Noisy Interactive Quantum Communication

by

Ala Shayeghi

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Mathematics

Waterloo, Ontario, Canada, 2020

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Salman Beigi Associate Professor, School of Mathematics, Institute for Research in Fundamental Sciences (IPM)
Supervisor(s):	Ashwin Nayak Professor, Dept. of Combinatorics & Optimization, Institute for Quantum Computing, University of Waterloo
Internal Member:	Debbie Leung Professor, Dept. of Combinatorics & Optimization, Institute for Quantum Computing, University of Waterloo Jon Yard Associate Professor, Dept. of Combinatorics & Optimization, Institute for Quantum Computing, University of Waterloo
Internal-External Member:	Daniel Gottesman Adjunct faculty, Dept. of Physics & Astronomy, University of Waterloo Faculty member, Perimeter Institute

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

We consider the problem of implementing two-party interactive quantum communication over noisy channels, a necessary endeavor if we wish to fully reap quantum advantages for communication. For an arbitrary protocol with n messages, designed for *noiseless* qudit channels (where d is arbitrary), our main result is a simulation method that fails with probability less than $2^{-\Theta(n\epsilon)}$ and uses a qudit channel $n(1+\Theta(\sqrt{\epsilon}\,))$ times, of which an ϵ fraction can be corrupted adversarially. The simulation is thus capacity achieving to leading order, and we conjecture that it is optimal up to a constant factor in the $\sqrt{\epsilon}$ term. Furthermore, the simulation is in a model that does not require pre-shared resources such as randomness or entanglement between the communicating parties. Surprisingly, this outperforms the best known overhead of $1+O(\sqrt{\epsilon \log\log 1/\epsilon})$ in the corresponding *classical* model, which is also conjectured to be optimal [Haeupler, FOCS'14]. Our work also improves over the best previously known quantum result where the overhead is a non-explicit large constant [Brassard *et al.*, SICOMP'19] for small ϵ.

Acknowledgements

I would like to express my gratitude to all those without whom I would not have been able to complete this research. First and foremost, my thanks are due to my supervisor, Ashwin Nayak, for his continuous support throughout my degree. I would like to thank my other collaborators in this project, Debbie Leung, Dave Touchette, Penghui Yao, and Nengkun Yu. I am especially grateful to Dave for lots of fruitful discussions. I would also like to thank Salman Beigi, Daniel Gottesman, Debbie Leung, and Jon Yard as members of my examining committee. In particular, I would like to thank Debbie for her insight and support throughout these years.

I am deeply grateful to my parents for their unfailing love and never-ending support. Without you, I would not be the person I am today. Above all I would like to thank my wife for her love, endless support, and encouragements. Thank you for being my best friend, Shima.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

1.1.1 The main questions

Quantum communication offers the possibility of distributed computation with extraordinary *provable* savings in communication as compared with classical communication (see, e.g., [56] and the references therein). Most often, if not always, the savings are achieved by protocols that assume access to *noiseless* communication channels. In practice, though, imperfection in channels is inevitable. Is it possible to make the protocols robust to noise while maintaining the advantages offered by quantum communication? If so, what is the cost of making the protocols robust, and how much noise can be tolerated? In this thesis, we address these questions in the context of quantum communication protocols involving two parties, in the low noise regime. Following convention, we call the two parties Alice and Bob.

1.1.2 Channel coding theory as a special case

In the special case when the communication is one-way (say, from Alice to Bob), techniques for making the message noise-tolerant, via error correcting codes, have been studied for a long time. Coding allows us to simulate a noiseless communication protocol using a noisy channel, under certain assumptions about the noise process (such as having a memoryless channel). Typically, such simulation is possible when the *error rate* (the fraction of the messages corrupted) is lower than a certain threshold. A desirable goal is to also maximize the *communication rate* (also called the *information rate*), which is the length of the original message, as a fraction of the length of its encoding. In the classical setting, Shannon established the capacity (i.e., the optimal communication rate) of *arbitrarily accurate* transmission, in the limit of *asymptotically large* number of channel uses, through

the Noisy Coding Theorem [61]. Since then, researchers have discovered many explicit codes with desirable properties such as good rate, and efficient encoding and decoding procedures (see, for example, [65, 4]). Analogous results have been developed over the past two decades in the quantum setting. In particular, capacity expressions for a quantum channel transmitting classical data [42, 60] or quantum data [51, 63, 25] have been derived. Even though it is not known how we may evaluate these capacity expressions for a general quantum channel, useful error correcting codes have been developed for many channels of interest (see, for example, [23, 22, 8, 11]). Remarkably, quantum effects give rise to surprising phenomena without classical counterparts, including *superadditivity* [27, 41], and *superactivation* [64]. All of these highlight the non-trivial nature of coding for noisy quantum channels.

1.1.3 Communication complexity as a special case

In general two-party protocols, data are transmitted in each direction alternately, potentially over a number of rounds. In a computation problem, the number of rounds may grow as a function of the input size. Such protocols are at the core of several important areas including distributed computation, cryptography, interactive proof systems, and communication complexity. For example, in the case of the Disjointness function, a canonical task in the two-party communication model, an n-bit input is given to each party, who jointly compute the function with as little communication as possible. The optimal quantum protocol for this task consists of $\Theta\left(\sqrt{n}\right)$ rounds of communication, each with a constant length message [21, 43, 1], and such a high level of interaction has been shown to be necessary [45, 44, 18]. Furthermore, quantum communication leads to provable advantages over the classical setting, without any complexity-theoretic assumptions. For example, some specially crafted problems (see, for example, [55, 56]) exhibit exponential quantum advantages, and others display the power of quantum interaction by showing that just one additional round can sometimes lead to exponential savings [45].

1.1.4 The problem, and motivation for the investigation

In this thesis, we consider two-party interactive communication protocols using *noisy* communication. The goal is to effectively implement an interactive communication protocol to arbitrary accuracy despite noise in the available channels. We want to minimize the number of uses of the noisy channel, and the complexity of the coding operations. The motivation is two-fold and applies to both the classical and the quantum setting. First, this problem is a natural generalization of channel coding from the 1-way to the 2-way setting, with the "capacity" being the best ratio of the number of channel uses in the original protocol divided by that needed in the noisy implementation. Here, we consider the combined number of channel uses in both directions. Note that this scenario is different from "assisted capacities" where some auxiliary noiseless resources such as a classical side

channel for quantum transmission are given to the parties for free. Second, we would like to generalize interactive protocols to the noisy communication regime. If an interactive protocol can be implemented using noisy channels while preserving the complexity, then the corresponding communication complexity results become robust against channel noise. In particular, an important motivation is to investigate whether the quantum advantage in interactive communication protocols is robust against quantum noise. Due to the ubiquitous nature of quantum noise and fragility of quantum data, noise-resilience is of fundamental importance for the realization of quantum communication networks. The coding problem for interactive quantum communication was first studied in [16]. In Section 1.3, we elaborate on this work and the questions that arise from it.

1.2 Fundamental difficulties in coding for quantum interactive communication

For some natural problems the optimal interactive protocols require a lot of interaction. For example, distributed quantum search over n items [21, 43, 1] requires $\Theta(\sqrt{n})$ rounds of constant-sized messages [45, 44, 18]. How can we implement such highly interactive protocols over noisy channels? What are the major obstacles?

1.2.1 Standard error correcting codes are inapplicable

In both the classical and quantum settings, standard error correcting codes are inapplicable. To see this, first suppose we encode each message separately. Then the corruption of even a single encoded message can already derail the rest of the protocol. Thus, for the entire protocol to be simulated with high fidelity, we need to reduce the decoding error for each message to be inversely proportional to the length of the protocol, say n. For constant size messages, the overhead of coding then grows with the problem size n, increasing the complexity and suppressing the rate of simulation to 0 as n increases. The situation is even worse with adversarial errors: the adversary can invest the entire error budget to corrupt the shortest critical message, and it is impossible to tolerate an error rate above $\approx 1/\text{number of rounds}$, no matter what the rate of communication is. To circumvent this barrier, one must employ a coding strategy acting collectively over many messages. However, most of these are generated dynamically during the protocol and are unknown to the sender earlier. Furthermore, error correction or detection may require communication between the parties, which is also corruptible. The problem is thus reminiscent of fault-tolerant computation in that the steps needed to implement error correction are themselves subject to errors.

1.2.2 The no-cloning quantum problem

A fundamental property of quantum mechanics is that learning about an unknown quantum state from a given specimen disturbs the state [6]. In particular, an unknown quantum state cannot be cloned [26, 68]. This affects our problem in two fundamental ways. First, any logical quantum data leaked into the environment due to the noisy channel cannot be recovered by the communicating parties. Second, the parties hold a joint quantum state that evolves with the protocol, but they cannot make copies of the joint state without corrupting it.

1.3 Prior classical and quantum work

Despite the difficulties in coding for interactive communication, many interesting results have been discovered over the last 25 years, with a notable extension in the quantum setting.

1.3.1 Classical results showing positive rates

Schulman first raised the question of simulating noiseless interactive communication protocols using noisy channels in the classical setting [57, 58, 59]. He developed *tree codes* to work with messages that are determined one at a time, and generated dynamically during the course of the interaction. These codes have constant overhead, and the capacity is thus a positive constant. Furthermore, these codes protect data against adversarial noise that corrupts up to a $\frac{1}{240}$ fraction of the channel uses. This tolerable noise rate was improved by subsequent work, culminating to the results by Braverman and Rao [20]. They showed that $< \frac{1}{4}$ adversarial errors can be tolerated provided one can use large constant alphabet sizes and that this bound on noise rate is optimal.

1.3.2 Classical results with efficient encoding and decoding

The aforementioned coding schemes are not known to be computationally efficient, as they are built on tree codes; the computational complexity of encoding and decoding tree codes is unknown. Other computationally efficient encoding schemes have been developed [13, 15, 14, 32, 33, 35]. The communication rates under various scenarios have also been studied [17, 36, 28, 30]. However, the rates do not approach the capacity expected of the noise rate.

1.3.3 Classical results with optimal rates

Kol and Raz [46] first established coding with rate approaching 1 as the noise parameter goes to 0, for the binary symmetric channel. Haeupler [37] extended the above result to adversarial binary channels corrupting at most an ϵ fraction of the symbols, with communication rate $1 - O\left(\sqrt{\epsilon \log\log\left(\frac{1}{\epsilon}\right)}\right)$, which is conjectured to be optimal. For oblivious adversaries, this increases to $1 - \mathrm{O}(\sqrt{\epsilon})$. Further studies of capacity have been conducted, for example, in [40, 5]. For further details about recent results on interactive coding, see the extensive survey by Gelles [31].

1.3.4 Quantum results showing positive rates

All coding for classical interactive protocols relies on "backtracking": if an error is detected, the parties go back to an earlier stage of the protocol and resume from there. Backtracking is impossible in the quantum setting due to the no cloning principle described in the previous subsection. There is no generic way to make copies of the quantum state at earlier stages without restarting the protocol. Brassard, Nayak, Tapp, Touchette, and Unger [16] provided the first coding scheme with constant overhead by using two ideas. The first idea is to teleport each quantum message. This splits the quantum data into a protected quantum share and an unprotected classical share that is transmitted through the noisy channels using tree codes. Second, backtracking is replaced by *rewinding* of steps to *return* to a desirable earlier stage; i.e., the joint quantum state is evolved back to that of an earlier stage, which circumvents the no-cloning theorem. This is possible since local operations can be made unitary, and communication can be reversed (up to more noise). Together, a positive simulation rate (or constant overhead) can be achieved. Brassard *et al.* aim at achieving a high noise tolerance with a non-vanishing communication rate. In contrast, in this thesis, we focus on optimizing the communication rate in the low-noise regime. In the noisy analogue to the Cleve-Buhrman communication model where entanglement is free, Brassard *et al.* show that any error rate strictly less than $1/2$ can be tolerated. Moreover, in the noisy analogue to the Yao (plain) model, a noisy quantum channel with one-way quantum capacity $Q > 0$ can be used to simulate an n-message protocol given $\mathrm{O}\left(\frac{1}{Q}n\right)$ uses of the channel. However, the communication rate is sub-optimal and the coding complexity is unknown due to the use of tree codes. The rate is further reduced by a large constant in order to match the quantum and classical data in teleportation, and in coordinating the action of the parties (advancing or reversing the protocol).

1.4 Results in this thesis, overview of techniques, and our contributions

Inspired by the recent results on rate optimal coding for the classical setting [46, 37] and the rate sub-optimal coding in the quantum setting [16], a fundamental question is: can we likewise avoid the loss of communication rate for interactive *quantum* protocols? In particular, is it possible to protect quantum data without pre-shared free entanglement, and if we have to generate it at a cost, can we still achieve communication rates approaching 1 as the error rate vanishes? Further, can erroneous steps be reversed with noisy resources, and with negligible overhead as the error rate vanishes? What is the complexity of rate optimal protocols, if one exists? Are there other new obstacles?

Our main result in this thesis addresses all these questions. We focus on alternating protocols, in which Alice and Bob exchange qudits back and forth in alternation.

Theorem 1.4.1. *Consider any alternating two-party communication protocol* Π *in the plain quantum model, communicating* n *messages over a noiseless channel with an alphabet* Σ *of arbitrary size* d*. We provide a simulation protocol* Π' *which given* Π*, simulates it with probability at least* $1 - 2^{-\Theta(n\epsilon)}$*, over any fully adversarial error quantum channel with the same alphabet* Σ *and error rate* ϵ*. The simulation uses* $n\left(1 + \Theta\left(\sqrt{\epsilon}\right)\right)$ *rounds of communication, and therefore achieves a communication rate of* $1 - \Theta\left(\sqrt{\epsilon}\right)$*.*

1.4.1 Outline of the ideas and our contributions

We start by studying a simpler setting where the input protocol Π and the noisy communication channel operate on the same communication alphabet of polynomial size in n, the length of Π. This simplifies the algorithm while still capturing the main challenges we need to address. The analysis is easier to follow and shares the same outline and structure with our main result, namely simulation of noiseless interactive communication over constant-size alphabets. The framework we develop in this simpler model, sets the stage for a smooth transition to the small alphabet case. Our rate optimal protocol requires a careful combination of ideas to overcome various obstacles. Some of these ideas are well-established, some are not so well known, some require significant modifications, and some are new. A priori, it is not clear whether these previously developed tools would be useful in the context of the problem.

For the clarity of presentation, we first, in Chapter 3, introduce our main ideas in a simpler communication model, where Alice and Bob have access to free entanglement and communicate over a fully adversarial error classical channel. We introduce several key ideas while developing a basic solution to approach the optimal rate in this scenario. Inspired by [16], we use teleportation to protect the communication and the simulation is actively rewound whenever an error is detected. We develop a framework which allows the two parties to obtain a global view of the simulation by locally maintaining a classical data

structure. We adapt ideas due to Haeupler [37] to efficiently update this data structure over the noisy channel and evolve the simulation.

Then, in Chapter 4, we extend these ideas to the plain model of quantum communication with large alphabets. In the plain quantum model, Alice and Bob communicate over a fully adversarial error quantum channel and do not have access to any pre-shared resources such as entanglement or shared randomness. As a result any such resources need to be established through extra communication. This in particular makes it more challenging to achieve a high communication rate in this setting. Surprisingly, an adaptation of an old technique called the *Quantum Vernam Cipher* (QVC) [50] turns out to be the perfect method to protect quantum data in our application. QVC allows the two parties to recycle and reuse entanglement as needed throughout the simulation. Building on the ideas introduced in the teleportation-based protocol, one of our main contributions in this model is developing a mechanism to reliably recycle entanglement in a communication efficient way.

Finally, in Chapter 5, we extend our results to the case where the communication alphabet is of constant size. One barrier in applying the large alphabet simulation protocol in the small alphabet case is that synchronizing the classical data requires exchanging logarithmic position information in the length of the input protocol. In the large alphabet case this corresponds to sending a constant number of symbols but when the communication alphabet has a constant size this would lead to a vanishing simulation rate. Following Ref. [37], we use *meeting point based rewinding* to circumvent this barrier. Combining our framework for the large alphabet case with a carefully modified version of meeting point-based rewinding mechanism leads to a solution for the small alphabet case. A priori, there is little reason to expect that the simulation framework and the tools developed for each successive case extend to the next. However, the extensions are surprisingly seamless and without serious obstacles, culminating in the final result. This testifies to the power of the framework and the choice of the tools we deploy.

1.4.2 Remarks on our results

Besides resolving the question concerning rate optimal coding for quantum interactive communication in the low-noise regime, our work achieves a few additional goals. First, our main result is achieved in the plain quantum model, where the two parties have no pre-shared resources. Remarkably, our rate outperforms the conjectured optimal bound in the corresponding plain classical model! Intuitively, this is possible in the quantum setting because a secret key can be obtained from low noise quantum communication (or from entanglement) and then more efficient hashing can be performed. Second, our result is the first of its kind for establishing the capacity for a noisy quantum channel used in both directions to leading order. Third, our teleportation-based simulation protocol provides the first computationally efficient interactive coding scheme in the quantum setting. Moreover, our protocol for the plain model of quantum communication is computationally efficient modulo the initial entanglement distribution. In fact, the initial subroutine which only

involves one-way communication and uses standard quantum error correction to establish the entanglement used in the simulation is the only part of the simulation which we do not know how to perform in a computationally efficient way.

Chapter 2

Preliminaries

We assume that the reader is familiar with the quantum formalism for finite dimensional systems; for a thorough treatment, we refer the interested reader to good introductions in a quantum information theory context [53, Chapter 2], [66, Chapter 2] [67, Chapters 3, 4, 5].

2.1 Mathematical notation

Let $r, s \in \mathbb{N}$ with $r \leq s$. We use $[s]$ to denote the set $\{1, \ldots, s\}$. The notation $r : s$ is used to denote the string of consecutive integers from r to s. Let $\mathcal{A}$ be a d-dimensional Hilbert space with computational basis $\{|0\rangle, \ldots, |d-1\rangle\}$. Let X and Z be the operators such that $\mathrm{X}|k\rangle \stackrel{\text{def}}{=} |k+1\rangle$ and $\mathrm{Z}|k\rangle \stackrel{\text{def}}{=} e^{\mathrm{i}\cdot 2\pi \frac{k}{d}}|k\rangle$. The generalized Pauli operators, also known as the Heisenberg-Weyl operators, are defined as $\left\{\mathrm{X}^j\mathrm{Z}^k\right\}_{0\leq j,k\leq d-1}$. Let $\Sigma = \{0, \ldots, d-1\}$. For $N \in \mathbb{N}$, the operators in

$$\mathcal{P}_{d,N} \stackrel{\text{def}}{=} \{\mathrm{X}^{j_1}\mathrm{Z}^{k_1} \otimes \cdots \otimes \mathrm{X}^{j_N}\mathrm{Z}^{k_N}\}_{j_l,k_l\in\Sigma\,,\, l\in[N]} \tag{2.1}$$

form a basis for the space of operators on $\mathcal{A}^{\otimes N}$. For $E \in \mathcal{P}_{d,N}$, we denote by $\mathrm{wt}(E)$ the weight of E, i.e., the number of $\mathcal{A}$ subsystems on which E acts non-trivially. For $j, k \in \Sigma$, we represent the single qudit Pauli error $\mathrm{X}^j\mathrm{Z}^k$ by the string $jk \in \Sigma^2$. Similarly, a Pauli error on multiple qudits is represented by a string in $(\Sigma^2)^*$. The Fourier transform operator F is defined to be the operator such that $\mathrm{F}|j\rangle \stackrel{\text{def}}{=} \frac{1}{\sqrt{d}}\sum_{k=0}^{d-1} e^{\mathrm{i}\cdot 2\pi\frac{jk}{d}}|k\rangle$.

Proposition 2.1.1. *Let $\left\{\mathrm{X}^j\mathrm{Z}^k\right\}_{0\leq j,k\leq d-1}$ be the set of generalized Pauli operators on a d-dimensional Hilbert space. It holds that $\mathrm{X}^j\mathrm{Z}^k = e^{-\mathrm{i}\cdot 2\pi\frac{jk}{d}}\mathrm{Z}^k\mathrm{X}^j$, $\mathrm{F}\mathrm{X}^j\mathrm{F}^\dagger = \mathrm{Z}^j$ and $\mathrm{F}\mathrm{Z}^j\mathrm{F}^\dagger = \mathrm{X}^{-j}$ for every $j, k \in \{|0\rangle, \ldots, |d-1\rangle\}$.*

Definition 2.1.2. Let $\mathcal{A}, \mathcal{B}$ be d-dimensional Hilbert spaces with computational bases $\{|i\rangle_A\}_{0\leq i\leq d-1}$ and $\{|i\rangle_B\}_{0\leq i\leq d-1}$, respectively. The set of Bell states in $\mathcal{A}\otimes\mathcal{B}$ is defined as

$$\left\{\left|\phi^{j,k}\right\rangle_{AB} \stackrel{\text{def}}{=} \left(\mathrm{X}_A^j \mathrm{Z}_A^k \otimes \mathbb{1}\right)|\phi\rangle_{AB} : 0 \leq j,k \leq d-1\right\} \quad ,$$

where $|\phi\rangle_{AB} \stackrel{\text{def}}{=} \frac{1}{\sqrt{d}}\sum_{i=0}^{d-1}|i\rangle_A|i\rangle_B$. We extend the definition of $\left|\phi^{j,k}\right\rangle$ to all $j,k \in \mathbb{Z}$, as $\left|\phi^{j,k}\right\rangle \stackrel{\text{def}}{=} \left|\phi^{j \pmod d, k \pmod d}\right\rangle$. By a ($d$-dimensional) *maximally entangled state (MES)*, we mean the state $|\phi\rangle$ even though all the Bell states are maximally entangled.

We may verify that $\left|\phi^{j,k}\right\rangle = \frac{1}{\sqrt{d}}\sum_{t=0}^{d-1} e^{2\pi \mathrm{i}\frac{tk}{d}}|t+j,t\rangle$.

Proposition 2.1.3. *The Bell states* $\left\{\left|\phi^{j,k}\right\rangle\right\}_{0\leq j,k\leq d-1}$ *form an orthonormal basis in* $\mathcal{A}\otimes\mathcal{B}$.

Proposition 2.1.4. *For any unitary operator U on register A, it holds that*

$$(U\otimes\mathbb{1})|\phi\rangle_{AB} = (\mathbb{1}\otimes U^\top)|\phi\rangle_{AB} \quad ,$$

where $U^\top = \sum_{j,k}\langle j|U|k\rangle|k\rangle\langle j|$. *In particular,* $(\mathrm{F}\otimes\mathbb{1})|\phi\rangle_{AB} = (\mathbb{1}\otimes\mathrm{F})|\phi\rangle_{AB}$.

A *quantum instrument* is a generalized notion of a non-destructive quantum measurement defined by a collection of completely positive maps $\{M_a\}_{a\in\Gamma}$, where Γ is the set of measurement outcomes and $\sum_{a\in\Gamma} M_a$ defines a quantum channel, i.e., a completely positive and trace preserving map. The outcome of the measurement on a state ρ is $a\in\Gamma$ with probability $\mathrm{Tr}\left[M_a(\rho)\right]$ and the output state after the measurement conditioned on outcome a is given by $\mathrm{Tr}\left[M_a\left(\rho\right)\right]^{-1} M_a\left(\rho\right)$. The action of such a quantum instrument is expressed as a quantum channel of the form

$$M(\rho) = \sum_{a\in\Gamma} M_a(\rho)\otimes|a\rangle\langle a| \quad ,$$

where the second (classical) register contains the outcome of the measurement.

2.2 Quantum Communication Model

The definitions for the noiseless and noisy quantum communication models are copied from Ref. [16]. We refer the reader to this reference for a more detailed discussion of the relationship of the noiseless quantum communication model to well-studied quantum communication complexity models such as Yao model and the Cleve-Buhrman model.

2.2.1 Noiseless Communication Model

In the *noiseless quantum communication model* that we want to simulate, there are five quantum registers: the A register held by Alice, the B register held by Bob, the C register, which is the communication register exchanged back-and-forth between Alice and Bob and initially held by Alice, the E register held by a potential adversary Eve, and finally the R register, a reference system which purifies the state of the $ABCE$ registers throughout the protocol. The initial state $|\psi_{\text{init}}\rangle^{ABCER} \in \mathcal{H}(A \otimes B \otimes C \otimes E \otimes R)$ is chosen arbitrarily from the set of possible inputs, and is fixed at the outset of the protocol, but possibly unknown (totally or partially) to Alice and Bob. Note that to allow for composition of quantum protocols in an arbitrary environment, we consider arbitrary quantum states as input, which may be entangled with systems RE. A protocol Π is then defined by the sequence of unitary operations $U_1, U_2, \cdots, U_{n+1}$, with U_i for odd i known at least to Alice (or given to her in a black box) and acting on registers AC, and U_i for even i known at least to Bob (or given to him in a black box) and acting on registers BC. For simplicity, we assume that n is even. We can modify any protocol to satisfy this property, while increasing the total cost of communication by at most one communication of the C register. The unitary operators of protocol Π can be assumed to be public information, known to Eve. On a particular input state $|\psi_{\text{init}}\rangle$, the protocol generates the final state $|\psi_{\text{final}}\rangle^{ABCER} = U_{n+1} \cdots U_1 |\psi_{\text{init}}\rangle^{ABCER}$, for which at the end of the protocol the A and C registers are held by Alice, the B register is held by Bob, and the E register is held by Eve. The reference register R is left untouched throughout the protocol. The output of the protocol resides in systems ABC, i.e., $\Pi(|\psi_{\text{init}}\rangle) = \text{Tr}_{ER}\left(|\psi_{\text{final}}\rangle\langle\psi_{\text{final}}|^{ABCER}\right)$, and by a slight abuse of notation we also represent the induced quantum channel from $ABCE$ to ABC simply by Π. This is depicted in Figure 2.1. Note that while the protocol only acts on ABC, we wish to maintain correlations with the reference system R, while we simply disregard what happens on the E system assumed to be in Eve's hand. Since we consider local computation to be free, the sizes of A and B can be arbitrarily large, but still of finite size, say m_A and m_B qubits, respectively. Since we are interested in a high communication rate, we do not want to restrict ourselves to the case of a single-qubit communication register C, since converting a general protocol to one of this form can incur a factor of two overhead. We thus consider alternating protocols in which the register C is of fixed size, say d dimensions, and is exchanged back-and-forth. We believe that non-alternating protocols can also be simulated by adapting our techniques, but we leave this extension to future work. Note that both the Yao and the Cleve-Buhrman models of quantum communication complexity can be recast in this framework; see Ref. [16].

We later embed length n protocols into others of larger length $\tilde{n} > n$. To perform such *noiseless protocol embedding*, we define some dummy registers $\tilde{A}$, $\tilde{B}$, $\tilde{C}$ isomorphic to A, B, C, respectively. $\tilde{A}$ and $\tilde{C}$ are part of Alice's scratch register and $\tilde{B}$ is part of Bob's scratch register. Then, for any quantum registers D, $\tilde{D}$ associated with isomorphic Hilbert spaces, let $\text{SWAP}_{D \leftrightarrow \tilde{D}}$ denote the unitary operation that swaps the $D, \tilde{D}$ registers. Recall that n is assumed to be even. In a noiseless protocol embedding, for $i \in \{1, 2, \cdots n-1\}$, we leave U_i

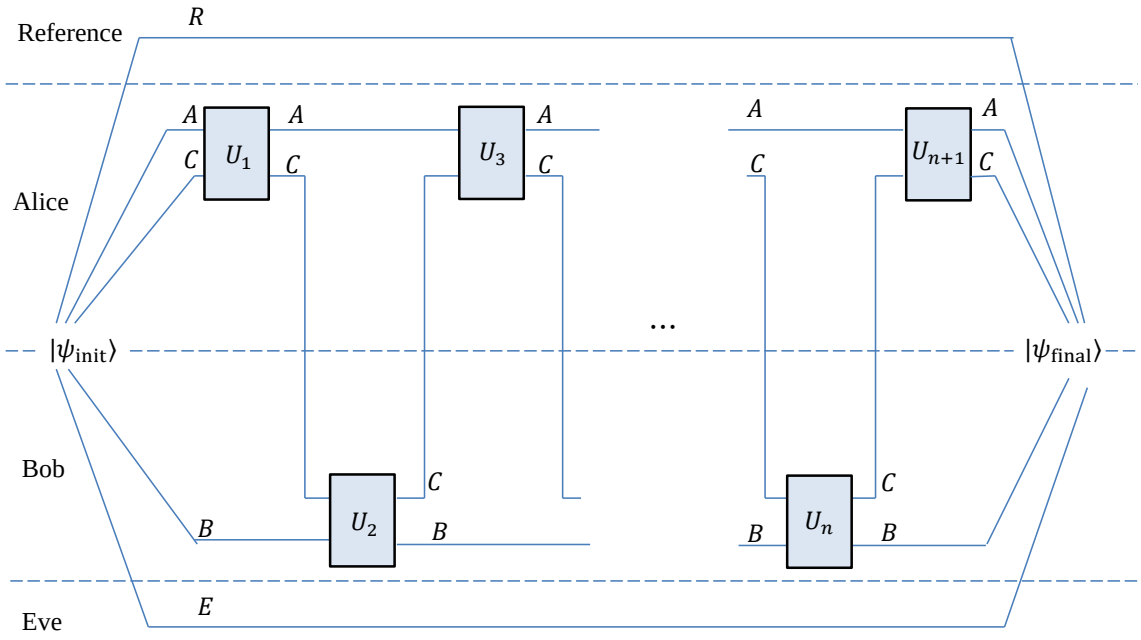

Figure 2.1: Depiction of a quantum protocol in the noiseless communication model.

untouched. We replace U_n by $(\text{SWAP}_{B\leftrightarrow\tilde{B}}U_n)$ and U_{n+1} by $(\text{SWAP}_{AC\leftrightarrow\tilde{A}\tilde{C}}U_{n+1})$. Finally, for $i \in \{n+2, n+3, \cdots \tilde{n}+1\}$, we define $U_i = \text{I}$, the identity operator. This embedding is important in the setting of interactive quantum coding for the following reasons. First, a robust protocol against transmission noise may require more than n rounds of interaction to successfully simulate an input protocol of length n. Adding these U_i for $i > n$ makes the protocol well defined for $\tilde{n}+1$ steps. Therefore, ensuring that we never run out of steps of the input protocol to simulate. Then, swapping the important registers into the safe registers $\tilde{A}$, $\tilde{B}$, $\tilde{C}$ ensures that the important registers are never affected by noise arising after the first $n+1$ steps have been applied. Hence, in our simulation, as long as we succeed in implementing the first $n+1$ steps without errors, the simulation will succeed since the $\tilde{A}$, $\tilde{B}$, $\tilde{C}$ registers will then contain the output of the simulation, with no error acting on these registers.

2.2.2 Noisy Communication Model

There are many possible models for noisy communication. For our main results, we focus on one in particular, analogous to the Yao model with no shared entanglement but noisy quantum communication, which we call the *plain quantum model*. In Chapter 3, we consider and define an alternative model.

For simplicity, we formally define in this section what we sometimes refer to as *alternating* communication models, in which Alice and Bob take turns in transmitting the communication register to each other, and this is the model in which most of our protocols are defined. Our definitions easily adapt to somewhat more general models which we call *oblivious* communication models, following Ref. [20]. In these models, Alice and Bob do not necessarily transmit their messages in alternation, but nevertheless in a fixed order and of fixed sizes known to all (Alice, Bob and Eve) depending only on the round, and not on the particular input or the actions of Eve. Communication models with a dependence on inputs or actions of Eve are called *adaptive* communication models.

Plain Quantum Model In the *plain quantum model*, input registers $ABCE$ are shared between Alice (AC), Bob (B) and Eve (E) and the reference register R contains the purification of the input. These registers are initially in the $|\psi_{\text{init}}\rangle$ state which is the initial state of the input protocol to be simulated. The output registers $\tilde{A}\tilde{B}\tilde{C}$ are shared between Alice ($\tilde{A}\tilde{C}$) and Bob ($\tilde{B}$). These are the registers introduced in the noiseless protocol embedding described above and at the end of the simulation contain the output state. The reference register R is left untouched throughout. In additions to these registers, Alice has workspace A', Bob has workspace B', the adversary Eve has workspace E', and there is some quantum communication register C' of some fixed size d' dimensions (we will consider only $d' = d$ in this work), exchanged back and forth between them n' times, passing through Eve's hand each time. Alice and Bob can perform arbitrary local processing between each transmission, whereas Eve's processing when the C' register passes through her hand is limited by the noise model as described below. Alice and Bob also possess registers C_{A} and C_{B}, respectively, acting as virtual communication register C from the original protocol Π of length n to be simulated. The communication rate of the simulation is given by the ratio $\frac{n \log d}{n' \log d'}$.

We are interested in two models of errors, adversarial and random noise. In the *adversarial* noise model, we are mainly interested in an adversary Eve with a bound $\epsilon n'$ on the number of errors that she introduces on the quantum communication register C' that passes through her hand. The fraction ϵ of corrupted transmissions is called the error rate. More formally, an adversary in the quantum model with error rate bounded by $\epsilon \in [0, 1]$ is specified by a sequence of instruments $\mathcal{N}_1^{E'C'_1}, \ldots, \mathcal{N}_{n'}^{E'C'_{n'}}$ acting on register E' of arbitrary dimension d'' and the communication register C' of dimension d' in protocols of length n'. For any

density operator ρ on $\mathcal{H}(E' \otimes C'^{\otimes n'})$, the action of such an adversary is

$$\mathcal{N}_1^{E'C'_1} \cdots \mathcal{N}_{n'}^{E'C'_{n'}}(\rho) = \sum_i G_i \rho G_i^\dagger \ , \tag{2.2}$$

for i ranging over some finite set, subject to $\sum_i G_i^\dagger G_i = \mathbb{1}^{E'C'^{\otimes n'}}$, where each G_i is of the form

$$G_i = \sum_{F \in \mathcal{P}_{d''},1} \sum_{\substack{H \in \mathcal{P}_{d',n'} \\ \mathrm{wt(H)} \leq \epsilon \mathrm{n}'}} \alpha_{F,H}^i F^{E'} \otimes H^{C'^{\otimes n'}} \ . \tag{2.3}$$

In the random noise model, we consider n' independent and identically distributed uses of a noisy quantum channel acting on register C', half the time in each direction. Eve's workspace register E' (including her input register E) can be taken to be trivial in this noise model. We only analyze the protocols in this thesis against adversarial noise, however, using concentration of measure arguments, it is straightforward to show that the same protocols can be used to obtain similar results in the random noise model as well.

For both noise models, we say that the simulation succeeds with error δ if for any input, the output in register $\tilde{A}\tilde{B}\tilde{C}$ at the end of the simulation is the same as the output obtained by running the protocol Π on the same input, while also maintaining correlations with system R, up to error δ in trace distance.

Note that adversaries in the quantum model can inject fully quantum errors since the messages are quantum, in contrast to adversaries corrupting classical messages which are restricted to be modifications of classical symbols. On the other hand, for classical messages the adversary can read all the messages without the risk of corrupting them, whereas in the quantum model, any attempt to "read" messages will result in an error in general on some quantum message.

2.3 Protocols over qudits

In this section, we revisit two quantum communication protocols, both of which are essential to our simulation algorithms and analyze the effect of noise on these protocols.

2.3.1 Quantum teleportation over noisy qudit channels

The protocol given here is an extension of quantum teleportation to qudits. Readers may refer to Chapter 6 in Ref. [67] for more details.

Definition 2.3.1. (Quantum teleportation protocol)

Alice possesses an arbitrary d-dimensional qudit in state $|\psi\rangle_A$, which she wishes to communicate to Bob. They share an MES in the state $|\phi\rangle_{A_1B_1}$.

1. Alice performs a measurement on registers AA_1 with respect to the Bell basis $\left\{\left|\phi^{j,k}\right\rangle\right\}_{j,k}$.

2. She transmits the measurement outcome (j,k) to Bob.

3. Bob applies the unitary transformation $\mathrm{Z}_{B_1}^k \mathrm{X}_{B_1}^j$ on his state to recover $|\psi\rangle$.

In the rest of the thesis the measurements implemented in Definition 2.3.1 are referred to as the *teleportation measurements* and the receiver's unitary transformation to recover the target state is referred to as *teleportation decoding operation*.

If Bob receives (j',k') due to a corruption on Alice's message, the state he gets after decryption will be the following:

$$\mathrm{Z}_B^{k'} \mathrm{X}_B^{j'} \mathrm{X}_B^{d-j} \mathrm{Z}_B^{d-k} |\psi\rangle = e^{\mathrm{i}\cdot\frac{2\pi}{d}(j'-j)k'} \mathrm{X}^{j'-j} \mathrm{Z}^{k'-k} |\psi\rangle \quad . \tag{2.4}$$

2.3.2 Quantum Vernam cipher over noisy qudit channels

In this section, we revisit *quantum Vernam cipher* (QVC) introduced by Leung [50], which is a quantum analog of Vernam cipher (one-time-pad). For a unitary operation U, the controlled gate c-U is defined as

$$(\text{c-}U)_{AB} |j\rangle_A |k\rangle_B \stackrel{\text{def}}{=} |j\rangle U^j |k\rangle \quad .$$

The extension of quantum Vernam cipher to qudit systems goes as follows.

Definition 2.3.2. (Quantum Vernam cipher)

Alice possesses an arbitrary d-dimensional qudit in state $|\psi\rangle_A$, which she wishes to communicate to Bob. They share an MES pair in the state $|\phi\rangle_{A_1B_1} |\phi\rangle_{A_2B_2}$, with Alice and Bob holding registers A_1A_2 and B_1B_2, respectively.

1. Alice applies the unitary transformation $(\text{c-Z})_{A_2A} (\text{c-X})_{A_1A}$.

2. She transmits the register A to Bob.

3. Bob applies the unitary transformation $(\text{c-X}^{-1})_{B_1B} (\text{c-Z}^{-1})_{B_2B}$.

Quantum Vernam cipher uses entanglement as the key to encrypt quantum information sent through an insecure quantum channel. In sharp contrast with the classical Vernam cipher, the quantum key can be recycled securely. Note that if no error occurs on Alice's message, then Bob recovers the state $|\psi\rangle$ perfectly, and at the end of the protocol the MES pair remain intact. The scheme detects and corrects for arbitrary transmission errors, and

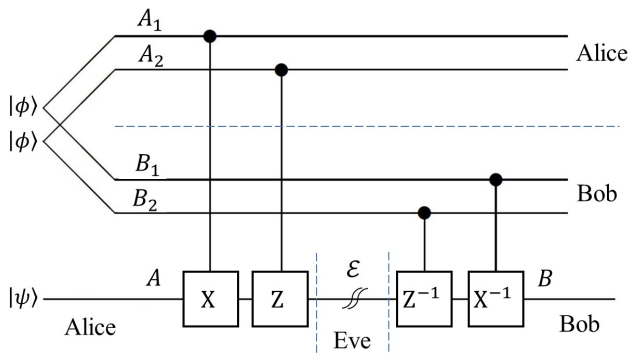

Figure 2.2: Sending one qudit through quantum channel $\mathcal{E}$ using quantum Vernam cipher.

it only requires local operations and classical communication between the sender and the receiver.

In particular, if Alice's message is corrupted by the Pauli error $\mathrm{X}^j\mathrm{Z}^k$, the joint state after Bob's decryption is

$$\begin{aligned}
& \left(\text{c-X}^{-1}\right)_{B_1A}\left(\text{c-Z}^{-1}\right)_{B_2A}\mathrm{X}^j\mathrm{Z}^k\,(\text{c-Z})_{A_2A}\,(\text{c-X})_{A_1A}\,|\phi\rangle_{A_1B_1}\,|\phi\rangle_{A_2B_2}\,|\psi\rangle_A \\
= \;& \frac{1}{d}\sum_{t,t'=0}^{d-1}\left(\text{c-X}^{-1}\right)_{B_1A}\left(\text{c-Z}^{-1}\right)_{B_2A}\mathrm{X}^j\mathrm{Z}^k\,(\text{c-Z})_{A_2A}\,(\text{c-X})_{A_1A}\,|t\rangle_{A_1}\,|t\rangle_{B_1}\,|t'\rangle_{A_2}\,|t'\rangle_{B_2}\,|\psi\rangle_A \\
= \;& \frac{1}{d}\sum_{t,t'=0}^{d-1}|t\rangle_{A_1}\,|t\rangle_{B_1}\,|t'\rangle_{A_2}\,|t'\rangle_{B_2}\,\mathrm{X}^{-t}\mathrm{Z}^{-t'}\mathrm{X}^j\mathrm{Z}^k\mathrm{Z}^{t'}\mathrm{X}^t\,|\psi\rangle_A \\
= \;& \frac{1}{d}\sum_{t,t'=0}^{d-1}e^{\mathrm{i}\cdot\frac{2\pi}{d}(kt-jt')}\,|t\rangle_{A_1}\,|t\rangle_{B_1}\,|t'\rangle_{A_2}\,|t'\rangle_{B_2}\,\mathrm{X}^j\mathrm{Z}^k\,|\psi\rangle \\
= \;& \left|\phi^{0,k}\right\rangle_{A_1B_1}\left|\phi^{0,-j}\right\rangle_{A_2B_2}\otimes\mathrm{X}^j\mathrm{Z}^k\,|\psi\rangle \quad . \qquad (2.5)
\end{aligned}$$

Note that by Eq. (2.5), there is a one-to-one correspondence between the Pauli errors and the state of the maximally entangled pair. An X^j error on the cipher-text is reflected in the state of the second MES as a Z^{-j} error and a Z^k error on the cipher-text is reflected in the state of the first MES as a Z^k error. Note that for every integer s we have

$$\left(\mathrm{F}\otimes\mathrm{F}^\dagger\right)\left|\phi^{0,s}\right\rangle \quad = \quad \left(\mathrm{F}\mathrm{Z}^s\otimes\mathrm{F}^\dagger\right)|\phi\rangle \quad = \quad \left(\mathrm{F}\mathrm{Z}^s\mathrm{F}^\dagger\otimes\mathbb{1}\right)|\phi\rangle \quad = \quad \left(\mathrm{X}^{-s}\otimes\mathbb{1}\right)|\phi\rangle \quad .$$

Therefore, in order to extract the error syndrome, it suffices for Alice and Bob to apply F and $\mathrm{F}^\dagger$, respectively, on their marginals of the MESs and measure them in the computational basis. By comparing their measurement outcomes they can determine the Pauli error.

When quantum Vernam cipher is used for communication of multiple messages, it is possible to detect errors without disturbing the state of the MES pairs at the cost of an additional

fresh MES. This error detection procedure allows for recycling of MESs which is crucial in order to achieve a high communication rate, as explained in Section 4.3.2. Here we describe a simplified version of the detection procedure. First we need the following two lemma.

Proposition 2.3.3. *It holds that*

$$(c\text{-X})_{A_1A_2} \cdot (c\text{-X})_{B_1B_2} \left|\phi^{j_1,k_1}\right\rangle_{A_1B_1} \left|\phi^{j_2,k_2}\right\rangle_{A_2B_2} = \left|\phi^{j_1,k_1-k_2}\right\rangle_{A_1B_1} \left|\phi^{j_1+j_2,k_2}\right\rangle_{A_2B_2}.$$

In particular,

$$(c\text{-X})_{A_1A_2} \cdot (c\text{-X})_{B_1B_2} \left|\phi^{0,k_1}\right\rangle_{A_1B_1} \left|\phi^{0,k_2}\right\rangle_{A_2B_2} = \left|\phi^{0,k_1-k_2}\right\rangle_{A_1B_1} \left|\phi^{0,k_2}\right\rangle_{A_2B_2}.$$

Proof.

$$\begin{aligned}
& (\text{c-X})_{A_1A_2} \cdot (\text{c-X})_{B_1B_2} \left|\phi^{j_1,k_1}\right\rangle_{A_1B_1} \left|\phi^{j_2,k_2}\right\rangle_{A_2B_2} \\
= & \frac{1}{d} \sum_{t_1,t_2=0}^{d-1} (\text{c-X})_{A_1A_2} \cdot (\text{c-X})_{B_1B_2} e^{\mathrm{i}\cdot\frac{2\pi}{d}(t_1k_1+t_2k_2)} |t_1+j_1\rangle_{A_1} |t_1\rangle_{B_1} |t_2+j_2\rangle_{A_2} |t_2\rangle_{B_2} \\
= & \frac{1}{d} \sum_{t_1,t_2=0}^{d-1} e^{\mathrm{i}\cdot\frac{2\pi}{d}(t_1k_1+t_2k_2)} |t_1+j_1\rangle_{A_1} |t_1\rangle_{B_1} |t_2+j_2+t_1+j_1\rangle_{A_2} |t_2+t_1\rangle_{B_2} \\
= & \left|\phi^{j_1,k_1-k_2}\right\rangle_{A_1B_1} \left|\phi^{j_1+j_2,k_2}\right\rangle_{A_2B_2}.
\end{aligned}$$

□

Suppose that Alice and Bob start with m copies of the MES $|\phi\rangle$ and use them in pairs to communicate messages using QVC over a noisy channel. By Eq. (2.5) all the MESs remain in $\mathsf{span}\left\{\left|\phi^{0,k}\right\rangle : 0 \le k \le d-1\right\}$. This invariance is crucial to the correctness of our simulation. Let $\left|\phi^{0,k_i}\right\rangle_{A_iB_i}$ be the state of the i-th MES after the communication is done. In order to detect errors, Alice and Bob use an additional MES $|\phi\rangle_{A_0B_0}$. For $i = 1, ..., m$, Alice and Bob apply $(\text{c-X})_{A_0A_i}$ and $(\text{c-X})_{B_0B_i}$, respectively. By Proposition 2.3.3, the joint state of the register A_0B_0 will be $\left|\phi^{0,-\sum_{i=1}^m k_i}\right\rangle_{A_0B_0}$. Now, all Alice and Bob need to do is to apply F and $F^\dagger$ on registers A_0 and B_0, respectively, and measure their marginal states in the computational basis. By comparing their measurement outcomes they can decide whether any error has occurred. In this procedure the MESs used as the keys in QVC are not measured. Note that if the corruptions are chosen so that $\sum_{i=1}^m k_i = 0 \bmod d$ then this procedure fails to detect the errors. We will analyze a modified version of this error detection procedure in detail in Section 4.3.2 which allows error detection with high probability independent of the error syndrome.

2.4 Small-bias and k-wise independence

Definition 2.4.1. Let $X = X_1 \dots X_n$ be a random variable distributed over $\{0,1\}^n$ and $J \subseteq [n]$ be a non-empty set. The *bias* of J with respect to distribution X, denoted $\text{bias}_J(X)$, is defined as

$$\text{bias}_J(X) \stackrel{\text{def}}{=} \left| \Pr\left(\sum_{i \in J} X_i = 1\right) - \Pr\left(\sum_{i \in J} X_i = 0\right) \right| \ ,$$

where the summation is mod 2. For $J = \emptyset$, bias is defined to be zero, i.e., $\text{bias}_\emptyset(X) = 0$.

Definition 2.4.2 (small-bias probability space). Let $\delta \in [0,1]$. A distribution X over $\{0,1\}^n$ is called a *δ-biased probability space* if $\text{bias}_J(X) \leq \delta$, for all non-empty subsets $J \subseteq [n]$.

Definition 2.4.3. Let p and q be probability distributions over the same (countable) set Ω. The L_1-distance between p and q is defined as

$$\|p - q\|_1 \stackrel{\text{def}}{=} \sum_{x \in \Omega} |p(x) - q(x)| \ , \tag{2.6}$$

and the L_2-distance between p and q is defined as

$$\|p - q\|_2 \stackrel{\text{def}}{=} \sqrt{\sum_{x \in \Omega} (p(x) - q(x))^2} \ . \tag{2.7}$$

We say p and q are δ-close in L_1-distance if $\|p - q\|_1 \leq \delta$. Similarly, p and q are said to be δ-close in L_2-distance if $\|p - q\|_2 \leq \delta$.

We will make use of the following proposition providing an alternative characterization of the L_1-distance between two probability distributions.

Proposition 2.4.4. *Let p and q be probability distributions over some (countable) set Ω, then*

$$\|p - q\|_1 = 2 \sup_{A \subseteq \Omega} |p(A) - q(A)| \ .$$

Intuitively, a small-bias random variable is statistically close to being uniformly distributed. The following lemma quantifies this statement.

Proposition 2.4.5 ([3])**.** *Let X be a δ-biased distribution over $\{0,1\}^n$ and let U denote the uniform distribution over $\{0,1\}^n$. Then X is δ-close in L_2-distance and $\left(2^{n/2}\delta\right)$-close in L_1-distance to U.*

Definition 2.4.6 (k-wise independence). A distribution X over $\{0,1\}^n$ is called *k-wise independent* if for any subset $J \subseteq [n]$ such that $|J| = k$, X_J, the restriction of X to the subset J is uniformly distributed.

The following is a direct corollary of Proposition 2.4.5.

Proposition 2.4.7 ([3]). *Any δ-biased random variable $X \in \{0,1\}^n$ is ϵ-close in L_1-distance to being k-wise independent for $\epsilon = 2^{k/2}\delta$.*

We use the following lemma in our algorithms to stretch uniformly random strings to much longer small-bias pseudo-random strings.

Lemma 2.4.8 ([52]). *For every $\delta \in (0,1)$, there exists a deterministic algorithm which given $\mathrm{O}\left(\log n + \log \frac{1}{\delta}\right)$ uniformly random bits outputs a δ-biased pseudo-random string of n bits.*

Chapter 3

Teleportation-based coding scheme via large alphabet classical channels

In this chapter, we focus on the teleportation-based quantum communication model with a polynomial-size alphabet. In this model, Alice and Bob share an unlimited number of copies of the maximally entangled state (MES) $|\phi\rangle$ before the protocol begins. The parties effectively send each other qudit messages by using an MES and then sending two classical symbols from the communication alphabet per qudit. The complexity of the protocol is the number of classical symbols exchanged, while the MESs are available for free. We call this model noiseless if the classical channel is noiseless. A large communication alphabet allows the parties to send logarithmic amount of information by communicating only a constant number of symbols from the alphabet. This simplifies our algorithm while allowing us to address the main challenges of interactive communication in this setting. In Chapter 5, we extend the framework we develop in this chapter to constant-size communication alphabets by carefully adapting the existing machinery in the classical setting.

3.1 Overview

At a high level, we adapt ideas due to Brassard *et al.* [16] and Haeupler [37] to design the simulation protocol. Namely, the simulation protocol Π' tries to construct the joint quantum state of the parties in the original protocol Π by evolving Π. When a transmission error is detected, we actively reverse earlier local operations in Π', as in Ref. [16]. In addition, we make communication robust to noise by adapting ideas from Ref. [37]. In the simulation protocol Π', both parties conduct the original conversation from Π as if there were no noise, except for the following:

- At regular intervals they exchange concise "summaries" of their (potentially different) views of the conversation up to that point.

- If the summaries are consistent, they continue the conversation.
- If the summaries are inconsistent, a transmission error is presumed to have occurred at some point in the simulation. The parties then rewind the simulation to an earlier stage of the conversation and resume from there.

This template can be interpreted as an error-correcting code over many messages, with trivial, and most importantly, *message-wise* encoding. The two-way summaries correspond to error syndromes for a large number of messages, thereby preserving the rate.

Next, we provide a more detailed description of the classical protocol of Ref. [37] for noisy interactive communication which can be helpful in understanding our algorithm. The input protocol is divided into smaller blocks of $r = \Theta_\epsilon(1)$ messages. At any point in the simulation, each party has a partial transcript corresponding to their view of the conversation so far. Let T_A and T_B denote Alice's and Bob's partial transcripts, respectively. Starting from empty strings T_A and T_B, at the beginning of each iteration, the two parties compare their partial transcripts through hashing. As described above, if their hash values do match, they continue their original conversation for another block of length r. Otherwise, they need to backtrack to a common prefix of T_A and T_B. One way to achieve this is to discard the last block of their transcripts when they suspect that an error has occurred. However, T_A and T_B are not necessarily of the same length and this simple strategy may lead to backtracking all the way to empty strings T_A and T_B. In the large alphabet setting, Alice and Bob can send logarithmic transcript length information by communicating only a constant number of symbols from the alphabet. In the scenario above, the party with a longer transcript keeps backtracking until both transcripts are equally long and then they both backtrack until they reach a common prefix. In Chapter 5 we will explain how the rewinding steps are coordinated when the communication is over a constant-size alphabet.

In the classical setting, the simulation works by limiting the maximum amount of communication that is rendered useless by a single error to $\mathrm{O}_\epsilon(1)$, where ϵ is the error-rate. As the error-rate vanishes, the communication rate goes to 1. In addition, the consistency tests are efficient, consisting of evaluation of simple hash functions.

Before we describe our simulation protocol in more detail (in Section 3.3), we discuss some of its important aspects.

3.1.1 Insufficiency of simply combining [16] and [37]

Suppose we wish to simulate an interactive protocol Π that uses noiseless classical channels in the teleportation-based model. When implementing Π with noisy classical channels, it is *not sufficient* to apply the Haeupler template to the classical messages used in teleportation, and rewind as in Ref. [16] when an error is suspected. The reason is that, in Ref. [16], each message is expanded to convey different types of actions in one step (for example, whether the parties are evolving the original protocol forward or reversing it). This also

helps maintain the matching between the classical messages and the corresponding MES, and the matching between the registers containing the two halves of the MESs. However, this method incurs a large constant factor overhead in the communication which we wish to avoid.

3.1.2 New difficulties in rate-optimal simulations

Due to errors in communication, the parties need to actively rewind the simulation to correct errors on their joint quantum state. This itself can lead to a situation where the parties may not agree on how they proceed with the simulation (to rewind simulation or to proceed forward). In order to move on, both parties first need to know what the other party has done so far in the simulation. This allows them to obtain a global view of the current joint state and decide on their next action. In Ref. [16], this reconciliation step was facilitated by the extra information sent by each party and the use of tree codes. This mechanism is not available to us.

3.1.3 Framework

Our first new idea is to introduce sufficient yet concise data structures so that the parties can detect inconsistencies in (1) the stage in which they are in the protocol, (2) what type of action they should be taking, (3) histories leading to the above, (4) histories of measurement outcomes generated by one party versus the potentially different (corrupted) received instruction for teleportation decoding, (5) which system contains the next MES to be used, (6) a classical description of the joint quantum state, which is only partially known to each party. Each of Alice and Bob maintain her/his data (we collectively call these $D_{\mathsf{A}}, D_{\mathsf{B}}$ respectively, here), and also an estimate of the other party's data ($\widetilde{D_{\mathsf{B}}}, \widetilde{D_{\mathsf{A}}}$ respectively). Without channel noise, these data are equal to their estimates.

3.1.4 A major new obstacle: out-of-sync teleportation

At every step in the simulation protocol Π', Alice and Bob may engage in one of three actions based on their current view of the simulation: a forward step in Π, step in reverse, or the exchange of classical information. However, due to adversary's corruptions Alice and Bob may have inconsistent views of the simulation. This leads to a new difficulty: errors in the summaries can trigger Alice and Bob to engage in different actions. In particular, it is possible that one party tries to teleport while the other expects classical communication, with only one party consuming his/her half of an MES. They then become out-of-sync over which MESs to use. This kind of problem, to the best of our knowledge, has not been encountered before, and it is not clear if quantum data can be protected from such error. (For example, Alice may try to teleport a message into an MES that Bob already "used" earlier.) One of our main technical contributions is to show that the quantum data

can always be located and recovered when Alice and Bob resolve the inconsistencies in their data $(D_{\mathsf{A}}, \widetilde{D_{\mathsf{B}}})$ and $(\widetilde{D_{\mathsf{A}}}, D_{\mathsf{B}})$ in the low noise regime. This is particularly surprising since quantum data can potentially leak irreversibly to the environment (or the adversary): Alice and Bob potentially operate in an open system due to channel noise, and out-of-sync teleportation a priori does not protect the messages so sent.

3.1.5 Tight rope between robustness and rate

The simulation maintains sufficient data structures to store information about each party's view so that Alice and Bob can overcome all the obstacles described above. The simulation makes progress so long as Alice's and Bob's views are consistent. The robustness of the simulation requires that the consistency checks be frequent and sensitive enough so that errors are caught quickly. On the other hand, to optimize interactive channel capacity, the checks have to remain communication efficient and not too frequent neither. We also put in some redundancy in the data structures to simplify the analysis. This calls for delicate analysis in which we balance the two. In more detail, let c denote the number of communicated symbols for consistency checks in each iteration. Roughly speaking, for every r symbols of Π, $r + c$ symbols are communicated in the simulation protocol. The other source of losing the simulation rate is the communication wasted by transmission errors. A single error is sufficient to waste $r + c$ communicated symbols in at least one iteration. This informal argument suggests that any such successful coding scheme would require communication of at least

$$\frac{n}{r}(r + c) + n\epsilon(r + c)$$

symbols. This expression is minimized when $r \approx \sqrt{c/\epsilon}$. This also implies that the communication rate of $1 - \Theta(\sqrt{\epsilon})$ is the best we can hope to achieve using protocols based on the rewind-if-error paradigm above.

3.2 Result

The following is our main result in the teleportation-based model for simulation of any n-round noiseless communication protocol over an adversarial channel that corrupts any ϵ fraction of the transmitted symbols.

Theorem 3.2.1. *Consider any n-round alternating communication protocol Π in the teleportation-based model, communicating messages over a noiseless channel with an alphabet Σ of bit-size $\Theta(\log n)$. Algorithm 2 is a computationally efficient coding scheme which given Π, simulates it with probability at least $1 - 2^{-\Theta(n\epsilon)}$, over any fully adversarial error channel with alphabet Σ and error rate ϵ. The simulation uses $n(1 + \Theta(\sqrt{\epsilon}))$ rounds of communication, and therefore achieves a communication rate of $1 - \Theta(\sqrt{\epsilon})$. Furthermore, the computational complexity of the coding operations is $\mathrm{O}(n^2)$.*

3.3 Description of Protocol

We follow the notation associated with quantum communication protocols introduced in Section 2.2 in the description below.

Recall that in the teleportation-based quantum communication model, Alice and Bob implement a protocol Π_0 with prior shared entanglement and quantum communication by substituting teleportation for quantum communication. For simplicity, we assume that Π_0 is alternating, and begins with Alice. In the implementation Π of Π_0, the message register C from Π_0 has two counterparts, C_{A} and C_{B}, held by Alice and Bob, respectively. The unitary operations on AC in Π_0 are applied by Alice on AC_{A} in Π. When Alice sends the qudit in C to Bob in Π_0, she applies the teleportation measurement to C_{A} and her share of the next available MES, and sends the measurement outcome to Bob in Π. Then Bob applies a decoding operation on his share of the MES, based on the message received, and swaps the MES register with C_{B}. Bob and Alice's actions in Π when Bob wishes to do a local operation and send a qudit to Alice in Π_0 are analogously defined. For ease of comparison with the joint state in Π_0, we describe the joint state of the registers in Π (or its simulation over a noisy channel) in terms of registers ABC. There, C stands for C_{A} if Alice is to send the next message or all messages have been sent, and for C_{B} if Bob is to send the next message.

Starting with such a protocol Π in the teleportation-based model, we design a simulation protocol Π' which uses a noisy classical channel. The simulation works with *blocks* of even number of messages. By a *block* of size r (for even r) of Π, we mean a sequence of r local operations and messages alternately sent in Π by Alice and Bob, starting with Alice.

Roughly speaking, Alice and Bob run the steps of the original protocol Π as is, in blocks of size $r \stackrel{\text{def}}{=} \Theta(\frac{1}{\sqrt{\epsilon}})$, with r even. They exchange summary information between these blocks, in order to check whether they agree on the operations that have been applying to the quantum registers ABC in the simulation. The MESs used for teleportations are correspondingly divided into blocks of r MESs, implicitly numbered from 1 to r: the odd numbered ones are used to simulate quantum communication from Alice to Bob, and the even numbered ones from Bob to Alice. If either party detects an error in transmission, they may run a block of Π in reverse, or simply communicate classically to help recover from the error. The classical communication is also conducted in sequences equal in length to the ones involving a block of Π. A block of Π' refers to any of these types of sequences.

3.3.1 Metadata

Alice uses an iteration in Π' for one out of four different types of operations: evolving the simulation by running a block of Π in the forward direction (denoted a "+1" block); reversing the simulation by applying inverses of unitary operations of Π (denoted a "-1" block); synchronizing with Bob on the number of MESs used so far by applying identity operators between rounds of teleportation or reversing such an iteration (denoted a "0"

block, with 0 standing for the application of unitary operations U_i^0 which are $\mathbb{1}^{AC}$); catching up on the description of the protocol so far by exchanging classical data with Bob (denoted a "C" block, with C standing for "classical"). Alice records the sequence of types of iterations as her "metadata" in the string $\mathit{FullMA} \in \{\pm 1, 0, \mathsf{C}\}^*$. FullMA gets extended by one symbol for each new iteration of the simulation protocol Π'. The number of blocks of r MESs Alice has used is denoted q_{MA} which corresponds to the number of non-C symbols in FullMA. Similarly, Bob maintains data FullMB and q_{MB}.

FullMA and FullMB may not agree due to the transmission errors. To counter this, the two players exchange information about their metadata at the end of each block. Hence, Alice also holds $\widetilde{MB}$ and $q_{\widetilde{MB}}$ as her best estimation of Bob's metadata and the number of MESs he has used, respectively. Similarly, Bob holds $\widetilde{MA}$ and $q_{\widetilde{MA}}$. We use these data to control the simulation; before taking any action in Π', Alice checks if her guess $\widetilde{MB}$ equals FullMB. Bob does the analogous check for his data.

3.3.2 Number of MESs used

Once both parties reconcile their view of each other's metadata with the actual data, they might detect a discrepancy in the number of MESs they have used. The three drawings in Figure 3.1 represent the $\lceil \frac{n}{2r}(1 + \mathrm{O}(r\epsilon)) \rceil$ blocks of $r = \mathrm{O}(\sqrt{1/\epsilon})$ MESs at different points in the protocol: first, before the protocol begins; second, when Alice and Bob have used the same number of MESs; and third, when they are not synchronized, say, Alice has used more blocks of MESs than Bob. A difference in q_{MA} and q_{MB} indicates that the joint state of the protocol Π can no longer be recovered from registers $AC_{\mathsf{A}}C_{\mathsf{B}}B$ alone. Since one party did not correctly complete the teleportation operations, the (possibly erroneous) joint state may be thought of as having "leaked" into the partially measured MESs which were used by only one party. We will elaborate on this scenario in Section 3.3.5.

3.3.3 Pauli data

The last piece of information required to complete the description of what has happened so far on the quantum registers ABC is about the Pauli operators corresponding to teleportation, which we call the "Pauli data". These Pauli data contain information about the teleportation measurement outcomes as well as about the teleportation decoding operations. Since incorrect teleportation decoding may arise due to the transmission errors, we must allow the parties to apply Pauli corrections at some point. We choose to concentrate such Pauli corrections on the receiver's side at the end of each teleportation. These Pauli corrections are computed from the history of all classical data available, before the evolution or reversal of Π in a block starts. The measurement data are directly transmitted over the noisy classical communication channel and the decoding data are directly taken to be the data received over the noisy channel. If there is no transmission error, the decoding Pauli

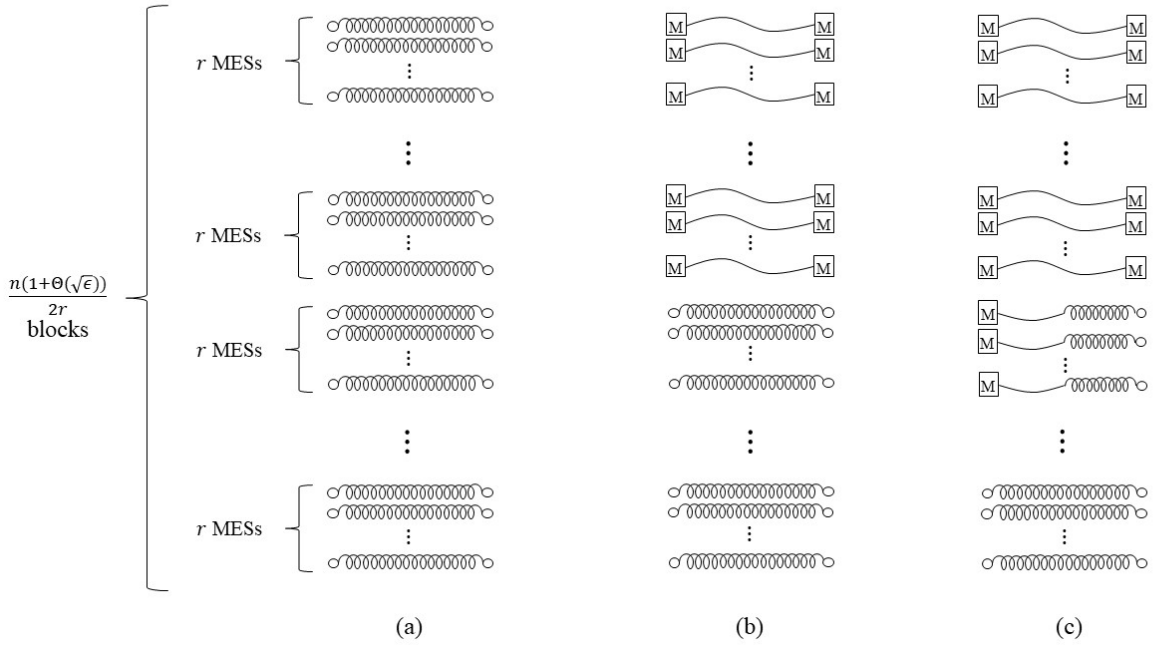

Figure 3.1: These figures represent the MES blocks at different stages of the protocol. The systems depicted by circles have not been used yet for teleportation, those depicted by squares have been used already. (either "Measured" or teleportation-decoded.) Figure (a) represents the MES blocks at the beginning of the protocol, when none have been used. Figure (b) represents them when Alice and Bob have used the same number of them; this is the desired situation. Figure (c) represents a situation when Alice and Bob are out of sync; e.g., Alice has used more MES blocks than Bob. They then work to get back in sync before resuming the simulation.

operation should correspond to the inverse of the effective measurement Pauli operation and cancel out to yield a noiseless quantum channel. Figure 3.2 depicts the different types of Pauli data in a block corresponding to type $+1$ for Alice and -1 for Bob. The Pauli operations applied on Alice's side are in the following order:

> teleportation measurement for the first qudit she sends,
> decoding operation for the first qudit she receives,
> correction operation for the same qudit (the first qudit she receives);
>
> teleportation measurement for the second qudit she sends,
> decoding operation for the second qudit she receives,
> correction operation for the same qudit (the second qudit she receives);
>
> and so on.

The Pauli operations applied on Bob's side are in a different order:

decoding operation for the first qudit he receives,
correction operation for the same qudit (the first qudit he receives),
teleportation measurement for the first qudit he teleports;

decoding operation for the second qudit he receives,
correction operation for the same qudit (the second qudit he receives),
teleportation measurement for the second qudit he sends;

and so on.

Alice records as her Pauli data in the string $FullPA \in (\Sigma^{3r})^*$, the sequence of Pauli operators that are applied on the quantum register on her side. Each block of $FullPA$ is divided into 3 parts of r symbols from the alphabet set Σ. The first part corresponds to the $\frac{r}{2}$ teleportation measurement outcomes with two symbols for each measurement outcome. Each of the $\frac{r}{2}$ teleportation decoding operations are represented by two symbols in the second part. Finally, the third part contains two symbols for each of the $\frac{r}{2}$ Pauli corrections. Similarly, Bob records the sequence of Pauli operators applied on his side in $FullPB$. As described above, the measurement outcomes and the decoding Pauli operations are available to the sender and the receiver, respectively. Based on the message transcript in Π' so far, Alice maintains her best guess $\widetilde{PB}$ for Bob's Pauli data and Bob maintains his best guess $\widetilde{PA}$ for Alice's Pauli data. These data also play an important role in the simulation. Before taking any action in Π', Alice checks if her guess $\widetilde{PB}$ equals $FullPB$. Bob does the analogous check for his data.

Alice and Bob check and synchronize their classical data, i.e., the metadata and Pauli data, by employing the ideas underlying the Haeupler algorithm [37]. Once they agree on each other's metadata and Pauli data, they both possess enough information to compute the content of the quantum register (to the best of their knowledge).

3.3.4 Hashing for string comparison

We use randomized hashes to compare strings and catch disagreements probabilistically. The hash values can be viewed as summaries of the strings to be compared. Usually, a random bit string called the *seed* is used to select a function from the family of hash functions. We say a *hash collision* occurs when a hash function outputs the same value for two unequal strings. In the large alphabet case (Chapters 3 and 4), we use the following family of hash functions based on the ϵ-biased probability spaces constructed in [52].

Lemma 3.3.1 (from [52]). *For any l, any alphabet Σ, and any probability $0 < p < 1$, there exist $s = \Theta(\log(l \log |\Sigma|) + \log \frac{1}{p})$, $o = \Theta(\log \frac{1}{p})$, and a simple function h, which given an s-bit uniformly random seed S maps any string over Σ of length at most l into an o-bit output, such that the collision probability of any two l-symbol strings over Σ is at most p. In short:*

$$\forall l, \Sigma, 0 < p < 1 : \quad \exists s = \Theta(\log(l \log |\Sigma|) + \log \frac{1}{p}) \ , \ o = \Theta(\log \frac{1}{p}) \ , \ h : \{0,1\}^s \times \Sigma^{\leq l} \mapsto \{0,1\}^o$$

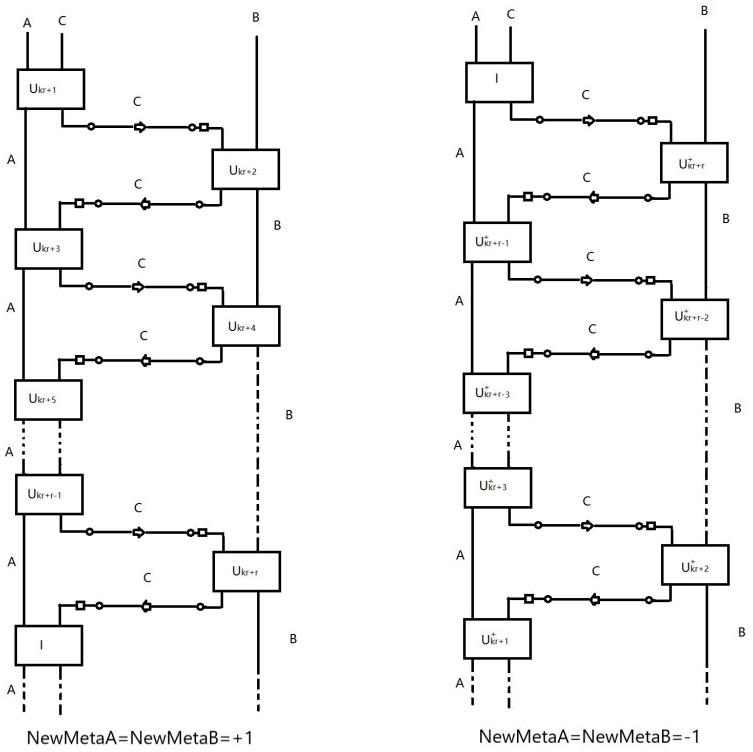

Figure 3.2: Representation of the teleportation-based scheme for a size r block. The figure on the left corresponds to Alice and Bob having blocks of type $+1$, the most common block type, and the one on the right to a block of type -1 for both. The large rectangles correspond to unitary operations of the original protocol or their inverses, or even an identity operator, being applied by Alice or by Bob to AC or BC, respectively. Bob has $r/2$ rectangles and applies a unitary operation or an inverse in each of them whenever he has a block of type ± 1. Alice has $r/2+1$ rectangles and uses the first $r/2$ to apply unitary operations in a block of type $+1$ and apply an identity on the last one, while she applies an identity in the first one and inverses of unitary operations in the $r/2$ last ones in a block of type -1. This is so that a -1 block for Alice can be the inverse of a $+1$ block for Alice, and vice-versa. The small circles correspond to the Pauli operations due to teleportation measurement and teleportation decoding, with the teleportation being from Alice to Bob on odd numbered MESs and from Bob to Alice on even numbered MESs. The small squares on the receiver side right after the teleportation decoding circle corresponds to the Pauli corrections made in order to try to correct errors in previous blocks.

$$\text{s.t.} \quad \forall X, Y \in \Sigma^{\leq l}, X \neq Y, S \in \{0,1\}^s \text{ i.i.d. Bernoulli}(1/2): \quad P[h_S(X) = h_S(Y)] \leq p \ .$$

In our application, the hash family of Lemma 3.3.1 is used to compare $\Theta(n)$-symbol strings,

where n is the length of the input protocol. Therefore, in the large alphabet setting, the collision probability can be chosen to be as low as $p = 1/\text{poly}\,(n)$, while still allowing the hash values to be exchanged using only a constant number of symbols. In the teleportation-based model, where Alice and Bob have access to free pre-shared entanglement, they generate the seeds by measuring the MESs they share in the computational basis.

3.3.5 Out-of-Sync Teleportation

Basic out-of-sync scenario

Consider an iteration in which Alice believes she should implement a $+1$ block, while Bob believes he has to resolve an inconsistency in their classical data. Alice will simulate one block of the input protocol Π, consuming the next block of MESs. On the other hand, Bob will try to resolve the inconsistency through classical communication alone, and not access the quantum registers. Thus Alice will treat Bob's messages as the outcomes of his teleportation measurements, and she performs the teleportation decoding operations according to these messages. The situation is even worse, since Alice sends quantum information to Bob through teleportation of which Bob is unaware, and Bob views the teleportation measurement outcomes sent by Alice as classical information about Alice's local Pauli data and metadata corresponding to previous iterations. Note that at this point the quantum state in registers ABC may potentially be lost. This scenario could continue for several iterations and derail the simulation completely. To recover from such a situation, especially to retrieve the quantum information in the unused MESs at his end, it would seem that Alice and Bob would have to rewind the simulation steps in Π' (and not only the steps of the original protocol Π) to an appropriate point in the past. This rewinding itself would be subject to error, and the situation seems hopeless. Nonetheless, we provide a simple solution to address this kind of error, which translates out-of-sync teleportation to errors in implementing the forward simulation or rewinding of the original protocol Π.

As explained earlier, Alice and Bob first reconcile their view of the history of the simulation stored in their metadata. Through this, suppose they both discover the discrepancy in the number of MESs used. (There are other scenarios as well; for example, they may both think that $q_{MA} = q_{MB}$. These scenarios lead to further errors, but the simulation protocol Π' eventually discovers the difference in MESs used.) In the scenario in which Alice and Bob both discover that $q_{MA} \neq q_{MB}$, they try to "gather" the quantum data hidden in the partially used MESs back into the registers ABC. In more detail, suppose Bob has used fewer MESs than Alice, and he discovers this at the beginning of the i-th iteration. Let $E_1 E_2 \cdots E_r$ be registers with Bob that hold the halves of the *first* block of MESs that Alice has used but Bob has not. Note that $E_1, E_3, \ldots, E_{r-1}$ contain quantum information teleported by Alice, and $E_2, E_4, \ldots, E_r$ are MES-halves intended for teleportation by Bob. The MES-halves corresponding to $E_2, E_4, \ldots, E_r$ have already been used by Alice to "complete" the teleportations she assumed Bob has performed. Say Alice used this block of MESs in the i'-th iteration. In the i-th iteration, Bob teleports the qudit E_1 using the MES-half E_2, E_3

with E_4, and so on. That is, Bob teleports qudit E_j using the MES-half E_{j+1} in increasing order of j, for all odd $j \in [r]$, as if the even numbered MESs had not been used by Alice. The effect of this teleportation is the same as if Alice and Bob had *both* tried to simulate the local operations and communication from the original protocol in the i'-th iteration (in the forward direction or to correct the joint state), *except that the following also happened independently of channel error*:

1. the Pauli operations used by Bob to decode $E_1, E_3, \ldots, E_{r-1}$ were all the identity,
2. the unitary operations used by Bob on the registers BC were all the identity, and
3. the Pauli operations applied by Alice for decoding Bob's teleportation were unrelated to the outcome of Bob's teleportation measurements.

This does not guarantee correctness of the joint state in ABC, but has the advantage that quantum information in the MES-halves $E_1, E_3, \ldots, E_{r-1}$ that is required to restore correctness is redirected back into the registers ABC. In particular, the difference in the number of MESs used by the two parties is reduced, while the errors in the joint quantum state in ABC potentially increase. The errors in the joint state are eventually corrected by reversing the incorrect unitary operations, as in the case when the teleportations are all synchronized.

To understand the phenomenon described above, consider a simpler scenario where Bob wishes to teleport a qudit $|\xi\rangle$ in register B_1 to Alice using an MES in registers E'_1E_1, after which Alice applies the unitary operation V to register E'_1. If they follow the corresponding sequence of operations, the final state would be $V|\xi\rangle$, stored in register E'_1. Instead suppose they do the following. First, Alice applies V to register E'_1, *then* Bob measures registers B_1E_1 in the generalized Bell basis and gets measurement outcome (j, k). He sends this outcome to Alice. We may verify the state of register E'_1 conditioned on the outcome is $V(\mathrm{X}^j\mathrm{Z}^k)|\xi\rangle$. Thus, the quantum information in ξ is redirected to the correct register, albeit with a Pauli error (that is known to Alice because of his message). In particular, Alice may later reverse V to correctly decode the teleported state. The chain of teleportation steps described in the previous paragraph has a similar effect.

3.3.6 First representation of the quantum registers

A first representation for the content of the quantum registers ABC in Π' can be obtained directly and explicitly from the metadata and the Pauli data, and is denoted $JS1$, as in Eq. (3.1) below, with JS standing for "joint state". We emphasize that this is the state conditioned on the outcomes of the teleportation measurements as well as the transcript of classical messages received by the two parties. However, the form $JS1$ is essentially useless for deciding the next action that the simulation protocol Π' should take, but it can be simplified into a more useful representation. This latter form, denoted $JS2$, as

in Eq. (3.2) below, directly corresponds to the further actions we may take in order to evolve the simulation of the original protocol or to actively reverse previous errors. We first consider $JS1$ and $JS2$ in the case when $q_{MA} = q_{MB}$.

We sketch how to obtain $JS1$ from *FullMA*, *FullMB*, *FullPA* and *FullPB* (when $q_{MA} = q_{MB}$). Each block of r MESs which have been used by both Alice and Bob corresponds to a bracketed expression $[*j]$ for some content "$*j$" corresponding to the j-th block that we describe below. The content of the quantum registers is then the ABC part of

$$JS1 = [*q_{MA}] \cdots [*2][*1] \, |\psi_{\text{init}}\rangle^{ABCER}, \tag{3.1}$$

with $|\psi_{\text{init}}\rangle^{ABCER}$ being the initial state of the original protocol. (To be accurate, the representation corresponds to the sequence of operations that have been applied to $|\psi_{\text{init}}\rangle$, and knowledge of $|\psi_{\text{init}}\rangle$ is not required to compute the representation.) It remains to describe the content $*j$ of the j-th bracket. It contains from right to left $\frac{r}{2}$ iterations of the following:

> Alice's unitary operation - Alice's teleportation measurement outcome -
> Bob's teleportation decoding - Bob's Pauli correction - Bob's unitary operation
> - Bob's teleportation measurement outcome -
> Alice's teleportation decoding - Alice's Pauli correction.

It also allows for an additional unitary operation of Alice on the far left when she is implementing a block of type -1; we elaborate on this later. If Alice's block type is $+1$, all her unitary operations are consecutive unitary operations from the original protocol (with the index of the unitary operations depending on the number of ± 1 in *FullMA*), while if it is -1, they are inverses of such unitary operations. If Alice's block type is 0, all unitary operations are equal to the identity on registers AC_{A}. Similar properties hold for Bob's unitary operations on registers BC. Alice's block type corresponds to the content of the j-th non-C element in *FullMA*, and Bob's to the content of the j-th non-C element in *FullMB*. Alice's Pauli data corresponds to the content of the j-th block in *FullPA*, and Bob's to the content of the j-th block in *FullPB*. The precise rules by which Alice and Bob determine their respective types for a block in Π', and which blocks of Π (if any) are involved, are deferred to the next section. Note that when $q_{MA} = q_{MB}$, the first q_{MA} MES blocks have been used by both parties but not necessarily in the same iterations. Nevertheless, the remedial actions the parties have taken to recover from out-of-sync teleportation have reduced the error on the joint state to transmission errors as if all the teleportations were synchronized and the adversary had introduced those additional errors; see Section 3.3.5.

To give a concrete example, suppose from her classical data, Alice determines that in *her* j-th non-C block of Π', she should actively reverse the unitary operations of block k of Π to correct some error in the joint state. So her j-th non-C block of Π' is of type -1. Suppose Alice's Pauli data in the j-th block of *FullPA* correspond to Pauli operators $p_{\mathsf{A},1}p_{\mathsf{A},2}\cdots p_{\mathsf{A},3r/2}$ in the order affecting the joint state; that is, the Pauli operators $p_{\mathsf{A},1}$, $p_{\mathsf{A},4}$, $\ldots$, $p_{\mathsf{A},3(r/2-1)+1}$ correspond to the sequence of Alice's teleportation

measurement outcomes, the Pauli operators $p_{\mathsf{A},2}$, $p_{\mathsf{A},5}$, $\ldots$, $p_{\mathsf{A},3(r/2-1)+2}$ are her teleportation decoding operations and $p_{\mathsf{A},3}$, $p_{\mathsf{A},6}$, $\ldots$, $p_{\mathsf{A},3r/2}$ are her Pauli corrections, respectively. Consider Bob's j-th non-C block of Π'. Note that this may be a different block of Π' than Alice's j-th non-C block. Suppose from *his* classical data, Bob determines that in his j-th non-C block of Π', he should apply the unitary operations of block l of Π to evolve the joint state further. So his j-th non-C block of Π' is of type $+1$. Suppose Bob's Pauli data in the j-th block of *FullPB* correspond to Pauli operators $p_{\mathsf{B},1}p_{\mathsf{B},2}\cdots p_{\mathsf{B},3r/2}$, in the order affecting the joint state; that is, the Pauli operators $p_{\mathsf{B},1}$, $p_{\mathsf{B},4}$, $\ldots$, $p_{\mathsf{B},3(r/2-1)+1}$ are Bob's decoding operations and $p_{\mathsf{B},2}$, $p_{\mathsf{B},5}$, $\ldots$, $p_{\mathsf{B},3(r/2-1)+2}$ are his Pauli corrections and $p_{\mathsf{B},3}$, $p_{\mathsf{B},6}$, $\ldots$, $p_{\mathsf{B},3r/2}$ correspond to his teleportation measurement outcomes, respectively. Then from *FullMA*, *FullMB*, *FullPA*, *FullPB*, we can compute a description of the joint state as in Eq. (3.1), with $*j$ equal to

$$
\begin{aligned}
&U_{kr+1}^{-1} \\
&\times \left(p_{\mathsf{A},3(r/2-1)+3} \;\; p_{\mathsf{A},3(r/2-1)+2}\right) \left(p_{\mathsf{B},3(r/2-1)+3} \;\; U_{lr+r} \;\; p_{\mathsf{B},3(r/2-1)+2} \;\; p_{\mathsf{B},3(r/2-1)+1}\right) \\
&\qquad \times \left(p_{\mathsf{A},3(r/2-1)+1} \;\; U_{kr+3}^{-1}\right) \\
&\times \cdots \\
&\times \left(p_{\mathsf{A},3(s-1)+3} \;\; p_{\mathsf{A},3(s-1)+2}\right) \left(p_{\mathsf{B},3(s-1)+3} \;\; U_{lr+2s} \;\; p_{\mathsf{B},3(s-1)+2} \;\; p_{\mathsf{B},3(s-1)+1}\right) \\
&\qquad \times \left(p_{\mathsf{A},3(s-1)+1} \;\; U_{kr+(r-2s+3)}^{-1}\right) \\
&\times \cdots \\
&\times \left(p_{\mathsf{A},6} \;\; p_{\mathsf{A},5}\right) \left(p_{\mathsf{B},6} \;\; U_{lr+4} \;\; p_{\mathsf{B},5} \;\; p_{\mathsf{B},4}\right) \left(p_{\mathsf{A},4} \;\; U_{kr+(r-1)}^{-1}\right) \\
&\times \left(p_{\mathsf{A},3} \;\; p_{\mathsf{A},2}\right) \left(p_{\mathsf{B},3} \;\; U_{lr+2} \;\; p_{\mathsf{B},2} \;\; p_{\mathsf{B},1}\right) \left(p_{\mathsf{A},1} \;\; \mathbb{1}\right) \quad .
\end{aligned}
$$

Note that Alice and Bob are not necessarily able to compute the state $JS1$. Instead, they use their best guess for the other party's metadata and Pauli data in the procedure described in this section to compute their estimates $JS1^{\mathrm{A}}$ and $JS1^{\mathrm{B}}$ of $JS1$, respectively. Note that Alice and Bob will not compute their estimates of $JS1$ unless they believe that they both know each other's metadata and Pauli data and have used the same number of MES blocks.

3.3.7 Second representation of the quantum registers

To obtain $JS2$ from $JS1$, we first look inside each bracket and recursively cancel consecutive Pauli operators inside the bracket. In case a bracket evaluates to the identity operator on registers ABC, we remove it. Once each bracket has been cleaned up in this way, we recursively try to cancel consecutive brackets if their contents correspond to the inverse of one another (assuming that no two U_i of the original protocol are the same or inverses of one another). Once no such cancellation works out anymore, what we are left with is

representation $JS2$, which is of the following form (when $q_{MA} = q_{MB}$):

$$JS2 = [\#b] \cdots [\#1][U_{gr} \cdots U_{(g-1)r+2} U_{(g-1)r+1}] \cdots [U_r \cdots U_2 U_1] \, |\psi_{\text{init}}\rangle^{ABCER} . \tag{3.2}$$

Here, the first g brackets starting from the right correspond to the "good" part of the simulation, while the last b brackets correspond to the "bad" part of the simulation, the part that Alice and Bob have to actively rewind later. The integer g is determined by the left-most bracket such that along with its contents, those of the brackets to the right equal the sequence of unitary operations $U_1, U_2, \ldots, U_{gr}$ from the original protocol Π in reverse. The brackets to the left of the last g brackets are all considered bad blocks. Thus, the content of $[\#1]$ is not $[U_{(g+1)r} \cdots U_{gr+1}]$, while the contents of $[\#2]$ to $[\#b]$ are arbitrary and have to be actively rewound before Alice and Bob can reverse the content of $[\#1]$.

Once the two parties synchronize their metadata, the number of MESs they have used and their Pauli data, they compute their estimates of $JS1$. Alice uses $JS1^{\text{A}}$ in the above procedure to compute her estimate $JS2^{\text{A}}$ of $JS2$. Similarly, Bob computes $JS2^{\text{B}}$ from $JS1^{\text{B}}$. These in turn determine their course of action in the simulation as described next. If $b > 0$, they actively reverse the incorrect unitary operators in the last bad block, while assuming the other party does the same. They start by applying the inverse of $[\#b]$, choosing appropriately whether to have a type ± 1 or 0 block, and also choosing appropriate Pauli corrections. Else, if $b = 0$, they continue implementing unitary operations U_{gr+1} to $U_{(g+1)r}$ of the original input protocol Π to evolve the simulation. Note that each player has their independent view of the joint state, and takes actions assuming that their view is correct. In this process, Alice and Bob use their view of the joint state to predict each other's next action in the simulation and extend their estimates of each other's metadata and Pauli data accordingly.

We describe a few additional subtleties on how the parties access the quantum register in a given block, as represented in Figure 3.2. First, each block begins and ends with Alice holding register C and being able to perform a unitary operation. In $+1$ blocks, she applies a unitary operation at the beginning and not at the end, whereas in -1 blocks she applies the inverse of a unitary operation at the end and not at the beginning. This is in order to allow a -1 block to be the inverse of a $+1$ block, and vice-versa. Second, whenever Alice and Bob are not synchronized in the number of MESs they have used so far, as explained in Section 3.3.5, the party who has used more will wait for the other to catch up by creating a new type C block while the party who has used less will try to catch up by creating a type 0 block, sequentially feeding the C register at the output of a teleportation decoding to the input of the next teleportation measurement. Notice that due to errors in communication, it might happen that $+1$ blocks are used to correct previous erroneous -1 blocks and 0 blocks are used to correct previous erroneous 0 blocks. As illustrated in Figure 3.2, the block on the right is the inverse of the one on the left if the corresponding Pauli operators are inverses of each other.

3.3.8 Representations of quantum registers while out-of-sync

We now define the $JS1$ and $JS2$ representations of the joint state in the case when $q_{MA} \neq q_{MB}$. Note that in this case, conditioned on the classical data with the two parties, $JS1$ and $JS2$ represent a pure state. However, in addition to the $ABCER$ registers, we must also include the half-used MES registers in the representation. Let $u \stackrel{\text{def}}{=} |q_{MA} - q_{MB}|$. For concreteness, suppose that $q_{MA} > q_{MB}$. Then the $JS1$ representation is of the following form:

$$JS1 = [*q_{MA}] \cdots [*q_{MB}] \cdots [*2][*1] \, |\psi_{\text{init}}\rangle^{ABCER} \quad . \tag{3.3}$$

The content of the first q_{MB} brackets from the right, corresponding to the MES blocks which have been used by both parties are obtained as described in Subsection 3.3.6. The leftmost u brackets correspond to the MES blocks which have been used only by Alice. We refer to these blocks as the *ugly* blocks. These brackets contain Alice's unitary operations from the input protocol, her teleportation decoding operations and Pauli correction operations in her last u non-classical iterations of the simulation. Additionally, they contain the u blocks of MES registers used only by Alice. In each of these blocks, the registers indexed by an odd number have been measured on Alice's side and the state of the MES register has collapsed to a state which is obtained from Alice's Pauli data.

The representation $JS2$ is obtained from $JS1$ as follows: We denote by $[@u] \cdots [@1]$ the leftmost u brackets corresponding to the ugly blocks. We use the procedure described in Subsection 3.3.7 on the rightmost q_{MB} brackets in $JS1$ to obtain $JS2$ of the following form:

$$JS2 = [@u] \cdots [@1][\#b] \cdots [\#1][U_{gr} \cdots U_{(g-1)r+2} U_{(g-1)r+1}] \cdots [U_r \cdots U_2 U_1] \, |\psi_{\text{init}}\rangle^{ABCER} \quad , \tag{3.4}$$

with g *good* blocks, and b *bad* blocks, for some non-negative integers g, b.

Thus, in the rest of this section, we assume that $JS2$ is of the form of Eq. (3.4) at the end of each iteration for some non-negative integers g, b, u which are given by

$$g \stackrel{\text{def}}{=} \text{the number of good unitary blocks in } JS2, \tag{3.5}$$

$$b \stackrel{\text{def}}{=} \text{the number of bad unitary blocks in } JS2, \text{ and} \tag{3.6}$$

$$u \stackrel{\text{def}}{=} |q_{MA} - q_{MB}|. \tag{3.7}$$

We point out that Alice and Bob compute their estimates of $JS1$ and $JS2$ only if, based on their view of the simulation so far, they believe that they have used the same number of MES blocks. Therefore, whenever computed, $JS1^{\text{A}}$, $JS1^{\text{B}}$ and $JS2^{\text{A}}$, $JS2^{\text{B}}$ are always of the forms described in Subsections 3.3.6 and 3.3.7, respectively.

Notice that if there are no transmission errors or hash collisions and Alice and Bob do as described earlier in this section after realizing that $q_{MA} > q_{MB}$, then the ugly blocks

$[@u]\cdots[@2]$ remain as they were while block [@1] becomes a standard block of unitary operations acting on registers ABC only, quite probably being a new bad block, call it $[\#b+1]$. More generally, if there is either a transmission error or a hash collision, Bob might not realize that $q_{MA} > q_{MB}$. Then he might either have a C type of iteration in which case block [@1] also remain as is, or else it is a $+1$, -1 or 0 (non-C) type of iteration and then he may apply non-identity Pauli operations and unitary operations on registers BC, which still results in block [@1] becoming a standard block of unitary operations acting on registers ABC only. Similarly if there is either a transmission error or a hash collision, Alice might not realize that $q_{MA} > q_{MB}$. Then she might have a non-C type of iteration in which case a new ugly block, call it $[@u+1]$, would be added to the left of $[@u]$.

3.3.9 Summary of main steps

The different steps that Alice and Bob follow in the simulation protocol Π' are summarized in Algorithm 1. Recall that each party runs the simulation algorithm based on their view of the simulation so far.

Algorithm 1: Main steps in one iteration of the simulation for the large alphabet teleportation-based model

1. Agree on the history of the simulation contained in the metadata, i.e., ensure $FullMA = \widetilde{MA}$ and $FullMB = \widetilde{MB}$. This involves Algorithm 5—**rewindMD**, and Algorithm 6—**extendMD**.
2. Synchronize the number of MESs used, in particular, ensure $q_{MA} = q_{\widetilde{MB}}$ and $q_{MB} = q_{\widetilde{MA}}$. This involves Algorithm 7—**syncMES**.
3. Agree on Pauli data for all the teleportation steps and additional Pauli corrections for addressing channel errors, i.e., ensure $FullPA = \widetilde{PA}$ and $FullPB = \widetilde{PB}$. This is done via Algorithm 8—**rewindPD** and Algorithm 9—**extendPD**.
4. Compute the best guess for $JS1$ and $JS2$. If there are any "bad" blocks in the guess for $JS2$, reverse the last bad block of unitary operations. I.e., implement quantum rewinding so that $b = 0$ in $JS2$. This is done in Algorithm 11—**simulate**.
5. If no "bad" blocks remain, implement the next block of the original protocol. This results in an increase in g in $JS2$, and is also done through Algorithm 11—**simulate**.

The algorithms mentioned in each step are presented in the next section. Figure 3.3 summarizes the main steps in flowchart form.

In every iteration exactly one of the steps listed in Algorithm 1 is conducted. Alice and Bob skip one step to the next only if the goal of the step has been achieved through the *previous* iterations. The simulation protocol is designed so that unless there is a transmission error or a hash collision in comparing a given type of data, Alice and Bob will go down these steps in tandem, while never returning to a previous step. For instance, once Alice and Bob

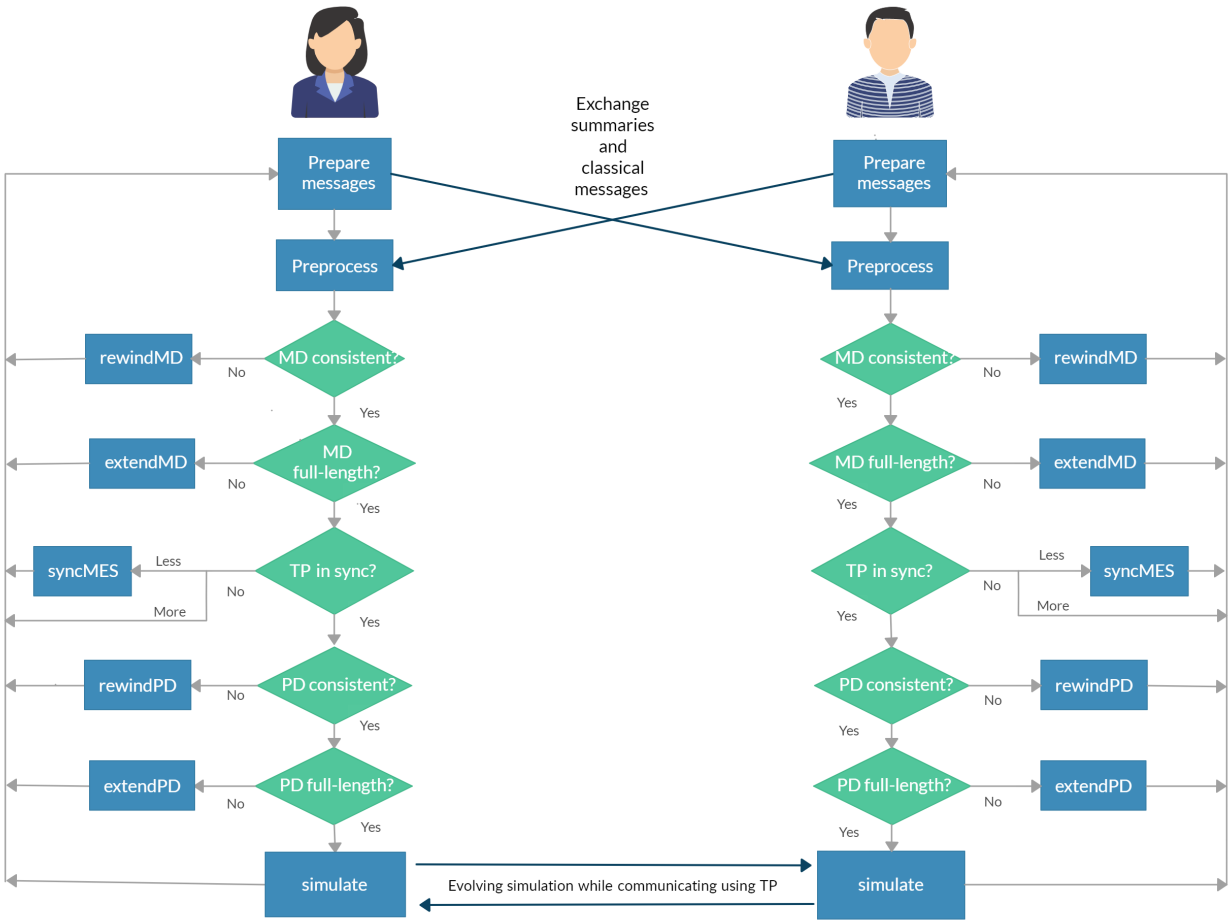

Figure 3.3: Flowchart of the teleportation-based scheme for high rate noisy interactive quantum communication. Most of the communication is spent actually trying to simulate the protocol, in the simulate subroutine.

achieve the goal of step 1, as long as no transmission error or hash collision occurs, their metadata will remain synchronized while they are conducting any of the next steps. This is in fact a crucial property which we utilize in the analysis of the algorithm. In particular, to ensure this property, Alice and Bob need to synchronize the number of MESs they have used *before* synchronizing their Pauli data.

3.4 Algorithm

In this section, we present our simulation protocol Π' in the teleportation-based model when the communication alphabet is polynomial-size. We first introduce the data structure used in our algorithm in this model, which summarizes the definition of the variables appearing

in the pseudo-code.

3.4.1 Data structure

- **Metadata:** In every iteration $NewMetaA \in \{\pm 1, 0, \mathsf{C}\}$ corresponds to Alice's block type which determines how the simulation of the input protocol proceeds locally on Alice's side. $NewMetaA = \mathsf{C}$ corresponds to a classical iteration, in which Alice does not access the quantum registers. $NewMetaA \in \{\pm 1, 0\}$ determines the exponent of the unitary operators from the input protocol Π applied by Alice in the current iteration of the simulation. Alice records her metadata in $FullMA$ which is concatenated with $NewMetaA$ in every iteration and has length i after i iterations. Her best guess of Bob's block type in the current iteration is denoted by $\widetilde{NewMetaB}$. Alice maintains a guess for Bob's metadata in $\widetilde{MB}$ which gets modified or corrected as she gains more information through interaction with Bob. Note that $\widetilde{MB}$ is not necessarily full-length in every iteration and its length may decrease. $\ell_{\widetilde{MB}}$ denotes the length of $\widetilde{MB}$. Bob's local data, $NewMetaB$, $FullMB$, $\widetilde{NewMetaA}$, $\widetilde{MA}$ and $\ell_{\widetilde{MA}}$ are defined similarly.

  Alice maintains a guess ℓ_{MA} for the length of $\widetilde{MA}$, which is with Bob. We define MA to be the prefix of $FullMA$ of length ℓ_{MA}, i.e., $MA \stackrel{\text{def}}{=} FullMA\,[1 : \ell_{MA}]$. When MA appears in any of the algorithms in this section, it is implicitly computed by Alice from $FullMA$ and ℓ_{MA}. The number of MES blocks used by Alice for teleportation is denoted by q_{MA}. We use $q_{\widetilde{MB}}$ to denote Alice's guess of the number of MES blocks used by Bob. Note that q_{MA} and $q_{\widetilde{MB}}$ are the number of 0, 1 and -1 symbols in MA and $\widetilde{MB}$, respectively. Bob's MB, ℓ_{MB}, q_{MB} and $q_{\widetilde{MA}}$ are defined similarly.

- **Pauli data:** In every iteration $NewPauliA \in (\Sigma^r)^3$ consists of three parts: The first part corresponds to the outcomes of Alice's teleportation measurements in the current iteration; the second part corresponds to the received transmissions which determine the teleportation decoding operation and the last part which corresponds to Pauli corrections.

  The Pauli data are recorded locally by Alice in $FullPA$. Starting from the empty string, $FullPA$ is concatenated with $NewPauliA$ whenever Alice implements a non-C iteration. Alice's best guess for Bob's $NewPauliB$ in each iteration is denoted by $\widetilde{NewPauliB}$. She maintains a string $\widetilde{PB}$ as an estimate of Bob's Pauli data. The length of $\widetilde{PB}$ is denoted by $\ell_{\widetilde{PB}}$. Alice also maintains ℓ_{PA}, her estimate for the length of $\widetilde{PA}$, which is with Bob. PA denotes the prefix of $FullPA$ of length ℓ_{PA}, i.e., $PA \stackrel{\text{def}}{=} FullPA\,[1 : \ell_{PA}]$. When PA appears in any of the algorithms in this section, it is implicitly computed by Alice from $FullPA$ and ℓ_{PA}. Bob's local Pauli data $NewPauliB$, $FullPB$, $\widetilde{NewPauliA}$, $\widetilde{PA}$, $\ell_{\widetilde{PA}}$, ℓ_{PB} , PB are defined similarly.

  A critical difference between the metadata and the Pauli data is that the metadata assigns one symbol for each block while the Pauli data assigns $3r$ symbols for each

block.

- We use H with the corresponding data as subscript to denote the hashed data, e.g., H_{MA} denotes the hash value of the string MA.
- The data with $'$ denote the received data after transmission over the noisy channel, e.g., ℓ'_{MB} denotes what Alice receives when Bob sends ℓ_{MB}.
- The variable $\mathit{Itertype} \in \{\mathrm{MD}, \mathrm{PD}, \mathrm{MES}, \mathrm{SIM}\}$ determines the iteration type for the party: MD and PD correspond to iterations where metadata and Pauli data are processed or modified, MES is used for iterations where the party is trying to catch up on the number of used MESs, and SIM corresponds to iterations where the party proceeds with evolving the simulation of Π by applying a block of unitary operators from Π or the inverse of such a block of unitary operators in order to fix an earlier error.
- The variable $\mathit{RewindExtend} \in \{\mathsf{R}, \mathsf{E}\}$ determines in classical iterations if a string of the local metadata or Pauli data is extended or rewound in the current iteration.

3.4.2 Pseudo-code

This section contains the pseudo-codes for the main algorithm and the subroutines that each party runs locally in the simulation protocol. The subroutines are the following: Preprocess, which determines what will happen locally to the classical and quantum data in the current iteration of the simulation; rewindMD and extendMD, which process the local metadata; syncMES which handles the case when the two parties do not agree on the number of MES blocks they have used; rewindPD and extendPD process the local Pauli data; and finally, simulate, in which the player moves on with the simulation of the input protocol according to the information from subroutine Computejointstate of Preprocess. When a party believes that the classical data are fully synchronized, he or she uses the subroutine Computejointstate to extract the necessary information to decide how to evolve the joint quantum state next. This information includes estimates of $JS1$ and $JS2$ defined in Eqs. (3.1) and (3.2), respectively, $\mathit{NewMetaA}$, $\mathit{RewindExtend}$, $\widetilde{\mathit{NewMetaB}}$, Block which represents the index of the block of unitary operations from the input protocol Π the party will perform, P_{Corr} representing Alice's Pauli corrections and $\widetilde{P_{\mathrm{Corr}}}$ representing Alice's guess of Bob's Pauli corrections.

For the subroutines used in the simulation protocol, we list all the global variables accessed by the subroutine as the **Input** at the beginning of the subroutine. Whenever applicable, the relation between the variables when the subroutine is called is stated as the **Promise** and the global variables which are modified by the subroutine are listed as the **Output**.

Remark 3.4.1. The amount of communication in each iteration of Algorithm 2 is independent of the iteration type.

Remark 3.4.2. Since in every iteration of Algorithm 2 the lengths of *FullMA* and *FullMB* increase by 1, in order to be able to catch up on the metadata, Alice and Bob need to communicate two symbols at a time when extending the metadata. This is done by encoding the two symbols into strings of length r of the channel alphabet Σ using the mapping encodeMD in Algorithm 4 and decoding it using the mapping decodeMD in Algorithm 6.

Algorithm 2: Main algorithm (Alice's side)

Input: n round protocol Π in teleportation-based model over polynomial-size alphabet Σ

Initialize
$r \leftarrow \Theta\left(1/\sqrt{\epsilon}\right)$;
$R_{\text{total}} \leftarrow \left\lceil \frac{n}{2r} + \Theta(n\epsilon) \right\rceil$;
$q_{MA}, \ell_{MA}, \ell_{\widetilde{MB}}, \ell_{PA}, \ell_{\widetilde{PB}} \leftarrow 0$;
$MA, \widetilde{MB}, PA, \widetilde{PB} \leftarrow \emptyset$;

$h \leftarrow$ hash function of Lemma 3.3.1 with $p = 1/n^5$ and $o = s = \Theta(\log n)$;

Measure $\Theta\left(R_{\text{total}}\right)$ MESs in the computational basis and record the binary representation of the outcomes in $S_1, \ldots, S_{4R_{\text{total}}}$;
// $4R_{\text{total}}$ **seeds of length** s **for the hash function** h

For $i = 1 \to R_{\text{total}}$
▷ **Preprocessing phase**

$H_{MA} \leftarrow h_{S_{4i-3}}\left(MA\right)$;
$H_{\widetilde{MB}} \leftarrow h_{S_{4i-2}}\left(\widetilde{MB}\right)$;
$H_{PA} \leftarrow h_{S_{4i-1}}\left(PA\right)$;
$H_{\widetilde{PB}} \leftarrow h_{S_{4i}}\left(\widetilde{PB}\right)$;
Send
$$\left(H_{MA}, \ell_{MA}, H_{\widetilde{MB}}, \ell_{\widetilde{MB}}, H_{PA}, \ell_{PA}, H_{\widetilde{PB}}, \ell_{\widetilde{PB}}\right);$$
Receive
$$\left(H'_{\widetilde{MA}}, \ell'_{\widetilde{MA}}, H'_{MB}, \ell'_{MB}, H'_{\widetilde{PA}}, \ell'_{\widetilde{PA}}, H'_{PB}, \ell'_{PB}\right);$$

3.5 Analysis

Our analysis is inspired by a potential function argument used in Ref. [37] to track the progress of the simulation. In order to show the correctness of the above algorithm, we condition on some view of the metadata and Pauli data, i.e., *FullMA*, *MA*, $\widetilde{MA}$, *FullMB*,

Algorithm 3: Main algorithm (Alice's side, cont. from previous page)

```
11
12      Preprocess;
13      if Itertype ≠ SIM then
14          Send msg;
15          Receive msg′;
                                        // messages are communicated alternately
                                                                    ▷ Case i.A
16      if Itertype = MD and RewindExtend = R then
17          rewindMD;
                                                                    ▷ Case i.B
18      else if Itertype = MD and RewindExtend = E then
19          extendMD;
                                                                    ▷ Case ii.A
20      else if Itertype = MES and NewMetaA = C then
21          return;
                                                                    ▷ Case ii.B
22      else if Itertype = MES and NewMetaA = 0 then
23          syncMES;
                                                                    ▷ Case iii.A
24      else if Itertype = PD and RewindExtend = R then
25          rewindPD;
                                                                    ▷ Case iii.B
26      else if Itertype = PD and RewindExtend = E then
27          extendPD;
                                        // Classical data are synchronized
                                                                    ▷ Case iv
28      else
29          simulate.
30  return Main algorithm;
```

Algorithm 4: Preprocess (Alice's side)

Input:

$$\left(\begin{array}{c} H_{MA}, \ell_{MA}, H_{\widetilde{MB}}, \ell_{\widetilde{MB}}, H_{PA}, \ell_{PA}, H_{\widetilde{PB}}, \ell_{\widetilde{PB}} \\ H'_{\widetilde{MA}}, \ell'_{\widetilde{MA}}, H'_{MB}, \ell'_{MB}, H'_{\widetilde{PA}}, \ell'_{\widetilde{PA}}, H'_{PB}, \ell'_{PB} \\ \mathit{FullMA}, \widetilde{MB}, \mathit{FullPA}, \widetilde{PB}, q_{MA} \end{array}\right)$$

Output:

$$\left(\mathit{Itertype}, \mathit{RewindExtend}, \mathit{NewMetaA}, \mathit{FullMA}, \ell_{MA}, \widetilde{\mathit{NewMetaB}}, \ell_{\widetilde{MB}}, \mathit{msg}\right)$$

if $\left(H_{MA}, H_{\widetilde{MB}}\right) = \left(H'_{\widetilde{MA}}, H'_{MB}\right)$ and $\ell_{MA} = \ell'_{\widetilde{MA}} = \ell_{\widetilde{MB}} = \ell'_{MB} = i-1$ **then**
  Compute $q_{\widetilde{MB}}$;

▷ `Processing metadata`

▷ `Case i.A`

if $\left(H_{MA}, H_{\widetilde{MB}}, \ell_{MA}, \ell_{\widetilde{MB}}\right) \neq \left(H'_{\widetilde{MA}}, H'_{\mathrm{MB}}, \ell'_{\widetilde{MA}}, \ell'_{MB}\right)$ **then**
  $\mathit{Itertype} \leftarrow \mathrm{MD}$;
  $\mathit{RewindExtend} \leftarrow \mathsf{R}$;
  $\mathit{NewMetaA} \leftarrow \mathsf{C}$;
  $\mathit{FullMA} \leftarrow (\mathit{FullMA}, \mathit{NewMetaA})$;
  $\mathit{msg} \leftarrow$ dummy message of length r;

▷ `Case i.B`

else if $(\ell_{MA} < i-1)$ or $\left(\ell_{\widetilde{MB}} < i-1\right)$ **then**
  $\mathit{Itertype} \leftarrow \mathrm{MD}$;
  $\mathit{RewindExtend} \leftarrow \mathsf{E}$;
  $\mathit{NewMetaA} \leftarrow \mathsf{C}$;
  $\mathit{FullMA} \leftarrow (\mathit{FullMA}, \mathit{NewMetaA})$;
  if $\ell_{MA} < i-1$ **then**
    $\mathit{msg} \leftarrow \mathrm{encodeMD}\left(\mathit{FullMA}\left[\ell_{MA}+1, \ell_{MA}+2\right]\right)$; `// Encode MD in` Σ^r
  else
    $\mathit{msg} \leftarrow$ dummy message of length r;

▷ `Comparing number of used MES blocks`

▷ `Case ii.A`

else if $q_{MA} > q_{\widetilde{MB}}$ **then**
  $\mathit{Itertype} \leftarrow \mathrm{MES}$;
  $\mathit{NewMetaA} \leftarrow \mathsf{C}$;
  $\mathit{FullMA} \leftarrow (\mathit{FullMA}, \mathit{NewMetaA})$;
  $\ell_{MA} \leftarrow \ell_{MA} + 1$;
  $\widetilde{\mathit{NewMetaB}} \leftarrow 0$;
  $\widetilde{MB} \leftarrow \left(\widetilde{MB}, \widetilde{\mathit{NewMetaB}}\right)$;
  $\ell_{\widetilde{MB}} \leftarrow \ell_{\widetilde{MB}} + 1$;
  $\mathit{msg} \leftarrow$ dummy message of length r;

Algorithm 4: Preprocess (Alice's side, cont. from previous page)

▷ `Case ii.B`

else if $q_{MA} < q_{\widetilde{MB}}$ **then**
  $Itertype \leftarrow \mathrm{MES}$;
  $NewMetaA \leftarrow \mathsf{0}$;
  $FullMA \leftarrow (FullMA, NewMetaA)$;
  $\ell_{MA} \leftarrow \ell_{MA} + 1$;
  $\widetilde{NewMetaB} \leftarrow \mathsf{C}$;
  $\widetilde{MB} \leftarrow \left(\widetilde{MB}, \widetilde{NewMetaB}\right)$;
  $\ell_{\widetilde{MB}} \leftarrow \ell_{\widetilde{MB}} + 1$;
  $msg \leftarrow$ dummy message of length r;

▷ `Processing Pauli data`

▷ `Case iii.A`

else if $\left(H_{PA}, H_{\widetilde{PB}}, \ell_{PA}, \ell_{\widetilde{PB}}\right) \neq \left(H'_{\widetilde{PA}}, H'_{PB}, \ell'_{\widetilde{PA}}, \ell'_{PB}\right)$ **then**
  $Itertype \leftarrow \mathrm{PD}$;
  $RewindExtend \leftarrow \mathsf{R}$;
  $NewMetaA \leftarrow \mathsf{C}$;
  $FullMA \leftarrow (FullMA, NewMetaA)$;
  $\ell_{MA} \leftarrow \ell_{MA} + 1$;
  $\widetilde{NewMetaB} \leftarrow \mathsf{C}$;
  $\widetilde{MB} \leftarrow \left(\widetilde{MB}, \widetilde{NewMetaB}\right)$;
  $\ell_{\widetilde{MB}} \leftarrow \ell_{\widetilde{MB}} + 1$;
  $msg \leftarrow$ dummy message of length r;

▷ `Case iii.B`

else if $(\ell_{PA} < 3q_{MA} \cdot r)$ or $\left(\ell_{\widetilde{PB}} < 3q_{\widetilde{MB}} \cdot r\right)$ **then**
  $Itertype \leftarrow \mathrm{PD}$;
  $RewindExtend \leftarrow \mathsf{E}$;
  $NewMetaA \leftarrow \mathsf{C}$;
  $FullMA \leftarrow (FullMA, NewMetaA)$;
  $\ell_{MA} \leftarrow \ell_{MA} + 1$;
  $\widetilde{NewMetaB} \leftarrow \mathsf{C}$;
  $\widetilde{MB} \leftarrow \left(\widetilde{MB}, \widetilde{NewMetaB}\right)$;
  $\ell_{\widetilde{MB}} \leftarrow \ell_{\widetilde{MB}} + 1$;
  if $\ell_{PA} < 3q_{MA} \cdot r$ **then**
    $msg \leftarrow FullPA\,[\ell_{PA} + 1, \ell_{PA} + r]$

Algorithm 4: Preprocess (Alice's side, cont. from previous page)

▷ `Processing joint quantum state`
▷ `Case iv`

else
  $Itertype \leftarrow$ SIM;
  computejointstate;
  $FullMA = (FullMA, NewMetaA)$;
  $\ell_{MA} \leftarrow \ell_{MA} + 1$;
  $\widetilde{MB} \leftarrow \left(\widetilde{MB}, \widetilde{NewMetaB}\right)$;
  $\ell_{\widetilde{MB}} \leftarrow \ell_{\widetilde{MB}} + 1$;
return Preprocess;

Algorithm 5: rewindMD (Alice's side)

Input: $\left(H_{MA}, \ell_{MA}, H_{\widetilde{MB}}, \ell_{\widetilde{MB}}, H'_{\widetilde{MA}}, \ell'_{\widetilde{MA}}, H'_{MB}, \ell'_{MB}\right)$
Promise: $\left(H_{MA}, H_{\widetilde{MB}}, \ell_{MA}, \ell_{\widetilde{MB}}\right) \neq \left(H'_{\widetilde{MA}}, H'_{MB}, \ell'_{\widetilde{MA}}, \ell'_{MB}\right)$.
Output: (ℓ_{MA}, ℓ'_{MB})
if $\ell_{MA} \neq \ell'_{\widetilde{MA}}$ or $\ell_{\widetilde{MB}} \neq \ell'_{MB}$ **then**
  if $\ell_{MA} > \ell'_{\widetilde{MA}}$ **then**
    $\ell_{MA} \leftarrow \ell_{MA} - 1$;
  if $\ell_{\widetilde{MB}} > \ell'_{MB}$ **then**
    $\ell_{\widetilde{MB}} \leftarrow \ell_{\widetilde{MB}} - 1$;
else
  if $H_{MA} \neq H'_{\widetilde{MA}}$ **then**
    $\ell_{MA} \leftarrow \ell_{MA} - 1$;
  if $H_{\widetilde{MB}} \neq H'_{MB}$ **then**
    $\ell_{\widetilde{MB}} \leftarrow \ell_{\widetilde{MB}} - 1$;
return rewindMD;

Algorithm 6: extendMD (Alice's side)

Input: $\left(\ell_{MA}, \ell_{\widetilde{MB}}, \widetilde{MB}, msg', i\right)$

Promise: $\left(H_{MA}, H_{\widetilde{MB}}, \ell_{MA}, \ell_{\widetilde{MB}}\right) = \left(H'_{\widetilde{MA}}, H'_{MB}, \ell'_{\widetilde{MA}}, \ell'_{MB}\right)$,
$\ell_{MA} < i-1 \quad \text{or} \quad \ell_{\widetilde{MB}} < i-1.$

Output: $\left(\ell_{MA}, \widetilde{MB}, \ell_{\widetilde{MB}}\right)$

if $\ell_{MA} < i-1$ **then**
  $\ell_{MA} \leftarrow \ell_{MA} + 2$;
else if $\ell_{MA} = i-1$ **then**
  $\ell_{MA} \leftarrow \ell_{MA} + 1$;
if $\ell_{\widetilde{MB}} < i-1$ **then**
  $\widetilde{MB}\left[\ell_{\widetilde{MB}}+1, \ell_{\widetilde{MB}}+2\right] \leftarrow \text{decodeMD}\left(msg'\right)$; `// decode MD from` Σ^r
  $\ell_{\widetilde{MB}} \leftarrow \ell_{\widetilde{MB}} + 2$;
else if $\ell_{\widetilde{MB}} = i-1$ **then**
  $\widetilde{MB} \leftarrow \left(\widetilde{MB}, \mathsf{C}\right)$;
  $\ell_{\widetilde{MB}} \leftarrow \ell_{\widetilde{MB}} + 1$;
return extendMD;

Algorithm 7: syncMES (Alice's side)

Input: $\left(FullPA, q_{MA}\right)$

Promise: $\left(H_{MA}, H_{\widetilde{MB}}, \ell_{MA}, \ell_{\widetilde{MB}}\right) = \left(H'_{\widetilde{MA}}, H'_{MB}, \ell'_{\widetilde{MA}}+1, \ell'_{MB}+1\right)$,
$\ell_{MA} = \ell_{\widetilde{MB}} = i$, $q_{MA} < q_{\widetilde{MB}}$.

Output: $q_{MA}, NewPauliA, FullPA$

Recall that $A'B'C'$ are the registers that are used to generate the joint quantum state of the protocol being simulated, and C' is the communication register;
Let $E_1 E_2 \cdots E_r$ be the r registers with Alice that contain halves of the block of r MESs with indices in the interval $(q_{\text{MA}} \cdot r,\ (q_{\text{MA}}+1) \cdot r]$;
Teleport C' using E_1; then teleport E_2 using E_3, E_4 using E_5, and so on (i.e., teleport E_j using E_{j+1} for even $j \in [r-2]$), and then store E_r in register C' ;
`// See Section 3.3.5 for the rationale, and Bob's analogue of this step`
Store the teleportation measurement outcomes in $m \in \Sigma^r$;
$NewPauliA \leftarrow (m, 0^r, 0^r)$;
$FullPA \leftarrow (FullPA, NewPauliA)$;
$q_{MA} \leftarrow q_{MA} + 1$;
return syncMES;

Algorithm 8: rewindPD (Alice's side)

Input: $\left(H_{PA}, \ell_{PA}, H_{\widetilde{PB}}, \ell_{\widetilde{PB}}, H'_{\widetilde{PA}}, \ell'_{\widetilde{PA}}, H'_{PB}, \ell'_{PB}\right)$

Promise: $\left(H_{MA}, H_{\widetilde{MB}}, \ell_{MA}, \ell_{\widetilde{MB}}\right) = \left(H'_{\widetilde{MA}}, H'_{MB}, \ell'_{\widetilde{MA}} + 1, \ell'_{MB} + 1\right)$,
$\ell_{MA} = \ell_{\widetilde{MB}} = i$, $q_{MA} = q_{\widetilde{MB}}$,
$\left(H_{PA}, H_{\widetilde{PB}}, \ell_{PA}, \ell_{\widetilde{PB}}\right) \neq \left(H'_{\widetilde{PA}}, H'_{PB}, \ell'_{\widetilde{PA}}, \ell'_{PB}\right)$.

Output: $\left(\ell_{PA}, \ell_{\widetilde{PB}}\right)$

if $\ell_{PA} \neq \ell'_{\widetilde{PA}}$ or $\ell_{\widetilde{PB}} \neq \ell'_{PB}$ **then**
  if $\ell_{PA} > \ell'_{\widetilde{PA}}$ **then**
    $\ell_{PA} \leftarrow \ell_{PA} - r$;
  if $\ell_{\widetilde{PB}} > \ell'_{PB}$ **then**
    $\ell_{\widetilde{PB}} \leftarrow \ell_{\widetilde{PB}} - r$;
else
  if $H_{PA} \neq H'_{\widetilde{PA}}$ **then**
    $\ell_{PA} \leftarrow \ell_{PA} - r$;
  if $H_{\widetilde{PB}} \neq H'_{PB}$ **then**
    $\ell_{\widetilde{PB}} \leftarrow \ell_{\widetilde{PB}} - r$;
return rewindPD;

Algorithm 9: extendPD (Alice's side)

Input: $\left(\ell_{PA}, \ell_{\widetilde{PB}}, \widetilde{PB}, q_{MA}, q_{\widetilde{MB}}, msg'\right)$

Promise: $\left(H_{MA}, H_{\widetilde{MB}}, \ell_{MA}, \ell_{\widetilde{MB}}\right) = \left(H'_{\widetilde{MA}}, H'_{MB}, \ell'_{\widetilde{MA}} + 1, \ell'_{MB} + 1\right)$,
$\ell_{MA} = \ell_{\widetilde{MB}} = i$, $q_{MA} = q_{\widetilde{MB}}$,
$\left(H_{PA}, H_{\widetilde{PB}}, \ell_{PA}, \ell_{\widetilde{PB}}\right) = \left(H'_{\widetilde{PA}}, H'_{PB}, \ell'_{\widetilde{PA}}, \ell'_{PB}\right)$,
$\ell_{PA} < 3q_{MA} \cdot r \quad \text{or} \quad \ell_{\widetilde{PB}} < 3q_{\widetilde{MB}} \cdot r$.

Output: $\left(\ell_{PA}, \widetilde{PB}, \ell_{\widetilde{PB}}\right)$

if $\ell_{PA} < 3q_{MA} \cdot r$ **then**
  $\ell_{PA} \leftarrow \ell_{PA} + r$;
if $\ell_{\widetilde{PB}} < 3q_{\widetilde{MB}} \cdot r$ **then**
  $\widetilde{PB}\left[\ell_{\widetilde{PB}} + 1 : \ell_{\widetilde{PB}} + r\right] \leftarrow msg'$;
  $\ell_{\widetilde{PB}} \leftarrow \ell_{\widetilde{PB}} + r$;
return extendPD;

Algorithm 10: Computejointstate (Alice's side)

Input: $\left(FullMA, \widetilde{MB}, FullPA, \widetilde{PB}\right)$

Promise: $\left(H_{MA}, H_{\widetilde{MB}}, \ell_{MA}, \ell_{\widetilde{MB}}\right) = \left(H'_{\widetilde{MA}}, H'_{MB}, \ell'_{\widetilde{MA}}, \ell'_{MB}\right)$,
$\ell_{MA} = \ell_{\widetilde{MB}} = i - 1$, $q_{MA} = q_{\widetilde{MB}}$,
$\left(H_{PA}, H_{\widetilde{PB}}, \ell_{PA}, \ell_{\widetilde{PB}}\right) = \left(H'_{\widetilde{PA}}, H'_{PB}, \ell'_{\widetilde{PA}}, \ell'_{PB}\right)$,
$\ell_{PA} = \ell'_{\widetilde{PA}} = 3q_{MA} \cdot r$, $\ell_{\widetilde{PB}} = \ell'_{PB} = 3q_{\widetilde{MB}} \cdot r$.

Output: $\left(JS1^{\mathrm{A}}, JS2^{\mathrm{A}}, NewMetaA, \widetilde{NewMetaB}, Block, RewindExtend, P_{\mathrm{Corr}}, \widetilde{P_{\mathrm{Corr}}}\right)$

Compute $JS1^{\mathrm{A}}$;
Compute $JS2^{\mathrm{A}}$;
Compute $NewMetaA$;
Compute $RewindExtend$;
Compute $\widetilde{NewMetaB}$;
Compute $Block$;
Compute P_{Corr};
Compute $\widetilde{P_{\mathrm{Corr}}}$;
`// Refer to Sections 3.3.6, 3.3.7 to see how these variables are computed`
return Computejointstate;

Algorithm 11: simulate (Alice's side)

Input: $\left(q_{MA}, FullPA, \ell_{PA}, \widetilde{PB}, \ell_{\widetilde{PB}}, RewindExtend, NewMetaA, Block, P_{\mathrm{Corr}}, \widetilde{P_{\mathrm{Corr}}}\right)$

Promise: $\left(H_{MA}, H_{\widetilde{MB}}, \ell_{MA}, \ell_{\widetilde{MB}}, q_{MA}\right) = \left(H'_{\widetilde{MA}}, H'_{MB}, \ell'_{\widetilde{MA}} + 1, \ell'_{MB} + 1, q_{\widetilde{MB}}\right)$,
$\ell_{MA} = \ell_{\widetilde{MB}} = i$, $\left(H_{PA}, H_{\widetilde{PB}}, \ell_{PA}, \ell_{\widetilde{PB}}\right) = \left(H'_{\widetilde{PA}}, H'_{PB}, \ell'_{\widetilde{PA}}, \ell'_{PB}\right)$,
$\ell_{PA} = \ell_{\widetilde{PB}} = 3q_{MA} \cdot r$

Output: $\left(FullPA, \ell_{PA}, \widetilde{PB}, \ell_{\widetilde{PB}}\right)$

Continue the simulation of the input protocol according to $Block$, $NewMetaA$ and P_{Corr};
Record all teleportation measurement outcomes in α;
Record all received Bob's teleportation measurement outcomes in β;

Algorithm 11: simulate (Alice's side, cont. from previous page)

$NewPauliA \leftarrow (\alpha, \beta, P_{\text{Corr}})$;
$FullPA \leftarrow (FullPA, NewPauliA)$;
$\ell_{PA} \leftarrow \ell_{PA} + 3r$;
$\widetilde{NewPauliB} \leftarrow \left(\beta, \alpha, \widetilde{P_{\text{Corr}}}\right)$;
$\widetilde{PB} \leftarrow \left(\widetilde{PB}, \widetilde{NewPauliB}\right)$;
$\ell_{\widetilde{PB}} \leftarrow \ell_{\widetilde{PB}} + 3r$;
$q_{MA} \leftarrow q_{MA} + 1$;
return simulate;

MB, $\widetilde{MB}$, $FullPA$, PA, $\widetilde{PA}$, $FullPB$, PB and $\widetilde{PB}$. We define a potential function Φ as

$$\Phi \stackrel{\text{def}}{=} \Phi_{\text{Q}} + \Phi_{\text{MD}} + \Phi_{\text{PD}} \ ,$$

where Φ_{MD} and Φ_{PD} measure the correctness of the two parties' current estimate of each other's metadata and Pauli data, respectively, and Φ_{Q} measures the progress in reproducing the joint state of the input protocol. We define

$$md_{+}^{\text{A}} \stackrel{\text{def}}{=} \text{ the length of the longest prefix where } MA \text{ and } \widetilde{MA} \text{ agree;} \tag{3.8}$$

$$md_{+}^{\text{B}} \stackrel{\text{def}}{=} \text{ the length of the longest prefix where } MB \text{ and } \widetilde{MB} \text{ agree;} \tag{3.9}$$

$$md_{-}^{\text{A}} \stackrel{\text{def}}{=} \max\{\ell_{MA}, \ell_{\widetilde{MA}}\} - md_{+}^{\text{A}}; \tag{3.10}$$

$$md_{-}^{\text{B}} \stackrel{\text{def}}{=} \max\{\ell_{MB}, \ell_{\widetilde{MB}}\} - md_{+}^{\text{B}}; \tag{3.11}$$

$$pd_{+}^{\text{A}} \stackrel{\text{def}}{=} \lfloor \frac{1}{r} \times \text{ the length of the longest prefix where } PA \text{ and } \widetilde{PA} \text{ agree} \rfloor; \tag{3.12}$$

$$pd_{+}^{\text{B}} \stackrel{\text{def}}{=} \lfloor \frac{1}{r} \times \text{ the length of the longest prefix where } PB \text{ and } \widetilde{PB} \text{ agree} \rfloor; \tag{3.13}$$

$$pd_{-}^{\text{A}} \stackrel{\text{def}}{=} \frac{1}{r} \max\{\ell_{PA}, \ell_{\widetilde{PA}}\} - pd_{+}^{\text{A}}; \tag{3.14}$$

$$pd_{-}^{\text{B}} \stackrel{\text{def}}{=} \frac{1}{r} \max\{\ell_{PB}, \ell_{\widetilde{PB}}\} - pd_{+}^{\text{B}}. \tag{3.15}$$

Also, recall that

$$g \stackrel{\text{def}}{=} \text{the number of good unitary blocks in } JS2, \tag{3.16}$$

$$b \stackrel{\text{def}}{=} \text{the number of bad unitary blocks in } JS2, \text{ and} \tag{3.17}$$

$$u \stackrel{\text{def}}{=} |q_{MA} - q_{MB}|, \tag{3.18}$$

with q_{MA} and q_{MB} the number of non-C iterations for Alice and Bob, respectively.

Now we are ready to define the components of the potential function. At the end of the i-th iteration, we let

$$\Phi_{\mathrm{Q}} \stackrel{\mathrm{def}}{=} g - b - 5u \ , \tag{3.19}$$

$$\Phi_{\mathrm{MD}} \stackrel{\mathrm{def}}{=} md_{+}^{\mathrm{A}} - 3md_{-}^{\mathrm{A}} + md_{+}^{\mathrm{B}} - 3md_{-}^{\mathrm{B}} - 2i \ , \tag{3.20}$$

$$\Phi_{\mathrm{PD}} \stackrel{\mathrm{def}}{=} pd_{+}^{\mathrm{A}} - pd_{-}^{\mathrm{A}} + pd_{+}^{\mathrm{B}} - pd_{-}^{\mathrm{B}} - 3q_{MA} - 3q_{MB} \ , \tag{3.21}$$

$$\Phi \stackrel{\mathrm{def}}{=} \Phi_{\mathrm{Q}} + \Phi_{\mathrm{MD}} + \Phi_{\mathrm{PD}} \ . \tag{3.22}$$

where g, b and and u are defined in Eqs. (3.16), (3.17), and (3.18).

Lemma 3.5.1. *Throughout the algorithm, it holds that*

- $\Phi_{\mathrm{MD}} \leq 0$ *with equality if and only if Alice and Bob have full knowledge of each other's metadata, i.e.,* $md_{+}^{\mathrm{A}} = md_{+}^{\mathrm{B}} = i$ *and* $md_{-}^{\mathrm{A}} = md_{-}^{\mathrm{B}} = 0$.
- $\Phi_{\mathrm{PD}} \leq 0$ *with equality if and only if Alice and Bob have full knowledge of each other's Pauli data, i.e.,* $pd_{+}^{\mathrm{A}} = 3q_{MA}$, $pd_{+}^{\mathrm{B}} = 3q_{MB}$ *and* $pd_{-}^{\mathrm{A}} = pd_{-}^{\mathrm{B}} = 0$.

Proof. The first statement follows from the property that $md_{+}^{\mathrm{A}}, md_{+}^{\mathrm{B}} \leq i$, and the second statement holds since $pd_{+}^{\mathrm{A}} \leq 3q_{MA}$ and $pd_{+}^{\mathrm{B}} \leq 3q_{MB}$. □

Note that if $g - b - u \geq n/2r$, the noiseless protocol embedding described in Section 2.2.1, guarantees that not only is the correct final state of the original protocol produced and swapped into the safe registers $\tilde{A}$, $\tilde{B}$ and $\tilde{C}$, but also they remain untouched by the bad and ugly blocks of the simulation. Therefore, by Lemma 3.5.1, for successful simulation of an n-round protocol it suffices to have $\Phi \geq n/2r$, at the end of the simulation.

The main result of this section is the following:

Theorem 3.2.1 (Restated). *Consider any n-round alternating two-party communication protocol Π in the teleportation-based model, communicating messages over a noiseless channel with an alphabet Σ of bit-size $\Theta(\log n)$. Algorithm 2 is a computationally efficient coding scheme which given Π, simulates it with probability at least $1 - 2^{-\Theta(n\epsilon)}$, over any fully adversarial error channel with alphabet Σ and error rate ϵ. The simulation uses $n(1 + \Theta(\sqrt{\epsilon}))$ rounds of communication, and therefore achieves a communication rate of $1 - \Theta(\sqrt{\epsilon})$. Furthermore. the computational complexity of the coding operations is $\mathrm{O}(n^2)$.*

Proof Outline. We prove that any iteration without an error or hash collision increases the potential by at least one while any iteration with error or hash collision reduces the potential by at most some fixed constant. As in Ref. [37], with very high probability the number of hash collisions is at most $\mathrm{O}(n\epsilon)$, the same order of magnitude as the number of errors, therefore negligible. Finally, our choice of the total number of iterations, $R_{total} \stackrel{\mathrm{def}}{=} \lceil n/2r + \kappa n\epsilon \rceil$ (for a sufficiently large constant κ), guarantees an overall potential increase of at least $n/2r$. As explained above, this suffices to prove successful simulation of the input protocol.

Lemma 3.5.2. *Each iteration of the Main Algorithm (Algorithm 2) without a hash collision or error increases the potential* Φ *by at least* 1*.*

Proof. Note that in an iteration with no error or hash collision, Alice and Bob agree on the iteration type. Moreover, if *Itertype* = MD or PD (Case i or iii), they also agree on whether they extend or rewind the data (the subcase A or B), and if *Itertype* = MES (Case ii), then exactly one of them is in Case A and the other one is in Case B. We analyze the potential function in each of the cases, keeping in mind that we only encounter Case ii or later cases once the metadata of the two parties are consistent and of full length, and similarly, that we encounter Case iv once the parties have used the same number of MESs and the Pauli data with the two parties are consistent and of full length. Lemma 3.5.1 guarantees that Φ_{MD} becomes 0 on entering Case ii, and that $\Phi_{\mathrm{MD}} = \Phi_{\mathrm{PD}} = 0$ on entering Case iv.

- Alice and Bob are in Case i.A:
  - Φ_{PD} and Φ_{Q} stay the same.
  - i increases by 1.
  - md_+^{A} and md_+^{B} stay the same.
  - None of md_-^{A} and md_-^{B} increases, and at least one decreases by 1.

  Therefore, Φ_{MD} increases at least by $3 - 2 = 1$, and so does Φ.
- Alice and Bob are in Case i.B:
  - Φ_{PD} and Φ_{Q} stay the same.
  - i increases by 1.
  - md_-^{A} and md_-^{B} stay at 0.
  - At least one of ℓ_{MA} or ℓ_{MB} is smaller than $i - 1$; If only $\ell_{MA} < i - 1$, then md_+^{A} increases by 2, and md_+^{B} by 1. The case where only $\ell_{MB} < i - 1$ is similar. If both are smaller than $i - 1$, then md_+^{A} and md_+^{B} both increase by 2.

  Therefore, Φ_{MD} increases by at least $3 - 2 = 1$, and so does Φ.
- Alice is in Case ii.A, Bob is in Case ii.B:
  - Φ_{MD} stays at 0.
  - q_{MB} increases by 1.
  - q_{MA}, pd_+^{A}, pd_-^{A}, pd_+^{B}, pd_-^{B} all stay the same.
  - g remains the same, b increases by at most 1, and u decreases by 1.

  Therefore, Φ_{Q} increases by at least $5 - 1 = 4$, and Φ_{PD} decreases by 3. So Φ increases by at least 1.

- Alice is in Case ii.B, Bob is in Case ii.A: This case is similar to the one above.
- Alice and Bob are in Case iii.A
  - Φ_{MD} stays at 0, and Φ_{Q} stays the same
  - pd_+^{A}, pd_+^{B}, q_{MA} and q_{MB} stay the same.
  - None of pd_-^{A} and pd_-^{B} increases, and at least one decreases by 1.

  Therefore, Φ_{PD} increases by at least 1, and so does Φ.
- Alice and Bob are in Case iii.B
  - Φ_{MD} stays at 0, and Φ_{Q} stays the same.
  - pd_-^{A}, pd_-^{B} stay at 0, and q_{MA}, q_{MB} stay the same.
  - At least one of the following holds: $\ell_{PA} < 3q_{MA} \cdot r$, in which case pd_+^{A} increases by 1 (otherwise it remains unchanged), or $\ell_{PB} < 3q_{MB} \cdot r$, and then pd_+^{B} increases by 1 (otherwise it remains unchanged).

  Therefore, Φ_{PD} increases by at least 1, and so does Φ.
- Alice and Bob are in Case iv
  - Φ_{MD} and Φ_{PD} stay at 0.
  - u stays at 0
  - Either g stays the same and b decreases by 1 (when $b \neq 0$) or b stays at 0 and g increases by 1.

  Therefore, Φ_{Q} increases by 1, and so does Φ.

Hence Φ increases at least by 1 for each iteration of the algorithm without a hash collision or error.

□

Lemma 3.5.3. *Each iterations of Algorithm 2, regardless of the number of hash collisions and errors, decreases the potential* Φ *by at most* 45.

Proof. At each step, i increases by 1 while, in the worst case, g, md_+^{A},md_+^{B}, pd_+^{A} and pd_+^{B} decrease by at most 1, b, u, q_{MA} and q_{MB} increase by at most 1, md_-^{A} and md_-^{B} increase by at most 3 and pd_-^{A} and pd_-^{B} increase by at most 4. Hence, Φ_{Q}, Φ_{MD} and Φ_{PD} decrease at most by 7, 22, and 16, respectively. So in total, Φ decreases by at most 45. □

The following lemma is from [37].

Lemma 3.5.4. *The number of iterations of Algorithm 2 suffering from a hash collision is at most* $6n\epsilon$ *with probability at least* $1 - 2^{-\Theta(\epsilon n)}$.

Proof of Theorem 3.2.1: Let $R_{\text{total}} = \lceil \frac{n}{2r} \rceil + 368n\epsilon$. The total number of iterations is less than $2n$, so the total number of iterations with an error is at most $2n\epsilon$. By Lemma 3.5.4, with probability at least $1-2^{-\Theta(\epsilon n)}$, the number of iterations with a hash collision is at most $6n\epsilon$. Therefore, by Lemma 3.5.2, in the remaining $R_{\text{total}} - 8n\epsilon = \lceil \frac{n}{2r} \rceil + 360n\epsilon$ iterations, the potential Φ increases by at least one. The potential decreases only when there is an error or hash collision and it decreases by at most 45. So at the end of the simulation, we have

$$g - b - u \geq \Phi_{\text{Q}} \geq \Phi \geq R_{\text{total}} - 8n\epsilon - 45 \times 8n\epsilon \geq \frac{n}{2r} \ .$$

Hence the simulation is successful. Furthermore, note that the amount of communication in each iteration is independent of the iteration type and is always $2r + \Theta(1)$ symbols: in every iteration each party sends $\Theta(1)$ symbols to communicate the hash values and the lengths of the metadata and Pauli data in line 9 of Algorithm 2; each party sends another r symbols, either in line 14 of Algorithm 2, if *Itertype* $\neq$ SIM or in Algorithm 11 to communicate the teleportation measurement outcomes. So the total number of communicated symbols is

$$R_{\text{total}} \cdot (2r + \Theta(1)) = \left(\lceil \frac{n}{2r} \rceil + \Theta(n\epsilon) \right) (2r + \Theta(1)) = n(1 + \Theta(\sqrt{\epsilon})) \ , \tag{3.23}$$

as claimed. □

Chapter 4

Recycling-based coding scheme via large alphabet quantum channels

In this chapter, we focus on the plain quantum communication model with polynomial-size alphabet. Our simulation protocol in this model is obtained by adapting the framework introduced in Chapter 3, using additional tools to overcome the new challenges in this communication model.

4.1 Overview

4.1.1 Teleportation is inapplicable

Switching from the teleportation-based model to the plain quantum model, suppose we are given a protocol Π using noiseless quantum communication, and we are asked to provide a protocol Π' using noisy quantum channels under the strongly adversarial model described earlier. In the absence of free entanglement, how can we protect quantum data from leaking to the environment without incurring a non-negligible overhead? First, note that some form of protection is necessary, as discussed in Section 1.2. Second, teleportation would be too expensive to use, since it incurs an overhead of at least 3: we have to pay for the MES as well as the classical communication required.

Surprisingly, an old and relatively unknown idea called the Quantum Vernam Cipher (QVC) [50] turns out to be a perfect alternative method to protect quantum data with negligible overhead as the noise rate approaches 0.

4.1.2 Quantum Vernam Cipher (QVC)

Suppose Alice and Bob share two copies of the MES $|\phi^{0,0}\rangle$, each over two d-dimensional systems. For Alice to send a message to Bob, she applies a controlled-X operation with her

half of the first MES as control, and the message as the target. She applies a controlled-Z operation from her half of the second MES to the message. When Bob receives the message, he reverses the controlled operations using his halves of the MESs. The operations are similar for the opposite direction of communication. A detailed description is provided in Section 2.3.2.

QVC is designed so that given access to an authenticated classical channel from Alice to Bob, Bob can determine and correct any error in the transmission of the quantum message. This can simply be done by measuring Z^l type changes to one half of the two MES. They can also run QVC many times to send multiple messages and determine the errors in a large block using a method called "random hashing", and recycle the MESs if the error rate (as defined in our adversarial model) is low. This is a crucial property of QVC and leads to one of the earliest (quantum) key recycling results known. What makes QVC particularly suitable for our problem is that encoding and decoding are performed message-wise, while error detection can be done in large blocks, and entanglement can be recycled if no error is detected. It may thus be viewed as a natural quantum generalization to Haeupler's consistency checks.

As an aside, in Appendix E of Ref. [50], the relative merits of teleportation and QVC were compared, and it was determined that entanglement generation over an insecure noisy quantum channel followed by teleportation is more entanglement efficient than QVC with entanglement recycling in some test settings. However, this difference vanishes for low noise. Furthermore, the comparison assumes authenticated noiseless classical communication to be free. QVC requires an amount of classical communication for the consistency checks which vanishes with the noise parameter (but this cost was not a concern in that study). Furthermore, QVC was also proposed as an authentication scheme, but the requirement for interaction to authenticate and to recycle the key or entanglement was considered a disadvantage, compared to non-interactive schemes. (Those are only efficient for large block length, and cannot identify the error when one is detected. So, these authentication schemes are inapplicable). We thus provide renewed insight into QVC when interaction is natural (while it is considered expensive in many other settings).

4.1.3 Entanglement recycling and adaptations of QVC for the current problem

In the current scenario, we have neither free MESs nor an authenticated classical channel. Instead, Alice and Bob start the protocol by distributing the MESs they need, using a high rate quantum error correcting code over the low-noise channel. Then, they run the input protocol Π as is over the noisy channel, while frequently checking for errors by performing *quantum hashing* [8, 50], using the same noisy quantum channel instead of an authenticated classical channel. If they detect an inconsistency, assuming that the errors are most likely recent, they measure a small block of MESs in the recent past to determine the errors. They continue this process until they get matching quantum hash values indicating (with

constant probability) that they have located and identified all the errors and the remaining MESs can be recycled and reused to encrypt the messages. Frequent quantum hashing allows Alice and Bob to boost their confidence about recyclability of the earlier MESs and reuse MESs in a cyclic way. Note that for successful simulation it is crucial to ensure that the recycled MESs are indeed not corrupted and that Alice and Bob recycle the same sequence of MESs. One of our main contributions in this chapter is developing a framework for recycling entanglement in a communication efficient way. We show that entanglement generation of $\mathrm{O}\left(n\sqrt{\epsilon}\right)$ MESs, where n is the length of the input protocol Π and ϵ is the noise parameter, is sufficient to last through the whole simulation.

4.1.4 Framework

As in the case of the teleportation-based protocols, due to transmission errors and collisions, Alice and Bob do not necessarily always agree on their actions in the simulation. Therefore, in every iteration both parties need to obtain a global view of the history of the simulation so far to correctly decide their next actions. They achieve this goal by maintaining a similar data structure as in the teleportation-based case. The data structure now contains additional information to keep track of their measurements, which Alice and Bob use in the recycling process.

4.1.5 Additional out-of-sync problems

Due to transmission errors introduced by the adversary, Alice and Bob may get out of sync in QVC. In such a scenario, the QVC operations are performed by only one party and the quantum data intended to be sent to the other party leaks into the MES registers used to encrypt the messages. Furthermore, the parties may not agree on the subset of MESs they have already measured when they perform quantum hashing. As we will explain in Section 4.3.4, in the worst case, this can further lead to the leakage of the quantum data into all the MES registers involved in the quantum hashing procedure.

We show that, surprisingly, once again the quantum data can be recovered once Alice and Bob reconcile the differences in the data structure developed for the task. This is in spite of the fact that there is no reason to expect out-of-sync QVC to be sufficient to protect the quantum data from leaking to the environment when encoding and decoding operations are performed incorrectly and quantum data is sent via the noisy quantum channel.

4.2 Result

The following is our main result in the plain quantum model with polynomial-size communication alphabet for simulation of any n-round noiseless communication protocol over a fully adversarial channel of error-rate ϵ defined in Section 2.2.2.

Theorem 4.2.1. *Consider any n-round alternating communication protocol Π in the plain quantum model, communicating messages over a noiseless channel with an alphabet Σ of bit-size $\Theta(\log n)$. Algorithm 14 is a quantum coding scheme which given Π, simulates it with probability at least $1 - 2^{-\Theta(n\epsilon)}$, over any fully adversarial error channel with alphabet Σ and error rate ϵ. The simulation uses $n(1 + \Theta(\sqrt{\epsilon}))$ rounds of communication, and therefore achieves a communication rate of $1 - \Theta(\sqrt{\epsilon})$.*

4.3 Description of Protocol

4.3.1 General Description

Our simulation of noiseless protocols in the plain quantum model of communication proceeds using the same idea of running $\mathrm{O}_\epsilon(1)$ rounds of the input protocol as is, while checking if the adversary has corrupted the communication during the previous iterations and if necessary, actively rewinding the simulation to correct errors. The quantum messages are protected using QVC against corruptions by the adversary. In order to detect potential transmission errors, the MES pairs used as the key in QVC may be measured after each communication round. The measurement outcomes may be stored and later on compared to obtain the error syndrome. Therefore, using a data structure similar to the one introduced in the previous section, one can obtain a coding scheme for simulating any protocol in the plain quantum model. However, this approach is not efficient in using the entanglement. Recall that in the plain quantum model, the parties do not pre-share any entanglement, hence they need to establish the shared MESs through extra communication. Rather than measuring the MES pairs immediately after each round of communication, we use the quantum hashing procedure described in Section 4.3.2 to check whether any transmission error has occurred so far. Note that if Alice and Bob detect an error, they need to determine the error and eventually actively rewind the simulation to correct it and resume the simulation from there. However, similar to the teleportation-based protocol, due to transmission errors Alice and Bob may not always agree on how they proceed with the simulation in every iteration. Thus, in every iteration before taking further actions, each party needs to know the actions of the other party so far. More accurately, they first need to obtain a global view of their joint quantum state. Alice and Bob locally maintain a similar data structure as in the teleportation-based protocol containing metadata and Pauli data, which needs to be synchronized in the algorithm. They first need to ensure they have full knowledge of each other's metadata. Then similar to the teleportation-based case, in order to avoid out-of-sync scenarios in communication using QVC, it is crucial for them to synchronize the number of MESs they have used (see Section 4.3.3). We denote by $\ell^{\mathrm{A}}_{\mathrm{QVC}}$ and $\ell^{\mathrm{B}}_{\mathrm{QVC}}$, the number of blocks of MES pairs used by Alice and Bob, respectively. After ensuring $\ell^{\mathrm{A}}_{\mathrm{QVC}} = \ell^{\mathrm{B}}_{\mathrm{QVC}}$ (to the best of their knowledge), they compare their quantum hash values to check for errors. Note that quantum hashing does not indicate in which round of communication the error has occurred. If the hash values do not match, they measure the

last block of MES pairs which has not gained as much trust as the older blocks through quantum hashing. This effectively collapses the adversary's action on this block to Pauli errors. The effective errors can be determined jointly from the measurement outcomes, which are recorded by Alice and Bob as part of their local Pauli data. Otherwise, if the hashes do match, then Alice and Bob synchronize their Pauli data. Note that similar to the teleportation-based protocol, the Pauli corrections performed by Alice and Bob are also recorded as part of the Pauli data. Together with the metadata, this gives Alice and Bob all the information they need to compute their estimate of the current joint state. Hence, they can determine how to proceed with the simulation next.

Recycling entanglement and recycling data. An important complication arising in simulation of protocols in the plain quantum model is that in order to achieve the simulation rate of $1 - \Theta(\sqrt{\epsilon})$, we cannot afford to access a new MES pair in every round of communication using QVC. This is where we use a crucial property of QVC, namely the key recycling property. In communication using QVC if no error occurs on the message then the pair of MESs used will remain intact, hence they can be recycled and used in later rounds. Otherwise, at some point Alice and Bob need to measure the MES pair to get the error syndrome, in which case the pair cannot be recycled. By performing quantum hashing regularly and carefully keeping track of the measured MES blocks, Alice and Bob can recycle MES pairs as needed and run the simulation by establishing a smaller number of MESs at the beginning of the protocol. The blocks of MES pairs used in QVC are implicitly indexed from 1 to L_{QVC}, where L_{QVC} denotes the total number of MES blocks reserved for communication using QVC.

In order to correctly simulate the input protocol we need to ensure that the recycling is successful in every iteration, namely that the same MES blocks are recycled by the two parties in every iteration and that they are indeed not corrupted when being recycled. Note that if the two parties reuse an MES pair which has been corrupted due to an earlier transmission error in QVC, then even if Alice and Bob detect the error and measure the MES pair, they have no way of knowing whether the error has occurred the last time the MES pair were used or in an earlier round. Moreover, if a block of MES pairs has been locally measured by only one party, say Alice, then the other party, Bob, needs to measure the block and avoid this block when encrypting future messages using QVC.

We modify the metadata so that it contains additional information to keep track of each party's measurements. Alice maintains a string $RA \in \{\mathsf{S}, \mathsf{M}\}^*$, where the M symbol corresponds to a measured MES block and S is used for an MES block still in superposition. In every iteration, Alice concatenates RA with an S symbol corresponding to her next fresh/recycled MES block and records the index of the MES block in a string $IndexA \in [L_{\text{QVC}}]^*$. Both RA and $IndexA$ are of length i after i iterations. If she measures an MES block in the current iteration, she changes the corresponding S symbol in RA to M. Similarly, Bob maintains the strings RB and $IndexB$. The strings $IndexA$ and $IndexB$ serve as queues of MES blocks to be used/reused by Alice and Bob, respectively. Note that MES

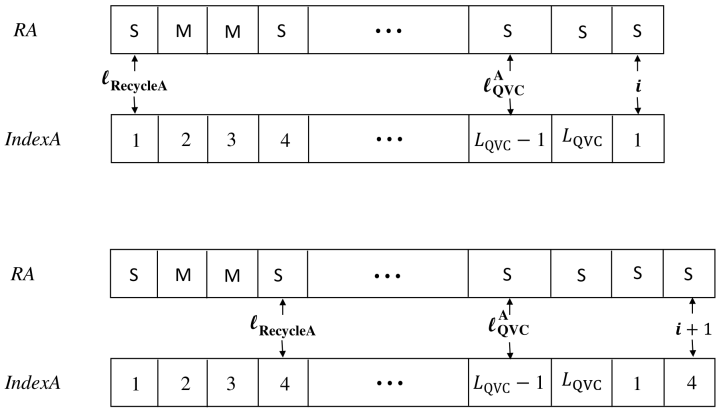

Figure 4.1: The strings RA and $IndexA$ in two consecutive iterations $L_{\text{QVC}}+1$ and $L_{\text{QVC}}+2$. The pointer ℓ_{RecycleA} moves to the next S symbol in RA to find the next MES block to recycle.

blocks in these queues are not necessarily used immediately or even in the same iteration by the two parties. In order to achieve successful recycling, Alice and Bob need to ensure that the strings RA and RB are the same (at least in a long enough prefix). Using her full-length estimate $\widetilde{MB}$ of $FullMB$, Alice computes an estimate $\widetilde{RB}$ of RB. Note that if $\widetilde{MB} = FullMB$ then $\widetilde{RB} = RB$. Similarly, Bob computes his estimate $\widetilde{RA}$ of RA from his full-length estimate $\widetilde{MA}$ of $FullMA$. After synchronizing their metadata and ensuring that they have used the same number of MES blocks, they synchronize their recycling data.

After L_{QVC} iterations, Alice and Bob start recycling MESs in a circular way. In order to achieve this, Alice uses a recycling pointer ℓ_{RecycleA}. In every iteration, this pointer moves forward on the strings RA and $IndexA$ until it reaches the next S symbol in RA. The corresponding MES block is recycled and its index is recorded at the end of $IndexA$. Similarly, Bob uses a pointer ℓ_{RecycleB} together with RB and $IndexB$ to recycle entanglement. Frequent hashing of the metadata allows them to be highly confident that their recycling data agree in a sufficiently long prefix. Furthermore, quantum hashing ensures that with high probability all the recycled MES blocks are indeed reusable. Figure 4.1 depicts the status of the strings RA and $IndexA$ in two consecutive iterations and how the pointers move on these strings in the recycling process described above.

Note that the synchronization of the recycling data is slightly different from the synchronization of the metadata and the Pauli data. For the latter, Alice and Bob just need to know each other's local data, i.e., Alice needs to know $FullMB$ and $FullPB$ and Bob needs to know $FullMA$ and $FullPA$ and the corresponding data need not be the same. Whereas

for the recycling data, Alice and Bob need to learn each other's data and match them, i.e., they need to ensure $RA = RB$. Moreover, unlike *FullMA*, *FullMB* and *FullPA*, *FullPB*, earlier parts of the strings RA and RB get modified during the simulation as Alice and Bob perform measurements to extract error syndromes.

Entanglement distribution. At the outset of the simulation, Alice and Bob use Algorithm 12 below to share $\Theta(n\sqrt{\epsilon})$ copies of the MES $|\phi^{0,0}\rangle$, defined in Definition 2.1.2. To establish the shared MESs, one party, say Alice, creates the states locally and sends half of each MES to the other party using an appropriate quantum error correcting code of distance $4n\epsilon$ and constant rate. Note that such a code is guaranteed to exist by the quantum Gilbert-Varshamov bound [29].

Algorithm 12: Robust Entanglement Distribution

```
C ← Error Correcting Code with rate 1 − Θ(H(ϵ)) and distance 4nϵ;
if Alice then
    Prepare Θ(n√ϵ) MESs in registers A, B′ each holding half of every MES;
    Transmit C(B′) to Bob;
else if Bob then
    Receive C′(B′);
    Decode C′(B′) into register B;
return Robust Entanglement Distribution;
```

We remark that the initial entanglement distribution is the only part of the simulation protocol in the plain quantum model which we do not know how to perform efficiently.

The shared MESs are used as follows:

- $\Theta(n\sqrt{\epsilon})$ MESs are used in pairs to serve as the key for encryption of messages using QVC. They are divided into $L_{\text{QVC}} = \Theta(n\epsilon)$ blocks of $2r$ MES pairs, where $r = \Theta(\frac{1}{\sqrt{\epsilon}})$. In each block the MES pairs are implicitly numbered from 1 to $2r$. The odd-numbered pairs are used in QVC to send messages from Alice to Bob and the even-numbered pairs are used to send messages from Bob to Alice,
- $\Theta(n\sqrt{\epsilon})$ MESs are reserved to be used in quantum hashing, and
- the remaining $\Theta(n\sqrt{\epsilon})$ MESs are measured in the computational basis by both parties to obtain a common random string to be used as the seed for classical and quantum hashing.

We show that with the limited error budget of the adversary, the $\Theta(n\sqrt{\epsilon})$ MESs established at the beginning of simulation are sufficient to successfully simulate the input protocol. This allows us to achieve a simulation rate of $1 - \Theta(\sqrt{\epsilon})$. Figure 4.2 shows the MES blocks in different stages of the simulation.

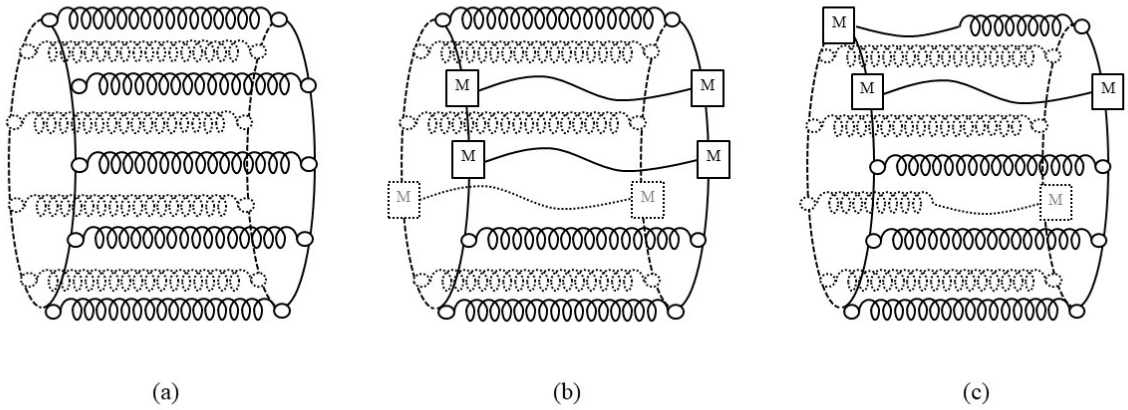

Figure 4.2: These figures represent the blocks of MES pairs at different stages of the protocol. To simplify the figure, we represent each block by a single MES. Note that these are used in a circular pattern, corresponding to recycling some of the previously used blocks of MES pairs. Those depicted as circles are assumed to be good and usable for QVC, those depicted by squares have been measured already in order to extract the error syndrome. Figure (a) represents the MES blocks at the beginning of the protocol, when none have been measured. Figure (b) represents them when Alice and Bob agree on which ones have been measured and have used the same amount of them for QVC, which is the desired state. Figure (c) represents a situation when Alice and Bob have gotten out-of-sync, e.g., Alice has measured some blocks that Bob has not (and maybe used QVC more often than Bob). They then work to get back in sync before resuming the simulation.

4.3.2 Quantum Hashing

By performing local measurements on the MES pairs used as the keys for encrypting the messages in QVC and comparing the measurement outcomes on both sides, one can extract the error syndrome corresponding to the corruptions introduced over the noisy communication channel. As explained in the previous subsection, although this allows the two parties to detect errors immediately, it is not efficient for our application. In Subsection 2.3.2, we introduced an error detection procedure which allows the parties to check for corruptions when QVC is used over several rounds of communication to send multiple messages at the cost of losing only one MES which is measured at the end. However, this error detection procedure is not directly useful in our application since the adversary

can always choose the corruptions in a way that makes it impossible for Alice and Bob to detect errors; see subsection 2.3.2. Instead, Alice and Bob use the *quantum hashing* procedure described below to check whether there is an undetected error. To avoid the adversary from hiding her corruptions from the detection procedure above, Alice and Bob choose a random subset of the MESs and try to detect errors in this subset rather than all MESs used in QVC. More precisely, quantum hashing involves the following steps in our algorithm. At the beginning of the i-th iteration, Alice and Bob pick a fresh MES serving as the control system used in the error detection procedure. Recall that Alice and Bob locally maintain the strings *IndexA* and *IndexB*, respectively, corresponding to their recycled MES blocks. Alice uses the shared randomness established at the outset of the protocol to choose a random subset of the MES registers contained in the blocks specified by $IndexA\left[\ell_{\text{RecycleA}}+1:\ell^{\text{A}}_{\text{QVC}}\right]$. Using her recycling data RA, she locally determines the MESs in this random subset which she has not measured already. She performs her operations in the detection procedure described in Subsection 2.3.2 only on these MESs. Bob does the same locally, based on *IndexB* and RB. Alice and Bob store their measurement outcomes in QHA and QHB, respectively, and exchange the values. We prove that except with exponentially small probability, recycling is successful throughout the execution of the algorithm. As we will see, this implies that in every iteration, $\ell_{\text{RecycleA}}=\ell_{\text{RecycleB}}$ and $IndexA = IndexB$. However, RA and RB do not necessarily match in every iteration. Therefore, in some iterations, Alice and Bob may perform quantum hashing on different subsets of the MESs. Moreover, if the two parties are not synchronized in the number of MES blocks they have used, they might perform quantum hashing on an MES that is only used by one party but not the other. We discuss these out-of-sync quantum hashing scenarios in Subsection 4.3.4. We remark that Alice and Bob do not need to perform quantum hashing on any of the MESs within the blocks specified by $IndexA\left[1:\ell_{\text{RecycleA}}\right]$. In fact, if according to $RA\left[1:\ell_{\text{RecycleA}}\right]$, an MES block is measured then it does not appear in $IndexA\left[\ell_{\text{RecycleA}}+1:\ell^{\text{A}}_{\text{QVC}}\right]$. Otherwise, it appears exactly once. The same statement holds for RB and *IndexB*.

Alice and Bob compare their quantum hash values only when they believe they have full knowledge of each other's metadata, have used the same number of MES blocks and agree on their recycling data. If they get the same hash values, they assume no error has occurred. Otherwise, they believe they have detected an error. Note that similar to all other types of data that are communicated over the noisy channel, the quantum hash values may get corrupted by the adversary. We say a *quantum hash collision* has occurred in an iteration only when recycling has been successful so far, Alice and Bob are indeed synchronized in their metadata, the number of MES blocks they have used and their recycling data and $QHA = QHB$ despite the fact that there are non-measured MES blocks which are not in the $\left|\phi^{0,0}\right\rangle^{\otimes 4r}$ state.

The following lemma shows that in every iteration of the algorithm, assuming successful recycling up to that point, the probability of a quantum hash collision is at most $1/2$.

Lemma 4.3.1. *Let $m\in\mathbb{N}$ and $k=k_1\dots k_m\in\{0,1,\dots,d-1\}^m$. Suppose that Alice*

and Bob share the states $|\phi^{0,0}\rangle_{A_0B_0}, \left|\phi^{0,k_1}\right\rangle_{A_1B_1}, \ldots, \left|\phi^{0,k_m}\right\rangle_{A_mB_m}$ *and a random variable* S *distributed over* $\{0,1\}^m$*, independent of* k*, interpreted as a subset of* $[m]$*. Alice and Bob apply* $c\text{-}\mathrm{X}_{A_0A_i}$ *and* $c\text{-}\mathrm{X}_{B_0B_i}$*, for all* $i \in S$*. Then they apply the quantum Fourier transform operator* F *and its inverse* $\mathrm{F}^\dagger$ *on* A_0 *and* B_0*, respectively. They measure the registers in the computational basis with outcomes* QHA *and* QHB*, respectively. Then for* $k = 0^m$*, independent of the random variable* S*, we have*

$$\Pr[QHA = QHB] = 1 \ .$$

Moreover, for uniformly random S*, for all* $k \neq 0^m$*, we have*

$$\Pr[QHA = QHB] \leq \frac{1}{2} \ .$$

Proof. Let $S = S_1S_2 \ldots S_m$. By Lemma 2.3.3, the state in register A_0B_0 before the measurements is

$$\begin{aligned}\left(\mathrm{F} \otimes \mathrm{F}^\dagger\right)\left|\phi^{0,-\sum_{i=1}^m S_ik_i}\right\rangle_{A_0B_0} &= \left(\mathrm{F}\mathrm{Z}^{-\sum_{i=1}^m S_ik_i} \otimes \mathrm{F}^\dagger\right)\left|\phi^{0,0}\right\rangle = \left(\mathrm{F}\mathrm{Z}^{-\sum_{i=1}^m S_ik_i}\mathrm{F}^\dagger \otimes \mathbb{1}\right)\left|\phi^{0,0}\right\rangle \\ &= \left(\mathrm{X}^{-\sum_{i=1}^m S_ik_i} \otimes \mathbb{1}\right)\left|\phi^{0,0}\right\rangle = \left|\phi^{-\sum_{i=1}^m S_ik_i,0}\right\rangle \ ,\end{aligned}$$

where the first and the last equality follow from Definition 2.1.2; the second equality holds by Proposition 2.1.4 and the fact that $\mathrm{F} = \mathrm{F}^T$. The third equality follows from Proposition 2.1.1. Hence

$$\Pr[QHA = QHB] = \Pr\left[\sum S_ik_i = 0 \mod d\right] \ .$$

If $k = 0^m$ then the above probability equals 1. Suppose that k is non-zero in a non-empty subset J of coordinates in $[m]$. Consider the set Z of all $s \in \{0,1\}^m$ such that $\sum s_ik_i = 0 \mod d$. Note that the minimum Hamming distance of elements of Z restricted to J is at least 2, since otherwise there exists $j \in J$ such that d divides k_j, contradicting $k_j \in [d-1]$. Fix $j \in J$ and let $e_j \in \{0,1\}^m$ be the string which is 1 in the j-th coordinate and zero everywhere else. For every $s \in Z$, the string $s + e_j$ is not in Z. Therefore, $|Z| \leq 2^{m-1}$ and for S uniformly distributed over $\{0,1\}^m$ we have

$$\Pr\left[\sum S_ik_i = 0 \mod d\right] \leq \frac{1}{2} \ .$$

Note that the above bound is tight when $|J| = 1$. Finally, by Lemma 2.3.3 the state in the registers $A_1B_1, \ldots, A_mB_m$ remains unchanged. □

In order to reduce the collision probability to a smaller constant, quantum hashing may be repeated for a constant number of times in every iteration with fresh control MESs and independent random subsets S.

Classical seeds needed for quantum hashing. Alice and Bob perform quantum hashing and communicate the hash values in every iteration but they only compare their hash values in a subset of iterations. We choose to do so in order to avoid the two parties from getting out-of-sync on which MES register to use in quantum hashing. As the hashing procedure only consumes a constant number of MESs in each iteration, the total number of MESs used in quantum hashing in the entire simulation is $\Theta(R_{\text{total}}) = \Theta(n\sqrt{\epsilon})$, and they constitute a constant fraction of the MESs distributed at the outset of the protocol; see Subsection 4.3.1. On the other hand, generating independent $\Theta(rt)$-bit seeds, with $r \in \Theta(1/\sqrt{\epsilon})$ and $t \in \Theta(n\epsilon)$, for each of the R_{total} iterations would require $\Theta(n^2\epsilon)$ bits of shared randomness. The shared randomness is obtained by measuring a fraction of the MESs established at the beginning of the algorithm. Even in the large-alphabet case, sharing $\Theta(n^2\epsilon)$ bits of randomness would require too much communication.

To circumvent this obstacle Alice and Bob start with a smaller number of i.i.d. random bits and extend them to a much longer pseudo-random string. In more detail, they measure $\Theta(n\sqrt{\epsilon})$ MESs in the computational basis and record the binary representation of the outcomes in R'. Then they each use the deterministic algorithm of Lemma 2.4.8 with $\delta = 2^{-\Theta(n\sqrt{\epsilon})}$, to obtain a shared δ-biased string R' of length $\Theta(rtR_{\text{total}}) = \Theta(n^2\epsilon)$. The following lemma bounds the collision probability when instead of a uniformly random seed, a δ-biased seed is used in quantum hashing. Note that in our application of Lemma 4.3.2, we have $m = \mathrm{O}(rt) = \mathrm{O}(n\sqrt{\epsilon})$.

Lemma 4.3.2. *Suppose that the random variable S in Lemma 4.3.1 is δ-biased. Then for all $k \neq 0^m$, we have*

$$\Pr[QHA = QHB] \leq \frac{1}{2} + 2^{m/2}\delta \ .$$

Proof. Let U denote the uniform distribution on $\{0,1\}^m$ and Z be the subset of all $s \in \{0,1\}^m$ such that $\sum s_i k_i = 0 \mod d$. By Propositions 2.4.5 and 2.4.4, we have

$$|U(Z) - S(Z)| \leq \frac{1}{2}\|U - S\|_1 \leq \frac{1}{2} \times 2^{m/2}\|U - S\|_2 \leq 2^{m/2}\delta \ .$$

Therefore, by Lemma 4.3.1, we have

$$\Pr[QHA = QHB] = \Pr\Big[\sum S_i k_i = 0 \mod d\Big] = S(Z) \leq \frac{1}{2} + 2^{m/2}\delta \ .$$

□

4.3.3 Out-of-Sync Quantum Vernam Cipher

Consider the scenario where Alice, based on her view of the simulation so far, implements a $+1$ block, while Bob believes their classical data are not consistent and therefore implements a C iteration. Alice simulates a block of the input protocol Π while using the next block

of MES pairs in the queue *IndexA* to encrypt her messages using QVC. At the same time, Bob tries to reconcile the inconsistency through classical communication and does not send his messages using QVC. In this scenario, they also interpret the messages they receive incorrectly. Alice believes Bob is also encrypting his messages using QVC and she applies QVC decoding operations and her Pauli corrections on Bob's messages. Moreover, she potentially applies unitary operations of the input protocol on her local registers. Meanwhile, Bob treats Alice's messages as classical information about the data he believes they need to synchronize. More importantly, since he does not perform the QVC operations (decoding operations in odd rounds and encoding operations in even rounds) on his side, in each round the corresponding MES pair becomes entangled with the message register. So crucial information for continuing the simulation spreads to multiple registers. Moreover, this scenario could continue for several iterations. Nonetheless, we provide a simple way to redirect the quantum information back to the ABC registers, while effectively reducing this type of error to corruptions introduced in the joint state due to transmission errors by the adversary. Once reduced to such errors, Alice and Bob can actively rewind the incorrect part of the simulation and resume from there.

As explained earlier, the first step for Alice and Bob is to ensure they have full knowledge of each other's metadata. Once they both achieve this goal, they discover the discrepancy in the number of MES blocks they have used. Suppose Bob has used fewer blocks of MES pairs than Alice, i.e., $\ell^{\mathrm{A}}_{\mathrm{QVC}} > \ell^{\mathrm{B}}_{\mathrm{QVC}}$ and he discovers this at the beginning of the i-th iteration. Let $E_1 E_2 \dots E_{4r}$ be the registers with Bob containing halves of the $4r$ MESs in the first MES block that Alice has used, say in the i'-th iteration, but Bob has not so far. Note that in iteration i', Alice has used the MES pairs corresponding to $E_1E_2, E_5E_6, \dots, E_{4r-3}E_{4r-2}$ on her side to encrypt quantum information using QVC and she has performed QVC decoding operations on Bob's messages and her marginal of MES pairs corresponding to $E_3E_4, E_7E_8, \dots, E_{4r-1}E_{4r}$. In the i-th iteration, Alice and Bob both send dummy messages to each other. Let $C_1, C_2, \dots, C_r$ denote the r message registers sent from Alice to Bob after communication over the noisy channel. For every $j \in [r]$, upon receiving C_j, Bob applies QVC decoding operations on C_j and $E_{4j-3}E_{4j-2}$, and then applies QVC encoding operations on C_j and $E_{4j-1}E_{4j}$, i.e., he applies

$$(\text{c-Z})_{E_{4j}C_j} (\text{c-X})_{E_{4j-1}C_j} \left(\text{c-X}^{-1}\right)_{E_{4j-3}C_j} \left(\text{c-Z}^{-1}\right)_{E_{4j-2}C_j},$$

and then he discards the message register C_j. The effect of these operations is the same as if Alice and Bob had both used the MES block in sync, i.e., in the i'-th iteration, *except the following also happened independently of channel error*:

1. Alice's messages in the i'-th iteration were replaced by $C_1, \dots, C_r$ and Bob applied his QVC decoding operations on these dummy messages rather than the messages Alice intended to communicate,

2. the unitary operations used by Bob on the registers BC were all identity, and

3. Bob's messages were replaced by his messages of the i'-th iteration and Alice's QVC decoding operations were applied on these (classical) messages.

The above procedure redirects the quantum information leaked to the MES registers back to the ABC registers, while introducing errors which act exactly the same as transmission errors introduced by the adversary. As in the case of corruptions by the adversary, once Alice and Bob measure the MES block, the error collapses to a Pauli error which can be determined by comparing the measurement outcomes by Alice and Bob. We choose to perform the measurements at the end of the i-th iteration, rather than leaving the algorithm to detect the error through quantum hashing (as in the case of transmission errors).

4.3.4 Out-of-Sync Quantum Hashing

Consider the scenario in which Alice and Bob have used the same number of MES blocks for communication using QVC but have measured different subsets of MES blocks. Suppose that when they perform quantum hashing, the random subset of MESs they choose contains an MES in registers A_1B_1, which has been measured by only one party, say Alice. Let $V_{\mathrm{A}}V_{\mathrm{B}}$ be the registers used as the control registers by Alice and Bob in quantum hashing. Alice and Bob compare their quantum hash values only if they believe they have measured the same subset of MES blocks. Therefore, if they compare their hash values in this iteration, it is due to a transmission error or a metadata hash collision. Note that in this scenario Bob applies a controlled-X operation on the partially measured MES, while Alice who has measured her marginal does not. Since A_1 is already measured by Alice, after Bob's controlled-X operation the registers A_1B_1 do not get entangled with $V_{\mathrm{A}}V_{\mathrm{B}}$. However, the state in the $V_{\mathrm{A}}V_{\mathrm{B}}$ registers will be mapped to $|\phi^{0,a}\rangle$, for a random $a \in \{0, 1, \ldots, d-1\}$ corresponding to Alice's measurement outcome. This (quite probably) results in Alice and Bob taking incorrect actions from which they can recover once they realize the inconsistency in their classical data. The algorithm is designed to ensure that the register B_1 is measured by Bob and A_1B_1 is not reused by the two parties in future iterations. Moreover, Bob's controlled-X operation does not change the outcome of his measurement on B_1. This ensures that Alice and Bob can correctly learn any potential error on the message register in the corresponding communication round once they learn each other's Pauli data.

A subtler scenario occurs when Alice and Bob perform quantum hashing when they have used different numbers of MES blocks. Suppose that $\ell^{\mathrm{A}}_{\mathrm{QVC}} > \ell^{\mathrm{B}}_{\mathrm{QVC}}$, i.e., Alice has used more MES blocks and the random subset of MESs they choose for quantum hashing contains an MES which has been only used on Alice's side. As explained in the previous section, when QVC operations are performed only on one side, the MES pair used as the key becomes entangled with the message register and the quantum information in the message register leaks to these half-used MES pairs. In the current scenario, once quantum hashing is done the information leaks into the additional MES in registers $V_{\mathrm{A}}V_{\mathrm{B}}$. Even worse, since the same MES register is used as the control system when applying the controlled-X operations

on all the MESs in the random subset, the information may leak even further into those registers as well. Surprisingly, the simple solution we provided to recover from out-of-sync QVC resolves this issue as well. The first measure we need to take is to ensure quantum hashing is performed in a sequential way on the MESs in the random subset, starting from the MES used earliest to the latest one. This ensures that the states in the MES registers which have been used both by Alice and Bob do not get disturbed. However, the remaining MESs in the random subset become entangled with $V_{\mathrm{A}}V_{\mathrm{B}}$ and potentially each other. We need to ensure that these MES registers are not reused in future iterations. Once the two parties synchronize their metadata and realize that they are out of sync on the number of MESs they have used, Bob completes QVC on his side as described in the previous section and immediately measures his marginal of the MES block. This ensures that he will never reuse this block in future iterations. The algorithm is designed so that by the time Alice needs to decide whether to recycle this MES block or not, she will have measured her marginal of the MES registers. We prove that except with probability $2^{-\Theta(n\epsilon)}$ recycling is successful in all iterations and such a block of MES registers is never recycled.

Despite the fact that quantum hashing is performed before Bob completes the QVC operations on his side, this procedure has the same effect as if Bob had completed QVC *before* the quantum hashing was performed. To understand this phenomenon, consider the following simpler scenario. Suppose that Alice and Bob share 3 copies of the MES $|\phi^{0,0}\rangle$ in registers A_1B_1, A_2B_2 and $V_{\mathrm{A}}V_{\mathrm{B}}$. Alice uses the MES pair in registers A_1B_1 and A_2B_2 as the key to encrypt a message in register C using QVC and sends the message register to Bob. Suppose that the adversary applies the Pauli error $\mathrm{X}^a\mathrm{Z}^b$ on the message register for some $a, b \in \{0, 1, \ldots, d-1\}$. Now suppose that before Bob applies his QVC decoding operations, Alice applies c-X on $V_{\mathrm{A}}A_1$ with V_{A} being the control system. Then their joint state is

$$\left(\text{c-X}^{-1}\right)_{B_1C}\left(\text{c-Z}^{-1}\right)_{B_2C}(\text{c-X})_{V_{\mathrm{A}}A_1}\left(\mathrm{X}^a\mathrm{Z}^b\right)_C(\text{c-Z})_{A_2C}(\text{c-X})_{A_1C} \left|\phi^{0,0}\right\rangle_{V_{\mathrm{A}}V_{\mathrm{B}}}\left|\phi^{0,0}\right\rangle_{A_1B_1}\left|\phi^{0,0}\right\rangle_{A_2B_2}|\psi\rangle_C \quad .$$

Note that $(\text{c-X})_{V_{\mathrm{A}}A_1}$ commutes with Bob's QVC decoding operation $\left(\text{c-X}^{-1}\right)_{B_1C}\left(\text{c-Z}^{-1}\right)_{B_2C}$. Therefore, by Eq. (2.5) their joint state is given by

$$(\text{c-X})_{V_{\mathrm{A}}A_1}\left(\mathrm{X}^a\mathrm{Z}^b\right)_C\left|\phi^{0,0}\right\rangle_{V_{\mathrm{A}}V_{\mathrm{B}}}\left|\phi^{0,b}\right\rangle_{A_1B_1}\left|\phi^{0,-a}\right\rangle_{A_2B_2}|\psi\rangle_C \quad .$$

Note that $V_{\mathrm{A}}V_{\mathrm{B}}$ and A_1B_1 are entangled as a result of the controlled-X operation. Nevertheless, Alice and Bob still extract the correct error syndrome when they measure A_1, A_2 and B_1, B_2, respectively, and compare their measurement outcomes. This is due to the fact that the error on the message register is reflected in the MES pair as phase errors and the phase error in each MES can still be detected correctly by local measurements in the Fourier basis even after the controlled-X operation is applied. In the out-of-sync quantum hashing scenario described above a similar effect occurs. Finally, note that when Alice and Bob do not agree on the number of MES blocks they have used, they do not compare their quantum hash values unless a transmission error or a metadata hash collision occurs.

4.3.5 First representation of the joint quantum state

As in Section 3.3, we start by introducing a first representation of the joint state, denoted $JS1$, which in turn is simplified into a more informative representation. This latter representation, denoted $JS2$, is the representation which Alice and Bob need to compute correctly in order to make progress in simulation of the input protocol Π and decide their next action in Π'. Recall that due to the recycling of MESs, each block of MES pairs may be used multiple times to encrypt messages using QVC. The representations $JS1$ and $JS2$ defined below are valid only if the recycling has been successful so far in Π', namely that Alice and Bob have recycled the same block of MES registers in every iteration and that these registers were indeed in the $|\phi^{0,0}\rangle$ state when recycled. We prove that except with probability $2^{-\Theta(n\epsilon)}$ the recycling is successful throughout the algorithm.

Recall that in the adversarial noise model, the adversary Eve can introduce arbitrary errors on the quantum communication register C' that passes through her hand subject to the constraints given by Eq. (2.2) and Eq. (2.3). Furthermore, as explained in Section 4.3.3, the algorithm is designed so that once the two parties agree on the number of blocks of MES pairs they have used, the error in the joint state due to out-of-sync QVC in any iteration is translated to a transmission error on the message registers, as if the MES blocks were used in sync and transmission errors were introduced by the adversary. We emphasize that in both cases, the error on the message register is a mixture of linear combinations of Pauli errors and once Alice and Bob measure a block of MES pairs to extract (part of) the syndrome, the error on the corresponding message register collapses to a Pauli error. Then the joint state can be written in terms of a new mixture of linear combinations of Pauli errors conditioned on the measurement outcomes, which are recorded in the Pauli data by the two parties. To simplify the joint state representation and the analysis of the algorithm, without loss of generality, we focus on a fixed but arbitrary error syndrome W in any such linear combination of Pauli errors arising in the simulation protocol Π'. We prove the correctness of the algorithm for any such error syndrome which by linearity implies the correctness of the algorithm against any adversary defined in Section 2.2.2. Let $E \in \mathcal{P}_{d,n'}$ be a Pauli error with $\mathrm{wt}\,(E) \leq \epsilon n'$. In the remainder of this chapter, we assume E is the error introduced by the adversary into the n' communicated qudits in Π'.

We first define the representations $JS1$ and $JS2$ after i iterations of the algorithm in the case when the two parties have used the same number of blocks of MES pairs, i.e., $\ell^{\mathrm{A}}_{\mathrm{QVC}} = \ell^{\mathrm{B}}_{\mathrm{QVC}}$. In Subsection 4.3.7, we explain how these representations are modified when Alice and Bob are out of sync in the number of MES blocks they have used.

We sketch how to obtain $JS1$ from *FullMA*, *FullMB*, *FullPA* and *FullPB*, when the error syndrome is given by $W \in (\Sigma^2)^*$ defined below in terms of E, conditioned on some view of the classical data. Recall that when $\ell^{\mathrm{A}}_{\mathrm{QVC}} = \ell^{\mathrm{B}}_{\mathrm{QVC}}$, there is a one-to-one correspondence between the state of each MES register and the Pauli error on the message register in the corresponding communication round. Therefore, in order to simplify the representations, without introducing any ambiguity, we omit the MES registers from the representations $JS1$ and $JS2$. The first representation $JS1$ of the joint state after i iterations (when

$\ell^{\mathrm{A}}_{\mathrm{QVC}} = \ell^{\mathrm{B}}_{\mathrm{QVC}}$) is given by

$$JS1 = [*\ell^{\mathrm{A}}_{\mathrm{QVC}}] \cdots [*2][*1] \, |\psi_{\mathrm{init}}\rangle^{ABCER}, \tag{4.1}$$

where $|\psi_{\mathrm{init}}\rangle^{ABCER}$ is the initial state of the original input protocol Π and the content of each bracket is described below. The j-th bracket corresponds to the j-th block of MES pairs which have been used by both Alice and Bob and contains from right to left r iterations of the following:

Alice's unitary operation -
Pauli error on Alice's message -
Bob's Pauli correction - Bob's unitary operation -
Pauli error on Bob's message -
Alice's Pauli correction.

Similar to the teleporation-based protocol, we allow for an additional unitary operation by Alice on the far left when she implements a block of type -1. Using the same rules described in Section 3.3.7, in each bracket, the block of unitary operations of the input protocol Π applied by Alice (if any) and her block type (± 1 or 0) can be computed from *FullMA*. Moreover, her Pauli corrections are recorded in *FullPA* and the block of *FullPA* containing these Pauli corrections can be located using *FullMA*. Each block of *FullPA* may correspond to two different types of iterations: when Alice measures a block of MES pairs to extract the error syndrome she concatenates *FullPA* with a new block containing her measurement outcomes (with no Pauli corrections), whereas in iterations in which she communicates using QVC, she may apply Pauli corrections in between and records the Pauli corrections in *FullPA*. Therefore, *FullMA* may be used to distinguish these two different types of blocks and locate the corresponding Pauli corrections in *FullPA*. Similarly, in each bracket, the block of unitary operations of Π applied by Bob, his block type and his Pauli corrections are obtained from *FullMB* and *FullPB*. Finally, in *JS*1, the Pauli errors on the messages in each bracket are specified in terms of the error syndrome $W = W_1 W_2 \ldots W_{\ell^{\mathrm{A}}_{\mathrm{QVC}}} \in (\Sigma^2)^{2r \times \ell^{\mathrm{A}}_{\mathrm{QVC}}}$ defined below. The communication in each iteration of the algorithm has two parts. In the first part, the parties use a constant number of rounds to communicate the pointers and hash values. Any transmission error introduced by the adversary on these messages only affects the actions of the two parties in the current and future iterations, which will be recorded in the metadata and Pauli data and reflected in the joint state representation. The second part involves $2r$ rounds of communication, in which either classical information is communicated (e.g., to reconcile inconsistencies in the data structures) or QVC is used to communicate quantum information (on one side or both). Transmission errors on these messages can directly modify the joint state and need to be separately taken into account in the joint state representation. Let $W' = W'_1 W'_2 \ldots W'_{R_{\mathrm{total}}}$ denote the error syndrome corresponding to the restriction of E to these messages over the R_{total} iterations of the algorithm, where each W'_j is a string in $(\Sigma^2)^{2r}$ representing a Pauli error on $2r$ qudits. For every $j \in \left[\ell^{\mathrm{A}}_{\mathrm{QVC}}\right]$, if the j-th block of MES pairs has been used in

sync on both sides, say in iteration j', we let $W_j = W'_{j'}$. Otherwise, the j-th block of MES pairs has been used out of sync and we define W_j to be the error syndrome arising on the message registers in the corresponding communication rounds due to the remedial actions the parties take to recover from out-of-sync QVC; see Section 4.3.3 for more details. Each $W_j \in (\Sigma^2)^{2r}$ specifies the $2r$ Pauli errors in the j-th bracket from the right.

Note that in order to compute the representation $JS1$, one needs to know *FullMA*, *FullMB*, *FullPA*, *FullPB* and W. This information is not necessarily available to Alice and Bob at any point during the simulation. In fact, we use the representation in order to analyze the progress in the simulation. Alice and Bob compute their best guess for $JS1$ based on their estimates of each other's classical data. They only compute their estimates of $JS1$ when they believe that they have full knowledge of each other's metadata and Pauli data, fully agree on the recycling data, have used the same number of MES blocks and have measured all blocks of MES pairs which were corrupted in communication using QVC or due to out-of-sync QVC.

Alice's estimate $JS1^{\mathrm{A}}$ of $JS1$ is of the same form as in Eq. (4.1), except she uses her best guess of *FullMB* and *FullPB* in the above procedure. Moreover, the string W in $JS1$ is replaced by $W^{\mathrm{A}} = W_1^{\mathrm{A}} \dots W_{\ell_{\mathrm{QVC}}^{\mathrm{A}}}^{\mathrm{A}} \in (\Sigma^2)^{2r \times \ell_{\mathrm{QVC}}^{\mathrm{A}}}$ computed by Alice as follows. For every $j \in \left[\ell_{\mathrm{QVC}}^{\mathrm{A}}\right]$,

- if $RA\,[j] = \mathsf{S}$, then she sets $W_j^{\mathrm{A}} = (0^2)^{2r}$. Recall that when Alice computes $JS1^{\mathrm{A}}$, she believes that they have used the same number of MES blocks and have both already measured all blocks of MES pairs which were corrupted in communication using QVC. Therefore, in Alice's view the remaining rounds of communication using QVC have not been corrupted.
- Otherwise, $RA\,[j] = \widetilde{RB}\,[j] = \mathsf{M}$, i.e., Alice has measured the corresponding block of MES pairs and believes Bob has measured them as well. Using *FullMA* and $\widetilde{MB}$, Alice locates the corresponding measurement outcomes in *FullPA* and $\widetilde{PB}$ and sets W_j^{A} to be the error syndrome obtained from the measurement outcomes.

Note that if Alice computes $JS1^{\mathrm{A}}$ in an iteration with no transmission errors or hash collisions, then the computed representation $JS1^{\mathrm{A}}$ is indeed equal to $JS1$. Bob computes $JS1^{\mathrm{B}}$ similarly using his best estimate of *FullMA* and *FullPA* in the above procedure.

4.3.6 Second representation of the joint quantum state

The representation $JS2$ is obtained from $JS1$ as follows. In $JS1$, starting from the rightmost bracket, we recursively try to cancel consecutive brackets if their contents correspond to inverse of one another. Once no further such cancellation is possible, what we are left with is the $JS2$ representation, which is of the following form (when $\ell_{\mathrm{QVC}}^{\mathrm{A}} = \ell_{\mathrm{QVC}}^{\mathrm{B}}$):

$$JS2 = [\#b] \cdots [\#1][U_{gr} \cdots U_{(g-1)r+2} U_{(g-1)r+1}] \cdots [U_r \cdots U_2 U_1]\, |\psi_{\mathrm{init}}\rangle^{ABCER}, \tag{4.2}$$

where g is the largest integer such that the concatenation of the first g brackets starting from the right equals the sequence $U_{gr}, \ldots, U_2, U_1$ of unitary operations of Π. As in Section 3.3, we refer to these brackets as the "good" blocks, and the remaining b brackets which need to be actively rewound are called the "bad" blocks.

Once the parties have synchronized their classical data (to the best of their knowledge) as described earlier, they have all the information they need to compute their best guesses $JS1^{\mathrm{A}}$ and $JS1^{\mathrm{B}}$ of $JS1$. Using the same procedure described above, from $JS1^{\mathrm{A}}$ Alice computes her estimate $JS2^{\mathrm{A}}$ of $JS2$. Similarly, Bob computes $JS2^{\mathrm{B}}$ from $JS1^{\mathrm{B}}$. Note that the two parties may have different views of their joint state, based on which they decide how to further evolve the state in Π'. The rules by which Alice and Bob decide their respective types (± 1 or 0) for the next block in Π', and which blocks of unitary operations of Π (if any) are involved, are the same as the teleportation-based protocol; see Section 3.3.7.

4.3.7 Representation of the joint state while out-of-sync

We now define the representations $JS1$ and $JS2$ in the case when $\ell^{\mathrm{A}}_{\mathrm{QVC}} \neq \ell^{\mathrm{B}}_{\mathrm{QVC}}$. Note that in this case similar to the teleportation-based protocol, conditioned on the classical data with Alice and Bob and a fixed error syndrome E by the adversary, $JS1$ and $JS2$ represent a pure state. However, in addition to the $ABCER$ registers we need to include the MES blocks which have been used by only one party. Let $u \stackrel{\mathrm{def}}{=} |\ell^{\mathrm{A}}_{\mathrm{QVC}} - \ell^{\mathrm{B}}_{\mathrm{QVC}}|$. For concreteness suppose that $\ell^{\mathrm{A}}_{\mathrm{QVC}} > \ell^{\mathrm{B}}_{\mathrm{QVC}}$. Then the $JS1$ representation is of the following form:

$$JS1 = [*\ell^{\mathrm{A}}_{\mathrm{QVC}}] \cdots [*\ell^{\mathrm{B}}_{\mathrm{QVC}}] \cdots [*2][*1] \, |\psi_{\mathrm{init}}\rangle^{ABCER} \quad . \tag{4.3}$$

The content of the first $\ell^{\mathrm{B}}_{\mathrm{QVC}}$ brackets from the right corresponding to the MES blocks which have been used by both parties are obtained as described in Subsection 4.3.5. The leftmost u brackets, correspond to the MES blocks which have been used only by Alice. We refer to these blocks as the *ugly* blocks. These brackets contain Alice's unitary operations from the input protocol Π, her Pauli correction operations and QVC encoding and decoding operations, as well as all the MES registers involved in these iterations which remain untouched on Bob's side.

The representation $JS2$ is obtained from $JS1$ as follows: We denote by $[@u] \cdots [@1]$ the leftmost u brackets corresponding to the ugly blocks. We use the procedure described in Subsection 4.3.6 on the rightmost $\ell^{\mathrm{B}}_{\mathrm{QVC}}$ brackets in $JS1$ to obtain $JS2$ of the following form:

$$JS2 = [@u] \cdots [@1][\#b] \cdots [\#1][U_{gr} \cdots U_{(g-1)r+2} U_{(g-1)r+1}] \cdots [U_r \cdots U_2 U_1] \, |\psi_{\mathrm{init}}\rangle^{ABCER} \quad , \tag{4.4}$$

for some non-negative integers g and b, which we refer to as the number of *good* blocks and the number of *bad* blocks in $JS2$ representation, respectively. We point out that Alice and Bob do not compute their estimates of $JS1$ and $JS2$ unless, based on their view of the simulation so far, they believe that they have used the same number of MES

blocks. Therefore, whenever computed, $JS1^{\mathrm{A}}$, $JS1^{\mathrm{B}}$ and $JS2^{\mathrm{A}}$, $JS2^{\mathrm{B}}$ are always of the forms described in Subsections 4.3.5 and 4.3.6, respectively. Note that Alice and Bob can realize that $\ell^{\mathrm{A}}_{\mathrm{QVC}} \neq \ell^{\mathrm{B}}_{\mathrm{QVC}}$ by learning each other's metadata. Then if they do as described in Subsection 4.3.3, if no error or collision occurs, they will reduce the number of ugly blocks in $JS2$ by one. In this case, block [@1] turns into a standard block of unitary operations, potentially introducing a new bad block $[\#b+1]$. If there is either a transmission error or a hash collision, however, Bob might not realize that $\ell^{\mathrm{A}}_{\mathrm{QVC}} > \ell^{\mathrm{B}}_{\mathrm{QVC}}$. Then if he has a $+1$, -1 or 0 type of iteration then he may apply non-identity Pauli operations and unitary operations on registers BC, which still results in block [@1] becoming a standard block of unitary operations acting on registers ABC only. Otherwise, block [@1] remains as is. Similarly if there is an error or a hash collision, Alice might not realize that $\ell^{\mathrm{A}}_{\mathrm{QVC}} > \ell^{\mathrm{B}}_{\mathrm{QVC}}$. Then she might use her next block of MES registers to communicate using QVC in which case a new ugly block, call it $[@u+1]$, would be added to the left of $[@u]$.

4.3.8 Constant collision probability for classical hashing suffices

As in the teleportation-based algorithm, in our algorithm in the plain model the hash function of Lemma 3.3.1 is used to check for inconsistencies in the classical data maintained by Alice and Bob. Recall that in Chapter 3, the collision probability p for the hash function h of Lemma 3.3.1 is chosen to be $1/\mathrm{poly}(n)$. The output of the hash function is of length $o = \Theta\left(\log \frac{1}{p}\right) = \Theta(\log n)$ bits. Therefore, in the large-alphabet case the hash values corresponding to the classical data can be communicated using only a constant number of rounds. However, in the small-alphabet case, using a logarithmic number of rounds to communicate the hash values leads to a vanishing simulation rate. In our algorithm in this section, we use the hash family of Lemma 3.3.1 with a constant collision probability and show that $p = \Theta(1)$ suffices to keep the number of hash collisions low. We address this issue in the simpler large-alphabet setting to simplify the proof in the small-alphabet case at the conceptual level.

Following Haeupler [37], we circumvent the barrier explained above using the observation that hashing only makes one-sided errors. In other words, collision only occurs when the data to be compared are not equal, which in turn is a result of corruptions by the adversary. As the error budget of the adversary is bounded by $2n\epsilon$, one would expect the total number of rounds in which the classical data being compared are not equal to be bounded by $\mathrm{O}(n\epsilon)$. In fact, this allows us to have a constant collision probability while keeping the number of hash collisions in the same order as the number of transmission errors.

4.3.9 Summary of main steps

In Algorithm 13, we summarize the outline of the steps which are followed by Alice and Bob in the simulation. Note that since synchronizing recycling data creates new Pauli data,

we choose to do this step before synchronizing the Pauli data. Similar to the teleportation-based case, the algorithm is designed so that unless there is a transmission error or a hash collision in comparing a given type of data, Alice and Bob will simultaneously go down these steps while never returning to a previous step. This in fact is a crucial property used in the analysis of the algorithm.

Algorithm 13: Main steps in one iteration of the simulation for the large alphabet recycling-based model

1. Agree on the history of the simulation contained in metadata, i.e., ensure $FullMA = \widetilde{MA}$ and $FullMB = \widetilde{MB}$. This involves Algorithm 5—**rewindMD** and Algorithm 6—**extendMD**.
2. Synchronize the number of MES blocks used in QVC, in particular, ensure $\ell^{\mathrm{A}}_{\mathrm{QVC}} = \widetilde{\ell^{\mathrm{B}}_{\mathrm{QVC}}}$ and $\ell^{\mathrm{B}}_{\mathrm{QVC}} = \widetilde{\ell^{\mathrm{A}}_{\mathrm{QVC}}}$. This is done via Algorithm 19—**Q-syncMES**.
3. Agree on the measurement pointers and the recycling data up to the pointers, in particular, ensure $(\ell_{RA}, RA[1:\ell_{RA}]) = \left(\widetilde{\ell_{RB}}, \widetilde{RB}\left[1:\widetilde{\ell_{RB}}\right]\right)$ and $(\ell_{RB}, RB[1:\ell_{RB}]) = \left(\widetilde{\ell_{RA}}, \widetilde{RA}\left[1:\widetilde{\ell_{RA}}\right]\right)$. This involves Algorithm 20—**rewindRD**.
4. Ensure no undetected quantum error from earlier rounds exists. This is done by ensuring $QHA = QHB$ and involves Algorithm 21—**measuresyndrome**.
5. Ensure $\ell_{RA} = \ell^{\mathrm{A}}_{\mathrm{QVC}}$ and $\ell_{RB} = \ell^{\mathrm{B}}_{\mathrm{QVC}}$. This is achieved via Algorithm 22—**extendRD**.
6. Agree on Pauli data, in particular, ensure $FullPA = \widetilde{PA}$ and $FullPB = \widetilde{PB}$. This is done via Algorithm 23—**Q-rewindPD** and Algorithm 24—**Q-extendPD**.
7. Compute the best guess for $JS1$ and $JS2$. If there are any "bad" blocks in the guess for $JS2$, reverse the last bad block of unitary operations. I.e., implement quantum rewinding so that $b = 0$ in $JS2$. This is done in Algorithm 26—**Q-simulate**.
8. If no "bad" blocks remain, implement the next block of rounds of the original protocol. This results in an increase in g in $JS2$, and is also done through Algorithm 26—**Q-simulate**.

Note that although in step 1 we use the same algorithms for synchronizing the metadata as in the teleportation-based case (rewindMD and extendMD), the alphabet over which the metadata strings are defined are now different (see Subsection 4.4.1). The algorithms mentioned in the remaining steps are presented in the next section. Figure 4.3 summarizes the main steps in flowchart form.

4.4 Algorithm

In this section, we present our simulation protocol in the plain quantum communication model over large alphabets. First, we introduce the data structure and the variables appearing in the pseudo-code.

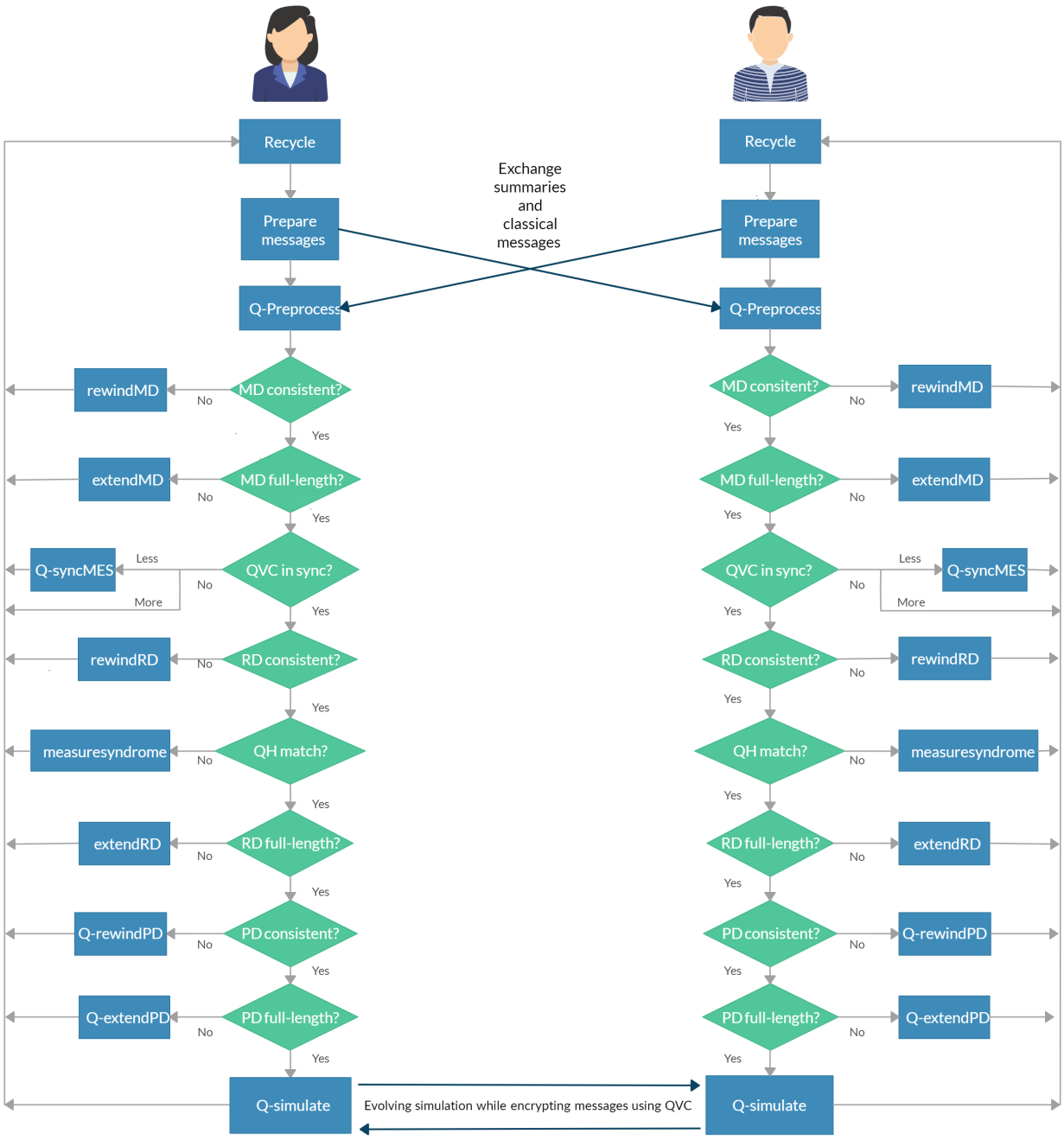

Figure 4.3: Flowchart of the recycling-based scheme for high rate noisy interactive quantum communication.

4.4.1 Data structure

Alice and Bob maintain a data structure obtained by modifying the one introduced in Section 3.4.1.

- **Metadata:** The metadata now contain new symbols corresponding to recycling operations and error detection measurements. In every iteration $NewMetaA \in \{\pm 1, 0, \mathsf{C}, 0_{\mathsf{ED}}, \mathsf{M}, \mathsf{C}'\}$ specifies Alice's action in the current iteration. Similar to the teleportation-based algorithm, in an iteration with $NewMetaA = \mathsf{C}$, Alice does not access the quantum registers and if $NewMetaA \in \{\pm 1, 0\}$, then the unitary operators of the original protocol applied locally by Alice in the current iteration have exponent $NewMetaA$. If $NewMetaA = \mathsf{M}$ Alice measures the block of MES registers specified by her measurement pointer and moves the pointer back by 1. $NewMetaA = \mathsf{C}'$ corresponds to a classical iteration for Alice in which she just moves her measurement pointer forward by 1. Finally, in an iteration with $NewMetaA = 0_{\mathsf{ED}}$, Alice completes QVC on her side as explained in Section 4.3.3. The notation 0_{ED} is used to emphasize that similar to an iteration with $NewMetaA = 0$ no unitary operator of the original protocol is applied by Alice, but she measures the block of MES registers at the end of the iteration for error detection.

  Alice records her metadata in $FullMA$ which is concatenated with $NewMetaA$ in each iteration and has length i at the end of the i-th iteration. Similar to the teleportation-based algorithm, Alice maintains a string $\widetilde{MB}$ as an estimate of Bob's metadata, which is not necessarily full-length. The length of $\widetilde{MB}$ is denoted by $\ell_{\widetilde{MB}}$. Alice also maintains ℓ_{MA}, her estimate for the length of $\widetilde{MA}$, which is with Bob. MA is defined as the prefix of $FullMA$ of length ℓ_{MA}, i.e., $MA \stackrel{\text{def}}{=} FullMA\,[1 : \ell_{MA}]$. When MA appears in any of the algorithms in this section, it is implicitly computed by Alice from $FullMA$ and ℓ_{MA}. We denote by $\ell^{\mathrm{A}}_{\mathrm{QVC}}$ the number of iterations in which Alice has performed QVC so far. Note that $\ell^{\mathrm{A}}_{\mathrm{QVC}}$ can be computed from $FullMA$ and is equal to the number of ± 1, 0, 0_{ED} symbols in $FullMA$. Bob's local metadata variables are defined similarly.

- **Recycling data:** Quantum recycling data are used to decide whether a block of MES pairs can be reused for communication using QVC. The recycling data alphabet consists of the symbol M corresponding to a measured block of MES pairs and the symbol S, used for a block of MES pairs still in superposition. Alice records her recycling data in a string RA, which is of length i after i iterations. The string RA is also computable from $FullMA$. Alice maintains a queue, $IndexA$, of indices corresponding to MES blocks to be used/reused in QVC. Similar to RA, the string $IndexA$ is of length i after i iterations. In every iteration, the recycling subroutine returns the variable $NextIndexA$ which contains the index of the recycled MES block by Alice and the string $IndexA$ is concatenated with this index. If no such index exists in an iteration, $NextIndexA$ is assigned the value $\perp$ and the protocol aborts. The variable ℓ_{RecycleA} is a pointer on RA and $IndexA$ used by Alice in the recycling subroutine to determine $NextIndexA$. Alice moves this pointer forward until it reaches the next S symbol in RA (if it exists) and records the value of $IndexA$ in this coordinate into $NextIndexA$. In every iteration, if the protocol does not abort, RA is concatenated with an S symbol corresponding to the recycled MES block and if an MES block is

measured the corresponding element of RA is changed from S to M. Alice maintains a measurement pointer, ℓ_{RA}, which together with *IndexA* specify the block of MES pairs to be measured in iterations with $NewMetaA = \mathsf{M}$. In each iteration, the pointer ℓ_{RA} may stay the same or move backward or forward by 1 and can be computed from *FullMA*. The measurement pointers also serve as reference points up to which Alice and Bob compare their recycling data in each iteration. Bob's recycling data variables are defined similarly. Alice computes $\widetilde{RB}$ and $\widetilde{\ell_{RB}}$ as her estimate of Bob's RB and ℓ_{RB}, respectively, based on her full-length estimate $\widetilde{MB}$ of *FullMB*. Note that if $\widetilde{MB} = FullMB$ then Alice's estimates of Bob's recycling data are correct.

- **Pauli data:** In any iteration, new Pauli data is generated on Alice's side if and only if she measures a block of MES pairs or performs QVC locally. *NewPauliA* has three parts: If a block of r MES pairs is measured locally by Alice in the current iteration then the measurement outcome, $(m_1, m_2) \in \Sigma^{4r}$, is recorded in the first two parts of *NewPauliA*, and the third part contains $\perp^{2r}$, corresponding to no Pauli corrections. Otherwise, if Alice performs QVC then $\perp^{2r}$ is recorded in each of the first two parts and the third part similar to the teleportation-based protocol specifies the Pauli corrections.

  Alice records her Pauli data in *FullPA*. Starting from the empty string, *FullPA* is concatenated with *NewPauliA* whenever Alice measures an MES block or performs QVC. She maintains a string $\widetilde{PB}$ as an estimate of Bob's Pauli data. The length of $\widetilde{PB}$ is denoted by $\ell_{\widetilde{PB}}$. Alice also maintains ℓ_{PA}, her estimate for the length of $\widetilde{PA}$, which is with Bob. PA denotes the prefix of *FullPA* of length ℓ_{PA}, i.e., $PA \stackrel{\text{def}}{=} FullPA\,[1 : \ell_{PA}]$. When PA appears in any of the algorithms in this section, it is implicitly computed by Alice from *FullPA* and ℓ_{PA}. We define $q_{MA} \stackrel{\text{def}}{=} |FullPA|\,/6r$. Note that q_{MA} can be computed from *FullMA*. Alice computes her estimate $q_{\widetilde{MB}}$ of q_{MB} using her full-length estimate $\widetilde{MB}$ of *FullMB*. Bob's Pauli data variables are defined similarly.

- As in Chapter 3, we use H with different variables as subscript to represent hash values, e.g., H_{MA} represents a hash value corresponding to MA. We use QHA and QHB to represent quantum hash values. The data variables with a superscript $'$ denote the received data after transmission over the noisy channel, e.g., ℓ'_{MB} denotes what Alice receives when Bob sends ℓ_{MB}.

- The variable *Itertype* takes two new values: RD corresponding to recycling data synchronization and QH corresponding to an iteration in which the received quantum hash value does not match the locally computed quantum hash value.

Finally, L_{QVC} denotes the total number of MES blocks to be used as the keys in QVC.

Remark 4.4.1. We point out an additional subtlety in interpreting the Pauli data by the two parties. Recall that when Alice and Bob compute their estimates of the joint state, they use the metadata to locate in the Pauli data, the measurement outcomes for each block

of measured MES pairs. However, due to transmission errors or collisions, it is possible to have an inconsistency between the metadata and the Pauli data. For instance, $\widetilde{MA}$ may indicate that a specific block of $\widetilde{PA}$ contains the measurement outcomes on an MES block, whereas it actually has $\perp^{2r}$ in the first two parts and corresponds to an iteration in which Alice has performed QVC. In any such scenario, Alice and Bob interpret the $\perp$ symbols as 0 and compute the joint state. This most likely introduces new errors on the joint state from which Alice and Bob can recover once they obtain a correct view of the simulation.

4.4.2 Pseudo-code

This section contains the pseudo-code for the main algorithm and the subroutines that each party runs locally in the simulation protocol. For each subroutine, we list all the global variables accessed by the subroutine as the **Input** at the beginning of the subroutine. Whenever applicable, the relation between the variables when the subroutine is called is stated as the **Promise** and the global variables which are modified by the subroutine are listed as the **Output**.

Algorithm 14: Q-Main (Alice's side)

Input: n round protocol Π in plain quantum model over polynomial-size alphabet Σ

Q-Initialization;
For $i = 1 \to R_{\text{total}}$
  if $i \leq L_{\text{QVC}}$ **then**
    $NextIndexA \leftarrow i$; `// No recycling in first` L_{QVC} `iterations`
  else
    Recycle;
    if $NextIndexA = \perp$ **then**
      Abort;
  $IndexA \leftarrow (IndexA, NextIndexA)$;
  $RA \leftarrow (RA, \mathsf{S})$;
  ▷ `computing hash values`
  $H_{MA} \leftarrow h_{S_{4i-3}}(MA)$;
  $H_{\widetilde{MB}} \leftarrow h_{S_{4i-2}}\left(\widetilde{MB}\right)$;
  $H_{PA} \leftarrow h_{S_{4i-1}}(PA)$;
  $H_{\widetilde{PB}} \leftarrow h_{S_{4i}}\left(\widetilde{PB}\right)$;
  Quantum-hash;

Algorithm 14: Q-Main (Alice's side, cont. from previous page)

16

17 Send

$$\left(H_{MA}, \ell_{MA}, H_{\widetilde{MB}}, \ell_{\widetilde{MB}}, H_{PA}, \ell_{PA}, H_{\widetilde{PB}}, \ell_{\widetilde{PB}}, QHA\right);$$

18 Receive

$$\left(H'_{\widetilde{MA}}, \ell'_{\widetilde{MA}}, H'_{MB}, \ell'_{MB}, H'_{\widetilde{PA}}, \ell'_{\widetilde{PA}}, H'_{PB}, \ell'_{PB}, QHB'\right);$$

19 **Q-Preprocess**; ▷ `Determining iteration type`

20 **if** *Itertype* $\neq$ SIM **then**

21 Send *msg*;

22 Receive *msg*′;

`// messages are communicated alternately`

▷ `Case i.A`

23 **if** *Itertype* = MD and *RewindExtend* = R **then**

24 **rewindMD**;

▷ `Case i.B`

25 **else if** *Itertype* = MD and *RewindExtend* = E **then**

26 **extendMD**;

▷ `Case ii.A`

27 **else if** *Itertype* = MES and *NewMetaA* = C **then**

28 return;

▷ `Case ii.B`

29 **else if** *Itertype* = MES and *NewMetaA* = 0_{ED} **then**

30 **Q-syncMES**;

▷ `Case iii`

31 **else if** *Itertype* = RD and *RewindExtend* = R **then**

32 **rewindRD**;

▷ `Case iv`

33 **else if** *Itertype* = QH **then**

34 **measuresyndrome**;

Algorithm 14: Q-Main (Alice's side, cont. from previous page)

```
35
                                                          ▷ Case v
36      else if Itertype = RD and RewindExtend = E then
37          extendRD;
                                                          ▷ Case vi.A
38      else if Itertype = PD and RewindExtend = R then
39          rewindPD;
                                                          ▷ Case vi.B
40      else if Itertype = PD and RewindExtend = E then
41          extendPD;
                                                          ▷ Case vii
42      else
43          Q-Simulate;
44  return Q-Main;
```

Algorithm 15: Q-Preprocess (Alice's side)

Input:

$$\begin{pmatrix} H_{MA}, \ell_{MA}, H_{\widetilde{MB}}, \ell_{\widetilde{MB}}, H_{PA}, \ell_{PA}, H_{\widetilde{PB}}, \ell_{\widetilde{PB}}, QHA \\ H'_{\widetilde{MA}}, \ell'_{\widetilde{MA}}, H'_{MB}, \ell'_{MB}, H'_{\widetilde{PA}}, \ell'_{\widetilde{PA}}, H'_{PB}, \ell'_{PB}, QHB' \\ FullMA, \ell^{\mathrm{A}}_{\mathrm{QVC}}, \widetilde{MB}, RA, \ell_{RA}, FullPA, \widetilde{PB} \end{pmatrix}$$

Output: $\left(Itertype, RewindExtend, NewMetaA, FullMA, \ell_{MA}, \widetilde{NewMetaB}, \widetilde{MB}, \ell_{\widetilde{MB}}, msg\right)$

```
1  if (H_MA, H_~MB) = (H'_~MA, H'_MB) and ℓ_MA = ℓ'_~MA = ℓ_~MB = ℓ'_MB = i − 1 then
2      Compute ℓ^B_QVC, R~B, ℓ~_RB, q_~MB;
                                                          ▷ Processing Metadata
                                                          ▷ Case i.A
3  if (H_MA, H_~MB, ℓ_MA, ℓ_~MB) ≠ (H'_~MA, H'_MB, ℓ'_~MA, ℓ'_MB) then
4      Itertype ← MD;
5      RewindExtend ← R;
6      NewMetaA ← C;
7      FullMA ← (FullMA, NewMetaA);
8      msg ← dummy message of length r;
```

Algorithm 15: Q-Preprocess (Alice's side, cont. from previous page)

▷ `Case i.B`

else if $(\ell_{MA} < i-1)$ or $\left(\ell_{\widetilde{MB}} < i-1\right)$ **then**
  $\mathit{Itertype} \leftarrow \mathrm{MD}$;
  $\mathit{RewindExtend} \leftarrow \mathsf{E}$;
  $\mathit{NewMetaA} \leftarrow \mathsf{C}$;
  $\mathit{FullMA} \leftarrow (\mathit{FullMA}, \mathit{NewMetaA})$;
  if $\ell_{MA} < i-1$ **then**
    $msg \leftarrow \mathrm{encodeMD}\left(\mathit{FullMA}\left[\ell_{MA}+1, \ell_{MA}+2\right]\right)$; `// Encode MD in` Σ^r
  else
    $msg \leftarrow$ dummy message of length r;

▷ `Comparing number of used MES blocks`

▷ `Case ii.A`

else if $\ell^{\mathrm{A}}_{\mathrm{QVC}} > \widetilde{\ell^{\mathrm{B}}_{\mathrm{QVC}}}$ **then**
  $\mathit{Itertype} \leftarrow \mathrm{MES}$;
  $\mathit{NewMetaA} \leftarrow \mathsf{C}$;
  $\mathit{FullMA} \leftarrow (\mathit{FullMA}, \mathit{NewMetaA})$;
  $\ell_{MA} \leftarrow \ell_{MA} + 1$;
  $\widetilde{\mathit{NewMetaB}} \leftarrow \mathsf{0}_{\mathsf{ED}}$;
  $\widetilde{MB} \leftarrow \left(\widetilde{MB}, \widetilde{\mathit{NewMetaB}}\right)$;
  $\ell_{\widetilde{MB}} \leftarrow \ell_{\widetilde{MB}} + 1$;
  $msg \leftarrow$ dummy message of length r;

▷ `Case ii.B`

else if $\ell^{\mathrm{A}}_{\mathrm{QVC}} < \widetilde{\ell^{\mathrm{B}}_{\mathrm{QVC}}}$ **then**
  $\mathit{Itertype} \leftarrow \mathrm{MES}$;
  $\mathit{NewMetaA} \leftarrow \mathsf{0}_{\mathsf{ED}}$;
  $\mathit{FullMA} \leftarrow (\mathit{FullMA}, \mathit{NewMetaA})$;
  $\ell_{MA} \leftarrow \ell_{MA} + 1$;
  $\widetilde{\mathit{NewMetaB}} \leftarrow \mathsf{C}$;
  $\widetilde{MB} \leftarrow \left(\widetilde{MB}, \widetilde{\mathit{NewMetaB}}\right)$;
  $\ell_{\widetilde{MB}} \leftarrow \ell_{\widetilde{MB}} + 1$;
  $msg \leftarrow$ dummy message of length r;

Algorithm 15: Q-Preprocess (Alice's side, cont. from previous page)

▷ `Processing recycling data`

▷ `Case iii`

else if $(\ell_{RA}, RA[1:\ell_{RA}]) \neq \left(\widetilde{\ell_{RB}}, \widetilde{RB}\left[1:\widetilde{\ell_{RB}}\right]\right)$ **then**
  $Itertype \leftarrow$ RD;
  $RewindExtend \leftarrow \mathsf{R}$;
  if $\ell_{RA} > \widetilde{\ell_{RB}}$ **then**
    $NewMetaA \leftarrow \mathsf{M}$;
    $\widetilde{NewMetaB} \leftarrow \mathsf{C}$;
  else if $\ell_{RA} < \widetilde{\ell_{RB}}$ **then**
    $NewMetaA \leftarrow \mathsf{C}$;
    $\widetilde{NewMetaB} \leftarrow \mathsf{M}$;
  else
    $NewMetaA \leftarrow \mathsf{M}$;
    $\widetilde{NewMetaB} \leftarrow \mathsf{M}$;
  $FullMA \leftarrow (FullMA, NewMetaA)$;
  $\ell_{MA} \leftarrow \ell_{MA} + 1$;
  $\widetilde{MB} \leftarrow \left(\widetilde{MB}, \widetilde{NewMetaB}\right)$;
  $\ell_{\widetilde{MB}} \leftarrow \ell_{\widetilde{MB}} + 1$;
  $msg \leftarrow$ dummy message of length r;

▷ `Case iv`

else if $QHA \neq QHB'$ **then**
  $Itertype \leftarrow$ QH;
  $NewMetaA \leftarrow \mathsf{M}$;
  $FullMA \leftarrow (FullMA, NewMetaA)$;
  $\ell_{MA} \leftarrow \ell_{MA} + 1$;
  $\widetilde{NewMetaB} \leftarrow \mathsf{M}$;
  $\widetilde{MB} \leftarrow \left(\widetilde{MB}, \widetilde{NewMetaB}\right)$;
  $\ell_{\widetilde{MB}} \leftarrow \ell_{\widetilde{MB}} + 1$;
  $msg \leftarrow$ dummy message of length r;

Algorithm 15: Q-Preprocess (Alice's side, cont. from previous page)

▷ `Case v`

else if $\ell_{RA} < \ell^{\mathrm{A}}_{\mathrm{QVC}}$ **then**
- $Itertype \leftarrow \mathrm{RD}$;
- $RewindExtend \leftarrow \mathsf{E}$;
- $NewMetaA \leftarrow \mathsf{C}'$;
- $FullMA \leftarrow (FullMA, NewMetaA)$;
- $\ell_{MA} \leftarrow \ell_{MA} + 1$;
- $\widetilde{NewMetaB} \leftarrow \mathsf{C}'$;
- $\widetilde{MB} \leftarrow \left(\widetilde{MB}, \widetilde{NewMetaB}\right)$;
- $\ell_{\widetilde{MB}} \leftarrow \ell_{\widetilde{MB}} + 1$;
- $msg \leftarrow$ dummy message of length r;

▷ `Processing Pauli data`

▷ `Case vi.A`

else if $\left(H_{PA}, H_{\widetilde{PB}}, \ell_{PA}, \ell_{\widetilde{PB}}\right) \neq \left(H'_{\widetilde{PA}}, H'_{PB}, \ell'_{\widetilde{PA}}, \ell'_{PB}\right)$ **then**
- $Itertype \leftarrow \mathrm{PD}$;
- $RewindExtend \leftarrow \mathsf{R}$;
- $NewMetaA \leftarrow \mathsf{C}$;
- $FullMA \leftarrow (FullMA, NewMetaA)$;
- $\ell_{MA} \leftarrow \ell_{MA} + 1$;
- $\widetilde{NewMetaB} \leftarrow \mathsf{C}$;
- $\widetilde{MB} \leftarrow \left(\widetilde{MB}, \widetilde{NewMetaB}\right)$;
- $\ell_{\widetilde{MB}} \leftarrow \ell_{\widetilde{MB}} + 1$;
- $msg \leftarrow$ dummy message of length r;

▷ `Case vi.B`

else if $(\ell_{PA} < 6q_{MA} \cdot r)$ or $\left(\ell_{\widetilde{PB}} < 6q_{\widetilde{MB}} \cdot r\right)$ **then**
- $Itertype \leftarrow \mathrm{PD}$;
- $RewindExtend \leftarrow \mathsf{E}$;
- $NewMetaA \leftarrow \mathsf{C}$;
- $FullMA \leftarrow (FullMA, NewMetaA)$;
- $\ell_{MA} \leftarrow \ell_{MA} + 1$;
- $\widetilde{NewMetaB} \leftarrow \mathsf{C}$;
- $\widetilde{MB} \leftarrow \left(\widetilde{MB}, \widetilde{NewMetaB}\right)$;
- $\ell_{\widetilde{MB}} \leftarrow \ell_{\widetilde{MB}} + 1$;
- **if** $\ell_{PA} < 6q_{MA} \cdot r$ **then**
  - $msg \leftarrow FullPA\,[\ell_{PA} + 1, \ell_{PA} + r]$

Algorithm 15: Q-Preprocess (Alice's side, cont. from previous page)

▷ `Case vii`

else
  Q-Computejointstate;
  $Itertype \leftarrow$ SIM;
  $FullMA = (FullMA, NewMetaA)$;
  $\ell_{MA} \leftarrow \ell_{MA} + 1$;
  $\widetilde{MB} \leftarrow \left(\widetilde{MB}, \widetilde{NewMetaB}\right)$;
  $\ell_{\widetilde{MB}} \leftarrow \ell_{\widetilde{MB}} + 1$;
return Q-Preprocess;

Algorithm 16: Q-Initialization (Alice's side)

Initialize
  $L_{\text{QVC}} \leftarrow \Theta(n\epsilon)$;
  $r \leftarrow \Theta(1/\sqrt{\epsilon})$;
  $R_{\text{total}} \leftarrow \lceil n/2r + \Theta(n\epsilon) \rceil$;
  $t \leftarrow \Theta(n\epsilon)$;
  $\ell_{MA}, \ell_{\widetilde{MB}}, \ell_{PA}, \ell_{\widetilde{PB}}, \ell^{\text{A}}_{\text{QVC}}, \ell_{RA}, \ell_{\text{RecycleA}} \leftarrow 0$;
  $FullMA, \widetilde{MB}, FullPA, \widetilde{PB}, RA, IndexA \leftarrow \emptyset$;
$h \leftarrow$ hash function of Lemma 3.3.1 with $p = \Theta(1), o = \Theta(1), s = \Theta(\log n)$;
Robust Entanglement Distribution ;
Reserve $L_{\text{QVC}} \cdot 4r$ MES pairs to be used as the keys in QVC ;
Reserve $10R_{\text{total}}$ MESs to be used in quantum hashing ;
Measure $\Theta(R_{\text{total}})$ MESs in the computational basis and record the binary representation of the outcomes in $S_1, \ldots, S_{4R_{\text{total}}}$;
  `//` $4R_{\text{total}}$ `seeds of length` s `for the hash function` h
Measure the remaining $\Theta(n\sqrt{\epsilon})$ MESs in the computational basis and record the binary representation of the outcomes in R' ;
Stretch R' to a δ-biased pseudo-random string $R = R_1, \ldots, R_{10R_{\text{total}}}$ using the deterministic algorithm of Lemma 2.4.8 where $\delta = 2^{-\Theta\left(\frac{n}{r}\right)}$;
  `//` $10R_{\text{total}}$ `seeds of length` $4rL_{\text{QVC}}$ `used in quantum hashing`
return Q-Initialization;

Algorithm 17: Recycle (Alice's side)

Input: $\left(RA, IndexA, \ell_{\text{RecycleA}}, \ell^{\text{A}}_{\text{QVC}}\right)$
Output: $(\ell_{\text{RecycleA}}, NextIndexA)$

$\ell_{\text{RecycleA}} \leftarrow \ell_{\text{RecycleA}} + 1$;
while $\left(\ell_{\text{RecycleA}} < \ell^{\text{A}}_{\text{QVC}}\right)$ and $(RA\left[\ell_{\text{RecycleA}}\right] = \mathsf{M})$ **do**
  $\ell_{\text{RecycleA}} \leftarrow \ell_{\text{RecycleA}} + 1$;
if $\ell_{\text{RecycleA}} = \ell^{\text{A}}_{\text{QVC}}$ **then**
  $NextIndexA \leftarrow \perp$;
else
  $NextIndexA \leftarrow IndexA\left[\ell_{\text{RecycleA}}\right]$;
return Recycle;

Algorithm 18: Quantum-hash (Alice's side)

Input: $\left(RA, IndexA, \ell^{\text{A}}_{\text{QVC}}, \ell_{\text{RecycleA}}, R\right)$
Output: QHA

$QHA \leftarrow \emptyset$;
for $k = 1 \rightarrow 10$ **do**
  Choose a fresh MES from "Quantum Hash" category, and let V_{A} denote the register containing Alice's half of the state;
  for $j = 1 \rightarrow 4r\left(\ell^{\text{A}}_{\text{QVC}} - \ell_{\text{RecycleA}}\right)$ **do** `// Hashing blocks` $\ell_{\text{RecycleA}} + 1 : \ell^{\text{A}}_{\text{QVC}}$
    if $R_{10i+k}\left[j\right] = 1$ and $RA\left[\ell_{\text{RecycleA}} + \lceil \frac{j}{4r} \rceil\right] \neq \mathsf{M}$ **then**
      Apply $(\text{c-X})_{V_{\text{A}}A_b}$, where $b = 4r \cdot IndexA\left[\ell_{\text{RecycleA}} + \lfloor \frac{j}{4r} \rfloor\right] + (j \bmod 4r)$;
  Apply the Fourier transform operator F on V_{A}, measure it in the computational basis and record the outcome in qh;
  $QHA \leftarrow (QHA, qh)$;
return Quantum-hash;

Algorithm 19: Q-syncMES (Alice's side)

Input: $\left(IndexA, \ell^{\mathrm{A}}_{\mathrm{QVC}}, msg', FullPA\right)$

Promise: $\left(H_{MA}, H_{\widetilde{MB}}, \ell_{MA}, \ell_{\widetilde{MB}}\right) = \left(H'_{\widetilde{MA}}, H'_{MB}, \ell'_{\widetilde{MA}} + 1, \ell'_{MB} + 1\right)$,
$\ell_{MA} = \ell_{\widetilde{MB}} = i$, $\ell^{\mathrm{A}}_{\mathrm{QVC}} < \widetilde{\ell^{\mathrm{B}}_{\mathrm{QVC}}}$

Output: $\left(\ell^{\mathrm{A}}_{\mathrm{QVC}}, NewPauliA, FullPA\right)$

Let C_0 be the communication register at the beginning of the current iteration (which is in Alice's possession) and for every $j \in [r]$, let C_j denote the communication register containing $msg'(j)$, Bob's j-th message in this iteration;

$\ell^{\mathrm{A}}_{\mathrm{QVC}} \leftarrow \ell^{\mathrm{A}}_{\mathrm{QVC}} + 1$;

Let $E_1 E_2 \cdots E_{4r}$ be the registers with Alice that contain halves of the $4r$ MESs in the block indexed by $IndexA\left[\ell^{\mathrm{A}}_{\mathrm{QVC}}\right]$;

Apply

$$(\text{c-Z})_{E_2 C_0} (\text{c-X})_{E_1 C_0}$$

For every $j \in [r-1]$, upon receiving C_j apply

$$(\text{c-Z})_{E_{4j+2} C_j} (\text{c-X})_{E_{4j+1} C_j} \left(\text{c-X}^{-1}\right)_{E_{4j-1} C_j} \left(\text{c-Z}^{-1}\right)_{E_{4j} C_j}$$

Upon receiving C_r apply

$$\left(\text{c-X}^{-1}\right)_{E_{4r-1} C_r} \left(\text{c-Z}^{-1}\right)_{E_{4r} C_r}$$

// See Section 4.3.3 for the rationale and Bob's analogue of above steps

Apply the Fourier transform operator to $E_1, E_2, \cdots, E_{4r}$ and measure them in the computational basis. Store the measurement outcomes in $(m_1, m_2) \in \Sigma^{4r}$;

$RA\left[\ell^{\mathrm{A}}_{\mathrm{QVC}}\right] \leftarrow \mathsf{M}$;

$NewPauliA \leftarrow (m_1, m_2, \perp^{2r})$;

$FullPA \leftarrow (FullPA, NewPauliA)$;

return Q-syncMES;

Algorithm 20: rewindRD (Alice's side)

Input: $(NewMetaA, RA, \ell_{RA}, IndexA, FullPA)$

Promise: $\left(H_{MA}, H_{\widetilde{MB}}, \ell_{MA}, \ell_{\widetilde{MB}}, \ell^{\mathrm{A}}_{\mathrm{QVC}}\right) = \left(H'_{\widetilde{MA}}, H'_{MB}, \ell'_{\widetilde{MA}} + 1, \ell'_{MB} + 1, \widetilde{\ell^{\mathrm{B}}_{\mathrm{QVC}}}\right)$,
$\ell_{MA} = \ell_{\widetilde{MB}} = i$, $(\ell_{RA}, RA\left[1 : \ell_{RA}\right]) \neq \left(\widetilde{\ell_{RB}}, \widetilde{RB}\left[1 : \widetilde{\ell_{RB}}\right]\right)$

Output: $(NewPauliA, FullPA, RA, \ell_{RA})$

if $NewMetaA = \mathsf{M}$ and $RA\left[\ell_{RA}\right] = \mathsf{S}$ **then**
  Sequentially apply the Fourier transform operator to all the MESs in the block indexed by $IndexA\left[\ell_{RA}\right]$ and measure them in the computational basis ;
  Store the measurement outcomes in $(m_1, m_2) \in \Sigma^{4r}$;
  $NewPauliA \leftarrow (m_1, m_2, \perp^{2r})$;
  $FullPA \leftarrow (FullPA, NewPauliA)$;
  $RA\left[\ell_{RA}\right] \leftarrow \mathsf{M}$;
$\ell_{RA} \leftarrow \ell_{RA} - 1$;
return rewindRD;

Algorithm 21: measuresyndrome (Alice's side)

Input: $(RA, \ell_{RA}, IndexA, FullPA)$

Promise: $\left(H_{MA}, H_{\widetilde{MB}}, \ell_{MA}, \ell_{\widetilde{MB}}, \ell^{\mathrm{A}}_{\mathrm{QVC}}\right) = \left(H'_{\widetilde{MA}}, H'_{MB}, \ell'_{\widetilde{MA}} + 1, \ell'_{MB} + 1, \widetilde{\ell^{\mathrm{B}}_{\mathrm{QVC}}}\right)$,
$\ell_{MA} = \ell_{\widetilde{MB}} = i$, $(\ell_{RA}, RA\left[1 : \ell_{RA}\right]) = \left(\widetilde{\ell_{RB}}, \widetilde{RB}\left[1 : \widetilde{\ell_{RB}}\right]\right)$,
$QHA \neq QHB'$

Output: $(NewPauliA, FullPA, RA, \ell_{RA})$

if $RA\left[\ell_{RA}\right] = \mathsf{S}$ **then**
  Sequentially apply the Fourier transform operator to all the MESs in the block indexed by $IndexA\left[\ell_{RA}\right]$ and measure them in the computational basis ;
  Store the measurement outcomes in $(m_1, m_2) \in \Sigma^{4r}$;
  $NewPauliA \leftarrow (m_1, m_2, \perp^{2r})$;
  $FullPA \leftarrow (FullPA, NewPauliA)$;
  $RA\left[\ell_{RA}\right] \leftarrow \mathsf{M}$;
$\ell_{RA} \leftarrow \ell_{RA} - 1$
return measuresyndrome;

Algorithm 22: extendRD (Alice's side)

Input: ℓ_{RA}

Promise: $\left(H_{MA}, H_{\widetilde{MB}}, \ell_{MA}, \ell_{\widetilde{MB}}, \ell^{\mathrm{A}}_{\mathrm{QVC}}\right) = \left(H'_{\widetilde{MA}}, H'_{MB}, \ell'_{\widetilde{MA}} + 1, \ell'_{MB} + 1, \widetilde{\ell^{\mathrm{B}}_{\mathrm{QVC}}}\right)$,
$\ell_{MA} = \ell_{\widetilde{MB}} = i$, $(\ell_{RA}, RA\left[1 : \ell_{RA}\right]) = \left(\widetilde{\ell_{RB}}, \widetilde{RB}\left[1 : \widetilde{\ell_{RB}}\right]\right)$,
$QHA = QHB'$, $\ell_{RA} < \ell^{\mathrm{A}}_{\mathrm{QVC}}$

Output: ℓ_{RA}

$\ell_{RA} \leftarrow \ell_{RA} + 1$;
return extendRD;

Algorithm 23: Q-rewindPD (Alice's side)

Input: $\left(H_{PA}, \ell_{PA}, H_{\widetilde{PB}}, \ell_{\widetilde{PB}}, H'_{\widetilde{PA}}, \ell'_{\widetilde{PA}}, H'_{PB}, \ell'_{PB}\right)$

Promise: $\left(H_{MA}, H_{\widetilde{MB}}, \ell_{MA}, \ell_{\widetilde{MB}}, \ell^{\mathrm{A}}_{\mathrm{QVC}}\right) = \left(H'_{\widetilde{MA}}, H'_{MB}, \ell'_{\widetilde{MA}} + 1, \ell'_{MB} + 1, \widetilde{\ell^{\mathrm{B}}_{\mathrm{QVC}}}\right)$,
$\ell_{MA} = \ell_{\widetilde{MB}} = i$, $(\ell_{RA}, RA\left[1 : \ell_{RA}\right]) = \left(\widetilde{\ell_{RB}}, \widetilde{RB}\left[1 : \widetilde{\ell_{RB}}\right]\right)$,
$QHA = QHB'$, $\ell_{RA} = \ell^{\mathrm{A}}_{\mathrm{QVC}}$,
$\left(H_{PA}, H_{\widetilde{PB}}, \ell_{PA}, \ell_{\widetilde{PB}}\right) \neq \left(H'_{\widetilde{PA}}, H'_{PB}, \ell'_{\widetilde{PA}}, \ell'_{PB}\right)$.

Output: $\left(\ell_{PA}, \ell_{\widetilde{PB}}\right)$

if $\ell_{PA} \neq \ell'_{\widetilde{PA}}$ or $\ell_{\widetilde{PB}} \neq \ell'_{PB}$ **then**
  if $\ell_{PA} > \ell'_{\widetilde{PA}}$ **then**
    $\ell_{PA} \leftarrow \ell_{PA} - r$;
  if $\ell_{\widetilde{PB}} > \ell'_{PB}$ **then**
    $\ell_{\widetilde{PB}} \leftarrow \ell_{\widetilde{PB}} - r$;
else
  if $H_{PA} \neq H'_{\widetilde{PA}}$ **then**
    $\ell_{PA} \leftarrow \ell_{PA} - r$;
  if $H_{\widetilde{PB}} \neq H'_{PB}$ **then**
    $\ell_{\widetilde{PB}} \leftarrow \ell_{\widetilde{PB}} - r$;
return Q-rewindPD;

Algorithm 24: Q-extendPD (Alice's side)

Input: $\left(\ell_{PA}, \ell_{\widetilde{PB}}, \widetilde{PB}, q_{MA}, q_{\widetilde{MB}}, msg'\right)$

Promise: $\left(H_{MA}, H_{\widetilde{MB}}, \ell_{MA}, \ell_{\widetilde{MB}}, \ell^{\mathrm{A}}_{\mathrm{QVC}}\right) = \left(H'_{\widetilde{MA}}, H'_{MB}, \ell'_{\widetilde{MA}} + 1, \ell'_{MB} + 1, \widetilde{\ell^{\mathrm{B}}_{\mathrm{QVC}}}\right)$,
$\ell_{MA} = \ell_{\widetilde{MB}} = i$, $(\ell_{RA}, RA\,[1 : \ell_{RA}]) = \left(\widetilde{\ell_{RB}}, \widetilde{RB}\left[1 : \widetilde{\ell_{RB}}\right]\right)$,
$QHA = QHB'$, $\ell_{RA} = \ell^{\mathrm{A}}_{\mathrm{QVC}}$,
$\left(H_{PA}, H_{\widetilde{PB}}, \ell_{PA}, \ell_{\widetilde{PB}}\right) = \left(H'_{\widetilde{PA}}, H'_{PB}, \ell'_{\widetilde{PA}}, \ell'_{PB}\right)$,
$\ell_{PA} < 6q_{MA} \cdot r \quad \text{or} \quad \ell_{\widetilde{PB}} < 6q_{\widetilde{MB}} \cdot r.$

Output: $\left(\ell_{PA}, \widetilde{PB}, \ell_{\widetilde{PB}}\right)$

1 **if** $\ell_{PA} < 6q_{MA} \cdot r$ **then**
2 $\quad \ell_{PA} \leftarrow \ell_{PA} + r$;
3 **if** $\ell_{\widetilde{PB}} < 6q_{\widetilde{MB}} \cdot r$ **then**
4 $\quad \widetilde{PB}\left[\ell_{\widetilde{PB}} + 1 : \ell_{\widetilde{PB}} + r\right] \leftarrow msg'$;
5 $\quad \ell_{\widetilde{PB}} \leftarrow \ell_{\widetilde{PB}} + r$;
6 **return Q-extendPD**;

Algorithm 25: Q-computejointstate (Alice's side)

Input: $\left(FullMA, \widetilde{MB}, RA, FullPA, \widetilde{PB}\right)$

Promise: $\left(H_{MA}, H_{\widetilde{MB}}, \ell_{MA}, \ell_{\widetilde{MB}}, \ell^{\mathrm{A}}_{\mathrm{QVC}}\right) = \left(H'_{\widetilde{MA}}, H'_{MB}, \ell'_{\widetilde{MA}}, \ell'_{MB}, \widetilde{\ell^{\mathrm{B}}_{\mathrm{QVC}}}\right)$,
$\ell_{MA} = \ell_{\widetilde{MB}} = i - 1$, $(\ell_{RA}, RA\,[1 : \ell_{RA}]) = \left(\widetilde{\ell_{RB}}, \widetilde{RB}\left[1 : \widetilde{\ell_{RB}}\right]\right)$,
$QHA = QHB'$, $\ell_{RA} = \ell^{\mathrm{A}}_{\mathrm{QVC}}$,
$\left(H_{PA}, H_{\widetilde{PB}}, \ell_{PA}, \ell_{\widetilde{PB}}\right) = \left(H'_{\widetilde{PA}}, H'_{PB}, \ell'_{\widetilde{PA}}, \ell'_{PB}\right)$,
$\ell_{PA} = 6q_{MA} \cdot r$, $\ell_{\widetilde{PB}} = 6q_{\widetilde{MB}} \cdot r$,

Output: $\left(JS1^{\mathrm{A}}, JS2^{\mathrm{A}}, NewMetaA, \widetilde{NewMetaB}, Block, RewindExtend, P_{\mathrm{Corr}}, \widetilde{P_{\mathrm{Corr}}}\right)$

1 Compute $JS1^{\mathrm{A}}$;
2 Compute $JS2^{\mathrm{A}}$;
3 Compute $NewMetaA$;
4 Compute $\widetilde{NewMetaB}$;
5 Compute $RewindExtend$;
6 Compute $Block$;
7 Compute P_{Corr};
8 Compute $\widetilde{P_{\mathrm{Corr}}}$;
`// Refer to Sections 4.3.5, 4.3.6 to see how these variables are computed`
9 **return Q-computejointstate**;

Algorithm 26: Q-simulate (Alice's side)

Input:

$$\left(\ell^{\mathrm{A}}_{\mathrm{QVC}}, \mathit{IndexA}, \mathit{FullPA}, \ell_{PA}, \widetilde{PB}, \ell_{\widetilde{PB}}, \mathit{NewMetaA}, \mathit{RewindExtend}, \mathit{Block}, P_{\mathrm{Corr}}, \widetilde{P_{\mathrm{Corr}}}\right)$$

Promise: $\left(H_{MA}, H_{\widetilde{MB}}, \ell_{MA}, \ell_{\widetilde{MB}}, \ell^{\mathrm{A}}_{\mathrm{QVC}}\right) = \left(H'_{\widetilde{MA}}, H'_{MB}, \ell'_{\widetilde{MA}}+1, \ell'_{MB}+1, \widetilde{\ell^{\mathrm{B}}_{\mathrm{QVC}}}\right)$,
$\ell_{MA} = \ell_{\widetilde{MB}} = i$, $(\ell_{RA}, RA[1:\ell_{RA}]) = \left(\widetilde{\ell_{RB}}, \widetilde{RB}\left[1:\widetilde{\ell_{RB}}\right]\right)$,
$QHA = QHB'$, $\ell_{RA} = \ell^{\mathrm{A}}_{\mathrm{QVC}}$,
$\left(H_{PA}, H_{\widetilde{PB}}, \ell_{PA}, \ell_{\widetilde{PB}}\right) = \left(H'_{\widetilde{PA}}, H'_{PB}, \ell'_{\widetilde{PA}}, \ell'_{PB}\right)$,
$\ell_{PA} = 6q_{MA}\cdot r$, $\ell_{\widetilde{PB}} = 6q_{\widetilde{MB}}\cdot r$,

Output: $\left(\ell^{\mathrm{A}}_{\mathrm{QVC}}, \mathit{NewPauliA}, \mathit{FullPA}, \ell_{PA}, \widetilde{\mathit{NewPauliB}}, \widetilde{PB}, \ell_{\widetilde{PB}}, \ell_{RA}\right)$

$\ell^{\mathrm{A}}_{\mathrm{QVC}} \leftarrow \ell^{\mathrm{A}}_{\mathrm{QVC}} + 1$;
Continue the simulation of the noiseless protocol according to the output of **Q-computejointstate** using the block of MESs indexed by $\mathit{IndexA}\left[\ell^{\mathrm{A}}_{\mathrm{QVC}}\right]$;
$\mathit{NewPauliA} \leftarrow (\perp^{2r}, \perp^{2r}, P_{\mathrm{Corr}})$;
$\mathit{FullPA} \leftarrow (\mathit{FullPA}, \mathit{NewPauliA})$;
$\ell_{PA} \leftarrow \ell_{PA} + 6r$;
$\widetilde{\mathit{NewPauliB}} \leftarrow \left(\perp^{2r}, \perp^{2r}, \widetilde{P_{\mathrm{Corr}}}\right)$;
$\widetilde{PB} \leftarrow \left(\widetilde{PB}, \widetilde{\mathit{NewPauliB}}\right)$;
$\ell_{\widetilde{PB}} \leftarrow \ell_{\widetilde{PB}} + 6r$;
$\ell_{RA} \leftarrow \ell_{RA} + 1$;
return Q-simulate;

4.5 Analysis

To simplify the analysis of the algorithm, without loss of generality, we assume that the error introduced by the adversary on the n' message registers in Π' is an arbitrary Pauli error of weight at most $\epsilon n'$. We prove the correctness of the algorithm for any such error syndrome, which by linearity implies the correctness of the algorithm against any adversary defined in Section 2.2.2. In order to track the simulation progress and show the correctness of the algorithm, we condition on some view of the local classical data recorded by Alice and Bob.

Similar to Section 3.5, the analysis of Algorithm 14 is in terms of potential functions which measure the correctness of the two players' views of what has happened so far in the simulation and quantify the progress in reproducing the joint state of the input protocol. We recall the following definitions from Section 3.5:

$$md_+^{\mathrm{A}} \stackrel{\mathrm{def}}{=} \text{ the length of the longest prefix where } MA \text{ and } \widetilde{MA} \text{ agree;} \tag{4.5}$$

$$md_+^{\mathrm{B}} \stackrel{\mathrm{def}}{=} \text{ the length of the longest prefix where } MB \text{ and } \widetilde{MB} \text{ agree;} \tag{4.6}$$

$$md_-^{\mathrm{A}} \stackrel{\mathrm{def}}{=} \max\{\ell_{MA}, \ell_{\widetilde{MA}}\} - md_+^{\mathrm{A}} \ ; \tag{4.7}$$

$$md_-^{\mathrm{B}} \stackrel{\mathrm{def}}{=} \max\{\ell_{MB}, \ell_{\widetilde{MB}}\} - md_+^{\mathrm{B}} \ ; \tag{4.8}$$

$$pd_+^{\mathrm{A}} \stackrel{\mathrm{def}}{=} \lfloor \frac{1}{r} \times \text{ the length of the longest prefix where } PA \text{ and } \widetilde{PA} \text{ agree} \rfloor; \tag{4.9}$$

$$pd_+^{\mathrm{B}} \stackrel{\mathrm{def}}{=} \lfloor \frac{1}{r} \times \text{ the length of the longest prefix where } PB \text{ and } \widetilde{PB} \text{ agree} \rfloor; \tag{4.10}$$

$$pd_-^{\mathrm{A}} \stackrel{\mathrm{def}}{=} \frac{1}{r} \max\{\ell_{PA}, \ell_{\widetilde{PA}}\} - pd_+^{\mathrm{A}} \ ; \tag{4.11}$$

$$pd_-^{\mathrm{B}} \stackrel{\mathrm{def}}{=} \frac{1}{r} \max\{\ell_{PB}, \ell_{\widetilde{PB}}\} - pd_+^{\mathrm{B}} \ . \tag{4.12}$$

We recall the definition of g, b, u from Subsection 4.3.7:

$$g \stackrel{\mathrm{def}}{=} \text{the number of good blocks in } JS2, \tag{4.13}$$

$$b \stackrel{\mathrm{def}}{=} \text{the number of bad blocks in } JS2, \text{ and} \tag{4.14}$$

$$u \stackrel{\mathrm{def}}{=} |\ell_{\mathrm{QVC}}^{\mathrm{A}} - \ell_{\mathrm{QVC}}^{\mathrm{B}}| \ , \tag{4.15}$$

We define

$$\begin{aligned} rd^+ \stackrel{\mathrm{def}}{=} \max\{j : \ & j \leq \min\{\ell_{RA}, \ell_{RB}\} \ , \ RA[1:j] = RB[1:j] \\ & , W_k = 0^{4r} \text{ for all } k \leq j \text{ with } RA[k] = \mathsf{S}\} \ ; \end{aligned} \tag{4.16}$$

$$rd^- \stackrel{\mathrm{def}}{=} \max\{\ell_{RA}, \ell_{RB}\} - rd^+ \ , \tag{4.17}$$

where W in Eq. 4.16 is the string corresponding to the error syndrome defined in Subsection 4.3.5. At the end of the i-th iteration, we let

$$\Phi_{\mathrm{MD}} \stackrel{\mathrm{def}}{=} 2i - md_{+}^{\mathrm{A}} + 3md_{-}^{\mathrm{A}} - md_{+}^{\mathrm{B}} + 3md_{-}^{\mathrm{B}} \ , \tag{4.18}$$

$$\Phi_{\mathrm{RD}} \stackrel{\mathrm{def}}{=} \ell_{\mathrm{QVC}}^{\mathrm{A}} + \ell_{\mathrm{QVC}}^{\mathrm{B}} + 13rd^{-} - 2rd^{+} \ , \tag{4.19}$$

$$\Phi_{\mathrm{PD}} \stackrel{\mathrm{def}}{=} 6q_{MA} + 6q_{MB} - pd_{+}^{\mathrm{A}} + pd_{-}^{\mathrm{A}} - pd_{+}^{\mathrm{B}} + pd_{-}^{\mathrm{B}} \ , \tag{4.20}$$

$$\Phi_{\mathrm{Q}} \stackrel{\mathrm{def}}{=} g - b - 9u \ , \tag{4.21}$$

$$\Phi \stackrel{\mathrm{def}}{=} \Phi_{\mathrm{Q}} - \Phi_{\mathrm{MD}} - \Phi_{\mathrm{RD}} - \Phi_{\mathrm{PD}} \ . \tag{4.22}$$

The following lemma states an important property of potential functions Φ_{MD}, Φ_{RD} and Φ_{PD} defined above which we use in the analysis of the algorithm.

Lemma 4.5.1. *Throughout the algorithm, it holds that*

- $\Phi_{\mathrm{MD}} \geq 0$ *with equality if and only if Alice and Bob have full knowledge of each other's metadata, i.e.,* $md_{+}^{\mathrm{A}} = md_{+}^{\mathrm{B}} = i$ *and* $md_{-}^{\mathrm{A}} = md_{-}^{\mathrm{B}} = 0$.
- $\Phi_{\mathrm{RD}} \geq 0$ *with equality if and only if Alice and Bob have used the same number of MES blocks* ($\ell_{\mathrm{QVC}}^{\mathrm{A}} = \ell_{\mathrm{QVC}}^{\mathrm{B}}$), *their measurement pointers* ℓ_{RA} *and* ℓ_{RB} *agree and are equal to* $\ell_{\mathrm{QVC}}^{\mathrm{A}}$, *they fully agree on the recycling data* ($RA = RB$) *and* $W_k = 0^{4r}$ *for all* $k \leq \ell_{\mathrm{QVC}}^{\mathrm{A}}$ *with* $RA[k] = \mathsf{S}$, *i.e.,* $\ell_{\mathrm{QVC}}^{\mathrm{A}} = \ell_{\mathrm{QVC}}^{\mathrm{B}} = rd^{+}$ *and* $rd^{-} = 0$.
- $\Phi_{\mathrm{PD}} \geq 0$ *with equality if and only if Alice and Bob have full knowledge of each other's Pauli data, i.e.,* $pd_{+}^{\mathrm{A}} = 6q_{MA}$, $pd_{+}^{\mathrm{B}} = 6q_{MB}$ *and* $pd_{-}^{\mathrm{A}} = pd_{-}^{\mathrm{B}} = 0$.

Proof. The first statement follows from the property that $md_{-}^{\mathrm{A}}, md_{-}^{\mathrm{B}} \geq 0$ and $md_{+}^{\mathrm{A}}, md_{+}^{\mathrm{B}} \leq i$. The second statement holds since $rd^{-} \geq 0$, $rd^{+} \leq \min\{\ell_{RA}, \ell_{RB}\}$ and the property that $\ell_{RA} \leq \ell_{\mathrm{QVC}}^{\mathrm{A}}$ and $\ell_{RB} \leq \ell_{\mathrm{QVC}}^{\mathrm{B}}$. The third statement follows since $pd_{-}^{\mathrm{A}}, pd_{-}^{\mathrm{B}} \geq 0$, $pd_{+}^{\mathrm{A}} \leq 6q_{MA}$, and $pd_{+}^{\mathrm{B}} \leq 6q_{MB}$. □

In order to avoid ambiguity, whenever necessary we use a superscript i to indicate the value of the variables of the algorithm at the end of the i-th iteration. For instance, we denote Alice's recycling data at the end of the i-th iteration by RA^i. Before presenting the analysis of Algorithm 14, we formally define successful recycling.

Definition 4.5.2. We say recycling is successful in the i-th iteration of Algorithm 14 if the following hold:

- The algorithm does not abort in the i-th iteration, i.e., $NextIndexA^i, NextIndexB^i \neq \perp$,
- $NextIndexA^i = NextIndexB^i$,

- The block of MES registers indexed by $NextIndexA^i$ are in the $|\phi^{0,0}\rangle^{\otimes 4r}$ state at the beginning of the i-th iteration.

Note that the conditions of Definition 4.5.2 are all satisfied in the first L_{QVC} iterations of the algorithm as well. Note that if recycling is successful in the first i iterations of the algorithm then $\ell^j_{\mathrm{RecycleA}} = \ell^j_{\mathrm{RecycleB}}$, for all $j \leq i$ and we have $IndexA^i = IndexB^i$. Moreover, for every $i \geq L_{\mathrm{QVC}}$, we have $IndexA^i\,[1 : L_{\mathrm{QVC}}] = IndexB^i\,[1 : L_{\mathrm{QVC}}] = 1 : L_{\mathrm{QVC}}$.

Proof Outline of Theorem 4.2.1. In order to prove successful simulation of an n-round protocol, it suffices to show that $\Phi \geq n/2r$, at the end of the simulation. In Section 3.5 we showed that in the teleportation-based protocol, except with exponentially small probability, the total number of hash collisions is $\mathrm{O}(n\epsilon)$. Then, for sufficiently large number of iterations, to prove the correctness it was sufficient to show that in any iteration with no error or hash collision the potential function increases by at least one, while any iteration with errors or hash collisions decreases the potential by at most some fixed constant. However, this statement is not necessarily true for Algorithm 14 if the recycling of MESs has not been successful in an earlier iteration. In fact, the potential function is defined in terms of $JS2$ which is a valid representation of the joint state at any stage in the simulation only if recycling has been successful so far. Therefore, to use such an argument, one needs to prove successful recycling first. On the other hand, to prove successful recycling in an iteration, we need to bound the number of iterations with a hash collision, as well as the number of iterations dedicated to "recovery" from hash collisions and transmission errors. Therefore, the analysis of the recycling-based protocol involves an inductive argument.

The analysis in this section involves constants

$$c_1 < c_2 < c_3 < c_4 < c_5 < c_6 < c_7 < c_8 < c_9 \ ,$$

chosen such that c_i is sufficiently large depending only on c_j with $j < i$.

Definition 4.5.3. We say an iteration of Algorithm 14 suffers from a *metadata hash collision* when $H_{MA} = H_{\widetilde{MA}}$ despite the fact that $MA \neq \widetilde{MA}$, or $H_{MB} = H_{\widetilde{MB}}$ despite the fact that $MB \neq \widetilde{MB}$. Note that we distinguish between the above scenario and when, for instance, $H_{MA} = H'_{\widetilde{MA}}$ due to a transmission error on $H_{\widetilde{MA}}$, despite the two might have similar effects.

The following lemma bounds the number of iterations with a metadata hash collision.

Lemma 4.5.4. *The number of iterations of Algorithm 14 suffering from a metadata hash collision is at most $c_1 n\epsilon$ with probability at least $1 - 2^{-\Theta(n\epsilon)}$.*

Proof. We call an iteration a *dangerous iteration of type I* if $md^{\mathrm{A}}_{-} + md^{\mathrm{B}}_{-} \neq 0$, at the beginning of the iteration. Note that metadata hash collisions can only occur in type I dangerous iterations. Let d_{I} denote the number of such iterations. It suffices to prove that

$$\Pr\left(d_{\mathrm{I}} > c_1 n\epsilon\right) \leq 2^{-\Theta(n\epsilon)} \ .$$

Note that in any iteration $md_-^{\mathrm{A}} + md_-^{\mathrm{B}}$ increases by at most 6. Moreover, in an iteration with $md_-^{\mathrm{A}} + md_-^{\mathrm{B}} \neq 0$, if $md_-^{\mathrm{A}} + md_-^{\mathrm{B}}$ decreases, it decreases by at least 1. Therefore, in at least $d_{\mathrm{I}}/7$ iterations, $md_-^{\mathrm{A}} + md_-^{\mathrm{B}}$ increases or remains unchanged at a nonzero value. Note that $md_-^{\mathrm{A}} + md_-^{\mathrm{B}} > 0$ increases or remains unchanged only if a transmission error or a metadata hash collision occurs. Moreover, when $md_-^{\mathrm{A}} + md_-^{\mathrm{B}}$ increases from zero in an iteration, it is due to a transmission error. The number of iterations is less than $2n$. So the total number of iterations with transmission errors is at most $2n\epsilon$. This implies that in all the remaining iterations, i.e., at least $d_{\mathrm{I}}/7 - 2n\epsilon$ iterations a metadata hash collision occurs. Since the algorithm uses independent seeds in each iteration and the probability of collision is chosen to be 0.1, the expected number of collisions is at most $d_{\mathrm{I}}/10$. If $d_{\mathrm{I}} > c_1 n\epsilon$ for a sufficiently large c_1, then the Chernoff bound implies that the probability of having so many collisions is at most $2^{-\Theta(n\epsilon)}$. □

Definition 4.5.5. We refer to an iteration of Algorithm 14 as a *recovery iteration of type I* if at least one of Alice or Bob conducts one of the cases i.A, i.B, ii.A, or ii.B.

We use the following lemma to bound the number of type I recovery iterations.

Lemma 4.5.6. *Suppose that in the first i iterations of Algorithm 14, the number of iterations suffering from a metadata hash collision is at most $c_1 n\epsilon$. Then the number of type I recovery iterations in the first i iterations is at most $c_2 n\epsilon$.*

Proof. Let

$$\Phi_{\mathrm{I}} \stackrel{\mathrm{def}}{=} u + \Phi_{\mathrm{MD}} \ .$$

By Lemma 4.5.1 and the definition of u in Eq. (4.15), Φ_{I} is always non-negative and is equal to zero if and only if Alice and Bob know each other's full metadata and have used the same number of MES blocks for QVC.

Note that if $\Phi_{\mathrm{I}} = 0$ at the beginning of an iteration, then the iteration is a type I recovery iteration only if a transmission error in communication of metadata messages occurs. The total number of such iterations is at most $2n\epsilon$.

Let β_{I} denote the number of iterations in the first i iterations starting with $\Phi_{\mathrm{I}} > 0$. Note that in any iteration, Φ_{I} increases or remains unchanged at a nonzero value only if a metadata hash collision or a transmission error occurs. In each iteration, regardless of the number of errors and collisions, Φ_{I} increases by at most 23. Moreover, if Φ_{I} decreases, it decreases by at least 1. Assuming the number of metadata hash collisions is at most $c_1 n\epsilon$, this implies that the number of iterations in which Φ_{I} decreases is at most $23\,(c_1 + 2)\,n\epsilon$. So we have $\beta_{\mathrm{I}} \leq 24\,(c_1 + 2)\,n\epsilon$.

Therefore, the total number of type I recovery iterations is at most $c_2 n\epsilon$, where $c_2 \stackrel{\mathrm{def}}{=} 24\,(c_1 + 2) + 2$. □

Definition 4.5.7. We say an iteration of Algorithm 14 suffers from a *quantum hash collision* when recycling has been successful so far, Alice and Bob know each other's

metadata, have used the same number of MES blocks ($\ell^{\mathrm{A}}_{\mathrm{QVC}} = \ell^{\mathrm{B}}_{\mathrm{QVC}}$) and agree on their measurement pointers and their recycling data up to the measurement pointers ($\ell_{RA} = \ell_{RB}$ and $RA\,[1 : \ell_{RA}] = RB\,[1 : \ell_{RB}]$) but despite the fact that there is an undetected quantum error from earlier iterations ($W_k \neq 0^{4r}$ for some $k \leq \ell_{RA}$ with $RA\,[k] = \mathsf{S}$), their quantum hash values match, i.e., $QHA = QHB$. Note that we distinguish between the above scenario and when $QHA = QHB'$ due to a transmission error on QHB.

We use the following lemma to bound the number of iterations suffering from a quantum hash collision.

Lemma 4.5.8. *Suppose that recycling is successful in the first i iterations of Algorithm 14 and the number of iterations suffering from a metadata hash collision is at most $c_1 n\epsilon$. Then the number of iterations suffering from a quantum hash collision in the first i iterations is at most $c_3 n\epsilon$ with probability at least $1 - 2^{-\Theta(n\epsilon)}$.*

Proof. We call an iteration a *dangerous iteration of type II* if $rd^- \neq 0$, at the beginning of the iteration. Note that quantum hash collisions can only occur in type II dangerous iterations. Let d_{II} denote the number of such iterations in the first i iterations of Algorithm 14. It suffices to prove that

$$\Pr\left(d_{\mathrm{II}} > c_3 n\epsilon\right) \leq 2^{-\Theta(n\epsilon)} \ .$$

Note that in any iteration rd^- increases by at most 2. Moreover, in an iteration with $rd^- \neq 0$, if rd^- decreases, it decreases by at least 1. Therefore, in at least $d_{\mathrm{II}}/3$ iterations, rd^- increases or remains unchanged at a nonzero value. Note that, assuming successful recycling in the previous iterations, $rd^- > 0$ increases or remains unchanged in an iteration only if

- A metadata hash collision or a transmission error on metadata messages (i.e., $H_{k_{MA}}$, H_{MA_1}, H_{MA_2}, $H_{k_{\widetilde{MA}}}$, $H_{\widetilde{MA}_1}$, $H_{\widetilde{MA}_2}$, $H_{k_{MB}}$, H_{MB_1}, H_{MB_2}, $H_{k_{\widetilde{MB}}}$, $H_{\widetilde{MB}_1}$, $H_{\widetilde{MB}_2}$) occurs, or else,
- The iteration is a type I recovery iteration. Alice and Bob are still reconciling an earlier inconsistency in their metadata and they are both in case i.A or case i.B, or one of them is in case ii.A and the other one in case ii.B. Else,
- A transmission error on quantum hash values or a quantum hash collision occurs. At least one party does not realize that $rd^- > 0$ and conducts one of the cases v, vi.A, vi.B, or vii.

Moreover, assuming successful recycling in the previous iterations, the value of rd^- increases from zero in an iteration only if

- A metadata hash collision occurs and Alice and Bob act based on incorrect estimates of each other's recycling data, or else,

- A transmission error on metadata messages occurs, or else,
- A transmission error on quantum hash values occurs and only one party conducts case iv, or else,
- A transmission error on the Pauli data messages (i.e., $H_{k_{PA}}$, H_{PA_1}, H_{PA_2}, $H_{k_{\widetilde{PA}}}$, $H_{\widetilde{PA}_1}$, $H_{\widetilde{PA}_2}$, $H_{k_{PB}}$, H_{PB_1}, H_{PB_2}, $H_{k_{\widetilde{PB}}}$, $H_{\widetilde{PB}_1}$, $H_{\widetilde{PB}_2}$) occurs and one party conducts case vi.A or vi.B while the other is in case vii. Else,
- A transmission error occurs on the communicated QVC messages when both parties conduct case vii.

Assuming the number of metadata hash collisions is at most $c_1 n\epsilon$, by Lemma 4.5.6, the total number of type I recovery iterations is at most $c_2 n\epsilon$. The total number of transmission errors is at most $2n\epsilon$. Therefore, in at least $d_{\mathrm{II}}/3 - (c_1 + c_2 + 2)\, n\epsilon$ iterations a quantum hash collision occurs.

The shared random string used as the classical seed for quantum hashing is δ-biased with $\delta = 2^{-\Theta\left(n\sqrt{\epsilon}\right)}$. By Lemma 2.4.8, the seeds are also $\delta^{\Theta(1)}$-statistically close to being $\Theta\left(R_{\mathrm{total}}\right)$-wise independent. Therefore, all hashing steps are statistically close to being fully independent. Combined with Lemma 4.3.2, this implies that the expected number of quantum hash collisions is at most $10^{-3} d_{\mathrm{II}}$. For sufficiently large c_3, if $d_{\mathrm{II}} > c_3 n\epsilon$, the Chernoff bound implies that the probability of having at least $d_{\mathrm{II}}/3 - (c_1 + c_2 + 2)\, n\epsilon$ quantum hash collisions is at most $2^{-\Theta(n\epsilon)}$. □

Definition 4.5.9. We refer to an iteration of Algorithm 14 as a *recovery iteration of type II* if it is not a type I recovery iteration and at least one of Alice or Bob conducts one of the cases iii, iv, or v.

We use the following lemma to bound the number of type II recovery iterations.

Lemma 4.5.10. *Suppose that in the first i iterations of Algorithm 14, recycling is successful, the number of iterations suffering from a metadata hash collision is at most $c_1 n\epsilon$ and the number of iterations suffering from a quantum hash collision is at most $c_3 n\epsilon$. Then the total number of type II recovery iterations in the first i iterations is at most $c_4 n\epsilon$.*

Proof. Note that by Lemma 4.5.1, Φ_{RD} is always non-negative. If at the beginning of an iteration $\Phi_{\mathrm{RD}} = 0$, then the iteration is a type II recovery iteration only if

- $\Phi_{\mathrm{MD}} > 0$ but due to a metadata hash collision or a transmission error on metadata messages both Alice and Bob do not realize that. In this case they compute their estimates of each other's recycling data based on incorrect estimates of each other's metadata.
- $\Phi_{\mathrm{MD}} = 0$ but a transmission error in communication of quantum hashes occurs.

Therefore, in the first i iterations the total number of type II recovery iterations starting with $\Phi_{\mathrm{RD}} = 0$ is at most $(c_1 + 2)\, n\epsilon$.
Let β_{RD} denote the number of iterations starting with $\Phi_{\mathrm{RD}} > 0$ in the first i iterations. Assuming successful recycling in the preceding iterations, $\Phi_{\mathrm{RD}} > 0$ increases or remains unchanged in an iteration only if

- The iteration is a type I recovery iteration, or else,
- $\Phi_{\mathrm{MD}} > 0$ but due to a metadata hash collision or transmission errors on metadata messages, Alice and Bob don't realize that and act based on their incorrect estimates of each other's recycling data.
- $\Phi_{\mathrm{MD}} = 0$, i.e., Alice and Bob have correct estimates of each other's recycling data but a quantum hash collision or a transmission error on quantum hash values occurs.

Moreover, Φ_{RD} increases from zero in an iteration only if

- A transmission error occurs, or else,
- A metadata hash collision occurs and the two parties act based on incorrect estimates of each other's recycling data.

Therefore, the number of iterations in the first i iterations with Φ_{RD} increasing or remaining unchanged at a nonzero value is at most $(c_1 + c_2 + c_3 + 2)\, n\epsilon$. Note that in each iteration, regardless of the number of errors and collisions, Φ_{RD} increases by at most 30. Moreover, if Φ_{RD} decreases, it decreases by at least 1. This implies that the number of iterations in which Φ_{RD} decreases is at most $30\,(c_1 + c_2 + c_3 + 2)\, n\epsilon$. So, we have $\beta_{\mathrm{RD}} \leq 31\,(c_1 + c_2 + c_3 + 2)\, n\epsilon$.
Therefore, the total number of recovery iterations of type II in the first i iterations is at most $c_4 n\epsilon$, where $c_4 \stackrel{\mathrm{def}}{=} 31\,(c_1 + c_2 + c_3 + 2) + (c_1 + 2)$. □

Definition 4.5.11. We say an iteration of Algorithm 14 suffers from a *Pauli data hash collision* when recycling has been successful so far, Alice and Bob know each other's metadata, agree on the number of MES blocks they have used ($\ell^{\mathrm{A}}_{\mathrm{QVC}} = \ell^{\mathrm{B}}_{\mathrm{QVC}}$), agree on their recycling data, their measurement pointers satisfy $\ell_{RA} = \ell_{RB} = \ell^{\mathrm{A}}_{\mathrm{QVC}}$, all the non-measured MES blocks are in the $|\phi^{0,0}\rangle^{\otimes 4r}$ state and $H_{PA} = H_{\widetilde{PA}}$ despite the fact that $PA \neq \widetilde{PA}$ or $H_{PB} = H_{\widetilde{PB}}$ despite the fact that $PB \neq \widetilde{PB}$. Note that we distinguish between the above scenario and when for instance $H_{PA} = H'_{\widetilde{PA}}$ due to a transmission error on $H_{\widetilde{PA}}$.

We use the following lemma to bound the number of iterations suffering from a Pauli data hash collision.

Lemma 4.5.12. *Suppose that in the first i iterations of Algorithm 14, recycling is successful, the number of iterations suffering from a metadata hash collision is at most $c_1 n\epsilon$ and the number of iterations suffering from a quantum hash collision is at most $c_3 n\epsilon$. Then the number of iterations suffering from a Pauli data hash collision in the first i iterations is at most $c_5 n\epsilon$ with probability at least $1 - 2^{-\Theta(n\epsilon)}$.*

Proof. We call an iteration a *dangerous iteration of type III* if $pd_-^{\mathrm{A}} + pd_-^{\mathrm{B}} \neq 0$, at the beginning of the iteration. Note that Pauli data hash collisions can only occur in type III dangerous iterations. Let d_{III} denote the number of such iterations in the first i iterations of Algorithm 14. We prove that

$$\Pr\left(d_{\mathrm{III}} > c_5 n\epsilon\right) \leq 2^{-\Theta(n\epsilon)} \ .$$

Note that in any iteration $pd_-^{\mathrm{A}} + pd_-^{\mathrm{B}}$ increases by at most 8. Moreover, in an iteration with $pd_-^{\mathrm{A}} + pd_-^{\mathrm{B}} \neq 0$, if $pd_-^{\mathrm{A}} + pd_-^{\mathrm{B}}$ decreases, it decreases by at least 1. Therefore, in at least $d_{\mathrm{III}}/9$ iterations, $pd_-^{\mathrm{A}} + pd_-^{\mathrm{B}}$ increases or remains unchanged at a nonzero value. Note that when $pd_-^{\mathrm{A}} + pd_-^{\mathrm{B}} > 0$ increases or remains unchanged in an iteration, it is due to one of the following reasons:

- The iteration is a type I recovery iteration, or else,
- The iteration is a type II recovery iteration, or else,
- A Pauli data hash collision or a transmission error on Pauli data messages (i.e., H_{PA}, ℓ_{PA}, H_{PB}, ℓ_{PB}, $H_{\widetilde{PB}}$, $\ell_{\widetilde{PB}}$, $H_{\widetilde{PA}}$, $\ell_{\widetilde{PA}}$) occurs.

Moreover, $pd_-^{\mathrm{A}} + pd_-^{\mathrm{B}}$ increases from zero in an iteration only if

- A metadata hash collision or transmission error on the metadata messages occurs, or else,
- A transmission error on the quantum hash values occurs, or else,
- A transmission error on the Pauli data messages occurs, or else,
- Both parties conduct case vi.B and due to a transmission error, at least one of Alice or Bob extends her/his estimate of the other party's Pauli data incorrectly.

Assuming the number of iterations of Algorithm 14 suffering from a metadata hash collision is at most $c_1 n\epsilon$ and the number of iterations suffering from a quantum hash collision is at most $c_3 n\epsilon$, the total number of type I and type II recovery iterations is at most $(c_2 + c_4)\, n\epsilon$. The number of transmission errors is at most $2n\epsilon$. Therefore, in at least $d_{\mathrm{III}}/9 - (c_1 + c_2 + c_4 + 2)\, n\epsilon$ iterations a Pauli data hash collision occurs.

Since the algorithm uses independent seeds in each iteration and the probability of a collision is chosen to be 0.1, the expected number of Pauli data hash collisions is at most $d_{\mathrm{III}}/10$. For sufficiently large c_5, if $d_{\mathrm{III}} > c_5 n\epsilon$, the Chernoff bound implies that the probability of having so many Pauli data hash collisions in the first i iterations is at most $2^{-\Theta(n\epsilon)}$. □

Definition 4.5.13. We refer to an iteration of Algorithm 14 as a *recovery iteration of type III* if it is not a type I or type II recovery iteration and at least one of Alice or Bob conducts one of the cases vi.A or vi.B.

We use the following lemma to bound the number of type III recovery iterations.

Lemma 4.5.14. *Suppose that in the first i iterations of Algorithm 14, recycling is successful and the number of iterations suffering from metadata , quantum and Pauli data hash collisions is at most $c_1 n\epsilon$, $c_3 n\epsilon$ and $c_5 n\epsilon$, respectively. Then the total number of type III recovery iterations in the first i iterations is at most $c_6 n\epsilon$.*

Proof. Note that by Lemma 4.5.1, Φ_{PD} is always non-negative and it is equal to zero if and only if Alice and Bob have full knowledge of each other's Pauli data. If at the beginning of an iteration $\Phi_{\text{PD}} = 0$, then the iteration is a type III recovery iteration only if

- A transmission error occurs on the Pauli data messages, or else,
- A metadata hash collision or a transmission error on metadata messages occurs. In this case at least one party incorrectly believes that his/her estimate of the other party's Pauli data is not full-length.

Therefore, in the first i iterations the total number of type III recovery iterations starting with $\Phi_{\text{PD}} = 0$ is at most $(c_1 + 2)\, n\epsilon$.

Let β_{PD} denote the number of iterations starting with $\Phi_{\text{PD}} > 0$ in the first i iterations. Note that in any iterations $\Phi_{\text{PD}} > 0$ increases or remains unchanged only if

- The iteration is a type I recovery iteration, or else,
- The iteration is a type II recovery iteration, or else,
- A Pauli data hash collision or a transmission error on Pauli data messages occurs.

Moreover, Φ_{PD} increases from zero in an iteration only if

- A metadata hash collision or transmission error on the metadata messages occurs, or else,
- The iteration is a type I recovery iteration in which one party conducts case ii.A and the other case ii.B, or else,
- A transmission error on the quantum hash values occurs, or else,
- The iteration is a type II recovery iteration in which both parties conduct case iii or both conduct case iv, or else,

- A transmission error on the Pauli data messages occurs.

Therefore, the number of iterations in the first i iterations with Φ_{PD} increasing or remaining unchanged at a nonzero value is at most $(c_1 + c_2 + c_4 + c_5 + 2)\, n\epsilon$. Note that in each iteration, regardless of the number of errors and collisions, Φ_{PD} increases by at most 22. Moreover, if Φ_{PD} decreases, it decreases by at least 1. This implies that the number of iterations in which Φ_{PD} decreases is at most $22\,(c_1 + c_2 + c_4 + c_5 + 2)\, n\epsilon$. So, we have $\beta_{\mathrm{PD}} \leq 23\,(c_1 + c_2 + c_4 + c_5 + 2)\, n\epsilon$.

Therefore, the total number of recovery iterations of type III in the first i iterations is at most $c_6 n\epsilon$, where $c_6 \stackrel{\mathrm{def}}{=} 23\,(c_1 + c_2 + c_4 + c_5 + 2) + (c_1 + 2)$. □

Let ω_i denote the number of iterations suffering from a transmission error or a hash collision in the first i iterations, plus the number of recovery iterations of type I, II, or III, in the first i iterations. Note that the bounds and probabilities in Lemmas 4.5.4–4.5.14 are all independent of the iteration number i. As a corollary we have:

Corollary 4.5.15. *There exist $q = 2^{-\Theta(n\epsilon)}$ and a constant c_7 such that, for every $i \in [R_{\mathrm{total}}]$, assuming successful recycling in the first i iterations of Algorithm 14, except with probability at most q, we have $\omega_i \leq c_7 n\epsilon$.*

The following lemma is the last ingredient we need to prove successful recycling in every iteration of the simulation. Recall that RA^i and RB^i denote the recycling data of Alice and Bob, respectively, at the end of the i-th iteration.

Lemma 4.5.16. *Let $t = c_8 n\epsilon$ where $c_8 > 3c_7$. Then for every $i \in [R_{\mathrm{total}}]$ where $i \geq t$, if recycling is successful in the first $i-1$ iterations and $\omega_{i-1} \leq c_7 n\epsilon$, then:*

1. *$RA^i[1 : i-t] = RB^i[1 : i-t]$, i.e., at the end of iteration i, the recycling data of Alice and Bob agree in a prefix of length at least $i - t$.*
2. *$RA^i[1 : i-t] = RA^{i+1}[1 : i-t]$, i.e., the prefix of RA of length $i - t$ does not get modified in the next iteration. The same statement holds for RB.*
3. *For every $k \in [i-t]$ such that $RA^i[k] = RB^i[k] = \mathsf{S}$, we have $W_k = 0^{4r}$.*

Proof. Part 1:

Toward contradiction, suppose that there exists $t' \in [t, i-1]$ such that $RA^i[i-t'] \neq RB^i[i-t']$. Without loss of generality, assume that $RA^i[i-t'] = \mathsf{M}$ and $RB^i[i-t'] = \mathsf{S}$. Suppose that the last time Alice's measurement pointer ℓ_{RA} was equal to $i - t'$ was t_2 iterations earlier, i.e., iteration $i - t_2$. In that iteration, ℓ_{RA} has distance $t_1 \stackrel{\mathrm{def}}{=} t' - t_2$ from $i - t_2$, the iteration number. Note that the distance between the iteration number and ℓ_{RA} increases only in (some) recovery iterations and it increases by at most 2: the distance remains the same if Alice is in case v or case vii. Otherwise, it increases by 1 if ℓ_{RA} does not

move and increases by 2 when it moves back. This implies that in the first $i - t_2$ iterations, there have been at least $t_1/2$ recovery iterations. In the t_2 iterations after that, there is an inconsistency in the recycling data which does not get resolved. In any of these iterations one of the following holds:

- A metadata hash collision or a transmission error on metadata messages occurs, or else,
- The iteration is a type I recovery iteration and Alice and Bob are still trying to reconcile an inconsistency in their metadata, or else,
- The iteration is a type II recovery iteration. In this case, no metadata transmission error or collision occurs and the iteration is not a type I recovery iteration. So Alice and Bob know each other's recycling data and aware of the inconsistency, they are trying to resolve it.

Therefore, in the first $i - 1$ iterations, the number of recovery iterations plus the number of iterations suffering from a transmission error or a collision is at least

$$t_1/2 + t_2 - 1 \geq t'/2 - 1 \geq t/2 - 1 \geq \frac{c_8}{2} n\epsilon - 1 \ ,$$

contradicting $\omega_{i-1} \leq c_7 n\epsilon$. Note that here we implicitly use the reasonable assumption that $n\epsilon$ is at least a constant.

Part 2:

Suppose that recycling is successful in the first $i - 1$ iterations and $\omega_{i-1} \leq c_7 n\epsilon$. Note that by the same argument as in part 1, at the end of iteration $i + 1$, the difference between the measurement pointers and the iteration number is at most $2\omega_{i-1} + 4 \leq 2c_7 n\epsilon + 4 \leq t$. Therefore, the prefixes of both RA and RB of length $i - t$ do not get modified in the next iteration.

Part 3:

Note that the difference between the iteration number and $\ell^{\mathrm{A}}_{\mathrm{QVC}}$ increases only in (some) recovery iterations and it increases by at most 1: The difference remains the same if Alice is in case ii.B or case vii. Otherwise, $\ell^{\mathrm{A}}_{\mathrm{QVC}}$ remains unchanged and the distance increases by 1. The number of recovery iterations in the first i iterations is at most $\omega_{i-1} + 1 < t$. Therefore, at the end of the i-th iteration we have $\min\{\ell^{\mathrm{A}}_{\mathrm{QVC}}, \ell^{\mathrm{B}}_{\mathrm{QVC}}\} > i - t$. Toward contradiction, suppose that there exists $t' \in [t, i-1]$ such that $RA^i[i - t'] = RB^i[i - t'] = \mathsf{S}$ and $W_{i-t'} \neq 0^{4r}$. This is due to one of the following scenarios:

- The corresponding block of MES registers has been used by both parties for communication using QVC, but in different iterations (out-of-sync QVC).
- It has been used in the same iteration by both parties for communication using QVC but transmission errors have occurred on the messages.

In any case, suppose that the last time one of the pointers $\ell^{\mathrm{A}}_{\mathrm{QVC}}$ or $\ell^{\mathrm{B}}_{\mathrm{QVC}}$ was equal to $i-t'$ was t_2 iterations earlier, i.e., iteration $i-t_2$ and without loss of generality, suppose that $\ell^{\mathrm{A}}_{\mathrm{QVC}}$ is that pointer. In iteration $i-t_2$, the pointer $\ell^{\mathrm{A}}_{\mathrm{QVC}}$ has distance $t_1 \stackrel{\mathrm{def}}{=} t'-t_2$ from $i-t_2$, the iteration number. This implies that in the first $i-t_2$ iterations, there have been at least t_1 recovery iterations. In the t_2 iterations after that, the block of MES registers indexed by $\mathit{IndexA}\left[i-t'\right]$ are not measured by any of the parties. In any of these iterations one of the following holds:

- A metadata hash collision or a transmission error on metadata messages occurs, or else,
- The iteration is a type I recovery iteration and Alice and Bob are still trying to reconcile an inconsistency in their metadata, or else,
- A quantum hash collision or a transmission error on quantum hash values occurs, or else,
- The iteration is a type II recovery iteration. In this case, no metadata transmission error or collision occurs and the iteration is not a type I recovery iteration. So Alice and Bob know each other's recycling data. So Alice and Bob are both be in case iii or both in case iv.

The above argument implies that in the first $i-1$ iterations, the number of recovery iterations plus the number of iterations suffering from a transmission error or a collision is at least

$$t_1+t_2-1=t'-1\geq t-1\geq c_8 n\epsilon-1 \ ,$$

contradicting $\omega_{i-1}\leq c_7 n\epsilon$. □

We are now ready to prove that except with exponentially small probability recycling is successful in every iteration of the algorithm. Recall that we denote Alice's recycling pointer at the end of iteration i by $\ell^i_{\mathrm{RecycleA}}$. We use m^{A}_i to denote the number of M symbols in $RA^i\left[1:\ell^i_{\mathrm{RecycleA}}\right]$. Similarly, $\ell^i_{\mathrm{RecycleB}}$ denotes Bob's recycling pointer at the end of iteration i and the number of M symbols in $RB^i\left[1:\ell^i_{\mathrm{RecycleB}}\right]$ is denoted by m^{B}_i.

Lemma 4.5.17. *Let $L_{\mathrm{QVC}}=c_9 n\epsilon$, where $c_9>c_7+c_8$. Then with probability at least $1-2^{-\Theta(n\epsilon)}$, recycling is successful throughout the execution of Algorithm 14.*

Proof. The proof is based on induction on the iteration number i. Note that recycling starts from iteration $L_{\mathrm{QVC}}+1$.

Base case ($i=L_{\mathrm{QVC}}+1$): Note that the conditions of Definition 4.5.2 are satisfied in the first L_{QVC} iterations of the algorithm and we have

$$\mathit{IndexA}\left[1:L_{\mathrm{QVC}}\right]=\mathit{IndexB}\left[1:L_{\mathrm{QVC}}\right]=1:L_{\mathrm{QVC}} \ .$$

Therefore, by Corollary 4.5.15, except with probability at most $q = 2^{-\Theta(n\epsilon)}$, we have $\omega_{L_{\mathrm{QVC}}} \leq c_7 n\epsilon$. Assuming $\omega_{L_{\mathrm{QVC}}} \leq c_7 n\epsilon$, by Lemma 4.5.16,

$$RA^{L_{\mathrm{QVC}}+1}\left[1 : L_{\mathrm{QVC}} + 1 - t\right] = RB^{L_{\mathrm{QVC}}+1}\left[1 : L_{\mathrm{QVC}} + 1 - t\right] \ ,$$

and for every $k \in [L_{\mathrm{QVC}} + 1 - t]$ such that $RA^{L_{\mathrm{QVC}}+1}[k] = RB^{L_{\mathrm{QVC}}+1}[k] = \mathsf{S}$, we have $W_k = 0^{4r}$. Note that the number of M symbols in $RA^{L_{\mathrm{QVC}}+1}\left[1 : L_{\mathrm{QVC}} + 1 - t\right]$ and $RB^{L_{\mathrm{QVC}}+1}\left[1 : L_{\mathrm{QVC}} + 1 - t\right]$ is at most the number of type I and type II recovery iterations so far, hence at most

$$\omega_{L_{\mathrm{QVC}}+1} \leq \omega_{L_{\mathrm{QVC}}} + 1 \leq c_7 n\epsilon + 1 < L_{\mathrm{QVC}} + 1 - t \ .$$

As shown in Part 3 of Lemma 4.5.16, at the end of iteration L_{QVC}, we have

$$\min\left\{\ell^{\mathrm{A}}_{\mathrm{QVC}}, \ell^{\mathrm{B}}_{\mathrm{QVC}}\right\} > L_{\mathrm{QVC}} - t \ .$$

Therefore, after running the $\mathsf{Recycle}$ subroutine, the algorithm does not abort and at the end of iteration $L_{\mathrm{QVC}} + 1$, the recycling pointers are equal, i.e., $\ell^{L_{\mathrm{QVC}}+1}_{\mathrm{RecycleA}} = \ell^{L_{\mathrm{QVC}}+1}_{\mathrm{RecycleB}}$. Together with the fact that $IndexA\left[1 : L_{\mathrm{QVC}}\right] = IndexB\left[1 : L_{\mathrm{QVC}}\right]$, this implies that the conditions of Definition 4.5.2 are satisfied. Therefore, there exists an event $\mathcal{E}$ of probability at most $q = 2^{-\Theta(n\epsilon)}$ such that if $\neg\mathcal{E}$ then,

- recycling is successful in iteration $L_{\mathrm{QVC}} + 1$,
- $\ell^{L_{\mathrm{QVC}}+1}_{\mathrm{RecycleA}} = \ell^{L_{\mathrm{QVC}}+1}_{\mathrm{RecycleB}}$, and
- $\ell^{L_{\mathrm{QVC}}+1}_{\mathrm{RecycleA}} = m^{\mathrm{A}}_{L_{\mathrm{QVC}}+1} + 1$ and $\ell^{L_{\mathrm{QVC}}+1}_{\mathrm{RecycleB}} = m^{\mathrm{B}}_{L_{\mathrm{QVC}}+1} + 1$.

For $L_{\mathrm{QVC}} < i \leq R_{\mathrm{total}}$, let $\mathcal{T}_i$ be the following statement in terms of the iteration number i:

- Recycling is successful in the first i iterations of the algorithm,
- $\ell^{i}_{\mathrm{RecycleA}} = \ell^{i}_{\mathrm{RecycleB}}$, i.e., the recycling pointers are equal at the end of iteration i, and
- $\ell^{i}_{\mathrm{RecycleA}} = m^{\mathrm{A}}_{i} + i - L_{\mathrm{QVC}}$ and $\ell^{i}_{\mathrm{RecycleB}} = m^{\mathrm{B}}_{i} + i - L_{\mathrm{QVC}}$.

Induction hypothesis: For $L_{\mathrm{QVC}} < i \leq R_{\mathrm{total}}$, there exists an event $\mathcal{E}_i$ of probability at most $(i - L_{\mathrm{QVC}}) \cdot q$ such that if $\neg\mathcal{E}_i$ then $\mathcal{T}_i$ holds.

Inductive step: Assuming $\neg\mathcal{E}_i$, by Corollary 4.5.15, except with probability at most $q = 2^{-\Theta(n\epsilon)}$, we have $\omega_i \leq c_7 n\epsilon$. Let $\mathcal{E}'$ be the event that $\omega_i > c_7 n\epsilon$. Note that $\Pr\left(\mathcal{E}'|\neg\mathcal{E}_i\right) \leq q$. Suppose further that $\neg\mathcal{E}'$. Since $\ell^{i}_{\mathrm{RecycleA}} = m^{\mathrm{A}}_{i} + i - L_{\mathrm{QVC}}$, we have

$$i - t - \ell^{i}_{\mathrm{RecycleA}} = L_{\mathrm{QVC}} - t - m^{\mathrm{A}}_{i} \geq L_{\mathrm{QVC}} - t - \omega_i \geq \Omega(n\epsilon) \ .$$

By induction hypothesis, we also have $i - t - \ell^i_{\text{RecycleB}} \geq \Omega(n\epsilon)$. As shown in Part 3 of Lemma 4.5.16, at the end of iteration i, we have $\min\left\{\ell^{\text{A}}_{\text{QVC}}, \ell^{\text{B}}_{\text{QVC}}\right\} > i - t$. By part 1 of Lemma 4.5.16, we have $RA^{i+1}[1 : i + 1 - t] = RB^{i+1}[1 : i + 1 - t]$. Since $\omega_{i-1} \leq \omega_i \leq c_7 n\epsilon$, by part 2 of Lemma 4.5.16, we have $RA^{i+1}[1 : i - t] = RA^i[1 : i - t]$ and $RB^{i+1}[1 : i - t] = RB^i[1 : i - t]$. Therefore, the algorithm does not abort in iteration $i + 1$ and we have $\ell^{i+1}_{\text{RecycleA}} = \ell^{i+1}_{\text{RecycleB}}$. Moreover, $\ell^{i+1}_{\text{RecycleA}} = m^{\text{A}}_{i+1} + (i + 1) - L_{\text{QVC}}$ and $\ell^{i+1}_{\text{RecycleB}} = m^{\text{B}}_{i+1} + (i + 1) - L_{\text{QVC}}$. Note that since recycling is successful in the first i iterations, at the beginning of iteration $i + 1$, we have $IndexA = IndexB$. So $NextIndexA^{i+1} = NextIndexB^{i+1} \neq \perp$, i.e., the first and second conditions of Definition 4.5.2 are satisfied for iteration $i + 1$.

By part 3 of Lemma 4.5.16, for every $k \in [i + 1 - t]$ such that $RA^{i+1}[k] = RB^{i+1}[k] = \mathsf{S}$, we have $W_k = 0^{4r}$. Note that in the strings $IndexA$ and $IndexB$, while each index in $[L_{\text{QVC}}]$ may appear several times before the recycling pointers, it can only appear at most once after these pointers. Therefore, the block of MES registers indexed by $NextIndexA^{i+1}$ is indeed in the $|\phi^{0,0}\rangle^{\otimes 4r}$ state when it is recycled in iteration $i + 1$ and the third condition of Definition 4.5.2 is also satisfied.

For $\mathcal{E}_{i+1} \stackrel{\text{def}}{=} \mathcal{E}_i \vee \mathcal{E}'$, we have

$$\Pr(\mathcal{E}_{i+1}) \leq \Pr(\mathcal{E}_i) + \Pr(\mathcal{E}'|\neg\mathcal{E}_i) \leq (i - L_{\text{QVC}}) \cdot q + q = (i + 1 - L_{\text{QVC}}) \cdot q \ .$$

By the above argument, if $\neg\mathcal{E}_{i+1}$ then $\mathcal{T}_{i+1}$ holds. Note that for $L_{\text{QVC}} < i \leq R_{\text{total}}$, we have

$$(i - L_{\text{QVC}}) \cdot q = 2^{-\Theta(n\epsilon)} \ .$$

□

Lemma 4.5.18. *Assuming successful recycling throughout the execution of Algorithm 14, each iteration with no transmission error or hash collision increases the potential function* Φ *defined in Eq.* (4.22) *by at least* 1.

Proof. Note that in an iteration with no error or hash collision Alice and Bob agree on the iteration type. Moreover, if $Itertype = \text{MD}, \text{RD}$ or PD, they also agree on whether they extend or rewind the data and if $Itertype = \text{MES}$ (Case ii), then exactly one of them is in sub-case A and the other one in sub-case B. We analyze the potential function in each case keeping in mind the hierarchy of the cases; e.g., Case ii or later cases are encountered only if Alice and Bob have full knowledge of each other's metadata. Lemma 4.5.1 guarantees that $\Phi_{\text{MD}} = 0$ on entering Case ii, $\Phi_{\text{MD}} = \Phi_{\text{RD}} = 0$ on entering Case vi and $\Phi_{\text{MD}} = \Phi_{\text{RD}} = \Phi_{\text{PD}} = 0$ on entering Case vii.

- Alice and Bob are in Case i.A:
  - Φ_{RD}, Φ_{PD} and Φ_{Q} stay the same.
  - i increases by 1.

- md_+^{A} and md_+^{B} stay the same.
- None of md_-^{A} and md_-^{B} increases and at least one decreases by 1.

Therefore, Φ_{MD} decreases by at least $3-2=1$ and Φ increases by at least 1.

- Alice and Bob are in Case i.B:
  - Φ_{RD}, Φ_{PD} and Φ_{Q} stay the same.
  - i increases by 1.
  - md_-^{A} and md_-^{B} stay at 0.
  - At least one of ℓ_{MA} or ℓ_{MB} is smaller than $i-1$; If only $\ell_{MA} < i-1$, then md_+^{A} increases by 2, and md_+^{B} by 1. The case where only $\ell_{MB} < i-1$ is similar. If both are smaller than $i-1$, then md_+^{A} and md_+^{B} both increase by 2.

  Therefore, Φ_{MD} decreases by at least $3-2=1$ and Φ increases by at least 1.

- Alice is in Case ii.A, Bob is in Case ii.B:
  - Φ_{MD} stays at 0.
  - rd^+ and rd^- stay the same.
  - $\ell_{\mathrm{QVC}}^{\mathrm{A}}$ stays the same and $\ell_{\mathrm{QVC}}^{\mathrm{B}}$ increases by 1.
  - q_{MA} stays the same and q_{MB} increases by 1.
  - $pd_+^{\mathrm{A}}, pd_-^{\mathrm{A}}, pd_+^{\mathrm{B}}, pd_-^{\mathrm{B}}$ stay the same.
  - g stays the same, b increases by at most 1 and u decreases by 1.

  Therefore, Φ_{RD}, Φ_{PD} and Φ_{Q} increase by 1, 6 and at least 8, respectively. So Φ increases by at least 1.

- Alice is in Case ii.B, Bob is in Case ii.A: This case is similar to the one above.

- Alice and Bob are in Case iii:
  - Φ_{MD} stays at 0.
  - rd^+, $\ell_{\mathrm{QVC}}^{\mathrm{A}}$ and $\ell_{\mathrm{QVC}}^{\mathrm{B}}$ stay the same.
  - rd^- decreases by 1.
  - $pd_+^{\mathrm{A}}, pd_-^{\mathrm{A}}, pd_+^{\mathrm{B}}, pd_-^{\mathrm{B}}$ stay the same.
  - q_{MA} and q_{MB} do not decrease. q_{MA} increases by 1 if $RA\,[\ell_{RA}] = \mathsf{S}$. Similarly, q_{MB} increases by 1 if $RB\,[\ell_{RB}] = \mathsf{S}$.
  - Φ_{Q} stays the same.

  Therefore, Φ_{RD} decreases by 13 and Φ_{PD} increases by at most 12. So Φ increases by at least 1.

- Alice and Bob are in Case iv:
  - Φ_{MD} stays at 0.
  - rd^{+}, $\ell^{\text{A}}_{\text{QVC}}$ and $\ell^{\text{B}}_{\text{QVC}}$ stay the same.
  - rd^{-} decreases by 1.
  - $pd^{\text{A}}_{+}, pd^{\text{A}}_{-}, pd^{\text{B}}_{+}, pd^{\text{B}}_{-}$ stay the same.
  - q_{MA} and q_{MB} do not decrease. They both increase by 1, if $RA\,[\ell_{RA}] = RB\,[\ell_{RB}] = \mathsf{S}$.
  - Φ_{Q} stays the same.

  Therefore, Φ_{RD} decreases by 13 and Φ_{PD} increases by at most 12. So Φ increases by at least 1.

- Alice and Bob are in Case v:
  - Φ_{MD} stays at 0.
  - rd^{-} stays at 0 and rd^{+} increases by 1.
  - $\ell^{\text{A}}_{\text{QVC}}$ and $\ell^{\text{B}}_{\text{QVC}}$ stay the same.
  - Φ_{PD} and Φ_{Q} stay the same.

  Therefore, Φ_{RD} decreases by 2 and Φ increases by 2.

- Alice and Bob are in Case vi.A:
  - Φ_{MD} and Φ_{RD} stay at 0.
  - $pd^{\text{A}}_{+}, pd^{\text{B}}_{+}, q_{MA}, q_{MB}$ stay the same.
  - None of pd^{A}_{-} and pd^{B}_{-} increases and at least one decreases by 1.
  - Φ_{Q} stays the same.

  Therefore, Φ_{PD} decreases by at least 1. So Φ increases by at least 1.

- Alice and Bob are in Case vi.B:
  - Φ_{MD} and Φ_{RD} stay at 0.
  - q_{MA}, q_{MB} stay the same and pd^{A}_{-}, pd^{B}_{-} stay at 0.
  - At least one of the following holds: $\ell_{PA} < 6q_{MA} \cdot r$, in which case pd^{A}_{+} increases by 1 (otherwise it remains unchanged), or $\ell_{PB} < 6q_{MB} \cdot r$, and then pd^{B}_{+} increases by 1 (otherwise it remains unchanged).
  - Φ_{Q} stays the same.

  Therefore, Φ_{PD} decreases by at least 1. So Φ increases by at least 1.

- Alice and Bob are in Case vii:
  - Φ_{MD}, Φ_{RD} and Φ_{PD} stay at 0.
  - u stays at 0.
  - If $b \neq 0$ then g stays the same and b decreases by 1, otherwise, b stays at 0 and g increases by 1.

  Therefore, Φ_{Q} increases by 1 and so does Φ.

So assuming successful recycling throughout the execution of the algorithm, the potential function Φ increases by at least 1 in every iteration with no transmission error or hash collision. □

Lemma 4.5.19. *Assuming successful recycling throughout the execution of Algorithm 14, each iteration of the algorithm, regardless of the number of hash collisions and transmission errors, decreases the potential function Φ by at most 85.*

Proof. In any iteration, i increases by 1, while g, md_+^{A}, md_+^{B}, rd^+, pd_+^{A} and pd_+^{B} decrease by at most 1; b, u, $\ell_{\text{QVC}}^{\text{A}}$, $\ell_{\text{QVC}}^{\text{B}}$, q_{MA} and q_{MB} increase by at most 1; md_-^{A} and md_-^{B} increase by at most 3; rd^- increases by at most 2; and pd_-^{A} and pd_-^{B} increase by at most 4. Hence, Φ_{MD}, Φ_{RD}, Φ_{PD} increase by at most 22, 30 and 22, respectively, and Φ_{Q} decreases by at most 11. So in total, Φ decreases by at most 85. □

Finally, we are ready to prove the main result of this section.

Theorem 4.2.1 (Restated). *Consider any n-round alternating communication protocol Π in the plain quantum model, communicating messages over a noiseless channel with an alphabet Σ of bit-size $\Theta(\log n)$. Algorithm 14 is a quantum coding scheme which given Π, simulates it with probability at least $1-2^{-\Theta(n\epsilon)}$, over any fully adversarial error channel with alphabet Σ and error rate ϵ. The simulation uses $n\left(1+\Theta\left(\sqrt{\epsilon}\right)\right)$ rounds of communication, and therefore achieves a communication rate of $1-\Theta\left(\sqrt{\epsilon}\right)$.*

Proof. Let $R_{\text{total}} = \left\lceil \frac{n}{2r} \right\rceil + 86\left(c_1+c_3+c_5+2\right)n\epsilon$. By Lemma 4.5.17, recycling is successful throughout the execution of the algorithm with probability at least $1-2^{-\Theta(n\epsilon)}$. Assuming successful recycling, by Lemmas 4.5.4, 4.5.8 and 4.5.12, the total number of iterations with a hash collision is at most $c_1+c_3+c_5$ except with probability $2^{-\Theta(n\epsilon)}$. Since the number of iterations is less than $2n$, the total number of iterations with a transmission error is at most $2n\epsilon$. Therefore, by Lemma 4.5.18, in the remaining $R_{\text{total}} - \left(c_1+c_3+c_5+2\right)n\epsilon$ iterations the potential function Φ increases by at least 1. The potential function decreases in an iteration only if a hash collision or a transmission error occurs and by Lemma 4.5.19, it decreases by at most 85. So at the end of the simulation, we have

$$g-b-u \geq \Phi_{\text{Q}} \geq \Phi \geq R_{\text{total}} - \left(c_1+c_3+c_5+2\right)n\epsilon - 85\left(c_1+c_3+c_5+2\right)n\epsilon \geq \frac{n}{2r} \ .$$

Therefore the simulation is successful with probability at least $1 - 2^{-\Theta(n\epsilon)}$. The cost of entanglement distribution is $\Theta\left(n\sqrt{\epsilon}\right)$. Moreover, the amount of communication in each iteration is independent of the iteration type and is always $(2r + \Theta(1))$: in every iteration each party sends $\Theta(1)$ symbols to communicate the hash values and the value of the pointers in line 17 of Algorithm 14; each party sends another r symbols either in line 21 of Algorithm 14, if *Itertype* $\neq$ SIM or in Algorithm 26. Hence, we have

$$\begin{aligned}\text{Total number of communicated qudits} &= \Theta\left(n\sqrt{\epsilon}\right) + R_{\text{total}} \cdot (2r + \Theta(1)) \\ &= \Theta\left(n\sqrt{\epsilon}\right) + \left(\left\lceil \frac{n}{2r} + \Theta\left(n\epsilon\right)\right\rceil\right)(2r + \Theta(1)) \\ &= n\left(1 + \Theta\left(\sqrt{\epsilon}\right)\right) \quad .\end{aligned}$$

□

Chapter 5

Interactive coding over small communication alphabets

In this chapter, we focus on the simulation of interactive communication over constant-size alphabets.

5.1 Overview

5.1.1 Challenges in capacity approaching coding over small alphabets

In the large alphabet case, it is possible to send $\mathrm{O}(\log n)$ bits of information by communicating only a constant number of symbols. However, such a powerful resource is no longer available to us when the communication alphabet is of constant size.

Exchanging length information to coordinate rewinding is inapplicable. In noisy interactive communication based on the rewind-if-error paradigm, the idea is to exchange summaries of the conversation at regular intervals and rewind the conversation back to an earlier stage, whenever an inconsistency is detected. The rewinding subroutine in this simple template, however, needs to be designed carefully in order to achieve a high communication rate. A crucial property for the rewinding procedure is that the rewinding length should be proportional to the amount of communication which triggers it. This prevents the adversary from derailing the simulation by falsely triggering super-constant rewinding steps while investing only a constant number of transmission errors each time. The simplest way to achieve the property above is to rewind a constant number of steps each time an inconsistency is detected. However, if Alice and Bob have transcripts of different lengths and rewind the same number of steps each time, they do not get closer until they

have rewound the correct prefix of their transcripts as well. Therefore, they first need to learn which party is ahead, and then the party with the longer transcript has to rewind until they have equally long transcripts. The adversary's corruptions can introduce transcript length differences of up to $\Theta(n\epsilon)$ steps between the two parties. Therefore, unless we use a different approach, coordinating the rewinding steps seems to require communicating logarithmic-size length information. When the communication alphabet is of constant size, this corresponds to a logarithmic amount of communication for each rewinding step, which leads to a vanishing simulation rate.

Long seeds needed for hashing. It is straightforward to see that the amount of communication in each iteration of the simulation algorithm should not grow with n, the length of the input protocol. To see this, let R_{total} denote the total number of iterations and m be the number of qudits communicated per iteration. A single transmission error by the adversary can make a whole iteration completely useless. The error budget of the adversary thus suffices to corrupt $\epsilon m R_{\text{total}}$ iterations. Therefore, unless $m < 1/\epsilon$, the adversary would be able to stall the whole simulation by corrupting every iteration. The communication in each iteration includes the hash values that Alice and Bob exchange in order to compare their local data. So, these hash functions must have an output length of at most $\mathrm{O}_\epsilon(1)$ symbols from the communication alphabet Σ, or equivalently, $\mathrm{O}_\epsilon(1)$ bits in the small alphabet case.

The classical data that are compared through hashing are $\mathrm{O}(n)$ bits long. Intuitively, in order to detect inconsistencies in such long strings with a nontrivial collision probability, using hash values that are only $\mathrm{O}_\epsilon(1)$ bits long, Alice and Bob need to use a very long random seed for each hashing step. In fact, a straightforward argument implies the following lower bound on the length of the seed.

Lemma 5.1.1. *[37] For any hash function with any nontrivial collision probability strictly less than* 1*, the seed length s satisfies $s \geq \log \frac{\ell \log|\Sigma|}{o}$, where $\ell \log|\Sigma|$ is the bit-length of the strings it hashes and o is the bit-length of the output.*

For output length $o \in \mathrm{O}_\epsilon(1)$, the seed length has to grow as a logarithmic function of the length of the strings to be hashed. In the plain communication model, however, Alice and Bob need to establish their shared randomness through extra communication. This leads to a vanishing simulation rate. One way to avoid using such long random seeds in each iteration is to hash only a few recent blocks of the data. However, this only works well in the random noise model. In the adversarial error case, the adversary can hide errors from both parties for up to $\Theta(n\epsilon)$ iterations. Therefore, Alice and Bob need to hash a longer suffix of length at least $\Omega(n\epsilon)$ of their data to ensure they can detect inconsistencies with high probability.

5.1.2 Meeting point-based rewinding

In this section we explain a rewinding mechanism introduced by Haeupler [37] which allows the two parties to coordinate their rewinding actions without exchanging their transcript length information. We first focus on a simplified scenario to introduce the main ideas behind *meeting point-based rewinding*.

Suppose that Alice is given T_{A} and Bob is given T_{B}, where T_{A} and T_{B} are strings over some finite alphabet. Let ℓ^+ be the length of the longest prefix in which T_{A} and T_{B} are equal and let ℓ^- denote their disagreement length, i.e., $\ell^- = \max\{|T_{\mathrm{A}}|, |T_{\mathrm{B}}|\} - \ell^+$. We are interested in the typical scenario encountered in protocols based on the rewind-if-error paradigm, where ℓ^- is small compared to ℓ^+. Alice and Bob need to agree on a meeting point close to ℓ^+, up to which T_{A} and T_{B} match. To further simplify the task, suppose that the two parties have access to a noiseless communication channel. In meeting point-based rewinding, Alice and Bob use a *rewinding step size* k. The rewinding step size defines meeting points on T_{A} and T_{B} that are multiples of k. Alice and Bob want to meet at the same meeting point without having to communicate their transcript lengths. It is straightforward to see that for $k \geq \ell^-$, Alice and Bob have at least one common meeting point among their last two meeting points. Starting from $k = 1$, in each iteration Alice and Bob use hashing to compare T_{A} and T_{B} up to the last two meeting points corresponding to k. They repeat this process while increasing k by 1 in each iteration, until they find a common meeting point to rewind to. We remark that comparing at least the last two meeting points in every iteration is crucial to ensure that the rewinding process terminates within ℓ^- iterations.

In Section 5.3, we explain how the rewinding scheme above is adapted to be used as a subroutine in noisy interactive communication for synchronizing classical data.

5.1.3 Hashing with pseudo-random seeds

In the plain quantum communication model, in order to establish the shared randomness and the MESs required for the simulation protocol, Alice locally creates sufficient copies of the MES $|\phi^{0,0}\rangle$ and sends half of all the copies to Bob using a "good" quantum error correcting code. A fraction of these MESs are measured in the computational basis to obtain a shared random string used as the seed in the hashing process. The independent seeds required in every iteration for hashing the classical data are $\Theta(\log n)$ bits long. Therefore, in the small alphabet case, the amount of communication required to establish such a long uniformly random bit string is super-linear in n, leading to a vanishing communication rate. We use Haeupler's approach in the classical case [37] to circumvent this obstacle. Instead of directly distributing such a large amount of randomness, we establish a much shorter uniformly random string and then stretch it to a sufficiently long small bias pseudo-random string, which is statistically indistinguishable from being independent. This is achieved using the deterministic algorithm of Lemma 2.4.8. In order to compare the classical data we use the following simple hash function:

Definition 5.1.2. (Inner product hash function [37]) For input length parameter L and output length parameter o, the inner product hash function

$$\text{iphash} : \{0,1\}^{2oL} \times \{0,1\}^{\leq L} \to \{0,1\}^{o}$$

is defined as follows: Any input binary string X of length $l \leq L$ is first concatenated with its length to obtain $\tilde{X} = (X, |X|)$ of length $\tilde{l} = l + \lceil \log l \rceil \leq 2L$. Then given any binary seed S of length $2oL$, the i-th output bit is given by

$$\left(\text{iphash}_S(X)\right)[i] \stackrel{\text{def}}{=} \left\langle \tilde{X}, S\left[i \cdot 2L + 1 : i \cdot 2L + \tilde{l}\right] \right\rangle \quad , \quad i \in \{0, \ldots, o-1\} \ .$$

It is easy to see that the collision probability for a uniformly random seed is exactly 2^{-o}. Moreover, by Definition 2.4.2, if the seed is sampled from a δ-biased distribution then the collision probability remains bounded by $2^{-o} + \delta$. In our algorithm, we set $o \in \Theta(1)$ for a constant collision probability. Hence, we need $\Theta(n)$-bit long seeds for every hashing step. The benefit of using the inner product hash function is the easier analysis when the seed is replaced by a (pseudo-random) small-bias string, which allows us to translate the bias to the outcome of the hashing steps.

5.2 Results

Our main results concerning noisy interactive quantum communication over constant-size alphabets are coding schemes for simulation of noiseless interactive communication over any fully adversarial channel of error-rate ϵ, in both the teleportation-based model and the plain quantum model.

Theorem 5.2.1. *Consider any n-round alternating communication protocol Π in the teleportation-based model, designed for communication over a noiseless channel with an alphabet Σ of constant size. We provide a computationally efficient coding scheme which given Π, simulates it with probability at least $1 - 2^{-\Theta(n\epsilon)}$, over any fully adversarial error channel with alphabet Σ and error rate ϵ. The simulation uses $n\left(1 + \Theta\left(\sqrt{\epsilon}\right)\right)$ rounds of communication, and therefore achieves a communication rate of $1 - \Theta\left(\sqrt{\epsilon}\right)$. Furthermore. the computational complexity of the coding operations is $O\left(n^2\right)$.*

Our main result in the plain model of quantum communication is the following:

Theorem 5.2.2. *Consider any n-round alternating communication protocol Π in the plain quantum model, communicating messages over a noiseless channel with a constant size alphabet Σ. Algorithm 28 is a quantum coding scheme which given Π, simulates it with probability at least $1 - 2^{-\Theta(n\epsilon)}$, over any fully adversarial error channel with alphabet Σ and error rate ϵ. The simulation uses $n\left(1 + \Theta\left(\sqrt{\epsilon}\right)\right)$ rounds of communication, and therefore achieves a communication rate of $1 - \Theta\left(\sqrt{\epsilon}\right)$.*

Note that in the classical setting with constant-size alphabet and no pre-shared randomness, the conjectured optimal simulation rate is $1 - \Theta(\sqrt{\epsilon})$ for oblivious channels and $1 - \Theta\left(\sqrt{\epsilon \log\log \frac{1}{\epsilon}}\right)$ for fully adversarial channels [37]. Our simulation in the plain quantum model outperforms the best known protocol in the corresponding classical setting. This advantage may be interpreted as follows. When using quantum communication, Alice and Bob can establish MESs and measure them to generate the hash seeds. One advantage of establishing the shared randomness in this way is that the seeds remain unknown to the adversary. Thus the adversary is not able to create hash collisions purposely. This is in contrast to the classical case with no pre-shared randomness, where the seeds need to be communicated over the classical channel and the adversary gets to know the seeds. The knowledge of the seeds enables the adversary to introduce errors which remain undetected with certainty. As a result, in Ref. [37] another layer of hashing is added to the algorithm for the oblivious noise model to protect against fully adversarial noise, dropping the simulation rate from $1 - \Theta(\sqrt{\epsilon})$ to $1 - \Theta\left(\sqrt{\epsilon \log\log 1/\epsilon}\right)$.

Since the steps in adapting the large alphabet simulation protocols to the small alphabet case are similar in both communication models above, in the remainder of this chapter we focus on the more involved recycling-based protocol for the plain quantum model. We remark that the same modifications and a similar analysis can be used for the teleportation-based model to prove Theorem 5.2.1.

5.3 Description of Protocol

The structure of our simulation protocol in the plain quantum communication model over small alphabets is quite similar to the large alphabet case. Alice and Bob maintain the same classical data structure to keep track of the evolution of their joint state. They distribute a sufficient amount of entanglement at the beginning of the protocol using standard quantum error correction. The shared entanglement is partially used as a source of randomness in comparing the local data in each iteration. The two parties prevent the adversary from irreversibly corrupting the simulation of the input protocol by protecting their messages using QVC. As before, quantum hashing is used to detect any errors introduced by the adversary on the QVC messages. The MES pairs used in QVC are recycled and reused as needed using the machinery introduced in Subsection 4.3.1. However, as mentioned earlier, synchronizing the classical data is more involved when the communication alphabet is of constant size. In this section, we only focus on the modifications that are necessary in order to use the simulation protocol of Chapter 4 in the small alphabet case. In particular, we explain how the meeting point-based rewinding scheme introduced by Haeupler [37] can be adapted to be used as a subroutine in our simulation protocol. For a detailed description of the remaining steps of the algorithm, please refer to Section 4.3.

5.3.1 Meeting point-based rewinding in noisy interactive communication

In Section 5.1.2, we provided a high level description of meeting point-based rewinding when a noiseless communication channel is available. In this section, we revisit meeting point-based rewinding and discuss the necessary changes for the scheme to be used in interactive communication against adversarial noise.

Alice and Bob compare their classical data in every iteration by exchanging the corresponding hashes and conduct the rewinding subroutine whenever they detect an inconsistency. In the rewinding subroutine, Alice and Bob maintain step sizes k_{A} and k_{B}, respectively, which they increase in each rewinding iteration until they find a common meeting point. Once they find such a meeting point, they rewind back and reset their step sizes to 1. When communicating over a noisy channel, however, corruptions may cause only one party to rewind, potentially increasing the disagreement length ℓ^- and/or decreasing ℓ^+. In such a scenario, the two parties will also get out-of-sync in their rewinding step sizes. Recall that the adversary can cause a transcript disagreement of length $\ell^- \in \Omega(n\epsilon)$. Therefore, k_{A} and k_{B} can be very large and in order to keep them synchronized, Alice and Bob cannot send them directly in each iteration. Instead, the parties use hashing to compare their step sizes and reset them if they detect too many discrepancies. We refer to this as a *status reset* due to a *mismatch transition*. Alice and Bob maintain *mismatch count variables*, E_{A} and E_{B}, respectively, to keep track of the number of iterations in which they have observed non-matching step size hashes. Alice resets k_{A} and E_{A} once E_{A} is greater than $k_{\mathrm{A}}/2$. Similarly, Bob does a status reset once E_{B} is greater than $k_{\mathrm{B}}/2$. This ensures that the number of corruptions the adversary needs to invest to falsely trigger a mismatch transition is proportional to the amount of communication spent while increasing the corresponding step size to its value before the transition.

Now, with a procedure for the parties to agree on the same rewinding step size k, we explain how they decide when to rewind to a meeting point. As explained earlier, a communication efficient rewinding scheme against adversarial noise needs to guarantee that a long rewinding step cannot be triggered by a small amount of communication. Therefore, Alice and Bob need to compare their meeting points multiple times to gain enough confidence before rewinding. In particular, we need the number of meeting point hash comparisons to be proportional to the rewinding length, i.e., proportional to k. In order to achieve this, for every k_{A}, a "coarse-grained" step size $\hat{k}_{\mathrm{A}} \stackrel{\text{def}}{=} 2^{\lfloor \log k_{\mathrm{A}} \rfloor}$ is defined and Alice's meeting points are defined as multiples of $\hat{k}_{\mathrm{A}}$ instead of k_{A}. Similarly, Bob's meeting points are now defined as multiples of $\hat{k}_{\mathrm{B}} \stackrel{\text{def}}{=} 2^{\lfloor \log k_{\mathrm{B}} \rfloor}$. In each rewinding iteration with consistent step size hashes, Alice and Bob compare hashes of their data up to their last two meeting points. We refer to this phase of the rewinding subroutine as the *voting phase*. Alice and Bob maintain a *vote count variable* for each of their meeting points. They vote for a meeting point each time it appears to match one of the other party's meeting points. A party rewinds back to a meeting point if it receives at least 80 percent of all the possible votes at the current coarse-grained step size $\hat{k}$, i.e., $0.8 \times \hat{k}/2 = 0.4\hat{k}$ votes. After rewinding back, the step size,

the mismatch count variable, and the vote count variables of the transitioning party (or parties) get reset. We refer to this as a status reset due to a *meeting point transition*.

5.3.2 Entanglement distribution and generating the pseudo-random seed

As in the large alphabet case, at the outset of the simulation, Alice and Bob use Algorithm 12 to share $\Theta(n\sqrt{\epsilon})$ copies of the MES $|\phi^{0,0}\rangle$. The shared MESs are used as follows:

- $\Theta\left(n\sqrt{\epsilon}\right)$ MESs are used in pairs to serve as the key for encryption of messages using QVC. They are divided into $L_{\text{QVC}} = \Theta\left(n\epsilon\right)$ blocks of $2r$ MES pairs, where $r = \Theta(\frac{1}{\sqrt{\epsilon}})$. In each block the MES pairs are implicitly numbered from 1 to $2r$. The odd-numbered pairs are used in QVC to send messages from Alice to Bob and the even-numbered pairs are used to send messages from Bob to Alice.
- $\Theta(n\sqrt{\epsilon})$ MESs are reserved to be used in quantum hashing.
- The remaining $\Theta(n\sqrt{\epsilon})$ MESs are measured in the computational basis by both parties to obtain a common random string to be used as the seed for classical and quantum hashing.

Quantum hashing in each iteration of the algorithm requires independent and uniformly random $\Theta\left(rL_{\text{QVC}}\right)$-bit seeds, with $r \in \Theta\left(1/\sqrt{\epsilon}\right)$ and $L_{\text{QVC}} \in \Theta\left(n\epsilon\right)$. Moreover, hashing the classical data using the inner product hash function of Definition 5.1.2 requires independent $\Theta\left(n\right)$-bit uniformly random seeds. To avoid distributing $R_{\text{total}} \cdot \Theta\left(rL_{\text{QVC}} + n\right) = \Theta\left(n\sqrt{\epsilon}\right)\Theta\left(n\sqrt{\epsilon} + n\right) = \Theta\left(n^2\sqrt{\epsilon}\right)$ i.i.d. random bits directly, Alice and Bob measure $\Theta\left(n\sqrt{\epsilon}\right)$ MESs in the computational basis and record the binary representation of the outcomes in R'. Then they use the deterministic algorithm of Lemma 2.4.8 with $\delta = 2^{-\Theta\left(n\sqrt{\epsilon}\right)}$, to stretch their shared uniformly random string R' to a δ-biased binary string of length $\Theta\left(n^2\sqrt{\epsilon}\right)$. The pseudo-random string is divided into two part: The classical seed $R = R_1 \cdots R_{10R_{\text{total}}}$ used in quantum hashing, where each R_j is of length $4rL_{\text{QVC}}$; and the seed $S = S_1 \cdots S_{12R_{\text{total}}}$ used by iphash, where each S_j is of length $\Theta\left(n\right)$.

5.3.3 Summary of main steps

In Algorithm 27, we summarize the outline of the steps which are followed by the two parties in the simulation. The hierarchy of the steps are the same as before and Alice and Bob skip one step to the next only if the goal of the step has been achieved through the previous iterations.

Note that, except for the rewinding subroutines for the metadata and the Pauli data, the algorithms mentioned in each step are the same subroutines presented in Section 4.4.2.

Algorithm 27: Main steps in one iteration of the simulation for the small alphabet recycling-based model

1 Agree on the history of the simulation contained in metadata, i.e., ensure $FullMA = \widetilde{MA}$ and $FullMB = \widetilde{MB}$. This involves Algorithm 33—**rewindMD-small-alphabet** and Algorithm 6—**extendMD**.

2 Synchronize the number of MES blocks used in QVC, in particular, ensure $\ell^{\mathrm{A}}_{\mathrm{QVC}} = \widetilde{\ell^{\mathrm{B}}_{\mathrm{QVC}}}$ and $\ell^{\mathrm{B}}_{\mathrm{QVC}} = \widetilde{\ell^{\mathrm{A}}_{\mathrm{QVC}}}$. This is done via Algorithm 19—**Q-syncMES**.

3 Agree on the measurement pointers and the recycling data up to the pointers, in particular, ensure $(\ell_{RA}, RA\,[1:\ell_{RA}]) = \left(\widetilde{\ell_{RB}}, \widetilde{RB}\left[1:\widetilde{\ell_{RB}}\right]\right)$ and $(\ell_{RB}, RB\,[1:\ell_{RB}]) = \left(\widetilde{\ell_{RA}}, \widetilde{RA}\left[1:\widetilde{\ell_{RA}}\right]\right)$. This involves Algorithm 20—**rewindRD**.

4 Ensure no undetected quantum error from earlier rounds exists. This is done by ensuring $QHA = QHB$ and involves Algorithm 21—**measuresyndrome**.

5 Ensure $\ell_{RA} = \ell^{\mathrm{A}}_{\mathrm{QVC}}$ and $\ell_{RB} = \ell^{\mathrm{B}}_{\mathrm{QVC}}$. This is achieved via Algorithm 22—**extendRD**.

6 Agree on Pauli data, in particular, ensure $FullPA = \widetilde{PA}$ and $FullPB = \widetilde{PB}$. This is done via Algorithm 34—**rewindPD-small-alphabet** and Algorithm 24—**Q-extendPD**.

7 Compute the best guess for $JS1$ and $JS2$. If there are any "bad" blocks in the guess for $JS2$, reverse the last bad block of unitary operations. I.e., implement quantum rewinding so that $b = 0$ in $JS2$. This is done in Algorithm 26—**Q-simulate**.

8 If no "bad" blocks remain, implement the next block of rounds of the original protocol. This results in an increase in g in $JS2$, and is also done through Algorithm 26—**Q-simulate**.

5.4 Algorithm

5.4.1 Data structure

In addition to the variables introduced in Subsection 4.4.1, Alice and Bob maintain *rewinding data* which they use to coordinate the rewinding process when they reconcile inconsistencies in their metadata or Pauli data.

- **Metadata rewinding variables:** Alice's rewinding step size corresponding to MA is denoted by k_{MA}. In the rewinding subroutine, k_{MA} either increases by 1 or gets reset to the value 1 (when a status reset occurs). We define $\hat{k}_{MA} \stackrel{\text{def}}{=} 2^{\lfloor \log k_{MA} \rfloor}$. The two meeting points corresponding to $\hat{k}_{MA}$ at the beginning of each iteration are defined as $mp1_{MA} \stackrel{\text{def}}{=} \hat{k}_{MA} \lfloor \frac{\ell_{MA}}{\hat{k}_{MA}} \rfloor$ and $mp2_{MA} \stackrel{\text{def}}{=} mp1_{MA} - \hat{k}_{MA}$. We denote by MA_1 and MA_2 the prefixes of MA (or equivalently $FullMA$) of lengths $mp1_{MA}$ and $mp2_{MA}$, respectively, i.e., $MA_1 \stackrel{\text{def}}{=} MA\,[1 : mp1_{MA}]$ and $MA_2 \stackrel{\text{def}}{=} MA\,[1 : mp2_{MA}]$. Bob's variables $k_{\widetilde{MA}}$, $mp1_{\widetilde{MA}}$, $mp2_{\widetilde{MA}}$, $\widetilde{MA}_1$, and $\widetilde{MA}_2$ corresponding to $\widetilde{MA}$ are defined similarly. At the beginning of every iteration, Alice and Bob exchange hash values corresponding to their rewinding step sizes and local data up to the meeting points. Following our convention, we use H with the corresponding variable as a subscript to denote the hash values, e.g., $H_{k_{MA}}$ is the hash value corresponding to k_{MA}. The variables with a superscript $'$ denote the received data after transmission over the noisy channel, e.g., $H'_{k_{MA}}$ denotes what Bob receives when Alice sends $H_{k_{MA}}$. Alice's mismatch count variable E_{MA} keeps track of iterations with non-matching step sizes. In the rewinding subroutine, Alice increases E_{MA} by 1 whenever $H_{k_{MA}} \neq H'_{k_{\widetilde{MA}}}$. Otherwise, if $H_{k_{MA}} = H'_{k_{\widetilde{MA}}}$, she enters the meeting point voting phase. We denote Alice's vote count variables corresponding to MA_1 and MA_2 by v_{MA_1} and v_{MA_2}, respectively. Each vote count increases by 1 in the voting phase, if the corresponding hash value matches one of the meeting point hashes she receives from Bob. The variables v_{MA_1}, v_{MA_2}, and E_{MA} get reset to 0 with each status reset. Alice's rewinding data corresponding to $\widetilde{MB}$ and Bob's rewinding data corresponding to $\widetilde{MA}$ and MB are defined and updated similarly.

- **Pauli data rewinding variables:** These variables are defined in the same way as the metadata rewinding data. The only difference is that the unit length in defining the meeting points is now a block of length r. For instance, Alice's meeting points on PA are now multiples of $\hat{k}_{PA}r$ instead of $\hat{k}_{PA}$ and her two closest meeting points corresponding to coarse-grained step size $\hat{k}_{PA}$ are now defined as $mp1_{PA} \stackrel{\text{def}}{=} \hat{k}_{PA}r \lfloor \frac{\ell_{PA}}{\hat{k}_{PA}r} \rfloor$ and $mp2_{PA} \stackrel{\text{def}}{=} mp1_{PA} - \hat{k}_{PA}r$.

5.4.2 Pseudo-code

This section contains the pseudo-code for the main algorithm and the subroutines that are different from those presented in Section 4.4.2, for the large alphabet case.

Algorithm 28: Q-Main-small-alphabet (Alice's side)

Input: n round protocol Π in plain quantum model over constant-size alphabet Σ

Q-Initialization-small-alphabet;
For $i = 1 \to R_{\text{total}}$
  if $i \leq L_{\text{QVC}}$ **then**
    $NextIndexA \leftarrow i$; `// No recycling in first` L_{QVC} `iterations`
  else
    Recycle;
    if $NextIndexA = \perp$ **then**
      Abort;
  $IndexA \leftarrow (IndexA, NextIndexA)$;
  $RA \leftarrow (RA, \mathsf{S})$;
  ▷ `computing hash values`
  $\left(H_{k_{MA}}, H_{k_{\widetilde{MB}}}, H_{k_{PA}}, H_{k_{\widetilde{PB}}}\right) \leftarrow$
  $\left(h_{S_{12i-11}}(k_{MA}), h_{S_{12i-10}}\left(k_{\widetilde{MB}}\right), h_{S_{12i-9}}(k_{PA}), h_{S_{12i-8}}\left(k_{\widetilde{PB}}\right)\right)$;
  $(mp1_{MA}, mp2_{MA}, MA_1, MA_2, H_{MA_1}, H_{MA_2}) \leftarrow$
  MPhash $(k_{MA}, \ell_{MA}, S_{12i-7}, S_{12i-6}, MA, 1)$;
  $\left(mp1_{\widetilde{MB}}, mp2_{\widetilde{MB}}, \widetilde{MB}_1, \widetilde{MB}_2, H_{\widetilde{MB}_1}, H_{\widetilde{MB}_2}\right) \leftarrow$
  MPhash $\left(k_{\widetilde{MB}}, \ell_{\widetilde{MB}}, S_{12i-5}, S_{12i-4}, \widetilde{MB}, 1\right)$;
  $(mp1_{PA}, mp2_{PA}, PA_1, PA_2, H_{PA_1}, H_{PA_2}) \leftarrow$
  MPhash $(k_{PA}, \ell_{PA}, S_{12i-3}, S_{12i-2}, PA.r)$;
  $\left(mp1_{\widetilde{PB}}, mp2_{\widetilde{PB}}, \widetilde{PB}_1, \widetilde{PB}_2, H_{\widetilde{PB}_1}, H_{\widetilde{PB}_2}\right) \leftarrow$
  MPhash $\left(k_{\widetilde{PB}}, \ell_{\widetilde{PB}}, S_{12i-1}, S_{12i}, \widetilde{PB}, r\right)$;
  Quantum-hash;

Algorithm 28: Q-Main-small-alphabet (Alice's side, cont. from previous page)

17

18 Send

$$\left(H_{k_{MA}}, H_{MA_1}, H_{MA_2}, H_{k_{\widetilde{MB}}}, H_{\widetilde{MB}_1}, H_{\widetilde{MB}_2}, QHA, H_{k_{PA}}, H_{PA_1}, H_{PA_2}, H_{k_{\widetilde{PB}}}, H_{\widetilde{PB}_1}, H_{\widetilde{PB}_2}\right);$$

19 Receive

$$\left(H'_{k_{\widetilde{MA}}}, H'_{\widetilde{MA}_1}, H'_{\widetilde{MA}_2}, H'_{k_{MB}}, H'_{MB_1}, H'_{MB_2}, QHB', H'_{k_{\widetilde{PA}}}, H'_{\widetilde{PA}_1}, H'_{\widetilde{PA}_2}, H'_{k_{PB}}, H'_{PB_1}, H'_{PB_2}\right);$$

▷ `Determining iteration type`

20 **Q-Preprocess-small-alphabet**;

21 **if** *Itertype* $\neq$ SIM **then**

22 Send *msg*;

23 Receive *msg*′;

`// messages are communicated alternately`

▷ `Case i.A`

24 **if** *Itertype* = MD and *RewindExtend* = R **then**

25 **rewindMD-small-alphabet**;

▷ `Case i.B`

26 **else if** *Itertype* = MD and *RewindExtend* = E **then**

27 **extendMD**;

▷ `Case ii.A`

28 **else if** *Itertype* = MES and *NewMetaA* = C **then**

29 return;

▷ `Case ii.B`

30 **else if** *Itertype* = MES and *NewMetaA* = 0_{ED} **then**

31 **Q-syncMES**;

▷ `Case iii`

32 **else if** *Itertype* = RD and *RewindExtend* = R **then**

33 **rewindRD**;

▷ `Case iv`

34 **else if** *Itertype* = QH **then**

35 **measuresyndrome**;

▷ `Case v`

36 **else if** *Itertype* = RD and *RewindExtend* = E **then**

37 **extendRD**;

Algorithm 28: Q-Main-small-alphabet (Alice's side, cont. from previous page)

38

▷ `Case vi.A`

39 **else if** *Itertype* = PD and *RewindExtend* = R **then**

40 **rewindPD-small-alphabet**;

▷ `Case vi.B`

41 **else if** *Itertype* = PD and *RewindExtend* = E **then**

42 **extendPD**;

▷ `Case vii`

43 **else**

44 **Q-Simulate**;

45 **return Q-Main-small-alphabet**;

Algorithm 29: Q-Preprocess-small-alphabet (Alice's side)

Input:

$$\begin{pmatrix} H_{MA}, \ell_{MA}, H_{\widetilde{MB}}, \ell_{\widetilde{MB}}, H_{PA}, \ell_{PA}, H_{\widetilde{PB}}, \ell_{\widetilde{PB}}, QHA \\ H'_{\widetilde{MA}}, \ell'_{\widetilde{MA}}, H'_{MB}, \ell'_{MB}, H'_{\widetilde{PA}}, \ell'_{\widetilde{PA}}, H'_{PB}, \ell'_{PB}, QHB' \\ FullMA, \ell^{\mathrm{A}}_{\mathrm{QVC}}, \widetilde{MB}, RA, \ell_{RA}, FullPA, \widetilde{PB} \end{pmatrix}$$

Output:

$$\left(Itertype, RewindExtend, NewMetaA, FullMA, \ell_{MA}, \widetilde{NewMetaB}, \widetilde{MB}, \ell_{\widetilde{MB}}, msg\right)$$

1 **if** $(k_{MA}, H_{k_{MA}}, H_{MA_1}) = \left(1, H'_{k_{\widetilde{MA}}}, H'_{\widetilde{MA}_1}\right)$ and $\left(k_{\widetilde{MB}}, H_{k_{\widetilde{MB}}}, H_{\widetilde{MB}_1}\right) = \left(1, H'_{k_{MB}}, H'_{MB_1}\right)$ and $\ell_{MA} = \ell_{\widetilde{MB}} = i-1$ **then**

2 Compute $\widetilde{\ell^{\mathrm{B}}_{\mathrm{QVC}}}$, $\widetilde{RB}$, $\widetilde{\ell_{RB}}$, $q_{\widetilde{MB}}$;

▷ `Processing Metadata`

▷ `Case i.A`

3 **if** $(k_{MA}, H_{k_{MA}}, H_{MA_1}) \neq \left(1, H'_{k_{\widetilde{MA}}}, H'_{\widetilde{MA}_1}\right)$ or $\left(k_{\widetilde{MB}}, H_{k_{\widetilde{MB}}}, H_{\widetilde{MB}_1}\right) \neq \left(1, H'_{k_{MB}}, H'_{MB_1}\right)$ **then**

4 $Itertype \leftarrow$ MD;

5 $RewindExtend \leftarrow$ R;

6 $NewMetaA \leftarrow$ C;

7 $FullMA \leftarrow (FullMA, NewMetaA)$;

8 $msg \leftarrow$ dummy message of length r;

Algorithm 29: Q-Preprocess-small-alphabet (Alice's side, cont. from previous page)

▷ **Case i.B**

else if $(\ell_{MA} < i-1)$ or $\left(\ell_{\widetilde{MB}} < i-1\right)$ **then**
  $\mathit{Itertype} \leftarrow \mathrm{MD}$;
  $\mathit{RewindExtend} \leftarrow \mathsf{E}$;
  $\mathit{NewMetaA} \leftarrow \mathsf{C}$;
  $\mathit{FullMA} \leftarrow (\mathit{FullMA}, \mathit{NewMetaA})$;
  if $\ell_{MA} < i-1$ **then**
    $msg \leftarrow \mathrm{encodeMD}\left(\mathit{FullMA}\left[\ell_{MA}+1, \ell_{MA}+2\right]\right)$; **// Encode MD in** Σ^r
  else
    $msg \leftarrow$ dummy message of length r;

▷ **Comparing number of used MES blocks**

▷ **Case ii.A**

else if $\ell^{\mathrm{A}}_{\mathrm{QVC}} > \widetilde{\ell^{\mathrm{B}}_{\mathrm{QVC}}}$ **then**
  $\mathit{Itertype} \leftarrow \mathrm{MES}$;
  $\mathit{NewMetaA} \leftarrow \mathsf{C}$;
  $\mathit{FullMA} \leftarrow (\mathit{FullMA}, \mathit{NewMetaA})$;
  $\ell_{MA} \leftarrow \ell_{MA} + 1$;
  $\widetilde{\mathit{NewMetaB}} \leftarrow \mathsf{0}_{\mathsf{ED}}$;
  $\widetilde{MB} \leftarrow \left(\widetilde{MB}, \widetilde{\mathit{NewMetaB}}\right)$;
  $\ell_{\widetilde{MB}} \leftarrow \ell_{\widetilde{MB}} + 1$;
  $msg \leftarrow$ dummy message of length r;

▷ **Case ii.B**

else if $\ell^{\mathrm{A}}_{\mathrm{QVC}} < \widetilde{\ell^{\mathrm{B}}_{\mathrm{QVC}}}$ **then**
  $\mathit{Itertype} \leftarrow \mathrm{MES}$;
  $\mathit{NewMetaA} \leftarrow \mathsf{0}_{\mathsf{ED}}$;
  $\mathit{FullMA} \leftarrow (\mathit{FullMA}, \mathit{NewMetaA})$;
  $\ell_{MA} \leftarrow \ell_{MA} + 1$;
  $\widetilde{\mathit{NewMetaB}} \leftarrow \mathsf{C}$;
  $\widetilde{MB} \leftarrow \left(\widetilde{MB}, \widetilde{\mathit{NewMetaB}}\right)$;
  $\ell_{\widetilde{MB}} \leftarrow \ell_{\widetilde{MB}} + 1$;
  $msg \leftarrow$ dummy message of length r;

Algorithm 29: Q-Preprocess-small-alphabet (Alice's side, cont. from previous page)

▷ `Processing recycling data`

▷ `Case iii`

else if $(\ell_{RA}, RA[1:\ell_{RA}]) \neq \left(\widetilde{\ell_{RB}}, \widetilde{RB}\left[1:\widetilde{\ell_{RB}}\right]\right)$ **then**
  $Itertype \leftarrow$ RD;
  $RewindExtend \leftarrow \mathsf{R}$;
  if $\ell_{RA} > \widetilde{\ell_{RB}}$ **then**
    $NewMetaA \leftarrow \mathsf{M}$;
    $\widetilde{NewMetaB} \leftarrow \mathsf{C}$;
  else if $\ell_{RA} < \widetilde{\ell_{RB}}$ **then**
    $NewMetaA \leftarrow \mathsf{C}$;
    $\widetilde{NewMetaB} \leftarrow \mathsf{M}$;
  else
    $NewMetaA \leftarrow \mathsf{M}$;
    $\widetilde{NewMetaB} \leftarrow \mathsf{M}$;
  $FullMA \leftarrow (FullMA, NewMetaA)$;
  $\ell_{MA} \leftarrow \ell_{MA} + 1$;
  $\widetilde{MB} \leftarrow \left(\widetilde{MB}, \widetilde{NewMetaB}\right)$;
  $\ell_{\widetilde{MB}} \leftarrow \ell_{\widetilde{MB}} + 1$;
  $msg \leftarrow$ dummy message of length r;

▷ `Case iv`

else if $QHA \neq QHB'$ **then**
  $Itertype \leftarrow$ QH;
  $NewMetaA \leftarrow \mathsf{M}$;
  $FullMA \leftarrow (FullMA, NewMetaA)$;
  $\ell_{MA} \leftarrow \ell_{MA} + 1$;
  $\widetilde{NewMetaB} \leftarrow \mathsf{M}$;
  $\widetilde{MB} \leftarrow \left(\widetilde{MB}, \widetilde{NewMetaB}\right)$;
  $\ell_{\widetilde{MB}} \leftarrow \ell_{\widetilde{MB}} + 1$;
  $msg \leftarrow$ dummy message of length r;

Algorithm 29: Q-Preprocess-small-alphabet (Alice's side, cont. from previous page)

▷ `Case v`

else if $\ell_{RA} < \ell^{\mathrm{A}}_{\mathrm{QVC}}$ **then**
- $Itertype \leftarrow \mathrm{RD}$;
- $RewindExtend \leftarrow \mathsf{E}$;
- $NewMetaA \leftarrow \mathsf{C}'$;
- $FullMA \leftarrow (FullMA, NewMetaA)$;
- $\ell_{MA} \leftarrow \ell_{MA} + 1$;
- $\widetilde{NewMetaB} \leftarrow \mathsf{C}'$;
- $\widetilde{MB} \leftarrow \left(\widetilde{MB}, \widetilde{NewMetaB}\right)$;
- $\ell_{\widetilde{MB}} \leftarrow \ell_{\widetilde{MB}} + 1$;
- $msg \leftarrow$ dummy message of length r;

▷ `Processing Pauli data`

▷ `Case vi.A`

else if $(k_{PA}, H_{k_{PA}}, H_{PA_1}) \neq \left(1, H'_{k_{\widetilde{PA}}}, H'_{\widetilde{PA}_1}\right)$ or $\left(k_{\widetilde{PB}}, H_{k_{\widetilde{PB}}}, H_{\widetilde{PB}_1}\right) \neq \left(1, H'_{k_{PB}}, H'_{PB_1}\right)$ **then**
- $Itertype \leftarrow \mathrm{PD}$;
- $RewindExtend \leftarrow \mathsf{R}$;
- $NewMetaA \leftarrow \mathsf{C}$;
- $FullMA \leftarrow (FullMA, NewMetaA)$;
- $\ell_{MA} \leftarrow \ell_{MA} + 1$;
- $\widetilde{NewMetaB} \leftarrow \mathsf{C}$;
- $\widetilde{MB} \leftarrow \left(\widetilde{MB}, \widetilde{NewMetaB}\right)$;
- $\ell_{\widetilde{MB}} \leftarrow \ell_{\widetilde{MB}} + 1$;
- $msg \leftarrow$ dummy message of length r;

▷ `Case vi.B`

else if $(\ell_{PA} < 6q_{MA} \cdot r)$ or $\left(\ell_{\widetilde{PB}} < 6q_{\widetilde{MB}} \cdot r\right)$ **then**
- $Itertype \leftarrow \mathrm{PD}$;
- $RewindExtend \leftarrow \mathsf{E}$;
- $NewMetaA \leftarrow \mathsf{C}$;
- $FullMA \leftarrow (FullMA, NewMetaA)$;
- $\ell_{MA} \leftarrow \ell_{MA} + 1$;
- $\widetilde{NewMetaB} \leftarrow \mathsf{C}$;
- $\widetilde{MB} \leftarrow \left(\widetilde{MB}, \widetilde{NewMetaB}\right)$;
- $\ell_{\widetilde{MB}} \leftarrow \ell_{\widetilde{MB}} + 1$;
- **if** $\ell_{PA} < 6q_{MA} \cdot r$ **then**
  - $msg \leftarrow FullPA\,[\ell_{PA} + 1, \ell_{PA} + r]$

Algorithm 29: Q-Preprocess-small-alphabet (Alice's side, cont. from previous page)

▷ `Case vii`

else
  Q-Computejointstate;
  $FullMA = (FullMA, NewMetaA)$;
  $\ell_{MA} \leftarrow \ell_{MA} + 1$;
  $\widetilde{MB} \leftarrow \left(\widetilde{MB}, \widetilde{NewMetaB}\right)$;
  $\ell_{\widetilde{MB}} \leftarrow \ell_{\widetilde{MB}} + 1$;
return Q-Preprocess-small-alphabet;

Algorithm 30: Q-Initialization-small-alphabet (Alice's side)

Initialize
  $L_{\text{QVC}} \leftarrow \Theta(n\epsilon)$;
  $r \leftarrow \Theta(1/\sqrt{\epsilon})$;
  $R_{\text{total}} \leftarrow \lceil n/2r + \Theta(n\epsilon) \rceil$;
  $t \leftarrow \Theta(n\epsilon)$;
  $k_{MA}, k_{\widetilde{MB}}, k_{PA}, k_{\widetilde{PB}} \leftarrow 1$;
  $E_{MA}, E_{\widetilde{MB}}, E_{PA}, E_{\widetilde{PB}} \leftarrow 0$;
  $v_{MA_1}, v_{MA_2}, v_{\widetilde{MB}_1}, v_{\widetilde{MB}_2}, v_{PA_1}, v_{PA_2}, v_{\widetilde{PB}_1}, v_{\widetilde{PB}_2} \leftarrow 0$;
  $\ell_{MA}, \ell_{\widetilde{MB}}, \ell_{PA}, \ell_{\widetilde{PB}}, \ell^{\text{A}}_{\text{QVC}}, \ell_{RA}, \ell_{\text{RecycleA}} \leftarrow 0$;
  $FullMA, \widetilde{MB}, FullPA, \widetilde{PB}, RA, IndexA \leftarrow \emptyset$;
$h \leftarrow$ iphash function from Definition 5.1.2 with $L = \Theta(n), o = \Theta(1), s = \Theta(n)$;
Robust Entanglement Distribution ; `// Distribute` $\Theta(n\sqrt{\epsilon})$ `MESs`
Reserve $L_{\text{QVC}} \cdot 4r$ MES pairs to be used as QVC encryption keys ;
Reserve $10R_{\text{total}}$ MESs to be used in quantum hashing ;
Measure the remaining $\Theta(n\sqrt{\epsilon})$ MESs in the computational basis and record the binary representation of the outcomes in R' ;
Stretch R' to a δ-biased pseudo-random string of length $\Theta(R_{\text{total}}(rL_{\text{QVC}} + n))$ using the deterministic algorithm of Lemma 2.4.8 where $\delta = 2^{-\Theta\left(\frac{n}{r}\right)}$;
Divide the pseudo-random string into $S = S_1 \cdots S_{12R_{\text{total}}}$ and $R = R_1 \cdots R_{10R_{\text{total}}}$;
  `//` $12R_{\text{total}}$ `seeds of length` $\Theta(n)$ `bits used by iphash`
  `//` $10R_{\text{total}}$ `seeds of length` $4rL_{\text{QVC}}$ `bits used in quantum hashing`
return Q-Initialization-small-alphabet;

Algorithm 31: MPhash$(k, \ell, S_1, S_2, T, \theta)$

$\hat{k} \leftarrow 2^{\lfloor \log k \rfloor}$;
$mp1 \leftarrow \hat{k}\theta \left\lfloor \frac{\ell}{\hat{k}\theta} \right\rfloor$;
$mp2 \leftarrow mp1 - \hat{k}\theta$;
$T_1 \leftarrow T\,[1 : mp1]$;
$T_2 \leftarrow T\,[1 : mp2]$;
$H_1 \leftarrow h_{S_1}(T_1)$;
$H_2 \leftarrow h_{S_2}(T_2)$;
return $(\boldsymbol{mp1}, \boldsymbol{mp2}, \boldsymbol{T_1}, \boldsymbol{T_2}, \boldsymbol{H_1}, \boldsymbol{H_2})$;

Algorithm 32: MPrewind$(E, k, H_k, H'_k, H_1, H_2, H'_1, H'_2, mp1, mp2, v_1, v_2, \ell)$

$\hat{k} \leftarrow 2^{\lfloor \log k \rfloor}$;
if $H_k \neq H'_k$ **then**
    $E \leftarrow E + 1$;
                                                ▷ `Voting phase`
else
    if $H_1 \in \{H'_1, H'_2\}$ **then**
        $v_1 \leftarrow v_1 + 1$;
    else if $H_2 \in \{H'_1, H'_2\}$ **then**
        $v_2 \leftarrow v_2 + 1$;
                                                ▷ `Transition phase`
if $E > k/2$ **then**
    Reset Status: $k, E, v_1, v_2 \leftarrow 0$; `// Mismatch transition`
else if $k = \hat{k}$ **then**
    if $v_1 \geq 0.4\hat{k}$ **then**
        $\ell \leftarrow mp1$;
        Reset Status: $k, E, v_1, v_2 \leftarrow 0$; `// Meeting point transition`
    else if $v_2 \geq 0.4\hat{k}$ **then**
        $\ell \leftarrow mp2$;
        Reset Status: $k, E, v_1, v_2 \leftarrow 0$; `// Meeting point transition`
    else
        $v_1, v_2 \leftarrow 0$;
        `// Disagreement length is too long for current rewinding step size`
$k \leftarrow k + 1$;
return $(\boldsymbol{E}, \boldsymbol{k}, \boldsymbol{v_1}, \boldsymbol{v_2}, \boldsymbol{\ell})$;

Algorithm 33: **rewindMD-small-alphabet** (Alice's side)

Input:

$$\begin{pmatrix} E_{MA}, k_{MA}, H_{k_{MA}}, H'_{k_{\widetilde{MA}}}, H_{MA_1}, H_{MA_2}, H'_{\widetilde{MA}_1}, H'_{\widetilde{MA}_2}, mp1_{MA}, mp2_{MA}, v_{MA_1}, v_{MA_2}, \ell_{MA} \\ E_{\widetilde{MB}}, k_{\widetilde{MB}}, H_{k_{\widetilde{MB}}}, H'_{k_{MB}}, H_{\widetilde{MB}_1}, H_{\widetilde{MB}_2}, H'_{MB_1}, H'_{MB_2}, mp1_{\widetilde{MB}}, mp2_{\widetilde{MB}}, v_{\widetilde{MB}_1}, v_{\widetilde{MB}_2}, \ell_{\widetilde{MB}} \end{pmatrix}$$

Promise: $(k_{MA}, H_{k_{MA}}, H_{MA_1}) \neq \left(1, H'_{k_{\widetilde{MA}}}, H'_{\widetilde{MA}_1}\right)$ or
$\left(k_{\widetilde{MB}}, H_{k_{\widetilde{MB}}}, H_{\widetilde{MB}_1}\right) \neq \left(1, H'_{k_{MB}}, H'_{MB_1}\right)$

Output: $\left(E_{MA}, k_{MA}, v_{MA_1}, v_{MA_2}, \ell_{MA}, E_{\widetilde{MB}}, k_{\widetilde{MB}}, v_{\widetilde{MB}_1}, v_{\widetilde{MB}_2}, \ell_{\widetilde{MB}}\right)$

if $(k_{MA}, H_{k_{MA}}, H_{MA_1}) \neq \left(1, H'_{k_{\widetilde{MA}}}, H'_{\widetilde{MA}_1}\right)$ **then**
  $(E_{MA}, k_{MA}, v_{MA_1}, v_{MA_2}, \ell_{MA}) \leftarrow$ **MPrewind**$\Big(E_{MA}, k_{MA}, H_{k_{MA}}, H'_{k_{\widetilde{MA}}}, H_{MA_1}, H_{MA_2}, H'_{\widetilde{MA}_1}, H'_{\widetilde{MA}_2}, mp1_{MA}, mp2_{MA}, v_{MA_1}, v_{MA_2}, \ell_{MA}\Big)$;

if $\left(k_{\widetilde{MB}}, H_{k_{\widetilde{MB}}}, H_{\widetilde{MB}_1}\right) \neq \left(1, H'_{k_{MB}}, H'_{MB_1}\right)$ **then**
  $\left(E_{\widetilde{MB}}, k_{\widetilde{MB}}, v_{\widetilde{MB}_1}, v_{\widetilde{MB}_2}, \ell_{\widetilde{MB}}\right) \leftarrow$ **MPrewind**$\Big(E_{\widetilde{MB}}, k_{\widetilde{MB}}, H_{k_{\widetilde{MB}}}, H'_{k_{MB}}, H_{\widetilde{MB}_1}, H_{\widetilde{MB}_2}, H'_{MB_1}, H'_{MB_2}, mp1_{\widetilde{MB}}, mp2_{\widetilde{MB}}, v_{\widetilde{MB}_1}, v_{\widetilde{MB}_2}, \ell_{\widetilde{MB}}\Big)$;

return **rewindMD-small-alphabet**;

Algorithm 34: **rewindPD-small-alphabet** (Alice's side)

Input:

$$\begin{pmatrix} E_{PA}, k_{PA}, H_{k_{PA}}, H'_{k_{\widetilde{PA}}}, H_{PA_1}, H_{PA_2}, H'_{\widetilde{PA}_1}, H'_{\widetilde{PA}_2}, mp1_{PA}, mp2_{PA}, v_{PA_1}, v_{PA_2}, \ell_{PA} \\ E_{\widetilde{PB}}, k_{\widetilde{PB}}, H_{k_{\widetilde{PB}}}, H'_{k_{PB}}, H_{\widetilde{PB}_1}, H_{\widetilde{PB}_2}, H'_{PB_1}, H'_{PB_2}, mp1_{\widetilde{PB}}, mp2_{\widetilde{PB}}, v_{\widetilde{PB}_1}, v_{\widetilde{PB}_2}, \ell_{\widetilde{PB}} \end{pmatrix}$$

Promise: $(k_{PA}, H_{k_{PA}}, H_{PA_1}) \neq \left(1, H'_{k_{\widetilde{PA}}}, H'_{\widetilde{PA}_1}\right)$ or
$\left(k_{\widetilde{PB}}, H_{k_{\widetilde{PB}}}, H_{\widetilde{PB}_1}\right) \neq \left(1, H'_{k_{PB}}, H'_{PB_1}\right)$

Output: $\left(E_{PA}, k_{PA}, v_{PA_1}, v_{PA_2}, \ell_{PA}, E_{\widetilde{PB}}, k_{\widetilde{PB}}, v_{\widetilde{PB}_1}, v_{\widetilde{PB}_2}, \ell_{\widetilde{PB}}\right)$

if $(k_{PA}, H_{k_{PA}}, H_{PA_1}) \neq \left(1, H'_{k_{\widetilde{PA}}}, H'_{\widetilde{PA}_1}\right)$ **then**
  $(E_{PA}, k_{PA}, v_{PA_1}, v_{PA_2}, \ell_{PA}) \leftarrow$ **MPrewind**$\Big(E_{PA}, k_{PA}, H_{k_{PA}}, H'_{k_{\widetilde{PA}}}, H_{PA_1}, H_{PA_2}, H'_{\widetilde{PA}_1}, H'_{\widetilde{PA}_2}, mp1_{PA}, mp2_{PA}, v_{PA_1}, v_{PA_2}, \ell_{PA}\Big)$;

Algorithm 34: rewindPD-small-alphabet (Alice's side, cont. from previous page)

3 **if** $\left(k_{\widetilde{PB}}, H_{k_{\widetilde{PB}}}, H_{\widetilde{PB}_1}\right) \neq \left(1, H'_{k_{PB}}, H'_{PB_1}\right)$ **then**

4 $\quad \left(E_{\widetilde{PB}}, k_{\widetilde{PB}}, v_{\widetilde{PB}_1}, v_{\widetilde{PB}_2}, \ell_{\widetilde{PB}}\right) \leftarrow$ **MPrewind**$\big(E_{\widetilde{PB}}, k_{\widetilde{PB}}, H_{k_{\widetilde{PB}}}, H'_{k_{PB}}, H_{\widetilde{PB}_1}, H_{\widetilde{PB}_2}, H'_{PB_1}, H'_{PB_2}, mp1_{\widetilde{PB}}, mp2_{\widetilde{PB}}, v_{\widetilde{PB}_1}, v_{\widetilde{PB}_2}, \ell_{\widetilde{PB}}\big)$;

5 **return rewindPD-small-alphabet**;

5.5 Analysis

As in the large alphabet case, our analysis relies on a potential function argument. In order to track the simulation progress and show the correctness of the algorithm, we condition on some view of the local classical data recorded by the two parties. Without loss of generality, as in Section 4.5, we analyze the algorithm assuming that the error introduced by the adversary on the n' message registers exchanged during the simulation is an arbitrary Pauli error of weight at most $\epsilon n'$.

The overall proof structure is exactly the same as the large alphabet case. Using an inductive argument, we show that recycling is successful throughout the execution of the algorithm. Moreover, we show that the number of iterations with a hash collision and the number of recovery iterations are both of the same order as the number of transmission errors introduced by the adversary, i.e., $\mathrm{O}(n\epsilon)$. The proof then proceeds by showing that in every iteration, the potential function Φ increases by at least 1, if no error or hash collision occurs; while it decreases by at most a constant otherwise. For R_{total} sufficiently large, we show that once the algorithm terminates, the potential function Φ is large enough to imply successful simulation.

The overall potential function Φ is defined as

$$\Phi \stackrel{\text{def}}{=} \Phi_{\mathrm{Q}} - \Phi_{\mathrm{MD}} - \Phi_{\mathrm{RD}} - \Phi_{\mathrm{PD}} \ , \tag{5.1}$$

where Φ_{Q} and Φ_{RD} are defined as in Section 4.5. The potential functions Φ_{MD} and Φ_{PD}, however, are modified to measure the progress in synchronizing the metadata and the Pauli data using meeting point-based rewinding.

We recall the following definitions from Section 4.5:

$$md_+^{\mathrm{A}} \stackrel{\text{def}}{=} \text{ the length of the longest prefix where } MA \text{ and } \widetilde{MA} \text{ agree;} \tag{5.2}$$

$$md_+^{\mathrm{B}} \stackrel{\text{def}}{=} \text{ the length of the longest prefix where } MB \text{ and } \widetilde{MB} \text{ agree;} \tag{5.3}$$

$$md_-^{\mathrm{A}} \stackrel{\text{def}}{=} \max\{\ell_{MA}, \ell_{\widetilde{MA}}\} - md_+^{\mathrm{A}}; \tag{5.4}$$

$$md_-^{\mathrm{B}} \stackrel{\text{def}}{=} \max\{\ell_{MB}, \ell_{\widetilde{MB}}\} - md_+^{\mathrm{B}}; \tag{5.5}$$

$$pd_+^{\mathrm{A}} \stackrel{\text{def}}{=} \lfloor \frac{1}{r} \times \text{ the length of the longest prefix where } PA \text{ and } \widetilde{PA} \text{ agree} \rfloor; \tag{5.6}$$

$$pd_{+}^{\mathrm{B}} \stackrel{\text{def}}{=} \lfloor \frac{1}{r} \times \text{ the length of the longest prefix where } PB \text{ and } \widetilde{PB} \text{ agree} \rfloor; \tag{5.7}$$

$$pd_{-}^{\mathrm{A}} \stackrel{\text{def}}{=} \frac{1}{r} \max\{\ell_{PA}, \ell_{\widetilde{PA}}\} - pd_{+}^{\mathrm{A}}; \tag{5.8}$$

$$pd_{-}^{\mathrm{B}} \stackrel{\text{def}}{=} \frac{1}{r} \max\{\ell_{PB}, \ell_{\widetilde{PB}}\} - pd_{+}^{\mathrm{B}}; \tag{5.9}$$

$$\begin{aligned} rd^{+} \stackrel{\text{def}}{=} \max\{j :\ & j \leq \min\{\ell_{RA}, \ell_{RB}\}\ ,\ RA\,[1:j] = RB\,[1:j] \\ & , W_k = 0^{4r} \text{ for all } k \leq j \text{ with } RA\,[k] = \mathsf{S}\}; \end{aligned} \tag{5.10}$$

$$rd^{-} \stackrel{\text{def}}{=} \max\{\ell_{RA}, \ell_{RB}\} - rd^{+}; \tag{5.11}$$

where W in Eq. (5.10) is the string corresponding to the error syndrome defined in Subsection 4.3.5. Also, recall that

$$g \stackrel{\text{def}}{=} \text{the number of good blocks in } JS2, \tag{5.12}$$

$$b \stackrel{\text{def}}{=} \text{the number of bad blocks in } JS2, \text{ and} \tag{5.13}$$

$$u \stackrel{\text{def}}{=} |\ell_{\mathrm{QVC}}^{\mathrm{A}} - \ell_{\mathrm{QVC}}^{\mathrm{B}}|. \tag{5.14}$$

The potential functions Φ_{MD} and Φ_{PD} now involve new variables which allow us to track the progress made in meeting point-based rewinding. We define

$$k_{\mathrm{MD}}^{\mathrm{A}} \stackrel{\text{def}}{=} k_{MA} + k_{\widetilde{MA}}; \tag{5.15}$$

$$k_{\mathrm{MD}}^{\mathrm{B}} \stackrel{\text{def}}{=} k_{MB} + k_{\widetilde{MB}}; \tag{5.16}$$

$$E_{\mathrm{MD}}^{\mathrm{A}} \stackrel{\text{def}}{=} E_{MA} + E_{\widetilde{MA}}; \tag{5.17}$$

$$E_{\mathrm{MD}}^{\mathrm{B}} \stackrel{\text{def}}{=} E_{MB} + E_{\widetilde{MB}}. \tag{5.18}$$

Similarly,

$$k_{\mathrm{PD}}^{\mathrm{A}} \stackrel{\text{def}}{=} k_{PA} + k_{\widetilde{PA}}; \tag{5.19}$$

$$k_{\mathrm{PD}}^{\mathrm{B}} \stackrel{\text{def}}{=} k_{PB} + k_{\widetilde{PB}}; \tag{5.20}$$

$$E_{\mathrm{PD}}^{\mathrm{A}} \stackrel{\text{def}}{=} E_{PA} + E_{\widetilde{PA}}; \tag{5.21}$$

$$E_{\mathrm{PD}}^{\mathrm{B}} \stackrel{\text{def}}{=} E_{PB} + E_{\widetilde{PB}}. \tag{5.22}$$

We introduce variables $BV_{\mathrm{MD}}^{\mathrm{A}}$, $BV_{\mathrm{MD}}^{\mathrm{B}}$, $BV_{\mathrm{PD}}^{\mathrm{A}}$, and $BV_{\mathrm{PD}}^{\mathrm{B}}$ which keep track of the number of iterations with a *bad voting step*. These variables account for the errors and collisions happening in the voting phase of the subroutine MPrewind. In each iteration, $BV_{\mathrm{MD}}^{\mathrm{A}}$ increases by 1 if both Alice and Bob conduct the subroutine MPrewind on the rewinding data corresponding to MA and $\widetilde{MA}$, respectively, and for some $j \in \{1, 2\}$, at least one of the following happens:

- $MA_j \notin \left\{\widetilde{MA}_1, \widetilde{MA}_2\right\}$ but Alice increases v_{MA_j} in the voting phase,

- $\widetilde{MA}_j \notin \{MA_1, MA_2\}$ but Bob increases $v_{\widetilde{MA}_j}$ in the voting phase,
- $MA_j \in \left\{\widetilde{MA}_1, \widetilde{MA}_2\right\}$ but despite conducting the voting phase, Alice does not increase v_{MA_j},
- $\widetilde{MA}_j \in \{MA_1, MA_2\}$ but despite conducting the voting phase, Bob does not increase $v_{\widetilde{MA}_j}$.

We reset $BV^{\mathrm{A}}_{\mathrm{MD}}$ to 0 whenever a status reset occurs on the rewinding data corresponding to MA or $\widetilde{MA}$. The variables $BV^{\mathrm{B}}_{\mathrm{MD}}$, $BV^{\mathrm{A}}_{\mathrm{PD}}$, and $BV^{\mathrm{B}}_{\mathrm{PD}}$ are defined similarly. Note that the values of these variables are not known to either party.

We also define the variables $BI^{\mathrm{A}}_{\mathrm{PD}}$ and $BI^{\mathrm{B}}_{\mathrm{PD}}$ to facilitate the analysis of the algorithm. In each iteration, the variable $BI^{\mathrm{A}}_{\mathrm{PD}}$ increases by 1 if exactly one of the following occurs:

- Alice conducts the subroutine MPrewind on the rewinding data corresponding to MA.
- Bob conducts the subroutine MPrewind on the rewinding data corresponding to $\widetilde{MA}$.

We reset $BI^{\mathrm{A}}_{\mathrm{PD}}$ to 0 whenever a status reset occurs on the rewinding data corresponding to MA or $\widetilde{MA}$. The variable $BI^{\mathrm{B}}_{\mathrm{PD}}$ is defined similarly.

Finally, we are ready to define the components of the potential function Φ. Let

$$1 < \lambda_1 < \lambda_2 < \lambda_3 < \lambda_4 < \lambda_5 \ ,$$

be constants chosen such that λ_k is sufficiently large depending only on λ_j with $j < k$. At the end of the i-th iteration, we let

$$\Phi_{\mathrm{MD}} \stackrel{\mathrm{def}}{=} \Phi^{\mathrm{A}}_{\mathrm{MD}} + \Phi^{\mathrm{B}}_{\mathrm{MD}} \ , \tag{5.23}$$

where

$$\Phi^{\mathrm{A}}_{\mathrm{MD}} \stackrel{\mathrm{def}}{=} \begin{cases} i - md^{\mathrm{A}}_{+} + \lambda_2 md^{\mathrm{A}}_{-} - \lambda_1\left(k^{\mathrm{A}}_{\mathrm{MD}} - 2\right) + \lambda_4 E^{\mathrm{A}}_{\mathrm{MD}} + \lambda_5 BV^{\mathrm{A}}_{\mathrm{MD}} \ , & \text{if } k_{MA} = k_{\widetilde{MA}} \quad (5.24a) \\ i - md^{\mathrm{A}}_{+} + \lambda_2 md^{\mathrm{A}}_{-} + 0.9\lambda_3\left(k^{\mathrm{A}}_{\mathrm{MD}} - 2\right) - \lambda_3 E^{\mathrm{A}}_{\mathrm{MD}} + \lambda_5 BV^{\mathrm{A}}_{\mathrm{MD}} \ , & \text{if } k_{MA} \neq k_{\widetilde{MA}} \quad (5.24b) \end{cases}$$

$$\Phi^{\mathrm{B}}_{\mathrm{MD}} \stackrel{\mathrm{def}}{=} \begin{cases} i - md^{\mathrm{B}}_{+} + \lambda_2 md^{\mathrm{B}}_{-} - \lambda_1\left(k^{\mathrm{B}}_{\mathrm{MD}} - 2\right) + \lambda_4 E^{\mathrm{B}}_{\mathrm{MD}} + \lambda_5 BV^{\mathrm{B}}_{\mathrm{MD}} \ , & \text{if } k_{MB} = k_{\widetilde{MB}} \quad (5.25a) \\ i - md^{\mathrm{B}}_{+} + \lambda_2 md^{\mathrm{B}}_{-} + 0.9\lambda_3\left(k^{\mathrm{B}}_{\mathrm{MD}} - 2\right) - \lambda_3 E^{\mathrm{B}}_{\mathrm{MD}} + \lambda_5 BV^{\mathrm{B}}_{\mathrm{MD}} \ , & \text{if } k_{MB} \neq k_{\widetilde{MB}} \quad (5.25b) \end{cases}$$

Similarly, for constants

$$1 < \eta_1 < \eta_2 < \eta_3 < \eta_4 < \eta_5 < \eta_6 \ ,$$

chosen such that η_k is sufficiently large depending only on η_j with $j < k$, we define

$$\Phi_{\mathrm{PD}} \stackrel{\mathrm{def}}{=} \Phi^{\mathrm{A}}_{\mathrm{PD}} + \Phi^{\mathrm{B}}_{\mathrm{PD}} \ , \tag{5.26}$$

where

$$\Phi^{\mathrm{A}}_{\mathrm{PD}} \stackrel{\mathrm{def}}{=} \begin{cases} 6q_{MA} - pd^{\mathrm{A}}_{+} + \eta_2 pd^{\mathrm{A}}_{-} - \eta_1\left(k^{\mathrm{A}}_{\mathrm{PD}} - 2\right) + \eta_4 E^{\mathrm{A}}_{\mathrm{PD}} + \eta_5 BV^{\mathrm{A}}_{\mathrm{PD}} + \eta_6 BI^{\mathrm{A}}_{\mathrm{PD}} \ , \\ \qquad\qquad \text{if } k_{PA} = k_{\widetilde{PA}} \quad (5.27\text{a}) \\ 6q_{MA} - pd^{\mathrm{A}}_{+} + \eta_2 pd^{\mathrm{A}}_{-} + 0.9\eta_3\left(k^{\mathrm{A}}_{\mathrm{PD}} - 2\right) - \eta_3 E^{\mathrm{A}}_{\mathrm{PD}} + \eta_5 BV^{\mathrm{A}}_{\mathrm{PD}} + \eta_6 BI^{\mathrm{A}}_{\mathrm{PD}} \ , \\ \qquad\qquad \text{if } k_{PA} \neq k_{\widetilde{PA}} \quad (5.27\text{b}) \end{cases}$$

$$\Phi^{\mathrm{B}}_{\mathrm{PD}} \stackrel{\mathrm{def}}{=} \begin{cases} 6q_{MB} - pd^{\mathrm{B}}_{+} + \eta_2 pd^{\mathrm{B}}_{-} - \eta_1\left(k^{\mathrm{B}}_{\mathrm{PD}} - 2\right) + \eta_4 E^{\mathrm{B}}_{\mathrm{PD}} + \eta_5 BV^{\mathrm{B}}_{\mathrm{PD}} + \eta_6 BI^{\mathrm{B}}_{\mathrm{PD}} \ , \\ \qquad\qquad \text{if } k_{PB} = k_{\widetilde{PB}} \quad (5.28\text{a}) \\ 6q_{MB} - pd^{\mathrm{B}}_{+} + \eta_2 pd^{\mathrm{B}}_{-} + 0.9\eta_3\left(k^{\mathrm{B}}_{\mathrm{PD}} - 2\right) - \eta_3 E^{\mathrm{B}}_{\mathrm{PD}} + \eta_5 BV^{\mathrm{B}}_{\mathrm{PD}} + \eta_6 BI^{\mathrm{B}}_{\mathrm{PD}} \ , \\ \qquad\qquad \text{if } k_{PB} \neq k_{\widetilde{PB}} \quad (5.28\text{b}) \end{cases}$$

We remark that the definitions above for Φ_{MD} and Φ_{PD} are modified versions of a similar potential function appearing in Ref. [37]. Our definition slightly simplifies the proof and in the case of Φ_{PD} allows us to deal with out-of-sync rewinding scenarios which we will discuss in detail.

Finally, Φ_{RD} and Φ_{Q} are defined as

$$\Phi_{\mathrm{RD}} \stackrel{\mathrm{def}}{=} \ell^{\mathrm{A}}_{\mathrm{QVC}} + \ell^{\mathrm{B}}_{\mathrm{QVC}} + 13rd^{-} - 2rd^{+} \ , \tag{5.29}$$

$$\Phi_{\mathrm{Q}} \stackrel{\mathrm{def}}{=} g - b - 9u \ . \tag{5.30}$$

In order to avoid ambiguity, we may use a superscript i to indicate the value of the variables of the algorithm at the *end* of the i-th iteration. For instance, we denote Alice's recycling data at the end of the i-th iteration by RA^i. Our analysis in this section involves constants

$$c_1 < c_2 < c_3 < c_4 < c_5 < c_6 < c_7 < c_8 < c_9 \ ,$$

chosen such that c_i is sufficiently large depending only on c_j with $j < i$.

Lemma 5.5.1. *Throughout the algorithm, we have*

- $\Phi_{\mathrm{MD}} \in -\mathrm{O}(n\epsilon)$.
- $\Phi_{\mathrm{RD}} \geq 0$ *with equality if and only if Alice and Bob have used the same number of MES blocks* ($\ell^{\mathrm{A}}_{\mathrm{QVC}} = \ell^{\mathrm{B}}_{\mathrm{QVC}}$)*, their measurement pointers* ℓ_{RA} *and* ℓ_{RB} *agree, they fully agree on the recycling data* ($RA = RB$)*, and* $W_k = 0^{4r}$ *for all* $k \leq \ell^{\mathrm{A}}_{\mathrm{QVC}}$ *with* $RA\,[k] = \mathsf{S}$*, i.e.,* $\ell^{\mathrm{A}}_{\mathrm{QVC}} = \ell^{\mathrm{B}}_{\mathrm{QVC}} = rd^{+}$ *and* $rd^{-} = 0$.

- $\Phi_{\mathrm{PD}} \in -\mathrm{O}(n\epsilon)$.

Proof. The variables appearing in the potential function $\Phi^{\mathrm{A}}_{\mathrm{MD}}$ are always non-negative. Moreover, at the end of the i-th iteration we always have $md^{\mathrm{A}}_{+} \leq i$, $E_{MA} \leq k_{MA}/2$, and $E_{\widetilde{MA}} \leq k_{\widetilde{MA}}/2$. This implies that, at the end of any iteration, if $k_{MA} \neq k_{\widetilde{MA}}$, then

$$\Phi^{\mathrm{A}}_{\mathrm{MD}} \geq 0.9\lambda_3 \left(k^{\mathrm{A}}_{\mathrm{MD}} - 2\right) - \lambda_3 E^{\mathrm{A}}_{\mathrm{MD}} \geq \lambda_3 \left(0.9k^{\mathrm{A}}_{\mathrm{MD}} - 0.5k^{\mathrm{A}}_{\mathrm{MD}} - 1.8\right) \geq -\lambda_3 \ ,$$

where the last inequality follows since $k^{\mathrm{A}}_{\mathrm{MD}} \geq 2$. Let $\lambda_2 = 8\lambda_1$ and suppose that $k_{MA} = k_{\widetilde{MA}} = k$ for some $k \geq 1$. Then $\Phi^{\mathrm{A}}_{\mathrm{MD}} < 0$ implies that $k > 4md^{\mathrm{A}}_{-}$. However, the rewinding step size cannot grow so large compared to md^{A}_{-} unless too many transmission errors occur. Note that, in every iteration, the variables k_{MA} and $k_{\widetilde{MA}}$ each either increase by 1, or get reset to 1, or remain unchanged at 1. Therefore, when $k_{MA} = k_{\widetilde{MA}} = k$, in the last k iteration both parties have had matching step sizes, increasing from 1 to k. So k_{MA} and $k_{\widetilde{MA}}$ can be so large compared to md^{A}_{-} only if in at least 0.2 of the iterations in which the step sizes increased from $2^{\lfloor \log md^{\mathrm{A}}_{-} \rfloor + 1}$ to $2^{\lfloor \lg k \rfloor}$, the step size hashes or the meeting point hashes were corrupted to appear non-matching. Since the total number of errors is at most $2n\epsilon$, this implies that

$$2n\epsilon \geq 0.2 \left(2^{\lfloor \lg k \rfloor} - 2^{\lfloor \log md^{\mathrm{A}}_{-} \rfloor + 1}\right) \geq 0.2 \left(k/2 - 2md^{\mathrm{A}}_{-}\right) \ .$$

Hence, we have

$$\Phi^{\mathrm{A}}_{\mathrm{MD}} \geq \lambda_1 \left(8md^{\mathrm{A}}_{-} - 2k\right) \geq -40\lambda_1 n\epsilon \ .$$

Similarly, one can show that $\Phi^{\mathrm{B}}_{\mathrm{MD}} \geq -40\lambda_1 n\epsilon$, concluding the proof of the first statement.

The second statement holds since $rd^{-} \geq 0$, $rd^{+} \leq \min\{\ell_{RA}, \ell_{RB}\}$ and the property that $\ell_{RA} \leq \ell^{\mathrm{A}}_{\mathrm{QVC}}$ and $\ell_{RB} \leq \ell^{\mathrm{B}}_{\mathrm{QVC}}$.

Finally, we prove the last statement. Note that for Pauli data we have $pd^{\mathrm{A}}_{+} \leq 6q_{MA}$ and $pd^{\mathrm{B}}_{+} \leq 6q_{MB}$. Moreover, at the end of every iteration, we always have $E_{PA} \leq k_{PA}/2$, and $E_{\widetilde{PA}} \leq k_{\widetilde{PA}}/2$. Therefore, when $k_{PA} \neq k_{\widetilde{PA}}$,

$$\Phi^{\mathrm{A}}_{\mathrm{PD}} \geq 0.9\eta_3 \left(k^{\mathrm{A}}_{\mathrm{PD}} - 2\right) - \eta_3 E^{\mathrm{A}}_{\mathrm{PD}} \geq \eta_3 \left(0.9k^{\mathrm{A}}_{\mathrm{PD}} - 0.5k^{\mathrm{A}}_{\mathrm{PD}} - 1.8\right) \geq -\eta_3 \ ,$$

where the last inequality holds since $k^{\mathrm{A}}_{\mathrm{PD}} \geq 2$.

Suppose that $k_{PA} = k_{\widetilde{PA}} = k$ for some $k \geq 1$. For η_4 sufficiently large compared to η_1, if $E^{\mathrm{A}}_{\mathrm{PD}} > k^{\mathrm{A}}_{\mathrm{PD}}/200$, then $\Phi^{\mathrm{A}}_{\mathrm{PD}} \geq 0$. Suppose that $E^{\mathrm{A}}_{\mathrm{PD}} \leq k^{\mathrm{A}}_{\mathrm{PD}}/200$ and let $\eta_2 = 16\eta_1$. Note that $\Phi^{\mathrm{A}}_{\mathrm{PD}} < 0$ implies that $k > 8pd^{\mathrm{A}}_{-}$. Similar to the argument used for the first statement, we show that the step sizes k_{PA} and $k_{\widetilde{PA}}$ cannot get too large under these conditions unless too many transmission errors occur. However, the proof is more complicated now since we no longer have the guarantee that k_{PA} and $k_{\widetilde{PA}}$ have been equal in the last k iterations. In fact, unlike k_{MA} and $k_{\widetilde{MA}}$, for the Pauli data, the step sizes k_{PA} and $k_{\widetilde{PA}}$ may remain unchanged at a value greater than 1 in an iteration.

We prove that $k \leq 50n\epsilon$, then we have

$$\Phi_{\mathrm{PD}}^{\mathrm{A}} \geq -2\eta_1 k \geq -100\eta_1 n\epsilon \ .$$

If $k \leq 10n\epsilon$ then we are done. Suppose that $k > 10n\epsilon$. Consider the last iteration in which at least one of k_{PA} or $k_{\widetilde{PA}}$ were reset. Note that in each of the following iterations the difference between k_{PA} and $k_{\widetilde{PA}}$ has either remained the same or changed by 1. In particular, the distance decreases by 1 only if the party with the smaller step size increases his/her step size by 1, while the step size for the other party remains the same. But this can happen only if a transmission error occurs. Let d be the maximum difference between k_{PA} and $k_{\widetilde{PA}}$ after the last time any of them were reset. Note that at least d transmission errors must have happened to decrease their difference to zero eventually. Let $\hat{k} = 2^{\lfloor k \rfloor}$. Consider the first iteration starting with $k_{PA}, k_{\widetilde{PA}} \geq \hat{k}/2$. Without loss of generality, suppose that Alice was the second party to reach the step size $\hat{k}/2$. In this iteration, the difference between k_{PA} and $k_{\widetilde{PA}}$ was at most d. Therefore, in at least $\hat{k}/2 - d$ of the following iterations we have $\hat{k}_{PA} = \hat{k}_{\widetilde{PA}} = \hat{k}/2$. Moreover, pd_{-}^{A} has remained the same in these iterations. Note that

$$\hat{k}/2 \geq k/4 > 2pd_{-}^{\mathrm{A}} \geq 2^{\lfloor \log pd_{-}^{\mathrm{A}} \rfloor + 1} \ .$$

Therefore, in at least $\hat{k}/2 - d$ iterations with $\hat{k}_{PA} = \hat{k}_{\widetilde{PA}} = \hat{k}/2$, the step size has been sufficiently large compared to pd_{-}^{A} to guarantee the existence of at least one matching meeting point. Alice did not reset her step size when k_{PA} reached the value $\hat{k}$, which means that her number of votes for the meeting point was strictly less than $0.8\hat{k}/2$. In at most E_{MA} iterations Alice has not entered the voting step. Let d' be the number of iterations in which Alice entered the voting step in the subroutine MPrewind but Bob's hash value corresponding to the correct meeting point was corrupted to appear non-matching. Then we have

$$\left(\hat{k}/2 - d\right) - E_{MA} - d' < 0.8\hat{k}/2 \ .$$

Since $E_{PA} \leq E_{\mathrm{PD}}^{\mathrm{A}} \leq \hat{k}/50$, this implies that $0.08\hat{k} < d + d' \leq 2n\epsilon$. Hence, we have $k \leq 2\hat{k} < 50n\epsilon$.

Similarly, one can prove that $\Phi_{\mathrm{PD}}^{\mathrm{B}} \geq -100\eta_1 n\epsilon$, which concludes the proof of the last statement. □

Definition 5.5.2. We say an iteration of Algorithm 28 suffers from a *metadata hash collision* if any of the following happens:

- $H_{k_{MA}} = H_{k_{\widetilde{MA}}}$ despite $k_{MA} \neq k_{\widetilde{MA}}$,
- $H_{MA_1} = H_{\widetilde{MA}_j}$ despite $MA_1 \neq \widetilde{MA}_j$, for $j \in \{1, 2\}$,
- $H_{MA_2} = H_{\widetilde{MA}_j}$ despite $MA_2 \neq \widetilde{MA}_j$, for $j \in \{1, 2\}$,
- $H_{k_{MB}} = H_{k_{\widetilde{MB}}}$ despite $k_{MB} \neq k_{\widetilde{MB}}$,

- $H_{MB_1} = H_{\widetilde{MB}_j}$ despite $MB_1 \neq \widetilde{MB}_j$, for $j \in \{1, 2\}$,
- $H_{MB_2} = H_{\widetilde{MB}_j}$ despite $MB_2 \neq \widetilde{MB}_j$, for $j \in \{1, 2\}$,

Note that we distinguish between, for instance, the first scenario above and when $H_{k_{MA}} = H'_{k_{\widetilde{MA}}}$ due to a transmission error on $H_{k_{\widetilde{MA}}}$, despite the two might have similar effects.

We recall the definition of successful recycling from Section 4.5.

Definition 5.5.3. We say recycling is successful in the i-th iteration of Algorithm 28, if the following hold:

- The algorithm does not abort in the i-th iteration, i.e., $NextIndexA^i, NextIndexB^i \neq \perp$,
- $NextIndexA^i = NextIndexB^i$,
- The block of MES registers indexed by $NextIndexA^i$ are in the $|\phi^{0,0}\rangle^{\otimes 4r}$ state at the beginning of the i-th iteration.

The conditions of Definition 5.5.3 are all satisfied in the first L_{QVC} iterations of the algorithm as well. Note that if recycling is successful in the first i iterations of the algorithm then $\ell^j_{\text{RecycleA}} = \ell^j_{\text{RecycleB}}$, for all $j \leq i$ and we have $IndexA^i = IndexB^i$. Moreover, for every $i \geq L_{\text{QVC}}$, we have $IndexA^i[1 : L_{\text{QVC}}] = IndexB^i[1 : L_{\text{QVC}}] = 1 : L_{\text{QVC}}$.

We use the same definition for quantum hash collision as in Section 4.5:

Definition 5.5.4. We say an iteration of Algorithm 28 suffers from a *quantum hash collision* when recycling has been successful so far, Alice and Bob know each other's metadata, have used the same number of MES blocks ($\ell^{\text{A}}_{\text{QVC}} = \ell^{\text{B}}_{\text{QVC}}$) and agree on their measurement pointers and their recycling data up to the measurement pointers ($\ell_{RA} = \ell_{RB}$ and $RA[1 : \ell_{RA}] = RB[1 : \ell_{RB}]$) but despite the fact that there is an undetected quantum error from earlier iterations ($W_k \neq 0^{4r}$ for some $k \leq \ell_{RA}$ with $RA[k] = \mathsf{S}$), their quantum hash values match, i.e., $QHA = QHB$. Note that we distinguish between the above scenario and when $QHA = QHB'$ due to a transmission error on QHB.

Definition 5.5.5. We say an iteration of Algorithm 28 suffers from a *Pauli data hash collision* when recycling has been successful so far, Alice and Bob know each other's metadata, agree on the number of MES blocks they have used ($\ell^{\text{A}}_{\text{QVC}} = \ell^{\text{B}}_{\text{QVC}}$), agree on their recycling data, their measurement pointers satisfy $\ell_{RA} = \ell_{RB} = \ell^{\text{A}}_{\text{QVC}}$, and all the non-measured MES blocks are in the $|\phi^{0,0}\rangle^{\otimes 4r}$ state but any of the following happens:

- $H_{k_{PA}} = H_{k_{\widetilde{PA}}}$ despite $k_{PA} \neq k_{\widetilde{PA}}$,
- $H_{PA_1} = H_{\widetilde{PA}_j}$ despite $PA_1 \neq \widetilde{PA}_j$, for $j \in \{1, 2\}$,

- $H_{PA_2} = H_{\widetilde{PA}_j}$ despite $PA_2 \neq \widetilde{PA}_j$, for $j \in \{1,2\}$,
- $H_{k_{PB}} = H_{k_{\widetilde{PB}}}$ despite $k_{PB} \neq k_{\widetilde{PB}}$,
- $H_{PB_1} = H_{\widetilde{PB}_j}$ despite $PB_1 \neq \widetilde{PB}_j$, for $j \in \{1,2\}$,
- $H_{PB_2} = H_{\widetilde{PB}_j}$ despite $PB_2 \neq \widetilde{PB}_j$, for $j \in \{1,2\}$.

We define $\Psi_{\mathrm{MD}}^{\mathrm{A}}$, $\Psi_{\mathrm{MD}}^{\mathrm{B}}$, and Ψ_{MD} at the end of the i-th iteration as follows:

$$\Psi_{\mathrm{MD}}^{\mathrm{A}} \stackrel{\mathrm{def}}{=} i - md_{+}^{\mathrm{A}} + md_{-}^{\mathrm{A}} + \left(k_{\mathrm{MD}}^{\mathrm{A}} - 2\right) \ , \tag{5.31}$$

$$\Psi_{\mathrm{MD}}^{\mathrm{B}} \stackrel{\mathrm{def}}{=} i - md_{+}^{\mathrm{B}} + md_{-}^{\mathrm{B}} + \left(k_{\mathrm{MD}}^{\mathrm{B}} - 2\right) \ , \tag{5.32}$$

$$\Psi_{\mathrm{MD}} \stackrel{\mathrm{def}}{=} \Psi_{\mathrm{MD}}^{\mathrm{A}} + \Psi_{\mathrm{MD}}^{\mathrm{B}} \ . \tag{5.33}$$

Note that we always have $\Psi_{\mathrm{MD}}^{\mathrm{A}}, \Psi_{\mathrm{MD}}^{\mathrm{B}} \geq 0$, and hence $\Psi_{\mathrm{MD}} \geq 0$.

Lemma 5.5.6. *Each iteration of Algorithm 28, regardless of the number of hash collisions and transmission errors, increases the potential functions $\Phi_{\mathrm{MD}}^{\mathrm{A}}$ and $\Phi_{\mathrm{MD}}^{\mathrm{B}}$ by at most a constant. Moreover, in any iteration with no transmission errors or hash collisions,*

- *if $\Psi_{\mathrm{MD}} = 0$ at the beginning of the iteration then Φ_{MD} remains the same,*
- *otherwise, if $\Psi_{\mathrm{MD}}^{\mathrm{A}} > 0$ then $\Phi_{\mathrm{MD}}^{\mathrm{A}}$ decreases by at least 1. Similarly, if $\Psi_{\mathrm{MD}}^{\mathrm{B}} > 0$ then $\Phi_{\mathrm{MD}}^{\mathrm{B}}$ decreases by at least 1.*

Proof. First note that in an iteration with no transmission errors, if $\Psi_{\mathrm{MD}} = 0$ at the beginning of the iteration, then Alice and Bob do not conduct the subroutine MPrewind on the rewinding data corresponding to MA and $\widetilde{MA}$, respectively. In this case, the iteration number, md_{+}^{A}, and md_{+}^{B} each increase by 1, while the remaining variables appearing in Φ_{MD} remain unchanged. Hence, Φ_{MD} remains the same.

In the remainder of the proof, we focus on $\Phi_{\mathrm{MD}}^{\mathrm{A}}$. The potential function $\Phi_{\mathrm{MD}}^{\mathrm{B}}$ behaves similarly. We consider all the remaining possible scenarios in the iteration at hand. We denote by md_{+}^{A}, md_{-}^{A}, k_{MA}, $k_{\widetilde{MA}}$, E_{MA}, $E_{\widetilde{MA}}$, and $BV_{\mathrm{MD}}^{\mathrm{A}}$ the values at the beginning of the iteration and denote by $\overline{md_{+}^{\mathrm{A}}}$, $\overline{md_{-}^{\mathrm{A}}}$, $\overline{k_{MA}}$, $\overline{k_{\widetilde{MA}}}$, $\overline{E_{MA}}$, $\overline{E_{\widetilde{MA}}}$, and $\overline{BV_{\mathrm{MD}}^{\mathrm{A}}}$ the values at the end of the iteration.

- **Scenario 1:** Alice and Bob do not conduct the subroutine MPrewind on the rewinding data corresponding to MA and $\widetilde{MA}$, respectively: In this case, at the beginning of the iteration, we have $k_{MA} = k_{\widetilde{MA}} = 1$, $H_{k_{MA}} = H'_{k_{\widetilde{MA}}}$, $H_{MA_1} = H'_{\widetilde{MA}_1}$, $H_{k_{\widetilde{MA}}} = H'_{k_{MA}}$, and $H_{\widetilde{MA}_1} = H'_{MA_1}$. The variables k_{MA} and $k_{\widetilde{MA}}$ remain unchanged, and $E_{\mathrm{MD}}^{\mathrm{A}}$, $BV_{\mathrm{MD}}^{\mathrm{A}}$ remain 0. The remaining variables appearing in $\Phi_{\mathrm{MD}}^{\mathrm{A}}$ vary by at most a constant.

Hence, regardless of the number of errors and hash collisions, $\Phi^{\mathrm{A}}_{\mathrm{MD}}$ increases by at most a constant. If the iteration is error and collision free and $\Psi^{\mathrm{A}}_{\mathrm{MD}} > 0$, then we have $md^{\mathrm{A}}_{-} = 0$ and $md^{\mathrm{A}}_{+} < i - 1$ at the beginning of the iteration. Both parties conduct Case i.B, the iteration number increases by 1, md^{A}_{+} increases by 2, and the remaining variables remain unchanged. Therefore, $\Phi^{\mathrm{A}}_{\mathrm{MD}}$ decreases by at least 1.

In the remaining scenarios, at least one of Alice or Bob conducts the subroutine MPrewind on the rewinding data corresponding to MA or $\widetilde{MA}$, respectively.

- **Scenario 2:** No status reset on the variables corresponding to MA and $\widetilde{MA}$ occurs: In this scenario, all the variables appearing in $\Phi^{\mathrm{A}}_{\mathrm{MD}}$ change by at most a constant and so does $\Phi^{\mathrm{A}}_{\mathrm{MD}}$. If the iteration has no errors or hash collisions then both parties conduct Case i.A. The variables $BV^{\mathrm{A}}_{\mathrm{MD}}, md^{\mathrm{A}}_{-}, md^{\mathrm{A}}_{+}$ remain unchanged while the iteration number, k_{MA}, and $k_{\widetilde{MA}}$ increase by 1. If $k_{MA} = k_{\widetilde{MA}}$ at the beginning of the iteration then $E^{\mathrm{A}}_{\mathrm{MD}}$ remains unchanged. Therefore, $\Phi^{\mathrm{A}}_{\mathrm{MD}}$ decreases by at least $2\lambda_1 - 1$, which is at least 1 for sufficiently large λ_1. Otherwise, if $k_{MA} \neq k_{\widetilde{MA}}$ then $E^{\mathrm{A}}_{\mathrm{MD}}$ increases by 2, leading to a potential decrease by at least $2\left(\lambda_3 - 0.9\lambda_3\right) - 1 = 0.2\lambda_3 - 1$, which is at least 1 for sufficiently large λ_3.

- **Scenario 3:** $k_{MA} \neq k_{\widetilde{MA}}$ and due to a mismatch transition or a meeting point transition, exactly one status reset on the variables corresponding to MA or $\widetilde{MA}$ occurs: Without loss of generality, suppose that Alice does the status reset. We consider the following two possibilities:

  - $\overline{k_{MA}} = \overline{k_{\widetilde{MA}}} = 1$: This can only happen in an iteration starting with $k_{\widetilde{MA}} = 1$ in which Bob does not conduct MPrewind on the rewinding data corresponding to $\widetilde{MA}$. Note that in this case $E_{\widetilde{MA}} = 0$. Bob's variables $k_{\widetilde{MA}}$ and $E_{\widetilde{MA}}$ remain unchanged. The variable md^{A}_{+} does not decrease due to Bob's actions, while he increases md^{A}_{-} by at most 2. On the other hand, Alice's variables k_{MA}, E_{MA}, and BV_{MA} get reset. The variable md^{A}_{+} decreases by at most $2k_{MA}$ due to Alice's actions and she increases md^{A}_{-} by at most $2k_{MA}$. The variable $BV^{\mathrm{A}}_{\mathrm{MD}}$ gets reset to 0. Therefore, we have

$$\begin{aligned}\Delta\Phi^{\mathrm{A}}_{\mathrm{MD}} &\leq 1 + 2k_{MA} + \lambda_2 \max\{2k_{MA}, 2\} - 0.9\lambda_3\left(k_{MA} - 1\right) + \lambda_3 E_{MA} \\ &\leq 1 - k_{MA}\left[0.9\lambda_3 - 0.5\lambda_3 - 2\lambda_2 - 2\right] + 0.9\lambda_3 - 0.5\lambda_3 \\ &= 1 + 0.4\lambda_3 - k_{MA}\left[0.4\lambda_3 - 2\lambda_2 - 2\right] \ ,\end{aligned}$$

    where the inequality holds since at the beginning of every iteration we always have $E_{MA} \leq 0.5k_{MA} - 0.5$. Since $k_{MA} \geq 2$, for sufficiently large λ_3, $\Phi^{\mathrm{A}}_{\mathrm{MD}}$ decreases by at least 1.

  - $\overline{k_{MA}} \neq \overline{k_{\widetilde{MA}}}$: In this case $k_{\widetilde{MA}}$ increases by 1, while $E_{\widetilde{MA}}$ increases by at most 1. Moreover, Bob's actions do not have any effect on the value of md^{A}_{+} and md^{A}_{-}.

So Bob's contribution increases $\Phi^{\mathrm{A}}_{\mathrm{MD}}$ by at most $0.9\lambda_3$. Using a similar argument as in the above case, Alice's contribution to $\Delta\Phi^{\mathrm{A}}_{\mathrm{MD}}$ is at most

$$0.4\lambda_3 - k_{MA}\left[0.4\lambda_3 - 2\lambda_2 - 2\right] \ ,$$

which is a decreasing function of k_{MA} for sufficiently large λ_3, and is equal to $2\lambda_2 + 2$ for $k_{MA} = 1$. Therefore, regardless of the number of errors and hash collisions, we have

$$\Delta\Phi^{\mathrm{A}}_{\mathrm{MD}} \leq 0.9\lambda_3 + 2\lambda_2 + 2 + 1 \ .$$

Suppose that the iteration has no transmission errors or hash collisions. Then $k_{\widetilde{MA}}$ and $E_{\widetilde{MA}}$ increase by 1. Therefore, Bob's contribution decreases $\Phi^{\mathrm{A}}_{\mathrm{MD}}$ by at least $0.1\lambda_3$. If $k_{MA} \geq 2$ then Alice's contribution also decreases $\Phi^{\mathrm{A}}_{\mathrm{MD}}$ by at least a constant. Therefore, $\Phi^{\mathrm{A}}_{\mathrm{MD}}$ decreases by at least 1. Otherwise, if $k_{MA} = 1$ at the beginning of the iteration, Alice's contribution increases $\Phi^{\mathrm{A}}_{\mathrm{MD}}$, but this is dominated by the decrease due to Bob's contribution. In more detail, in this case Alice's variable E_{MA} is 0 at the beginning of the iteration and it remain the same, k_{MA} remains at 1, md^{A}_{+} decreases by at most 2, and md^{A}_{-} increases by at most 2. Therefore, $\Phi^{\mathrm{A}}_{\mathrm{MD}}$ decreases by at least $0.1\lambda_3 - 2\lambda_2 - 2 - 1$, which is at least 1 for sufficiently large λ_3.

- **Scenario 4:** $k_{MA} \neq k_{\widetilde{MA}}$ and status resets occur on both sides due to mismatch or meeting point transitions: At the end of the iteration we have $\overline{k_{MA}} = \overline{k_{\widetilde{MA}}} = 1$ and $\overline{E^{\mathrm{A}}_{\mathrm{MD}}} = \overline{BV^{\mathrm{A}}_{\mathrm{MD}}} = 0$. At the beginning of every iteration, we have $E_{MA} \leq 0.5k_{MA} - 0.5$ and $E_{\widetilde{MA}} \leq 0.5k_{\widetilde{MA}} - 0.5$ and therefore, $E^{\mathrm{A}}_{\mathrm{MD}} \leq 0.5k^{\mathrm{A}}_{\mathrm{MD}} - 1$. Moreover, md^{A}_{+} decreases by at most $2\max\{k_{MA}, k_{\widetilde{MA}}\} \leq 2k^{\mathrm{A}}_{\mathrm{MD}}$ and similarly, md^{A}_{-} increases by at most $2k^{\mathrm{A}}_{\mathrm{MD}}$. Hence, we have

$$\begin{aligned}
\Delta\Phi^{\mathrm{A}}_{\mathrm{MD}} &\leq 1 + 2k^{\mathrm{A}}_{\mathrm{MD}} + \lambda_2\left(2k^{\mathrm{A}}_{\mathrm{MD}}\right) - 0.9\lambda_3\left(k^{\mathrm{A}}_{\mathrm{MD}} - 2\right) + \lambda_3 E^{\mathrm{A}}_{\mathrm{MD}} \\
&\leq -k^{\mathrm{A}}_{\mathrm{MD}}\left(0.9\lambda_3 - 0.5\lambda_3 - 2\lambda_2 - 2\right) + 1.8\lambda_3 - \lambda_3 + 1 \\
&\leq -0.4\lambda_3 + 6\lambda_2 + 7 \ ,
\end{aligned}$$

where the last inequality follows since $k^{\mathrm{A}}_{\mathrm{MD}} \geq 3$. Thus, $\Phi^{\mathrm{A}}_{\mathrm{MD}}$ deceases by at least 1 for sufficiently large λ_3.

- **Scenario 5:** $k_{MA} = k_{\widetilde{MA}} = k$ and at least one status reset due to a mismatch transition occurs: Note that this can only happen if due to transmission errors $E^{\mathrm{A}}_{\mathrm{MD}}$ increases despite having matching step sizes. We show that $\Phi^{\mathrm{A}}_{\mathrm{MD}}$ increases by at most a constant in this scenario. Suppose without loss of generality that a mismatch transition occurs on Alice's side. The variable $BV^{\mathrm{A}}_{\mathrm{MD}}$ gets reset to 0. This can only contribute to decreasing $\Phi^{\mathrm{A}}_{\mathrm{MD}}$. Alice's E_{MA} gets reset to 0 due to the mismatch transition, which implies that $E_{MA} + 1 > k/2$. So $E_{MA} \geq 0.5k - 0.5$ at the beginning of the iteration and resetting E_{MA} leads to a decrease by $0.5\lambda_4\left(k-1\right)$ in $\Phi^{\mathrm{A}}_{\mathrm{MD}}$. Bob's $E_{\widetilde{MA}}$ gets reset or increases by at most 1. Note that in any case this cannot increase

$\Phi^{\mathrm{A}}_{\mathrm{MD}}$. The remaining variables change by at most $2k$ while being weighted by smaller constants. In particular, the iteration number increases by 1 and md^{A}_{+} and md^{A}_{-} change by at most $2k$. Alice's k_{MA} gets reset which increases $\Phi^{\mathrm{A}}_{\mathrm{MD}}$ by $\lambda_1 (k-1)$ and Bob's $k_{\widetilde{MA}}$ gets reset or increases by at most 1, which increases $\Phi^{\mathrm{A}}_{\mathrm{MD}}$ by less than $(\lambda_1 + 0.9\lambda_3) k$. Therefore, we have

$$\Delta\Phi^{\mathrm{A}}_{\mathrm{MD}} \leq 1 + 2k + \lambda_2 (2k) + (2\lambda_1 + 0.9\lambda_3) k - 0.5\lambda_4 (k-1) \quad .$$

If $k > 1$ then for λ_4 sufficiently large, the potential function $\Phi^{\mathrm{A}}_{\mathrm{MD}}$ decreases by at least 1. The case of $k = 1$ is trivial since all the variables change by at most some constant and so does $\Phi^{\mathrm{A}}_{\mathrm{MD}}$.

- **Scenario 6:** $k_{MA} = k_{\widetilde{MA}} = k$, exactly one status reset occurs, and the reset is due to a meeting point transition: Without loss of generality, we assume that Alice does the transition. Note that in this case k_{MA} and $k_{\widetilde{MA}}$ have been equal in the last k iterations while increasing from 1 to k.

  First, suppose that $\overline{k_{MA}} \neq \overline{k_{\widetilde{MA}}}$. Then $k_{\widetilde{MA}}$ increases to $k+1$, which increases $\Phi^{\mathrm{A}}_{\mathrm{MD}}$ by less than $(0.9\lambda_3 + \lambda_1) k$. Bob's $E_{\widetilde{MA}}$ increases by at most 1, but due to the re-weighting this does not increase $\Phi^{\mathrm{A}}_{\mathrm{MD}}$. Finally, Bob's actions do not have any effect on md^{A}_{+} and md^{A}_{-}. Alice's E_{MA} gets reset to 0, k_{MA} gets reset to 1, the iteration number increases by 1, $BV^{\mathrm{A}}_{\mathrm{MD}}$ gets reset to 0, and the remaining variables change by at most $2k$.

  If the meeting point to which Alice backtracks does not match any of Bob's meeting points in the current iteration then, in the last $k/2$ iterations, Alice has incorrectly increased her vote count for this meeting point at least $0.8k/2$ times. This implies that $BV^{\mathrm{A}}_{\mathrm{MD}} \geq 0.4k$. Therefore, for sufficiently large λ_5, the decrease in $\Phi^{\mathrm{A}}_{\mathrm{MD}}$ due to resetting $BV^{\mathrm{A}}_{\mathrm{MD}}$ dominates any other potential change and $\Phi^{\mathrm{A}}_{\mathrm{MD}}$ decreases by at least 1.

  Otherwise, if the meeting point to which Alice backtracks does indeed match one of Bob's meeting points in the current iteration, then Bob should have also backtracked to the same meeting point. The fact that Bob does not have enough votes for the meeting point implies that $E_{\widetilde{MA}} + BV^{\mathrm{A}}_{\mathrm{MD}} > 0.2 (k/2)$. Therefore, the decrease due to the re-weighting of $E_{\widetilde{MA}}$ and resetting $BV^{\mathrm{A}}_{\mathrm{MD}}$ dominates all other potential changes. In more detail, we have:

  - If $E_{\widetilde{MA}} \geq k/20$ then the decrease by at least $(\lambda_4 + \lambda_3) E_{\widetilde{MA}}$ due to the re-weighting of $E_{\widetilde{MA}}$ dominates the potential change due to i, md^{A}_{+}, md^{A}_{-}, and $k^{\mathrm{A}}_{\mathrm{MD}}$. Moreover, resetting E_{MA} and $BV^{\mathrm{A}}_{\mathrm{MD}}$ does not increases $\Phi^{\mathrm{A}}_{\mathrm{MD}}$. Hence, regardless of the number of errors and collisions, we have

    $$\begin{aligned}\Delta\Phi^{\mathrm{A}}_{\mathrm{MD}} &\leq 1 + 2k + \lambda_2 (2k) + [\lambda_1 (k-1) + (0.9\lambda_3 + \lambda_1) (k)] - (\lambda_4 + \lambda_3) E_{\widetilde{MA}} \\ &\leq -[\lambda_4/20 - 0.85\lambda_3 - 2\lambda_2 - 2\lambda_1 - 2] k + 1 \quad .\end{aligned}$$

Therefore, for sufficiently large λ_4, the potential function $\Phi^{\mathrm{A}}_{\mathrm{MD}}$ decreases by at least 1.

- If $E_{\widetilde{MA}} < k/20$ then $BV^{\mathrm{A}}_{\mathrm{MD}}$ is at least $0.1k - k/20 = 0.05k$. Therefore, for sufficiently large λ_5, the decrease in $\Phi^{\mathrm{A}}_{\mathrm{MD}}$ due to resetting $BV^{\mathrm{A}}_{\mathrm{MD}}$ dominates any other potential change and $\Phi^{\mathrm{A}}_{\mathrm{MD}}$ decreases by at least 1.

It remains to consider the case where $\overline{k_{MA}} = \overline{k_{\widetilde{MA}}} = 1$. In the scenario we are considering, no status reset occurs on Bob's side in the current iteration. So $\overline{k_{\widetilde{MA}}} = 1$ only if $k = 1$ and Bob does not conduct the subroutine MPrewind on the rewinding data corresponding to $\widetilde{MA}$. But this can only happen if a transmission error or a hash collision occurs. In this case, all the variables change by at most a constant and therefore, so does $\Phi^{\mathrm{A}}_{\mathrm{MD}}$.

- **Scenario 7:** $k_{MA} = k_{\widetilde{MA}} = k$ and status resets occur on both sides due to meeting point transitions: In this case, the variables $E^{\mathrm{A}}_{\mathrm{MD}}$,$BV^{\mathrm{A}}_{\mathrm{MD}}$ get reset to 0. Therefore, any change in these variables can only decrease $\Phi^{\mathrm{A}}_{\mathrm{MD}}$. The variables k_{MA} and $k_{\widetilde{MA}}$ get reset to 1. This increases $\Phi^{\mathrm{A}}_{\mathrm{MD}}$ by $2\lambda_1\,(k-1)$. The remaining variables change by at most $2k$. We consider the following different possibilities:

  - $\overline{md^{\mathrm{A}}_{-}} \neq 0$: In the last $k/2$ iterations, at least one party has incorrectly voted more than $0.8\,(k/2)$ times for a meeting point that does not match any of the other party's meeting points in the current iteration. Therefore, we have $BV^{\mathrm{A}}_{\mathrm{MD}} \geq 0.4k$ and for sufficiently large λ_5, the decrease due to resetting $BV^{\mathrm{A}}_{\mathrm{MD}}$ dominates any other change in the potential function and $\Phi^{\mathrm{A}}_{\mathrm{MD}}$ decreases by at least 1.
  - $\overline{md^{\mathrm{A}}_{-}} = 0$ and $k < 8md^{\mathrm{A}}_{-}$: In this case $\Phi^{\mathrm{A}}_{\mathrm{MD}}$ decreases by at least

$$-1 - 2k + \lambda_2 md^{\mathrm{A}}_{-} - 2\lambda_1\,(k-1) \geq [\lambda_2/8 - 2\lambda_1 - 2]\,k - 1 \ ,$$

    which is at least 1 for sufficiently large λ_2 and every $k \geq 1$.
  - $\overline{md^{\mathrm{A}}_{-}} = 0$ and $k \geq 8md^{\mathrm{A}}_{-}$: First note that if $E^{\mathrm{A}}_{\mathrm{MD}} \geq k/100$ then the decrease due to resetting $E^{\mathrm{A}}_{\mathrm{MD}}$ guarantees an overall decrease by at least 1, for large enough λ_4. Suppose that $E^{\mathrm{A}}_{\mathrm{MD}} < k/100$. Recall that if $k_{MA} = k_{\widetilde{MA}} \geq 2^{\lfloor \log md^{\mathrm{A}}_{-} \rfloor + 1}$ then Alice and Bob have at least one matching meeting point. Moreover, $k \geq 8md^{\mathrm{A}}_{-}$ implies that $k/4 \geq 2^{\lfloor \log md^{\mathrm{A}}_{-} \rfloor + 1}$. This implies that Alice and Bob should have backtracked to a matching meeting point earlier. In particular, in at least 0.2 of the iterations in which k_{MA} and $k_{\widetilde{MA}}$ have increased from $2^{\lfloor \log md^{\mathrm{A}}_{-} \rfloor + 1}$ to $k/2$, i.e., in at least $0.2\,(k/4)$ iterations, the vote counts for their matching meeting points have not increased. So $BV^{\mathrm{A}}_{\mathrm{MD}} \geq 0.05k - E^{\mathrm{A}}_{\mathrm{MD}} \geq 0.04k$ and for sufficiently large λ_5, the decrease due to resetting $BV^{\mathrm{A}}_{\mathrm{MD}}$ guarantees an overall decrease in $\Phi^{\mathrm{A}}_{\mathrm{MD}}$ by at least 1.

This completes the proof. □

The following lemma bounds the number of iterations with a metadata hash collision.

Lemma 5.5.7. *The number of iterations of Algorithm 28 suffering from a metadata hash collision is at most $c_1 n\epsilon$ with probability at least $1 - 2^{-\Theta(n\epsilon)}$.*

Proof. We call an iteration a *dangerous iteration of type I* if at the beginning of the iteration, either $md_-^{\mathrm{A}} + md_-^{\mathrm{B}} \neq 0$ or $k_{\mathrm{MD}}^{\mathrm{A}} + k_{\mathrm{MD}}^{\mathrm{B}} > 4$. Note that metadata hash collisions can only occur in these iterations. Let h_{MD} denote the total number of iterations with a metadata hash collision and d_{I} denote the total number of type I dangerous iterations. It suffices to prove that

$$\Pr\left(d_{\mathrm{I}} > c_1 n\epsilon\right) \leq 2^{-\Theta(n\epsilon)} \ .$$

Note that in type I dangerous iterations $\Psi_{\mathrm{MD}} > 0$. Therefore, by Lemma 5.5.6, in these iterations $\Phi_{\mathrm{MD}}^{\mathrm{A}}$ decreases by at least 1, if no error or collision occurs while it increases by at most a fixed constant c_{MD}^{+} otherwise. Hence, the total change in Φ_{MD} in type I dangerous iterations is at most $c_{\mathrm{MD}}^{+}\left(h_{\mathrm{MD}} + 2n\epsilon\right) - \left(d_{\mathrm{I}} - h_{\mathrm{MD}} - 2n\epsilon\right)$. In iterations that are not type I dangerous, Φ_{MD} increases only if a transmission error occurs. Moreover, if the iteration is error free then Φ_{MD} remains unchanged when $\Psi_{\mathrm{MD}} = 0$, and it decreases by at least 1 otherwise. The total contribution of non-dangerous iterations to the final value of Φ_{MD} is thus at most $c_{\mathrm{MD}}^{+}\left(2n\epsilon\right)$. By Lemma 5.5.1, Φ_{MD} is bounded below by a value in $-\mathrm{O}\left(n\epsilon\right)$. Therefore, we have $\left(1 + c_{\mathrm{MD}}^{+}\right) h_{\mathrm{MD}} - d_{\mathrm{I}} \in -\mathrm{O}\left(n\epsilon\right)$. For sufficiently large c_1, if $d_{\mathrm{I}} > c_1 n\epsilon$ then $h_{\mathrm{MD}} \geq \frac{d_{\mathrm{I}}}{2\left(1+c_{\mathrm{MD}}^{+}\right)}$. However, for collision probability $p = 2^{-o}$ satisfying $p < \frac{1}{20\left(1+c_{\mathrm{MD}}^{+}\right)}$, the probability of having such a large fraction of collisions during the dangerous iterations is very small. In particular, hash collisions are $2^{-\Theta\left(n\sqrt{\epsilon}\right)}$-statistically close to being dominated by independent Bernoulli(p) variables. Together with the Chernoff bound this implies that the probability of having such a large deviation from the expected number of collisions is at most $2^{-\Theta(n\epsilon)}$. □

Definition 5.5.8. We refer to an iteration of Algorithm 28 as a *recovery iteration of type I* if at least one of Alice or Bob conducts one of the cases i.A, i.B, ii.A, or ii.B.

We use the following lemma to bound the number of type I recovery iterations.

Lemma 5.5.9. *Suppose that in the first i iterations of Algorithm 28, the number of iterations suffering from a metadata hash collision is at most $c_1 n\epsilon$. Then the number of type I recovery iterations in the first i iterations is at most $c_2 n\epsilon$.*

Proof. Let

$$\Phi_{\mathrm{I}} \stackrel{\mathrm{def}}{=} \Phi_{\mathrm{MD}} + u \ ,$$
$$\Psi_{\mathrm{I}} \stackrel{\mathrm{def}}{=} \Psi_{\mathrm{MD}} + u \ .$$

By Lemma 5.5.1 and the definition of u in Eq. (5.14), we always have $\Phi_{\mathrm{I}} \in -\mathrm{O}\left(n\epsilon\right)$. Moreover, Ψ_{I} is always non-negative and is equal to zero if and only if $k_{MA} = k_{\widetilde{MA}} = k_{MB} =$

$k_{\widetilde{MB}} = 1$, Alice and Bob know each other's full metadata and have used the same number of MES blocks for QVC.

Note that if $\Psi_{\mathrm{I}} = 0$ at the beginning of an iteration, then the iteration is a type I recovery iteration only if a transmission error in communication of metadata messages occurs. The total number of such iterations is thus at most $2n\epsilon$.

Let β_{I} denote the number of iterations in the first i iterations starting with $\Psi_{\mathrm{I}} > 0$. We prove an upper bound on β_{I}. In any iteration, u increases by at most 1. Hence, by Lemma 5.5.6, in each iteration, regardless of the number of errors and collisions, Φ_{I} increases by at most some fixed constant.

In iterations starting with $\Psi_{\mathrm{I}} = 0$, the potential function Φ_{I} remains the same unless a transmission error occurs. Therefore, the accumulated potential in such iterations is at most $\mathrm{O}(n\epsilon)$.

In iterations starting with $\Psi_{\mathrm{I}} > 0$, the potential function Φ_{I} increases only if a transmission error or a metadata hash collision occurs. Assuming that the total number of metadata hash collisions is at most $c_1 n\epsilon$, such iterations increase Φ_{I} by at most $\mathrm{O}(n\epsilon)$. In the remaining iterations, i.e., the iterations starting with $\Psi_{\mathrm{I}} > 0$ in which no errors or metadata hash collisions occur, the potential function Φ_{I} decreases by at least 1. But $\Phi_{\mathrm{I}} \geq -\mathrm{O}(n\epsilon)$ implies that the number of such iterations is also at most $\mathrm{O}(n\epsilon)$.

Therefore, the total number of type I recovery iterations in the first i iterations is at most $c_2 n\epsilon$, for sufficiently large c_2. □

We use the following lemma to bound the number of iterations suffering from a quantum hash collision. The proof is similar to that of Lemma 4.5.8 in the large alphabet case. We repeat the proof with the notations of the small alphabet case.

Lemma 5.5.10. *Suppose that recycling is successful in the first i iterations of Algorithm 28 and the number of iterations suffering from a metadata hash collision is at most $c_1 n\epsilon$. Then the number of iterations suffering from a quantum hash collision in the first i iterations is at most $c_3 n\epsilon$ with probability at least $1 - 2^{-\Theta(n\epsilon)}$.*

Proof. We call an iteration a *dangerous iteration of type II* if $rd^- \neq 0$, at the beginning of the iteration. Note that quantum hash collisions can only occur in type II dangerous iterations. Let d_{II} denote the number of such iterations in the first i iterations of Algorithm 28. It suffices to prove that

$$\Pr\left(d_{\mathrm{II}} > c_3 n\epsilon\right) \leq 2^{-\Theta(n\epsilon)} \ .$$

Note that in any iteration rd^- increases by at most 2. Moreover, in an iteration with $rd^- \neq 0$, if rd^- decreases, it decreases by at least 1. Therefore, in at least $d_{\mathrm{II}}/3$ iterations, rd^- increases or remains unchanged at a nonzero value. Note that, assuming successful recycling in the previous iterations, $rd^- > 0$ increases or remains unchanged in an iteration only if

- A metadata hash collision or a transmission error on metadata messages (i.e., $H_{k_{MA}}$, H_{MA_1}, H_{MA_2}, $H_{k_{\widetilde{MA}}}$, $H_{\widetilde{MA}_1}$, $H_{\widetilde{MA}_2}$, $H_{k_{MB}}$, H_{MB_1}, H_{MB_2}, $H_{k_{\widetilde{MB}}}$, $H_{\widetilde{MB}_1}$, $H_{\widetilde{MB}_2}$) occurs, or else,
- The iteration is a type I recovery iteration. Alice and Bob are still reconciling an earlier inconsistency in their metadata and they are both in case i.A or case i.B, or one of them is in case ii.A and the other one in case ii.B. Else,
- A transmission error on quantum hash values or a quantum hash collision occurs. At least one party does not realize that $rd^- > 0$ and conducts one of the cases v, vi.A, vi.B, or vii.

Moreover, assuming successful recycling in the previous iterations, the value of rd^- increases from zero in an iteration only if

- A metadata hash collision occurs and Alice and Bob act based on incorrect estimates of each other's recycling data, or else,
- A transmission error on metadata messages occurs, or else,
- A transmission error on quantum hash values occurs and only one party conducts case iv, or else,
- A transmission error on the Pauli data messages (i.e., $H_{k_{PA}}$, H_{PA_1}, H_{PA_2}, $H_{k_{\widetilde{PA}}}$, $H_{\widetilde{PA}_1}$, $H_{\widetilde{PA}_2}$, $H_{k_{PB}}$, H_{PB_1}, H_{PB_2}, $H_{k_{\widetilde{PB}}}$, $H_{\widetilde{PB}_1}$, $H_{\widetilde{PB}_2}$) occurs and one party conducts case vi.A or vi.B while the other is in case vii. Else,
- A transmission error occurs on the communicated QVC messages when both parties conduct case vii.

Assuming the number of metadata hash collisions is at most $c_1 n\epsilon$, by Lemma 5.5.9, the total number of type I recovery iterations is at most $c_2 n\epsilon$. The total number of transmission errors is at most $2n\epsilon$. Therefore, in at least $d_{\mathrm{II}}/3 - (c_1 + c_2 + 2)\, n\epsilon$ iterations a quantum hash collision occurs.

The shared random string used as the classical seed for quantum hashing is δ-biased with $\delta = 2^{-\Theta\left(n\sqrt{\epsilon}\right)}$. By Lemma 2.4.8, the seeds are also $\delta^{\Theta(1)}$-statistically close to being $\Theta\left(R_{\mathrm{total}}\right)$-wise independent. Therefore, all hashing steps are statistically close to being fully independent. Combined with Lemma 4.3.2, this implies that the expected number of quantum hash collisions is at most $10^{-3} d_{\mathrm{II}}$. For sufficiently large c_3, if $d_{\mathrm{II}} > c_3 n\epsilon$, the Chernoff bound implies that the probability of having at least $d_{\mathrm{II}}/3 - (c_1 + c_2 + 2)\, n\epsilon$ quantum hash collisions is at most $2^{-\Theta(n\epsilon)}$. □

Definition 5.5.11. We refer to an iteration of Algorithm 28 as a *recovery iteration of type II* if it is not a type I recovery iteration and at least one of Alice or Bob conducts one of the cases iii, iv, or v.

We use the following lemma to bound the number of type II recovery iterations.

Lemma 5.5.12. *Suppose that in the first i iterations of Algorithm 28, recycling is successful, the number of iterations suffering from a metadata hash collision is at most $c_1 n\epsilon$ and the number of iterations suffering from a quantum hash collision is at most $c_3 n\epsilon$. Then the total number of type II recovery iterations in the first i iterations is at most $c_4 n\epsilon$.*

Proof. Note that by Lemma 5.5.1, Φ_{RD} is always non-negative. If at the beginning of an iteration $\Phi_{\mathrm{RD}} = 0$, then the iteration is a type II recovery iteration only if

- $\Psi_{\mathrm{MD}} > 0$ but due to a metadata hash collision or a transmission error on metadata messages both Alice and Bob do not realize that. In this case they compute their estimates of each other's recycling data based on incorrect estimates of each other's metadata.
- $\Psi_{\mathrm{MD}} = 0$, i.e., Alice and Bob have correct estimates of each other's recycling data but a transmission error in communication of quantum hashes occurs.

Therefore, in the first i iterations the total number of type II recovery iterations starting with $\Phi_{\mathrm{RD}} = 0$ is at most $(c_1 + 2)\, n\epsilon$.
Let β_{RD} denote the number of iterations starting with $\Phi_{\mathrm{RD}} > 0$ in the first i iterations. Assuming successful recycling in the preceding iterations, $\Phi_{\mathrm{RD}} > 0$ increases or remains unchanged in an iteration only if

- The iteration is a type I recovery iteration, or else,
- $\Psi_{\mathrm{MD}} > 0$ but due to a metadata hash collision or transmission errors on metadata messages, Alice and Bob don't realize that and act based on their incorrect estimates of each other's recycling data.
- $\Psi_{\mathrm{MD}} = 0$, i.e., Alice and Bob have correct estimates of each other's recycling data but a quantum hash collision or a transmission error on quantum hash values occurs.

Moreover, Φ_{RD} increases from zero in an iteration only if

- A transmission error occurs, or else,
- A metadata hash collision occurs and the two parties act based on incorrect estimates of each other's recycling data.

Therefore, the number of iterations in the first i iterations with Φ_{RD} increasing or remaining unchanged at a nonzero value is at most $(c_1 + c_2 + c_3 + 2)\, n\epsilon$. Note that in each iteration, regardless of the number of errors and collisions, Φ_{RD} increases by at most

30. Moreover, if Φ_{RD} decreases, it decreases by at least 1. This implies that the number of iterations in which Φ_{RD} decreases is at most $30\,(c_1+c_2+c_3+2)\,n\epsilon$. So, we have $\beta_{\mathrm{RD}} \leq 31\,(c_1+c_2+c_3+2)\,n\epsilon$.
Therefore, the total number of recovery iterations of type II in the first i iterations is at most $c_4 n\epsilon$, where $c_4 \stackrel{\text{def}}{=} 31\,(c_1+c_2+c_3+2)+(c_1+2)$. □

We define $\Psi_{\mathrm{PD}}^{\mathrm{A}}$, $\Psi_{\mathrm{PD}}^{\mathrm{B}}$, and Ψ_{PD} as follows:

$$\Psi_{\mathrm{PD}}^{\mathrm{A}} \stackrel{\text{def}}{=} 6q_{MA} - pd_{+}^{\mathrm{A}} + pd_{-}^{\mathrm{A}} + \left(k_{\mathrm{PD}}^{\mathrm{A}} - 2\right) \ , \tag{5.34}$$

$$\Psi_{\mathrm{PD}}^{\mathrm{B}} \stackrel{\text{def}}{=} 6q_{MB} - pd_{+}^{\mathrm{B}} + pd_{-}^{\mathrm{B}} + \left(k_{\mathrm{PD}}^{\mathrm{B}} - 2\right) \ , \tag{5.35}$$

$$\Psi_{\mathrm{PD}} \stackrel{\text{def}}{=} \Psi_{\mathrm{PD}}^{\mathrm{A}} + \Psi_{\mathrm{PD}}^{\mathrm{B}} \ . \tag{5.36}$$

Note that we always have $\Psi_{\mathrm{PD}}^{\mathrm{A}}, \Psi_{\mathrm{PD}}^{\mathrm{B}} \geq 0$, and hence $\Psi_{\mathrm{PD}} \geq 0$. The following lemma characterizes the behaviour of Φ_{PD} in each iteration under different possible scenarios. The analyses of the first 5 scenarios are along similar lines as those appearing in Lemma 5.5.6. However, the last two scenarios are more involved. Introducing the variables $BI_{\mathrm{PD}}^{\mathrm{A}}$ and $BI_{\mathrm{PD}}^{\mathrm{B}}$ facilitates the analysis.

Lemma 5.5.13. *Each iteration of Algorithm 28, regardless of the number of hash collisions and transmission errors, increases the potential functions $\Phi_{\mathrm{PD}}^{\mathrm{A}}$ and $\Phi_{\mathrm{PD}}^{\mathrm{B}}$ by at most a constant. Moreover, in any iteration with no transmission errors or hash collisions that is not a type I or type II recovery iteration,*

- *if $\Psi_{\mathrm{PD}} = 0$ at the beginning of the iteration then Φ_{PD} remains the same,*
- *otherwise, if $\Psi_{\mathrm{PD}}^{\mathrm{A}} > 0$ then $\Phi_{\mathrm{PD}}^{\mathrm{A}}$ decreases by at least 1. Similarly, if $\Psi_{\mathrm{PD}}^{\mathrm{B}} > 0$ then $\Phi_{\mathrm{PD}}^{\mathrm{B}}$ decreases by at least 1.*

Proof. First note that in an iteration with no transmission errors or hash collisions that is not a type I or type II recovery iteration, if $\Psi_{\mathrm{PD}} = 0$ at the beginning of the iteration, then both Alice and Bob conduct Case vii. In this case, the variables q_{MA} and q_{MB} increase by 1, while pd_{+}^{A} and pd_{+}^{B} increase by 6. The remaining variables appearing in Φ_{PD} remain unchanged. Hence, Φ_{PD} remains the same.

In the remainder of the proof, we focus on $\Phi_{\mathrm{PD}}^{\mathrm{A}}$. The potential function $\Phi_{\mathrm{PD}}^{\mathrm{B}}$ behaves similarly. We consider all the remaining possible scenarios in the iteration at hand. We denote by pd_{+}^{A}, pd_{-}^{A}, k_{PA}, $k_{\widetilde{PA}}$, E_{PA}, $E_{\widetilde{PA}}$, and $BV_{\mathrm{PD}}^{\mathrm{A}}$ the values at the beginning of the iteration and denote by $\overline{pd_{+}^{\mathrm{A}}}$, $\overline{pd_{-}^{\mathrm{A}}}$, $\overline{k_{PA}}$, $\overline{k_{\widetilde{PA}}}$, $\overline{E_{PA}}$, $\overline{E_{\widetilde{PA}}}$, and $\overline{BV_{\mathrm{PD}}^{\mathrm{A}}}$ the values at the end of the iteration.

- **Scenario 1:** Alice and Bob do not conduct the subroutine MPrewind on the rewinding data corresponding to PA and $\widetilde{PA}$, respectively: In this case, all the variables

appearing in $\Phi_{\text{PD}}^{\text{A}}$ vary by at most a constant. Hence, regardless of the number of errors and hash collisions, $\Phi_{\text{PD}}^{\text{A}}$ increases by at most a constant. If the iteration is not a type I or type II recovery iteration, has no errors or hash collisions, and starts with $\Psi_{\text{PD}}^{\text{A}} > 0$, then we have $pd_{-}^{\text{A}} = 0$ and $pd_{+}^{\text{A}} < 6q_{MA}$ at the beginning of the iteration. Both parties conduct Case vi.B and pd_{+}^{A} increases by 1 while the remaining variables remain unchanged. Therefore, $\Phi_{\text{PD}}^{\text{A}}$ decreases by at least 1.

In the remaining scenarios, at least one of Alice or Bob conducts the subroutine MPrewind on the rewinding data corresponding to PA or $\widetilde{PA}$, respectively.

- **Scenario 2:** No status reset on the variables corresponding to PA and $\widetilde{PA}$ occurs: In this scenario, all the variables appearing in $\Phi_{\text{PD}}^{\text{A}}$ change by at most a constant and so does $\Phi_{\text{PD}}^{\text{A}}$. If the iteration has no transmission errors or hash collisions and is not a type I or type II recovery iterations then both parties conduct Case vi.A. The variables $BI_{\text{PD}}^{\text{A}}$, $BV_{\text{PD}}^{\text{A}}$, pd_{-}^{A}, pd_{+}^{A}, and q_{MA} remain unchanged while k_{PA} and $k_{\widetilde{PA}}$ increase by 1. If $k_{PA} = k_{\widetilde{PA}}$ at the beginning of the iteration then E_{PD}^{A} remains unchanged. Therefore, $\Phi_{\text{PD}}^{\text{A}}$ decreases by at least $2\eta_1$, which is at least 1 for sufficiently large η_1. Otherwise, if $k_{PA} \neq k_{\widetilde{PA}}$ then E_{MD}^{A} increases by 2, leading to a potential decrease by at least $2\left(\eta_3 - 0.9\eta_3\right) = 0.2\eta_3$, which is at least 1 for sufficiently large η_3.

- **Scenario 3:** $k_{PA} \neq k_{\widetilde{PA}}$ and due to a mismatch or meeting point transition, exactly one status reset on the variables corresponding to PA or $\widetilde{PA}$ occurs: Suppose that Alice does the status reset. We consider the following two possibilities:

  - $\overline{k_{PA}} = \overline{k_{\widetilde{PA}}} = 1$: This can only happen in an iteration starting with $k_{\widetilde{PA}} = 1$ in which Bob does not conduct MPrewind on the rewinding data corresponding to $\widetilde{PA}$. The variable q_{MA} remains the same. Note that in this case, $E_{\widetilde{PA}} = 0$. Bob's variables $k_{\widetilde{PA}}$, and $E_{\widetilde{PA}}$ remain unchanged. The variable pd_{+}^{A} does not decrease due to Bob's actions, while he increases pd_{-}^{A} by at most 1. On the other hand, Alice's variables k_{PA}, E_{PA}, and the variables $BV_{\text{PD}}^{\text{A}}$, $BI_{\text{PD}}^{\text{A}}$ get reset. Due to Alice's actions, pd_{+}^{A} decrease by at most $2k_{PA}$ and pd_{-}^{A} increases by at most $2k_{PA}$. Therefore, we have

$$\begin{aligned}\Delta\Phi_{\text{PD}}^{\text{A}} &\leq 2k_{PA} + \eta_2 \max\{2k_{PA}, 1\} - 0.9\eta_3\left(k_{PA} - 1\right) + \eta_3 E_{PA} \\ &\leq -k_{PA}\left[0.9\eta_3 - 0.5\eta_3 - 2\eta_2 - 2\right] + 0.9\eta_3 - 0.5\eta_3 \\ &= 0.4\eta_3 - k_{PA}\left[0.4\eta_3 - 2\eta_2 - 2\right] \ ,\end{aligned}$$

    where the inequality holds since at the beginning of every iteration we always have $E_{PA} \leq 0.5k_{PA} - 0.5$. Since $k_{PA} \geq 2$, for sufficiently large η_3, $\Phi_{\text{PD}}^{\text{A}}$ decreases by at least 1.

  - $\overline{k_{PA}} \neq \overline{k_{\widetilde{PA}}}$: If Bob conducts the subroutine MPrewind on the rewinding data corresponding to $\widetilde{PA}$ then $k_{\widetilde{PA}}$ increases by 1, while $E_{\widetilde{PA}}$ increases by at most 1.

Otherwise, $k_{\widetilde{PA}}$ remains unchanged at a value greater than 1 and $E_{\widetilde{PA}}$ remains the same. In any case, $k_{\widetilde{PA}}$ and $E_{\widetilde{PA}}$ increase by at most 1. Moreover, Bob's actions do not have any effect on the value of pd_+^{A} and pd_-^{A}. So Bob's contribution increases $\Phi_{\mathrm{PD}}^{\mathrm{A}}$ by at most $0.9\eta_3$. Using a similar argument as in the above case, Alice's contribution to $\Delta\Phi_{\mathrm{PD}}^{\mathrm{A}}$ is at most

$$0.4\eta_3 - k_{PA}\left[0.4\eta_3 - 2\eta_2 - 2\right] \ ,$$

which is a decreasing function of k_{PA} and is equal to $2\eta_2 + 2$ when $k_{PA} = 1$. Therefore, regardless of the number of errors and hash collisions, $\Phi_{\mathrm{PD}}^{\mathrm{A}}$ increases by at most

$$0.9\eta_3 + 2\eta_2 + 2 \ .$$

Suppose that the iteration is not a type I or type II recovery iteration and has no transmission errors or hash collisions. Then $k_{\widetilde{PA}}$ and $E_{\widetilde{PA}}$ increase by 1. Therefore, Bob's contribution decreases $\Phi_{\mathrm{PD}}^{\mathrm{A}}$ by $0.1\eta_3$. If $k_{PA} \geq 2$ then Alice's contribution also decreases $\Phi_{\mathrm{PD}}^{\mathrm{A}}$ by at least a constant. Therefore, $\Phi_{\mathrm{PD}}^{\mathrm{A}}$ decreases by at least 1. Otherwise, if $k_{PA} = 1$ at the beginning of the iteration, Alice's contribution increases $\Phi_{\mathrm{PD}}^{\mathrm{A}}$, but it is dominated by the decrease due to Bob's contribution. In more detail, in this case Alice's variables E_{PA} is 0 at the beginning of the iteration and it remains the same, k_{PA} remains at 1, pd_+^{A} decreases by at most 2, and pd_-^{A} increases by at most 2. Therefore, $\Phi_{\mathrm{PD}}^{\mathrm{A}}$ decreases by at least $0.1\eta_3 - 2\eta_2 - 2$, which is at least 1 for sufficiently large η_3.

If the status reset occurs on Bob's side the proof would be similar with the role of Alice and Bob switched, except in this case q_{MA} increases by at most 1.

- **Scenario 4:** $k_{PA} \neq k_{\widetilde{PA}}$ and status resets occur on both sides on the variables corresponding to PA and $\widetilde{PA}$ due to mismatch or meeting point transitions: In this case, $\overline{k_{PA}} = \overline{k_{\widetilde{PA}}} = 1$ and $\overline{E_{\mathrm{PD}}^{\mathrm{A}}} = \overline{BV_{\mathrm{PD}}^{\mathrm{A}}} = \overline{BI_{\mathrm{PD}}^{\mathrm{A}}} = 0$. At the beginning of every iteration, we have $E_{PA} \leq 0.5k_{PA} - 0.5$ and $E_{\widetilde{PA}} \leq 0.5k_{\widetilde{PA}} - 0.5$ and therefore, $E_{\mathrm{PD}}^{\mathrm{A}} \leq 0.5k_{\mathrm{PD}}^{\mathrm{A}} - 1$. Moreover, pd_+^{A} decreases by at most $2\max\{k_{PA}, k_{\widetilde{PA}}\} \leq 2k_{\mathrm{PD}}^{\mathrm{A}}$ and similarly, pd_-^{A} increases by at most $2k_{\mathrm{PD}}^{\mathrm{A}}$. The variable q_{MA} remains unchanged. Hence, we have

$$\begin{aligned}
\Delta\Phi_{\mathrm{PD}}^{\mathrm{A}} &\leq 2k_{\mathrm{PD}}^{\mathrm{A}} + \eta_2\left(2k_{\mathrm{PD}}^{\mathrm{A}}\right) + 0.9\eta_3\left(k_{\mathrm{PD}}^{\mathrm{A}} - 2\right) - \eta_3 E_{\mathrm{PD}}^{\mathrm{A}} \\
&\leq -k_{\mathrm{PD}}^{\mathrm{A}}\left(0.9\eta_3 - 0.5\eta_3 - 2\eta_2 - 2\right) + 1.8\eta_3 - \eta_3 \\
&\leq -0.4\eta_3 + 6\eta_2 + 6 \ ,
\end{aligned}$$

where the last inequality follows since $k_{\mathrm{PD}}^{\mathrm{A}} \geq 3$. Thus, $\Phi_{\mathrm{PD}}^{\mathrm{A}}$ deceases by at least 1 for sufficiently large η_3.

- **Scenario 5:** $k_{PA} = k_{\widetilde{PA}} = k$ and at least one status reset due to a mismatch transition occurs: Note that this can only happen if due to transmission errors $E_{\mathrm{PD}}^{\mathrm{A}}$ increases

despite having matching step sizes. We show that $\Phi^{\mathrm{A}}_{\mathrm{PD}}$ increases by at most a constant in this scenario.

Suppose that a mismatch transition occurs on Alice's side. The variables $BV^{\mathrm{A}}_{\mathrm{PD}}$ and $BI^{\mathrm{A}}_{\mathrm{PD}}$ get reset to 0. This can only contribute to decreasing $\Phi^{\mathrm{A}}_{\mathrm{PD}}$. Alice's E_{PA} gets reset to 0, which implies that $E_{PA}+1 > k/2$. So $E_{PA} \geq 0.5k - 0.5$ at the beginning of the iteration and resetting E_{PA} leads to a decrease by $0.5\eta_4\,(k-1)$ in $\Phi^{\mathrm{A}}_{\mathrm{PD}}$. Bob's $E_{\widetilde{PA}}$ gets reset or increases by at most 1. Note that in any case this cannot increase $\Phi^{\mathrm{A}}_{\mathrm{PD}}$. The remaining variables change by at most $2k$ while being weighted by smaller constants. In particular, q_{MA} stays the same and pd^{A}_{+} and pd^{A}_{-} change by at most $2k$. Alice's k_{PA} gets reset which increases $\Phi^{\mathrm{A}}_{\mathrm{PD}}$ by $\eta_1\,(k-1)$ and Bob's $k_{\widetilde{PA}}$ gets reset or increases by at most 1, which increases $\Phi^{\mathrm{A}}_{\mathrm{PD}}$ by less than $(\eta_1 + 0.9\eta_3)\,k$. Therefore, we have

$$\Delta\Phi^{\mathrm{A}}_{\mathrm{MD}} \leq 2k + \eta_2\,(2k) + (2\eta_1 + 0.9\eta_3)\,k - 0.5\eta_4\,(k-1) \quad .$$

If $k > 1$ then for η_4 sufficiently large, the potential function $\Phi^{\mathrm{A}}_{\mathrm{PD}}$ decreases by at least 1. The case of $k = 1$ is trivial since all the variables change by at most some constant and so does $\Phi^{\mathrm{A}}_{\mathrm{PD}}$.

If no mismatch transition occurs on Alice's side the proof would be similar with the role of Alice and Bob switched, except in this case q_{MA} increases by at most 1.

- **Scenario 6:** $k_{PA} = k_{\widetilde{PA}} = k$, exactly one status reset occurs, and the reset is due to a meeting point transition: Suppose that the meeting point transition occurs on Alice's side. Note that in this case it is not necessarily true that k_{PA} and $k_{\widetilde{PA}}$ have been equal in the last k iterations while increasing from 1 to k. Therefore, the proof is more involved compared to Scenario 6 in Lemma 5.5.6.

  First, suppose that $\overline{k_{PA}} \neq \overline{k_{\widetilde{PA}}}$. The variables $BV^{\mathrm{A}}_{\mathrm{PD}}$ and $BI^{\mathrm{A}}_{\mathrm{PD}}$ get reset to 0. In the current iteration, if Bob conducts the subroutine MPrewind on the rewinding data corresponding to $\widetilde{PA}$ then $k_{\widetilde{PA}}$ increases by 1 and $E_{\widetilde{PA}}$ increases by at most 1. Otherwise, $k_{\widetilde{PA}}$ remains unchanged at a value greater than 1, and $E_{\widetilde{PA}}$ remains the same. Due to re-weighting, an increase by at most 1 in $k_{\widetilde{PA}}$ increases $\Phi^{\mathrm{A}}_{\mathrm{PD}}$ by less than $(0.9\eta_3 + \eta_1)\,k$. Due to the same reason, an increase by at most 1 in $E_{\widetilde{PA}}$ does not increase $\Phi^{\mathrm{A}}_{\mathrm{PD}}$. Finally, q_{MA} remains the same, k_{PA} gets reset to 1, E_{PA} gets reset to 0, and pd^{A}_{+} and pd^{A}_{-} change by at most $2k$.

  If $BI^{\mathrm{A}}_{\mathrm{PD}} > k/100$ then, for sufficiently large η_6, the decrease due to resetting $BI^{\mathrm{A}}_{\mathrm{PD}}$ dominates any other change in the potential function and $\Phi^{\mathrm{A}}_{\mathrm{PD}}$ decreases by at least 1. Suppose that $BI^{\mathrm{A}}_{\mathrm{PD}} \leq k/100$.

  Note that in each iteration after the last status reset, the distance between k_{PA} and $k_{\widetilde{PA}}$ changes by at most 1. Moreover, in these iterations, the variable $BI^{\mathrm{A}}_{\mathrm{PD}}$ increases by 1 whenever the distance changes. Since $k_{PA} = k_{\widetilde{PA}}$ at the beginning of the current iteration, this implies that the maximum distance between k_{PA} and $k_{\widetilde{PA}}$ after the last status reset was at most $BI^{\mathrm{A}}_{\mathrm{PD}}$. We consider two case:

- The meeting point to which Alice backtracks does not match any of Bob's meeting points in the current iteration: Consider the iterations in which $k_{PA} \geq k/2$ has increased by 1. In these $k/2$ iterations, Alice has voted for this meeting point at least $0.8k/2$ times. In at most $BI^{\mathrm{A}}_{\mathrm{PD}}$ of these iterations $k_{\widetilde{PA}} < k/2$. So Alice has voted for her meeting point at least $0.4k - BI^{\mathrm{A}}_{\mathrm{PD}}$ times in iterations with $\hat{k}_{\widetilde{PA}} = \hat{k}_{PA} = k/2$. But in these iterations her meeting point does not match any of Bob's meeting points. The variable $BV^{\mathrm{A}}_{\mathrm{PD}}$ increases by 1 when Alice votes for an incorrect meeting point except when Bob does not conduct the subroutine MPrewind on the rewinding data corresponding to $\widetilde{PA}$. Therefore, $BV^{\mathrm{A}}_{\mathrm{PD}} \geq 0.4k - 2BI^{\mathrm{A}}_{\mathrm{PD}} \geq 0.38k$. For sufficiently large η_5, the decrease in $\Phi^{\mathrm{A}}_{\mathrm{PD}}$ due to resetting $BV^{\mathrm{A}}_{\mathrm{PD}}$ dominates any other potential change and $\Phi^{\mathrm{A}}_{\mathrm{PD}}$ decreases by at least 1.
- The meeting point to which Alice backtracks does indeed match one of Bob's meeting points: In this case, Bob should have backtracked to the same meeting point as well. Consider the iterations in which $k_{\widetilde{PA}} \geq k/2$ has increased by 1. Among these $k/2$ iterations, in at least $k/2 - BI^{\mathrm{A}}_{\mathrm{PD}}$ iterations k_{PA} was also at least $k/2$, i.e., $\hat{k}_{\widetilde{PA}} = \hat{k}_{PA} = k/2$. In at least $k/2 - 2BI^{\mathrm{A}}_{\mathrm{PD}}$ of these iterations Alice has conducted the subroutine MPrewind on the rewinding data corresponding to PA. Therefore, the fact that Bob does not have enough votes for the meeting point implies that

$$k/2 - 2BI^{\mathrm{A}}_{\mathrm{PD}} - E_{\widetilde{MA}} - BV^{\mathrm{A}}_{\mathrm{PD}} < 0.8k/2 \ .$$

  Therefore, $E_{\widetilde{PA}} + BV^{\mathrm{A}}_{\mathrm{PD}} > 0.1k - 0.02k = 0.08k$ and the decrease due to the re-weighting of $E_{\widetilde{PA}}$ or resetting $BV^{\mathrm{A}}_{\mathrm{PD}}$ dominates all other potential changes. In more detail, we have:

  - If $E_{\widetilde{PA}} \geq k/20$ then the decrease due to the re-weighting of $E_{\widetilde{MA}}$ dominates the potential change due to md^{A}_{+}, md^{A}_{-}, and $k^{\mathrm{A}}_{\mathrm{PD}}$. Moreover, resetting E_{PA}, $BV^{\mathrm{A}}_{\mathrm{PD}}$, and $BI^{\mathrm{A}}_{\mathrm{PD}}$ does not increase $\Phi^{\mathrm{A}}_{\mathrm{PD}}$. Hence, we have

$$\begin{aligned}\Delta\Phi^{\mathrm{A}}_{\mathrm{PD}} &\leq 2k + \eta_2\,(2k) + [\eta_1\,(k-1) + (0.9\eta_3 + \eta_1)\,(k)] \\ &\quad - (\eta_4 + \eta_3)\,E_{\widetilde{PA}} \\ &\leq -\left[\eta_4/20 - 0.85\eta_3 - 2\eta_2 - 2\eta_1 - 2\right]k \ ,\end{aligned}$$

    which is at most -1, for every $k \geq 1$ and sufficiently large η_4.
  - If $E_{\widetilde{PA}} < k/20$ then $BV^{\mathrm{A}}_{\mathrm{PD}}$ is at least $0.08k - k/20 = 0.03k$. Therefore, for sufficiently large η_5, the decrease in $\Phi^{\mathrm{A}}_{\mathrm{PD}}$ due to resetting $BV^{\mathrm{A}}_{\mathrm{PD}}$ dominates any other potential change and $\Phi^{\mathrm{A}}_{\mathrm{PD}}$ decreases by at least 1.

It remains to consider the case where $\overline{k_{PA}} = \overline{k_{\widetilde{PA}}} = 1$. In the scenario we are considering, no status reset occurs on Bob's side in the current iteration. So $\overline{k_{\widetilde{PA}}} = 1$

only if $k = 1$ and Bob does not conduct the subroutine MPrewind on the rewinding data corresponding to $\widetilde{PA}$. But this can only happen if a transmission error or a hash collision occurs. In this case, all the variables change by at most a constant and therefore, so does $\Phi^{\mathrm{A}}_{\mathrm{PD}}$.

If the meeting point transition occurs on Bob's side the proof would be similar with the role of Alice and Bob switched, except in this case q_{MA} increases by at most 1.

- **Scenario 7:** $k_{PA} = k_{\widetilde{PA}} = k$ and status resets occur on both sides due to meeting point transitions: In this case, the variables $E^{\mathrm{A}}_{\mathrm{PD}}$,$BV^{\mathrm{A}}_{\mathrm{PD}}$, and $BI^{\mathrm{A}}_{\mathrm{PD}}$ get reset to 0. Therefore, any change in these variables can only decrease $\Phi^{\mathrm{A}}_{\mathrm{PD}}$. The variables k_{PA} and $k_{\widetilde{PA}}$ get reset to 1. This increases $\Phi^{\mathrm{A}}_{\mathrm{PD}}$ by $2\eta_1\,(k-1)$. The remaining variables change by at most $2k$. First note that if $BI^{\mathrm{A}}_{\mathrm{PD}} > k/100$ then, for sufficiently large η_6, the decrease due to resetting $BI^{\mathrm{A}}_{\mathrm{PD}}$ dominates any other change in the potential function and $\Phi^{\mathrm{A}}_{\mathrm{PD}}$ decreases by at least 1. Suppose that $BI^{\mathrm{A}}_{\mathrm{PD}} \le k/100$. We consider the following different possibilities:

  - $\overline{pd^{\mathrm{A}}_-} \neq 0$: In this case, at least one party has backtracked to a meeting point that does not match any of the other party's meeting points in the current iteration. As argued in Scenario 6, we have $BV^{\mathrm{A}}_{\mathrm{PD}} \ge 0.4k - 2BI^{\mathrm{A}}_{\mathrm{PD}} \ge 0.38k$. Hence, for sufficiently large η_5, the decrease in $\Phi^{\mathrm{A}}_{\mathrm{PD}}$ due to resetting $BV^{\mathrm{A}}_{\mathrm{PD}}$ dominates any other potential change and $\Phi^{\mathrm{A}}_{\mathrm{PD}}$ decreases by at least 1.
  - $\overline{pd^{\mathrm{A}}_-} = 0$ and $k < 8pd^{\mathrm{A}}_-$: In this case $\Phi^{\mathrm{A}}_{\mathrm{MD}}$ decreases by at least

    $$-2k + \eta_2 pd^{\mathrm{A}}_- - 2\eta_1\,(k-1) \ge [\eta_2/8 - 2\eta_1 - 2]\,k \ ,$$

    which is at least 1 for sufficiently large η_2 and every $k \ge 1$.
  - $\overline{pd^{\mathrm{A}}_-} = 0$ and $k \ge 8pd^{\mathrm{A}}_-$: In this case, Alice and Bob should have backtracked to a matching meeting point earlier. First note that if $E^{\mathrm{A}}_{\mathrm{PD}} \ge k/100$ then the decrease due to resetting $E^{\mathrm{A}}_{\mathrm{PD}}$ guarantees an overall decrease by at least 1, for large enough η_4. Suppose that $E^{\mathrm{A}}_{\mathrm{PD}} < k/100$. Note that if $\hat{k}_{PA} = \hat{k}_{\widetilde{PA}} \ge 2^{\lfloor \log pd^{\mathrm{A}}_- \rfloor + 1}$ at the beginning of an iteration then Alice and Bob have at least one matching meeting point. Moreover, $k \ge 8pd^{\mathrm{A}}_-$ implies that $k/4 \ge 2^{\lfloor \log pd^{\mathrm{A}}_- \rfloor + 1}$. Consider the first iteration after the last status reset starting with $k_{PA}, k_{\widetilde{PA}} \ge k/4$. Without loss of generality, suppose that $k_{PA} = k/4$ at the beginning of this iteration. Note that the maximum distance between k_{PA} and $k_{\widetilde{PA}}$ after the last status reset has been at most $BI^{\mathrm{A}}_{\mathrm{PD}}$. Therefore, in at least $k/4 - BI^{\mathrm{A}}_{\mathrm{PD}}$ iterations, $\hat{k}_{PA} = \hat{k}_{\widetilde{PA}} = k/4$ and the variable $k_{\widetilde{PA}}$ has increased by 1, i.e., Bob has conducted the subroutine MPrewind on the rewinding data corresponding to $\widetilde{PA}$. The fact that Bob did not have enough votes for any of his meeting points when $k_{\widetilde{PA}}$ reached the value $k/2$ implies that

    $$\left(k/4 - BI^{\mathrm{A}}_{\mathrm{PD}}\right) - E_{\widetilde{PA}} - BI^{\mathrm{A}}_{\mathrm{PD}} - BV^{\mathrm{A}}_{\mathrm{PD}} < 0.8\,(k/4) \ .$$

So we have $BV^{\mathrm{A}}_{\mathrm{PD}} > 0.02k$ and for sufficiently large η_5, the decrease due to resetting $BV^{\mathrm{A}}_{\mathrm{PD}}$ guarantees an overall decrease in $\Phi^{\mathrm{A}}_{\mathrm{PD}}$ by at least 1.

This completes the proof. □

We use the following lemma to bound the number of iterations suffering from a Pauli data hash collision.

Lemma 5.5.14. *Suppose that in the first i iterations of Algorithm 28, recycling is successful, the number of iterations suffering from a metadata hash collision is at most $c_1 n\epsilon$ and the number of iterations suffering from a quantum hash collision is at most $c_3 n\epsilon$. Then the number of iterations suffering from a Pauli data hash collision in the first i iterations is at most $c_5 n\epsilon$ with probability at least $1 - 2^{-\Theta(n\epsilon)}$.*

Proof. We call an iteration a *dangerous iteration of type III* if at the beginning of the iteration, either $pd^{\mathrm{A}}_{-} + pd^{\mathrm{B}}_{-} \neq 0$ or $k^{\mathrm{A}}_{\mathrm{PD}} + k^{\mathrm{B}}_{\mathrm{PD}} > 4$. Note that Pauli data hash collisions can only occur in these iterations. Let h_{PD} denote the number of iterations in the first i iterations suffering from a Pauli data hash collision. Let d_{III} denote the number of type III dangerous iterations in the first i iterations. It suffices to prove that

$$\Pr\left(d_{\mathrm{III}} > c_5 n\epsilon\right) \leq 2^{-\Theta(n\epsilon)} \ .$$

By the assumption, the number of iterations suffering from a metadata hash collision is at most $c_1 n\epsilon$ and the number of iterations suffering from a quantum hash collision is at most $c_3 n\epsilon$. Therefore, by Lemmas 5.5.9 and 5.5.12, the total number of type I and type II recovery iterations in the first i iterations is at most $(c_2 + c_4)\, n\epsilon$.

- In type III dangerous iterations $\Psi_{\mathrm{PD}} > 0$. Therefore, by Lemma 5.5.13, in these iterations Φ_{PD} decreases by at least 1, if no error or hash collision occurs and the iteration is not a type I or type II recovery iteration. In the remaining type III dangerous iterations Φ_{PD} increases by at most a fixed constant c^{+}_{PD}.
- In iterations that are not type III dangerous, if no transmission error, metadata hash collision or quantum hash collision occurs and the iteration is not a type I or type II recovery iteration, then Φ_{MD} remains unchanged when $\Psi_{\mathrm{MD}} = 0$, and it decreases by at least 1 when $\Psi_{\mathrm{MD}} > 0$. Otherwise, Φ_{PD} increases by at most c^{+}_{PD}.

Hence, the accumulated potential Φ_{PD} in the first i iterations is at most

$$c^{+}_{\mathrm{PD}}\left(h_{\mathrm{PD}} + (2 + c_1 + c_2 + c_3 + c_4)\, n\epsilon\right) - \left(d_{\mathrm{III}} - h_{\mathrm{PD}} - (2 + c_1 + c_2 + c_3 + c_4)\, n\epsilon\right) \ .$$

By Lemma 5.5.1, Φ_{PD} is bounded below by a value in $-\mathrm{O}\left(n\epsilon\right)$. Therefore, we have $\left(1 + c^{+}_{\mathrm{PD}}\right) h_{\mathrm{PD}} - d_{\mathrm{III}} \in -\mathrm{O}\left(n\epsilon\right)$. For sufficiently large c_5, if $d_{\mathrm{III}} > c_5 n\epsilon$ then $h_{\mathrm{PD}} \geq \frac{d_{\mathrm{III}}}{2\left(1 + c^{+}_{\mathrm{PD}}\right)}$.

However, for collision probability $p = 2^{-o}$ satisfying $p < \frac{1}{20\left(1+c^{+}_{\mathrm{PD}}\right)}$, the probability of having such a large fraction of collisions during the dangerous iterations is very small. In particular, hash collisions are $2^{-\Theta\left(n\sqrt{\epsilon}\right)}$-statistically close to being dominated by independent Bernoulli(p) variables. Therefore, Chernoff bound implies that the probability of having such a large number of collisions is at most $2^{-\Theta(n\epsilon)}$. □

Definition 5.5.15. We refer to an iteration of Algorithm 28 as a *recovery iteration of type III* if it is not a type I or type II recovery iteration and at least one of Alice or Bob conducts one of the cases vi.A or vi.B.

The following lemma is used to bound the number of type III recovery iterations.

Lemma 5.5.16. *Suppose that in the first i iterations of Algorithm 28, recycling is successful and the number of iterations suffering from metadata, quantum, and Pauli data hash collisions is at most $c_1 n\epsilon$, $c_3 n\epsilon$ and $c_5 n\epsilon$, respectively. Then the total number of type III recovery iterations in the first i iterations is at most $c_6 n\epsilon$.*

Proof. Note that by Lemma 5.5.1, we always have $\Phi_{\mathrm{PD}} \in -\mathrm{O}\left(n\epsilon\right)$. Moreover, Ψ_{PD} is always non-negative and it is equal to zero if and only if $k_{PA} = k_{\widetilde{PA}} = k_{PB} = k_{\widetilde{PB}} = 1$ and Alice and Bob have full knowledge of each other's Pauli data.

If at the beginning of an iteration $\Psi_{\mathrm{PD}} = 0$, then the iteration is a type III recovery iteration only if

- A transmission error occurs on the Pauli data messages, or else,
- A metadata hash collision or a transmission error on metadata messages occurs. In this case at least one party incorrectly believes that his/her estimate of the other party's Pauli data is not full-length.

Therefore, in the first i iterations the total number of type III recovery iterations starting with $\Psi_{\mathrm{PD}} = 0$ is at most $\left(c_1 + 2\right) n\epsilon$.

Let β_{PD} denote the number of iterations starting with $\Psi_{\mathrm{PD}} > 0$ in the first i iterations. We prove an upper bound on β_{PD}. By Lemma 5.5.13, in each iteration, regardless of the number of errors and collisions, Φ_{PD} increases by at most some fixed constant. Moreover, Φ_{PD} increases only in type I or Type II recovery iterations or iterations with errors or hash collisions. Therefore, by the assumption, the accumulated potential Φ_{PD} in such iterations is at most $\mathrm{O}\left(n\epsilon\right)$. In the remaining iterations starting with $\Psi_{\mathrm{PD}} > 0$, the potential function Φ_{PD} decreases by at least 1. However, $\Phi_{\mathrm{PD}} \in -\mathrm{O}\left(n\epsilon\right)$ implies that the number of such iterations is also at most $\mathrm{O}\left(n\epsilon\right)$.

Therefore, the total number of type III recovery iterations in the first i iterations is at most $c_6 n\epsilon$, for sufficiently large c_6. □

As in the large alphabet case, we let ω_i denote the number of iterations suffering from a transmission error or a hash collision in the first i iterations, plus the number of recovery iterations of type I, II, or III, in the first i iterations. Note that the constants and the probabilities in Lemmas 5.5.7, 5.5.9, 5.5.10, 5.5.12, 5.5.14, and 5.5.16 are all independent of the iteration number i. As a corollary we have:

Corollary 5.5.17. *There exist $q = 2^{-\Theta(n\epsilon)}$ and a constant c_7 such that, for every $i \in [R_{\text{total}}]$, assuming successful recycling in the first i iterations of Algorithm 28, except with probability at most q, we have $\omega_i \leq c_7 n\epsilon$.*

Next we prove that recycling is successful in every iteration of the simulation. First, we need the following lemma. Recall that we use RA and RB with the iteration number i as a super-script to denote their value at the end of the i-th iteration.

Lemma 5.5.18. *Let $t = c_8 n\epsilon$ where $c_8 > 3c_7$. Then for every $i \in [R_{\text{total}}]$ where $i \geq t$, if recycling is successful in the first $i-1$ iterations and $\omega_{i-1} \leq c_7 n\epsilon$, then:*

1. *$RA^i[1 : i-t] = RB^i[1 : i-t]$, i.e., at the end of iteration i, the recycling data of Alice and Bob agree in a prefix of length at least $i-t$.*
2. *$RA^i[1 : i-t] = RA^{i+1}[1 : i-t]$, i.e., the prefix of RA of length $i-t$ does not get modified in the next iteration. The same statement holds for RB.*
3. *For every $k \in [i-t]$ such that $RA^i[k] = RB^i[k] = \mathsf{S}$, we have $W_k = 0^{4r}$.*

Proof. The proof is exactly the same as the proof of Lemma 4.5.16, but in terms of different constants $c_1, \ldots, c_8$ used in the current section. □

Lemma 5.5.19. *Let $L_{\text{QVC}} = c_9 n\epsilon$, where $c_9 > c_7 + c_8$. Then with probability at least $1 - 2^{-\Theta(n\epsilon)}$, recycling is successful throughout the execution of Algorithm 28.*

Proof. The exact same inductive argument used in Lemma 4.5.17 is applicable here, using the constants $c_1, \ldots, c_9$ of this section. □

The following two lemmas characterize the behaviour of the overall potential function in each iteration, assuming successful recycling in every iteration of the simulation.

Lemma 5.5.20. *Assuming successful recycling throughout the execution of Algorithm 28, each iteration of the algorithm, regardless of the number of hash collisions and transmission errors, decreases the potential function Φ, defined in Eq. (5.1), by at most some fixed constant c^-.*

Proof. By Lemmas 5.5.6 and 5.5.13, in each iteration, the potential functions Φ_{MD} and Φ_{PD} increase by at most a fixed constant. Moreover, all the variables appearing in Φ_{RD} and Φ_{Q} change by at most a constant. Therefore, in any iteration, regardless of the number of errors and hash collisions, the potential function Φ decreases by at most some constant. □

Lemma 5.5.21. *Assuming successful recycling throughout the execution of Algorithm 28, each iteration with no transmission error or hash collision increases the potential function* Φ *by at least* 1.

Proof. Note that in an iteration with no error or hash collision Alice and Bob agree on the iteration type. Moreover, if *Itertype* = MD, RD or PD, they also agree on whether they extend or rewind the data and if *Itertype* = MES (Case ii), then exactly one of them is in sub-case A and the other one in sub-case B. We analyze the potential function in each case keeping in mind the hierarchy of the cases; e.g., Case ii or later cases are encountered only if Alice and Bob have full knowledge of each other's metadata. Note that $\Psi_{\text{MD}} = 0$ on entering Case ii, $\Psi_{\text{MD}} = \Phi_{\text{RD}} = 0$ on entering Case vi and $\Psi_{\text{MD}} = \Phi_{\text{RD}} = \Psi_{\text{PD}} = 0$ on entering Case vii.

- Alice and Bob are both in Case i.A or both in Case i.B:
  - $\Psi_{\text{MD}} > 0$, thus by Lemma 5.5.6, Φ_{MD} decreases by at least 1.
  - Φ_{RD}, Φ_{PD} and Φ_{Q} stay the same.

  Therefore, Φ increases by at least 1.
- Alice is in Case ii.A, Bob is in Case ii.B:
  - $\Psi_{\text{MD}} = 0$, thus by Lemma 5.5.6, Φ_{MD} stays the same.
  - rd^{+} and rd^{-} stay the same.
  - $\ell^{\text{A}}_{\text{QVC}}$ stays the same and $\ell^{\text{B}}_{\text{QVC}}$ increases by 1.
  - q_{MA} stays the same and q_{MB} increases by 1.
  - All other variables appearing in $\Phi^{\text{A}}_{\text{PD}}$ and $\Phi^{\text{B}}_{\text{PD}}$ stay the same.
  - g stays the same, b increases by at most 1, and u decreases by 1.

  Therefore, Φ_{RD}, Φ_{PD} and Φ_{Q} increase by 1, 6 and at least 8, respectively. So Φ increases by at least 1.
- Alice is in Case ii.B, Bob is in Case ii.A: This case is similar to the one above.
- Alice and Bob are in Case iii:
  - $\Psi_{\text{MD}} = 0$, thus by Lemma 5.5.6, Φ_{MD} stays the same.
  - rd^{+}, $\ell^{\text{A}}_{\text{QVC}}$ and $\ell^{\text{B}}_{\text{QVC}}$ stay the same.
  - rd^{-} decreases by 1.
  - q_{MA} and q_{MB} do not decrease. q_{MA} increases by 1, if $RA\left[\ell_{RA}\right] = \mathsf{S}$. Similarly, q_{MB} increases by 1, if $RB\left[\ell_{RB}\right] = \mathsf{S}$.
  - All other variables appearing in $\Phi^{\text{A}}_{\text{PD}}$ and $\Phi^{\text{B}}_{\text{PD}}$ stay the same.

- Φ_{Q} stays the same.

Therefore, Φ_{RD} decreases by 13 and Φ_{PD} increases by at most 12. So Φ increases by at least 1.

- Alice and Bob are in Case iv:
  - $\Psi_{\mathrm{MD}} = 0$, thus by Lemma 5.5.6, Φ_{MD} stays the same.
  - rd^{+}, $\ell^{\mathrm{A}}_{\mathrm{QVC}}$ and $\ell^{\mathrm{B}}_{\mathrm{QVC}}$ stay the same.
  - rd^{-} decreases by 1.
  - q_{MA} and q_{MB} do not decrease. They both increase by 1, if $RA\,[\ell_{RA}] = RB\,[\ell_{RB}] = \mathsf{S}$.
  - All other variables appearing in $\Phi^{\mathrm{A}}_{\mathrm{PD}}$ and $\Phi^{\mathrm{B}}_{\mathrm{PD}}$ stay the same.
  - Φ_{Q} stays the same.

  Therefore, as in Case iii, Φ_{RD} decreases by 13 and Φ_{PD} increases by at most 12. So Φ increases by at least 1.

- Alice and Bob are in Case v:
  - $\Psi_{\mathrm{MD}} = 0$, thus by Lemma 5.5.6, Φ_{MD} stays the same.
  - rd^{-} stays at 0 and rd^{+} increases by 1.
  - $\ell^{\mathrm{A}}_{\mathrm{QVC}}$ and $\ell^{\mathrm{B}}_{\mathrm{QVC}}$ stay the same.
  - Φ_{PD} and Φ_{Q} stay the same.

  Therefore, Φ_{RD} decreases by 2 and Φ increases by 2.

- Alice and Bob are both in Case vi.A or both in Case vi.B:
  - $\Psi_{\mathrm{MD}} = 0$, thus by Lemma 5.5.6, Φ_{MD} stays the same.
  - Φ_{RD} stays at 0.
  - $\Psi_{\mathrm{PD}} > 0$, thus by Lemma 5.5.13, Φ_{PD} decreases by at least 1.
  - Φ_{Q} stays the same.

  Therefore, Φ increases by at least 1.

- Alice and Bob are in Case vii:
  - $\Psi_{\mathrm{MD}} = 0$, thus by Lemma 5.5.6, Φ_{MD} stays the same.
  - Φ_{RD} stays at 0.
  - $\Psi_{\mathrm{PD}} = 0$, thus by Lemma 5.5.13, Φ_{PD} stays the same.
  - u stays at 0.

- If $b \neq 0$ then g stays the same and b decreases by 1, otherwise, b stays at 0 and g increases by 1.

Therefore, Φ_{Q} increases by 1 and so does Φ.

So assuming successful recycling throughout the execution of the algorithm, the potential function Φ increases by at least 1, in every iteration with no transmission error or hash collision. □

Finally, we are ready to prove our main result:

Theorem 5.2.1 (**Restated**). *Consider any n-round alternating communication protocol Π in the plain quantum model, communicating messages over a noiseless channel with a constant size alphabet Σ. Algorithm 28 is a quantum coding scheme which given Π, simulates it with probability at least $1-2^{-\Theta(n\epsilon)}$, over any fully adversarial error channel with alphabet Σ and error rate ϵ. The simulation uses $n\left(1+\Theta\left(\sqrt{\epsilon}\right)\right)$ rounds of communication, and therefore achieves a communication rate of $1-\Theta\left(\sqrt{\epsilon}\right)$.*

Proof. Let $R_{\mathrm{total}} = \left\lceil \frac{n}{2r} \right\rceil + \alpha n\epsilon$, where α is a sufficiently large constant to be determined. By Lemma 5.5.19, recycling is successful throughout the execution of the algorithm with probability at least $1-2^{-\Theta(n\epsilon)}$. Assuming successful recycling, by Lemmas 5.5.7, 5.5.10 and 5.5.14, the total number of iterations with a hash collision is at most $\left(c_1+c_3+c_5\right)n\epsilon$ except with probability $2^{-\Theta(n\epsilon)}$. Since the number of iterations is less than $2n$, the total number of iterations with a transmission error is at most $2n\epsilon$. Therefore, by Lemma 5.5.21, in the remaining $R_{\mathrm{total}} - \left(c_1+c_3+c_5+2\right)n\epsilon$ iterations the potential function Φ increases by at least 1. The potential function decreases in an iteration only if a hash collision or a transmission error occurs and by Lemma 5.5.20, it decreases by at most a fixed constant c^-. As shown in the proof of Lemma 5.5.1, $\Phi_{\mathrm{MD}} \geq -80\lambda_1 n\epsilon$, $\Phi_{\mathrm{RD}} \geq 0$, and $\Phi_{\mathrm{PD}} \geq -200\eta_1 n\epsilon$. For $\alpha \stackrel{\mathrm{def}}{=} \left(c^-+1\right)\left[c_1+c_3+c_5+2\right] + 80\lambda_1 + 200\eta_1$, at the end of the simulation, we have

$$\begin{aligned} g-b-u \geq \Phi_{\mathrm{Q}} &\geq \Phi - \left(80\lambda_1+200\eta_1\right)n\epsilon \\ &\geq \left[R_{\mathrm{total}} - \left(c_1+c_3+c_5+2\right)n\epsilon\right] - c^-\left(c_1+c_3+c_5+2\right)n\epsilon - \left(80\lambda_1+200\eta_1\right)n\epsilon \\ &\geq \frac{n}{2r} \ . \end{aligned}$$

Therefore, the simulation is successful with probability at least $1-2^{-\Theta(n\epsilon)}$. The cost of entanglement distribution is $\Theta\left(n\sqrt{\epsilon}\right)$. Moreover, the amount of communication in each iteration is independent of the iteration type and is always $(2r+\Theta(1))$: in every iteration each party sends $\Theta(1)$ symbols to communicate the hash values in line 18 of Algorithm 28; each party sends another r symbols either in line 22 of Algorithm 28, if *Itertype* $\neq$ SIM or

in Algorithm 26, i.e., the **Q-simulate** subroutine. Hence, we have

$$\begin{aligned}\text{Total number of communicated qudits} &= \Theta\left(n\sqrt{\epsilon}\right) + R_{\text{total}} \cdot (2r + \Theta(1)) \\ &= \Theta\left(n\sqrt{\epsilon}\right) + \left(\left\lceil \frac{n}{2r} + \Theta\left(n\epsilon\right)\right\rceil\right)(2r + \Theta(1)) \\ &= n\left(1 + \Theta\left(\sqrt{\epsilon}\right)\right) \quad .\end{aligned}$$

□

Chapter 6

Conclusions and outlook

6.1 Summary

In this thesis, we have studied simulation of noiseless two-party interactive quantum communication via low noise channels. A lower bound of $1 - \Theta(\sqrt{\epsilon})$ on the optimal achievable communication rate is proved in the plain model of quantum communication against adversarial noise of rate ϵ. To achieve this goal, we first studied the teleportation-based model in which the parties have access to free entanglement and the communication is over a noisy classical channel. In this model, we showed the same lower bound of $1 - \Theta(\sqrt{\epsilon})$ in the large alphabet case. Then we adapted the framework developed for the teleportation-based model to the plain quantum model in which the parties do not have access to pre-shared entanglement and communicate over a noisy quantum channel. We showed how quantum Vernam cipher can be used in the interactive communication setting to efficiently recycle and reuse entanglement, allowing us to simulate any input protocol with an overhead of only $1 + \Theta(\sqrt{\epsilon})$. Finally, we combined our framework with meeting point-based rewinding to extend our results to constant-size communication alphabets. Our coding scheme beats the currently best known overhead of $1 - \mathrm{O}(\sqrt{\epsilon \log\log 1/\epsilon})$, in the corresponding plain *classical* model, which is also conjectured to be optimal in [37].

6.2 Implications of our results

In this work, we have studied the capacity of noisy quantum channels to implement two-way communication. In particular, we studied the ability of quantum channels to simulate interactive two-party communication, with the channel available in both directions, but without any assistance by side resources, e.g., classical side channels. As discussed in Section 1.1.4, this can be seen as a generalization of channel coding. Quantum effects give rise to surprising phenomena which make it more challenging to characterize the capacity of quantum channels for reliable communication compared to their better understood classical

counterparts. For example, the qubit erasure channel with erasure probability $\frac{1}{2}$ has no one-way quantum capacity [7]. But when the channel can be used in either direction, noisy backwards classical communication becomes possible, and one can show a lower bound of $\frac{1}{10}$ on the capacity [7, 49]. A similar effect happens to the qubit depolarizing channel [8, 16]. Thus, comparing memoryless channels in the classical and the quantum setting, the one-way capacity of a classical channel is always an upper bound on its two-way capacity, while this does not necessarily hold for all quantum channels. These effects enrich the subject but also add to the challenge of determining the capacity of quantum channels for reliable communication. Not much is known about the two-way quantum capacity. For general memoryless quantum channels, the two-way capacity is only known to be upper bounded by the entanglement-assisted quantum capacity Q_E [9, 10], which is equal to the quantum feedback capacity [12]. This bound is not tight (for example, for very noisy qubit depolarizing channel, the two-way capacity vanishes but $Q_E > 0$). As discussed in Section 1.2.1, coding seems much harder in the interactive setting and our work presents important progress in the low-noise regime. For the qubit depolarizing channel with noise rate ϵ, in the low noise regime, the one-way capacity is $1 - H(\epsilon) + \epsilon \log 3 + \mathrm{O}(\epsilon^2)$ [48]. We have established an achievable rate of $1 - \Theta(\sqrt{\epsilon})$ against adversarial noise for interactive communication. If our conjectured optimality holds, in this case the interactive capacity will be lower than one-way capacity in the dependence on ϵ.

A further implication of our result is that quantum communication complexity is very robust against transmission noise at low error rates. In particular, for alternating protocols like those considered in this paper and most known protocols in the communication complexity setting, the overhead factor goes to one as the noise goes to zero, allowing one to get the full quantum advantage whenever such an advantage can be obtained.

6.3 Open questions

This relatively new area of research is still vastly unexplored. Here we present a few interesting research directions to pursue following this work.

Interactive capacity of quantum channels. Two questions stem directly from our work. First, we conjecture that a rate of $1 - \mathrm{O}(\sqrt{\epsilon})$ is optimal, up to leading order in dependence on ϵ. Is this conjecture true, and if so, what is the constant hidden in the O notation? Second, what is the optimal rate of communication in the high noise regime, for large ϵ?

The *interactive capacity* of a channel can be defined as the optimal asymptotic rate of simulating noiseless interactive communication over the noisy channel. The only non-trivial upper bound on the interactive capacity of channels is due to Kol and Raz [46] in the classical setting. They prove a tight upper bound on the interactive capacity of the binary symmetric channel by studying the communication cost of the *pointer jumping game*–a

well-studied problem in communication complexity–over this noisy channel. Computing this function requires a high level of interaction in the quantum setting [45], making it a "hard" task to achieve over noisy quantum channels and a promising candidate to study. The heart of the difficulty in proving such a result lies in the fact that as opposed to what is usually satisfactory in the (noiseless) communication complexity setting where we prove bounds that are tight up to multiplicative constants, here we need to prove a bound that is tight up to second order terms, while the communication is over a noisy channel.

Adaptive simulation of non-alternating protocols. Another important direction is concerning the fact that our coding scheme assumes that the protocol to be simulated is alternating, i.e., Alice and Bob alternate in sending qudits to each other. We believe that a lot of the machinery that we have developed should transpose very well to the more general setting where the protocol to be simulated has a more general structure, potentially with messages constructed from different number of qudits in different rounds. As discussed in Ref. [37], for such protocols in order to achieve a high communication rate it is necessary to use *adaptive* simulation protocols. In an adaptive protocol, the communication order is not predetermined and each party independently determines whether to speak or listen in the next round according to their view of the simulation. When the parties' views of the simulation differ this can lead to communication rounds in which both parties speak or listen simultaneously. The channel's behaviour must be carefully defined in each of these scenarios.

Interactive quantum communication against insertion-deletion errors. In the current work, we already have to deal with many types of synchronization errors at the teleportation, Quantum Vernam Cipher and quantum hashing level. An interesting question from this perspective is: what about synchronization errors over the channel itself?

Synchronization errors arising in data transmission are modeled by *insertion-deletion* channels. This error model, which has gained considerable attention in recent years, is a generalization of the better-understood model of corruption errors. Recently, there has been much interest in the classical interactive coding literature towards such type of errors [19, 39, 62]. In particular, Haeupler and Shahrasbi [38] introduced *synchronization strings*, novel combinatorial structures which were used to devise coding schemes against insertion-deletion errors for interactive classical communication [39] and later, for one-way quantum communication [47]. Inspired by these recent developments, we believe that our framework can be adapted to design an algorithm for protecting interactive quantum communication against adversarial insertion-deletion errors.

Multiparty noisy interactive quantum communication. A natural extension of the task considered in this work is multiparty distributed computation over arbitrary networks with noisy communication links. A naive strategy in the multiparty setting would be to break any given input protocol designed for noiseless communication into sub-protocols

between pairs of adjacent parties and then simulate every sub-protocol using a two-party interactive coding scheme. However, this approach runs into several fundamental problems. Each outgoing message of a party is a function of the received messages from all other parties in earlier rounds. As a result, a local error over a communication link in any part of the network can propagate throughout the network after some rounds of communication. Therefore, every locally detected error must be signaled to the other parties to trigger a *global* rewind. This would require a synchronization mechanism to maintain a global view of the simulation with some reasonable delay at every node. Depending on the topology of the underlying network, several algorithms have been proposed to coordinate the simulation (see, e.g., [54, 2]). We believe that the framework we have developed can be used to translate these ideas to the quantum setting and reproduce many of the known classical results. However, it would be more interesting to investigate whether purely quantum effects that are not available in the classical setting can be utilized to design new algorithms.

Private noisy interactive quantum communication. Consider two parties Alice and Bob who given access to a noisy communication channel wish to compute a function $f(x, y)$ of their respective private inputs x and y, while preserving their privacy in an information theoretic sense; that is, they do not want to reveal any information about their inputs to each other beyond what they learn from $f(x, y)$. Surprisingly, Gelles, Sahai, and Wadia [34] show that in the classical setting, it is not always possible to simultaneously achieve privacy and error-resilience against adversarial noise. Another no-go result [24] in this direction has been shown for "knowledge-preserving" simulation of interactive protocols against adversarial noise, where the goal is for the encoded protocol to carry the same amount of information as the original protocol.

Does quantum communication allow one to evade the classical no-go results obtained for private interactive communication in the cryptographic settings above? As we have seen in this work, the unique properties of quantum information can be helpful in the interactive communication setting, since we were able to achieve a higher communication rate over fully adversarial binary channels in the plain model than the conjectured upper bound in the corresponding plain classical setting.

Fully fault-tolerant interactive quantum communication. Considering a larger fault-tolerant setting due to the inherently fragile nature of quantum data, can we perform high rate interactive quantum communication when the local quantum computations are also noisy?

Many other interesting directions of research in the quantum setting stem from recent exciting developments in the classical setting, for example [13, 15, 14, 32, 33, 35, 17, 36, 28, 30, 40, 5]. We believe that our framework should be extendable to the study of many of these problems in the quantum setting.

References

[1] Scott Aaronson and Andris Ambainis. Quantum search of spatial regions. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, pages 200–209. IEEE, 2003.

[2] Noga Alon, Mark Braverman, Klim Efremenko, Ran Gelles, and Bernhard Haeupler. Reliable communication over highly connected noisy networks. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*, PODC '16, page 165–173, New York, NY, USA, 2016. Association for Computing Machinery.

[3] Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. Simple constructions of almost k-wise independent random variables. *Random Structures & Algorithms*, 3(3):289–304, 1992.

[4] Erdal. Arikan. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Transactions on Information Theory*, 55(7):3051–3073, July 2009.

[5] Young-Han Kim Assaf Ben-Yishai, Ofer Shayevitz. Interactive coding for Markovian protocols. In *55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 870–877, 2017.

[6] Charles H Bennett, Gilles Brassard, Richard Jozsa, Dominic Mayers, Asher Peres, Benjamin Schumacher, and William K Wootters. Reduction of quantum entropy by reversible extraction of classical information. *Journal of Modern Optics*, 41(12):2307–2314, 1994.

[7] Charles H. Bennett, David P. DiVincenzo, and John A. Smolin. Capacities of quantum erasure channels. *Phys. Rev. Lett.*, 78:3217–3220, Apr 1997.

[8] Charles H Bennett, David P DiVincenzo, John A Smolin, and William K Wootters. Mixed-state entanglement and quantum error correction. *Phys. Rev. A*, 54(5):3824–3851, 1996.

[9] Charles H Bennett, Peter W Shor, John A Smolin, and Ashish V Thapliyal. Entanglement-assisted classical capacity of noisy quantum channels. *Phys. Rev. Lett.*, 83(15):3081–3084, 1999.

[10] Charles H Bennett, Peter W Shor, John A Smolin, and Ashish V Thapliyal. Entanglement-assisted capacity of a quantum channel and the reverse Shannon theorem. *IEEE Transactions on Information Theory*, 48(10):2637–2655, 2002.

[11] Hector Bombin. Gauge color codes: optimal transversal gates and gauge fixing in topological stabilizer codes. *New Journal of Physics*, 17(8):083002, 2015.

[12] Garry Bowen. Quantum feedback channels. *IEEE Transactions on Information Theory*, 50(10):2429–2434, 2004.

[13] Zvika Brakerski and Yael Tauman Kalai. Efficient interactive coding against adversarial noise. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science*, pages 160–166. IEEE, 2012.

[14] Zvika Brakerski, Yael Tauman Kalai, and Moni Naor. Fast interactive coding against adversarial noise. *J. ACM*, 61(6):35:1–35:30, December 2014.

[15] Zvika Brakerski and Moni Naor. Fast algorithms for interactive coding. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 443–456. Society for Industrial and Applied Mathematics, 2013.

[16] Gilles Brassard, Ashwin Nayak, Alain Tapp, Dave Touchette, and Falk Unger. Noisy interactive quantum communication. *SIAM Journal on Computing*, 48(4):1147–1195, 2019.

[17] Mark Braverman and Klim Efremenko. List and unique coding for interactive communication in the presence of adversarial noise. *SIAM Journal on Computing*, 46(1):388–428, 2017.

[18] Mark Braverman, Ankit Garg, Young Kun Ko, Jieming Mao, and Dave Touchette. Near-optimal bounds on bounded-round quantum communication complexity of disjointness. In *Proceedings of the 2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, FOCS '15, pages 773–791, Washington, DC, USA, 2015. IEEE Computer Society.

[19] Mark Braverman, Ran Gelles, Jieming Mao, and Rafail Ostrovsky. Coding for interactive communication correcting insertions and deletions. *IEEE Transactions on Information Theory*, 63(10):6256–6270, 2017.

[20] Mark Braverman and Anup Rao. Toward coding for maximum errors in interactive communication. *IEEE Transactions on Information Theory*, 60(11):7248–7255, 2014.

[21] Harry Buhrman, Richard Cleve, and Avi Wigderson. Quantum vs. classical communication and computation. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 63–68. ACM, 1998.

[22] A Robert Calderbank, Eric M Rains, Peter W Shor, and Neil JA Sloane. Quantum error correction via codes over GF(4). *IEEE Transactions on Information Theory*, 44(4):1369–1387, 1998.

[23] A. Robert Calderbank and Peter W. Shor. Good quantum error-correcting codes exist. *Phys. Rev. A*, 54:1098–1105, Aug 1996.

[24] Kai-Min Chung, Rafael Pass, and Sidharth Telang. Knowledge-preserving interactive coding. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 449–458, 2013.

[25] Igor Devetak. The private classical capacity and quantum capacity of a quantum channel. *IEEE Transactions on Information Theory*, 51(1):44–55, 2005.

[26] Dennis Dieks. Communication by EPR devices. *Phys. Lett. A*, 92(6):271–272, 1982.

[27] David DiVincenzo, Peter Shor, and John Smolin. Quantum-channel capacity of very noisy channels. *Physical Review A*, 57(2):830–839, 1998.

[28] Klim Efremenko, Ran Gelles, and Bernhard Haeupler. Maximal noise in interactive communication over erasure channels and channels with feedback. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, ITCS '15, pages 11–20, New York, NY, USA, 2015. ACM.

[29] Keqin Feng and Zhi Ma. A finite gilbert-varshamov bound for pure stabilizer quantum codes. *IEEE Transactions on Information Theory*, 50(12):3323–3325, Dec 2004.

[30] Matthew Franklin, Ran Gelles, Rafail Ostrovsky, and Leonard J. Schulman. Optimal coding for streaming authentication and interactive communication. *IEEE Transactions on Information Theory*, 61(1):133–145, Jan 2015.

[31] Ran Gelles. Coding for interactive communication: A survey. *Foundations and Trends® in Theoretical Computer Science*, 13(1–2):1–157, 2017.

[32] Ran Gelles, Ankur Moitra, and Amit Sahai. Efficient and explicit coding for interactive communication. In *Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science*, pages 768–777. IEEE, 2011.

[33] Ran Gelles, Ankur Moitra, and Amit Sahai. Efficient coding for interactive communication. *IEEE Transactions on Information Theory*, 60(3):1899–1913, March 2014.

[34] Ran Gelles, Amit Sahai, and Akshay Wadia. Private interactive communication across an adversarial channel. *IEEE Transactions on Information Theory*, 61(12):6860–6875, 2015.

[35] Mohsen Ghaffari and Bernhard Haeupler. Optimal error rates for interactive coding ii: Efficiency and list decoding. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science*, pages 394–403. IEEE, 2014.

[36] Mohsen Ghaffari, Bernhard Haeupler, and Madhu Sudan. Optimal error rates for interactive coding i: Adaptivity and other settings. In *Proceedings of the Forty-sixth Annual ACM Symposium on Theory of Computing*, STOC '14, pages 794–803, New York, NY, USA, 2014. ACM.

[37] Bernhard Haeupler. Interactive channel capacity revisited. In *Proceedings of the 2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, FOCS '14, pages 226–235, Washington, DC, USA, 2014. IEEE Computer Society.

[38] Bernhard Haeupler and Amirbehshad Shahrasbi. Synchronization strings: Codes for insertions and deletions approaching the singleton bound. In *Proceedings of the 49th Annual ACM Symposium on Theory of Computing*, STOC 2017, pages 33–46, New York, NY, USA, 2017. ACM.

[39] Bernhard Haeupler, Amirbehshad Shahrasbi, and Ellen Vitercik. Synchronization strings: Channel simulations and interactive coding for insertions and deletions. In *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, volume 107 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 75:1–75:14, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[40] Bernhard Haeupler and Ameya Velingker. Bridging the capacity gap between interactive and one-way communication. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '17, pages 2123–2142, Philadelphia, PA, USA, 2017. Society for Industrial and Applied Mathematics.

[41] Matthew B Hastings. Superadditivity of communication capacity using entangled inputs. *Nature Physics*, 5(4):255, 2009.

[42] Alexander S Holevo. The capacity of the quantum channel with general signal states. *IEEE Transactions on Information Theory*, 44(1):269–273, 1998.

[43] Peter Høyer and Ronald De Wolf. Improved quantum communication complexity bounds for disjointness and equality. In *STACS 2002*, pages 299–310, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[44] Rahul Jain, Jaikumar Radhakrishnan, and Pranab Sen. A lower bound for the bounded round quantum communication complexity of set disjointness. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, pages 220–229. IEEE, 2003.

[45] Hartmut Klauck, Ashwin Nayak, Amnon Ta-Shma, and David Zuckerman. Interaction in quantum communication. *IEEE Transactions on Information Theory*, 53(6):1970–1982, 2007.

[46] Gillat Kol and Ran Raz. Interactive channel capacity. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, pages 715–724. ACM, 2013.

[47] Janet Leahy, Dave Touchette, and Penghui Yao. Quantum insertion-deletion channels. Technical Report arXiv:1901.00984 [quant-ph], arXiv.org, `https://arxiv.org/abs/1901.00984`, January 2019.

[48] Felix Leditzky, Debbie Leung, and Graeme Smith. Quantum and private capacities of low-noise channels. In *2017 IEEE Information Theory Workshop (ITW)*, pages 484–488, 2017.

[49] Debbie Leung, Joungkeun Lim, and Peter Shor. Capacity of quantum erasure channel assisted by backwards classical communication. *Phys. Rev. Lett.*, 103:240505, Dec 2009.

[50] Debbie W. Leung. Quantum vernam cipher. *Quantum Info. Comput.*, 2(1):14–34, December 2002.

[51] Seth Lloyd. Capacity of the noisy quantum channel. *Phys. Rev. A*, 55(3):1613–1622, 1997.

[52] Joseph Naor and Moni Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM Journal on Computing*, 22(4):838–856, 1993.

[53] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge University Press, Cambridge, UK, 2000.

[54] Sridhar Rajagopalan and Leonard Schulman. A coding theorem for distributed computation. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '94, page 790–799, New York, NY, USA, 1994. Association for Computing Machinery.

[55] Ran Raz. Exponential separation of quantum and classical communication complexity. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 358–367. ACM, 1999.

[56] Oded Regev and Bo'az Klartag. Quantum one-way communication can be exponentially stronger than classical communication. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 31–40. ACM, 2011.

[57] Leonard J Schulman. Communication on noisy channels: A coding theorem for computation. In *Proceedings of the 33rd Annual IEEE Symposium on Foundations of Computer Science*, pages 724–733. IEEE, 1992.

[58] Leonard J Schulman. Deterministic coding for interactive communication. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, pages 747–756. ACM, 1993.

[59] Leonard J Schulman. Coding for interactive communication. *IEEE Transactions on Information Theory*, 42(6):1745–1756, 1996.

[60] Benjamin Schumacher and Michael D Westmoreland. Sending classical information via noisy quantum channels. *Phys. Rev. A*, 56(1):131–138, 1997.

[61] Claude E. Shannon. A mathematical theory of communication. *Bell System Tech. J.*, 27:379–423, 623–656, 1948.

[62] Alexander A Sherstov and Pei Wu. Optimal interactive coding for insertions, deletions, and substitutions. *IEEE Transactions on Information Theory*, 65(10):5971–6000, 2019.

[63] Peter W Shor. The quantum channel capacity and coherent information. Lecture notes, MSRI Workshop on Quantum Computation, 2002.

[64] Graeme Smith and Jon Yard. Quantum communication with zero-capacity channels. *Science*, 321(5897):1812–1815, 2008.

[65] Norbert Stolte. *Rekursive Codes mit der Plotkin-Konstruktion und ihre Decodierung*. PhD thesis, TU Darmstadt, Fachbereich Elektrotechnik und Informationstechnik„ 2002.

[66] John Watrous. *The Theory of Quantum Information*. Cambridge University Press, May 2018.

[67] Mark M Wilde. *Quantum Information Theory*. Cambridge University Press, Cambridge, UK, 2013.

[68] William K Wootters and Wojciech H Zurek. A single quantum cannot be cloned. *Nature*, 299(5886):802–803, 1982.