \otimes \left|\text{rej}\right\rangle\left\langle\text{rej}\right| \Bigg).
\end{aligned} \tag{57}$$

From here we will move the permutations to act on the attack Paulis since they're all applied to the same register, C_2:

$$= \frac{1}{\left|\Pi_{(n+c+h)}\right|} tr_{C_1C_2}\Bigg(\sum_{\pi\in\Pi_{(n+c+h)}} \Big(\mathcal{P}^{\mathscr{U}}_{acc}\Big(\sum_{P\in\mathbb{P}_{(n+c+h)}} \alpha_P \pi^\dagger_{C_2} P_{C_2}\pi_{C_2} \otimes U_R^P\Big)\phi_{MRC_1C_2} \Big(\sum_{P\in\mathbb{P}_{(n+c+h)}} \overline{\alpha_P}\pi^\dagger_{C_2} P_{C_2}\pi_{C_2} \otimes U_R^{P\dagger}\Big)\mathcal{P}^{\mathscr{U}\dagger}_{acc}\Big) \otimes |\text{acc}\rangle\langle\text{acc}| + tr_M\Big(\mathcal{P}^{\mathscr{U}}_{rej}\Big(\sum_{P\in\mathbb{P}_{(n+c+h)}} \alpha_P \pi^\dagger_{C_2} P_{C_2}\pi_{C_2} \otimes U_R^P\Big)\phi_{MRC_1C_2} \Big(\sum_{P\in\mathbb{P}_{(n+c+h)}} \overline{\alpha_P}\pi^\dagger_{C_2} P_{C_2}\pi_{C_2} \otimes U_R^{P\dagger}\Big)\mathcal{P}^{\mathscr{U}\dagger}_{rej}\Big)\Omega_M \otimes |\text{rej}\rangle\langle\text{rej}| \Bigg). \tag{58}$$

Finally, we apply the projectors:

$$= \frac{1}{\left|\Pi_{(n+c+h)}\right|} tr_{C_1C_2}\Bigg(\sum_{\pi\in\Pi_{(n+c+h)}} \Big(\sum_{P|\pi^\dagger P\pi\in\mathbb{P}_{\mathscr{F}}} |\alpha_P|^2 (\pi^\dagger_{C_2} P_{C_2}\pi_{C_2} \otimes U_R^P)(\phi_{MRC_1C_2}) (\pi^\dagger_{C_2} P_{C_2}\pi_{C_2} \otimes U_R^{P\dagger})\Big) \otimes |\text{acc}\rangle\langle\text{acc}| + tr_M\Big(\sum_{P|\pi^\dagger P\pi\notin\mathbb{P}_{\mathscr{F}}} |\alpha_P|^2 (\pi^\dagger_{C_2} P_{C_2}\pi_{C_2} \otimes U_R^P)(\phi_{MRC_1C_2}) (\pi^\dagger_{C_2} P_{C_2}\pi_{C_2} \otimes U_R^{P\dagger})\Big)\Omega_M \otimes |\text{rej}\rangle\langle\text{rej}| \Bigg). \tag{59}$$

Lastly, we present a lemma that will allow us to consider the maximum number of permutations that can act on an attack and have the permuted attack be undetected on the traps.

Lemma 5.2.1. *For a fixed $P \in \mathbb{P}_{(n+c+h)}$, let η_P denote the number of permutations π of P such that $\pi^\dagger P\pi \in \mathbb{P}_{\mathscr{E}} \setminus \mathbb{P}_{\mathscr{F}}$. Then for all P:*

$$\eta_P \leq \binom{n}{t+1}(t+1)!((n+c+h)-(t+1))!\,. \tag{60}$$

An intuitive argument for the above lemma is that η_P can be upper-bounded by fixing a Pauli $P \in \{I,X\}^{(n+c+h)}$ of weight $t+1$. We show that a Pauli with greater weight will have $\leq \eta_P$ possible allowed permutations. To find the number of possible

allowed permutations we will consider the first n positions, where we require at least $t+1$ non-identity Paulis (for a total of $\binom{n}{t+1}(t+1)!$ permutations). The remaining positions are then simply permuted, since we have used all of the non-identity Paulis already, contributing a multiplicative factor of $((n+c+h)-(t+1))!$ permutations. This is formalized below (where we also consider general attack Paulis consisting of combinations of X, Y and Z).

Proof. In order to find an upper bound for η_P, we look to find the Pauli, P, that has the largest number of permutations, π, such that $\pi^\dagger P \pi \in \mathbb{P}_{\mathscr{E}} \setminus \mathbb{P}_{\mathscr{F}}$.

For a Pauli P with $\omega(P) = d$, we write $d = d_x + d_y + d_z + x_1 + y + z_1 + x_2 + z_2$ for values $d_x, d_y, d_z, x_1, y, z_1, x_2, z_2$ as follows:

1. d_x, d_y, d_z where $d_x + d_y + d_z = t+1$. These are the $t+1$ X, Y, and Z Paulis that must be applied to the first n qubits for the Pauli to be in $\mathbb{P}_{\mathscr{E}} \setminus \mathbb{P}_{\mathscr{F}}$.

2. y where $y + d_y$ is the total number of Y Paulis in P and y are the additional Y Paulis applied to the first n qubits. Note that Y Paulis cannot be applied to either set of traps without altering them.

3. x_1, x_2 where $x_1 + x_2 + d_x$ is the total number of X Paulis in P and x_1 are the additional X Paulis applied to the first n qubits and x_2 are the X Paulis applied to the $|+\rangle\langle+|^{\otimes h}$ traps.

4. z_1, z_2 where $z_1 + z_2 + d_z$ is the total number of Z Paulis in P and z_1 are the additional Z Paulis applied to the first n qubits and z_2 are the Z Paulis applied to the $|0\rangle\langle 0|^{\otimes c}$ traps.

Then the possible permutations on P are found by multiplying the following terms:

1. $\binom{n}{d_x, d_y, d_z, n-t-1} d_x! d_y! d_z!$, which is the number of ways to choose the required $t+1$ spots for the minimum number of Paulis applied to the first n qubits, multiplied by the number of ways of permuting each of the sets of X, Y, and Z Paulis. Note that this term simplifies to $\frac{n!}{(n-t-1)!}$,

2. $\binom{n-t-1}{x_1} x_1!$, the number of ways to apply x_1 additional X Paulis to the first n qubits,

3. $\binom{n-t-1-x_1}{y} y!$, the number of ways to apply y additional Y Paulis to the first n qubits,

4. $\binom{n-t-1-x_1-y}{z_1} z_1!$, the number of ways to apply z_1 additional Z Paulis to the first n qubits,

5. $\binom{h}{x_2} x_2!$, the number of ways to apply x_2 X Paulis to the h traps that will not be changed by them,

6. $\binom{c}{z_2} z_2!$, the number of ways to apply z_2 Z Paulis to the c traps that will not be changed by them, and

7. $((n+c+h)-(d_x+d_y+d_z+x_1+y+z_1+x_2+z_2))!$ the number of ways to permute the remaining identity qubits, which simplifies to $((n+c+h)-d)!$.

The product of these terms, once simplified, is:

$$\begin{aligned}\eta_P &= \frac{n!c!h!((n+c+h)-d)!}{(n-t-1-x_1-y-z_1)!(h-x_2)!(c-z_2)!} \\ &= \prod_{n-t-x_1-y-z_1}^{n} i \prod_{h-x_2+1}^{h} i \prod_{c-z_2+1}^{c} i \prod_{i=1}^{(n+c+h)-t-1-x_1-y-z_1-x_2-z_2} i \end{aligned} \tag{61}$$

Since t is fixed, in order to maximize the above expression, we need to minimize x_1, y, z_1, x_2, z_2. This is achieved by setting $x_1 = y = z_1 = x_2 = z_2 = 0$, and therefore $d = t+1$. We thus find that:

$$\begin{aligned}\eta_P &\leq \prod_{n-t}^{n} i \prod_{i=1}^{(n+c+h)-t-1} i \\ &= \binom{n}{t+1} (t+1)!((n+c+h)-(t+1))!. \end{aligned} \tag{62}$$

□

5.2.2 Security

We are now ready to present our main theorem:

Theorem 5.2.2. *Let $\{(\mathcal{E}_k^{M\to C}, \mathcal{D}_k^{C\to MF}) \mid k \in \mathcal{K}\}$ be the trap quantum message authentication scheme with parameter t, the number of bit or phase flip errors that the error correcting code applied to the input message qubit can correct. Then the trap code is an ϵ-secure quantum message authentication scheme, for $\epsilon = (\frac{1}{1+\frac{c}{n}+\frac{h}{n}-\frac{t}{n}})^{t+1}$.*

Proof. Using the simulator described above, we wish to show that:

$$D\left(\frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}}\mathscr{E}_k(\rho_{MR}), \mathscr{F}(\rho_{MR})\right) \leq \epsilon, \forall \rho_{MR}. \tag{63}$$

Consider a general attack U_{CR}, written as $U_{CR} = \sum_{P\in\mathbb{P}_{(n+c+h)}} \alpha_P P_C \otimes U_R^P$ with $\sum_{P\in\mathbb{P}_{(n+c+h)}} |\alpha_P|^2 = 1$. We will use ψ to represent $Enc_M(\rho_{MR}) \otimes |0\rangle\langle 0|^{\otimes c} \otimes |+\rangle\langle +|^{\otimes h}$. The real-world channel is then represented as:

$$\mathscr{E}_k^{MR\to MRF} : \rho_{MR} \mapsto \mathcal{D}_k\Big(\Big(\sum_{P\in\mathbb{P}_{(n+c+h)}} \alpha_P P_C \otimes U_R^P\Big)\mathcal{E}_k(\rho_{MR})\Big(\sum_{P\in\mathbb{P}_{(n+c+h)}} \overline{\alpha_P} P_C \otimes U_R^{P\dagger}\Big)\Big) \tag{64}$$

$$\begin{aligned} = \frac{1}{|\mathcal{K}|} tr_{0,+} \sum_{k\epsilon\mathcal{K}} \Bigg(& Dec_M\Big(\mathcal{P}_{acc}\pi_{k_1}^\dagger P_{k_2}\Big(\sum_{P\in\mathbb{P}_{(n+c+h)}} \alpha_P P_C \otimes U_R^P\Big)(P_{k_2}\pi_{k_1}\psi\pi_{k_1}^\dagger P_{k_2}) \\ & \Big(\sum_{P\in\mathbb{P}_{(n+c+h)}} \overline{\alpha_P} P_C \otimes U_R^{P\dagger}\Big) P_{k_2}\pi_{k_1}\mathcal{P}_{acc}^\dagger\Big) \otimes |\mathrm{acc}\rangle\langle \mathrm{acc}| \\ + tr_M\Big(\mathcal{P}_{rej}\pi_{k_1}^\dagger P_{k_2}\Big(\sum_{P\in\mathbb{P}_{(n+c+h)}} & \alpha_P P_C \otimes U_R^P\Big)(P_{k_2}\pi_{k_1}\psi\pi_{k_1}^\dagger P_{k_2}) \\ & \Big(\sum_{P\in\mathbb{P}_{(n+c+h)}} \overline{\alpha_P} P_C \otimes U_R^{P\dagger}\Big) P_{k_2}\pi_{k_1}\mathcal{P}_{rej}^\dagger\Big)\Omega_M \otimes |\mathrm{rej}\rangle\langle \mathrm{rej}| \Bigg). \end{aligned} \tag{65}$$

From here we apply the Pauli Twirl (Theorem 2.11.1):

$$= \frac{1}{|\mathcal{K}_1|} tr_{0,+} \sum_{k_1 \epsilon \mathcal{K}_1} \Bigg(Dec_M \Big(\mathcal{P}_{acc} \pi_{k_1}^\dagger \Big(\sum_{P \in \mathbb{P}_{(n+c+h)}} |\alpha_P|^2 (P_C \otimes U_R^P) \pi_{k_1} \psi \pi_{k_1}^\dagger (P_C \otimes U_R^{P\dagger}) \Big) \pi_{k_1} \mathcal{P}_{acc}^\dagger \Big) \otimes |\text{acc}\rangle \langle \text{acc}| + tr_M \Big(\mathcal{P}_{rej} \pi_{k_1}^\dagger \Big(\sum_{P \in \mathbb{P}_{(n+c+h)}} |\alpha_P|^2 (P_C \otimes U_R^P) \pi_{k_1} \psi \pi_{k_1}^\dagger (P_C \otimes U_R^{P\dagger}) \Big) \pi_{k_1} \mathcal{P}_{rej}^\dagger \Big) \Omega_M \otimes |\text{rej}\rangle \langle \text{rej}| \Bigg). \quad (66)$$

Since the permutations act on the same register as the attack Paulis, we can move the permutations to be considered to be acting on the Paulis instead of the message and traps:

$$= \frac{1}{|\mathcal{K}_1|} tr_{0,+} \sum_{k_1 \epsilon \mathcal{K}_1} \Bigg(Dec_M \Big(\mathcal{P}_{acc} \Big(\sum_{P \in \mathbb{P}_{(n+c+h)}} |\alpha_P|^2 (\pi_{k_1}^\dagger P_C \pi_{k_1} \otimes U_R^P) \psi (\pi_{k_1}^\dagger P_C \pi_{k_1} \otimes U_R^{P\dagger}) \Big) \mathcal{P}_{acc}^\dagger \Big) \otimes |\text{acc}\rangle \langle \text{acc}| + tr_M \Big(\mathcal{P}_{rej} \Big(\sum_{P \in \mathbb{P}_{(n+c+h)}} |\alpha_P|^2 (\pi_{k_1}^\dagger P_C \pi_{k_1} \otimes U_R^P) \psi (\pi_{k_1}^\dagger P_C \pi_{k_1} \otimes U_R^{P\dagger}) \Big) \mathcal{P}_{rej}^\dagger \Big) \Omega_M \otimes |\text{rej}\rangle \langle \text{rej}| \Bigg). \quad (67)$$

Finally we apply the projectors and notice that $\mathcal{K}_1 = \Pi_{(n+c+h)}$:

$$= \frac{1}{\left|\Pi_{(n+c+h)}\right|} tr_{0,+} \Bigg(\sum_{\pi \epsilon \Pi_{(n+c+h)}} \Bigg(Dec_M \Big(\sum_{P | \pi^\dagger P \pi \in \mathbb{P}_{\mathscr{E}}} |\alpha_P|^2 (\pi^\dagger P_C \pi \otimes U_R^P) \psi (\pi^\dagger P_C \pi \otimes U_R^{P\dagger}) \Big) \otimes |\text{acc}\rangle \langle \text{acc}| + tr_M \Big(\sum_{P | \pi^\dagger P \pi \in \mathbb{P}_{(n+c+h)} \setminus \mathbb{P}_{\mathscr{E}}} |\alpha_P|^2 (\pi^\dagger P_C \pi \otimes U_R^P) \psi (\pi^\dagger P_C \pi \otimes U_R^{P\dagger}) \Big) \Omega_M \otimes |\text{rej}\rangle \langle \text{rej}| \Bigg) \Bigg). \quad (68)$$

Then:

$$\frac{1}{2}\Big\|\frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}}\mathscr{E}_k(\rho_{MR})-\mathscr{F}(\rho_{MR})\Big\|_1$$

$$=\frac{1}{2}\Big\|\frac{1}{\left|\Pi_{(n+c+h)}\right|}\sum_{\pi\epsilon\Pi_{(n+c+h)}}\Bigg(tr_{0,+}\Big(Dec_M\Big(\sum_{P|\pi^\dagger P\pi\in\mathbb{P}_{\mathscr{E}}}|\alpha_P|^2\,(\pi^\dagger P_C\pi\otimes U_R^P)\psi$$

$$(\pi^\dagger P_C\pi\otimes U_R^{P\dagger})\Big)\otimes|\mathrm{acc}\rangle\langle\mathrm{acc}|$$

$$+tr_M\Big(\sum_{P|\pi^\dagger P\pi\in\mathbb{P}_{(n+c+h)}\setminus\mathbb{P}_{\mathscr{E}}}|\alpha_P|^2\,(\pi^\dagger P_C\pi\otimes U_R^P)\psi(\pi^\dagger P_C\pi\otimes U_R^{P\dagger})\Big)\Omega_M\otimes|\mathrm{rej}\rangle\langle\mathrm{rej}|\Big)$$

$$-tr_{C_1C_2}\Big(\sum_{P|\pi^\dagger P\pi\in\mathbb{P}_{\mathscr{F}}}|\alpha_P|^2\,(\pi_{C_2}^\dagger P_{C_2}\pi_{C_2}\otimes U_R^P)(\phi_{MRC_1C_2})$$

$$(\pi_{C_2}^\dagger P_{C_2}\pi_{C_2}\otimes U_R^{P\dagger})\Big)\otimes|\mathrm{acc}\rangle\langle\mathrm{acc}|$$

$$-tr_{MC_1C_2}\Big(\sum_{P|\pi^\dagger P\pi\notin\mathbb{P}_{\mathscr{F}}}|\alpha_P|^2\,(\pi_{C_2}^\dagger P_{C_2}\pi_{C_2}\otimes U_R^P)(\phi_{MRC_1C_2})$$

$$(\pi_{C_2}^\dagger P_{C_2}\pi_{C_2}\otimes U_R^{P\dagger})\Big)\Omega_M\otimes|\mathrm{rej}\rangle\langle\mathrm{rej}|\Bigg)\Big\|_1. \tag{69}$$

We will subtract the accepted states in the ideal protocol from those accepted in the real protocol and we will subtract the rejected states in the real protocol from the rejected states in the ideal protocol. Note that
$\mathbb{P}_{\mathscr{E}}\setminus\mathbb{P}_{\mathscr{F}}=\{P\otimes R\otimes Q|P\in\mathbb{P}_n,\omega(P)>t,R\in\{I,Z\}^{\otimes c},Q\in\{I,X\}^{\otimes h}\}$.

$$=\frac{1}{2}\Big\|\frac{1}{\left|\Pi_{(n+c+h)}\right|}\sum_{\pi\epsilon\Pi_{(n+c+h)}}\sum_{P|\pi^\dagger P\pi\in\mathbb{P}_{\mathscr{E}}\setminus\mathbb{P}_{\mathscr{F}}}\Bigg(tr_{0,+}\Big(Dec_M(|\alpha_P|^2\,(\pi^\dagger P_C\pi\otimes U_R^P)\psi$$

$$(\pi^\dagger P_C\pi\otimes U_R^{P\dagger}))\Big)\otimes|\mathrm{acc}\rangle\langle\mathrm{acc}|$$

$$-tr_{MC_1C_2}\Big(|\alpha_P|^2\,(\pi_{C_1}^\dagger P_{C_1}\pi_{C_1}\otimes U_R^P)(\phi_{MRC_1C_2})$$

$$(\pi_{C_1}^\dagger P_{C_1}\pi_{C_1}\otimes U_R^{P\dagger})\Big)\Omega_M\otimes|\mathrm{rej}\rangle\langle\mathrm{rej}|\Bigg)\Big\|_1. \tag{70}$$

Here we will use the triangle inequality to remove the sums from the trace distance:

$$\leq \frac{1}{2}\frac{1}{\left|\Pi_{(n+c+h)}\right|}\sum_{\pi\in\Pi_{(n+c+h)}}\sum_{P|\pi^\dagger P\pi\in\mathbb{P}_{\mathscr{E}}\setminus\mathbb{P}_{\mathscr{F}}}\Big\|tr_{0,+}\Big(Dec_M(|\alpha_P|^2(\pi^\dagger P_C\pi\otimes U_R^P)\psi (\pi^\dagger P_C\pi\otimes U_R^{P\dagger}))\Big)\otimes|\text{acc}\rangle\langle\text{acc}| - tr_{MC_1C_2}\Big(|\alpha_P|^2(\pi_{C_1}^\dagger P_{C_1}\pi_{C_1}\otimes U_R^P)(\phi_{MRC_1C_2}) (\pi_{C_1}^\dagger P_{C_1}\pi_{C_1}\otimes U_R^{P\dagger})\Big)\Omega_M\otimes|\text{rej}\rangle\langle\text{rej}|\Big\|_1. \quad (71)$$

Since the maximum trace distance between two states is 1 we have:

$$\leq \frac{1}{\left|\Pi_{(n+c+h)}\right|}\sum_{k_1\epsilon\mathcal{K}_1}\sum_{P|\pi^\dagger P\pi\in\mathbb{P}_{\mathscr{E}}\setminus\mathbb{P}_{\mathscr{F}}}|\alpha_P|^2. \quad (72)$$

Now if we let η_P be the number of permutations, π of P such that $\pi^\dagger P\pi\in\mathbb{P}_{\mathscr{E}}\setminus\mathbb{P}_{\mathscr{F}}$, then the above can be written as:

$$=\frac{1}{\left|\Pi_{(n+c+h)}\right|}\sum_{P\in\mathbb{P}_{(n+c+h)}}\eta_P\times|\alpha_P|^2. \quad (73)$$

Here, we cite Theorem 5.2.1, which gives us $\eta_P\leq\binom{n}{t+1}(t+1)!((n+c+h)-(t+1))!$. Thus, since $\sum_{P\in\mathbb{P}_{n+c+h}}|\alpha_P|^2=1$, the above expression can be bounded by:

$$\begin{aligned}
&\leq\frac{1}{(n+c+h)!}\binom{n}{t+1}(t+1)!(n+c+h-(t+1))!\\
&=\frac{\prod_{i=1}^{n}i\prod_{i=1}^{n+c+h-t-1}i}{\prod_{i=1}^{n-t-1}i\prod_{i=1}^{n+c+h}i}\\
&=\frac{\prod_{i=n-t}^{n}i}{\prod_{i=n+c+h-t}^{n+c+h}i}\\
&=\prod_{i=0}^{t}\frac{n-t+i}{n+c+h-t+i}\\
&\leq\prod_{i=0}^{t}\frac{n}{n+c+h-t}=\Big(\frac{1}{1+\frac{c}{n}+\frac{h}{n}-\frac{t}{n}}\Big)^{t+1}
\end{aligned} \quad (74)$$

Therefore, $D\Big(\frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}}\mathscr{E}_k(\rho_{MR}), \mathscr{F}(\rho_{MR})\Big) \leq (\frac{1}{1+\frac{c}{n}+\frac{h}{n}-\frac{t}{n}})^{t+1}, \forall \rho_{MR}.$ □

We note that the bound provided above is not tight. It was achieved by replacing each fraction in the sum with the single largest possible value, obtained by taking the largest possible numerator in the sum and dividing by the smallest possible denominator. This is clearly not a tight bound and given more information about c and h, it could be possible to easily find a tighter bound. However, while this is not tight, it is sufficient to show security.

We also note that as can be seen in [BW16], given $c = h = n$, we can simplify the bound to $\frac{1}{3^{t+1}}$. This is very similar to the bound in [BGS13] of $(\frac{2}{3})^{d/2}$. Note that the trap code in [BGS13] uses the error *detection* property of the code. Since a code of distance d can detect up to $d/2$ errors, this bound is consistent with our bound of $(\frac{1}{3})^{t+1}$.

Chapter 6

Conclusions

In this thesis, we have introduced quantum information and quantum message authentication. We discussed a security model that relies on a simulator. Finally, we have proven the security of both the Clifford and trap codes, using efficient simulators.

Following this work, there are a few questions that could be addressed in further work. Currently we study these two protocols in a noiseless setting, which is ultimately impractical in a real world setting. It would therefore be useful to have an understanding of how these codes could handle noise. The trap code in particular seems to be a good candidate for a QMA scheme for a noisy channel because of the original error correcting code applied to the message. There is likely a relatively simple extension of our current results that would find a bound on the amount of error that should be allowed on the traps to account for noise, but still catch adversarial attacks with high probability. This would proceed by finding a non-zero limit to the number of traps that could be accepted when changed as noise but still under the number that would need to be changed by an adversary to alter a message qubit.

Furthermore, there is the question of whether or not the trap code could be simplified by removing the Pauli encoding. The Paulis currently only serve to allow us to remove the cross terms that we would have when we write the attack unitary as the sum of Paulis. We suspect it is the case that the Paulis are not necessary and that the permutation would be sufficient to prove security. A formal proof, however, is left to future work.

Finally, we have mentioned that the efficient simulations are typically a crucial requirement to prove UC security. The next logical step would be to prove UC security using the results of this thesis.

Appendix A

Summary of Notation

Summary of the notation used, in order of appearance.

Symbol	Meaning
$\|\cdot\rangle$	Dirac's *ket* notation; column vector
$\|0\rangle$, $\|1\rangle$	Computational basis states
$\|+\rangle$, $\|-\rangle$	Hadamard basis states
$\|\psi\rangle$	Arbitrary pure qubit, $\|\psi\rangle = \alpha\|0\rangle + \beta\|1\rangle$ for $\alpha, \beta \in \mathbb{C}$
U	Unitary operator
I, X, Z, Y	Single qubit Pauli gates; identity, bit flip, phase flip, and bit and phase flip, respectively
H	Single qubit Hadamard gate; takes $\|0\rangle$ to $\|+\rangle$ and $\|1\rangle$ to $\|-\rangle$ and vice versa
M_m	Measurement projector corresponding to the outcome m
$\otimes$	Tensor product
$\|\Psi^+\rangle$, $\|\Psi^-\rangle$, $\|\Phi^+\rangle$, $\|\Phi^-\rangle$	Bell states; EPR pair; pair of maximally entangled qubits
ρ	Density matrix; most arbitrary representation of a qubit, or of multiple qubits

Symbol	Meaning
$\mathcal{H}$	Hilbert space
tr	trace of a matrix; sum of diagonal entries of a matrix
$\mathcal{B}(\mathcal{H})$	Space of bounded linear operators on a Hilbert space; quantum operations
$CPTP$ map	Completely positive trace preserving map; quantum channel
$D(\cdot,\cdot)$	Trace distance
$\|\cdot\|_1$	Trace norm
Π_n	Set of all n-qubit permutations
π	Permutation operator
$\mathbb{P}_n$	Set of all n-qubit Pauli operators
P, Q, R	Pauli matrices, dimension should be clear from context
$\mathcal{C}_n$	Clifford group on n qubits
C	Clifford operator
$\mathcal{E}$	Encoding map
$\mathcal{D}$	Decoding map
$\mathcal{K}$	Set of all keys
$\mathscr{E}$	Real world protocol
$\mathscr{F}$	Ideal world protocol
$\mathcal{P}$	Measurement projector
$Enc(\cdot)$	Error correcting code applied to $\cdot$
$Dec(\cdot)$	Decoding (of the error correcting code) applied to $\cdot$
$\mathbb{P}_{\mathscr{E}}$	Set of all Paulis that the real world protocol will accept being applied to the cipher system

Symbol	Meaning
$\mathbb{P}_{\mathscr{F}}$	Set of all Paulis that the simulator would accept being applied to the half EPR pairs
η_P	Number of permutations of P that would leave the effect of P on a cipher register undetected on the traps

Bibliography

[ABE10] Dorit Aharonov, Michael Ben-Or, and Elad Eban. Interactive proofs for quantum computations. In *Innovations in Computer Science—ICS 2010*, pages 453–469, 2010.

[BB84] Charles H. Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. In *International Conference on Computers, Systems and Signal Processing*, pages 175–179, 1984.

[BBCS01] Charles H. Bennett, Gilles Brassard, Claude Crépeau, and Marie-Hélène Skubiszewska. Practical quantum oblivious transfer. In *Advances in Cryptology—CRYPTO 1991*, pages 351–366, 2001.

[BCG+02] Howard Barnum, Claude Crépeau, Daniel Gottesman, Adam Smith, and Alain Tapp. Authentication of quantum messages. In *43rd Annual Symposium on Foundations of Computer Science—FOCS 2002*, pages 449–458, 2002.

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

[BCMS97] Gilles Brassard, Claude Crépeau, Dominic Mayers, and Louis Salvail. A brief review on the impossibility of quantum bit commitment, 1997. arXiv:quant-ph/9712023.

[BEM+07] Dagmar Bruß, Gábor Erdélyi, Tim Meyer, Tobias Riege, and Jörg Rothe. Quantum cryptography: A survey. *ACM Computing Surveys—CSUR*, 39(2), 2007.

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

[BGS13] Anne Broadbent, Gus Gutoski, and Douglas Stebila. Quantum one-time programs. In *Advances in Cryptology—CRYPTO 2013*, pages 344–360, 2013.

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

[Bro15] Anne Broadbent. *How to Verify a Quantum Computation.* 2015. arXiv:1509.09180.

[BS16] Anne Broadbent and Christian Schaffner. Quantum cryptography beyond quantum key distribution. *Designs, Codes and Cryptography*, 78:351–382, 2016.

[BW16] Anne Broadbent and Evelyn Wainewright. Efficient simulation for quantum message authentication. 2016. To appear in *9th International Conference on Information Theoretic Security—ICITS 2016*. arXiv:1607.03075.

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd Annual Symposium on Foundations of Computer Science—FOCS 2001*, pages 136–145, Oct. 2001.

[CGL99] Richard Cleve, Daniel Gottesman, and Hoi-Kwong Lo. How to share a quantum secret. *Physical Review Letters*, 83(3):648–651, Jul 1999.

[DCEL09] Christoph Dankert, Richard Cleve, Joseph Emerson, and Etera Livine. Exact and approximate unitary 2-designs and their application to fidelity estimation. *Physical Review A*, 80:012304, 2009.

[DNS12] Frédéric Dupuis, Jesper Buus Nielsen, and Louis Salvail. Actively secure two-party evaluation of any quantum operation. In *Advances in Cryptology—CRYPTO 2012*, pages 794–811, 2012.

[Feh10] Serge Fehr. Quantum cryptography. *Foundations of Physics*, 40(5):494–531, Jan 2010.

[GHW09] Daniel Greenberger, Klaus Hentschel, and Friedel Weinert. *Compendium of Quantum Physics: Concepts, Experiments, History and Philosophy*. Physics and astronomy online library. Springer Berlin Heidelberg, 2009.

[Got97] Daniel Gottesman. *Stabilizer Codes and Quantum Error Correction*. PhD thesis, California Institute of Technology, 1997.

[HLM11] Patrick Hayden, Debbie Leung, and Dominic Mayers. Universal composable security of quantum message authentication with key recycling. Presented at QCRYPT 2011, 2011.

[KS14] Rober Koenig and John A. Smolin. How to efficiently select an arbitrary clifford group element. *Journal of Mathematical Physics*, 55(12):122202, 2014.

[MPZ01] Chiara Macchiavello, G Massimo Palma, and Anton Zeilinger, editors. *Quantum Computation and Quantum Information Theory: Reprint Volume with Introductory Notes for ISI TRM Network School*. World Scientific, 2001.

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

[Unr10] Dominique Unruh. Universally composable quantum multi-party computation. In *Advances in Cryptology—EUROCRYPT 2010*, pages 486–505, 2010.

[Wat11] John Watrous. Guest column: An introduction to quantum information and quantum circuits. *ACM SIGACT News*, 42(2):52–67, 2011.