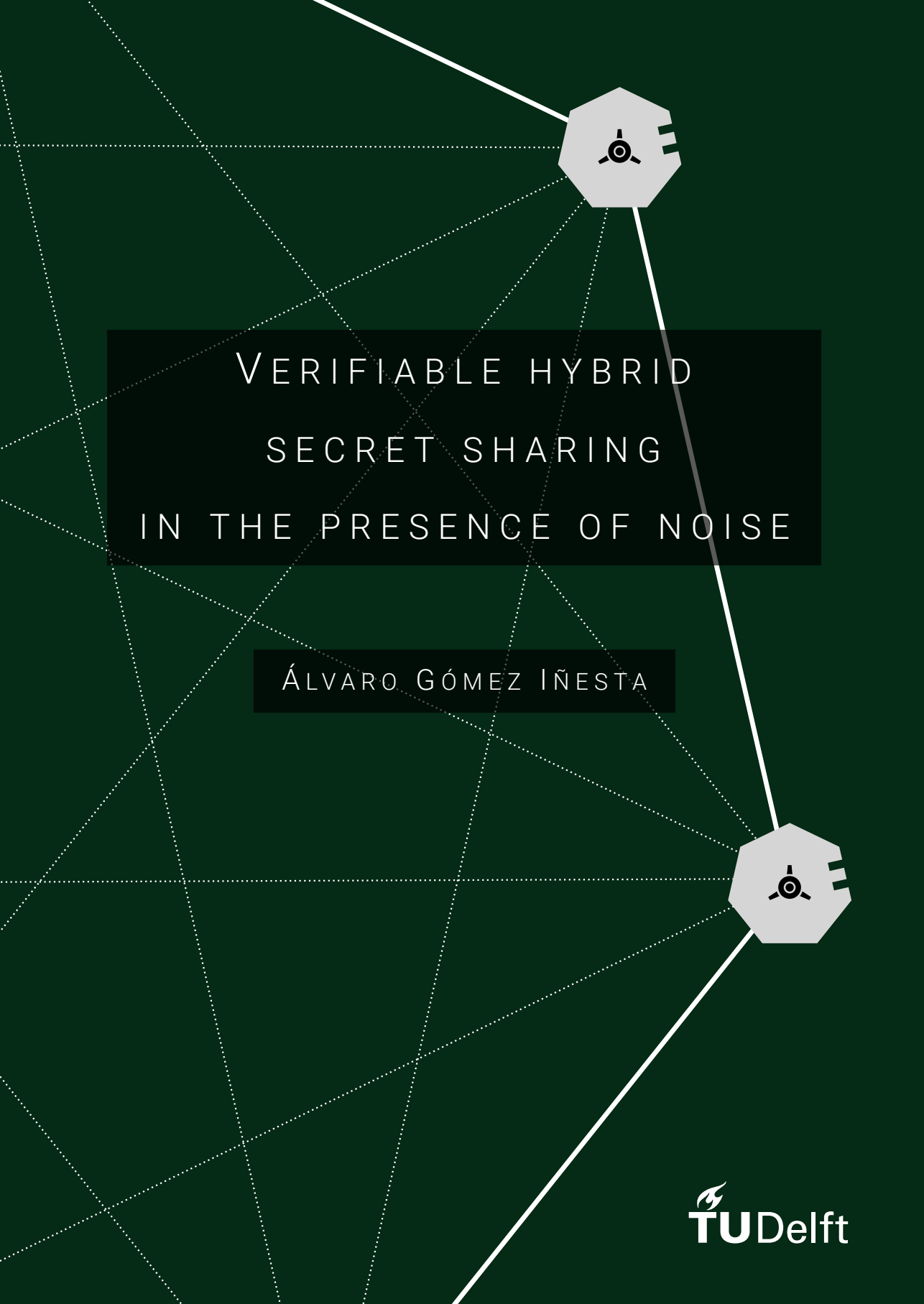


Verifiable hybrid secret sharing in the presence of noise

Álvaro Gómez Iñesta

TU Delft

Verifiable hybrid secret sharing in the presence of noise

by

Álvaro Gómez Iñesta

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday February 21st 2020, at 9:30h.

Student number:	4914686	
Project duration:	September 2, 2019 – February 7, 2020	
Thesis committee:	Prof. dr. S. D. C. Wehner,	TU Delft, supervisor
	Dr. D. Elkouss Coronas,	TU Delft
	Dr. M. T. Wimmer,	TU Delft
Daily supervisor:	PhD candidate V. Lipinska,	TU Delft

An electronic version of this thesis is available at
http://repository.tudelft.nl/.

TU Delft

Casimir research school

After thirty-nine years of looking at myself in the mirror
I have learnt nothing about me,
but I am becoming an expert in mirrors

— Neorrabioso

Abstract

Verifiable quantum secret sharing (VQSS) is the task of sharing a secret quantum state among the n nodes of a quantum network, in a way that it is possible to verify that the secret has been correctly distributed. A number of protocols that perform this task have been proposed. In particular, the verifiable hybrid secret sharing (VHSS) scheme proposed by Lipinska et al. (2019) realizes this task while reducing the number of required qubits, as compared to other existing protocols.
The VHSS scheme was proven secure for noiseless quantum networks. In this work, we analyze its performance in the presence of noise. To do that, we first define different noise models in which qubits can be randomly erased or depolarized. Then, we propose several modifications for the protocol that allow us to run it more efficiently on quantum networks described by such noise models.
Additionally, we explore a new approach to the VQSS task using approximate quantum error correction. We propose the verifiable trap secret sharing (VTSS) protocol, which combines the scheme by Lipinska et al. (2019) and the trap code by Broadbent et al. (2013). Our protocol achieves the same functionalities as the VHSS protocol and only requires that a strict majority of the nodes follow the protocol honestly, raising the maximum number of nodes that can cheat from $\lfloor \frac{n-1}{4} \rfloor$ to $\lfloor \frac{n-1}{2} \rfloor$. We can do this at the cost of increasing the probability of error in the protocol.

Acknowledgements

This report is the result of six months of work. During this time, I have been surrounded and supported by many wonderful people, and I would like to express my deepest gratitude to all of them.

First and foremost, I would like to thank my supervisor, Prof. Stephanie Wehner, and my daily supervisor, PhD candidate Victoria Lipinska, for giving me the opportunity to conduct my thesis in their group. Their excellent feedback and continuous support has been key to achieve all the goals of this project. Victoria, thank you for all the time invested in sharing your expertise with me and for the passion you transmitted—and sorry for those extremely long meetings!
I would also like to thank the rest of the people in the group, who warmly welcomed me and made my work much easier and enjoyable. In particular, thanks to Jérémy and Kenneth for our insightful discussions, and to Helena and Jenny for taking care of most of the administrative aspects of the project.
Finally, special thanks to Prof. Elkouss and Prof. Wimmer for kindly accepting the invitation to join the thesis committee and taking the time to read this report.

Huge thank you to all my friends—in particular to Maribel—for distracting me from work, making me laugh every day and being there for me. I hope you enjoyed my company and jokes as much I enjoyed yours. Finally, the people I should thank the most are my family. Papá, Mamá, Tori, thanks for putting up with my bad temper and for making me smile. Tori, thanks for being so genuine, there is still a lot we have to learn from you. Thank you to the rest of the family too, you are all essential. And thank you, Lara, for your love, for the memes, and for giving me the strength to carry on, no matter how hard the road ahead was. My deepest gratitude and all my love. *hH-h.*

This thesis has been financially supported by "la Caixa" Foundation (ID 100010434), fellowship code LCF/BQ/EU18/11650059, and the QuTech Academy Scholarship 2018-2020. I would like to conclude with my gratitude towards both institutions.

Álvaro Gómez Iñesta
Delft, February 2020

Contents

1 Introduction **1**

2 Theory of error correction **5**
2.1 Classical error-correcting codes 5
2.1.1 Classical linear codes 10
2.2 Quantum error-correcting codes 10
2.2.1 Calderbank-Shor-Steane codes 14
2.3 Approximate quantum error correction 15
2.3.1 Quantum authentication 16
2.3.2 The trap code 19

3 Secret sharing protocols **21**
3.1 From classical to quantum 22
3.2 Verifiable hybrid secret sharing (VHSS) 22
3.2.1 The protocol 23

4 Quantum noise in the VHSS protocol **33**
4.1 Erasures in the secret 34
4.2 Erasures in the secret and the ancillae 43
4.3 Depolarizing noise 50
4.4 Efficient qubit sharing 54

5 Approximate quantum error correction and verifiable quantum secret sharing **59**
5.1 Approximate verifiable quantum secret sharing 60
5.2 Verifiable trap secret sharing 61

6 Conclusions and outlook **71**

Bibliography **76**

A Derivation of $f_{\mathrm{v}}(n, t, q, t_{\mathrm{c}}, t_{\mathrm{d}})$ **77**

B Convergence of the sampling algorithm **85**

C Lower bounds for the binomial distribution **91**

List of Acronyms

AQEC	Approximate quantum error correction
AQECC	Approximate quantum error-correcting code
CE	'Channel erasure' (noise model defined in Chapter 4)
CD	'Channel depolarization' (noise model defined in Chapter 4)
CSS	Calderbank-Shor-Steane (refers to a type of quantum codes)
DS	'Depolarization on the secret' (noise model defined in Chapter 4)
DSA	'Depolarization on the secret and the ancillas' (noise model defined in Chapter 4)
EPR	Einstein-Podolsky-Rosen (refers to a maximally entangled 2-qubit state)
ES	'Erasures on the secret' (noise model defined in Chapter 4)
ESA	'Erasures on the secret and the ancillas' (noise model defined in Chapter 4)
PT	Purity-testing (refers to a type of quantum codes)
TTP	Trusted third party
QAS	Quantum authentication scheme
QEC	Quantum error correction
QECC	Quantum error-correcting code
QSS	Quantum secret sharing
VHSS	Verifiable hybrid secret sharing
VQSS	Verifiable quantum secret sharing
WQSS	Weak quantum secret sharing

1

Introduction

Verifiable secret sharing is a classical cryptography task for a network in which a dealer wants to share a secret with several nodes by giving a piece of information to each of them. We call each of these pieces of information a share. The dealer must distribute them in a way that (*i*) the secret can be reconstructed only when a sufficient number of shares are combined together, (*ii*) small groups of curious nodes cannot retrieve any information about the secret, and (*iii*) all the nodes can collectively check that their shares are consistent with some secret. Note that the nodes must be able to verify the consistency of their shares without revealing any information about the secret. Different protocols for verifiable secret sharing have been proposed [1, 2, 3, 4], and they are widely used as a subroutine in other cryptographic applications. Some of these applications are secure multi-party computation [5, 6, 7], byzantine agreement [8], and end-to-end auditable voting systems [9].

In the recent years the scientific community has experienced a growing interest in quantum technologies. In particular, a lot of effort is dedicated to the development of quantum networks that would enable remote quantum communication [10, 11, 12]. The practical use of these networks relies on the design of cryptographic protocols tailored to their needs. As in the classical framework, many quantum applications, such as secure quantum multi-party computation, require a *verifiable quantum secret sharing* (VQSS) subroutine. This task is similar to its classical analogue, but the secret to be shared is a quantum state, which is encoded into many other states that are distributed among the nodes of a quantum network.
The first proposal of a VQSS protocol is the one by Crépeau, Gottesman, and Smith (2002) [13]. However, this approach requires a large number of qubits at each node of the network. The nodes of early quantum networks will have limited quantum resources, and therefore it would be convenient to design a more efficient VQSS scheme in terms of the number of qubits required.
To remedy that, in 2019, Lipinska et al. proposed the *verifiable hybrid secret sharing* (VHSS) protocol [14]. Their approach preserves the same properties as the previous

1

VQSS and also achieves a reduction of the quantum resources by combining VQSS with classical encryption and classical secret sharing.

Currently, one of the technical issues that prevent the deployment of quantum networks is the presence of noise which influences the quantum states we wish to send in the network. In order to close the gap between theory and experimental implementations, we review the VHSS protocol and propose several modifications aimed at improving its performance when qubits are subject to different types of noise. In addition, the VHSS protocol can tolerate up to $\lfloor \frac{n-1}{4} \rfloor$ cheating nodes, where n is the total number of nodes, that can perform any joint operation on their shares and do not follow the protocol honestly. In the last part of this thesis, we investigate how to use approximate quantum error correction to increase the number of cheaters that the protocol tolerates.
Our efforts have been directed towards answering the following research questions:

> The VHSS protocol from [14] assumes noiseless quantum states. In which ways are the security statements affected when considering noisy qubits?

> How can we modify the protocol in order to improve its performance in noisy networks?

> The protocol tolerates up to $\lfloor \frac{n-1}{4} \rfloor$ cheaters. Can we use approximate error correction to lift this bound while keeping the number of quantum resources low enough?

Outline and contributions. The structure of this thesis is the following:

- In Chapter 2 we give a brief overview of the classical and quantum theory of error correction and introduce the concepts and definitions that are used throughout this thesis. First, in Section 2.1, we discuss the fundamentals of classical error-correcting codes. Then, in Sections 2.2 and 2.3, we focus on exact and approximate quantum error correction, respectively.
- Chapter 3 contains a review of the quantum secret sharing protocols mentioned before. In particular, the VHSS protocol is discussed in detail in Section 3.2.
- In Chapter 4 we present a large part of our contributions. In Sections 4.1 and 4.2 we consider several noise models in which qubits can be erased with certain probability while traveling from node to node. Then, we analyze the performance of the VHSS scheme in quantum networks described by these models. Importantly, we also provide modifications that enable the usage of this protocol in such networks. Next, in Section 4.3, we perform a similar analysis with a different type of noise models, in which qubits are depolarized

instead of erased. Finally, in Section 4.4, we propose a major modification of the protocol. Namely, we discuss an alternative strategy to distribute the quantum shares among the nodes in such a way that the effects of noise are reduced. This chapter gives an extensive answer to the first two research questions.

- In Chapter 5 we aim at increasing the number of cheaters that the protocol tolerates. In Section 5.1, we review previous work on this direction, and in Section 5.2 we present the rest of our contributions. We propose a novel scheme for verifiable quantum secret sharing that achieves the desired functionality while preserving the security of the protocol, answering to the third research question. Note that this last section is presented as an open discussion in which some formal proofs are still required.

2

Theory of error correction

The goal of most cryptographic protocols is to ensure the secrecy of the communication between two or more parties. This communication consists in information exchange over a physical medium called *channel*. Examples of channels are air, for oral conversations, and cables, for electrical signals travelling inside electronic devices.
The choice of a proper channel is crucial to successfully send and receive messages, since it might introduce some noise. Nevertheless, even in the presence of noise, it is possible to communicate by encoding the messages. One can employ a *code* to transform the original information into more robust encoded data. This way, the encoded message will protect the original information by adding some amount of redundancy. As a simple example, imagine a conversation between Alice and Bob in a noisy environment. If Alice's last sentence was distorted and Bob did not understand it, he might ask her to repeat it. If the problem persists, several repetitions may be necessary, yielding a simple *repetition code*.
Codes can be used to protect classical and quantum data. In the quantum framework, a quantum state can be encoded into a larger state, and the redundancy of the message is stored in the entanglement between the encoded qubits.

In this chapter, we start by reviewing the fundamentals of classical error-correcting codes (Section 2.1). Then, in Section 2.2, we discuss the basics of quantum error correction, focusing on the properties of Calderbank-Shor-Steane codes. Ultimately, in Section 2.3, we introduce the concepts of approximate quantum error correction and quantum authentication, as well as the trap code, a specific construction that is used later in Chapter 5.

2.1. Classical error-correcting codes

Cryptography is a field at the intersection of mathematics, physics, and computer science that we already explore as children, usually when we want to secretly

communicate in the presence of adult eavesdroppers. A common practice is to play around with words to create our own secret codes. For example, two siblings may refer to their '*parents*' as '*birds*' and to '*candy*' as '*a walk*'. Then, the sentence '*when the birds leave, we can go for a walk*' can be used to secretly propose going for candy when their parents leave home. This way, the siblings have constructed a code $C = \{birds,\ a\ walk\}$ which is associated with the following encoding map: *parents* $\rightarrow$ *birds*, *candy* $\rightarrow$ *a walk*.

Definition 2.1.1. A **classical error-correcting code** C over the finite alphabet Σ is a subset of Σ^n, where Σ^n is the set of all strings of length n generated by combining elements from Σ (each element can be repeated an arbitrary number of times) [15].

The elements of the code are called *codewords*. Codes are associated with an encoding map, which maps each of the possible messages to a different codeword. Consequently, we require the size of C to be the same as the size of the set of all possible messages. In many applications, $\Sigma = \mathbb{Z}_q$, where $\mathbb{Z}_q$ is the set of integers modulo q. Throughout this work, we will only consider classical codes over $\mathbb{Z}_2 = \{0, 1\}$.
The mapping of a message to a codeword is generally performed to make it more robust against noise. In classical codes defined over $\mathbb{Z}_2$, noise can produce two types of error: *bit flips* and *bit erasures*. The former changes 0s to 1s and vice versa, while the latter erases the bit. The fundamental difference between both of them is that bit erasures are flagged[1]. Hence, we always know the positions of the erasures within the string before performing any error detection operation. For bit flips, this information is not available a priori. We will refer to bit flips as errors and to bit erasures as erasures.

The error-correcting power of a code is measured with the code distance.

Definition 2.1.2. The **distance** d of a code is the minimum Hamming distance between any two codewords, i.e.,

$$d = \min_{\substack{x,y \in C \\ x \neq y}} d_{\text{H}}(x, y), \tag{2.1.1}$$

where the Hamming distance $d_{\text{H}}(x, y)$ between two strings x and y is the number of symbols in which they differ.

We can model the bit flip errors by introducing the error string e, which has 1s in the bit flip positions and 0s in the rest of them. The total number of errors is denoted by t, i.e., $d_{\text{H}}(0, e) = t$. Using this notation, the string obtained after adding errors to a codeword $x \in C$ is $x + e$, where the sum is a bitwise addition modulo 2. This string is at distance $d_{\text{H}}(x, x + e) = t$ from the original codeword. We say that a code can correct t errors iff $x + e$ is closer to x than to any other codeword y, for any x, y

[1] Deletion of a bit without raising a flag is also possible, although this type of error is not common in most physical setups and therefore we do not consider it in this thesis.

and e. Mathematically, this condition can be written as

$$\begin{array}{c} C \text{ can} \\ \text{correct } t \text{ errors} \end{array} \Leftrightarrow \begin{array}{c} d_{\text{H}}(x+e,y) > t,\ \forall x,y \in C \text{ s.t. } x \neq y \\ \text{and } \forall e \text{ s.t. } d_{\text{H}}(0,e) = t. \end{array} \tag{2.1.2}$$

Regarding erasures, we define the *punctured codeword* $x|_{\mathbf{b}}$ as a codeword that suffered from bit erasures in the positions specified by $\mathbf{b} = (b_1, b_2, \ldots, b_u)$, $b_i \in \mathbb{Z}_n$. The number of erasures is $u = |\mathbf{b}|$, and the distance between two different punctured codewords satisfies the following condition:

$$d_{\text{H}}(x|_{\mathbf{b}}, y|_{\mathbf{b}}) \geq d - u,\ \forall x,y \in C \text{ s.t. } x \neq y \text{ and } \forall \mathbf{b} \text{ s.t. } |\mathbf{b}| = u. \tag{2.1.3}$$

We say that a code can correct u erasures iff we can distinguish any two punctured codewords, i.e.,

$$\begin{array}{c} C \text{ can} \\ \text{correct } u \text{ erasures} \end{array} \Leftrightarrow \begin{array}{c} d_{\text{H}}(x|_{\mathbf{b}}, y|_{\mathbf{b}}) > 0,\ \forall x,y \in C \text{ s.t. } x \neq y \\ \text{and } \forall \mathbf{b} \text{ s.t. } |\mathbf{b}| = u. \end{array} \tag{2.1.4}$$

Theorem 2.1.1. *Any classical error-correcting code can correct t errors or $2t$ erasures iff it has distance $d \geq 2t+1$* [15].

Proof. We start by considering two different codewords $x, y \in C$. The error string is e and the total number of errors is t. First, we use the triangle inequality [16] to derive the following partial result:

$$\begin{aligned} d_{\text{H}}(x,y) \leq d_{\text{H}}(x,x+e) + d_{\text{H}}(x+e,y) \quad &\Leftrightarrow \quad d \leq t + d_{\text{H}}(x+e,y) \\ &\Leftrightarrow \quad d_{\text{H}}(x+e,y) \geq d - t,\ \forall x,y,e, \end{aligned} \tag{2.1.5}$$

where we have used that $d_{\text{H}}(x,y) \geq d$ for $x \neq y$ and $d_{\text{H}}(0,e) = t$.
($\Rightarrow$) Let us consider a code C that can correct t errors. Assume that the distance is $d < 2t+1$. Let x^* and y^* be two codewords such that $d_{\text{H}}(x^*,y^*) = d$. Then, there exists some e^* such that $d_{\text{H}}(x^*+e^*,y^*) = d-t$, with $t < d$. This e^* only has 1s in t positions in which x^* and y^* differ. Hence,

$$d_{\text{H}}(x^*+e^*,y^*) = d - t < 2t + 1 - t = t + 1 \quad \Leftrightarrow \quad d_{\text{H}}(x^*+e^*,y^*) \leq t. \tag{2.1.6}$$

However, we are considering a code that can correct t errors and therefore $d_{\text{H}}(x^* + e^*, y^*) > t$, $\forall x^*, y^*, e^*$ (see Eq. (2.1.2)). We have arrived to a contradiction, which means that our hypothesis about the distance was not true. Consequently, a code that can correct t errors must have distance $d \geq 2t+1$.
($\Leftarrow$) We consider now the case in which the distance of the code is $d \geq 2t+1$. Using the triangle inequality from Eq. (2.1.5):

$$d_{\text{H}}(x+e,y) \geq d - t \geq t + 1 > t. \tag{2.1.7}$$

Then, the condition from Eq. (2.1.2) is satisfied and the code can correct t errors.
($\Rightarrow$) Regarding erasures, let us first consider a code that can correct $2t$ erasures, and assume that $d < 2t$. Let x^* and y^* be two codewords such that $d_{\text{H}}(x^*,y^*) = d$. Then,

there exists some $\mathbf{b}$ with $|\mathbf{b}| = 2t$ such that all bits in which x^* and y^* differ are erased. Therefore, $d_{\text{H}}(x^*|_{\mathbf{b}^*}, y^*|_{\mathbf{b}^*}) = 0$. This contradicts Condition (2.1.4), which holds for any x, y, and $\mathbf{b}$. Consequently, the hypothesis about the distance was not correct and we conclude that $d \geq 2t + 1$.
($\Leftarrow$) On the other hand, consider now a code with distance $d \geq 2t + 1$. Using Eq. (2.1.3), we obtain

$$d_{\text{H}}(x|_{\mathbf{b}}, y|_{\mathbf{b}}) \geq d - 2t \geq 1, \ \forall x, y \in C \text{ s.t. } x \neq y \text{ and } \forall \mathbf{b} \text{ s.t. } |\mathbf{b}| = 2t,$$

which means that the code can correct $2t$ erasures, according to Eq. (2.1.4). □

An alternative statement of the previous theorem can be found in [16].

Corollary 2.1.1.1. *A code that can correct up to t errors or up to $2t$ erasures has distance $d \in \{2t + 1, 2t + 2\}$* [17].

Proof. A code C_1 with distance $d_1 = 2t + 3$ can correct $t + 1$ errors or $2(t + 1)$ erasures by virtue of Theorem 2.1.1. Then, a code C_2 that can correct t errors or $2t$ erasures ($d_2 \geq 2t + 1$) but not more must have distance $d_2 < 2t + 3$. Therefore, $d_2 \in \{2t + 1, 2t + 2\}$. □

After discussing the meaning of the distance of a code and how it is related to the number of errors it can tolerate, let us provide a definition of error-correcting code that is more convenient for our work than the one given at the beginning of the section:

Definition 2.1.3. An $[n, k, d]$ **classical error-correcting code** C over $\mathbb{Z}_2$ is a subset of $\mathbb{Z}_2^n$ of size $|C| = k$, i.e., it encodes k bits into n bits, and it has distance d.

For a better understanding of the the previous definition, consider the following example: the repetition code. The $[4, 1, 4]$ repetition code encodes 1 bit into 4 bits, and it has two codewords: 0000 and 1111. The distance of this code is 4, as this is the minimum distance between codewords. Similarly, any $[n, 1, d]$ repetition code has two codewords, one of them composed by n zeros and the other formed by n ones. Moreover, $d = n$. Figure 2.1 shows a graphical representation of this family of codes for $n \leq 4$.
Graphically, when a codeword is affected by a single error, the new string is one edge away from the original node. Then, we can correct t errors if the new string is still more than t edges away from the other codeword, i.e., when we can guess which was the original codeword with absolute certainty. For example, take the $n = 4$ code. A single error takes 0000 to any of the yellow nodes and 1111 to any of the blue ones. As a consequence, if we receive a string corresponding to a yellow node and we know that it has not suffered from more than one error, we conclude that the original codeword was 0000 and the error can be corrected. Similarly, blue nodes can be corrected to recover 1111. However, two errors take both codewords to a green node, which makes it impossible to determine whether the codeword was 0000 or 1111. Following this reasoning, the codes from Fig. 2.1 can correct up to $t = 1, 1, 0, 0$

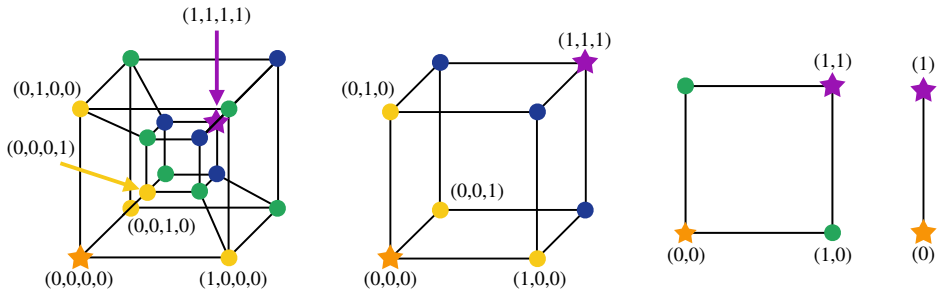


Figure 2.1. Visual representation of $[n, 1, d]$ repetition codes, with $n = d = 4, 3, 2, 1$, from left to right. The codewords are represented with stars. See main text for a detailed discussion.

errors, for $n = 4, 3, 2, 1$, respectively.
Now let us consider bit erasures. In order to correct them, we need the remaining string to correspond to a single codeword with no uncertainty. In the case of the repetition code, we can erase up to $n - 1$ bits, as the remaining string would be 0 for codeword 0000 and 1 for codeword 1111. Then, the codes tolerate up to $u = 3, 2, 1, 0$ erasures for $n = 4, 3, 2, 1$, respectively. Moreover, after erasing a bit from the $[n, 1, n]$ repetition code, the remaining bits form the $[n - 1, 1, n - 1]$ code (the next one to the right in Fig. 2.1). We say then that the $[n - 1, 1, n - 1]$ repetition code is the 1-punctured code of the $[n, 1, n]$ code.
This graphical derivation of t and u provides a helpful interpretation of Theorem 2.1.1 and Corollary 2.1.1.1, which state that any distance d code can correct up to $t = \lfloor (d - 1)/2 \rfloor$ errors.

Finally, another relevant concept that we need to introduce is the dual code.

Definition 2.1.4. The **dual code** of an $[n, k, d]$ classical error-correcting code C is

$$C^{\perp} := \{c^{\perp} \in \mathbb{Z}_2^n \,|\, \langle c^{\perp}, c \rangle = 0, \forall c \in C\},$$

where the inner product is defined as $\langle x, y \rangle := x_1 \cdot y_1 + x_2 \cdot y_2 + \cdots + x_n \cdot y_n \bmod 2$, $\forall x = x_1 x_2 \ldots x_n$, $y = y_1 y_2 \ldots y_n \in \mathbb{Z}_2^n$.

A corollary of the previous definition is the following:

$$\langle x^{\perp}, x \rangle \neq 0,\ x^{\perp} \in C^{\perp} \Rightarrow x \notin C. \tag{2.1.8}$$

On the other hand, note that it is possible that C and $C^{\perp}$ intersect, and even that $C = C^{\perp}$ in the case of self-dual codes [18].
We refer to the distance of the dual code as dual distance $d^{\perp}$. However, according to [19], "*there is no theoretical result known which combines the minimum distance of C and $C^{\perp}$*", and therefore we do not provide any general relation between both distances.

2.1.1. Classical linear codes

Classical linear codes are the main building block for the specific type of quantum codes that we will be using later. Linear codes are defined as a linear subspace of a vector space $\mathbb{F}_q^n$, where $\mathbb{F}_q$ is a finite field with q elements and $n \in \mathbb{Z}^+$, with $\mathbb{Z}^+ := \{1, 2, 3, \dots\}$. Since the finite field we use is $\mathbb{Z}_2$, we provide the following definition for linear codes.

Definition 2.1.5. An $[n, k, d]$ **linear code** C is a linear subspace of the vector space $\mathbb{Z}_2^n$ that has dimension $|C| = k$ and code distance d.

A useful property of linear codes is that a sum of codewords is also a codeword: $x + y \in C$, $\forall x, y \in C$. By linearity, we also find that the zero string[2] is always a codeword: $x + x = 0 \in C$, $\forall x \in C$.
The concept of code distance can be simplified in linear codes:

$$d = \min_{\substack{x,y\in C\\ x\neq y}} d_{\mathrm{H}}(x, y) = \min_{\substack{x,y\in C\\ x\neq y}} d_{\mathrm{H}}(x + y, y + y) = \min_{\substack{z\in C\\ z\neq 0}} d_{\mathrm{H}}(z, 0), \tag{2.1.9}$$

where we have used that $x + y = z \in C$ and $y + y = 0$ in the last step. The previous expression can be interpreted as follows: the distance of a linear code is the minimum number of 1s in a nonzero codeword.
The dual code $C^\perp$ of a code C is always linear, even if C is not [15]. In addition, if C is an $[n, k, d]$ linear code then $C^\perp$ is an $[n, n - k, d^\perp]$ linear code [15].

To conclude this section, we derive a set of addition rules that determine if a sum of strings is a codeword or not.
Let C be an $[n, k, d]$ linear code. Moreover, $x, y \in C$, with $x \neq y$, and $\tilde{x} \notin C$. Then:

$$x + y \in C, \tag{2.1.10a}$$

$$x + \tilde{x} \notin C. \tag{2.1.10b}$$

The first addition rule follows from the linearity of C. The second one can be proven as follows. Assume that $x + \tilde{x} \in C$, with $x \in C$ and $\tilde{x} \notin C$. Then, using linearity of C, we have that $x + (x + \tilde{x}) = \tilde{x} \in C$. However, one of the hypothesis was $\tilde{x} \notin C$. We have arrived to a contradiction, which means that the assumption $x + \tilde{x} \in C$ was wrong. Hence, we conclude that $x + \tilde{x} \notin C$.
It is not possible to state a simple addition rule for the case $\tilde{x} + \tilde{y}$, with $\tilde{x}, \tilde{y} \notin C$. As an example, take the dual code of the $[7, 4, 3]$ Hamming code: $C = \{$0000000, 0001111, 0110011, 0111100, 1010101, 1011010, 1100110, 1101001$\}$. The sum of the words $0000001 \notin C$ and $0000011 \notin C$ is $0000010 \notin C$. However, the sum of the words $0000011 \notin C$ and $0001100 \notin C$ is $0001111 \in C$.

2.2. Quantum error-correcting codes

As in classical communication, the exchange of quantum information through a quantum channel is subject to noise. In the classical case, this noise can either flip or

[2] We represent the zero string $0 \dots 0$ with a single 0 for clarity.

erase a bit. However, quantum channels can introduce a continuum of different errors in a qubit [20]. Moreover, two additional difficulties for quantum error correction are the no-cloning theorem [21], which prevents the creation of back-up copies of quantum states, and the loss of information in each measurement due to the collapse of the wave function. Despite these obstacles, quantum error correction (QEC) is still possible [20]. In the following paragraphs, we discuss the basics of QEC and later we focus on Calderbank-Shor-Steane (CSS) codes, a particular type of quantum error-correcting codes (QECC). We assume that the reader is familiar with quantum information theory and the Dirac notation. For an introduction to them, we refer the reader to [20].

When analyzing errors on qubits, we distinguish between independent and correlated error models. Independent errors happen independently on each qubit and do not affect the rest of them. Hence, they are useful to describe the evolution of non-interacting qubits. On the other hand, correlated errors are joint quantum operations performed over several qubits, possibly involving the use of extra ancillary qubits. All possible errors can be included in either of these two categories. For example, systematic inaccuracies in the implementation of a quantum gate U are modeled by applying the noisy gate $\tilde{U} = EU$, where E is a noise operator acting only on those qubits on which the gate is applied [22]. If U is a single-qubit gate, this is an independent noise model, whereas if U is a multi-qubit gate, this is a correlated noise model.

Independent noise models. Some noisy processes affect each qubit independently, and therefore it is possible to analyze them by considering the case of a single qubit. We distinguish between two general types of independent noise models: erasures and arbitrary errors. The concept of erasure is the same as in classical error correction: the qubit is lost and a classical flag is generated, indicating that the qubit has been erased.
Arbitrary errors correspond to rotations of the qubit in the Bloch sphere. As previously stated, there is a continuum of possible arbitrary errors. Nevertheless, the Pauli matrices and the 2×2 identity form an orthogonal basis for the complex Hilbert space of 2×2 matrices [20]. Therefore, an arbitrary error E acting on a single qubit can be decomposed as

$$E = e_I I + e_X X + e_Y Y + e_Z Z = e_I I + e_X X + e_{XZ} XZ + e_Z Z, \tag{2.2.1}$$

where the e_i are constants, the Pauli matrices are defined as

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \text{and} \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{2.2.2}$$

and we have used that $Y \propto XZ$. As a result, quantum errors, despite being continuous, can be corrected by a discrete set of operations. In particular, one can employ X and Z gates to correct them.
Independent errors can be modeled using quantum channels, which determine the evolution of a quantum state.

Definition 2.2.1. A **quantum channel** is a linear, completely positive, and trace-preserving (CPTP) map $T : \rho \to T(\rho)$ [23].

A fundamental property of a quantum channel is that, for any input density matrix ρ, the output $T(\rho)$ is also a valid density matrix. In this work, we will focus on two types of quantum channel: erasure and depolarizing channels.

Definition 2.2.2. A single-qubit **erasure channel** is quantum channel

$$T_{\text{erasure}} : \rho \ \to \ (1-q)\rho + q\,|\perp\rangle\langle\perp|\,,$$

where ρ is a 2×2 density matrix, $q \in [0, 1]$, and $\langle\perp|\,\rho\,|\perp\rangle = 0$.

Definition 2.2.3. A single-qubit **depolarizing channel** is quantum channel

$$T_{\text{depol}} : \rho \ \to \ (1-q)\rho + q\frac{\mathbb{I}}{2},$$

where ρ is a 2×2 density matrix, $\mathbb{I}$ is the 2×2 identity matrix, and $q \in [0, 1]$.

The previous definitions are given for a single-qubit state since we are working under the assumption of independent errors. The erasure channel erases the qubit with probability q. If this happens, a flag is issued, which allows us to identify which qubit was erased. Note that this increases the size of the output Hilbert space with respect to the input [24]. This channel is used as a simple model for physical setups in which the quantum information can be lost, e.g., optical fibers used for single-photon communication.
On the other hand, the depolarizing channel yields a completely mixed state with probability q [25]. Hence, all the quantum information is lost and no flag is issued, so it is not possible to know a priori if any qubits were affected. This constitutes a worst-case-scenario model that is well suited when we do not make any assumptions about the physical realization of the channel [24].

Correlated error models. Errors can be introduced in a correlated way, such that they are not properly described by any independent noise model. When correlated noise affects a set of t qubits, the problem becomes too complex to model, as the possibilities can be endless. Two basic examples are the undesired application of any multi-qubit gate and the random re-initialization of all t qubits. In this thesis, we do not assume any specific model for correlated errors. Instead, we will assume that any joint operation can be applied to all the qubits that are prone to correlated errors.

After a quantum state has suffered from errors, it is interesting to compare initial and final states. For that purpose, we introduce the fidelity, which measures how close is the final state to the original one.

Definition 2.2.4 (From [20])**.** The **fidelity** between two quantum states σ and ρ is defined as

$$F(\sigma, \rho) := \text{Tr}\sqrt{\sigma^{1/2}\rho\sigma^{1/2}}. \tag{2.2.3}$$

In many problems, the initial state is pure. In that case, the calculation of the fidelity can be simplified to

$$F(|\phi\rangle, \rho) = \mathrm{Tr}\sqrt{\langle\phi|\,\rho\,|\phi\rangle}. \tag{2.2.4}$$

As previously discussed, even though quantum errors are continuous, it is possible to correct them by applying a discrete set of operations. This idea is at the core of quantum error correction theory [18, 20]. We assume that the reader is familiar with this theory and therefore we only provide some definitions and theorems that are explicitly used in our work. We start by defining the concept of quantum error-correcting code (QECC):

Definition 2.2.5. An $[[n, k, d]]$ **quantum error-correcting code** $\mathcal{C}$ is a 2^k-dimensional subspace of a 2^n-dimensional Hilbert space $\mathcal{H}_n$. This code has distance d.

Note that the double square bracket notation $[[\cdot]]$ is used to distinguish quantum from classical codes, and the distance is still to be defined for quantum codes.
Let $\mathcal{H}_k$ be a 2^k-dimensional Hilbert space and $B_k = \{|\psi_i\rangle\}_{i=1,\dots,k}$ be an orthonormal basis. Let $B_{\mathcal{C}} = \{|\phi_i\rangle\}_{i=1,\dots,n}$ be the basis of $\mathcal{C}$. QECCs encode a pure k-qubit state from $\mathcal{H}_k$ into a pure n-qubit state of $\mathcal{C}$. The original k qubits are called the physical state, while the encoded ones are the *logical* state, denoted with a bar, e.g., $|\bar{0}\rangle$ and $|\bar{1}\rangle$ represent a single logical qubit[3]. As an example, take Shor's 9-qubit code [26], which encodes 1 physical qubit into a logical state composed by 9 physical qubits ($n = 9$, $k = 1$):

$$\begin{aligned} |0\rangle &\rightarrow |\bar{0}\rangle = \tfrac{1}{2\sqrt{2}}(|000\rangle + |111\rangle)^{\otimes 3} \\ |1\rangle &\rightarrow |\bar{1}\rangle = \tfrac{1}{2\sqrt{2}}(|000\rangle - |111\rangle)^{\otimes 3} \end{aligned}$$

The mapping is established by an encoding map, which is a bijection between each element of B_k and $B_{\mathcal{C}}$. The encoding enlarges the Hilbert space of our quantum state, spreading the information contained in the original qubits into correlations between the encoded qubits. This nonlocalization of the information via entanglement is key for quantum error correction, since it allows for qubits to be lost or modified without any information loss. In addition, any density matrix supported on the 2^k-dimensional subspace $\mathcal{C}$ is a valid codeword [22].

$\mathcal{C}$ can correct the set of errors $\mathcal{E} = \{E_i\}$ iff there exists a quantum operation R such that $(R \circ E_i)\,|\phi_k\rangle = |\phi_k\rangle$, $\forall E_i \in \mathcal{E}$, $|\phi_k\rangle \in \mathcal{C}$ [20]. However, it is not necessary to find R in order to assess the error-correcting properties of $\mathcal{C}$. The Knill-Laflamme conditions [27] are necessary and sufficient conditions for error correction. They state that $\mathcal{C}$ can correct the set of errors $\mathcal{E} = \{E_i\}$ iff

$$\langle\phi_k|\,E_i^\dagger E_j\,|\phi_l\rangle = C_{ij}\delta_{kl}, \tag{2.2.5}$$

[3] If $k > 1$, the logical states can be labeled with decimal or binary digits, e.g., $|\bar{0}\rangle$, $|\bar{1}\rangle$, $|\bar{2}\rangle$, and $|\bar{3}\rangle$, or $|\overline{00}\rangle$, $|\overline{01}\rangle$, $|\overline{10}\rangle$, and $|\overline{11}\rangle$, for $k = 2$.

where C_{ij} is a constant. If $\mathcal{E}$ constitutes a basis for t-error operators, i.e., if any operator introducing errors in t qubits can be written as a combination of E_i, then $\mathcal{C}$ can correct any t errors [28]. Moreover, if $\mathcal{C}$ can correct any t errors, it can alternatively correct any $2t$ qubit erasures [28]. Finally, we are ready to define the distance of a quantum code:

Definition 2.2.6. The **distance** of a quantum error-correcting code $\mathcal{C}$ is the minimum weight[4] of an error operator E_j such that

$$\langle \phi_k | E_j | \phi_l \rangle \neq C_j \delta_{kl}, \text{ for } |\phi_k\rangle \in \mathcal{C}, \tag{2.2.6}$$

where C_j is a constant.

This definition together with the Knill-Laflamme conditions imply that a quantum code with distance $d \geq 2t+1$ can correct t errors or $2t$ erasures [18].

2.2.1. Calderbank-Shor-Steane codes

A Calderbank-Shor-Steane (CSS) code [29, 30] is a special type of quantum code which is built upon two classical codes V and W that satisfy the following conditions [20]:

- V is an $[n, k_V, d_V]$ linear code that can correct up to t_V errors, i.e., $d_V \leq 2t_V + 1$.
- W is an $[n, k_W, d_W]$ linear code that can correct up to t_W errors, i.e., $d_W \leq 2t_W + 1$. Note that $W^\perp$ is an $[n, k_\perp, d_\perp]$ linear code, with $k_\perp = n - k_W$ (see Subsection 2.1.1).
- $W^\perp \subseteq V$.

These classical codes generate an $[[n, k, d]]$ CSS code, with $k = k_V + k_W - n$, whose logical states are defined as

$$|\bar{i}\rangle = \frac{1}{\sqrt{2^{k_\perp}}} \sum_{w \in W^\perp} |v_i + w\rangle, \tag{2.2.7}$$

where $v_i \in V$ and the sum is a bitwise addition modulo 2. We denote the CSS code generated by V and W as $\mathcal{C} := \mathrm{CSS}(V, W)$.
Interestingly, $\mathcal{C}$ can correct up to t_V bit flips and up to t_W phase flips [18]. As a consequence,

$$d \geq \min(d_V, d_W). \tag{2.2.8}$$

Since the formulation of CSS codes in 1996 [29, 30], their properties have been widely studied [20]. Here, we list those that are necessary for the secret sharing protocols discussed in Chapters 3, 4, and 5:

[4] The weight of an operator is the number of qubits acted on by a nontrivial Pauli matrix (X, Y, and Z) [18].

- The error correction can be divided into two separate stages: bit flip correction and phase flip correction [18].
- Certain logical operations $\bar{\Lambda}$ can be implemented by applying local operations Λ [14], i.e., $\bar{\Lambda} = \Lambda^{\otimes n}$, where n is the number of qubits. This property is called *transversality*. In particular, the CNOT gate is transversal.
- When a state $|\bar{i}\rangle \in \mathcal{C}$ is measured in the standard basis, the measurement outcome corresponds to a codeword from V. When measured in the Fourier basis, the outcome is a codeword from W [18].
- Applying a qubit-wise Fourier transform[5] $\mathcal{F}^{\otimes n}$ maps codewords from the original code $\mathrm{CSS}(V, W)$ to codewords from the code $\mathrm{CSS}(W, V)$ [14].

To conclude this section, let us mention that an encoded quantum state can be further protected against noise by performing a second encoding to each of the qubits, yielding a *concatenated* code.
The number of errors tolerated by a concatenated code grows exponentially with the total number of concatenations, but so does the number of qubits required. For example, g successive concatenations of an $[[n, 1, d]]$ code yields an $[[n^g, 1, d^g]]$ code that can correct up to $\lfloor \frac{d^g-1}{2} \rfloor$ errors [31]. If the codes used were $[[n_i, 1, d_i]]$, $i = 1, \ldots, g$, the concatenated code would be an $[[N, 1, D]]$ quantum code with $N = n_1 \cdot n_2 \cdot \ldots \cdot n_g$ and $D \geq d_1 \cdot d_2 \cdot \ldots \cdot d_g$ [18].

2.3. Approximate quantum error correction

The no-cloning theorem [21] establishes a fundamental limit to the theory of quantum error correction. Consider a single-qubit pure state $|\phi\rangle$ that is encoded using an $[[n, k, d]]$ quantum code. Let us assume that any $\lfloor \frac{n-1}{2} \rfloor$ encoded qubits could be used to reconstruct the original quantum state. In that case, we could take half of the encoded qubits to decode the state and obtain $|\phi\rangle$. However, the other half of the qubits could also be used to recover $|\phi\rangle$, obtaining a second copy of the original state. This result violates the no-cloning theorem and therefore it cannot be true. Consequently, the decoding of any encoded quantum state requires more than $\lfloor \frac{n-1}{2} \rfloor$ encoded qubits.
The QECCs presented so far are *exact* codes, meaning that they will correct any errors with probability 1, as long as there are not too many of them. From this point on, we refer to exact QECC simply as QECC.
As discussed in the previous section, a QECC that corrects any $2t$ erasures can alternatively correct t arbitrary errors. Since the no-cloning theorem establishes the maximum number of correctable erasures in $\lfloor \frac{n-1}{2} \rfloor$, the maximum number of correctable arbitrary errors is bounded from above by $\lfloor \frac{n-1}{4} \rfloor$.[6]

[5] The quantum Fourier transform maps an n-qubit state $|j\rangle$ to $\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \omega^{jk} |k\rangle$, with $\omega = e^{2\pi i/2^n}$. When applied to a single qubit, the mapping is $|0\rangle \to |+\rangle$, $|1\rangle \to |-\rangle$.

[6] Some other bounds that are more restrictive have been found. For example, the quantum Singleton one: $n - k \geq 2(d-1)$. See [18] for a general overview of these bounds.

While $t \leq \lfloor \frac{n-1}{4} \rfloor$ holds for any QECC, it is possible to design an approximate quantum error-correcting code (AQECC) that encodes k qubits into an n-qubit state and tolerates more than $\lfloor \frac{n-1}{4} \rfloor$ arbitrary errors. The price to pay is that the recovery of the encoded state has a small probability of error, which can be quantified by, for example, the fidelity between the decoded and the original states.

Definition 2.3.1. An $[[N, n; t, \varepsilon]]$ **approximate quantum error-correcting code** (AQECC) is a pair of quantum algorithms *Encode* and *Decode* such that [32]:

- *Encode* takes an n-qubit state, ρ, as input and outputs an N-qubit state denoted by $Encode(\rho)$.
- *Decode* takes an N-qubit state, σ, as input and outputs an n-qubit state denoted by $Decode(\sigma)$.

Moreover, for all initial states ρ and all operators $\mathbb{O}$ acting on at most t qubits,

$$F(\rho, \rho_{\text{out}}) \leq 1 - \varepsilon, \tag{2.3.1}$$

where $\rho_{\text{out}} = Decode(\mathbb{O}(Enc(\rho)))$.[7]

In [32], Crépeau, Gottesman, and Smith propose a construction of AQECCs based on quantum authentication schemes (QAS). Before introducing this construction, we must define what a QAS is.

2.3.1. Quantum authentication

The authentication of classical messages is a complex task that remained an open question until Wegman and Carter proposed cryptographically secure authentication schemes based on hash functions in 1981 [33]. The formulation of the problem, however, is simple: Alice wants to send a message to Bob in such a way that Bob can be sure that the message received was actually sent by Alice. The general way to do that is to send a tag attached to the message, which is used by Bob to verify the identity of the sender (see, e.g., [34] for further details).
In a quantum framework, the statement of the problem is slightly different: Alice wants to send a quantum state ρ to Bob in such a way that, if Bob accepts the received state, the fidelity between that state and ρ is close enough to 1. The way to do this is to encrypt the original quantum message combined with some tag qubits in a particular way, using a key from a set of keys $\mathcal{K}$ shared between Alice and Bob.

Definition 2.3.2. A **quantum authentication scheme** (QAS) is a pair of keyed quantum algorithms $Encrypt_k$ and $Decrypt_k$ of the form [35]

- $Encrypt_k$: $\rho \to U_k(\rho \otimes \sigma_k)U_k^\dagger$.
- $Decrypt_k$: $\rho' \to \mathrm{Tr}_T \left[(\mathbb{I}^{\otimes m} \otimes \Pi_k^{\text{acc}})\, U_k^\dagger \rho' U_k\, (\mathbb{I}^{\otimes m} \otimes \Pi_k^{\text{acc}}) \right]$
  $+ \mathrm{Tr}_{MT} \left[(\mathbb{I}^{\otimes m} \otimes \Pi_k^{\text{rej}})\, U_k^\dagger \rho' U_k\, (\mathbb{I}^{\otimes m} \otimes \Pi_k^{\text{rej}}) \right] \otimes |\perp\rangle\langle\perp|$

[7] Note that the original definition from [32] uses $\mathrm{Tr}(|\phi\rangle\langle\phi|\, \rho_{\text{out}}) \leq 1 - \varepsilon$, where $|\phi\rangle$ is the initial state. Since this is not relevant for our results, we use the fidelity for handiness.

where

- ρ is an m-qubit quantum state to be authenticated,
- k is a key from a set of keys $\mathcal{K}$,
- U_k is a key-dependent unitary,
- σ_k is a key-dependent τ-qubit tag state,
- ρ' is an $(m+\tau)$-qubit state,
- Tr_T and Tr_{MT} denote the partial trace over the tag (last τ qubits) and the message and tag (all qubits), respectively,
- Π_k^{acc} and Π_k^{rej} are orthogonal projectors onto the support of σ_k and its complement, respectively,
- and $|\perp\rangle\langle\perp|$ is an m-qubit flag state.

A QAS is defined by a keyed collection $\{(U_k, \sigma_k\}_{k\in\mathcal{K}}$ of unitaries and tag quantum states. The previous definition can be understood in the following terms. First, a quantum state ρ, the message, is encrypted by combining it with some tag qubits σ_k and applying a unitary U_k. Then, the decryption algorithm undoes the action of the unitary and checks the state of the tag, using the projectors Π_k^{acc} and Π_k^{rej}. If the tag lies in the acceptance subspace defined by Π_k^{acc}, the state is accepted and the tag is traced out. Otherwise, the whole state is traced out and the message is replaced by a flag state $|\perp\rangle\langle\perp|$ that means rejection of ρ'.

There exist several methods to perform the encryption and decryption of a QAS, depending on the functionalities that one wants to achieve, e.g., key recycling between different authenticated messages [36] and ciphertext authentication [37], in which the message is accepted only if the entire ciphertext (message and tag qubits) remains untouched. For a general overview of these types of authentication, see [35]. In this work, we will not require any of these properties. Instead, we only need our QAS to be plaintext authenticating, meaning that we accept the state if the message remains untouched. Therefore, we make use of Definition 1 from [35], which is, in turn, based on [38]:

Definition 2.3.3. A plaintext ε-authenticating quantum authentication scheme (ε-QAS) is a QAS defined by the keyed collection $\{(U_k, \sigma_k\}_{k\in\mathcal{K}}$ of unitaries and tags such that for all completely-positive maps $\mathbb{O}$ acting on the message, tag, and a side-information register, there exist completely-positive maps $\mathcal{S}_{\mathrm{acc}}$ and $\mathcal{S}_{\mathrm{rej}}$ such that $\mathcal{S}_{\mathrm{acc}} + \mathcal{S}_{\mathrm{rej}}$ is trace-preserving and

$$\frac{1}{2}\left|\left|\mathop{\mathbb{E}}_k\left[Decrypt_k \circ \mathbb{O} \circ Encrypt_k\right] - \left(\mathbb{I}^{\otimes m} \otimes \mathcal{S}_{\mathrm{acc}} + |\perp\rangle\langle\perp| \left(\mathrm{Tr}_M \otimes \mathcal{S}_{\mathrm{rej}}\right)\right)\right|\right|_\diamond \leq \varepsilon, \quad (2.3.2)$$

where $Encrypt_k$ and $Decrypt_k$ are the quantum algorithms defining the QAS, and $||\mathcal{A}||_\diamond := \sup_\rho \mathrm{Tr}\sqrt{\mathcal{A}\rho(\mathcal{A}\rho)^\dagger}$ is the diamond norm of a quantum channel $\mathcal{A}$.

This definition ensures that, for any message and any key, the decrypted state corresponds to the original message in the accept case, while it is discarded and replaced by a flag state if rejected, therefore providing plaintext authentication up to certain error ε. For other functionalities such as key recycling and ciphertext authentication, the previous definition needs to be strengthened [35].

A QAS can be used to design an AQECC, as stated before. First, an $[[n, k, d]]$ non-degenerate stabilizer code is used to encode a k-qubit state into an n-qubit state. Next, each of the n qubits is authenticated using an ε-QAS, yielding a new state with $n \cdot (1+\tau)$ qubits in total, where τ is the number of tag qubits used by the QAS. Then, this state is sent to the receiver, who needs the encryption key to retrieve the message. In order to send the key, a classical secret sharing scheme and a classical authentication subroutine are employed (see [32] for further details).
This whole construction allows the receiver to individually authenticate each of the encoded qubits and reject them if they were affected by any arbitrary error. Consequently, arbitrary errors are flagged and therefore can be treated as erasures, tolerating up to $d-1 \leq \lfloor \frac{n-1}{2} \rfloor$, and yielding an $[[n \cdot (1+\tau), k; d-1, \varepsilon_{\text{AQECC}}]]$ AQECC, with [32]

$$\varepsilon_{\text{AQECC}} = 2n^2\varepsilon. \tag{2.3.3}$$

Note that the size of the encoded state is proportional to τ, which also determines the error ε. Even though a larger τ generally reduces ε, the exact relation between these two variables depends on the inner structure of the QAS, and therefore the tradeoff between the error and the total number of encoded qubits must be analyzed for each particular case.

To conclude this subsection, let us discuss a way to build an ε-QAS that satisfies the requirements from Definition 2.3.3. To do that, we use a very specific type of unitaries, namely, the encoding maps of purity-testing codes.

Definition 2.3.4 (From [35])**.** A set of QECCs with encoding maps $\{U_k\}_{k\in\mathcal{K}}$, where $\mathcal{K}$ is a set of keys, is **purity-testing with error ε (ε-PT)** if, for any Pauli $P_l \in \mathbb{P}_{m+\tau} \backslash \{\mathbb{I}^{\otimes m+\tau}\}$,

$$\Pr_k\left[U_k^\dagger P_l U_k \in (\mathbb{P}_m \backslash \{\mathbb{I}^{\otimes m}\}) \otimes \{\mathbb{I}, Z\}^{\otimes \tau}\right] \leq \varepsilon, \tag{2.3.4}$$

for some m and τ, where $\mathbb{P}_m$ is the m-qubit Pauli group by and its elements are labeled as P_l, where l is a $2m$-bit string indicating the positions of bit flips and phase flips.

When the unitaries of an ε-PT code are used to build a QAS, m and τ are the number of message and tag qubits, respectively. Then, the interpretation of the previous definition is that, for ε-PT codes, the probability that an error is introduced in the message and not detected by the tag is at most ε.
A convenient method to construct a QAS is the encode-encrypt scheme proposed in Protocol 5.2 of [39] and formally defined in Construction 1 of [35]. This approach consists in encoding the message and the tag qubits using a code $\mathcal{C}_j$ from a family of

ε-PT codes with encoding maps $\{U_k\}_{k\in\mathcal{K}}$, where $j \in \mathcal{K}$ is the encoding key. Then, the encoded state is encrypted using one-time pad. The $Decrypt_k$ algorithm also consists of two steps: first, the one-time pad encryption is decrypted, and then the syndrome of the state in the code $\mathcal{C}_j$ is measured. If this syndrome is 0, no errors have been introduced and the state can be accepted. If the syndrome corresponds to a nonzero string, the state is rejected.
The set of codes $\{U_k\}_{k\in\mathcal{K}}$ that is used to build the QAS plays a key role in the relation between τ and ε. In [39], Barnum et al. define a ε-PT set of codes with

$$\varepsilon = \frac{2(m+\tau)}{\tau(2^\tau+1)}. \tag{2.3.5}$$

We refer to this set as the *efficient family* of ε-PT codes.

2.3.2. The trap code

In our work, we will employ a set of ε-PT codes based on the trap code proposed by Broadbent et al. [40].

Definition 2.3.5 (From [40])**.** The **trap code** is a QECC whose encoding map can be written as

$$U_k = \pi_k(E \otimes \mathbb{I}^{\otimes\tilde{n}} \otimes H^{\otimes\tilde{n}}), \tag{2.3.6}$$

where E is the encoding map of an $[[\tilde{n}, 1, \tilde{d}]]$ CSS code, k is a key specifying the permutation of $3\tilde{n}$ elements, and π_k is the permutation operator, which permutes the qubits of a state according to the permutation key k. This encoding map must be applied to $|\phi\rangle \otimes |0\rangle^{\otimes 3\tilde{n}-1}$, where $|\phi\rangle$ is the singe-qubit state to be encoded.

The trap code encodes a single-qubit state $|\phi\rangle$ into a $3\tilde{n}$-qubit state. As long as the permutation key is not revealed to an attacker that is trying to introduce errors in $|\phi\rangle$, this attacker might accidentaly introduce them in the $\tilde{n}$ tag qubits that remained as $|0\rangle$ after the encoding or in the $\tilde{n}$ tags that remained as $H|0\rangle = |+\rangle$. We call these $2\tilde{n}$ tags the *traps*. The traps at state $|0\rangle$ can be measured in the Z basis to reveal whether a bit flip has been applied to them. Similarly, traps at state $|+\rangle$ reveal phase flips. We say that a trap has been triggered whenever a $|0\rangle$ tag suffers from a bit flip or a $|+\rangle$ tag suffers from a phase flip.
A set of trap codes $\{U_k\}_{k\in\mathcal{K}}$ together with the set of keys $\mathcal{K}$ is ε-PT, with

$$\varepsilon \leq \left(\frac{2}{3}\right)^{\tilde{d}/2}, \tag{2.3.7}$$

where $\tilde{d}$ is the distance of the underlying CSS code [41]. Interestingly, it also constitutes an ε-QAS in which the unitaries U_k are the encoding maps of the trap code and the tag qubits are zeros, i.e., $\sigma_k = |0\rangle^{\otimes 3\tilde{n}-1}$ [40].

3

Secret sharing protocols

Secret sharing is a task in which a dealer wants to share a secret with n nodes in a network by giving a piece of the secret, which we call a *share*, to each of them. Moreover, this has to be done in a way that (i) the secret can be reconstructed only when a sufficient number of shares are combined together and (ii) any group of p nodes cannot retrieve any information at all about the secret. The first solutions to this problem were proposed by Shamir [42] and Blakley [43] in 1979.

Secret sharing is suitable for multi-party cryptographic protocols in which the nodes do not trust each other [13]. However, in these applications not even the dealer is necessarily trusted by the rest of the nodes. Hence, we would like our protocols to be *verifiable*, meaning that the honest nodes can verify if the dealer is honest, i.e., whether they hold shares that are consistent with some secret. For example, if all the hyperplanes in Blakley's approach were parallel, the shares would not be consistent with any secret.
The hard part of this problem is that the shares must be verified without revealing any information about the secret. Nevertheless, a solution based on oblivious transfer [44] was proposed by Chor et al. [1] in 1985. Since then, many applications have used verifiable secret sharing as a subroutine, e.g., secure multi-party computation [5, 6], byzantine agreement [8], and end-to-end auditable voting systems [9].

In this chapter, we review verifiable secret sharing protocols. In particular, we focus on protocols in which the shared secret is a single-qubit quantum state instead of a bitstring. In Section 3.1, we give a general overview of the quantum secret sharing task. Then, in Section 3.2, we explain a specific protocol in detail, namely, the verifiable hybrid secret sharing (VHSS) by Lipinska et al. [14].

3.1. From classical to quantum

Quantum secret sharing (QSS) is the quantum analogue of secret sharing. The fundamental difference is that the secret to be shared is a quantum state instead of a bitstring.
Hillery, Buzek, and Berthiaume published in 1998 the earliest proposal of a QSS scheme [45]. In 1999, Cleve, Gottesman, and Lo proposed another scheme [46], and they already pointed out the relevance of quantum error correction (QEC) in the context of secret sharing: "*Every QSS scheme is, in some sense, a quantum error-correcting code; however, some error-correcting codes are not secret sharing schemes, since they may contain sets of shares from which partial information about the secret can be obtained*". As in QEC, the information is not stored in the qubits themselves, but in the entanglement between them. In fact, quantum error-correcting codes (QECC) constitute a fundamental building block in the protocols that we discuss later.

The main limitation of the protocol from [46] is that it was not verifiable. In 2002, Crépeau, Gottesman, and Smith published their *verifiable quantum secret sharing* (VQSS) scheme [47, 13]. In this protocol, a quantum state is encoded using a concatenated Calderbank-Shor-Steane (CSS) code (see Subsection 2.2.1), and the verification stage consists in checking that the encoded qubits have not suffered from too many errors that prevent them to be decoded into a valid state.
In the protocol from [13], the number of qubits that each node must control simultaneously grows with the size of the network. Specifically, the node workspace size must be at least $\Omega(r^2 n \log n)$, where n is the total number of nodes and r is a security parameter of the protocol. This prevents its implementation in early quantum networks, in which nodes cannot control too many qubits at the same time.

Research has been done towards reducing the number of quantum resources in QSS [48]. An approach to achieve this in the VQSS task is the one proposed by Lipinska et al. [14]. They designed a protocol that combines the VQSS from [13] with classical encryption and classical secret sharing. We use this protocol as the starting point for most of the work developed throughout this thesis. Consequently, we extensively discuss it in the following section.

3.2. Verifiable hybrid secret sharing (VHSS)

The *verifiable hybrid secret sharing* (VHSS) protocol proposed by Lipinska et al. [14] combines a classical encryption and classical secret sharing with a quantum subroutine based on the VQSS protocol from [13].
As in the VQSS, the goal is to share a quantum state among several nodes in a way that (*i*) it can be reconstructed only when a sufficient number of shares are combined together, (*ii*) small groups of nodes cannot retrieve any information about the shared state, and (*iii*) it is possible to verify that honest parties hold consistent shares that can be decoded into a valid quantum state. The protocol runs on a network with n_c

nodes that can process classical data perfectly and n nodes that can hold quantum shares and perform quantum operations on them. Note that a single node could simultaneously work with classical and quantum information. We assume that nodes which can hold quantum shares can also hold classical bits, and therefore $n_c \geq n$.
The VHSS protocol is summarized as follows. First, the dealer D encrypts the quantum secret $|\psi\rangle$ using quantum one-time pad and a classical secret key s. Then, the classical key is distributed between n_c classical nodes, following a verifiable classical secret sharing (VCSS) scheme. In parallel, the encrypted quantum state is encoded into n qubits using a CSS code $\mathcal{C}$, which are distributed among the quantum nodes. These nodes perform a second encoding and redistribute the shares among themselves. After that, the verification stage takes place. Finally, if the protocol does not abort during verification, all quantum and classical shares are sent to the reconstructing node R, who reconstructs the classical key and then decodes and decrypts the quantum state.
The main advantages of this protocol with respect to the VQSS are the number of quantum resources required to run the protocol and the degree of security:

- In the VHSS, each of the n quantum nodes is only required to have a workspace of at most $3n$ simultaneous qubits, while the VQSS from [13] requires $\Omega(r^2 n \log n)$ (see [14] for further details about this comparison).
- The classical encryption ensures that the VHSS scheme achieves maximum secrecy, i.e., no group of $p \leq \lfloor \frac{n-1}{2} \rfloor$ nodes can retrieve any information about the secret.

3.2.1. The protocol

The original protocol is described step by step in [14]. In this subsection, we discuss each of the steps in order to provide a template over which we propose some modifications in the next chapters. The steps are divided into three stages: sharing, verification, and reconstruction. In addition, we assume that the nodes have access to an authenticated classical broadcast channel and a public randomness source, and that each pair of nodes is connected by an authenticated classical channel and a quantum channel (the latter is only required for each pair of quantum nodes).

SHARING. In this first stage, a pure single-qubit state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ is encrypted, encoded into n shares, and distributed among the nodes.

1. (Encryption) The dealer D performs quantum one-time pad on the secret state $|\psi\rangle$ using a classical secret key s to obtain the state $\Phi^{0,0}$.

The classical key is a two-bit string $s = xz$. After encryption, the quantum state is described by a maximally mixed state from the point of view of the nodes since they do not know s, i.e.,

$$\Phi^{0,0} = \frac{1}{4} \sum_{xz=\{0,1\}^2} X^x Z^z |\psi\rangle\langle\psi| Z^z X^x.$$

2. (VCSS) D distributes the classical key s among the n_c nodes using a verifiable classical secret sharing (VCSS) scheme.

In this work, we follow the approach of [14] and do not discuss the details of the VCSS protocol. Instead, we just assume that this subroutine does not leak any information about s to any set of at most p_c nodes, except with probability exponentially small in a security parameter r.

3. (Branch encoding and distribution) D encodes $\Phi^{0,0}$ into an n-qubit state $\Phi^{0,0}_{[1,n]}$ using the code $\mathcal{C}$. Then, it sends qubit $\Phi^{0,0}_i$ to node i, $\forall i = 1, \dots, n$.

The quantum code employed here is an $[[n, 1, d]]$ CSS code, $\mathcal{C} = \mathrm{CSS}(V, W)$, built upon two classical linear codes V and W that fulfill the conditions stated in Subsection 2.2.1: V is an $[n, k_V, d_V]$ linear code, W is an $[n, k_W, d_W]$ linear code, $W^\perp \subseteq V$, $k_V + k_W - n = 1$, and $d \geq \min(d_V, d_W)$. This CSS code can correct up to $t \leq \lfloor \frac{d-1}{2} \rfloor$ arbitrary errors or up to $u \leq d - 1$ erasures.
Note that the dealer can only introduce errors in the secret shares before sending $\Phi^{0,0}_{[1,n]}$ to the nodes. Hence, a dishonest dealer cannot tamper with the secret after step 3.

4. (Leaf encoding and distribution) Each node i encodes its share $\Phi^{0,0}_i$ into an n-qubit state $\Phi^{0,0}_{i_{[1,n]}}$ using the code $\mathcal{C}$. Then, it sends qubit $\Phi^{0,0}_{ij}$ to node j, $\forall j = 1, \dots, n$.

Steps 3 and 4 of the sharing stage are depicted in Fig. 3.1, which shows the distribution of qubits after each action. Due to the similarity with a tree, we refer to $\Phi^{0,0}_{[1,n]}$ and $\Phi^{0,0}_{i_{[1,n]}}$ as the branch and the leaf shares, respectively.
After step 4, no more errors can be introduced directly in the branch shares $\Phi^{0,0}_{[1,n]}$ since each qubit $\Phi^{0,0}_i$ has been further encoded into $\Phi^{0,0}_{i_{[1,n]}}$.

Verification. After the sharing stage, the nodes want to verify that they are holding consistent shares that later will allow for reconstruction of a quantum state. It is crucial to perform this check without retrieving any information about the secret, as we want it to remain unknown for the nodes throughout the protocol. Here, we only discuss the verification of the quantum shares, although the nodes also have to verify the classical shares of the VCSS subroutine [3] in parallel.

As discussed in Section 2.2, arbitrary errors can be described as a combination of bit flips and phase flips. Hence, in order to detect arbitrary errors in their shares, the nodes employ r ancillary qubits initialized in the state $|\bar{+}\rangle = \sum_{v \in V} |v\rangle$ to find

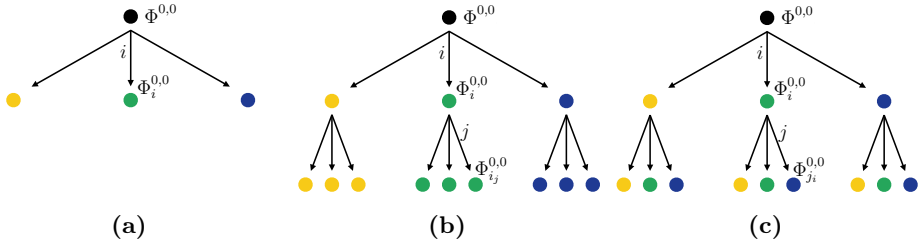


Figure 3.1. Sharing steps of the VHSS protocol in a toy network with $n = 3$ nodes (note that this is just a graphical example, since no CSS code exists for $n = 3$). **(a)** The dealer encodes the secret and shares the n-qubit state with the nodes. **(b)** The nodes encode their qubits and hold all the new shares. **(c)** Nodes redistribute their shares.

bit flip errors (Z basis), and r ancillas initialized in $|\bar{0}\rangle = \sum_{w\in W^\perp} |w\rangle$ to find phase flips (X basis).

for $m = 1, \ldots, r$:

5. (Preparation and distribution - Z basis) The dealer D prepares an n-qubit ancillary state $|\bar{+}\rangle_{[1,n]}^{0,m} = \sum_{v\in V} |v\rangle$. Then, it sends qubit $|\bar{+}\rangle_i^{0,m}$ to node i, $\forall i = 1, \ldots, n$.

6. (Leaf encoding and distribution - Z basis) Each node i encodes its share $|\bar{+}\rangle_i^{0,m}$ into an n-qubit state $|\bar{+}\rangle_{i_{[1,n]}}^{0,m}$ using $\mathcal{C}$. Then, it sends qubit $|\bar{+}\rangle_{i_j}^{0,m}$ to node j, $\forall j = 1, \ldots, n$.

7. (CNOTs - Z basis) A public random bit $b_{0,m}$ is generated. Then, each node i applies the following operation to their shares:

$$\mathrm{CNOT}^{b_{0,m}}(\Phi_{j_i}^{0,0}, |\bar{+}\rangle_{j_i}^{0,m}),\ \forall j = 1, \ldots, n, \tag{3.2.1}$$

   i.e., a CNOT is applied if $b_{0,m} = 1$.

8. (Measurements - Z basis) Each node i measures the n ancillary states it is holding (systems indexed with m and $l = 0$) in the Z basis and broadcasts the measurement outcomes. All measured qubits are discarded.

end

An ancillary tree of states is generated in steps 5-6 in the same way as it was done with the secret in steps 3-4. The goal of step 7 is to propagate bit flip errors from $\Phi_{j_i}^{0,0}$ to $|\bar{+}\rangle_{j_i}^{0,m}$ using CNOTs, so that they can be detected later in steps 18-19. After the leaf encoding (step 4), each n-qubit state $\Phi_{j_{[1,n]}}^{0,0}$ is a superposition of 2^{k_V} states of the form $|v_i\rangle$, $v_i \in V$. Similarly, $|\bar{+}\rangle_{j_{[1,n]}}$ is also a superposition of states of the

form $|v_k^+\rangle$, $v_k^+ \in V$. If there were some bit flips[1] on v_i or v_k^+, these bit strings would no longer be in V.
The CNOTs in step 7 can be applied separately to each share since it is a transversal operation for encoded states of a CSS code [14]. Moreover, it is also a linear operation and therefore it is possible to discuss how bit flips propagate from the secret shares to the ancillas by applying step 7 to a single combination of $|v_i\rangle$ and $|v_k^+\rangle$. After the CNOT, the resulting state can either be in V or not, depending on which qubits were affected by bit flips. This can be determined using the addition rules from Eqs. (2.1.10). All possible scenarios are summarized in Table 3.1, where checkmarks denote that the word is in V and crosses denote that it is not. When only one of the qubits has been affected by a bit flip, the final ancillary state captures this information. However, when both states have suffered from errors, the CNOT could compensate them and yield a final state in V.

Table 3.1. Possible scenarios in step 7 of the protocol (see main text). A checkmark means that the word is in V, a cross means that it is not, and a question mark means that it cannot be predetermined in a general case. Note that the sum is a bitwise addition modulo 2.

v_i^Φ	v_k^+	$v_i^\Phi + v_k^+$
✓	✓	✓
✓	×	×
×	✓	×
×	×	?

In order to prevent cheaters from maliciously introducing errors in the secret and in the ancillas that cancel out, the CNOT is applied at random. Moreover, the random bit $b_{l,m}$ is drawn after distributing all ancillas labeled with l, m, so cheaters do not know a priori to which shares they should apply the CNOT. By doing this, errors introduced in the ancillas are detected at least 50% of the time. Consequently, cheaters will only be able to pass verification holding a share that has suffered from bit flips with probability at most $2^{-\Omega(r)}$, where r is the number of rounds. For a more detailed discussion, see Appendix A of [14].
Finally, step 8 measures the ancillas to find the bit flip errors.

for $l = 1, \ldots, r$:

9. (Preparation and distribution - X basis) The dealer D prepares an n-qubit ancillary state $|\bar{0}\rangle_{[1,n]}^{l,0} = \sum_{w \in W^\perp} |w\rangle$. Then, it sends qubit $|\bar{0}\rangle_i^{l,0}$ to node i, $\forall i = 1, \ldots, n$.

[1] Recall that we assume at most t cheaters and therefore at most t bit flips.

10. (Leaf encoding and distribution - X basis) Each node i encodes its share $|\bar{0}\rangle_i^{l,0}$ into an n-qubit state $|\bar{0}\rangle_{i_{[1,n]}}^{l,0}$ using $\mathcal{C}$. Then, it sends qubit $|\bar{0}\rangle_{i_j}^{l,0}$ to node j, $\forall j = 1, \ldots, n$.

for $m = 1, \ldots, r$:

11. (Preparation and distribution - $l, m \neq 0$) The dealer D prepares an n-qubit ancillary state $|\bar{0}\rangle_{[1,n]}^{l,m} = \sum_{w \in W^\perp} |w\rangle$. Then, it sends qubit $|\bar{0}\rangle_i^{l,m}$ to node i, $\forall i = 1, \ldots, n$.

12. (Leaf encoding and distribution - $l, m \neq 0$) Each node i encodes its share $|\bar{0}\rangle_i^{l,m}$ into an n-qubit state $|\bar{0}\rangle_{i_{[1,n]}}^{l,m}$ using $\mathcal{C}$. Then, it sends qubit $|\bar{0}\rangle_{i_j}^{l,m}$ to node j, $\forall j = 1, \ldots, n$.

13. (CNOTs - $l, m \neq 0$) A public random bit $b_{l,m}$ is generated. Then, each node i applies the following operation to their shares:

$$\text{CNOT}^{b_{l,m}}(|\bar{0}\rangle_{j_i}^{l,0}, |\bar{0}\rangle_{j_i}^{l,m}), \ \forall j = 1, \ldots, n, \tag{3.2.2}$$

i.e., a CNOT is applied if $b_{l,m} = 1$.

14. (Measurements - $l, m \neq 0$) Each node i measures its n qubits indexed with l, m in the Z basis and broadcasts the measurement outcomes. All measured qubits are discarded.

15. (Fourier transform) Each node i applies the Fourier transform $\mathcal{F}$ to its remaining shares, obtaining $\Phi_{j_i}^{\mathcal{F},0,0}$ and $|\bar{0}\rangle_{j_i}^{\mathcal{F},l,0}$, $j = 0, \ldots, n$.

16. (CNOTs - X basis) A public random bit $b_{l,0}$ is generated. Then, each node i applies the following operation to their shares:

$$\text{CNOT}^{b_{l,0}}(\Phi_{j_i}^{\mathcal{F},0,0}, |\bar{0}\rangle_{j_i}^{\mathcal{F},l,0}), \ \forall j = 1, \ldots, n. \tag{3.2.3}$$

i.e., a CNOT is applied if $b_{l,0} = 1$.

17. (Measurements - X basis) Each node i measures the n ancillary states it is holding (systems indexed with l and $m = 0$) in the Z basis and broadcasts the measurement outcomes.

end

Steps 9-17 have a similar goal as steps 5-8. In this case, the states $|\bar{0}\rangle^{l,0}$ are used to find phase flips in the secret shares. To do that, nodes apply a Fourier transform to the states and look for bit flips instead. Nevertheless, it is also necessary to check that the states $|\bar{0}\rangle_{i_j}^{l,0}$ have not suffered from bit flips, so the nodes perform

r additional checks on each of them. If we did not perform these checks, bit flips on $|\bar{0}\rangle_{i_j}^{l,0}$ would transform into phase flips on $|\bar{0}\rangle_{i_j}^{\mathcal{F},l,0}$ that would propagate to $\Phi_{i_j}^{\mathcal{F},0,0}$ with the application of the CNOT in step 16. These phase flips would become again bit flips on $\Phi_{i_j}^{0,0}$ that would not be detected anymore. Therefore, if we do not ensure that the ancillas $|\bar{0}\rangle_{i_j}^{l,0}$ have not suffered from bit flips, we could introduce additional errors in the secret shares.

From steps 9-17, we see that each node i is required to hold only $3n$ qubits simultaneously at each round (l, m): n secret shares $\Phi_{j_i}^{0,0}$ $(j = 1, \ldots, n)$, n ancillary shares $|\bar{0}\rangle_{j_i}^{l,0}$, and n extra ancillas $|\bar{0}\rangle_{j_i}^{l,m}$. Table 3.2 shows all the $(r+1)^2$ quantum states that are used in a single leaf j_i of the protocol.

Table 3.2. Sketch of the states used in the verification step at leaf j_i. The total number of states per leaf is $(r+1)^2$.

$\Phi_{j_i}^{0,0}$	$\|\bar{+}\rangle_{j_i}^{0,1}$	$\ldots$	$\|\bar{+}\rangle_{j_i}^{0,m}$	$\ldots$	$\|\bar{+}\rangle_{j_i}^{0,r}$
$\|\bar{0}\rangle_{j_i}^{1,0}$	$\|\bar{0}\rangle_{j_i}^{1,1}$	$\ldots$	$\|\bar{0}\rangle_{j_i}^{1,m}$	$\ldots$	$\|\bar{0}\rangle_{j_i}^{1,r}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\|\bar{0}\rangle_{j_i}^{l,0}$	$\|\bar{0}\rangle_{j_i}^{l,1}$	$\ldots$	$\|\bar{0}\rangle_{j_i}^{l,m}$	$\ldots$	$\|\bar{0}\rangle_{j_i}^{l,r}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\|\bar{0}\rangle_{j_i}^{r,0}$	$\|\bar{0}\rangle_{j_i}^{r,1}$	$\ldots$	$\|\bar{0}\rangle_{j_i}^{r,m}$	$\ldots$	$\|\bar{0}\rangle_{j_i}^{r,r}$

18. (Decoding leaves - Z basis) Broadcasted values $v_{l,m,i,j}$ in steps 8 and 14 yield words $\mathbf{v}_{l,m,i} = v_{l,m,i,1} v_{l,m,i,2} \ldots v_{l,m,i,n}$ that should correspond to codewords from code V. For each word, the nodes:
    (a) Classically decode the word to obtain a single bit $a_{l,m,i}$.
    (b) If a bit flip occurred at position k of $\mathbf{v}_{l,m,i}$, include this position in a set B_i. We denote this as $B_i \leftarrow B_i \cup \{k\}$. If $|B_i| > t$, then $B \leftarrow B \cup \{i\}$.
19. (Decoding branches - Z basis) Decoded values $a_{l,m,i}$ form words $\mathbf{a}_{l,m}$, which should also correspond to codewords from V. If an error occurred in position k of $\mathbf{a}_{l,m}$, then $B \leftarrow B \cup \{k\}$.
20. (Decoding leaves - X basis) Broadcasted values $w_{l,0,i,j}$ in step 17 yield words $\mathbf{w}_{l,0,i}$ that should correspond to codewords from W. For each word, the nodes:
    (a) Classically decode the word to obtain a single bit $a_{l,0,i}$.
    (b) If a bit flip occurred at position k of $\mathbf{w}_{l,0,i}$, include this position in B_i, i.e., $B_i \leftarrow B_i \cup \{k\}$. If $|B_i| > t$, then $B \leftarrow B \cup \{i\}$.

21. (Decoding branches - X basis) Decoded values $a_{l,0,i}$ form words $\mathbf{a}_{l,0}$, which should also correspond to codewords from W. If an error occurred in position k of $\mathbf{a}_{l,0}$, then $B \leftarrow B \cup \{k\}$.

22. (Abort condition) If $|B| > t$, reject the dealer and abort the protocol. Otherwise, continue.

23. (Inverse Fourier transform) Nodes apply an inverse Fourier transform to revert the transform from step 15. Each node holds again $\Phi^{0,0}_{[1,n]_j}$.

Let us define C_D as the set of nodes whose shares have suffered from errors introduced by a *dishonest dealer*. If the dealer is honest, $|C_D| = 0$.
We also define a set C_{cheat} of *cheating nodes* which do not follow the protocol honestly and can introduce errors in their shares. They can do that by performing any joint operation on their qubits. Consequently, they can introduce correlated errors. The maximum number of errors that the CSS code $\mathcal{C}$ can correct is t, and therefore we assume that there are at most t cheaters.
Moreover, we define another set C_C which includes:

- Nodes holding a branch share ($\Phi^{0,0}_i$, $|\bar{+}\rangle^{0,m}_i$, or $|\bar{0}\rangle^{l,m}_i$) that contains errors not introduced by the dealer.

- Nodes holding a branch share ($\Phi^{0,0}_i$, $|\bar{+}\rangle^{0,m}_i$, or $|\bar{0}\rangle^{l,m}_i$) that cannot be reconstructed due to an excess of errors in the leaves ($\Phi^{0,0}_{i_{[1,n]}}$, $|\bar{+}\rangle^{0,m}_{i_{[1,n]}}$, or $|\bar{0}\rangle^{l,m}_{i_{[1,n]}}$).

In an ideal quantum network in which qubits do not experience any noise, branch shares only suffer from errors introduced by the dealer or by the nodes that hold them. Hence, the first type of nodes in C_C are cheaters. Moreover, a branch share of an honest node i can always be reconstructed, since cheaters cannot introduce errors on more than t shares of $\Phi^{0,0}_{i_{[1,n]}}$, $|\bar{+}\rangle^{0,m}_{i_{[1,n]}}$, or $|\bar{0}\rangle^{l,m}_{i_{[1,n]}}$. Therefore, the second type of nodes in C_C are cheaters that tampered with their own leaf shares. Hence, $C_C \subseteq C_{\text{cheat}}$ and $|C_C| \leq |C_{\text{cheat}}| \leq t$. Note also that the size of C_C can grow throughout the protocol.

The set $C = C_D \cup C_C$ corresponds to the set of all nodes holding branch shares that have suffered from errors or that cannot be reconstructed due to an excess of errors in the leaves. Hence, the branch shares held by nodes $i \in C$ cannot be used for reconstruction.
In steps 18-21 of the verification, the nodes perform a classical decoding of the measurement outcomes (see Figure 3.2) and create the set B of *apparent cheaters*. The protocol identifies $B = C$ with probability exponentially close to 1 in a security parameter r, and it aborts if the size of B is too large, meaning that there had been too many errors that prevent the reconstruction of the secret (see [14] for a formal proof).

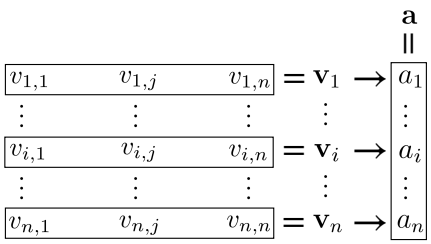


Figure 3.2. Sketch of the classical decoding in steps 18-21. The sketch corresponds to an arbitrary combination of l and m. For clarity, we have omitted these indices in the variables, e.g., $\mathbf{v}_i$ corresponds to $\mathbf{v}_{l,m,i}$. The measurement outcomes are represented by $v_{i,j}$, which form words $\mathbf{v}_i$ that are decoded as a_i. Finally, all bits a_i form the word $\mathbf{a}$.

RECONSTRUCTION. After passing verification, the nodes can attempt to recover the original quantum state shared by the dealer.

24. (Send to R) All nodes send their quantum and classical shares to the reconstructing node R.
25. (Reconstruct s) R reconstructs the classical key s following the VCSS scheme.
26. (Detect cheaters) for $i = 1, \ldots, n$:
    - 26.1. R runs an error-detecting circuit for code $\mathcal{C}$ on $\Phi^{0,0}_{i_{[1,n]}}$. For each error detected at position j: $B_i \leftarrow B_i \cup \{j\}$.
    - 26.2. If $|B_i| \leq t$, R corrects the errors in $\Phi^{0,0}_{i_{[1,n]}}$ and decodes it to obtain $\Phi^{0,0}_i$. Otherwise, $B \leftarrow B \cup \{i\}$.
27. (Reconstruct $\Phi^{0,0}$) R takes $n - 2t$ shares $\Phi^{0,0}_i$, $i \notin B$, at random and applies an erasure-recovery circuit to obtain $\Phi^{0,0}$.
28. (Decrypt) R decrypts $\Phi^{0,0}$ using s to obtain the original quantum state $|\psi\rangle$.

The reconstruction of the secret is performed by a reconstructing node R, which is assumed to act honestly—otherwise, we cannot give any guarantees about the completeness of the protocol.
To reconstruct the secret, R first completes the VCSS subroutine to obtain the classical key s. Then, it detects and corrects errors that were introduced by cheaters after verification. Reconstruction will be possible as long as the number of arbitrary errors in $\Phi^{0,0}_{[1,n]}$ is not larger than $2t \leq d - 1$, since $\mathcal{C}$ tolerates up to $d - 1$ erasures. Note however that the abort condition in step 22 of the verification stage is $|B| > t$ instead of $|B| > 2t$. This way, even in the case in which all errors are introduced by a dishonest dealer, the cheaters cannot make reconstruction to fail after a successful verification ($|B| \leq t$), since they can introduce at most t extra errors, yielding

$|B| \leq 2t$.
Finally, if there are at least $n - 2t$ branches $\Phi_i^{0,0}$, $i \notin B$, that have not suffered from errors, R decodes and decrypts the quantum secret.

4

Quantum noise in the VHSS protocol

In this chapter, we analyze how noisy communication channels can affect the performance of the VHSS protocol. We consider a situation in which each qubit suffers from noise when traveling from node to node. This noise is independent for each qubit.

Classical communication is more advanced and reliable than quantum. Hence, we assume that none of the bits involved in the protocol suffers from noise. This assumption is commonly adopted in quantum cryptography problems, since the levels of classical noise are negligible compared to the levels of quantum noise.

First, we consider a network in which qubits can be erased, e.g., a network in which qubits are transmitted in the form of single photons. We start by assuming that only the quantum secret shares are affected by noise, while the ancillas remain noiseless. In this scenario, the security statements made by Lipinska et al. [14] do not hold anymore, e.g., there is a non-negligible probability to abort the protocol in the verification phase even when the dealer is honest. In Section 4.1, we propose some modifications to the original protocol that should be adopted when the network is described by this noise model. We also analyze the probability of passing verification and use it to reformulate the security statements.

One might argue that most quantum networks running VHSS would distribute secret and ancilla shares over the same channels, and therefore they should be subject to the same noise model. For this reason, we propose two additional noise models: one in which channels are broken with certain probability and lose all the qubits, and another one in which all qubits in the protocol can be independently lost with certain probability. We then generalize the approach from the first noise model to these two, in Section 4.2.

In Section 4.3, we discuss how the depolarizing noise on each individual qubit affects the protocol. We show that our results regarding erasures can be adapted in the depolarizing noise models.

Finally, in Section 4.4 we propose an alternative sharing stage that improves the robustness of the protocol against all the previous types of noise.

4.1. Erasures in the secret

Let us assume that every time a qubit is sent from one node to another, there is a probability q that it is lost. Branch shares can be lost when the dealer distributes its qubits, while leaf shares can suffer from erasures when nodes redistribute their shares and when they send them to the reconstructor. In this section, we assume that only the secret shares can be lost. We can picture this as a quantum network whose channels are noisy during the sharing and the reconstruction stages, while, for some reason, noise drops to negligible levels ($q = 0$) during verification. This is not a common scenario, but allows us to get some insight into the problem before including erasures of the ancillas in the model (Section 4.2).

Definition 4.1.1. We define the **'Erasures on the Secret' (ES) model** as a noise model for quantum networks running the VHSS protocol in which: (*i*) each qubit has an independent probability q of being erased when traveling from node to node and (*ii*) only shares of the encoded secret can be erased, i.e., ancillary qubits do not suffer from noise.

PROTOCOL MODIFICATIONS. When the noise in the quantum network can be described by the ES model, it is convenient to perform some modifications over the original VHSS protocol from [14].
First, recall that the goal of the verification phase is to create a set of apparent cheaters B such that $B = C$, where C is the set of nodes whose branch shares cannot be used to reconstruct the secret. We keep the definitions of C_D, C_{cheat}, and C_C.[1] As in the original scheme, $C = C_D \cup C_C$. However, errors can be now randomly introduced by the communication channels and C_C can contain honest nodes, as opposed to the case of the ideal network in which only the dealer and the cheaters could tamper with the qubits and C_C only contained cheating nodes (see Subsection 3.2.1).
In contrast to the original scheme, which tolerates up to t cheaters, where t is the maximum number of errors tolerated by the CSS code, let us assume that there are at most t_c cheaters, with $t_c \leq t$. The motivation for this is to increase the probability that an honest dealer passes verification by decreasing the maximum number of tolerated cheaters. This assumption requires some steps of the protocol to be modified. In particular, the condition for passing verification must be $|B| \leq 2t - t_c$. This is a way to ensure that, even in a worst case scenario in which all t_c cheaters introduce as many errors as they can after verification, the size of B will not exceed $2t$, which is the maximum allowed for reconstruction. Note that the dealer can introduce up to $2t - t_c$ arbitrary errors and still pass verification.

[1] C_D: set of nodes whose branch shares carry errors introduced by the dealer.
C_{cheat}: set of cheating nodes.
C_C: set of nodes whose branch shares carry errors not introduced by the dealer or cannot be reconstructed due to an excess of errors in the leaves.

On the other hand, we can also take advantage of the fact that erasures are flagged. Honest nodes will publicly announce erasures. Then, if there were too many of them, nodes can abort the protocol before verification, if they happened in the sharing phase, or at the beginning of reconstruction, if they happened when sending all qubits to R. This way, they can avoid performing a large number of quantum operations as they know beforehand that the secret cannot be reconstructed.
Cheaters can decide to not report erasures. In that case, we could have more erasures than allowed and still not abort the protocol before verification or reconstruction. However, this is not problematic, since these erasures will act as arbitrary errors introduced by the cheaters and will be detected in the verification stage or in step 26 of reconstruction.
In addition, throughout the protocol, nodes will replace erased qubits by arbitrarily initialized states in order treat erasures as arbitrary errors. The reason to do this is that we would like to keep as many steps as possible from the original VHSS without any modification. By treating erasures as arbitrary errors, the reconstructor can run the same error-correcting circuits as in the original reconstruction stage. If we did not follow this approach, a careful analysis of the reconstruction stage would be required, as the quantum circuits employed for reconstruction might be different depending on which qubits were erased.

We include the previous considerations in the protocol with the following modifications for steps 3, 4, 22, and 24:

VHSS-ES PROTOCOL. VHSS protocol for networks described by an ES noise model. Here, we only list the steps that are different with respect to the original scheme by Lipinska et al. [14] (see Subsection 3.2.1).

⋮

3_{ES}. (Branch encoding and distribution)

- $3_{\text{ES}}.1$. D encodes $\Phi^{0,0}$ into an n-qubit state $\Phi^{0,0}_{[1,n]}$ using the code $\mathcal{C}$. Then, it sends qubit $\Phi^{0,0}_i$ to node i, $\forall i = 1, \dots, n$.
- $3_{\text{ES}}.2$. Erasures are publicly announced by the nodes. If an erasure happens on $\Phi^{0,0}_i$, node i replaces its share by an arbitrary qubit and $B \leftarrow B \cup \{i\}$.
- $3_{\text{ES}}.3$. If $|B| > 2t - t_{\text{c}}$, abort the protocol. Otherwise, continue.

⋮

4_{ES}. (Leaf encoding and distribution)

$4_{\text{ES}}.1$. Each node i encodes its share $\Phi_i^{0,0}$ into an n-qubit state $\Phi_{i_{[1,n]}}^{0,0}$ using the code $\mathcal{C}$. Then, it sends qubit $\Phi_{i_j}^{0,0}$ to node j, $\forall j = 1, \ldots, n$.

$4_{\text{ES}}.2$. Erasures are publicly announced by the nodes. If an erasure happens on $\Phi_{i_j}^{0,0}$, node j replaces this share by an arbitrary qubit and $B_i \leftarrow B_i \cup \{j\}$.

$4_{\text{ES}}.3$. If $|B_i| > t$, $B \leftarrow B \cup \{i\}$, $\forall i = 1, \ldots, n$.

$4_{\text{ES}}.4$. If $|B| > 2t - t_{\text{c}}$, abort the protocol. Otherwise, continue.

⋮

22_{ES}. (Abort condition) If $|B| > 2t - t_{\text{c}}$, reject the dealer and abort the protocol. Otherwise, continue.

⋮

24_{ES}. (Send to R)

$24_{\text{ES}}.1$. All nodes send their quantum and classical shares to the reconstructing node R.

$24_{\text{ES}}.2$. If an erasure happens on $\Phi_{i_j}^{0,0}$, R replaces this share by an arbitrary qubit and $B_i \leftarrow B_i \cup \{j\}$.

$24_{\text{ES}}.3$. If $|B_i| > t$, $B \leftarrow B \cup \{i\}$, $\forall i = 1, \ldots, n$.

$24_{\text{ES}}.4$. If $|B| > 2t$, abort the protocol. Otherwise, continue.

⋮

Security statements and design function. After being adapted to the ES model, the protocol must still be secure, sound, and complete. Here, we show that the VHSS-ES protocol meets these three security requirements. First, the secrecy of the protocol can be stated as follows:

Theorem 4.1.1 (Secrecy)**.** *In the verifiable hybrid secret sharing protocol for networks under the ES noise model,* VHSS-ES protocol*, when D is honest and there are at most t_{c} active cheaters in the verification phase, no group of at most p_{c} nodes learns nothing about D's secret state throughout the protocol except with probability exponentially small in the security parameter r, where p_{c} is the secrecy of the underlying classical scheme.*

Proof. The proof given in [14] still applies to VHSS-ES PROTOCOL. It is based on the security of the underlying VCSS scheme, which ensures that no group of at most p_c nodes can reconstruct the classical encryption key s that was used to apply a one-time pad on the quantum secret. This implies that the decoded quantum secret remains as a maximally mixed state if s is not known. □

Next, we formulate the soundness statement. The intuitive interpretation is that, if there are too many errors, the protocol will abort, except with probability $2^{-\Omega(r)}$. Let us state this formally:

Theorem 4.1.2 (Soundness)**.** *In the verifiable hybrid secret sharing protocol for networks under the ES noise model,* VHSS-ES PROTOCOL*, either the honest parties hold a consistently encoded secret or the protocol aborts, with probability at least* $1 - 2^{-\Omega(r)}$*. Moreover, after passing verification and sending all qubits to an honest* R*, either the shares are still consistent or the protocol aborts, with probability at least* $1 - 2^{-\Omega(r)}$*.*

Proof. Regarding the first part of the statement, we proceed in a similar way as Lipinska et al. did to prove the soundness statement of the original VHSS [14]. The statement can be reformulated as: *the probability that verification fails to identify* $B = C$ *is at most* $2^{-\Omega(r)}$. Failure can happen either in the underlying VCSS scheme or in the VQSS:

$$\Pr(\text{VCSS fails} \vee \text{VQSS fails}) \leq \Pr(\text{VCSS fails}) + \Pr(\text{VQSS fails}). \tag{4.1.1}$$

Regarding the classical subroutine, we also consider the protocol from [3], whose probability of failure is exponentially small on a security parameter r_{VCSS}. We choose $r_{\text{VCSS}} = r$ to obtain $\Pr(\text{VCSS fails}) \leq 2^{-\Omega(r)}$.
Lipinska et al. showed in Appendix A of [14] that $\Pr(\text{VQSS fails}) \leq 2^{-\Omega(r)}$. The proof consists in showing that the state held by the nodes after verification is close to a codeword in $\mathcal{C}$ with at most t errors on branch shares held by nodes in C, or that the protocol aborts. For the VHSS-ES PROTOCOL, the proof is the same except for a minor subtlety: bitstrings obtained from the measurement of the ancillas in the modified protocol can have up to $2t - t_c$ errors. This has no influence on the proof from [14] since all the measurement outcomes from steps 8 and 14 still correspond bitstrings that belong to a 2-GOOD$_V$ tree (similarly, measurement outcomes from step 17 correspond to bitstrings from a 2-GOOD$_W$ tree). Informally, this means that all bitstrings correspond to codewords from the code V (or W) up to $2t - t_c$ errors. For a formal definition of 2-GOOD trees, see Definition 1 from [13] and Definition 4, Proposition 2.10, and Lemma 2.11 from [47]. Finally, combining the results for the VCSS and the VQSS, we obtain

$$\Pr(\text{VCSS fails} \vee \text{VQSS fails}) \leq 2^{-\Omega(r)}. \tag{4.1.2}$$

Let us now show the second part of the statement: after sending all shares to R, either they remain consistent or the protocol aborts, except with a exponentially small probability. Right after passing verification, there is at least a probability of

$1 - 2^{-\Omega(r)}$ that the shares are consistent, as stated in the first part of the Theorem. In other words, nodes successfully identified $B = C$ with probability $1 - 2^{-\Omega(r)}$. Qubits can be erased while traveling to R and cheaters could have faked to be honest during verification and may introduce errors right before sending their qubits to R. Both events could enlarge C. To prove the second part of the statement, we must show that, if the nodes were holding consistent shares, the set B is always correctly updated by R, such that it matches the enlarged C.
First, recall that erasures are flagged and the reconstructor is honest. Hence, it would successfully update the sets B_i and B if any erasure happened.
The arbitrary errors introduced in the leaf shares by the cheaters are detected in step 26, as long as there are not more than $2t$ errors in total. We only look at those branches $i \notin B$, since the rest of them are not used for reconstruction. For these branches, $|B_i| \leq t$ after step 24_{ES}. Since there are at most $t_c \leq t$ cheaters, the maximum number of arbitrary errors in a branch $i \notin B$ is $2t$ for any value of t_c, and therefore R will correctly identify all errors.
As a result, if $B = C$ after verification, i.e., R will always update B correctly before trying to reconstruct the quantum secret. Therefore, it will either reconstruct the original state (if $|B| \leq t$) or abort (if $|B| > t$). Consequently, the probability of R holding consistent shares or aborting is at least $1 - 2^{-\Omega(r)}$. □

Before discussing the completeness of the modified protocol, it is convenient to compute some useful functions. Given a network under the ES noise model and the number of errors introduced by the dealer, t_d, the probability that less than $2t - t_c$ nodes are in the set C right after verification[2], i.e., $|C_v| \leq 2t - t_c$, is

$$\begin{aligned}\Pr(|C_v| \leq 2t - t_c) &\geq f_v(n, t, q, t_c, t_d) \\ &:= \Pr(|C_v| \leq 2t - t_c \mid \text{max. cheats}) \\ &= \sum_{x=0}^{2t-t_c} \sum_{y=0}^{3tn-2t^2} \Pr(u^{(1)} = x) \cdot \Pr(u^{(2)} = y) \\ &\qquad \cdot \Pr(|C_v| \leq 2t - t_c | u^{(1)}, u^{(2)}, \text{max. cheats}),\end{aligned} \tag{4.1.3}$$

where n is the number of nodes, t is the maximum number of errors tolerated by the CSS code employed, q is the probability of erasure in the ES model, t_c is the maximum number of cheaters, and 'max. cheats' means that all cheaters introduce errors in all the shares to which they have access. Each of the terms in the sum corresponds to a particular combination of the total number of branch erasures, $u^{(1)}$, and the total number of leaf erasures, $u^{(2)}$. Both of them are random variables, whose probability distributions are given by:

$$\Pr(u^{(1)} = x) = q^x \binom{n}{x} (1 - q)^{n-x}, \tag{4.1.4}$$

[2]We employ the subindex v on sets B and C when we refer to the state of those sets right after verification, since more elements could be included during the reconstruction phase.

$$\Pr(u^{(2)} = y) = q^y \binom{n^2}{y} (1-q)^{n^2-y}. \tag{4.1.5}$$

The analytical expression for $\Pr(|C_v| \leq 2t - t_c | u^{(1)}, u^{(2)}, \text{max. cheats})$ is more complicated and can be found in Appendix A, together with a full derivation of $f_v(n, t, q, t_c, t_d)$.

From the previous analytical expressions, the dependence of f_v on n, t, q, t_c, and t_d is not explicit enough. Therefore, we proceed with a graphical representation of the function in order to extract more useful information. The evaluation of the analytical expression of $\Pr(|C_v| \leq 2t - t_c | u^{(1)}, u^{(2)}, \text{max. cheats})$ given in Appendix A is computationally demanding and not reliable if done by brute force, due to the large amount of terms involving combinatorial factors of different orders of magnitude.
We have designed Algorithm 1 to evaluate $\Pr(|C_v| \leq 2t - t_c | u^{(1)}, u^{(2)}, \text{max. cheats})$ by sampling, and its output converges to the correct value for large enough number of samples:

Proposition 4.1.1. *Let $h(\mathbf{x}, N)$ be the output of Algorithm 1, with $\mathbf{x} := (n, t, q, t_c, t_d, u^{(1)}, u^{(2)})$, and let us define $\tilde{g}(\mathbf{x}, N) := \Pr(u^{(1)}) \cdot \Pr(u^{(2)}) \cdot h(\mathbf{x}, N)$ and $g(\mathbf{x}) := \Pr(u^{(1)}) \cdot \Pr(u^{(2)}) \cdot \Pr(|C| \leq 2t - t_c | u^{(1)}, u^{(2)}, \text{max. cheats})$, where $\Pr(u^{(1)})$ and $\Pr(u^{(2)})$ are the probability distributions given by Eqs. (4.1.4) and (4.1.5), respectively. Then,*

$$\Pr\Big[|\tilde{g}(\mathbf{x}, N) - g(\mathbf{x})| < \delta\Big] \geq 1 - N^{-1} \cdot \delta^{-2} \cdot \varepsilon, \tag{4.1.6}$$

for any $\mathbf{x}$, $N > 0$, and $\delta > 0$, with

$$\begin{aligned} \varepsilon \leq q^{2u^{(1)}+2u^{(2)}} \cdot (1-q)^{2n^2+2n-2u^{(1)}-2u^{(2)}} \cdot \binom{n}{u^{(1)}}^2 \cdot \binom{n^2}{u^{(2)}}^2 \\ \cdot \left[\binom{n}{u^{(1)}} \cdot \binom{n^2}{u^{(2)}} \cdot \binom{n}{t_d} \cdot \binom{n}{t_c} - 1 \right]. \end{aligned} \tag{4.1.7}$$

Proof. The proof can be found in Appendix B. □

Proposition 4.1.1 states that we can use Algorithm 1 to compute each of the terms in Eq. (4.1.3) with error smaller than δ, with probability at least $1 - N^{-1} \cdot \delta^{-2} \cdot \varepsilon$. Let us employ the symbol '$\sim$' to denote the order of magnitude of a variable. Since there are $(2t - t_c + 1) \cdot (3tn - 2t^2 + 1)$ terms in Eq. (4.1.3), the error in f_v is smaller than $\delta \cdot (2t - t_c + 1) \cdot (3tn - 2t^2 + 1) \sim 10\delta$, where we assume that $t \sim 1$ and $n \sim 10$. Therefore, it is possible to choose a value of δ that produces a reliable representation of f_v (e.g., $\delta = 10^{-3}$ yields an absolute error in f_v of $\sim 10^{-2}$) and then use Proposition 4.1.1 to calculate the number of samples required in Algorithm 1.

Using Algorithm 1, we can obtain graphical representations of f_v, and then use them to tune the parameters of the protocol in order to achieve the highest probability of passing verification holding consistent shares. Hence, we have obtained a useful tool for hyperparameter tuning and for network design.

Algorithm 1 - Approximating $\Pr(|C_{\mathrm{v}}| \leq 2t - t_{\mathrm{c}} | u^{(1)}, u^{(2)}, \text{max. cheats})$ by sampling.

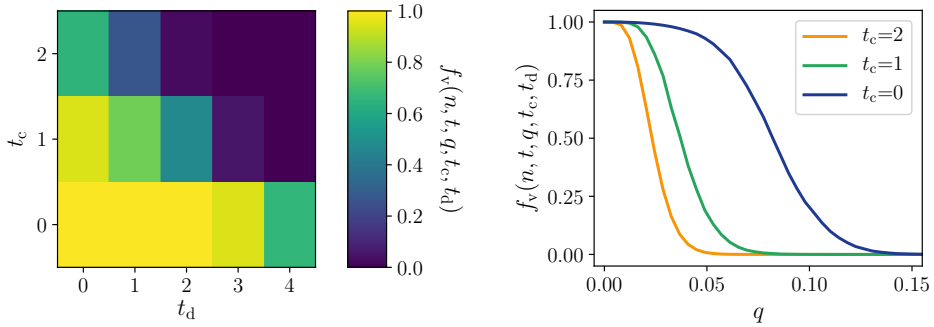
Inputs:

n: number of nodes in the network.
t: maximum number of errors the CSS code can correct.
q: probability of qubit erasure.
t_{c}: maximum number of cheaters.
t_{d}: number of errors introduced by the dealer.
$u^{(1)}$: total number of erasures in the branches.
$u^{(2)}$: total number of erasures in the leaves.
N: number of samples.

Create a variable $P \leftarrow 0$
for $s = 1, \ldots, N$ **do**
    Create an $n \times n$ array, V, with $u^{(2)}$ ones and $n^2 - u^{(2)}$ zeroes distributed uniformly at random
    Generate a $t_{\mathrm{c}} \times 1$ array, h, whose entries are integers between 0 and n with no repetition
    for $j = 1, \ldots, t_{\mathrm{c}}$ **do**
        $k \leftarrow h_j$
        $V_{ik} \leftarrow 1$, $\forall i = 1, \ldots, N$
    end for
    Sum all the values in each row of V to obtain an $n \times 1$ array v ($v_i = \sum_{j=1}^{n} V_{ij}$)
    Create an $n \times 1$ array, b, full of zeroes
    for $i = 1, \ldots, N$ **do**
        if $v_i > t$ **then** $b_i \leftarrow 1$
        end if
    end for
    Randomly choose $u^{(1)}$ elements of b and set them to 1 (regardless whether their previous value was 0 or 1)
    Randomly choose t_{d} elements of b and set them to 1 (regardless whether their previous value was 0 or 1)
    for $j = 1, \ldots, t_{\mathrm{c}}$ **do**
        $k \leftarrow h_j$
        $b_k \leftarrow 1$
    end for
    if $\sum_{i=1}^{n} b_i \leq 2t - t_{\mathrm{c}}$ **then** $P \leftarrow P + 1$
    end if
end for
Normalize P over the number of samples: $P \leftarrow P/N$
return P

As an example, take a network of $n = 18$ quantum nodes running VHSS with an $[[18, 1, 5]]$ CSS code ($t = 2$). Before running the protocol, we want to choose the values of the parameters that will maximize the probability of having $|C| \leq 2t - t_c$ right after the verification. We consider two different scenarios: (i) a network with fixed $q = 0.02$, and (ii) a network with an honest dealer.

In (i), we want to select a suitable value for t_c and t_d. For all possible combinations of values, we have $\varepsilon < 8 \cdot 10^{-2}$. A reasonable value for δ is 10^{-3}, as this provides a precision of 10^{-2} in the estimate of f_v, which is enough to assess the performance of the protocol. Moreover, we would like to ensure this with more than a 99.9% chance. Hence, using Proposition 4.1.1, we find that it is safe to use the algorithm with at least $N \sim 10^7$ samples. Note that this is a clear advantage in the computational cost since the evaluation of most of the terms in Eq. (4.1.3) already requires a loop over more than 10^7 different configurations of erasures for fixed $u^{(1)}$ and $u^{(2)}$, e.g., for $u^{(1)} = 0$ and $u^{(2)} = 5$ there are roughly 10^{13} different configurations.
Figure 4.1a shows $f_v(n, t, q, t_c, t_d)$ for $n = 18$, $t = 2$, $q = 0.02$, and different combinations of t_c and t_d. This can be employed to find the values that better fit our interests. For example, if one wants the probability that honest nodes hold consistent shares to be larger than 0.9, i.e., $\Pr(|C| \leq 2t - t_c) > 0.9$, it is enough to ensure that $f_v(n, t, q, t_c, t_d) > 0.9$. In Fig. 4.1a we observe that, in order to achieve it, we can tolerate at most 1 cheater if the dealer is honest and none if the dealer introduces up to 3 arbitrary errors. In any other case, we cannot ensure that $\Pr(|C| \leq 2t - t_c) > 0.9$.



(a) **(b)**

Figure 4.1. Design function for the VHSS-ES protocol built upon an $[[18, 1, 5]]$ CSS code. These plots can be used to tune the maximum number of cheaters, t_c, and the maximum number of errors introduced by the dealer, t_d), that the protocol tolerates, and possibly to determine whether improving the quality of the channels, i.e., reducing q, is necessary or not. **(a)** Quantum network with $q = 0.02$. **(b)** Quantum network with honest dealer ($t_d = 0$). Both subfigures were computed using Eq. (4.1.3) and Algorithm 1 with $N = 10^7$.

In situation (*ii*), the dealer is honest, and we would like to choose a proper value for t_c. Figure 4.1b shows $f_v(n, t, q, t_c, t_d)$ versus the probability of erasure q. As an example, if the network was characterized by $q = 0.15$, we observe that f_v is almost zero even if we do not tolerate any cheater, and therefore we cannot ensure that the secret would be properly shared among the honest nodes. Let us now assume that $q = 0.02$ and that we aim for $\Pr(|C| \leq 2t - t_c) > 0.9$. Then, by looking at which curves are above 0.9 at $q = 0.02$, we determine that it is possible to tolerate up to 1 cheater.
Note that Figure 4.1b still captures the case of the original VHSS protocol ($q = 0$), in which an honest dealer ($t_d = 0$) would always pass verification ($\Pr(|C| \leq 2t - t_c) = 1$).

After discussing the meaning and use cases of f_v, we are ready to state the completeness of the VHSS-ES PROTOCOL.

Theorem 4.1.3 (Completeness)**.** *In the verifiable hybrid secret sharing protocol for networks under the ES noise model,* VHSS-ES PROTOCOL*, if D is honest then it passes the verification phase with probability at least $f_v(n, t, q, t_c, 0)$. Moreover, if R is also honest, it reconstructs D's secret with probability at least*

$$(1 - 2^{-\Omega(r)}) \cdot (1 - q)^{(n-2t)(n-t)}, \tag{4.1.8}$$

where r is the security parameter.

Proof. The probability of passing verification for an honest dealer is:

$$\begin{aligned}
\Pr(\text{pass verification}) &:= \Pr(|B_v| \leq 2t - t_c) \\
&= \Pr(|C_v| \leq 2t - t_c) \cdot \Pr(|B_v| \leq 2t - t_c \,\big|\, |C_v| \leq 2t - t_c) \\
&\quad + \Pr(|C_v| > 2t - t_c) \cdot \Pr(|B_v| \leq 2t - t_c \,\big|\, |C_v| > 2t - t_c) \\
&\overset{a}{\geq} \Pr(|C_v| \leq 2t - t_c) \\
&\overset{b}{\geq} f_v(n, t, p, t_c, 0)
\end{aligned} \tag{4.1.9}$$

where, in step a, we have used that $\Pr(|B_v| \leq 2t - t_c \,\big|\, |C_v| \leq 2t - t_c) = 1$ and that the second term of the sum is non-negative; and in step b we have used the definition of $f_v(n, t, p, t_c, t_d)$ (see Eq. (4.1.3)) with $t_d = 0$, since we assume an honest dealer.

Let us now consider the second part of the statement. First, from Theorem 4.1.2, we know that the honest nodes hold consistent shares after verification (i.e., $|C| \leq 2t - t_c$) at least with probability $1 - 2^{-\Omega(r)}$. Then, the probability that R also holds consistent shares and therefore is able to reconstruct is at least $(1 - 2^{-\Omega(r)}) \cdot \Pr(X)$, where $\Pr(X)$ is the probability that the size of C remains less than or equal to $2t$ after sending all shares to R (we call this event X). As discussed in the proof of Theorem 4.1.2, R always updates the set B correctly. Therefore, $\Pr(X)$ can also be regarded as the probability that $|B| \leq 2t$ after sending all shares to R.
In order to reconstruct the secret, we need $|B| \leq 2t$. This implies that at least

$n-2t$ branch shares have not suffered from errors and can be reconstructed, i.e., at least $n-t$ of their leaf shares have not suffered from errors ($|B_i| \leq t$, $\forall i \notin B$). If these $(n-2t)(n-t)$ leaf shares did not suffer from stochastic erasures (we call this event Y), then the shares held by R are still consistent. Mathematically, $Y \Rightarrow X$. Therefore,

$$\Pr(X) \geq \Pr(Y) = (1-q)^{(n-2t)(n-t)}, \tag{4.1.10}$$

where $(1-q)^{(n-2t)(n-t)}$ is the probability that none of the $(n-2t)(n-t)$ leaf shares is erased.
We can conclude that the probability that R still holds consistent shares after verification is

$$(1-2^{-\Omega(r)}) \cdot \Pr(X) \geq (1-2^{-\Omega(r)}) \cdot (1-q)^{(n-2t)(n-t)}. \tag{4.1.11}$$

□

4.2. Erasures in the secret and the ancillae

In the previous section, we assumed that erasures only happened in the secret shares. A more realistic model should consider that the ancillas travel through the same communication channels and therefore are also subject to noise. In this section, we allow both secret and ancilla shares to be erased, giving rise to two new models:

Definition 4.2.1. We define the **'Channel Erasure' (CE) model** as a noise model for quantum networks running the VHSS protocol in which: (*i*) each pair of nodes j and k is connected by two opposite unidirectional channels, one from j to k and another one from k to j, (*ii*) each of these channels can be independently broken with probability q at the beginning of the protocol, erasing all qubits that are sent through it, and (*iii*) all channels are restored after verification but they can break again before reconstruction with probability q.

Definition 4.2.2. We define the **'Erasures on the Secret and the Ancillas' (ESA) model** as a noise model for quantum networks running the VHSS protocol in which each qubit has an independent probability q of being erased when traveling from node to node.

CE MODEL. Let us assume that the dealer is connected to node i by channel i and that each node i is connected to j by the unidirectional channel i_j. All qubits are distributed using these channels. We assume that, instead of erasing qubits with probability q, each channel is broken with probability q at the beginning of the protocol and therefore it losses all qubits, i.e., if channel i_j breaks down, shares $\Phi_{i_j}^{0,0}$, $|\bar{+}\rangle_{i_j}^{0,m}$ $(m=1,\ldots,r)$, and $|\bar{0}\rangle_{i_j}^{l,m}$ $(l=1,\ldots,r,\ m=0,\ldots,r)$ are erased, but $\Phi_{j_i}^{0,0}$, $|\bar{+}\rangle_{j_i}^{0,m}$, and $|\bar{0}\rangle_{j_i}^{l,m}$ are not. In addition, we assume that all channels are restored after verification. Hence, when the qubits are sent to the reconstructor, they can again be erased with probability q.
As in the ES model, we treat erasures as arbitrary errors. Erased shares are replaced

by new arbitrary qubits, but we use the fact that erasures are flagged to update the sets B_i and B every time a qubit is lost. To implement this, the sharing stage requires the same changes as with the ES model, i.e., steps 3 and 4 from the original protocol (see Subsection 3.2.1) must be replaced by steps 3_{ES} and 4_{ES} from the VHSS-ES PROTOCOL. Verification also requires the same abort condition as the VHSS-ES PROTOCOL besides an additional modification: whenever an ancillary qubit is lost, it has to be replaced by a new arbitrary qubit. However, it is not necessary to publicly announce erasures nor to update the sets B_i or B, since erased ancillas correspond to erased secret shares and therefore these actions are already performed in steps 3_{ES} and 4_{ES}. Finally, the reconstruction stage is identical to the reconstruction within the ES model, since each share can be lost independently with probability q when traveling to the reconstructor.
For completeness, we present all the previous modifications in the following box, in the same format as VHSS-ES PROTOCOL.

VHSS-CE PROTOCOL. VHSS protocol for networks described by a CE noise model. For simplicity, we only list here the steps that are different with respect to the VHSS-ES PROTOCOL.

⋮

5_{CE}. (Preparation and distribution - Z basis) The dealer D prepares an n-qubit ancillary state $|\bar{+}\rangle^{0,m}_{[1,n]} = \sum_{v\in V}|v\rangle$. Then, it sends qubit $|\bar{+}\rangle^{0,m}_{i}$ to node i, $\forall i = 1,\dots,n$. If share $|\bar{+}\rangle^{0,m}_{i}$ is erased, node i replaces it by an arbitrary qubit.

6_{CE}. (Leaf encoding and distribution - Z basis) Each node i encodes its share $|\bar{+}\rangle^{0,m}_{i}$ into an n-qubit state $|\bar{+}\rangle^{0,m}_{i_{[1,n]}}$ using $\mathcal{C}$. Then, it sends qubit $|\bar{+}\rangle^{0,m}_{i_j}$ to node j, $\forall j = 1,\dots,n$. If share $|\bar{+}\rangle^{0,m}_{i_j}$ is erased, node j replaces it by an arbitrary qubit.

⋮

9_{CE}. (Preparation and distribution - X basis) The dealer D prepares an n-qubit ancillary state $|\bar{0}\rangle^{l,0}_{[1,n]} = \sum_{w\in W^{\perp}}|w\rangle$. Then, it sends qubit $|\bar{0}\rangle^{l,0}_{i}$ to node i, $\forall i = 1,\dots,n$. If share $|\bar{0}\rangle^{l,0}_{i}$ is erased, node i replaces it by an arbitrary qubit.

10_{CE}. (Leaf encoding and distribution - X basis) Each node i encodes its share $|\bar{0}\rangle^{l,0}_{i}$ into an n-qubit state $|\bar{0}\rangle^{l,0}_{i_{[1,n]}}$ using $\mathcal{C}$. Then, it sends qubit $|\bar{0}\rangle^{l,0}_{i_j}$ to node j, $\forall j = 1,\dots,n$. If share $|\bar{0}\rangle^{l,0}_{i_j}$ is erased, node j replaces it by an arbitrary qubit.

$\vdots$

11_{CE}. (Preparation and distribution - $l, m \neq 0$) The dealer D prepares an n-qubit ancillary state $|\bar{0}\rangle_{[1,n]}^{l,m} = \sum_{w \in W^{\perp}} |w\rangle$. Then, it sends qubit $|\bar{0}\rangle_{i}^{l,m}$ to node i, $\forall i = 1, \ldots, n$. If share $|\bar{0}\rangle_{i}^{l,m}$ is erased, node i replaces it by an arbitrary qubit.

12_{CE}. (Leaf encoding and distribution - $l, m \neq 0$) Each node i encodes its share $|\bar{0}\rangle_{i}^{l,m}$ into an n-qubit state $|\bar{0}\rangle_{i_{[1,n]}}^{l,m}$ using $\mathcal{C}$. Then, it sends qubit $|\bar{0}\rangle_{ij}^{l,m}$ to node j, $\forall j = 1, \ldots, n$. If share $|\bar{0}\rangle_{ij}^{l,m}$ is erased, node j replaces it by an arbitrary qubit.

$\vdots$

The CE and the ES models describe different physical scenarios. However, their respective modified VHSS protocols achieve the same functionalities, meaning that the security statements (Theorems 4.1.1, 4.1.2, and 4.1.3) and the design function f_{v} of the ES model (see Section 4.1) also apply to the CE model. Let us discuss now why.

The fundamental difference between both models is that, in the CE model, whenever a secret share is erased, its corresponding ancillary share is also erased.
Let us assume that a branch share $\Phi_k^{0,0}$ has been erased when traveling from the dealer to node k. In both VHSS-ES PROTOCOL and VHSS-CE PROTOCOL, if k acts honestly, it will publicly announce the erasure and include k in the set B. After that, the k-th branch ancillas, $|\bar{+}\rangle_k^{0,m}$ and $|\bar{0}\rangle_k^{l,m}$, are not useful anymore. Therefore, the erasure of these ancillas has no effect over the protocol.
If node k is a cheater, it will only pass verification without reporting the erasure and including k in B with probability exponentially small in the security parameter r, regardless of the state of the ancillas $|\bar{+}\rangle_k^{0,m}$ and $|\bar{0}\rangle_k^{l,m}$, since the CNOTs between secret and ancilla shares are applied at random (see Subsection 3.2.1). Nevertheless, this situation is taken into account even in the original protocol, since the cheater itself could erase the qubit.
We conclude that, if a branch share $\Phi_k^{0,0}$ is erased, the state of its corresponding ancillas, $|\bar{+}\rangle_k^{0,m}$ and $|\bar{0}\rangle_k^{l,m}$, is not relevant. A similar argument can be given if a leaf share $\Phi_{ij}^{0,0}$ was erased. Hence, the VHSS-CE PROTOCOL and the VHSS-ES PROTOCOL satisfy the same security statements, and f_{v} given by Eq. (4.1.3) is valid for both models.

ESA MODEL. Let us now assume that qubits have an independent probabil-

ity q of being erased every time they travel from node to node. If they are lost, they are replaced by arbitrary qubits and the erasure is treated as an arbitrary error (although it is still flagged). This might seem much more complex to analyze than the ES and the CE models, since any of the ancillary qubits can be erased. However, a change of variables will allow us to apply the results of the ES model to the ESA model.
First, let us propose a modified VHSS protocol for the ESA model. The only change that we need to introduce with respect to the VHSS-CE PROTOCOL is that ancillary qubit erasures have to be publicly announced in order to properly update sets B_i and B. We give an explicit description of the protocol in the following box:

VHSS-ESA PROTOCOL. VHSS protocol for networks described by an ESA noise model. For simplicity, we only list here the steps that are different with respect to the VHSS-ES PROTOCOL.

$\vdots$

5_{ESA}. (Preparation and distribution - Z basis)

$5_{\text{ESA}}.1$. The dealer D prepares an n-qubit ancillary state $|\bar{+}\rangle_{[1,n]}^{0,m} = \sum_{v\in V}|v\rangle$. Then, it sends qubit $|\bar{+}\rangle_i^{0,m}$ to node i, $\forall i = 1,\dots,n$.

$5_{\text{ESA}}.2$. Erasures are publicly announced by the nodes. If an erasure happens on $|\bar{+}\rangle_i^{0,m}$, node i replaces its share by an arbitrary qubit and $B \leftarrow B \cup \{i\}$.

$5_{\text{ESA}}.3$. If $|B| > 2t - t_c$, abort the protocol. Otherwise, continue.

6_{ESA}. (Leaf encoding and distribution - Z basis)

$6_{\text{ESA}}.1$. Each node i encodes its share $|\bar{+}\rangle_i^{0,m}$ into an n-qubit state $|\bar{+}\rangle_{i_{[1,n]}}^{0,m}$ using $\mathcal{C}$. Then, it sends qubit $|\bar{+}\rangle_{ij}^{0,m}$ to node j, $\forall j = 1,\dots,n$.

$6_{\text{ESA}}.2$. Erasures are publicly announced by the nodes. If an erasure happens on $|\bar{+}\rangle_{ij}^{0,m}$, node j replaces its share by an arbitrary qubit and $B_i \leftarrow B_i \cup \{j\}$.

$6_{\text{ESA}}.3$. If $|B_i| > t$, $B \leftarrow B \cup \{i\}$, $\forall i = 1,\dots,n$.

$6_{\text{ESA}}.4$. If $|B| > 2t - t_c$, abort the protocol. Otherwise, continue.

$\vdots$

9_{ESA}. (Preparation and distribution - X basis)

$9_{\text{ESA}}.1$. The dealer D prepares an n-qubit ancillary state $|\bar{0}\rangle^{l,0}_{[1,n]} = \sum_{w\in W^{\perp}} |w\rangle$. Then, it sends qubit $|\bar{0}\rangle^{l,0}_{i}$ to node i, $\forall i = 1,\ldots,n$.

$9_{\text{ESA}}.2$. Erasures are publicly announced by the nodes. If an erasure happens on $|\bar{0}\rangle^{l,0}_{i}$, node i replaces its share by an arbitrary qubit and $B \leftarrow B \cup \{i\}$.

$9_{\text{ESA}}.3$. If $|B| > 2t - t_{\text{c}}$, abort the protocol. Otherwise, continue.

10_{ESA}. (Leaf encoding and distribution - X basis)

$10_{\text{ESA}}.1$. Each node i encodes its share $|\bar{0}\rangle^{l,0}_{i}$ into an n-qubit state $|\bar{0}\rangle^{l,0}_{i_{[1,n]}}$ using $\mathcal{C}$. Then, it sends qubit $|\bar{0}\rangle^{l,0}_{i_j}$ to node j, $\forall j = 1,\ldots,n$.

$10_{\text{ESA}}.2$. Erasures are publicly announced by the nodes. If an erasure happens on $|\bar{0}\rangle^{l,0}_{i_j}$, node j replaces its share by an arbitrary qubit and $B_i \leftarrow B_i \cup \{j\}$.

$10_{\text{ESA}}.3$. If $|B_i| > t$, $B \leftarrow B \cup \{i\}$, $\forall i = 1,\ldots,n$.

$10_{\text{ESA}}.4$. If $|B| > 2t - t_{\text{c}}$, abort the protocol. Otherwise, continue.

$\vdots$

11_{ESA}. (Preparation and distribution - $l, m \neq 0$)

$11_{\text{ESA}}.1$. The dealer D prepares an n-qubit ancillary state $|\bar{0}\rangle^{l,m}_{[1,n]} = \sum_{w\in W^{\perp}} |w\rangle$. Then, it sends qubit $|\bar{0}\rangle^{l,m}_{i}$ to node i, $\forall i = 1,\ldots,n$.

$11_{\text{ESA}}.2$. Erasures are publicly announced by the nodes. If an erasure happens on $|\bar{0}\rangle^{l,m}_{i}$, node i replaces its share by an arbitrary qubit and $B \leftarrow B \cup \{i\}$.

$11_{\text{ESA}}.3$. If $|B| > 2t - t_{\text{c}}$, abort the protocol. Otherwise, continue.

1_{ESA}. (Leaf encoding and distribution - $l, m \neq 0$)

$12_{\text{ESA}}.1$. Each node i encodes its share $|\bar{0}\rangle^{l,m}_{i}$ into an n-qubit state $|\bar{0}\rangle^{l,m}_{i_{[1,n]}}$ using $\mathcal{C}$. Then, it sends qubit $|\bar{0}\rangle^{l,m}_{i_j}$ to node j, $\forall j = 1,\ldots,n$.

$12_{\text{ESA}}.2$. Erasures are publicly announced by the nodes. If an erasure happens on $|\bar{0}\rangle_{ij}^{l,m}$, node j replaces its share by an arbitrary qubit and $B_i \leftarrow B_i \cup \{j\}$.

$12_{\text{ESA}}.3$. If $|B_i| > t$, $B \leftarrow B \cup \{i\}$, $\forall i = 1, \ldots, n$.

$12_{\text{ESA}}.4$. If $|B| > 2t - t_c$, abort the protocol. Otherwise, continue.

⋮

The VHSS-ESA PROTOCOL can be employed in the absence of noise and in networks described by any of the proposed noise models so far. This can be easily shown by checking that all the modified steps in the VHSS-ESA PROTOCOL reduce to the VHSS-CE PROTOCOL, the VHSS-ES PROTOCOL, or the original protocol from [14] when using the corresponding noise model. For example, if the ancillas are not affected by noise, the VHSS-ESA PROTOCOL automatically reduces to VHSS-ES PROTOCOL since the qubits cannot be erased when traveling from node to node.

Let us now show how to apply the analysis from Section 4.1 to the ESA model. A major difference between the ES and the ESA models in terms of the variables employed throughout the protocol is that, in the latter, ancilla erasures can enlarge the set C.

When qubits are noisy, it is not possible to distinguish whether an erasure is due to a noisy channel or a cheater. The VHSS-ES PROTOCOL already considers that cheaters can perform any operation on their shares. Therefore, in order to include the effect of noisy ancillas in the results derived for the ES model (Section 4.1), it is only necessary to analyze the state of the qubits held by honest nodes.

Let us assume that nodes i and j are honest. As discussed previously, the erasures of $\Phi_i^{0,0}$, $|\bar{+}\rangle_i^{0,m}$, and $|\bar{0}\rangle_i^{l,m}$ have the same effect over the set B: after any of these qubits is lost, node i will publicly announce it and i will be included in B. The probability that none of the aforementioned qubits is erased is $(1-q)^{(r+1)^2}$, since there are $(r+1)^2$ branch shares labeled with index i. Hence, the probability that at least one of those qubits is erased is $1-(1-q)^{(r+1)^2}$. Similarly, the erasures of $\Phi_{ij}^{0,0}$, $|\bar{+}\rangle_{ij}^{0,m}$, and $|\bar{0}\rangle_{ij}^{l,m}$ have the same effect, and j is included in B_i with probability $1-(1-q)^{(r+1)^2}$.

Then, we conclude that the verification stage of the ESA model is equivalent to an ES model in which each secret share can be erased with an effective probability

$$q_s := 1-(1-q)^{(r+1)^2} \tag{4.2.1}$$

when traveling from node to node in the sharing stage.

Finally, qubits can be independently erased with probability q while traveling to the reconstructor in both models. Since the reconstruction stages in the VHSS-ESA PROTOCOL and the VHSS-ES PROTOCOL are the same, the results derived for the ES model are valid for the ESA model.

After the previous discussion, we are ready to reformulate the security statements. First, Theorem 4.1.1 (secrecy) applies to VHSS-ESA PROTOCOL, since the proof relies on the secrecy of the underlying VCSS scheme and it is not affected by errors happening on the quantum shares.
Theorem 4.1.2 (soundness) also applies to the VHSS-ESA PROTOCOL, and the proof is the same as for the ES model, since the erasures on the ancillas are flagged and therefore the set B is always properly updated by the honest nodes throughout the protocol.

The function f_{v} employed in Section 4.1 was derived by analyzing the state of the shares in the sharing and verification stages. Therefore, Eq. (4.1.3) is valid if we use q_{s} instead of q, i.e., $f_{\mathrm{v}}(n,t,q_{\mathrm{s}},t_{\mathrm{c}},t_{\mathrm{d}})$ is a lower bound for $\Pr(|C_{\mathrm{v}}| \leq 2t - t_{\mathrm{c}})$ in the ESA model (recall that C_{v} denotes the set C right after verification). The derivation of the full expression is the same as for the ES model, except that the probability distributions of the total number of branch erasures, $u^{(1)}$, and the total number of leaf erasures, $u^{(2)}$, are now given by

$$\Pr(u^{(1)} = x) = q_{\mathrm{s}}^{x} \binom{n}{x} (1-q_{\mathrm{s}})^{n-x}, \tag{4.2.2}$$

$$\Pr(u^{(2)} = y) = q_{\mathrm{s}}^{y} \binom{n^2}{y} (1-q_{\mathrm{s}})^{n^2-y}. \tag{4.2.3}$$

Moreover, Algorithm 1 can still be used to evaluate $f_{\mathrm{v}}(n,t,q_{\mathrm{s}},t_{\mathrm{c}},t_{\mathrm{d}})$ by sampling, since it takes a fixed $u^{(1)}$ and a fixed $u^{(2)}$ as inputs and hence it does not involve any probability of erasure.
In the ESA model, we can therefore use f_{v} and Algorithm 1 to tune the parameters of the protocol in order to improve its performance, in the same way as we did within the ES model (see Fig. 4.1).

Finally, we state the completeness of the VHSS-ESA PROTOCOL:

Theorem 4.2.1 (Completeness)**.** *In the verifiable hybrid secret sharing protocol for networks under ESA noise model,* VHSS-ESA PROTOCOL*, if D is honest then it passes the verification phase with probability at least $f_{\mathrm{v}}(n,t,q_{\mathrm{s}},t_{\mathrm{c}},0)$, where f_{v} is given by Eq. (4.1.3) and $q_{\mathrm{s}} = 1-(1-q)^{(r+1)^2}$. Moreover, if R is also honest, it reconstructs D's secret with probability at least*

$$(1-2^{-\Omega(r)}) \cdot (1-q)^{(n-2t)(n-t)}, \tag{4.2.4}$$

where r is the security parameter.

Proof. The proof is the same as for Theorem 4.1.3 but considering that the probability of erasure in the sharing stage is now $q_{\mathrm{s}} = 1-(1-q)^{(r+1)^2}$, as previously discussed. □

Note that q_s is monotonously decreasing in q but it increases exponentially in r. Therefore, a larger value of the security parameter r decreases the probability of passing verification holding inconsistent shares, but it also decreases the probability of passing verification at all. This is actually an intuitive result: the more verification rounds r that are run, the higher the probability of adding node i to set B due to a random erasure. We find here a tradeoff that should be taken into consideration when implementing the VHSS-ESA PROTOCOL on a real quantum network.

4.3. Depolarizing noise

Let us now consider the case of a network in which the channels connecting any two nodes can depolarize the qubits. The natural analogues of the ES, the CE, and the ESA noise models are the DS ('Depolarization on the Secret'), CD ('Channel Depolarization') and the DSA ('Depolarization on the Secret and the Ancillas') models. In these noise models, erasure channels are replaced by depolarization channels. Let us explicitly define the three models:

Definition 4.3.1. We define the **'Depolarization on the Secret' (DS) model** as a noise model for quantum networks running the VHSS protocol in which: (*i*) each qubit has an independent probability q of being depolarized when traveling from node to node and (*ii*) only shares of the encoded secret can be depolarized, i.e., ancillary qubits do not suffer from noise.

Definition 4.3.2. We define the **'Channel Depolarization' (CD) model** as a noise model for quantum networks running the VHSS protocol in which: (*i*) each pair of nodes is connected by two opposite unidirectional channels, (*ii*) each of these channels can be independently broken with probability q at the beginning of the protocol, depolarizing all qubits that are sent through it, and (*iii*) all channels are restored after verification but they can break again before reconstruction with probability q.

Definition 4.3.3. We define the **'Depolarization on the Secret and the Ancillas' (DSA) model** as a noise model for quantum networks running the VHSS protocol in which each qubit has an independent probability q of being depolarized when traveling from node to node.

Throughout this work, erasures have been effectively treated as arbitrary errors. Therefore, when the network is described by the DS, the CD, or the DSA model, one could make use of the VHSS-ES PROTOCOL, the VHSS-CE PROTOCOL, or the VHSS-ESA PROTOCOL, respectively. However, a strong limitation arises from the fact that depolarized qubits are not flagged, and the security statements derived in the previous sections may not apply anymore. If too many qubits are depolarized, the protocol can fail. For example, after the nodes pass verification holding consistent shares, all qubits can be depolarized on their way to the reconstructor, possibly generating a valid but different encoded state. In that case, even if a secret is reconstructed, we cannot guarantee that it corresponds to the original state. As a result, the security statements that we can derive for the depolarization models are

weaker than the ones for the erasure models.

Let us first consider the DS model. We start by defining the function $S(z;n,q)$,

$$S(z;n,q) := \sum_{k=0}^{z} q^k (1-q)^{n-k} \binom{n}{k}, \tag{4.3.1}$$

which corresponds to the cumulative distribution function of a random variable that follows a binomial distribution with parameters n (number of trials) and q (probability of success per trial).
The secrecy statement, Theorem 4.1.1, still holds when qubits can be depolarized instead of erased, since its proof is based on the properties of the underlying VCSS scheme and it is not affected by quantum errors happening on the VQSS subroutine. Regarding soundness, we have the following statement:

Theorem 4.3.1 (Soundness)**.** *In the* VHSS-ES PROTOCOL*, when run on a network described by the DS noise model, either the honest parties hold a consistently encoded secret or the protocol aborts, with probability at least* $\left[S(2t-t_c;n,q)\right]^{n-t_c}\cdot(1-2^{-\Omega(r)})$*. Moreover, after passing verification and sending all qubits to an honest* R*, either the shares are still consistent with the same secret or the protocol aborts, with probability at least* $\left[S(2t-t_c;n,q)\right]^{n-t_c}\left[S(t-t_c;n,q)\right]^{n}\cdot(1-2^{-\Omega(r)})$*.*

Proof. First, note that the original VHSS from [14] and all our modified protocols already consider that branch shares $\Phi_i^{0,0}$ can be depolarized, as the dealer could have introduced any arbitrary errors on the qubits before sending them to the nodes. However, the same is not true for the leaves $\Phi_{i_{[1,n]}}^{0,0}$: in order to detect all arbitrary errors, there must be at most $2t$.[3] There are at most t_c cheaters, therefore, if more than $2t-t_c$ leaf shares from a single branch were depolarized, these errors may not be detected and this branch should not be trusted. In the ES, CE, and ESA models, erasures were flagged, so this situation was avoided by including branch i into the set of apparent cheaters B whenever the number of erasures in its leaves exceeded certain threshold. Unfortunately, depolarizations are not flagged and it is not possible to assess whether this happened or not. Nevertheless, as long as the branches held by the $n-t_c$ honest nodes do not suffer more than $2t-t_c$ stochastic depolarizations (we call this event T), the sharing and verification stages from VHSS-ES PROTOCOL work. This means that either the honest parties hold a consistently encoded secret or the protocol aborts (we call this event A), with probability

$$\begin{aligned}\Pr(\neg T)\cdot\Pr(A\mid\neg T)+\Pr(T)\cdot\Pr(A\mid T) &\geq \Pr(T)\cdot\Pr(A\mid T)\\ &= \left[S(2t-t_c;n,q)\right]^{n-t_c}\cdot(1-2^{-\Omega(r)}),\end{aligned} \tag{4.3.2}$$

where $\Pr(A\mid T)=1-2^{-\Omega(r)}$ was found in the proof of Theorem 4.1.2 (in that proof, $\Pr(T)=1$, since erasures were flagged). Moreover, we have also used that

[3] Recall that, in general, the underlying CSS code cannot detect more than $2t$ arbitrary errors.

$\Pr(T) = \big[S(2t - t_c; n, q)\big]^{n-t_c}$, since the probability of T is the probability that at most $2t - t_c$ out of n leaves are erased, which is described by the cumulative distribution function of a binomial distribution, in each of the $n - t_c$ branches encoded by the honest nodes.
After verification, qubits can still be depolarized on their way to the reconstructor. Again, these errors are not flagged and a branch should not be trusted if too many of its leaf shares were depolarized. Hence, we need that the leaf shares from each branch $i \notin B$ (those that could be used to reconstruct the secret) do not suffer more than $2t - t_c - t$ errors while traveling to the reconstructor (we call this event T_r), in order to be able to detect these errors. This way, we allow t_c cheaters to introduce errors and still have less than $2t$ in total, considering that branches $i \notin B$ have at most t leaf errors at the end of the verification stage. Finally, the shares held by R are still consistent with the same secret or the protocol aborts (event A_r) with probability $1 - 2^{-\Omega(r)}$, the same as in Theorem 4.1.2, as long as there is not an excess of depolarizations, i.e., given T and T_r. Hence,

$$\begin{aligned}
&\Pr(\neg(T \wedge T_r)) \cdot \Pr(A_r \mid \neg(T \wedge T_r)) + \Pr(T \wedge T_r) \cdot \Pr(A_r \mid T \wedge T_r) \\
&\quad \geq \Pr(T \wedge T_r) \cdot \Pr(A_r \mid T \wedge T_r) \\
&\quad = \Pr(T) \cdot \Pr(T_r \mid T) \cdot \Pr(A_r \mid T \wedge T_r) \\
&\quad \geq \big[S(2t - t_c; n, q)\big]^{n-t_c} \big[S(t - t_c - 1; n, q)\big]^{n} \cdot (1 - 2^{-\Omega(r)}),
\end{aligned} \tag{4.3.3}$$

where $\Pr(T) = \big[S(2t - t_c; n, q)\big]^{n-t_c}$, $\Pr(A_r \mid T \wedge T_r) = 1 - 2^{-\Omega(r)}$, and

$$\Pr(T_r \mid T) \geq \big[S(t - t_c; n, q)\big]^{n}, \tag{4.3.4}$$

since $S(t - t_c; n, q)$ is the probability that the leaves of a single branch $i \notin B$ do not suffer more than $t - t_c$ errors while traveling to R, and there are at most n of these branches. □

In the original VHSS and in the modified versions for erasure models, all errors can be detected. Consequently, either the nodes hold consistent shares or the protocol aborts with probability exponentially close to 1 in r. The strong limitations imposed by the fact that depolarizations are not flagged are already present in the soundness statement. There exists a non-negligible probability that the protocol fails. This can happen in several ways, e.g., nodes could hold inconsistent shares and attempt to reconstruct instead of aborting. In the case of an honest dealer, nodes could reconstruct a different secret instead of the one shared by D.

The function $S(z; n, q)$ is not easy to manipulate when the number of terms in the sum is large. In some situations, particularly for $n \gg 1$, one can use the following bound (see Appendix C for a detailed derivation):

$$S(z; n, q) \geq 1 - \left(\frac{z+1}{nq}\right)^{-(z+1)} \left(\frac{n - (z+1)}{n - nq}\right)^{z+1-n}, \quad q < \frac{z}{n} < 1. \tag{4.3.5}$$

This bound is particularly tight for small values of q [49], as displayed in Figure 4.2. This figure shows the probability that the honest parties hold a consistently encoded secret or the protocol aborts, except with probability exponentially small in r, versus q, for $n = 18$ and $t = \lfloor \frac{n-1}{4} \rfloor = 4$. Furthermore, this plot shows that one can improve the performance of the protocol by decreasing the maximum number of cheaters that are tolerated, since the blue lines ($t_c = 0$) are above the orange ones ($t_c = t$). For example, if the channels of our network have a probability $q = 0.1$ of depolarizing qubits, we can set $t_c = 0$ instead of $t_c = t$ to ensure that the probability of holding consistent shares or aborting is larger than 0.999.

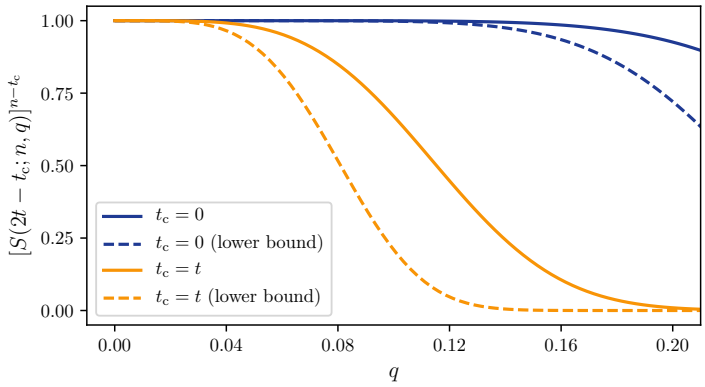


Figure 4.2. Probability that the honest parties hold a consistently encoded secret or the protocol aborts, except with probability exponentially small in r, when the network is described by the DS noise model (see Theorem 4.3.1). In this example, $n = 18$ and $t = \lfloor \frac{n-1}{4} \rfloor = 4$. Solid lines correspond to the exact definition of $S(z; n, q)$ from Eq. (4.3.1). Dashed lines were computed using the lower bound from Eq. (4.3.5), and they are only valid for $q < 0.44$ and $q < 0.22$ when $t_c = 0$ and $t_c = t$, respectively. We only represent the interval between $q = 0$ and $q = 0.2$ since generally we are not interested in larger values of q.

Let us now state the completeness of the protocol when using the DS model.

Theorem 4.3.2 (Completeness)**.** *In the* VHSS-ES PROTOCOL*, when run on a network described by the DS noise model, if D is honest then it passes the verification phase with probability at least $\left[S(2t - t_c; n, q)\right]^{n-t_c} \cdot f_v(n, t, q, t_c, 0)$, where f_v is given by Eq. (4.1.3). Moreover, if R is also honest, it reconstructs D's secret with probability at least*

$$(1 - 2^{-\Omega(r)}) \cdot \left[S(2t - t_c; n, q)\right]^{n-t_c} \cdot (1 - q)^{(n-2t)(n-t)}, \tag{4.3.6}$$

where r is the security parameter.

Proof. As discussed in the proof of the soundness statement, the completeness of the ES model (Theorem 4.1.3) can be applied to the DS model as long as there are not too many depolarizations. A sufficient condition is that each branch that was

encoded by an honest node does not suffer more than $2t - t_c$ leaf depolarizations. The probability of this event is $\left[S(2t - t_c; n, q)\right]^{n-t_c}$, as discussed in the proof of Theorem 4.3.1. Consequently, Theorem 4.1.3 holds at least with this probability, yielding this new completeness statement for the DS model. □

To conclude this section, let us briefly discuss the CD and DSA noise models. On the one hand, the same arguments that were used in the previous section to show that the CE and the ES models are equivalent can be used to show that the CD model is equivalent to the DS model. Therefore, the security statements derived in this section remain valid when using the CD model.
On the other hand, in the DSA model, the ancillary qubits can be depolarized. Let us consider one of the $(r+1)^2$ trees formed after the encoding of the encrypted secret or an ancilla. Let us assume that the leaf shares of any of the $n - t_c$ branches that were encoded by honest nodes suffer at most $2t - t_c$ depolarizations, otherwise we could not directly make a general statement about the security of the protocol, since the errors introduced may not be detected. By assuming this, each branch has at most $2t$ arbitrary errors in the leaf shares, and the verification stages of the DSA and the ESA models are equivalent (the flags from the erasures are not necessary because they can always be detected by the CSS code). This happens with probability $[S(2t - t_c; n, q)]^{(n-t_c)(r+1)^2}$, as there are $n - t_c$ branches and $(r+1)^2$ trees. Consequently, the first part of the soundness statement (Theorem 4.1.2) and the whole completeness statement (Theorem 4.2.1) of the ESA model hold for the DSA model at least with this probability.
The second part of the soundness statement involves sending secret shares to the reconstructor. Therefore, besides the previous assumption, we also require that the leaves do not suffer an excess of depolarizations while traveling to R. Then, this part of the statement holds with probability at least $[S(2t - t_c; n, q)]^{(n-t_c)(r+1)^2} \cdot [S(t - t_c; n, q)]^n$ (see the proof of Theorem 4.3.1 for the derivation of the second term).

4.4. Efficient qubit sharing

Throughout this chapter we have studied the VHSS protocol under different noise models. We have also proposed several changes that improve its performance in the presence of noise. In this section, we propose a final modification that makes the protocol more robust against any of the noise models proposed so far.
One of the most often recurring assumptions in this work is that qubits can experience an erasure or a depolarization at random only when traveling from node to node. Hence, it is obvious that, the less qubits that are shared, the less qubits that can suffer from noise.
The protocol upgrade that we propose here consists in taking advantage of the previous remark and performing the leaf distribution (step 4 from the original protocol, see Subsection 3.2.1) in a different way:

4_{EFF}. (Leaf encoding and distribution)

4_{EFF}.1. Each node i encodes its share $\Phi_i^{0,0}$ into an n-qubit state $\Phi_{i_{[1,n]}}^{0,0}$ using the code $\mathcal{C}$.

4_{EFF}.2. Each node i sends qubit $\Phi_{i_j}^{0,0}$ to node $[(i+j) \bmod n] + 1$, for $j = 0, \ldots, n-(t-t_c)-2$, while keeping $\Phi_{i_{[n-(t-t_c),n]}}^{0,0}$.

Recall that n is the number of nodes, t_c is the maximum number of cheaters, and t is the maximum number of correctable errors of the underlying CSS code.
The distribution of leaf ancillas (steps 6, 10, and 12) must be modified in the same way, but we omit it here for simplicity.
The rest of the protocol also needs to be accordingly modified, so that the operations to be applied on certain qubits (CNOTs and measurements) are the same but they are performed by the new holder. For example, in the original VHSS, shares $\Phi_{1_2}^{0,0}$, $|\bar{+}\rangle_{1_2}^{0,m}$, and $|\bar{0}\rangle_{1_2}^{l,m}$ are held by node 2 after the sharing stage. However, they can now be held by node 1, who is the new responsible for the operations to be performed on these qubits.
We will refer to a VHSS protocol that implements all the previous modifications as VHSS with *efficient sharing*.

Figure 4.3 shows an example of the leaf shares distribution when running the VHSS in a network with $n = 7$, $t = 3$, and $t_c = 2$ (note that this scheme is not valid since there does not exist any CSS code compatible with such parameters, but it produces a clear graphical representation). In the original protocol, each node i would encode its branch share Φ_i into n leaf qubits and send one of them to each node (Subfigures **(a)** and **(b)**). This way, $n-1 = 6$ qubits are prone to noise. Using the efficient sharing (subfigures **(c)** and **(d)**), each node i keeps $1+t-t_c = 2$ of the encoded qubits Φ_{i_j} and only $n-(1+t-t_c) = 5$ shares are transmitted to other nodes.

Due to the symmetry in the new distribution of leaf shares, every node holds the same total number of qubits as in the original VHSS. Hence, the number of quantum resources required is kept constant.

Let us now show the effects of this modification over the security statements of the erasure models from Section 4.2. The VHSS-ESA PROTOCOL reduces to any of the other schemes (VHSS-ES PROTOCOL and VHSS-CE PROTOCOL) if the network is described by the corresponding noise model. In the absence of noise, it reduces to the original protocol by Lipinska et al. [14]. Then, we take the VHSS-ESA PROTOCOL as reference, since this is the most general scheme and the results will also apply to the ES and CE noise models as well as to the original protocol.
On the one hand, if one sets $t_c = t$, the only effect of the efficient sharing is that nodes are labeled in a different order. Each node would keep one of its leaf shares and send the rest of them to the other nodes. Consequently, all the security statements

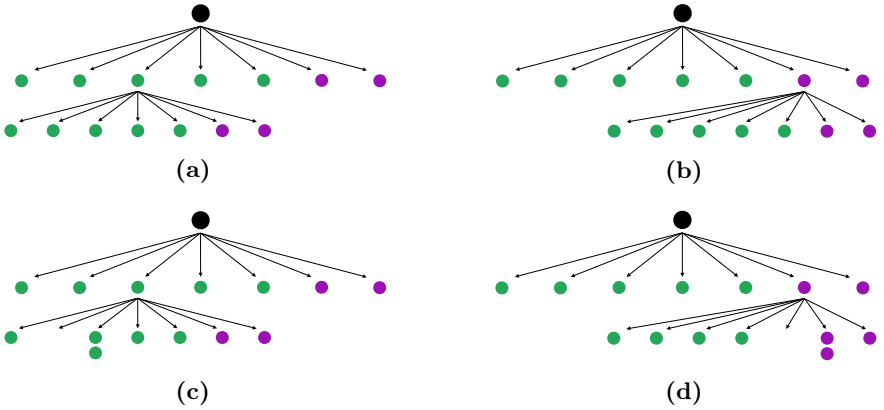


Figure 4.3. **(a-b)** Sketch of the leaf sharing in the VHSS protocol (see Section 3.2) using a CSS code of length $n = 7$ that tolerates up to $t = 3$ errors (note that this code does not exist, we only use it to have a clear graphical representation). Black circles represent qubits to be shared, e.g., the encrypted quantum secret Φ. These initial qubits are encoded into n-qubit states $\Phi_{[1,n]}$, and each share is sent to a different node. The arrows represent nodes 1 to 7, labeled from left to right. Green/purple circles correspond to qubits held by honest/cheating nodes (we assume that nodes 6 and 7 are cheaters). Each qubit Φ_i is encoded into n further qubits and distributed in the same way. Subfigures **(a)** and **(b)** show the leaf encoding of Φ_3 and Φ_6, respectively.
(c-d) Same as in **(a-b)** but using the efficient leaf distribution (see main text, Section 4.4). In **(c)**, node 2 does not hold any leaf share and node 3 holds two of them. In **(d)**, node 5 does not hold any leaf qubit while node 6 holds two of them.

hold. On the other hand, if $t_c < t$, each node keeps $1 + t - t_c$ of its own leaf shares. Let us assume that node i is honest and j is a cheater. After step 4_{NEW}, the whole set of cheaters holds at most t_c leaf shares from i and t leaf shares from j.
The sets B_i and B_j are only taken into account as long as $|B_i| \leq t$ and $|B_j| \leq t$. Therefore, they will always be updated properly in the verification, since erasures are flagged and cheaters are not able to introduce more than t errors in the leaf shares of each branch (we would not be able to detect the errors if there were more than $2t$ in total).
Moreover, the cheaters cannot introduce more than t errors after verification in any of the branches, and the sets B_i and B_j can still be updated correctly in the reconstruction stage. Hence, the proof of Theorem 4.1.2 (soundness) remains valid when using the VHSS-ESA PROTOCOL and the efficient sharing.
With the efficient sharing, less qubits are transmitted, and therefore the probability that at most $2t - t_c$ branches suffered from errors right after verification is larger, yielding

$$f_v^{\text{EFF}}(n, t, q, t_c, t_d) \geq f_v(n, t, q, t_c, t_d), \tag{4.4.1}$$

where f_v is the probability that at most $2t - t_c$ branches suffered from errors assuming that all cheaters introduced errors in their shares (see Eq. (4.1.3)), and f_v^{EFF} corresponds to the same probability when the efficient sharing is used. The completeness statement of the VHSS-ESA PROTOCOL, Theorem 4.2.1, can be derived in the same way when using the efficient sharing. However, instead of using f_v as a lower bound

for the probability that an honest dealer passes the verification phase, one can use $f_{\mathrm{v}}^{\mathrm{EFF}}$.
The efficient sharing was proposed in the latest stages of this thesis, so we do not compute $f_{\mathrm{v}}^{\mathrm{EFF}}$ explicitly. Nevertheless, the true advantage of this new distribution of qubits relies in this function: if one uses the efficient sharing, an honest dealer will be more likely to pass verification, in general.[4]
The secrecy statement, Theorem 4.1.1, also holds when using the efficient sharing, as the proof is based on the underlying VCSS protocol [14].

Regarding the depolarization models from Section 4.3, a similar analysis can be done. The only effect of the efficient sharing over the security statements is that the probability that an honest dealer passes verification is larger.

[4] In some cases, e.g., for $t_{\mathrm{c}} = t$, the probability that an honest dealer passes verification does not increase by using the efficient sharing. In any case, this probability never decreases by using it.

5

Approximate quantum error correction and verifiable quantum secret sharing

In this chapter, we assume that qubits are noiseless, and we focus on schemes that aim at increasing the maximum number of cheaters that can be tolerated in the VQSS task. The main idea behind these protocols is to use the concepts from approximate quantum error correction discussed in Section 2.3. Let us summarize them here.
Quantum authentication schemes (QAS) ensure that a quantum state has not been modified, up to a certain error. Assume that a quantum state Φ has been encoded with an $[[n, k, d]]$ quantum code $\mathcal{C}$, yielding an n-qubit state $\Phi_{[1,n]}$. If we use a QAS to authenticate each of the resulting qubits, the location of arbitrary errors can be identified: the QAS will reject qubits that have suffered from errors (except with a small probability). Hence, arbitrary errors raise a flag and they can be treated as erasures, so the code can tolerate up to $d - 1 \leq \lfloor \frac{n-1}{2} \rfloor$ instead of up to $\lfloor \frac{n-1}{4} \rfloor$. The main drawback of this approach is that the fidelity between the shared secret and the reconstructed state cannot be as high as in the VQSS [13] and VHSS [14] protocols discussed in Chapter 3.
This line of research has already been explored. Ben-Or et al. [7] propose a protocol for VQSS that can tolerate up to $\lfloor \frac{n-1}{2} \rfloor$ cheaters, achieving a fidelity exponentially close to 1 in some security parameters. However, this scheme requires a large number of qubits to be realized. Specifically, each node needs a workspace of at least $\Omega(r^2 n \log(n))$ qubits, where r is a security parameter.
Figure 5.1 shows a comparison of all these protocols in terms of node workspace size, maximum number of cheaters, and fidelity between the original secret and the reconstructed state.

Throughout this chapter, we summarize our efforts to design a protocol that keeps

the number of quantum resources as low as in the VHSS but increases the maximum number of cheaters up to $\lfloor \frac{n-1}{2} \rfloor$. First, in Section 5.1, we review the protocol proposed in [7], which requires a large number of qubits per node. In Section 5.2, we discuss why the VHSS protocol cannot be trivially modified by using a QAS to increase the number of cheaters. Then, we propose the *verifiable trap secret sharing* (VTSS), a new protocol based on the VHSS from [14] that achieves the desired functionalities by making use of the trap code from [40]. We present this section as a discussion and we leave formal proofs as an open question. In addition, we highly recommend the reader to review the concepts explained in Section 2.3 before continuing with this chapter, as we constantly use them.

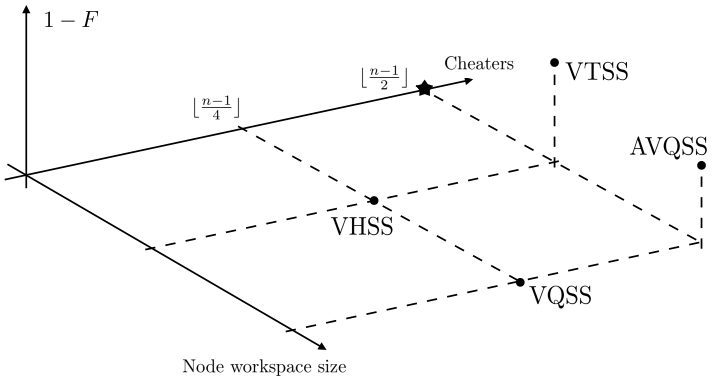


Figure 5.1. Qualitative comparison between the VQSS protocols discussed in this thesis. We assume that qubits do not suffer from noise. In the design of these protocols, three parameters play a key role: the node workspace size, the maximum number of cheaters that are tolerated, and the fidelity F between the shared secret and the reconstructed state.
An optimal protocol (indicated with a star in the diagram) would maximize these three parameters. The protocols placed in the sketch are the following. VQSS: protocol by Crépeau et al. [13]. VHSS: protocol by Lipinska et al. [14]. AVQSS: protocol by Ben-Or et al. [7]. VTSS: VTSS PROTOCOL, proposed in Section 5.2. Note that the fidelity of the VQSS and VHSS is not 1, but $1 - 2^{\Omega(r)}$, where r is the security parameter.

5.1. Approximate verifiable quantum secret sharing

One of the early proposals of a VQSS scheme tolerating up to $\lfloor \frac{n-1}{2} \rfloor$ cheaters is the one from [7]. This protocol relies on the fact that a QAS can be used to create an authenticated quantum channel between each pair of nodes. Let us briefly review the protocol.
They start by defining a subroutine for weak quantum secret sharing (WQSS). The difference between WQSS and VQSS is that, in the former, the dealer can prevent the nodes from reconstructing the secret at any time. The core of their WQSS consists in sharing authenticated zero states between the dealer and each node, that are later used to generate shared EPR pairs. Then, the dealer can teleport an encoded state

to the nodes using these EPR pairs. For a detailed description of this procedure, see the Protocol Zero-Share and Appendix B from [7].
The whole protocol, to which we refer as AVQSS, consists in the following general steps:

1. D initializes $\Omega(r)$ qu*p*its in the state $|0\rangle$, where r is a security parameter. Then, D encodes them using a quantum code of length n and sends the i-th share to node i using WQSS.

2. For each share received from D, each node initializes $\Omega(r)$ qu*p*its in the state $|0\rangle$. Then, it encodes them using a quantum code of length n and sends the i-th share to node i using WQSS.

3. Some quantum and classical operations are applied to conclude the verification stage (see Appendix C from [7]).

4. Nodes generate shared EPR pairs and send the half of each pair to the dealer.

5. D shares the secret with the nodes using the EPR pairs.

6. For reconstruction, a similar procedure is performed between the nodes and the reconstructor R: they create shared EPR pairs which are used to teleport their secret shares to R.

Note that the previous steps are missing relevant details that are not necessary for the upcoming discussion (see [7] for a complete description of the protocol).
During the first steps of the protocol, each node receives $\Omega(r)$ shares from the dealer. Then, it generates $\Omega(r)$ zeros and encodes them into n further shares. Each share has size $\Omega(\log(n))$, since they work with qu*p*its, and $p > n$. Consequently, this scheme requires a node workspace size of $\Omega(r^2 n \log(n))$.
In the spirit of creating protocols for early quantum networks in which the quantum resources are limited, redesigning this AVQSS protocol to use less qubits and achieve the same functionalities did not seem feasible within the timeframe of this thesis. Hence, we adopted a different approach to the problem and decided to work towards increasing the number of tolerable cheaters in the VHSS.

5.2. Verifiable trap secret sharing

A naive approach to the problem would be to directly substitute the CSS code employed in the VHSS protocol from [14] by an AQECC. Let us assume that the first encoding is still performed by the dealer using an $[[n, 1, d]]$ CSS code. Then, the nodes employ an AQECC for the leaf encoding, instead of using the same CSS code. By using an AQECC in the second encoding, we can lift the number of errors tolerated in the leaves from $\lfloor (d-1)/2 \rfloor$ to $d-1$. Nevertheless, before analyzing whether the verification and reconstruction stages would work in the same way as in the original protocol, we already find an important drawback, namely the large number of qubits that each node must handle.
Figure 5.2 shows the structure of this type of sharing. The dealer encodes a state

Φ into $\Phi_{[1,n]}$. Then, each branch share Φ_i is further encoded into $\Phi_{i_{[1,n]}}$ using an $[[n, 1, d]]$ non-degenerate stabilizer code. Finally, each leaf share Φ_{i_j} is authenticated, yielding $A(\Phi_{i_j})$. These final states are composed by $1 + \tau$ qubits, where τ is the number of tag qubits. Consequently, each node holds $n \cdot (1 + \tau)$ qubits.

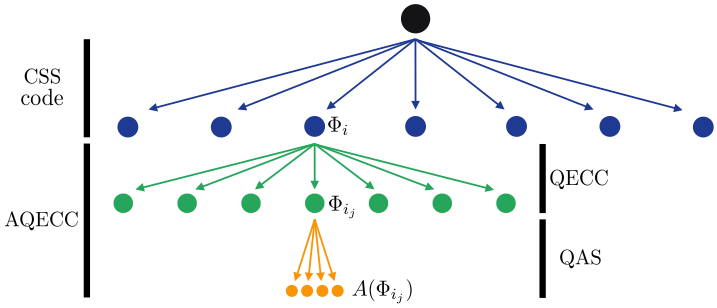


Figure 5.2. Sketch of the encoding in a VHSS scheme in which the second encoding is performed by an AQECC (see main text). The black circle represents a quantum state to be shared, e.g., the encrypted secret. Blue and green circles correspond to branch and leaf shares, respectively. Orange circles represent an authenticated leaf share—note that the number of qubits employed in this graphical representation does not necessarily correspond to a valid QAS.

The value of τ depends on the fidelity that one wants to achieve. Nevertheless, to obtain reasonable errors, large values of τ are required. As an example, consider the efficient family of codes (see Section 2.3). Using Eqs. (2.3.3) and (2.3.5), we can compute the error of the resulting AQECC:

$$\varepsilon_{\text{AQECC}} = 2n^2\varepsilon = 2n^2 \frac{2(m+\tau)}{\tau(2^\tau + 1)} \approx \frac{4n^2}{2^\tau}, \tag{5.2.1}$$

where we can do this approximation since $\tau \gg 1$. In order to keep this error small (it should be at least less than 1), τ must grow faster than $\log n$. Then, the number of qubits per node scales with n at least as fast as in the AVQSS from [7], so we would not be able to achieve the desired reduction of quantum resources.
If we used a set of trap codes based on an $[[\tilde{n}, 1, \tilde{d}]]$ CSS code instead of the efficient family, the error of the whole AQECC would be given by

$$\varepsilon_{\text{AQECC}} = 2n^2\varepsilon = 2n^2 \left(\frac{2}{3}\right)^{\tilde{d}/2} \geq 2n^2 \left(\frac{2}{3}\right)^{\tilde{n}/4}, \tag{5.2.2}$$

where we have used Eqs. (2.3.3) and (2.3.7), and the fact that $\tilde{d} \leq \lfloor \frac{\tilde{n}-1}{2} \rfloor$. In order to have $\varepsilon_{\text{AQECC}} < 1$ for small networks ($n < 100$), $\tilde{n}$ must grow faster than n, yielding a total of $n \cdot 3\tilde{n} > 4n^2$ number of qubits per node. Following the same reasoning as before, this scheme is not valid since the number of quantum resources scales even faster than in the AVQSS.

The conclusion of the previous discussion is that incorporating an AQECC to the protocol means that an extra level of encoding is required, due to the need for

authentication with a QAS. This implies a dramatical increase in the number of qubits per node. A cheaper alternative can be designed by using part of the AQECC instead of the whole construction. Hence, we take a similar approach to the previous one but without making use of a full AQECC: we employ a trap code for the second level of encoding, which acts as a QAS for each of the branch qubits (see Figure 5.3). We call this new proposal *verifiable trap secret sharing* protocol (VTSS PROTOCOL).

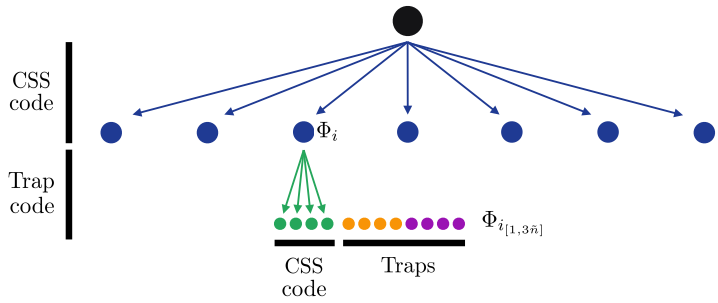


Figure 5.3. Sketch of the qubit encoding in the VTSS PROTOCOL, in which the second encoding is performed by a trap code (see main text). The black circle represents a quantum state to be shared, e.g., the encrypted secret. Blue circles correspond to shares encoded with the first CSS code. Green circles represent shares encoded using the second CSS code. Orange and purple circles correspond to the traps that detect errors in the X and the Z basis, respectively. The permutation of encoded qubits and traps is not represented here. Note that the number of qubits employed in this graphical representation does not necessarily correspond to a valid trap code.

The requirements for the VTSS scheme are the same as for the VHSS from [14], namely, we assume that the nodes have access to an authenticated classical broadcast channel and a public randomness source, and that each pair of nodes is connected by an authenticated classical channel and a quantum channel. In addition, a classical trusted third party (classical-TTP) is needed for the verification stage. This classical-TTP can be replaced by a secure multi-party computation subroutine [3]. Moreover, the protocol makes use of an $[[n, k, d]]$ CSS code $\mathcal{C} = \mathrm{CSS}(V, W)$ and a family of trap codes $\{\mathcal{C}_{k_i}\}_{k_i \in \mathcal{K}}$ based on an $[[\tilde{n}, 1, \tilde{d}]]$ CSS code $\mathrm{CSS}(V_{\mathrm{trap}}, W_{\mathrm{trap}})$. V, W and V_{trap}, W_{trap} are two pairs of classical codes meeting the conditions to generate a CSS code (see Subsection 2.2.1), and the set of keys $\mathcal{K}$ spans all possible permutations of $3\tilde{n}$ elements.

In the VHSS scheme, the dealer can be cheating. In the VTSS protocol, however, we assume that the dealer can be *faulty* but not a cheater: it can introduce errors in the quantum states but it does not collaborate with the cheaters (see Figure 5.4). This can be useful as a framework for distributed systems with consensus problems [8].

We provide a step-by-step description of the VTSS PROTOCOL in the following box:

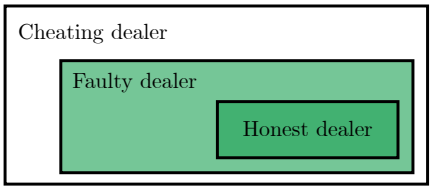


Figure 5.4. The dealer models that we consider in this work are three. A cheating dealer can perform any operation on the qubits and can collaborate with the rest of cheaters. A particular type of cheating dealer is the faulty one, which does not collaborate with other cheaters. An honest dealer follows the protocol honestly and does not introduce any errors. The VTSS protocol assumes that the dealer is faulty (regions highlighted in green).

VTSS PROTOCOL.

Sharing stage.

1. (Key generation) The dealer D generates n random keys $k_1, \ldots, k_n$. Each of them specifies a permutation of $3\tilde{n}$ elements.

2. (Key sharing) D sends all keys to the classical-TTP.

3. (Encryption) The dealer D performs quantum one-time pad on the secret state $|\psi\rangle$ using a classical secret key s to obtain the state $\Phi^{0,0}$.

4. (VCSS) D distributes the classical key s among the n nodes using a verifiable classical secret sharing (VCSS) scheme.

5. (Encoding and distribution)

   5.1. D encodes $\Phi^{0,0}$ into an n-qubit state $\Phi^{0,0}_{[1,n]}$ using the CSS code $\mathcal{C}$.

   5.2. D encodes each qubit $\Phi^{0,0}_i$ using the trap code $\mathcal{C}_{k_i}$, yielding $\Phi^{0,0}_{i_{[1,3\tilde{n}]}}$.

   5.3. D sends $\Phi^{0,0}_{i_{[1,3\tilde{n}]}}$ to node i.

Verification stage.

for $m = 1, \ldots, r$:

6. (Preparation and encoding - Z basis) The dealer D prepares an n-qubit ancillary state $|\bar{+}\rangle^{0,m}_{[1,n]} = \sum_{v\in V}|v\rangle$. Then, it encodes each qubit $|\bar{+}\rangle^{0,m}_{i}$ using $\mathcal{C}_{k_i}$, yielding $|\bar{+}\rangle^{0,m}_{i_{[1,3\tilde{n}]}}$.

7. (Distribution - Z basis) D sends $|\bar{+}\rangle^{0,m}_{i_{[1,3\tilde{n}]}}$ to node i.

8. (CNOTs - Z basis) A public random bit $b_{0,m}$ is generated. Then, each node i applies the following operation to their shares:

$$\mathrm{CNOT}^{b_{0,m}}(\Phi^{0,0}_{i_j}, |\bar{+}\rangle^{0,m}_{i_j}),\ \forall j = 1,\dots,3\tilde{n}, \tag{5.2.3}$$

   i.e., a CNOT is applied if $b_{0,m} = 1$.

9. (Measurements - Z basis) Each node i measures the $3\tilde{n}$ ancillary states it is holding (systems indexed with m and $l = 0$) in the Z basis, obtaining bit strings $\mathbf{v}^{0,m}_i$, and sends all the measurement outcomes to the classical-TTP. All measured qubits are discarded.

for $l = 1,\dots,r$:

10. (Preparation and encoding - X basis) The dealer D prepares an n-qubit ancillary state $|\bar{0}\rangle^{l,0}_{[1,n]} = \sum_{w\in W^\perp}|w\rangle$. Then, it encodes each qubit $|\bar{0}\rangle^{l,0}_{i}$ using $\mathcal{C}_{k_i}$, yielding $|\bar{0}\rangle^{l,0}_{i_{[1,3\tilde{n}]}}$.

11. (Distribution - X basis) D sends $|\bar{0}\rangle^{l,0}_{i_{[1,3\tilde{n}]}}$ to node i.

    for $m = 1,\dots,r$:

    12. (Preparation and encoding - $l, m \neq 0$) The dealer D prepares an n-qubit ancillary state $|\bar{0}\rangle^{l,m}_{[1,n]} = \sum_{w\in W^\perp}|w\rangle$. Then, it encodes each qubit $|\bar{0}\rangle^{l,m}_{i}$ using $\mathcal{C}_{k_i}$, yielding $|\bar{0}\rangle^{l,m}_{i_{[1,3\tilde{n}]}}$.

    13. (Distribution - $l, m \neq 0$) D sends $|\bar{0}\rangle^{l,m}_{i_{[1,3\tilde{n}]}}$ to node i.

    14. (CNOTs - $l, m \neq 0$) A public random bit $b_{l,m}$ is generated. Then, each node i applies the following operation to their shares:

$$\mathrm{CNOT}^{b_{0,m}}(|\bar{0}\rangle^{l,0}_{i_j}, |\bar{0}\rangle^{l,m}_{i_j}),\ \forall j = 1,\dots,3\tilde{n}, \tag{5.2.4}$$

        i.e., a CNOT is applied if $b_{l,m} = 1$.

15. (Measurements - $l, m \neq 0$) Each node i measures its $3\tilde{n}$ ancillary states indexed with l, m in the Z basis, obtaining bit strings $\mathbf{v}_i^{l,m}$, and sends all measurement outcomes to the classical-TTP. All measured qubits are discarded.

16. (Fourier transform) Each node i applies the Fourier transform $\mathcal{F}$ to its remaining shares, obtaining $\Phi_{i_j}^{\mathcal{F},0,0}$ and $|\bar{0}\rangle_{i_j}^{\mathcal{F},l,0}$, $j = 0, \ldots, 3\tilde{n}$.

17. (CNOTs - X basis) A public random bit $b_{l,0}$ is generated. Then, each node i applies the following operation to their shares:

$$\mathrm{CNOT}^{b_{l,0}}(\Phi_{i_j}^{\mathcal{F},0,0}, |\bar{0}\rangle_{i_j}^{\mathcal{F},l,0}),\ \forall j = 1, \ldots, 3\tilde{n}, \tag{5.2.5}$$

    i.e., a CNOT is applied if $b_{l,0} = 1$.

18. (Measurements - X basis) Each node i measures its $3\tilde{n}$ ancillary states indexed with l and $m = 0$ in the Z basis, obtaining bit strings $\mathbf{v}_i^{l,0}$, and sends all measurement outcomes to the classical-TTP. All measured qubits are discarded.

19. (Measurement permutations) The classical-TTP permutes each $\mathbf{v}_i^{l,m}$ according to k_i, yielding strings $\hat{\mathbf{v}}_i^{l,m}$.

20. (Check traps) For each i, the classical-TTP checks whether any of the last $2\tilde{n}$ bits of any $\hat{\mathbf{v}}_i^{l,m}$ is 1, i.e., if any trap has been triggered. If a trap has been triggered, then it includes i in the set of apparent cheaters B: $B \leftarrow B \cup \{i\}$.

21. (Leaf decoding - Z basis) For each i, the classical-TTP checks whether any $\hat{\mathbf{v}}_i^{l,m}$, $m \neq 0$, does not correspond to a codeword from V_{trap}. If this happens, then $B \leftarrow B \cup \{i\}$ and the classical-TTP creates a variable $a_i^{l,m} = \perp$. Otherwise, it classically decodes all the strings and saves the decoded values $a_i^{l,m}$.

22. (Branch decoding - Z basis) Decoded values $a_i^{l,m}$ form words $\mathbf{a}_{l,m}$, which should correspond to codewords from V. If an error occurred at position k of $\mathbf{a}_{l,m}$, then $B \leftarrow B \cup \{k\}$.

23. (Leaf decoding - X basis) For each i, the classical-TTP checks whether any $\hat{\mathbf{v}}_i^{l,0}$ does not correspond to a codeword from W_{trap}. If this happens, then $B \leftarrow B \cup \{i\}$ and the classical-TTP creates a variable $a_i^{l,0} = \perp$. Otherwise, it classically decodes all the strings and saves the decoded values $a_i^{l,m}$.

24. (Branch decoding - X basis) Decoded values $a_i^{l,0}$ form words $\mathbf{a}_{l,0}$, which should correspond to codewords from W. If an error occurred at position k of $\mathbf{a}_{l,0}$, then $B \leftarrow B \cup \{k\}$.

25. (Abort condition) The classical-TTP publicly announces the size of B. If $|B| > 2t$, reject the dealer and abort the protocol. Otherwise, continue.

26. (Inverse Fourier transform) Nodes apply an inverse Fourier transform to revert the transform from step 11. Each node i holds again $\Phi^{0,0}_{i_{[1,3\bar{n}]}}$.

Reconstruction stage.

22. (Send to R) All nodes send their quantum and classical shares to the reconstructing node R. The classical-TTP also sends all the keys to R.

23. (Reconstruct s) R reconstructs the classical encryption key s following the VCSS scheme.

24. (Check traps) For each $i \notin B$, R decodes $\Phi^{0,0}_{i_{[1,3\bar{n}]}}$ using $\mathcal{C}_{k_i}$. If any of the traps is triggered or an error is detected, then $B \leftarrow B \cup \{i\}$. If $|B| > 2t$, R aborts the protocol.

25. (Reconstruct $\Phi^{0,0}$) For $i \notin B$, R takes $n - 2t$ shares $\Phi^{0,0}_i$ at random and applies an erasure-recovery circuit to obtain $\Phi^{0,0}$.

26. (Decrypt) R decrypts $\Phi^{0,0}$ using s to obtain the original quantum state $|\psi\rangle$.

The main differences between the VTSS PROTOCOL and the VHSS protocol from [14] are the following:

- A set of random keys $k_1, \ldots, k_n$ is required. Therefore, the first step (key generation) is new with respect to the VHSS protocol.
- The second level encoding (step 4 from the VHSS scheme in Subsection 3.2.1 and step 5.2 from the VTSS PROTOCOL) is now performed by the dealer instead of the nodes, using a trap code instead of a CSS code.
- The nodes do not check their measurements outcomes. Instead, a classical-TTP with access to the keys $k_1, \ldots, k_n$ does it for them, since these keys are necessary for decoding.

As previously stated, the trap code acts as a QAS on the branch shares Φ_i. It is important to note that the trap code is particularly suitable for this protocol since the CNOT and the Fourier transform remain transversal—other QAS may not preserve this property.
Moreover, the trap code is secure as long as the key is not recycled [35]. This comes from the fact that the trap code is plaintext authenticating, meaning that the probability that a Pauli alters the message but leaves the tag qubits untouched is very small (see Subsection 2.3.1). However, certain attacks may find out the position

of some traps by applying a Pauli and checking whether the message is accepted or rejected. This way, they can retrieve partial information about the key. This problem can be solved by using a different key for each qubit that is authenticated, although this is not a desirable strategy if one wants to perform computations on simultaneously authenticated qubits, as they must be encoded under the same key.[1] There exist other QASs similar to the trap code that are ciphertext authenticating, i.e., they enable key-recycling [35]. However, in the VTSS PROTOCOL, we do not employ a ciphertext authenticating scheme, as it is not an essential property. Let us discuss why. Each branch share Φ_i is encoded using a trap code with a different permutation key k_i. Then, in the verification stage, the ancillary qubits are authenticated in the same way. If a cheating node i followed the strategy of applying a Pauli to any of the encoded qubits to obtain information about the key, this would be reflected in the measurement outcomes (except with a probability exponentially small in $\tilde{d}$). Then, the state $\Phi_{i_{[1,3\tilde{n}]}}$ would be rejected, leaking some information to node i but also leaving it out of the protocol.

5

The reason for assuming a faulty dealer instead of a cheating dealer becomes apparent after going through the protocol. If the dealer shared the permutation keys of the cheaters, $\{k_i\}_{i\in C_{\text{cheat}}}$[2], with them, they could easily introduce logical errors without being detected by the traps. In order to prevent this, we must assume that the dealer does not collaborate with the cheaters, although it can still introduce arbitrary errors in the quantum states.

To conclude, we present the security statements of the VTSS PROTOCOL. As stated before, we give informal proofs for them (except for the secrecy statement), while formal proofs remain an open question.

Theorem 5.2.1 (Secrecy)**.** *In the* VTSS PROTOCOL*, when D is honest and there are at most t_c active cheaters in the verification phase, no group of at most p_c nodes learns nothing about D's secret state throughout the protocol, where p_c is the secrecy of the underlying classical scheme, except with probability exponentially small in the security parameter r.*

Proof. In the VTSS PROTOCOL, the dealer encrypts its quantum state with a one-time pad before encoding it, in the same way as in the original VHSS scheme from [14]. Then, the encryption key is shared using a VCSS subroutine. Therefore, the proof of their security statement also applies to our protocol, since it is based on the security of the underlying VCSS scheme. □

Claim 5.2.1 (Soundness)**.** *In the* VTSS PROTOCOL*, either the nodes hold a consistently encoded secret or the protocol aborts, with probability at least $1-2^{-\Omega(r)}-\varepsilon^{\Omega(r)}$,*

[1] This is actually the motivation for Dulek and Speelman [35] to create a "strong trap code", which is ciphertext authenticating and yields a key-recycling QAS.
[2] Recall that C_{cheat} is a set containing the labels of the cheating nodes.

where $\varepsilon \leq (\frac{2}{3})^{\tilde{d}/2}$ *is the error of the trap code employed. Moreover, after passing verification and sending all qubits to an honest* R*, either the shares are still consistent or the protocol aborts, with probability at least* $1 - 2^{-\Omega(r)} - 2\varepsilon^{\Omega(r)}$.

Proof (informal). The first part of the statement can be rephrased as *the probability that the nodes pass verification holding inconsistent shares is at most* $2^{-\Omega(r)} + \varepsilon^{\Omega(r)}$. The branch shares $\Phi_i^{0,0}$ are authenticated before being sent to the nodes. Since the nodes do not know the permutation keys k_i, a cheater j can introduce errors in $\Phi^{0,0}_{j_{[1,3\tilde{n}]}}$ without being detected by the traps with probability at most $\varepsilon^{\Omega(r)}$. This ε is an upper bound for the probability of introducing an error in the message qubits and not in the traps (see Subsection 2.3.2). The exponential dependence on $\Omega(r)$ comes from the fact that we employ $\Omega(r)$ ancillas.
If the dealer introduces errors in the branch shares, this will be detected by the ancillas in the same way as in the original VHSS scheme. However, in our protocol, D can also tamper with the authenticated shares $\Phi^{0,0}_{i_{[1,3\tilde{n}]}}$. If D introduces an error in the message qubits, it could also introduce an error in the ancillas such that they always cancel out after applying the CNOT. However, the random bit that determines whether the CNOT is applied or not is drawn after the corresponding ancilla has been distributed. Therefore, these errors will not be detected with probability at most $2^{-\Omega(r)}$, as in the original VHSS protocol.
Combining both previous results, the probability that an error is introduced and not detected is at most $2^{-\Omega(r)} + \varepsilon^{\Omega(r)}$.

Let us now consider the second part of the statement. It can be reformulated as *the probability that* R *holds inconsistent shares and does not abort is at most* $2^{-\Omega(r)} + 2\varepsilon^{\Omega(r)}$. This can only happen in two situations: if the nodes pass verification holding inconsistent shares (event X) or if the nodes pass verification holding consistent shares but the cheaters manage to introduce some errors in their qubits without being detected by traps (event Y). X happens with probability at most $2^{-\Omega(r)} + \varepsilon^{\Omega(r)}$, as previously discussed. Regarding Y, the nodes pass verification holding consistent shares (event Y_1), and then cheaters introduce errors without falling into the traps (event Y_2) with probability $\varepsilon^{\Omega(r)}$. Since events X and Y cannot happen simultaneously and Y_1 and Y_2 are independent, the probability that R holds inconsistent shares is given by:

$$\begin{aligned} \Pr(X) + \Pr(Y) &\leq (2^{-\Omega(r)} + \varepsilon^{\Omega(r)}) + \Pr(Y_1)\Pr(Y_2) \\ &\leq (2^{-\Omega(r)} + \varepsilon^{\Omega(r)}) + \Pr(Y_2) \\ &= (2^{-\Omega(r)} + \varepsilon^{\Omega(r)}) + \varepsilon^{\Omega(r)} \\ &= 2^{-\Omega(r)} + 2\varepsilon^{\Omega(r)}. \end{aligned} \tag{5.2.6}$$

□

Claim 5.2.2 (Completeness)**.** *In the* VTSS PROTOCOL*, if* D *is honest then it passes the verification phase. Moreover, if* R *is also honest, it reconstructs* D*'s secret with probability at least* $1 - \varepsilon^{\Omega(r)}$.

Proof (informal). The probability that an honest dealer does not pass verification corresponds to the probability that $|B| > 2t$. However, since there are at most $2t$ cheaters, they cannot introduce any errors in more than $2t$ branch shares $\Phi_i^{0,0}$. Consequently, D passes verification with probability 1. If R is also honest, it reconstructs D's secret unless a cheater manages to introduce an error in its shares without being detected by the traps. This happens with probability at most $\varepsilon^{\Omega(r)}$, as discussed in the proof of Theorem 5.2.1. Therefore, D's secret can be reconstructed with probability at least $1 - \varepsilon^{\Omega(r)}$. □

6

Conclusions and outlook

Most of the verifiable quantum secret sharing (VQSS) protocols proposed so far, e.g., the ones from [13] and [14], are proven to be secure in ideal networks with a fixed set of cheating nodes. Throughout this thesis we have explored two directions in which these schemes can be improved, namely, the amount of noise they can tolerate and the maximum number of cheaters. Moreover, we did this while keeping the required node workspace size as low as possible. For this reason, the core of our work has been focused on verifiable hybrid secret sharing (VHSS) schemes, which combine VQSS with a classical encryption and classical secret sharing to reduce the number of quantum resources.

In Chapter 4, we analyzed the performance of the VHSS from [14] when it is run on noisy networks. First, we proposed several noise models in which qubits can be erased while traveling from node to node but do not suffer from stochastic arbitrary errors. The fact that erasures are flagged was used to modify the original protocol, obtaining new schemes tailored to the needs of each model (VHSS-ES protocol, VHSS-CE protocol, and VHSS-ESA protocol). The soundness and the secrecy of these modified schemes was found to be the same as for the original protocol. The completeness, however, was found to be slightly weaker, although this was already expected, since an honest dealer may not be able to pass verification when errors can happen at random on any qubit. Moreover, we derived a function f_{v} (Eq. 4.1.3) that corresponds to the probability of having a number of arbitrary errors that still allows for the reconstruction of the secret. As discussed in Section 4.1, this function can be used to tune the parameters of the protocol in order to maximize the probability of successfully sharing a secret in a quantum network that can lose qubits with certain probability.
A similar analysis was performed for several noise models in which qubits could be depolarized instead of erased. In this case, errors are not flagged and it is not possible to abort the protocol whenever too many qubits have been depolarized. Consequently, the security statements derived for the depolarization models are

noticeably weaker, as the protocol can fail in many ways, e.g., there is a nonzero probability that a logical error is introduced after the verification stage.
Furthermore, in all the previous noise models, qubits suffer from noise while they are traveling between nodes. To alleviate this, we designed a new sharing stage for the VHSS protocol that achieves the same functionalities while sending less qubits around (see Section 4.4).

Lastly, in Chapter 5, we took a different approach to the problem. We considered again noiseless networks and worked towards increasing the maximum number of cheaters tolerated in the VHSS protocol. We proposed the VTSS PROTOCOL, which raises the maximum number of tolerable cheaters from $\lfloor \frac{n-1}{4} \rfloor$ up to $\lfloor \frac{n-1}{2} \rfloor$, making use of a quantum authentication scheme based on the trap code from [40]. This chapter is presented as an open discussion in which some formal proofs are still required.

Future work still has to be done in the same directions. First, our noise models do not take into account the noise present in the quantum memories. A complete description of the physical layout must consider both the quantum noise introduced by these memories and by the communication channels. Therefore, this thesis can serve as the groundwork for a more advanced noise analysis. On the other hand, the use of quantum authentication schemes and approximate error correction for VQSS is a promising line of research that enables the presence of up to $\lfloor \frac{n-1}{2} \rfloor$ cheaters. More efficient approximate VQSS protocols in terms of computational cost and quantum resources than the current ones, e.g., [7], could be designed.
Finally, the simulation or implementation of the protocols discussed in this thesis in noisy networks would also be an interesting exercise. This could be used to evaluate the validity of the theoretical results and would potentially lead to new tools for improving the performance of the protocols.

References

[1] Benny Chor, Shafi Goldwasser, Silvio Micali, and Baruch Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults. In *26th Annual Symposium on Foundations of Computer Science (sfcs 1985)*, pages 383–395. IEEE, 1985.

[2] Paul Feldman. A practical scheme for non-interactive verifiable secret sharing. In *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*, pages 427–438. IEEE, 1987.

[3] Tal Rabin and Michael Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 73–85. ACM, 1989.

[4] Torben Pryds Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Annual international cryptology conference*, pages 129–140. Springer, 1991.

[5] David Chaum, Claude Crépeau, and Ivan Damgard. Multiparty unconditionally secure protocols. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 11–19, 1988.

[6] Wenliang Du and Mikhail J Atallah. Secure multi-party computation problems and their applications: a review and open problems. In *Proceedings of the 2001 workshop on New security paradigms*, pages 13–22, 2001.

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

[8] Pesech Feldman and Silvio Micali. An optimal probabilistic protocol for synchronous byzantine agreement. *SIAM Journal on Computing*, 26(4):873–933, 1997.

[9] Peter YA Ryan, Steve Schneider, and Vanessa Teague. End-to-end verifiability in voting systems, from theory to practice. *IEEE Security & Privacy*, 13(3):59–62, 2015.

[10] Stephan Ritter, Christian Nölleke, Carolin Hahn, Andreas Reiserer, Andreas Neuzner, Manuel Uphoff, Martin Mücke, Eden Figueroa, Joerg Bochmann, and Gerhard Rempe. An elementary quantum network of single atoms in optical cavities. *Nature*, 484(7393):195, 2012.

[11] Stephanie Wehner, David Elkouss, and Ronald Hanson. Quantum internet: A vision for the road ahead. *Science*, 362(6412):eaam9288, 2018.

[12] Wojciech Kozlowski and Stephanie Wehner. Towards large-scale quantum networks. In *Proceedings of the Sixth Annual ACM International Conference on Nanoscale Computing and Communication*, pages 1–7, 2019.

[13] Claude Crépeau, Daniel Gottesman, and Adam Smith. Secure multi-party quantum computation. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 643–652. ACM, 2002.

[14] Victoria Lipinska, Gláucia Murta, Jérémy Ribeiro, and Stephanie Wehner. Verifiable hybrid secret sharing with few qubits. *arXiv preprint arXiv:1911.09470*, 2019.

[15] Venkatesan Guruswami. Lecture notes for introduction to coding theory 15-859v: Linear codes. [PDF lecture notes]. *Carnegie Mellon University Lecture Notes*, 2010.

[16] Richard W Hamming. Error detecting and error correcting codes. *The Bell system technical journal*, 29(2):147–160, 1950.

[17] Madhu Sudan. Lecture notes for 6.897 algorithmic introduction to coding theory, lecture 2 [pdf lecture notes]. *MIT Lecture Notes*, 2001.

[18] John Preskill. Lecture notes for Physics 219: Quantum computation. Chapter 7. Quantum error correction [PDF lecture notes]. *Caltech Lecture Notes*, 1999.

[19] Liqun Chen. *Cryptography and Coding: 13th IMA International Conference, IMACC 2011, Oxford, UK, December 2011, Proceedings*, volume 7089. Springer Science & Business Media, 2011.

[20] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.

[21] William K Wootters and Wojciech H Zurek. A single quantum cannot be cloned. *Nature*, 299(5886):802, 1982.

[22] Dorit Aharonov. *Noisy Quantum Computation*. PhD thesis, The Hebrew University, 1999.

[23] Stéphane Attal. Lectures on Quantum Noise Theory (online course). Lecture 6: Quantum channels [PDF lecture notes]. 2014.

[24] Mark M Wilde. *From classical to quantum Shannon theory*. arXiv preprint arXiv:1106.1445, 2011.

[25] John Preskill. Lecture notes for Physics 219: Quantum computation. Chapter 3. Foundations of Quantum Theory II: measurement and evolution [PDF lecture notes]. *Caltech Lecture Notes*, 2015.

[26] Peter W Shor. Scheme for reducing decoherence in quantum computer memory. *Physical review A*, 52(4):R2493, 1995.

[27] Emanuel Knill and Raymond Laflamme. Theory of quantum error-correcting codes. *Physical Review A*, 55(2):900, 1997.

[28] Markus Grassl, Th Beth, and Thomas Pellizzari. Codes for the quantum erasure channel. *Physical Review A*, 56(1):33, 1997.

[29] A Robert Calderbank and Peter W Shor. Good quantum error-correcting codes exist. *Physical Review A*, 54(2):1098, 1996.

[30] Andrew M Steane. Error correcting codes in quantum theory. *Physical Review Letters*, 77(5):793, 1996.

[31] Simon J Devitt, William J Munro, and Kae Nemoto. Quantum error correction for beginners. *Reports on Progress in Physics*, 76(7):076001, 2013.

[32] Claude Crépeau, Daniel Gottesman, and Adam Smith. Approximate quantum error-correcting codes and secret sharing schemes. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 285–301. Springer, 2005.

[33] Mark N Wegman and J Lawrence Carter. New hash functions and their use in authentication and set equality. *Journal of computer and system sciences*, 22(3):265–279, 1981.

[34] Jonathan Katz, Alfred J Menezes, Paul C Van Oorschot, and Scott A Vanstone. *Handbook of applied cryptography*. CRC press, 1996.

[35] Yfke Dulek and Florian Speelman. Quantum ciphertext authentication and key recycling with the trap code. *arXiv preprint arXiv:1804.02237*, 2018.

[36] Sumegha Garg, Henry Yuen, and Mark Zhandry. New security notions and feasibility results for authentication of quantum data. In *Annual International Cryptology Conference*, pages 342–371. Springer, 2017.

[37] Gorjan Alagic, Tommaso Gagliardoni, and Christian Majenz. Unforgeable quantum encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 489–519. Springer, 2018.

[38] Frédéric Dupuis, Jesper Buus Nielsen, and Louis Salvail. Actively secure two-party evaluation of any quantum operation. In *Annual Cryptology Conference*, pages 794–811. Springer, 2012.

[39] Howard Barnum, Claude Crépeau, Daniel Gottesman, Adam Smith, and Alain Tapp. Authentication of quantum messages. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pages 449–458. IEEE, 2002.

[40] Anne Broadbent, Gus Gutoski, and Douglas Stebila. Quantum one-time programs. In *Annual Cryptology Conference*, pages 344–360. Springer, 2013.

[41] Anne Broadbent and Evelyn Wainewright. Efficient simulation for quantum message authentication. In *International Conference on Information Theoretic Security*, pages 72–91. Springer, 2016.

[42] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[43] George Robert Blakley. Safeguarding cryptographic keys. In *1979 International Workshop on Managing Requirements Knowledge (MARK)*, pages 313–318. IEEE, 1979.

[44] Michael O Rabin. How to exchange secrets with oblivious transfer. 1981.

[45] M. Hillery, V. Buzek, and A. Berthiaume. Quantum secret sharing, 1998.

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

[47] Adam Smith. *Multi-party quantum computation.* PhD thesis, Massachusetts Institute of Technology, 2001.

[48] Tomohiro Ogawa, Akira Sasaki, Mitsugu Iwamoto, and Hirosuke Yamamoto. Quantum secret sharing schemes and reversibility of quantum operations. *Physical review A*, 72(3):032318, 2005.

[49] Richard Arratia and Louis Gordon. Tutorial on large deviations for the binomial distribution. *Bulletin of mathematical biology*, 51(1):125–131, 1989.

[50] Charles Miller Grinstead and James Laurie Snell. *Introduction to probability.* American Mathematical Soc., 2012.

A

Derivation of $f_{\mathrm{v}}(n, t, q, t_{\mathrm{c}}, t_{\mathrm{d}})$

In this appendix we derive the function $f_{\mathrm{v}}(n, t, q, t_{\mathrm{c}}, t_{\mathrm{d}})$ presented in Section 4.1. Note that the analytical expressions derived here are not used in the main chapters, since the results are computationally demanding and not very reliable when evaluated by brute force, due to the large number of combinatorial factors involved. Therefore, instead of evaluating the form of f_{v} found in this appendix, we designed Algorithm 1 to obtain an estimate which is arbitrarily close to the analytical expression (see Proposition 4.1.1).
We derive the exact value of f_{v} by dividing the problem into subproblems that are easier to solve. We do this by employing the law of total probability (LTP) [50], which states that, if B_n, with $n = 1, 2, 3, \ldots, N$, are pairwise disjoint events whose union is the entire sample space, then

$$\Pr(A) = \sum_{n=1}^{N} \Pr(B_n) \cdot \Pr(A|B_n), \tag{A.0.1}$$

for any event A of the same probability space.

Recall that $f_{\mathrm{v}}(n, t, q, t_{\mathrm{c}}, t_{\mathrm{d}}) := \Pr(|C_{\mathrm{v}}| \leq 2t - t_{\mathrm{c}} \mid \text{max. cheats})$. This is the probability that $|C| \leq 2t - t_{\mathrm{c}}$ right after verification, assuming that all cheaters introduced errors in their shares.
We start by using the LTP to write f_{v} in the following form:

$$f_{\mathrm{v}}(n, t, q, t_{\mathrm{c}}, t_{\mathrm{d}}) = \sum_{x=0}^{2t-t_{\mathrm{c}}} \sum_{y=0}^{3tn-2t^2} \Big[\Pr(u^{(1)} = x) \cdot \Pr(u^{(2)} = y) \cdot \Pr(|C| \leq 2t - t_{\mathrm{c}} \mid u^{(1)} = x, u^{(2)} = y, \text{max. cheats}) \Big], \tag{A.0.2}$$

where $u^{(1)}$ is the total number of branch shares that suffered from stochastic erasures in the communication channels, and $u^{(2)}$ is the total number of stochastic erasures in the leaf shares. The first sum goes from 0 to $2t-t_{\mathrm{c}}$, since more than $2t-t_{\mathrm{c}}$ errors in the branches already make the condition $|C| \leq 2t-t_{\mathrm{c}}$ to fail, i.e., $\Pr(|C| \leq 2t-t_{\mathrm{c}} \mid u^{(1)} > 2t-t_{\mathrm{c}}) = 0$. Similarly, the maximum number of leaf erasures that allow for reconstruction is $3tn-2t^2$. This can be shown as follows. First, each of the $n-2t$ branch shares that are used to reconstruct the secret cannot suffer more than t leaf errors, while each of the other $2t$ branches can suffer up to n. This yields a maximum of $(n-2t)t+2tn = 3tn-2t^2$ leaf erasures: $\Pr(|C| \leq 2t-t_{\mathrm{c}} \mid u^{(2)} > 3tn-2t^2) = 0$. $\Pr(u^{(1)}=x)$ corresponds to a binomial probability distribution in which n branch shares have an independent probability q of being erased. Similarly, $\Pr(u^{(2)}=y)$ also corresponds to a binomial distribution in which n^2 leaf shares can be independently erased with probability q. Therefore,

$$\Pr(u^{(1)}=x) = q^x \binom{n}{x} (1-q)^{n-x}, \tag{A.0.3}$$

$$\Pr(u^{(2)}=y) = q^y \binom{n^2}{y} (1-q)^{n^2-y}. \tag{A.0.4}$$

Next, we need to compute $\Pr(|C| \leq 2t-t_{\mathrm{c}} \mid u^{(1)}, u^{(2)}, \text{max. cheats})$. In the remainder of this appendix, all probabilities are conditioned on 'max. cheats', but we will not write it explicitly for handiness. Using the LTP, this probability can be written as

$$\begin{aligned}\Pr(|C| \leq 2t-t_{\mathrm{c}} \mid u^{(1)}=x, u^{(2)}=y) = \sum_{\omega=\max(t_{\mathrm{d}},x)}^{\min(n,t_{\mathrm{d}}+x)} & \Pr(t^{(1\mathrm{d})}=\omega \mid u^{(1)}=x) \\ & \cdot \Pr(|C| \leq 2t-t_{\mathrm{c}} \mid t^{(1\mathrm{d})}=\omega, u^{(2)}=y).\end{aligned} \tag{A.0.5}$$

where $t^{(1\mathrm{d})}$ is the number of branches that suffered from stochastic erasures or from errors introduced by the dealer. A single branch can be affected by both types of errors. Therefore, the minimum value of $t^{(1\mathrm{d})}$ takes place when all the stochastic erasures happen in branches in which the dealer already introduced errors (or vice versa). Therefore, $t^{(1\mathrm{d})}$ cannot be less than the maximum between the number of stochastic erasures, $u^{(1)}=x$, and the number of errors introduced by the dealer, t_{d}. If all these errors happened in $t_{\mathrm{d}}+x$ different branches instead, the value of $t^{(1\mathrm{d})}$ would be maximum, with n as upper bound, since this is the total number of branches.

Let us now compute $\Pr(t^{(1\mathrm{d})}=\omega | u^{(1)}=x)$. The approach we take is to first calculate the total number of ways to combine $u^{(1)}$ erasures and t_{d} errors in the branches such that $t^{(1\mathrm{d})}=\omega$, and then divide it by the total number of possible

combinations. This yields

$$\Pr(t^{(1\mathrm{d})} = \omega | u^{(1)} = x) = \frac{\binom{n}{\max(t_\mathrm{d},x)}\binom{\max(t_\mathrm{d},x)}{t_\mathrm{d}+x-\omega}\binom{n-\max(t_\mathrm{d},x)}{\omega-\max(t_\mathrm{d},x)}}{\binom{n}{x}\binom{n}{t_\mathrm{d}}}. \tag{A.0.6}$$

Let us explain the previous expression factor by factor. First, $\binom{n}{x}\binom{n}{t_\mathrm{d}}$ is the total number of possible combinations of $u^{(1)}$ erasures and t_d errors happening in n branches. Regarding the numerator, the first factor accounts for the total number of ways of choosing $\max(t_\mathrm{d}, x)$ out of n branches. These branches are assumed to suffer from some kind of error. Moreover, among these $\max(t_\mathrm{d}, x)$ branches, we have to consider all the combinations of branches that suffered simultaneously from stochastic erasures and errors introduced by the dealer. There are a total of $t_\mathrm{d} + x - \omega$ of such branches, yielding the factor $\binom{\max(t_\mathrm{d},x)}{t_\mathrm{d}+x-\omega}$. Finally, from the other $n - \max(t_\mathrm{d}, x)$ branches, we choose the remaining $\omega - \max(t_\mathrm{d}, x)$ branches that suffered from errors.

In the previous step, we included the effect of the dealer. We must also include the errors introduced by the cheating nodes (recall that we assume that all of them introduce errors in their shares). To do that, we apply the LTP to $\Pr(|C| \leq 2t - t_\mathrm{c} \mid t^{(1\mathrm{d})} = \omega, u^{(2)})$ in the same way as we did in Equation (A.0.5):

$$\begin{aligned}\Pr(|C| \leq 2t - t_\mathrm{c} \mid t^{(1\mathrm{d})} = \omega, u^{(2)} = y) = \sum_{z=\max(t_\mathrm{c},\omega)}^{\min(n,t_\mathrm{c}+\omega)} &\Pr(t^{(1)} = z \mid t^{(1\mathrm{d})} = \omega) \\ &\cdot \Pr(|C| \leq 2t - t_\mathrm{c} \mid t^{(1)} = z, u^{(2)} = y),\end{aligned} \tag{A.0.7}$$

where

$$\Pr(t^{(1)} = z | t^{(1\mathrm{d})} = \omega) = \frac{\binom{n}{\max(t_\mathrm{c},\omega)}\binom{\max(t_\mathrm{c},\omega)}{t_\mathrm{c}+\omega-z}\binom{n-\max(t_\mathrm{c},\omega)}{z-\max(t_\mathrm{c},\omega)}}{\binom{n}{\omega}\binom{n}{t_\mathrm{c}}}. \tag{A.0.8}$$

In Eq. (A.0.7), we combine the branches affected by erasures and errors introduced by the dealer ($t^{(1\mathrm{d})}$ in total) with those affected by arbitrary errors introduced by the cheaters (t_c in total). This gives the total number of branches that suffered from any kind of error $t^{(1)}$. The reasoning that leads to Eq. (A.0.8) is the same one as for Eq. (A.0.6), but using $t^{(1)}$, $t^{(1\mathrm{d})}$, and t_c instead of $t^{(1\mathrm{d})}$, $u^{(1)}$, and t_d.

Next, we need to compute $\Pr(|C| \leq 2t - t_\mathrm{c} \mid t^{(1)} = z, u^{(2)} = y)$. We apply again the LTP as follows:

$$\begin{aligned}\Pr(|C| \leq 2t - t_\mathrm{c} \mid t^{(1)} = z, u^{(2)} = y) = \sum_{\eta=\max(0,y-nz)}^{\min(y,n(n-z))} &\Pr(u^{(2\mathrm{h})} = \eta | u^{(2)} = y, t^{(1)} = z) \\ &\cdot \Pr(|C| \leq 2t - t_\mathrm{c} \mid t^{(1)} = z, u^{(2\mathrm{h})} = \eta),\end{aligned} \tag{A.0.9}$$

A

where $u^{(2\mathrm{h})}$ is the total number of stochastic erasures that affect leaf shares from branches that were not affected by errors. This variable takes its minimum value when the n leaf shares from each of the $t^{(1)}$ branches that suffered from errors suffer from as many erasures as possible. If all erasures happened in those $nt^{(1)}$ leaf shares, then $u^{(2\mathrm{h})}=0$, and if there were some more leaf erasures (i.e., $u^{(2)}>nt^{(1)}$), then $u^{(2\mathrm{h})}=u^{(2)}-nt^{(1)}$. The maximum value of $u^{(2\mathrm{h})}$ corresponds to a situation in which the $n(n-t^{(1)})$ leaf shares from the branches that did not suffer from errors suffer as many erasures as possible. Then, this value is the minimum between $u^{(2)}$ and $n(n-t^{(1)})$.
To calculate $\Pr(u^{(2\mathrm{h})}=\eta|u^{(2)}=y,t^{(1)}=z)$, we proceed in a similar way as we did for Eq. (A.0.6). This probability corresponds to the number of configurations of $u^{(2)}=y$ erasures in the leaves such that y of them happen in the $n-t^{(1)}=n-z$ branches that were not affected by errors, divided by the total number of configurations with $u^{(2)}=y$. Hence,

$$\Pr(u^{(2\mathrm{h})}=\eta|u^{(2)}=y,t^{(1)}=z)=\frac{\binom{n(n-z)}{\eta}\binom{nz}{y-\eta}}{\binom{n^2}{y}}. \tag{A.0.10}$$

The denominator is the total number of ways of choosing y out of n^2 leaf shares. The first factor in the numerator accounts for the number of ways of choosing η out of the $n(n-z)$ leaves from branches that were not affected by errors. The second factor is the number of combinations of $y-\eta$ leaves from the other nz branches.

Next, we calculate $\Pr(|C|\leq 2t-t_{\mathrm{c}}\mid t^{(1)}=z,u^{(2\mathrm{h})}=\eta)$ as follows:

$$\Pr(|C|\leq 2t-t_{\mathrm{c}}\mid t^{(1)}=z,u^{(2\mathrm{h})}=\eta)=\sum_{\xi=0}^{2t-t_{\mathrm{c}}-z}\Pr(t^{(12)}=\xi|t^{(1)}=z,u^{(2\mathrm{h})}=\eta), \tag{A.0.11}$$

where the variable $t^{(12)}$ is the number of branch shares that cannot be reconstructed due to an excess of errors in their leaves. To ensure $|C|\leq 2t-t_{\mathrm{c}}$, $t^{(12)}$ cannot be greater than $2t-t_{\mathrm{c}}-z$, since z branches have already suffered from errors.
In order to find an analytical expression for $\Pr(t^{(12)}=\xi|t^{(1)}=z,u^{(2\mathrm{h})}=\eta)$, let us first introduce the following probability distribution:

$$\Pr(t_i=\tilde{t}_i|u_i=\tilde{u}_i)=\frac{\binom{n}{\max(t_{\mathrm{c}},\tilde{u}_i)}\binom{\max(t_{\mathrm{c}},\tilde{u}_i)}{t_{\mathrm{c}}+\tilde{u}_i-\tilde{t}_i}\binom{n-\max(t_{\mathrm{c}},\tilde{u}_i)}{\tilde{t}_i-\max(t_{\mathrm{c}},\tilde{u}_i)}}{\binom{n}{\tilde{u}_i}\binom{n}{t_{\mathrm{c}}}}, \tag{A.0.12}$$

where t_i is the total number of leaf errors in branch i, including the t_{c} leaves affected by errors introduced by the cheaters, and u_i is the number of stochastic leaf erasures in branch i. The previous probability distribution can be derived in the same way as Eq. (A.0.6), but using t_i, u_i, and t_{c} instead of $t^{(1\mathrm{d})}$, $u^{(1)}$, and t_{d}.

Using the LTP and Eq. (A.0.12), we can write

$$
\begin{aligned}
&\Pr(t^{(12)}=\xi|t^{(1)}=z,u^{(2\mathrm{h})}=\eta)= \\
&\quad=\binom{n-z}{\xi}\cdot\sum_{\tilde{t}_1=t+1}^{n}\sum_{\tilde{u}_1=\tilde{t}_1-t_\mathrm{c}}^{\tilde{t}_1}\Big[\Pr(t_1=\tilde{t}_1|u_1=\tilde{u}_1)\cdot\Pr(u_1=\tilde{u}_1) \\
&\cdot\ldots\cdot\sum_{\tilde{t}_i=t+1}^{n}\sum_{\tilde{u}_i=\tilde{t}_i-t_\mathrm{c}}^{\tilde{t}_i}\Big[\Pr(t_i=\tilde{t}_i|u_i=\tilde{u}_i)\cdot\Pr(u_i=\tilde{u}_i|\sum_{j=1}^{i-1}u_j=\sum_{j=1}^{i-1}\tilde{u}_j)\cdot\ldots \\
&\cdot\sum_{\tilde{t}_\xi=t+1}^{n}\sum_{\tilde{u}_\xi=\tilde{t}_\xi-t_\mathrm{c}}^{\tilde{t}_\xi}\Big[\Pr(t_\xi=\tilde{t}_\xi|u_\xi=\tilde{u}_\xi)\cdot\Pr(u_\xi=\tilde{u}_\xi|\sum_{j=1}^{\xi-1}u_j=\sum_{j=1}^{\xi-1}\tilde{u}_j) \\
&\cdot\sum_{\tilde{t}_{\xi+1}=X_{\xi+1}}^{Y_{\xi+1}}\sum_{\tilde{u}_{\xi+1}=\tilde{t}_{\xi+1}-t_\mathrm{c}}^{\tilde{t}_{\xi+1}}\Big\{\Pr(t_{\xi+1}=\tilde{t}_{\xi+1}|u_{\xi+1}=\tilde{u}_{\xi+1}) \\
&\qquad\cdot\Pr(u_{\xi+1}=\tilde{u}_{\xi+1}|\sum_{j=1}^{\xi+1-1}u_j=\sum_{j=1}^{\xi+1-1}\tilde{u}_j) \\
&\cdot\ldots\cdot\sum_{\tilde{t}_i=X_i}^{Y_i}\sum_{\tilde{u}_i=\tilde{t}_i-t_\mathrm{c}}^{\tilde{t}_i}\Big\{\Pr(t_i=\tilde{t}_i|u_i=\tilde{u}_i)\cdot\Pr(u_i=\tilde{u}_i|\sum_{j=1}^{i-1}u_j=\sum_{j=1}^{i-1}\tilde{u}_j)\cdot\ldots \\
&\cdot\sum_{\tilde{t}_{n-z}=X_{n-z}}^{Y_{n-z}}\sum_{\tilde{u}_{n-z}=\tilde{t}_{n-z}-t_\mathrm{c}}^{\tilde{t}_{n-z}}\Pr(t_{n-z}=\tilde{t}_{n-z}|u_{n-z}=\tilde{u}_{n-z}) \\
&\qquad\cdot\Pr(u_{n-z}=\tilde{u}_{n-z}|\sum_{j=1}^{n-z-1}u_j=\sum_{j=1}^{n-z-1}\tilde{u}_j)\Big\}\Big\}\Big]\Big]\Big],
\end{aligned}
\tag{A.0.13}
$$

where $X_i=\max(0,\eta-\sum_{j=1}^{i-1}\tilde{u}_j-t(n-z-i)))$, $Y_i=\min(t,t_\mathrm{c}+\eta-\sum_{j=1}^{i-1}\tilde{u}_j)$, and

$$
\Pr(u_i=\tilde{u}_i|\sum_{j=1}^{i-1}u_j=\sum_{j=1}^{i-1}\tilde{u}_j)=\frac{\binom{\eta-\sum_{j=1}^{i-1}\tilde{u}_j}{\tilde{u}_i}\binom{n(n-z)-\eta-\sum_{j=1}^{i-1}(n-\tilde{u}_j)}{n-\tilde{u}_i}}{\binom{n(n-z)-n(i-1)}{n}}. \tag{A.0.14}
$$

The previous expressions can be explained using the following thought experiment. Let us assume that we labeled the $n-z$ branches that did not suffer from errors in a particular way. Moreover, all the $u^{(2\mathrm{h})}$ stochastic leaf erasures are applied manually by us. We start by applying u_1 of these erasures to the leaf shares of branch 1. Next, u_2 leaf erasures are applied to branch 2, and so on. Note that $\sum_{i=1}^{n}u_i=u^{(2\mathrm{h})}$, and therefore the number of erasures applied to branch i is conditioned on the number of erasures applied to all the previous branches (e.g., if $\sum_{i=1}^{m}u_i=u^{(2\mathrm{h})}$, then $u_i=0$ for any $i>m$).

After distributing all these stochastic erasures, we want to compute the probability

A

that the first ξ branches have more than t arbitrary errors, considering stochastic erasures and errors introduced by the cheaters. The factor $\binom{n-z}{\xi}$ in Eq. (A.0.13) considers all possible ways of choosing these ξ branches out of the $n-z$. Then, we calculate the probability that each of those ξ branches have $t+1$ or more errors using the LTP. The probability for each branch i depends on the previous branches, since they must satisfy $\sum_{i=1}^{n}\tilde{u}_i=\eta$. This explains the first 2ξ sums. For each of the other $n-z-\xi$ branches, we require the total number of errors to be less than t. Note that the total number of errors cannot be larger than $t_{\mathrm{c}}+\eta-\sum_{j=1}^{i-1}\tilde{u}_j$, where the sum considers all the leaf erasures applied to previous branches, explaining the expression we gave for Y_i. Moreover, X_i must be at least $\eta-\sum_{j=1}^{i-1}\tilde{u}_j-t(n-z-i))$, otherwise there would be too many erasures left for the next branches and some of them would receive more than t erasures.
Finally, Equation (A.0.14) gives the probability that branch i has $\tilde{u}_i$ stochastic erasures given the number of erasures that happened in the previous branches (recall that $\sum_{i=1}^{n}\tilde{u}_i=\eta$). Imagine we had a bag with all the $n(n-z)$ leaves, and η of them have suffered from errors. The combinatorial number in the denominator accounts for all the possible ways to choose n leaves from a bag of $n(n-z)-n(i-1)$, where $n(i-1)$ considers the leaves that were extracted from the bag and assigned to branches preceding branch i. In the numerator, the first combination accounts for the ways of choosing $\tilde{u}_i$ erased leaves from a total of $\eta-\sum_{j=1}^{i-1}\tilde{u}_j$, where the sum represents the erased leaves assigned to previous branches. The second term is for the ways of choose $n-\tilde{u}_i$ leaves that were not affected by stochastic erasures from a total of $n(n-z)$ minus η erased leaves and minus $\sum_{j=1}^{i-1}(n-\tilde{u}_j)$ non-erased leaves that were retrieved from the bag for previous branches.

To sum up, the final solution is

$$
\begin{aligned}
\Pr(|C_{\mathrm{v}}|\leq 2t-t_{\mathrm{c}}) &\geq f_{\mathrm{v}}(n,t,q,t_{\mathrm{c}},t_{\mathrm{d}})\\
&=\sum_{x=0}^{2t-t_{\mathrm{c}}}\sum_{y=0}^{3tn-2t^2}\Bigg[\Pr(u^{(1)}=x)\cdot\Pr(u^{(2)}=y)\\
&\quad\cdot\sum_{\omega=\max(t_{\mathrm{d}},x)}^{\min(n,t_{\mathrm{d}}+x)}\Bigg[\Pr(t^{(1\mathrm{d})}=\omega|u^{(1)}=x)\\
&\quad\cdot\sum_{z=\max(t_{\mathrm{c}},\omega)}^{\min(n,t_{\mathrm{c}}+\omega)}\Bigg[\Pr(t^{(1)}=z|t^{(1\mathrm{d})}=\omega)\\
&\quad\cdot\sum_{\eta=\max(0,y-nz)}^{\min(y,n(n-z))}\Bigg[\Pr(u^{(2\mathrm{h})}=\eta|u^{(2)}=y,t^{(1)}=z)\\
&\quad\cdot\sum_{\xi=0}^{2t-t_{\mathrm{c}}-z}\Pr(t^{(12)}=\xi|u^{(2)}=y,t^{(1)}=z,u^{(2\mathrm{h})}=\eta)\Bigg]\Bigg]\Bigg]\Bigg],
\end{aligned}
\tag{A.0.15}
$$

where all the analytical expressions of the probability distributions involved have been found throughout this appendix.

The previous expression could be simplified but this is not a trivial task. Moreover, upper and lower bounds can be computed, although we did not find any tight bound with a simple analytical expression. We remark again that this is not problematic for our work, as we proved that the results obtained from Algorithm 1 are arbitrarily close to the analytical expressions found in this appendix (see Proposition 4.1.1).

B

Convergence of the sampling algorithm

In this appendix, we provide the proof of Proposition 4.1.1. Let us state it here again:

Proposition B.0.1. *Let $h(\mathbf{x}, N)$ be the output of Algorithm 1, with $\mathbf{x} := (n, t, q, t_c, t_d, u^{(1)}, u^{(2)})$, and let us define $\tilde{g}(\mathbf{x}, N) := \Pr(u^{(1)}) \cdot \Pr(u^{(2)}) \cdot h(\mathbf{x}, N)$ and $g(\mathbf{x}) := \Pr(u^{(1)}) \cdot \Pr(u^{(2)}) \cdot \Pr(|C| \leq 2t - t_c | u^{(1)}, u^{(2)}, \text{max. cheats})$, where $\Pr(u^{(1)})$ and $\Pr(u^{(2)})$ are the probability distributions given by Eqs. (4.1.4) and (4.1.5), respectively. Then,*

$$\Pr\Big[|\tilde{g}(\mathbf{x}, N) - g(\mathbf{x})| < \delta\Big] \geq 1 - N^{-1} \cdot \delta^{-2} \cdot \varepsilon, \tag{B.0.1}$$

for any $\mathbf{x}$, $N > 0$, and $\delta > 0$, with

$$\begin{aligned} \varepsilon \leq q^{2u^{(1)}+2u^{(2)}} \cdot (1-q)^{2n^2+2n-2u^{(1)}-2u^{(2)}} \cdot \binom{n}{u^{(1)}}^2 \cdot \binom{n^2}{u^{(2)}}^2 \\ \cdot \left[\binom{n}{u^{(1)}} \cdot \binom{n^2}{u^{(2)}} \cdot \binom{n}{t_d} \cdot \binom{n}{t_c} - 1 \right]. \end{aligned} \tag{B.0.2}$$

Proof. First, recall that the function $f_v(n, t, q, t_c, t_d)$ from Eq. (4.1.3) corresponds to the probability that $|C| \leq 2t - t_c$ in case that all t_c cheaters introduced errors in their shares. Note that this function can be written in terms of $g(\mathbf{x})$:

$$f_v(n, t, q, t_c, t_d) = \sum_{u^{(1)}=0}^{2t-t_c} \sum_{u^{(2)}=0}^{3tn-2t^2} g(\mathbf{x}) \tag{B.0.3}$$

Proposition B.0.1 states that $\tilde{g}(\mathbf{x}, N)$, which is computed using Algorithm 1, is close to $g(\mathbf{x})$ with high probability. Hence, this could be used to obtain an estimation of

$f_{\mathrm{v}}(n,t,q,t_{\mathrm{c}},t_{\mathrm{d}})$ by replacing all terms $g(\mathbf{x})$ by $\tilde{g}(\mathbf{x},N)$.
Our goal in this proof is to derive a lower bound for $\Pr\big[|\tilde{g}(\mathbf{x},N)-g(\mathbf{x})|<\delta\big]$. For handiness, let us define $P_1 := \Pr(u^{(1)})$, $P_2 := \Pr(u^{(2)})$, and $P_C := \Pr(|C| \leq 2t-t_{\mathrm{c}}|u^{(1)},u^{(2)},\text{max. cheats})$. P_C corresponds to the probability that less than $2t-t_{\mathrm{c}}$ branch qubits suffered from errors (either directly introduced on the branch or due to an excess of errors on its leaves), given the total number of erasures in the branches, $u^{(1)}$, and in the leaves, $u^{(2)}$. This can be written as

$$P_C = \sum_{i=1}^{M} \frac{\theta_i}{M}, \tag{B.0.4}$$

where M is the total number of configurations of errors (for fixed values of $u^{(1)}$ and $u^{(2)}$), i is a label that identifies a specific configuration, and θ_i is a binary variable that has value 1 only when configuration i satisfies $|C| \leq 2t-t_{\mathrm{c}}$. Instead of using the exact value of M, we will employ the following bound:

$$M \leq \binom{n}{u^{(1)}} \cdot \binom{n}{t_{\mathrm{d}}} \cdot \binom{n}{t_{\mathrm{c}}} \cdot \binom{n^2}{u^{(2)}}. \tag{B.0.5}$$

This bound does not consider the indistinguishability of all sources of errors (noisy channels, dealer, and cheaters), and it corresponds to the number of different ways of choosing $u^{(1)}$ branch shares out of n to be erased in the communication channels between nodes, $u^{(2)}$ leaf shares out of n^2 to be erased in the channels, t_{d} branch shares out of n in which the dealer introduces errors, and t_{c} cheaters out of n nodes. Figure B.1 shows three examples of configurations of errors for $n=7$ nodes, $u^{(1)}=1$, $u^{(2)}=5$, $t_{\mathrm{d}}=1$, and $t_{\mathrm{c}}=1$. Configurations **(a)** and **(b)** are equivalent and therefore should only be considered once when computing M. The upper bound that we give for M considers that both of these configurations are different.

To estimate P_c, Algorithm 1 generates a configuration of errors i and computes θ_i, i.e., it checks if $|C| \leq 2t-t_{\mathrm{c}}$. This process is repeated over N iterations. In order to avoid large memory requirements, the Algorithm does not store which configurations have been already checked. Therefore, it could choose the same configuration several times. Let us define the binary variable z_{ij}, with $i=1,\ldots,M$ and $j=1,\ldots,N$, which has value 1 only if configuration i was the one chosen at iteration j of the Algorithm. We can describe z_{ij} as a discrete random variable that takes value 0 with probability $\frac{M-1}{M}$ and value 1 with probability $\frac{1}{M}$.
We can write the output of Algorithm 1 as

$$h(\mathbf{x},N) = \sum_{i=1}^{M} \frac{\sum_{j=1}^{N} z_{ij}}{N} \theta_i, \tag{B.0.6}$$

where $\sum_{j=1}^{N} z_{ij}$ is the number of times that configuration i was sampled over the N iterations.

Now, we are ready to calculate the lower bound for $\Pr\big[|\tilde{g}(\mathbf{x},N)-g(\mathbf{x})|<\delta\big]$. We proceed as follows:

$$
\begin{aligned}
\Pr\big[|\tilde{g}(\mathbf{x},N)-g(\mathbf{x})|<\delta\big] &\overset{a}{=} \Pr\big[P_1\cdot P_2\cdot|h(\mathbf{x},N)-P_C|<\delta\big] \\
&= \Pr\left[|h(\mathbf{x},N)-P_C|<\frac{\delta}{P_1P_2}\right] \\
&\overset{b}{=} \Pr\left[\left|\sum_{i=1}^{M}\frac{\sum_{j=1}^{N}z_{ij}}{N}\theta_i-\sum_{i=1}^{M}\frac{\theta_i}{M}\right|<\frac{\delta}{P_1P_2}\right] \\
&= \Pr\left[\left|\sum_{i=1}^{M}\theta_i\left(\frac{\sum_{j=1}^{N}z_{ij}}{N}-\frac{1}{M}\right)\right|<\frac{\delta}{P_1P_2}\right] \\
&\overset{c}{\geq} \Pr\left[\sum_{i=1}^{M}\left|\theta_i\left(\frac{\sum_{j=1}^{N}z_{ij}}{N}-\frac{1}{M}\right)\right|<\frac{\delta}{P_1P_2}\right] \\
&\overset{d}{\geq} \Pr\left[\sum_{i=1}^{M}\left|\frac{\sum_{j=1}^{N}z_{ij}}{N}-\frac{1}{M}\right|<\frac{\delta}{P_1P_2}\right] \\
&\overset{e}{\geq} \Pr\left[\left|\frac{\sum_{j=1}^{N}z_{ij}}{N}-\frac{1}{M}\right|<\frac{\delta}{P_1P_2M}\right] \\
&= 1-\Pr\left[\left|\frac{\sum_{j=1}^{N}z_{ij}}{N}-\frac{1}{M}\right|\geq\frac{\delta}{P_1P_2M}\right] \\
&\overset{f}{\geq} 1-\frac{(M-1)P_1^2P_2^2}{N\delta^2} \\
&\overset{g}{=} 1-\frac{\varepsilon}{N\delta^2},
\end{aligned}
\tag{B.0.7}
$$

with the following steps:

a. We use the definitions of $\tilde{g}(\mathbf{x},N)$ and $g(\mathbf{x})$.

b. We use Eqs. (B.0.6) and (B.0.4).

c. For any finite sum $\sum_{i=0}^{N}|a_i|<\eta$, it is true that $\big|\sum_{i=0}^{N}a_i\big|<\eta$. Moreover, $\Pr(B)\geq\Pr(A)$ for any two probabilistic events A and B such that $A\Rightarrow B$. Then, $\Pr\Big(\big|\sum_{i=0}^{N}a_i\big|<\eta\Big)\geq\Pr\Big(\sum_{i=0}^{N}|a_i|<\eta\Big)$.

d. For any finite sum $\sum_{i=0}^{N}|a_i|<\eta$, it is true that $\sum_{i=0}^{N}|\theta_i a_i|<\eta$, with $\theta_i\in\{0,1\}$. As in the previous step, if $A\Rightarrow B$, then $\Pr(B)\geq\Pr(A)$. Hence, $\Pr\Big(\sum_{i=0}^{N}|\theta_i a_i|<\eta\Big)\geq\Pr\Big(\sum_{i=0}^{N}|a_i|<\eta\Big)$.

e. Using that $|a_i|<\frac{\eta}{M}$, $\forall i\in\{0,\ldots,M\}\Rightarrow\sum_{i=1}^{M}|a_i|<\eta$ and that, if $A\Rightarrow B$, then $\Pr(B)\geq\Pr(A)$, we obtain $\Pr\Big(\sum_{i=1}^{M}|a_i|<\eta\Big)\geq\Pr\Big(|a_i|<\frac{\eta}{M}\Big)$ for some $i\in\{0,\ldots,M\}$.

f. Chebyshev's inequality states that [50]

$$\Pr\Big(\big|X - \mathbb{E}[X]\big| \geq \varepsilon\Big) \leq \frac{\mathbb{V}[X]}{\varepsilon^2}, \tag{B.0.8}$$

for any discrete or continuous random variable X with expected value $\mathbb{E}[X]$ and variance $\mathbb{V}[X]$, and any $\varepsilon > 0$. The binary variables z_{ij} are discrete random variables with expected value $\mathbb{E}[z_{ij}] = \frac{1}{M}$ and variance $\mathbb{V}[z_{ij}] = \mathbb{E}[z_{ij}^2] - \mathbb{E}[z_{ij}]^2 = \frac{M-1}{M^2}$. These variables are independent for different values of j, since the samples are not correlated in the Algorithm. Hence, $Z = \sum_{j=1}^{N} \frac{z_{ij}}{N}$ is a random variable with expected value $\mathbb{E}[Z] = \mathbb{V}[z_{ij}] = \frac{1}{M}$ and variance $\mathbb{V}[Z] = \frac{\mathbb{V}[z_{ij}]}{N} = \frac{M-1}{NM^2}$ [50]. Then,

$$\Pr\Big(|Z - \frac{1}{M}| \geq \varepsilon\Big) \leq \frac{M-1}{NM^2\varepsilon^2}. \tag{B.0.9}$$

By setting $\varepsilon = \frac{\delta}{P_1 P_2 M}$, we obtain

$$\Pr\left[\left|\frac{\sum_{j=1}^{N} z_{ij}}{N} - \frac{1}{M}\right| \geq \frac{\delta}{P_1 P_2 M}\right] \leq \frac{(M-1)P_1^2 P_2^2}{N\delta^2}. \tag{B.0.10}$$

g. We define

$$\begin{aligned}\varepsilon &= (M-1)P_1^2 P_2^2 \\ &\leq q^{2u^{(1)}+2u^{(2)}} \cdot (1-q)^{2n^2+2n-2u^{(1)}-2u^{(2)}} \cdot \binom{n}{u^{(1)}}^2 \cdot \binom{n^2}{u^{(2)}}^2 \\ &\qquad \cdot \left[\binom{n}{u^{(1)}} \cdot \binom{n^2}{u^{(2)}} \cdot \binom{n}{t_{\mathrm{d}}} \cdot \binom{n}{t_{\mathrm{c}}} - 1\right],\end{aligned} \tag{B.0.11}$$

where we have used Eqs. (4.1.4), (4.1.5), and (B.0.5).

□

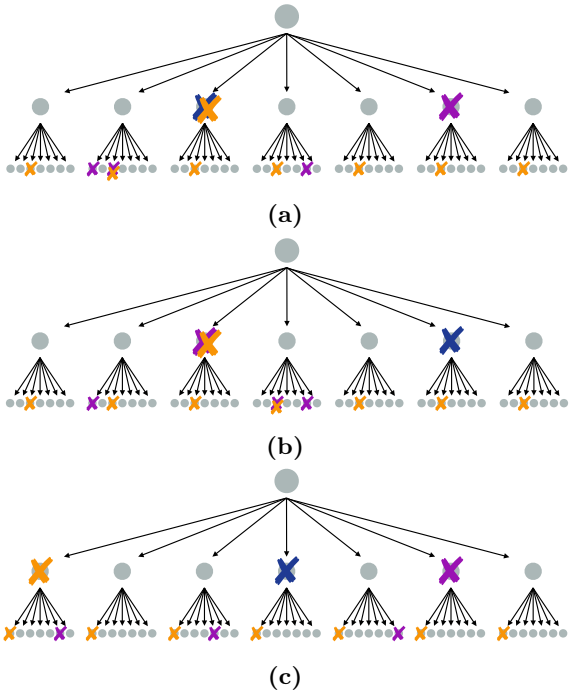


Figure B.1. Example of three configurations of errors for $n = 7$ nodes, $u^{(1)} = 1$, $u^{(2)} = 5$, $t_{\mathrm{d}} = 1$, and $t_{\mathrm{c}} = 1$. The top circle represents a qubit to be encoded, e.g. the encrypted secret. The first level of circles corresponds to the branch shares, and the second level, to the leaf shares. A purple cross represents that the qubit has suffered from a stochastic error introduced by the noisy channels. Blue and orange crosses denote that an error has been introduced in the share by the dealer and the cheaters, respectively. Note a single share can simultaneouly suffer from stochastic erasures and from arbitrary errors introduced by the dealer or the cheaters, as depicted in **(a)** and **(b)**. Configurations **(a)** and **(b)** are equivalent, while **(c)** is different. For the bound given in Eq. (B.0.5), all three configurations are different (see main text).

C

Lower bounds for the binomial distribution

Let X be a discrete random variable that follows a binomial distribution with parameters n and q. This distribution models a situation in which n identical experiments are run and each one has an independent probability of success q. The most common example for these experiments is a coin tossing. The random variable X corresponds to the total number of experiments that are successful.
The probability distribution of X is

$$P(X = k) = q^k(1-q)^{n-k}\binom{n}{k}, \tag{C.0.1}$$

and the cumulative distribution function is given by

$$S(z; n, q) := \sum_{k=0}^{z} q^k(1-q)^{n-k}\binom{n}{k}. \tag{C.0.2}$$

In [49], the authors provide lower bounds for the binomial distribution. In particular, they found that

$$P(X \geq z) \leq \exp\left(-nD\left[\frac{z}{n}||q\right]\right), \quad \text{for} \quad q < \frac{z}{n} < 1, \tag{C.0.3}$$

where $D(a||q)$ is the relative entropy between two Bernoulli distributions with parameters a and q, respectively,

$$D[a||q] = a\log\left(\frac{a}{q}\right) + (1-a)\log\left(\frac{1-a}{1-q}\right). \tag{C.0.4}$$

By rewriting Equation (C.0.3) in terms of the cumulative distribution function, we find the following lower bound:

$$\begin{aligned} S(z;n,q) &\leq 1 - \exp\left(- nD\left[\frac{z+1}{n}||q\right]\right) \\ &= 1 - \exp\left(- (z+1)\log\left(\frac{z+1}{nq}\right) + (z+1-n)\log\left(\frac{n-(z+1)}{n-nq}\right)\right) \\ &= 1 - \exp\left(- (z+1)\log\left(\frac{z+1}{nq}\right)\right)\exp\left((z+1-n)\log\left(\frac{n-(z+1)}{n-nq}\right)\right) \\ &= 1 - \left(\frac{z+1}{nq}\right)^{-(z+1)}\left(\frac{n-(z+1)}{n-nq}\right)^{z+1-n}, \end{aligned} \tag{C.0.5}$$

which is only valid for $q < \frac{z}{n} < 1$.