Interactive Proofs For Quantum Computations

Dorit Aharonov* Michael Ben-Or* Elad Eban* Urmila Mahadev†

April 17, 2017

Abstract

The widely held belief that BQP strictly contains BPP raises fundamental questions: upcoming generations of quantum computers might already be too large to be simulated classically. Gottesman asked ([Got04]): Is it possible to experimentally test that these systems perform as they should, if we cannot efficiently compute predictions for their behavior? As phrased by Vazirani in [Vaz07]: If computing predictions for quantum mechanics requires exponential resources, is quantum mechanics a falsifiable theory? In cryptographic settings, an untrusted future company wants to sell a quantum computer or perform a delegated quantum computation. Can the customer be convinced of correctness without the ability to compare results to predictions?

To provide answers to these questions, we define Quantum Prover Interactive Proofs (QPIP). Whereas in standard interactive proofs [GMR85] the prover is computationally unbounded, here our prover is in BQP, representing a quantum computer. The verifier models our current computational capabilities: it is a BPP machine, with access to only a few qubits. Our main theorem can be roughly stated as: "Any language in BQP has a QPIP which hides the computation from the prover". We provide two proofs. The simpler one uses a new (possibly of independent interest) quantum authentication scheme (QAS) based on random Clifford elements. This QPIP, however, involves two way quantum communication for polynomially many rounds. Our second protocol uses polynomial codes QAS due to Ben-Or, Crépeau, Gottesman, Hassidim, and Smith [BOCG$^+$06], combined with secure multiparty quantum computation techniques. This protocol involves quantum communication from the verifier to the prover at the start of the protocol, and classical communication throughout the rest of the protocol. Both protocols are inherently "blind": both the quantum circuit and the input remain unknown to the prover.

This is the journal version of work reported in 2008 ([ABOE08]) and presented in ICS 2010. The protocols are slightly modified from the original version, whereas some of the proofs required major modifications and corrections. Notably, the claim that the polynomial QPIP is fault tolerant was removed.

After deriving the results in [ABOE08], we learnt that Broadbent, Fitzsimons, and Kashefi [BFK08] have independently suggested "universal blind quantum computation" using completely different methods (measurement based quantum computation). Their construction implicitly implies similar implications. The protocol in [BFK08] was flawed but based on similar ideas, Fitzsimons and Kashefi have provided a protocol and proof of blind verifiable computation in [FK12]. The initial independent works ([ABOE08],[BFK08]) ignited a long line of research of blind and verifiable quantum computation, which we survey here, along with connections to various cryptographic problems. Importantly, the problems of making the results fault tolerant, as well as removing the need for quantum communication altogether, remain open.

*School of Computer Science, The Hebrew University of Jerusalem, Israel. {doria,benor,elade}@cs.huji.ac.il

†Department of Computer Science, UC Berkeley, California. mahadev@cs.berkeley.edu

1 Introduction

1.1 Motivation

As far as we know today, the quantum mechanical description of many-particle systems requires exponential resources to simulate. This has the following fundamental implication: the results of an experiment conducted on a many-particle physical system described by quantum mechanics cannot be predicted (in general) by classical computational devices in any reasonable amount of time. This important realization (or belief), which stands at the heart of the interest in quantum computation, led Gottesman [Got04] to ask: Can a classical verifier verify the correctness of quantum evolutions? The question was phrased by Vazirani [Vaz07] as: Is quantum mechanics a falsifiable physical theory? Assuming that small quantum systems obey quantum mechanics to an extremely high accuracy, it is still possible that the physical description of large systems deviates significantly from quantum mechanics. Since there is no efficient way to make the predictions of the experimental outcomes for most large quantum systems, there is no way to test or falsify this possibility experimentally, using the usual scientific paradigm of predict and compare.

This question has practical implications. Experimentalists who attempt to realize quantum computers would like to know how to test that their systems indeed perform the way they should. But most tests cannot be compared to any predictions! The tests whose predictions can in fact be computed do not actually test the more interesting aspects of quantum mechanics, namely those which cannot be simulated efficiently classically.

The problem arises in cryptographic situations as well. It is natural to expect that the first generations of quantum computers will be extremely expensive, and thus quantum computations would be delegated to untrusted companies. Is there any way for the customer to trust the outcome, without the need to trust the company which performed the computation, even though the customer cannot verify the outcome of the computation (since he cannot simulate it)? And even if the company is honest, can the customer detect innocent errors in such a computation? Given the amounts of grant money and prestige involved, the possibility of dishonesty of experimentalists and experimentalists' bias inside the academia should not be ignored either [Roo03, Wik08].

As Vazirani points out [Vaz07], an answer to these questions is already given in the form of Shor's factoring algorithm [Sho97]. Indeed, quantum mechanics does not seem to be falsifiable using the *usual* scientific paradigm, assuming that BQP is strictly lager than BPP. However, Shor's algorithm does provide a way for falsification, by means of an experiment which lies outside of the usual scientific paradigm: though its result cannot be *predicted* and then compared to the experimental outcome, it can be *verified* once the outcome of the experiment is known (by simply taking the product of the factors and checking that this gives the input integer).

This, however, does not fully address the issues raised above. Consider, for example, a company called *Q-Wave* which is trying to convince a customer that it has managed to build a quantum computer of 100 qubits. Such a system is already too big to simulate classically. However, any factoring algorithm that is run on a system of a 100 qubits can be easily performed by today's classical technology. For delegated quantum computations, how can Shor's algorithm help in convincing a customer of correctness of, say, the computation of the BQP complete problem of approximating the Jones polynomial [AJL06, AA06, FKLW01, BFLW09]? As for experimental results, it is difficult to rigorously state which aspects of quantum mechanics are exactly falsified or verified by the possibility to apply Shor's algorithm. Moreover, we are now facing a time in which small quantum computers of a few tens of qubits may very well be realizable; yet, factoring is still impossible in such systems, and we would nevertheless like to be able to test their evolution.

We thus pose the following main question: Can one be convinced of the correctness of the computation of *any*

polynomial quantum circuit? Alternatively, can one be convinced of the "correctness" of the quantum mechanical description of any quantum experiment that can be conducted in the laboratory, even though one cannot compute the predictions for the outcomes of this experiment? In this paper, we address the above fundamental question in a rigorous way and provide a positive answer to these questions, in a well defined framework. We do this by taking a computational point of view on the interaction between the supposed quantum computer, and the entity which attempts to verify that the quantum computer indeed computes what it should.

1.2 Quantum Prover Interactive Proofs (QPIP)

Interactive proof systems, defined by Goldwasser, Micali and Rackoff [GMR85], play a crucial role in the theory of computer science. Roughly, a language $\mathcal{L}$ is said to have an interactive proof if there exists a computationally unbounded prover (denoted $\mathbb{P}$) and a BPP verifier ($\mathbb{V}$) such that for any instance x in $\mathcal{L}$, $\mathbb{P}$ convinces $\mathbb{V}$ of the fact that $x \in \mathcal{L}$ with probability $\geq \frac{2}{3}$ (completeness). Otherwise, when $x \notin \mathcal{L}$, there does not exist a prover who can convince $\mathbb{V}$ that $x \in \mathcal{L}$ with probability higher than $\frac{1}{3}$ (soundness).

Quantum interactive proofs in which the prover is an *unbounded* quantum computer, and the *verifier* is a BQP machine have previously been studied in [Wat03]. The starting point of this work is the observation that Shor's factoring algorithm [Sho97] can be viewed as an interactive proof of a very different kind: one between a classical BPP verifier, and a quantum *polynomial time* (BQP) prover, in which the prover convinces the verifier of the factors of a given number (this can be easily converted to the usual IP formalism of membership in a language by converting the search problem to a decision problem in the standard way).

One might suspect that such an interactive proof exists for all problems inside $\mathsf{BQP} \cap \mathsf{NP}$ by asking the BQP prover to find the witness, which the classical verifier can then verify. We do not know this to be true; the trouble with this argument is that the fact that the problem is in $\mathsf{BQP} \cap \mathsf{NP}$ does not guarantee that the BQP machine can also *find* a witness efficiently - decision to search reductions are known only for NP-complete problems. In any case, it is widely believed that BQP is not contained in NP (and in fact not even in the polynomial hierarchy - see [Aar09] and references therein). The main goal of this paper is to generalize the interactive point of view of Shor's algorithm, as mentioned above, in order to show that a BPP verifier can be convinced of the result of *any* polynomial quantum circuit, using interaction with the BQP prover (the quantum computer). In other words, we would like to extend the above interactive proof (which is specific to factoring) to a BQP complete problem.

To this end we define a new model of quantum interactive proofs which we call quantum prover interactive proofs (QPIP). The simplest definition would be an interactive proof in which the prover is a BQP machine and the verifier a BPP classical machine. In some sense, this model captures the possible interaction between the quantum world (for instance, quantum systems in the lab) and the classical world. However, we do not know how to provide interactive proofs for all problems in BQP with only classical interaction; this is a major open problem (see Section 1.8). We therefore modify the model a little, and allow the verifier additional access to a constant number of qubits. The verifier can be viewed as modeling our current computational abilities, and so in some sense, the verifier represents "us".

Definition 1.1 *A Language $\mathcal{L}$ is said to have a Quantum Prover Interactive Proof (QPIP_κ) with completeness c and soundness s (where $c - s$ is a constant) if there exists a pair of algorithms $(\mathbb{P}, \mathbb{V})$, where $\mathbb{P}$ is the prover and $\mathbb{V}$ is the verifier, with the following properties:*

- *The prover $\mathbb{P}$ is a BQP machine, which also has access to a quantum channel which can transmit κ qubits.*
- *The verifier $\mathbb{V}$ is a hybrid quantum-classical machine. Its classical part is a BPP machine. The quantum part is a register of κ qubits, on which the verifier can perform arbitrary quantum operations and which has*

access to a quantum channel which can transmit κ qubits. At any given time, the verifier is not allowed to possess more than κ qubits. The interaction between the quantum and classical parts of the verifier is the usual one: the classical part controls which operations are to be performed on the quantum register, and outcomes of measurements of the quantum register can be used as input to the classical machine.

- *There is also a classical communication channel between the prover and the verifier, which can transmit polynomially many bits at any step.*
- *At any given step, either the verifier or the prover perform computations on their registers and send bits and qubits through the relevant channels to the other party.*

We require:

- ***Completeness**: if $x \in \mathcal{L}$, then after interacting with $\mathbb{P}$, $\mathbb{V}$ accepts with probability $\geq c$.*
- ***Soundness**: if $x \notin \mathcal{L}$, then the verifier rejects with probability $\geq 1 - s$ regardless of the prover $\mathbb{P}'$ (who has the same description as $\mathbb{P}$) with whom he is interacting.*

Abusing notation, we denote the class of languages for which such a proof exists also by QPIP_κ. Throughout the paper, when we refer to QPIP without a subscript, we are assuming the subscript is a constant c. We remark that our definition of QPIP_κ is asymmetric - the verifier is "convinced" only if the quantum circuit outputs 1. This asymmetry seems irrelevant in our context of verifying correctness of quantum computations. Indeed, it is possible to define a symmetric version of QPIP_κ (which we denote by $\mathsf{QPIP}_\kappa^{sym}$) in which the verifier is convinced of *correctness* of the prover's outcome whether or not $x \in \mathcal{L}$ rather than only if $x \in \mathcal{L}$; see Appendix A for the definition.

1.3 Main Results

Our main results are phrased in terms of the BQP complete problem $\mathsf{Q\text{-}CIRCUIT}_\gamma$:

Definition 1.2 *The promise problem $\mathit{Q\text{-}CIRCUIT}_\gamma$ consists of a quantum circuit made of a sequence of gates, $U = U_N \ldots U_1$, acting on n input bits. The task is to distinguish between two cases for all $x \in \{0,1\}^n$:*

$$\begin{aligned} \mathit{Q\text{-}CIRCUIT}_{YES} &: \|((|1\rangle\langle 1| \otimes \mathcal{I}_{n-1})\, U\, |x\rangle\|^2 \geq 1 - \gamma \\ \mathit{Q\text{-}CIRCUIT}_{NO} &: \|((|1\rangle\langle 1| \otimes \mathcal{I}_{n-1})\, U\, |x\rangle\|^2 \leq \gamma \end{aligned}$$

when we are promised that one of the two cases holds.

$\mathsf{Q\text{-}CIRCUIT}_\gamma$ is a BQP complete problem as long as $1 - 2\gamma > \frac{1}{poly(n)}$. Throughout this paper, if we refer to Q-CIRCUIT (without the parameter γ), we are assuming that γ satisfies the above inequality. Our main result is:

Theorem 1.1 *For $0 < \epsilon < 1$ and $\gamma < 1 - \epsilon$, the language $\mathit{Q\text{-}CIRCUIT}_\gamma$ has a $\mathit{QPIP}_{O(\log(\frac{1}{\epsilon}))}$ with completeness $1 - \gamma$ and soundness $\epsilon + \gamma$.*

We note that although we provide QPIP protocols only for the language Q-CIRCUIT (for which the initial state is always a standard basis state), our proofs can be extended to security for a modified language for which the initial state is an arbitrary quantum state. In addition, we prove soundness against an unbounded prover, rather than a BQP prover (as given in Definition 1.1). By setting ϵ to a constant, we obtain a QPIP_c for a constant c, which gives our main theorem:

Theorem 1.2 *There exists a constant c for which $\mathsf{BQP} = \mathit{QPIP}_c$.*

Proof: Since Q-CIRCUIT is BQP complete, the fact that BQP is in QPIP_c follows from Theorem 1.1. QPIP_c is trivially in BQP since the BQP machine can simulate both prover, verifier and their interactions. □

Thus, a BQP prover can convince the verifier of any language he can compute. Since BQP is closed under completion, we also get equality to the symmetric version of QPIP_c (see Appendix A for the proof):

Corollary 1.3 *There exists a constant c for which* $\mathsf{BQP} = \mathit{QPIP}_c^{sym}$

Our main tools for the proof of Theorem 1.1 are quantum authentication schemes (QAS) [BCG$^+$02]. Roughly, a QAS allows two parties to communicate in the following way. First, Alice sends an encoded quantum state to Bob. Then Bob decodes the state and decides whether or not it is valid. If the state was not altered along the way, then upon decoding, Bob gets the same state that Alice had sent (and declares it valid). If the state was altered, the scheme is ϵ-secure if Bob declares a wrong state valid with probability at most ϵ. The basic idea used to extend a QAS to a QPIP is that similar security can be achieved, even if the state needs to be rotated by unitary gates, as long as the verifier can control how the unitary gates affect the authenticated states.

1.3.1 Clifford QAS based QPIP

We start with a simple QAS (which we extend to a QPIP) based on random Clifford group operations (it is reminiscent of Clifford based quantum t-designs [AE07, ABW08]). The Clifford QAS based QPIP demonstrates some key ideas and might be of interest on its own due to its simplicity. However, the QPIP has the disadvantage that it requires two way quantum communication between the prover and the verifier.

We first describe the Clifford QAS. To encode a state of n qubits, Alice tensors the state with e qubits in the state $|0\rangle$, and applies a random Clifford operator on the $n+e$ qubits. To decode, Bob removes the Clifford operator chosen by Alice and checks if the e auxiliary qubits are in the state $|0\rangle^{\otimes e}$ (see Protocol 3.1 for a complete description of the QAS). We prove the following theorem:

Theorem 1.4 *The Clifford scheme given in Protocol 3.1 is a QAS with security $\epsilon = 2^{-e}$.*

This QAS might be interesting in its own right due to its simplicity. To construct a QPIP using this QAS, we simply use the prover as an untrusted storage device: the verifier asks the prover for the authenticated qubits on which he would like to apply the next gate, decodes them by applying the appropriate inverse Clifford operators, applies the gate, applies new random Clifford operators and sends the resulting qubits to the prover. As we show in the following theorem, this protocol (see Protocol 4.1 for full details) is a QPIP:

Theorem 1.5 *For $0 < \epsilon < 1$ and $\gamma < 1 - \epsilon$, Protocol 4.1 is a $\mathit{QPIP}_{O(\log(\frac{1}{\epsilon}))}$ with completeness $1 - \gamma$ and soundness $\gamma + \epsilon$ for $\mathit{Q\text{-}CIRCUIT}_\gamma$.*

1.3.2 Polynomial code QAS based QPIP

Our second type of QPIP uses a QAS due to Ben-Or, Crépeau, Gottesman, Hassidim and Smith [BOCG$^+$06]. This QAS is based on signed quantum polynomial codes (defined in Definition 2.6) , which are quantum polynomial codes [ABO97] of degree at most d multiplied by some random sign (1 or -1) at every coordinate (this is called the sign key k) and a random Pauli at every coordinate (the Pauli key). The QAS simply consists of Alice encoding a single qudit using the signed polynomial code and Bob checking if the received state is indeed encoded under the signed polynomial code (described in further detail in Protocol 5.1). We prove the following theorem:

Theorem 1.6 *The polynomial authentication scheme as described in Protocol 5.1 is a QAS with security $\epsilon = 2^{-d}$.*

The security proof of the polynomial code based QAS is subtle, and was missing from the original paper [BOCG$^+$06]; we provide it here.

To extend the polynomial based QAS to a QPIP, we first note that performing Clifford gates in this setting is very easy. Due to its algebraic structure, the signed polynomial code allows applying Clifford gates without knowing the sign key (if the same sign key is used for all registers). This was used in [BOCG$^+$06] for secure multiparty quantum computation; here we use it to allow the prover to perform gates without knowing the sign key or the Pauli key. To perform Toffoli gates, the verifier first creates authenticated magic states (used in [Sho96],[BK05],[BOCG$^+$06]) and sends them to the prover. The prover can apply a Toffoli gate by first applying Clifford operations between the computation qubits and a magic state, then measuring 3 of the computation qubits, and then adaptively applying a Clifford correction based on the measurement results (for more details on applying Toffoli gates using Toffoli states see Section 2.3.1). Note that since the prover obtains a measurement result encoded under the polynomial QAS, he must send it to the verifier to be decoded before he can perform an adaptive Clifford correction. It follows that with authenticated magic states and classical assistance from the verifier, the prover can perform universal computation using only Clifford group operations and measurements (universality was proved for qubits in [BK05] and for higher dimensions it was shown in [ABO97]).

The polynomial QPIP protocol (Protocol 6.1) goes as follows. The prover receives all authenticated qubits in the beginning. Those include the inputs to the circuit, as well as authenticated magic states required to perform Toffoli gates. The prover can then perform universal computation as described above. Except for the first round, any further communication between the verifier and prover (occuring when implementing the Toffoli gates) is thus classical. We show that this protocol is a QPIP in the following theorem:

Theorem 1.7 *For $0 < \epsilon < 1$ and $\gamma < 1 - \epsilon$, Protocol 6.1 is a $\mathsf{QPIP}_{O(\log(\frac{1}{\epsilon}))}$ protocol with completeness $1 - \gamma$ and soundness $\gamma + \epsilon$ for $\mathsf{Q\text{-}CIRCUIT}_\gamma$.*

We remark that in the study of the classical notion of IP, a natural question is to ask how powerful the prover must be to prove certain classes of languages. It is known that a PSPACE prover is capable of proving any language in PSPACE to a BPP verifier, and similarly, it is known that NP or #P restricted provers can prove any language which they can compute to a BPP verifier. This is not known for coNP, SZK or PH [AB09]. It is natural to ask what is the power of a BQP prover; our results imply that such a prover can prove the entire class of BQP (albeit to a verifier who is not entirely classical). Thus, we provide a characterization of the power of a BQP prover. We stress the open question of characterizing this power when the interaction between the prover and verifier is completely classical (discussed in Section 1.8).

1.3.3 Blindness

In the works [Chi01, AS06] a related question was raised: in our cryptographic setting, if we distrust the company performing the delegated quantum computation, we might want to keep both the input and the function which is being computed secret. Can this be done while maintaining the confidence in the outcome? A simple modification of our protocols to work on universal quantum circuits gives the following theorem, which we prove in Section 7:

Theorem 1.8 *Theorem 1.1 holds also in a blind setting, namely, the prover does not get any information regarding the function being computed and its input.*

We note that an analogous result for NP-hard problems was shown already in the late 80's to be impossible (in the setting of classical communication) unless the polynomial hierarchy collapses [AFK87].

To achieve Theorem 1.8, we modify our construction so that the circuit that the prover performs is a *universal quantum circuit*, i.e., a fixed sequence of gates which gets as input a description of a quantum circuit (with gates from a constant size universal set of gates) plus an input string to that circuit, and applies the input quantum circuit to the input string. Since the universal quantum circuit is fixed, it reveals nothing about the input quantum circuit or the input string to it. To prove blindness, we simply need to show that the input states provided to the prover by the verifier (and all messages provided to the prover during the protocol) do not leak information about the input to the universal circuit. This is done by showing that at all times, the prover's state is independent of the input to the universal circuit.

Proving blindness of the Clifford scheme is quite straightforward and done in the following theorem (which we prove in Section 7):

Theorem 1.9 (Blindness of the Clifford Based QPIP) *The state of the prover in the Clifford based QPIP (Protocol 4.1) is independent of the input to the circuit which is being computed throughout the protocol.*

On the other hand, proving blindness of the polynomial scheme is a bit more involved, due to the classical interaction:

Theorem 1.10 (Blindness of the Polynomial Based QPIP) *The state of the prover in the polynomial based QPIP (Protocol 6.1) remains independent of the input to the circuit which is being computed throughout the protocol.*

1.3.4 Interpretation

We will now present some corollaries which clarify the connection between the results and the motivating questions, and show that one can use the QPIP protocols designed here to address the various issues raised in Sec. 1.1.

We start with some basic questions. Conditioned that the verifier does not abort, does he know that the final state of the machine is very close to the correct state that was supposed to be the outcome of the computation? This unfortunately is not the case. It may be that the prover can make sure that the verifier aborts with very high probability, but when he does *not* abort, the outcome of the computation is wrong. However a weaker form of the above result (which achieves what is reasonable to hope for) does hold: if we know that the probability of not aborting is high, then we can deduce something about the probability of the final state being very close to the correct state.

Corollary 1.11 *For the Clifford based QPIP protocol with security parameter ϵ, if the verifier does not abort with probability $\geq \beta$ then the trace distance between the final density matrix conditioned on the verifier's accepting and the correct final state, is at most $\frac{\epsilon}{\beta}$.*

The proof is given in Section 8 - it is simple for the Clifford QPIP. For the polynomial scheme a similar corollary holds, with a proof which is more involved:

Corollary 1.12 *For the polynomial based QPIP protocol with security parameter ϵ, if the verifier does not abort with probability $\geq \beta$, and the correct final state is a standard basis state, then the trace distance between the final density matrix conditioned on the verifier's acceptance and the correct final state is at most $\frac{\epsilon}{\beta}$.*

The following corollary (which we prove in Appendix A) contains another implication of Theorem 1.2. We show that under a somewhat stronger assumption than BQP $\neq$ BPP, but still a widely believed assumption, it is possible to lower bound the computational power of the prover (and deduce that the prover is not within BPP) by efficiently testing the prover (assuming the prover passes the test with high probability).

Corollary 1.13 *Assume that there is a language $L \in$ BQP and there is a polynomial time sampleable distribution D on the instances of L on which any* BPP *machine errs with non negligible probability (e.g. the standard cryptographic assumption about the hardness of Factoring or Discrete Log). If the verifier runs a QPIP (with soundness $\gamma + \epsilon$ and completeness $1 - \gamma$) on Q-CIRCUIT$_\gamma$ on an instance drawn from D and does not abort with probability $\geq \beta$ (where $\beta - \frac{4\epsilon}{1-2\gamma}$ is a constant), then the prover's computational power cannot be simulated by a* BPP *machine.*

1.4 Proofs Overview

As mentioned, we rely on two different quantum authentication schemes (and their proofs of security) in order to derive and prove completeness and soundness of our two QPIPs.

1.4.1 Clifford QAS

To prove the security of the Clifford QAS (as stated in Theorem 1.4) we first prove in Lemma 3.2 that any non identity attack of Eve is mapped by the random Clifford operator to a uniform mixture over all non identity Pauli operators. We call this property of Clifford operators (as stated in Lemma 3.2) operator decohering by Cliffords or in short, *Clifford decoherence*. Next, we show that the uniform mixture over Pauli operators changes the auxiliary 0 states used in the Clifford authentication scheme with high probability, and the non identity attack is therefore likely to be detected by Bob's decoding procedure.

1.4.2 Clifford QPIP

To prove the soundness of the Clifford based QPIP (stated in Theorem 1.5), we use Clifford decoherence to reduce the soundness of the QPIP to the security of the QAS. We do this by showing in Claim 4.2 that Clifford decoherence (Lemma 3.2) allows us to shift all attacks of the prover to the end of the protocol (at which point we can simply apply the security proof of the QAS). This shifting is clearly possible if the prover's attack is the identity operator. If the prover's attack is non identity, Clifford decoherence maps the prover's attack to a uniform mixture over all non identity Pauli operators. It follows that after the verifier decodes the state sent by the prover, the state will essentially be maximally mixed, and whatever the verifier applies at this point commutes with the prover's attack which is currently acting on the state (this is shown in Lemma 4.6). Note that it is important that the verifier does not check the states for correctness in each round (by measuring the auxiliary qubits and checking whether they are 0); instead, he only checks in the final round. If the verifier had instead checked the states for correctness in each round, we could not have used the shifting technique due to Clifford decoherence, since the verifier would be applying a non unitary operator (measurement). We thereby obtain a simple QPIP, with a rather short proof.

The key disadvantage of this protocol is the two way quantum communication required in each round.

1.4.3 Polynomial based QAS

To strengthen the results, we use a polynomial based QPIP instead. The proof of the security of the corresponding QAS (Theorem 1.6) requires some care, due to a subtle point which was not addressed in [BOCG$^+$06]. To prove Theorem 1.6, we first prove in Lemma 5.2 that no non identity Pauli attack can preserve the signed polynomial code for more than 2 of the sign keys, and thus the sign key suffices in order to protect against any non identity Pauli attack of Eve's. Next, we need to show that the scheme is secure against general attacks. This, surprisingly, does not follow by linearity from the security against Pauli attacks (as is the case in quantum error correcting codes): if we omit the Pauli key we get an authentication scheme which is secure against Pauli attacks but not against general

attacks[1]. We proceed by showing (with some similarity to the Clifford based QAS) that the random Pauli key effectively translates Eve's attack to a mixture (not necessarily uniform like in the Clifford case) of Pauli operators acting on a state encoded by a random signed polynomial code. We call this property of random Pauli keys *Pauli decoherence* (see Lemma 5.1).

1.4.4 Polynomial based QPIP

We proceed to proving the soundness of the polynomial based QPIP (as stated in Theorem 1.7). Unlike in the Clifford case, the soundness of the polynomial QPIP cannot be directly reduced to the security of the polynomial QAS, because the prover's attack cannot be shifted to the end of the protocol in the polynomial QPIP. This is due to the weakness of Pauli decoherence relative to Clifford decoherence. In more detail, Clifford decoherence first maps the prover's attack to a convex sum over Pauli operators, and then further maps each non-identity Pauli operator to a uniform mixture over all non identity Pauli operators. Pauli decoherence only performs the first step: it maps the prover's attack to a convex sum over Pauli operators (which are weighted according to the original attack). This does not create a maximally mixed state, and therefore does not allow the same shifting of the prover's attack to the end of the protocol. Thus, the proof of Theorem 1.7 does not use the proof of the polynomial QAS (Theorem 1.6) as a black box, and in fact that proof is not strictly needed for the proof of Theorem 1.7. However, we included Theorem 1.6 and its proof for completeness (as it was not written before), and mainly because the two key ideas used in that proof will also be used in the proof of the polynomial based QPIP. Recall that these two key ideas are Pauli decoherence from Lemma 5.1 and security of the sign key against Pauli attacks from Lemma 5.2 (which takes up most of the technical effort involved in proving Theorem 1.6).

To prove soundness of the polynomial QPIP, we first note that if all of the classical messages sent from the verifier to the prover were fixed ahead of time, shifting the prover's attacks to the end of the protocol would be fine, as the verifier's messages to the prover do not depend on the prover's measurement results. Once we shift the prover's attacks, we can apply the two main ideas (Pauli decoherence from Lemma 5.1 and security of the sign key from Lemma 5.2) used in the proving the security of the polynomial QAS to obtain soundness of the polynomial QPIP. However, in the actual polynomial QPIP protocol, the classical interaction does depend on the prover's messages. We employ an idea from [FK12] (see Figure 7 in their paper): as part of the analysis, we fix the interaction transcript at the start of the protocol (to allow shifting of the prover's attacks) and then project onto this fixed interaction transcript at the end of the protocol to enforce consistency. This technique (which is formalized in Claim 6.1) essentially partitions the prover's Hilbert space according to the interaction transcript, and we can then apply the two key ideas used in proving the security of the polynomial QAS in each partition.

1.4.5 Blindness

Proving blindness of the Clifford scheme (as stated in Theorem 1.9) is quite straightforward and is done by showing that, due to the randomness of the Clifford encoding operator, the prover's state is maximally mixed at all times. This is because applying a random Clifford operator on a state results in a maximally mixed state (see Lemma 4.4). See Section 7 for the full proof.

Proving blindness of the polynomial scheme (as stated in Theorem 1.10) is a bit more involved due to the classical interaction - see Section 7. Without the classical interaction, we could use the randomness of the Pauli keys to show that the prover's state is always maximally mixed (this relies on the fact that applying a random Pauli operator to a state results in a maximally mixed state, as stated in Lemma 4.5). When we include classical interaction (for

[1]Without Pauli keys, the sign key can be determined up to ± 1 from a measurement of the state. This follows from the uniqueness of signed polynomials, which is proven in Fact 5.1.

the purpose of decoding measurement results), we need to show that the measurement results (even if altered by a malicious prover) do not leak information about the input state. This is due to the fact that measurement results are initially distributed uniformly at random, and even if a malicious prover attacks, his attack can be reduced to a convex sum over Pauli operators (due to Lemma 5.1), which preserves the uniform distribution. Note that we could have simplified this proof significantly by including extra randomness in the magic states, which would serve essentially as a one time pad for the decoded measurement results (this would have complicated the description of the polynomial QPIP protocol, Protocol 6.1) . However, it is interesting to note that this extra randomness is not needed for blindness, and that the protocol is blind due to the randomness of the measurement results and the Pauli keys.

1.4.6 Interpretation

We prove the corollaries (Corollary 1.11 and Corollary 1.12) given in Section 1.3.4 in Section 8. Both proofs rely on using the format of the prover's state, as shown in Claim 4.2 for the Clifford QPIP and Claim 6.1 for the polynomial QPIP, to first determine what the prover's state will look like conditioned on the verifier's acceptance. In the Clifford case, the trace distance can then be determined quite easily, due to the simplicity of Claim 4.2. The format of the prover's final state in the polynomial case is significantly more involved. The proof of Corollary 1.12 proceeds by analyzing the effect of two different types of Pauli attack operators (trivial and non trivial) in order to show that the trace distance between the final state after acceptance and the correct final state is correlated to the probability of acceptance and the security parameter.

1.5 Changes from Conference Version

This journal version is a corrected and elaborated version of the conference version, and a new author (U.M) was added. We describe here in detail the differences from the conference version.

1.5.1 Soundness

Clifford Scheme The main difference between the Clifford QPIP protocol in this version and theconference version is that in this version the verifier checks correctness (by measuring the auxiliary 0 states) only in the final round, whereas in the conference version, the verifier checked correctness each time he received qubits from the prover. The conference version of the protocol is actually not sound; the prover can cheat by deviating only slightly in each round. Since there are polynomially many rounds, this can add up to a significant deviation in the final state, without being detected, using essentially the zeno effect.

The security proof in the original version assumed that all attacks of the provercould be shifted to the end of the protocol; namely, that the prover only deviated at the end of the protocol. While this does not hold in the original protocol, in the new scheme this can be proven, which is what makes the proof go through. The final proof eventually turns out to go along similar lines to the original one, except for this change in the protocol.

Polynomial Scheme The protocol for the polynomial QPIP remained the same. However, the security proof needed to be changed dramatically, due to the same incorrect assumption used in the Clifford QPIP regarding shifting the prover's attacks to the end of the protocol (as mentioned above). Whereas in the Clifford scheme a minor change in the protocol sufficed to guarantee that this assumption actually holds, we did not have such a simple solution in the polynomial scheme. As described in Section 1.4, this is because the weakness of Pauli decoherence relative to Clifford decoherence: Pauli decoherence does not map the attack to a *uniform* mixture over Paulis, thus preventing shifting the prover's attacks to the end of the protocol. The polynomial QPIP proof therefore required major revisions, because we could not simply reduce to the security of the polynomial QAS.

1.5.2 Fault Tolerance

In the original version of the paper a scheme for making the protocol fault tolerant was proposed and was claimed to be secure. Unfortunately, there is a fatal flaw in the proof; we retract the claim about fault tolerance (see open questions in Section 1.8 for possible approaches left for future work).

We describe below the proposal for fault tolerance of [ABOE08] and the bug. The proposed protocol was: at the first stage of the protocol, authenticated qudits are sent from the verifier to the prover, one by one. As soon as the prover receives an authenticated qudit, he protects his qudits using his own concatenated error correcting codes so that the effective error in the encoded authenticated qudit is constant. This constant accuracy can be maintained for a long time by the prover, by performing error correction with respect to *his* error correcting code (see [ABO97]). Thus, polynomially many such authenticated states can be passed to the prover in sequence. A constant effective error is not good enough, but can be amplified to an arbitrary inverse polynomial by purification. Indeed, the prover cannot perform purification on his own since the purification compares authenticated qudits and the prover does not know the authentication code. However, the verifier can help the prover by using classical communication. This way the prover can reduce the effective error on his encoded authenticated qudits to inverse polynomial, and perform the usual fault tolerant construction of the given circuit, with the help of the verifier in performing the gates.

The problem with this approach is that the purification protocol could leak information about the sign key; during the purification protocol the verifier tells the prover which states are good enough for him to keep and which he should throw away. A cheating prover could lie on all of his messages to the verifier; eventually, he will figure out which of his lies will lead to the verifier accepting, and this should give him information about the sign key chosen by the verifier. Once the sign key is no longer hidden from the prover, the QPIP protocol is no longer secure. The problem seems to be difficult. In Section 1.8 we describe why several other possible avenues we tried, in order to achieve fault tolerance, failed; It remains open to achieve blind verifiable QPIPs in the noisy setting, even when we allow the verifier to hold a polylogarithmic quantum register, rather than a constant one.

1.6 Related Work

Related Work in Blindness and Verifiability The question of delegated blind computation was asked by Childs in [Chi01] and by Arrighi and Salvail in [AS06], who proposed schemes to deal with such scenarios. However [Chi01] does not deal with a cheating prover, so the protocol is not verifiable. Also, the setting is somewhat different; rather than limiting the quantum space of the verifier, the verifier is limited to only performing Pauli gates. In [AS06], Arrighi and Salvail provide a blind interactive quantum protocol in this setting for a restricted set of functions, and prove its security against a restricted set of attacks.

Independent work After deriving the results of the first version of this paper, we learned that Broadbent, Fitzsimons, and Kashefi [BFK08] have claimed related results. Using measurement based quantum computation, they construct a protocol for universal blind quantum computation. In their case, it suffices that the verifier's register consists of a single qubit. Their results have similar implications to ours in terms of the QPIP notion, though these are implicit in [BFK08]. However, their protocol was not secure against general attacks (as noted in [FK12]). However, based on similar ideas, Fitzsimons and Kashefi suggested a measurement based protocol which is both verifiable and blind, and prove its security in [FK12] (a key idea they used to prove security was also useful in our proof of the polynomial QPIP, as described in Section 1.4).

Follow-up Work Since the results presented here were first posted [ABOE08], together with the [BFK08] paper, there had been a surge of results investigating the notions of blind quantum computation, verifiable quantum computation, the ability to perform those in a noisy environment fault tolerantly, as well as experimental demonstrations.

As mentioned, Fitzimons and Kashefi gave a different QPIP protocol which is both verifiable and blind, based on measurement based quantum computation [FK12]. Our protocol seems to be simpler to state, but the [FK12] has the advantage of only requiring a single qubit at the verifier's end.

In [BGS12], Broadbent, Gutoski and Stebila provided a framework for analysing blind QPIPs in the context of one time quantum programs; a sketch for a proof of the blindness (but not of the verifiability) of a protocol very similar to our polynomial based protocol (Protocol 6.1) is given in that paper in Section 6.1. In [MF16], Morimae and Fitzsimons proposed a very nice and simple QPIP protocol which is just verifiable but not blind. Moreover, it requires the verifier only to be able to measure qubits in the standard or Hadamard basis; it is based on the idea of the prover generating the history state known from Kitaev's QMA proof ([KSV02]). Additional blind QPIP protocols were proposed in [HM15], [Mor14] and [Bro15].

A very interesting question which branched out from the results presented here, was taken by Reichardt, Unger and Vazirani[RUV12]. In their work, they proposed a protocol in which a BPP verifier could verify a BQP computation by only classical interaction, when interacting with *two* BQP entangled provers [RUV12]. Since then, there have been several papers which have explored the model of multiple BQP provers and a single BPP verifier (such as [Mck13], [GKW15], [HPDF15], [HH16]).

The difficulties in providing a fault tolerant blind verifiable protocol with only $O(1)$ qubits at the verifier's end seem hard to get around. There have been several attempts to suggest solutions, including the conference version of this paper [ABOE08] as well as [FK12, TFMI16] but to the best of our knowledge this problem remains importantly open, even when the verifier is allowed to hold a polylogarithmic quantum register.
.
As for experimental demonstrations, we mention a few: of blind computing in [BKB$^+$12],[GRB$^+$16] and of verifiable computing in [BFKW13]).

1.7 Fault Tolerance Open Questions and Attempts

Technically, the main open question raised by this work is to provide a fault tolerant version of these results. In work yet to be published ([ASMZ17]), it is shown that fault tolerance can be achieved if only one of the tasks (blindness or verification) is required. However, we do not know how to achieve fault tolerance for both tasks simultanuously. Moreover, we do not even know how to do this when allowing the verifier a quantum register of polylogarithmic size. There seems to be an inherent problem in any of the straightforward approaches to making our schemes fault tolerant, which we now explain.

We already discussed above why the approach which we suggested in the original paper, of purification with the help of the verifier (see Section 1.5.2), failed. Another attempt is to create a fault tolerant version of the Clifford protocol by running a fault tolerant circuit, which involved the prover passing the qubits back to the verifier for correction at every step of the circuit. This seemed to require the verifier to measure and check for errors when correcting, which compromises the soundness of the Clifford protocol as explained in Section 1.5.1 (recall from Section 1.4 that the verifier only checks for errors at the end of the current Clifford protocol).

We also attempted to create a fault tolerant version of the polynomial protocol by using blind computation to allow the prover to create the authenticated states on his own; the prover can then do everything in his lab fault tolerantly. This idea seemed troublesome because the prover did not have to honestly run the blind computation in order to create the authenticated states, and his dishonesty during the state creation phase could potentially compromise

security later on in the protocol.

Finally, we attempted to use standard fault tolerant techniques (e.g. [ABO97]) in order to simulate the QPIP protocol, both by the verifier and the prover. The protocol will start with the verifier (who now has a quantum register of polylogarithmic size) creating a fault tolerant encoding of his authenticated states, and sending those to the prover. The prover will act as expected by the QPIP protocol, but will keep correcting the state with respect to the code used for fault tolerance. Unfortunately, we do not yet know how to extend our security proofs to hold for this protocol, though it may be secure. A natural attempt to prove security would be to reduce the security in the noisy case to that of the ideal case. In other words, we would like to claim that if the protocol is insecure in the noisy setting, then the prover can also cheat in the noiseless setting (by simulating the noise). Unfortunately, we do not know how to claim that the prover can simulate the effect of the noise acting on the authenticated states, since the noise may depend on the private keys of the verifier (this is because the verifier's circuit to create the authenticated states depends on these keys). One might hope to use error correction techniques to remove the dependence of the noise on the keys, but this approach turns out to fail due to a very subtle issue - namely, due to teleportation-like effects, dependencies on the keys may propagate through the error correction to qubits which were previously subject to errors independent of the keys. Hence, we leave this approach for future investigation.

1.8 Conclusion and Open Questions

The results presented here introduced the notion of interactive proofs with quantum provers and this journal version provides rigorous proofs of the two QPIPs presented in [ABOE08]. These results show that the fundamental questions regarding the falsifiability of the high complexity regime of quantum mechanics, the ability to delegate quantum computations to untrusted servers, and the ability to test that experimental quantum systems behave as they should can all be done using interactive protocols between a BQP prover and a classical (BPP) verifier augmented with $O(1)$ qubits.

This work has revolutionary implications in the context of philosophy of science. It suggests that experiments can be conducted in a structured adaptive way, along the lines of interactive proofs [GMR85]; this can be called "interactive experiments" and suggests a new approach to confirmation of physical theories. Following discussions with us at preliminary stages of this work, Jonathan Yaari has studied "Interactive proofs with Nature" from the philosophy of science perspective [Yaa08]. The philosophical aspects of this possibility of interactive experiments suggested by our QPIP protocols were also discussed by Aharonov and Vazirani in [AV12]. A very interesting question is whether interactive experiments can be designed to test conjectured physical theories, even in the absence of full control of the physical system as is required in our protocols. A particularly interesting example is high T_c superconductivity, in which guesses regarding the governing Hamiltonian exist. It would be extremely interesting to be able to test the correctness of the Hamiltonian using such interactive techniques, without resorting to full fledged quantum computational power.

Perhaps the most important and intriguing open question that emerges from this work is whether it is possible to remove the necessity for even a small quantum register, and achieve similar results in the more natural QPIP model in which the verifier is entirely classical. This would have interesting fundamental implications regarding the ability of a classical system to learn and test a quantum system; it is likely that such a protocol might also have implications on the major open problem of quantum PCP [AAV13].

Finally, we can also ask whether it is possible to achieve blind (rather than verifiable) computation, in two different settings. The question of blind computation involves a client who would like to ask a BQP server to run a BQP circuit. The client does not wish to verify the result of the computation, but just to ensure that the server does

not learn anything about the computation, even though he is able to run the computation. If the client is a BQP machine (but does some amount of work which is independent of the size of the computation) and there is only one round of interaction, this problem is known as quantum fully homomorphic encryption. While there have been several results exploring this question, such as [DSS16], [BJ14] and [YPDF14] (which is an impossibility result regarding information theoretically secure quantum homomorphic encryption), quantum fully homomorphic encryption remains an open question. We can also change the model slightly by allowing classical interaction and restricting the client to be a BPP machine. This variant also remains open.

We remark that this area is notorious for the difficulty in providing rigorous protocols and proofs of security, as the arguments involved are very delicate and subtle. We hope that this journal version makes a useful contribution in this direction. We believe that the techniques presented here will be very useful in the vastly growing area of delegated quantum computation and quantum cryptographic protocols.

Paper Organization We start with some notations and background in Sec. 2. In Section 3, we present the Clifford QAS and prove its security. In Section 4, we present the Clifford QPIP and prove security. Sections 5 and 6 present the polynomial QAS and QPIP. Blind delegated quantum computation is proved in Section 7. The corollaries related to the interpretations of the results are proven in Section 8. Appendix A contains the definition of $\mathsf{QPIP}_{\kappa}^{sym}$ as well as the proofs of Corollary 1.3 and Corollary 1.13. Appendix B contains useful lemmas about Clifford and Pauli operators, Appendix C contains proofs of the technical lemmas required in Sections 3 and 4 and Appendix D contains proofs of correctness of the logical operators on signed polynomial codes. Finally, in Appendix E we provide a notation table; this is especially helpful in reading Section 6, as we introduce a significant amount of notation in that section.

2 Background

2.1 Quantum Authentication

Quantum authentication is a protocol by which a sender $\mathcal{A}$ and a receiver $\mathcal{B}$ are capable of verifying that the state sent by $\mathcal{A}$ had not been altered while transmitted to $\mathcal{B}$.

2.1.1 Quantum Security

If $\mathcal{B}$ is a quantum machine, we would like our authentication definition to capture the following two requirements. On the one hand, in the absence of intervention, the received state should be the same as the sent state and moreover, $\mathcal{B}$ should not abort. On the other hand, we want that when the adversary does intervene, then with all but a small probability (or in fact, distance in terms of density matrices), either $\mathcal{B}$ rejects or his received state is the same as that sent by $\mathcal{A}$.

This is formalized below for pure states; one can deduce the appropriate statement about fidelity of mixed states, or for states that are entangled to the rest of the world (see [BCG$^+$02] Appendix B).

Definition 2.1 *(adapted from Barnum et. al. [BCG$^+$02]). A quantum authentication scheme (QAS) from l to $m = l + e$ qubits, with security ϵ, is a pair of polynomial time quantum algorithms $\mathcal{A}$ and $\mathcal{B}$ together with a set of classical keys $\mathcal{K}$ such that:*

- *$\mathcal{A}$ takes as input a state $|\psi\rangle$ on l qubits and chooses $k \in \mathcal{K}$ uniformly at random. $\mathcal{A}$ then applies a unitary operator A_k on the state of m qubits $|\psi\rangle |0\rangle^{\otimes e}$ obtaining:*

$$A_k(|\psi\rangle |0\rangle^{\otimes e}) \tag{1}$$

- *$\mathcal{B}$ takes as input a state of m qubits and a classical key $k \in \mathcal{K}$. He applies a unitary operator B_k to the input state to obtain an output state of m qubits. $\mathcal{B}$ declares the state valid if the last e qubits of the output state lie in the space $|0\rangle\langle 0|^{\otimes e}$ and declares the state erroneous if the last e qubits lie in the space $\Pi_{ABR} = \mathcal{I} - |0\rangle\langle 0|^{\otimes e}$.*

We require:

- **Completeness***: For all keys $k \in \mathcal{K}$,*

$$B_k A_k(|\psi\rangle\langle\psi| \otimes |0\rangle\langle 0|^{\otimes e})A_k^\dagger B_k^\dagger = |\psi\rangle\langle\psi| \otimes |0\rangle\langle 0|^{\otimes e} \tag{2}$$

To quantify soundness, define the projections:

$$\Pi_1^{|\psi\rangle} = |\psi\rangle\langle\psi| \otimes I^{\otimes e} + \left(I^{\otimes l} - |\psi\rangle\langle\psi|\right) \otimes \Pi_{ABR} \tag{3}$$

$$\Pi_0^{|\psi\rangle} = (I^{\otimes l} - |\psi\rangle\langle\psi|) \otimes |0\rangle\langle 0|^{\otimes e} \tag{4}$$

Then

- **Soundness***: For any super-operator $\mathcal{O}$ (representing a possible intervention by the adversary), let ρ_B be defined by*

$$\rho_B = \frac{1}{|\mathcal{K}|}\sum_k B_k(\mathcal{O}(A_k(|\psi\rangle\langle\psi| \otimes |0\rangle\langle 0|^{\otimes e})A_k^\dagger))B_k^\dagger \tag{5}$$

Then the quantum authentication scheme is ϵ-secure if:

$$Tr(\Pi_1^{|\psi\rangle}\rho_B) \geq 1 - \epsilon \tag{6}$$

2.2 Pauli and Clifford Gates in F_2

The n-qubits Pauli group consists of all elements of the form $P = P_1 \otimes P_2 \otimes \ldots \otimes P_n$ where $P_i \in \{\mathcal{I}, X, Y, Z\}$, together with the multiplicative factors -1 and $\pm i$. We will use a subset of this group, which we denote as $\mathbb{P}_n$, which includes all operators $P = P_1 \otimes P_2 \otimes \ldots \otimes P_n$ but not the multiplicative factors.

The Pauli group $\mathbb{P}_n$ is a basis to the matrices acting on n-qubits. We can write any matrix U over a vector space $A \otimes B$ (where A is the space of n qubits) as $\sum_{P\in\mathbb{P}_n} P \otimes U_P$ where U_P is some (not necessarily unitary) matrix on B.

Let $\mathfrak{C}_n$ denote the n-qubit Clifford group. Recall that it is a finite subgroup of unitaries acting on n qubits generated by the Hadamard matrix-H, by $K = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$, and by controlled-NOT. The Clifford group is characterized by the property that it maps the Pauli group $\mathbb{P}_n$ to itself, up to a phase $\alpha \in \{\pm 1, \pm i\}$. That is: $\forall C \in \mathfrak{C}_n, P \in \mathbb{P}_n : \alpha CPC^\dagger \in \mathbb{P}_n$

Fact 2.1 *[DLT02] A random element from the Clifford group on n qubits can be sampled efficiently by choosing a string k of $poly(n)$ bits uniformly at random. The map from k to the group element represented as a product of Clifford group generators can be computed in classical polynomial time.*

2.3 Generalized Gates over F_q

Definition 2.2 *The generalized Pauli operators over F_q perform the following maps:*

$$X|a\rangle = |(a+1) \mod q\rangle \quad (7)$$
$$Z|a\rangle = \omega_q^a |a\rangle \quad (8)$$
$$Y = XZ \quad (9)$$

where $\omega_q = e^{2\pi i/q}$ is the primitive q-root of the unity.

We note that $ZX = \omega_q XZ$. The generalized Pauli group consists of generalized Pauli operators, together with the multiplicative factor ω_q. We use the same notation, $\mathbb{P}_n$, for the standard and generalized Pauli groups, as it will be clear by context which one is being used.

Definition 2.3 *For vectors x, z in F_q^m, we denote by $P_{x,z}$ the Pauli operator $Z^{z_1}X^{x_1} \otimes \ldots \otimes Z^{z_m}X^{x_m}$.*

We now define the other generalized gates we will need:

Definition 2.4 ***Generalized Gates***

1. *The generalized Fourier transform over F_q performs the following map on $a \in F_q$:*

$$F|a\rangle \stackrel{\text{def}}{=} \frac{1}{\sqrt{q}} \sum_b \omega_q^{ab} |b\rangle \quad (10)$$

2. *The generalized r- variant of the Fourier transform over F_q performs the following map on $a \in F_q$:*

$$F_r|a\rangle \stackrel{\text{def}}{=} \frac{1}{\sqrt{q}} \sum_b \omega_q^{rab} |b\rangle \quad (11)$$

3. *The generalized CNOT gate, which we denote as SUM, performs the following map on $a, b \in F_q$:*

$$SUM|a\rangle|b\rangle \stackrel{\text{def}}{=} |a\rangle|(a+b) \mod q\rangle \quad (12)$$

4. *The generalized Toffoli gate T performs the following map on $a, b, c \in F_q$:*

$$T|a\rangle|b\rangle|c\rangle \stackrel{\text{def}}{=} |a\rangle|b\rangle|c+ab\rangle \quad (13)$$

5. *The multiplication gate M_r (for $r \in F_q$, $r \neq 0$) performs the following map on $a \in F_q$:*

$$M_r|a\rangle \stackrel{\text{def}}{=} |ra\rangle \quad (14)$$

6. *The generalized controlled phase gate, which we denote as CPG, performs the following map on $a, b \in F_q$:*

$$CPG|a\rangle|b\rangle = \omega_q^{ab}|a\rangle|b\rangle \quad (15)$$

2.3.1 Toffoli Gate by Teleportation

If given a resource state (which we will also refer to as a magic state or a Toffoli state) of the following form:

$$\frac{1}{q}\sum_{a,b\in F_q}|a,b,ab\rangle \tag{16}$$

it is possible to apply a Toffoli gate using only Clifford operations and measurements. This can be done as follows. Assume we would like to apply the Toffoli gate to the state $|c,d,e\rangle$, resulting in $|c,d,e+cd\rangle$. We start with the following state:

$$\frac{1}{q}\sum_{a,b\in F_q}|a,b,ab,c,d,e\rangle$$

We then perform the following Clifford entangling operations: a SUM gate from register 6 to register 3, inverse sum gates from register 1 to 4 and register 2 to 5, and an inverse Fourier gate on register 6 resulting in:

$$\frac{1}{q}\sum_{a,b\in F_q}|a,b,ab\rangle \longrightarrow \frac{1}{\sqrt{q^3}}\sum_{a,b,l\in F_q}\omega^{-le}|a,b,ab+e,c-a,d-b,l\rangle \tag{17}$$

We then measure registers 4,5, and 6 obtaining measurement results x,y,z where x corresponds to register 4, etc.. The renormalized state after measurement on the unmeasured registers (the first three registers) is then:

$$\omega^{-ze}|c-x,d-y,(c-x)(d-y)+e\rangle \tag{18}$$

Then we apply the following correction to the state (on the first three remaining registers):

$$C_{x,y,z}\stackrel{\text{def}}{=}T(X^x\otimes X^y\otimes Z^z)T^\dagger=(X^xZ^{-yz}\otimes X^yZ^{-xz}\otimes X^{xy}Z^z)SUM^y_{1,3}SUM^x_{2,3}CPG^{-z}_{1,2} \tag{19}$$

where the subscript denotes the registers (the first is the control and second is the target). We note that the above correction involves Toffoli gates, but since they are acting by conjugation on Pauli operators, the expression is actually a Clifford operator. It is easy to check that after applying $C_{x,y,z}$ to the state in equation 18, the resulting state is:

$$|c,d,e+cd\rangle \tag{20}$$

2.4 Conjugation Properties of Generalized Gates

In this section, we describe how the gates above conjugate operators in the Pauli group. We begin with the *SUM* gate. It is easy to check that:

$$SUM(Z^{z_A}X^{x_A}\otimes Z^{z_B}X^{x_B})SUM^\dagger \quad=\quad (Z^{z_A-z_B}X^{x_A}\otimes Z^{z_B}X^{x_B+x_A}) \tag{21}$$

Next, the Fourier gate swaps the roles of the X and Z Pauli operators; i.e. for $r\in F_q$ ($r\neq 0$)

$$F_rZ^zX^xF_r^\dagger=X^{-r^{-1}z}Z^{rx} \tag{22}$$

Finally, the multiplication gate M_r (again for $r\in F_q$ where $r\neq 0$) has the following conjugation behavior:

$$M_rZ^zX^xM_r^\dagger=Z^{r^{-1}z}X^{rx} \tag{23}$$

2.5 Signed Polynomial Codes

We first define polynomial codes:

Definition 2.5 *Polynomial error correction code [ABO97]. Given m, d, q and $\{\alpha_1, \ldots, \alpha_m\}$ where α_i are distinct non zero values from F_q, the encoding of $a \in F_q$ is $|S_a\rangle$*

$$|S_a\rangle \stackrel{\text{def}}{=} \frac{1}{\sqrt{q^d}} \sum_{f:def(f)\leq d, f(0)=a} |f(\alpha_1), \ldots, f(\alpha_m)\rangle \tag{24}$$

We use here $m = 2d + 1$, in which case the code subspace is its own dual. It is easy to see that this code can detect up to d errors [ABO97]. In this paper, we will be using signed polynomial codes:

Definition 2.6 ([BOCG$^+$06]) *The signed polynomial code with respect to a string $k \in \{\pm 1\}^m$ is defined by:*

$$\left|S_a^k\right\rangle \stackrel{\text{def}}{=} \frac{1}{\sqrt{q^d}} \sum_{f:deg(f)\leq d, f(0)=a} |k_1 \cdot f(\alpha_1) \ldots k_m \cdot f(\alpha_m)\rangle$$

We again use $m = 2d + 1$. Similar to the polynomial code, the signed polynomial code can detect d errors and is self dual [BOCG$^+$06]. We will require the following encoding circuit:

Definition 2.7 *Let E_k be a unitary operator such that:*

$$E_k |a\rangle |0\rangle^{\otimes m-1} = \left|S_a^k\right\rangle$$

E_k is the encoding circuit, which we describe in further detail in Section 2.5.2. We will write ρ^k to denote that a density matrix ρ is encoded with the signed polynomial code with respect to k; i.e. if ρ is one qudit, then

$$\rho^k \stackrel{\text{def}}{=} E_k(\rho \otimes |0\rangle \langle 0|^{\otimes m-1}) E_k^\dagger \tag{25}$$

2.5.1 Signed Polynomial Code Logical Gates

For proofs of all claims and lemmas below, see Appendix D. We first provide the logical X operator:

Claim 2.1 *For $x \in F_q$ and $k \in \{-1, 1\}^m$, the logical X operator $\widetilde{X}_k^x$ obeys the following identity:*

$$\widetilde{X}_k^x \left|S_a^k\right\rangle \stackrel{\text{def}}{=} (X^{k_1 x} \otimes \cdots \otimes X^{k_m x}) \left|S_a^k\right\rangle = \left|S_{a+x}^k\right\rangle \tag{26}$$

Similarly for logical *SUM*, we consider the transitive application of controlled-sum, that is a *SUM* operation applied between the j'th register of $|S_a\rangle$ and $|S_b\rangle$.

Claim 2.2 *For all $k \in \{-1, 1\}^m$, the logical SUM operator $\widetilde{SUM}$ obeys the following identity:*

$$\widetilde{SUM} \left|S_a^k\right\rangle \left|S_b^k\right\rangle \stackrel{\text{def}}{=} (SUM)^{\otimes m} \left|S_a^k\right\rangle \left|S_b^k\right\rangle = \left|S_a^k\right\rangle \left|S_{a+b}^k\right\rangle \tag{27}$$

where each SUM gate in the tensor product acts between registers i and $m + i$ for $1 \leq i \leq m$.

Showing what is the logical Fourier transform on the signed polynomial code requires more work. We need the following lemma:

Lemma 2.3 *For any m distinct numbers $\{\alpha_i\}_1^m$ there exist* ***interpolation coefficients*** *$\{c_i\}_1^m$ such that*

$$\sum_{i=1}^{m} c_i f(\alpha_i) = f(0) \tag{28}$$

for any polynomial of degree $\leq m-1$.

We are now ready to define the logical Fourier transform.

Claim 2.4 *For all $k \in \{-1,1\}^m$, the logical Fourier operator $\widetilde{F}$ obeys the following identity:*

$$\widetilde{F}\left|S_a^k\right\rangle \stackrel{\text{def}}{=} F_{c_1} \otimes F_{c_2} \otimes \ldots \otimes F_{c_m} \left|S_a^k\right\rangle = \frac{1}{\sqrt{q}} \sum_b \omega_q^{ab} \left|\widetilde{S_b^k}\right\rangle \tag{29}$$

where $\left|\widetilde{S_b^k}\right\rangle$ is the encoding of b in a signed polynomial code of degree $m-d$ on m registers.

Finally, we define the logical Z operator.

Claim 2.5 *For $z \in F_q$ and $k \in \{-1,1\}^m$, the logical Pauli Z operator $\widetilde{Z}_k^z$ obeys the following identity:*

$$\widetilde{Z}_k^z \left|S_a^k\right\rangle \stackrel{\text{def}}{=} (Z^{k_1 c_1 z} \otimes \cdots \otimes Z^{k_m c_m z}) \left|S_a^k\right\rangle = \omega_q^{za} \left|S_a^k\right\rangle \tag{30}$$

2.5.2 Signed Polynomial Encoding Circuit

The encoding circuit E_k first applies a Fourier transform on the first d 0 states, and then interpolates to fill in the rest of the state. The following unitary operator performs the interpolation:

Definition 2.8 *Let D_k be a unitary operator such that*

$$D_k \left|a\right\rangle \left|k_2 f(\alpha_2), \ldots, k_{d+1} f(\alpha_{d+1})\right\rangle \left|0\right\rangle^{\otimes d} = \left|k_1 f(\alpha_1), \ldots, k_m f(\alpha_m)\right\rangle$$

such that $deg(f) \leq d$ and $f(0) = a$.

Now if F is the generalized Fourier transform (see equation 10) it is easy to check that

$$E_k = D_k(\mathcal{I} \otimes F^{\otimes d} \otimes \mathcal{I}) \tag{31}$$

We now describe D_k in further detail.

Claim 2.6 *The operator D_k can be written as a product of SUM operators controlled by registers $1, \ldots, d+1$ with target registers $1, d+2, \ldots, m$ and a multiplication operator on the first register. More explicitly:*

$$D_k = \prod_i SUM_{i,1}^{h_i(\alpha_1) k_i k_1} (M_{k_1 h_0(\alpha_1)} \otimes \mathcal{I}^{\otimes m-1}) \prod_{i,j} SUM_{i,j}^{h_i(\alpha_j) k_i k_j} \prod_j SUM_{1,j}^{h_0(\alpha_j) k_j} \tag{32}$$

where $2 \leq i \leq d+1$, $d+2 \leq j \leq m$ and for $i' \in \{0, 2, ..., d+1\}$ and $\alpha_0 = 0$

$$h_{i'}(x) = \prod_{\substack{l \in \{0,2,\ldots,d+1\} \\ l \neq i'}} \frac{x - \alpha_l}{\alpha_{i'} - \alpha_l} \tag{33}$$

Proof: Observe that

$$h_0(x)f(0) + \sum_{2\leq i\leq d+1} h_i(x)f(\alpha_i) = f(x) \tag{34}$$

It follows that for all $d+2 \leq j \leq m$, register j holds the following value after the controlled sum operations detailed above:

$$\begin{aligned} k_0 f(0)(h_0(\alpha_j)k_0k_j) + \sum_{2\leq i\leq d+1} k_i f(\alpha_i)(h_i(\alpha_j)k_ik_j) &= k_j(h_0(\alpha_j)f(0) + \sum_{2\leq i\leq d+1} h_i(\alpha_j)f(\alpha_i)) & (35)\\ &= k_j f(\alpha_j) & (36)\end{aligned}$$

Now we can see that the controlled sum operations have performed the following mapping:

$$\left|f(0), k_2f(\alpha_2), \ldots, k_{d+1}f(\alpha_{d+1}), 0^d\right\rangle \rightarrow |f(0), k_2f(\alpha_2), \ldots, k_mf(\alpha_m)\rangle \tag{37}$$

The only thing left to do is map the first register from $f(0)$ to $k_1f(\alpha_1)$. To do this, first multiply the first register by $k_1h_0(\alpha_1)$ by using the multiplication operation $M_{k_1h_0(\alpha_1)}$, where

$$M_{k_1h_0(\alpha_1)}\,|a\rangle = |k_1h_0(\alpha_1)a\rangle \tag{38}$$

The multiplication operator performs the following mapping:

$$|f(0), k_2f(\alpha_2), \ldots, k_mf(\alpha_m)\rangle \rightarrow |k_1h_0(\alpha_1)f(0), k_2f(\alpha_2), \ldots, k_mf(\alpha_m)\rangle \tag{39}$$

Then apply controlled sum operations from registers $\hat{i} \in \{2, \ldots, d+1\}$ to register 1 $h_{\hat{i}}(\alpha_1)k_ik_1$ times. Due to equation 34, the value in the first register after these operations is:

$$\begin{aligned} k_1h_0(\alpha_1)f(0) + \sum_{2\leq i\leq d+2} k_if(\alpha_i)(h_i(\alpha_1)k_ik_1) &= k_1(h_0(\alpha_1)f(0) + h_i(\alpha_1)f(\alpha_i)) & (40)\\ &= k_1f(\alpha_1) & (41)\end{aligned}$$

It follows that the final controlled sum operations have performed the following mapping:

$$|k_1h_0(\alpha_1)f(0), k_2f(\alpha_2), \ldots, k_mf(\alpha_m)\rangle \rightarrow |k_1f(\alpha_1), k_2f(\alpha_2), \ldots, k_mf(\alpha_m)\rangle \tag{42}$$

□

3 Clifford Authentication Scheme

We now define a quantum authentication scheme based on Clifford operations. Let $\mathcal{K}_m$ be the set of authentication keys, consisting of succinct descriptions of Clifford operations in $\mathfrak{C}_m$ (i.e. Clifford operations on m qubits); these descriptions exist due to Fact 2.1.

Protocol 3.1 ***Clifford based QAS****: Given is a state $|\psi\rangle$ on l qubits. Let $e \in \mathbb{N}$ be such that $2^{-e} = \epsilon$. We denote $m = l + e$. We denote by C_k the operator specified by a key $k \in \mathcal{K}_m$.*

- ***Encoding -*** *A_k: Alice applies C_k on the state $|\psi\rangle \otimes |0\rangle^{\otimes e}$.*
- ***Decoding -*** *B_k: Bob applies $C_k^\dagger$ to the received state. Bob measures the e auxiliary registers and declares the state valid if they are all 0, otherwise Bob aborts.*

Theorem 1.4 *The Clifford scheme given in Protocol 3.1 is a QAS with security $\epsilon = 2^{-e}$.*

3.1 The Overall Proof of Theorem 1.4

The completeness of this protocol is trivial. In the following proof, we show soundness by first showing that *any* attack of Eve can be decomposed into a distribution over Pauli attacks. We then show that averaging over the random Clifford operators maps a Pauli operator to a uniform distribution over Paulis; the effective transformation on the original state is an application of a random Pauli. These two facts are summarized by Claim 3.1. We conclude the proof of Theorem 1.4 by showing that any Pauli attack is detected with high probability.

*P*roof of **Theorem 1.4**: We denote the space of the message sent from Alice to Bob as M. Without loss of generality, we can assume that Eve adds to the message a system E (of arbitrary dimension) and performs a unitary transformation U on the joint system. We note that there is a representation of U as $\sum_{P\in\mathbb{P}_m} P\otimes U_P$, where P acts on the message space M, and U_P is not necessarily unitary and acts on the environment E. This is because the Pauli matrices form a basis for the $2^m\times 2^m$ matrix vector space. We first characterize the effect that Eve's attack has on the unencoded message: $|\psi\rangle\otimes|0\rangle^{\otimes e}$.

Claim 3.1 *Let $\rho = |\psi\rangle\langle\psi|\otimes|0\rangle^{\otimes e}$ be the state of Alice before the application of the Clifford operator. For any attack $U=\sum_{P\in\mathbb{P}_m} P\otimes U_P$ by Eve, Bob's state after decoding is $s\rho+\frac{1-s}{4^m-1}\sum_{P\neq\mathcal{I}} P\rho P^\dagger$, where $s = Tr(U_I\rho_E U_I^\dagger)$.*

We proceed with the proof of the theorem. From the above claim we know what Bob's state after Eve's intervention is and we would like to bound its projection on $P_0^{|\psi\rangle}$ (defined in equation 4):

$$\mathrm{Tr}\Big(\Pi_0^{|\psi\rangle}\big(s\rho+\frac{1-s}{4^m-1}\sum_{Q\in\mathbb{P}_m\setminus\{\mathcal{I}\}} Q\rho Q^\dagger\big)\Big) \;=\; s\mathrm{Tr}(\Pi_0^{|\psi\rangle}\rho)+\frac{1-s}{4^m-1}\sum_{Q\in\mathbb{P}_m\setminus\{\mathcal{I}\}}\mathrm{Tr}(\Pi_0^{|\psi\rangle}Q\rho Q^\dagger) \qquad (43)$$

By definition of $\Pi_0^{|\psi\rangle}$ we see that $\mathrm{Tr}(\Pi_0^{|\psi\rangle}\rho)=0$. On the other hand: $\mathrm{Tr}(\Pi_0^{|\psi\rangle}Q\rho Q^\dagger)\leq 1$ when Q does not flip any auxiliary qubits, and vanishes otherwise. The Pauli operators that do not flip auxiliary qubits can be written as $Q'\otimes Q''$ where $Q'\in\mathbb{P}_l$ and $Q''\in\{\mathcal{I},Z\}^{\otimes e}$. It follows that the number of such operators is exactly 4^l2^e. Omitting the identity $\mathcal{I}_m$ we are left with 4^l2^e-1 operators which are undetected by our scheme. We return to Eq. 43:

$$\ldots \;\leq\; \frac{(1-s)(4^l2^e-1)}{4^m-1} \qquad (44)$$

$$\leq\; \frac{1-s}{2^e} \qquad (45)$$

The security follows from the fact that $1-s\leq 1$, and hence the projection is bounded by $\frac{1}{2^e}$. This concludes the proof. □

We remark that the above proof in fact implies a stronger theorem: interventions that are very close to $\mathcal{I}$ are even more likely to keep the state in the space defined by $\Pi_1^{|\psi\rangle}$.

3.2 Proof of Claim 3.1

Let $U=\sum_{P\in\mathbb{P}_m} P\otimes U_P$ be the operator applied by Eve. We denote $\rho=|\psi\rangle\langle\psi|\otimes|0\rangle\langle 0|^{\otimes e}$ the state of Alice prior to encoding. Let us now write ρ_{Bob}, the state of Bob's system after decoding and before measuring the e auxiliary qubits. For clarity of reading we omit the normalization factor $|\mathfrak{C}_m|$ and denote the Clifford operation applied by Alice (Bob) C ($C^\dagger$):

$$\rho_{Bob} \;=\; \frac{1}{|\mathfrak{C}_m|}\mathrm{Tr}_E\Big(\sum_{C\in\mathfrak{C}_m}(C\otimes\mathcal{I}_E)^\dagger U\Big((C\otimes\mathcal{I}_E)\rho(C\otimes\mathcal{I}_E)^\dagger\otimes\rho_E\Big)U^\dagger(C\otimes\mathcal{I}_E)\Big) \qquad (46)$$

At this point, we require the following lemma which states that a random Clifford conjugating an operator has the effect of decohering (removing the cross terms) of the operator:

Lemma 3.2 (Clifford Decoherence) *Let ρ' be a density matrix on $m' > m$ qubits and let $U = \sum_{P \in \mathbb{P}_m} P \otimes U_P$ be a matrix acting on ρ' by conjugation. Then*

$$\frac{1}{|\mathfrak{C}_m|} \sum_{C \in \mathfrak{C}_m} (C \otimes \mathcal{I})^\dagger U (C \otimes \mathcal{I}) \rho' (C \otimes \mathcal{I})^\dagger U^\dagger (C \otimes \mathcal{I}) = (\mathcal{I} \otimes U_I) \rho' (\mathcal{I} \otimes U_I)^\dagger + \frac{1}{|\mathbb{P}_m| - 1} \sum_{P,Q \in \mathbb{P}_m \setminus \{\mathcal{I}\}} (P \otimes U_Q) \rho' (P \otimes U_Q)^\dagger$$

The proof of the above lemma is in Section 3.3. Applying Lemma 3.2, we have

$$\begin{aligned}
\rho_{Bob} &= \mathrm{Tr}_E[(\mathcal{I} \otimes U_I)\rho \otimes \rho_E (\mathcal{I} \otimes U_I)^\dagger + \frac{1}{|\mathbb{P}_m| - 1} \sum_{P,Q \in \mathbb{P}_m \setminus \{\mathcal{I}\}} (P \otimes U_Q)\rho \otimes \rho_E (P \otimes U_Q)^\dagger] && (47) \\
&= \rho \cdot \mathrm{Tr}(U_I \rho_E U_I^\dagger) + \frac{1}{|\mathbb{P}_m| - 1} \sum_{P,Q \in \mathbb{P}_m \setminus \{\mathcal{I}\}} P \rho P^\dagger \cdot \mathrm{Tr}(U_Q \rho_E U_Q^\dagger) && (48) \\
&= \rho \cdot \mathrm{Tr}(U_I \rho_E U_I^\dagger) + \frac{1}{|\mathbb{P}_m| - 1} \Big(\sum_{P \in \mathbb{P}_m \setminus \{\mathcal{I}\}} P \rho P^\dagger \Big) \cdot \Big(\sum_{Q \in \mathbb{P}_m \setminus \{\mathcal{I}\}} \mathrm{Tr}(U_Q \rho_E U_Q^\dagger) \Big) && (49)
\end{aligned}$$

We now require the following lemma, which allows us to trace out the extra space a prover may use as part of his attack. See Section 3.3 for the proof:

Lemma 3.3 *Let $U = \sum_{P \in \mathbb{P}_m} P \otimes U_P$ be a unitary operator acting on $m' > m$ qubits. For any density matrix τ acting on the last $m' - m$ qubits:*

$$\sum_{P \in \mathbb{P}_m} Tr(U_P \tau U_P^\dagger) = Tr(\tau) = 1 \tag{50}$$

We apply Lemma 3.3 to write Bob's state as:

$$s\rho + \frac{(1-s)}{4^m - 1} \sum_{P \in \mathbb{P}_m \setminus \{\mathcal{I}\}} \left(P \rho P^\dagger \right) \tag{51}$$

for $s = \mathrm{Tr}(U_{\mathcal{I}} \rho_E U_{\mathcal{I}}^\dagger)$, which concludes the proof of Claim 3.1.

3.3 Proofs of Technical Lemmas

In this section we prove Lemma 3.2 and Lemma 3.3.

3.3.1 Proof of Lemma 3.2

To prove Lemma 3.2, we require the following three lemmata, which we prove in Appendix C. The lemma below states that applying a random Clifford operator in $\mathfrak{C}_m$ (by conjugation) to a non identity Pauli operator $P \in \mathbb{P}_m$ maps it to a Pauli operator $Q \in \mathbb{P}_m$ chosen uniformly over all non-identity Pauli operators:

Lemma 3.4 (Pauli Partitioning by Cliffords) *For every $P, Q \in \mathbb{P}_m \setminus \{\mathcal{I}\}$ it holds that : $\left| \{ C \in \mathfrak{C}_m | C^\dagger P C = Q \} \right| = \frac{|\mathfrak{C}_m|}{|\mathbb{P}_m| - 1} = \frac{|\mathfrak{C}_m|}{4^m - 1}$.*

The following two lemmas describe the effect of conjugating an operator by a random Pauli or Clifford operator. The lemma below (Lemma 3.5) is used in the proof of the lemma after it (Lemma 3.6), and since it will be useful also when we handle the polynomial codes based protocol later on, we state it here with generalized Pauli operators:

Lemma 3.5 (Pauli Twirl) *Let $P \neq P'$ be generalized Pauli operators. For any density matrix ρ' on $m' > m$ qubits it holds that*

$$\sum_{Q \in \mathbb{P}_m} (Q^\dagger P Q \otimes \mathcal{I}) \rho' (Q^\dagger (P')^\dagger Q \otimes \mathcal{I}) = 0$$

Lemma 3.6 (Clifford Twirl) *Let $P \neq P'$ be Pauli operators. For any density matrix ρ' on $m' > m$ qubits it holds that*

$$\sum_{C \in \mathfrak{C}_m} (C^\dagger P C \otimes \mathcal{I}) \rho' (C^\dagger (P')^\dagger C \otimes \mathcal{I}) = 0$$

We now proceed to the proof of Lemma 3.2:

*P*roof of **Lemma 3.2**: We start with

$$\frac{1}{|\mathfrak{C}_m|} \sum_{C \in \mathfrak{C}_m} (C \otimes \mathcal{I})^\dagger U (C \otimes \mathcal{I}) \rho' (C \otimes \mathcal{I})^\dagger U^\dagger (C \otimes \mathcal{I}) \;=\; \frac{1}{|\mathfrak{C}_m|} \sum_{P,P' \in \mathbb{P}_m} \sum_{C \in \mathfrak{C}_m} (C^\dagger P C \otimes U_P) \rho' (C^\dagger P' C \otimes U_{P'})^\dagger$$

We use Lemma 3.6 and are left only with $P = P'$

$$\ldots = \frac{1}{|\mathfrak{C}_m|} \sum_{P \in \mathbb{P}_m} \sum_{C \in \mathfrak{C}_m} (C^\dagger P C \otimes U_P) \rho' (C^\dagger P C \otimes U_P)^\dagger \tag{52}$$

We first consider the case were $P = \mathcal{I}$, then:

$$\frac{1}{|\mathfrak{C}_m|} \sum_{C \in \mathfrak{C}_m} (C^\dagger P C \otimes U_P) \rho' (C^\dagger P C \otimes U_P)^\dagger = (\mathcal{I} \otimes U_I) \rho' (\mathcal{I} \otimes U_I)^\dagger \tag{53}$$

On the other hand when, $P \neq \mathcal{I}$ by Lemma 3.4:

$$\frac{1}{|\mathfrak{C}_m|} \sum_{C \in \mathfrak{C}_m} (C^\dagger P C \otimes U_P) \rho' (C^\dagger P C \otimes U_P)^\dagger = \frac{1}{|\mathbb{P}_m| - 1} \sum_{Q \in \mathbb{P} \setminus \{\mathcal{I}\}} (Q \otimes U_P) \rho' (Q \otimes U_P)^\dagger \tag{54}$$

□

3.3.2 Proof of Lemma 3.3

*P*roof of **Lemma 3.3**: We analyze the action of U on the density matrix $\frac{1}{2^m}\mathcal{I} \otimes \tau$. We first notice that U is a trace preserving operator, that is: $\frac{1}{2^m}\text{Tr}(U(\mathcal{I} \otimes \tau)U^\dagger) = \frac{1}{2^m}\text{Tr}(\mathcal{I} \otimes \tau) = \text{Tr}(\tau)$. On the other hand it holds that:

$$\begin{aligned}
\frac{1}{2^m}\text{Tr}\big(U(\mathcal{I} \otimes \tau)U^\dagger\big) &= \frac{1}{2^m} \sum_{P,P' \in \mathbb{P}_m} \text{Tr}\big((P \otimes U_P)(\mathcal{I} \otimes \tau)(P' \otimes U_{P'})^\dagger\big) &(55)\\
&= \frac{1}{2^m} \sum_{P,P' \in \mathbb{P}_m} \text{Tr}\big(P\mathcal{I}P'^\dagger \otimes U_P \tau U_{P'}^\dagger\big) &(56)\\
&= \frac{1}{2^m} \sum_{P,P' \in \mathbb{P}_m} \text{Tr}\big(PP'^\dagger\big)\text{Tr}\big(U_P \tau U_{P'}^\dagger\big) &(57)
\end{aligned}$$

If $P \neq P'$ then $\text{Tr}\big(PP'^\dagger\big) = 0$, and therefore:

$$\begin{aligned}
\ldots &= \frac{1}{2^m} \sum_{P \in \mathbb{P}_m} \text{Tr}\big(\mathcal{I}\big)\text{Tr}\big(U_P \tau U_P^\dagger\big) \\
&= \sum_{P \in \mathbb{P}_m} \text{Tr}\big(U_P \tau U_P^\dagger\big)
\end{aligned} \tag{58}$$

It follows that $\text{Tr}(\tau) = \sum_{P \in \mathbb{P}_m} \text{Tr}(U_P \tau U_P^\dagger)$, which concludes the proof.

□

4 Quantum Interactive Proofs with Clifford Authentication

Protocol 4.1 ***Clifford based Interactive Proof for Q-CIRCUIT****: Fix a security parameter ϵ. Given is a quantum circuit consisting of two-qubit gates, $U = U_N \ldots U_1$, acting on n input qubits with error probability reduced to $\leq \gamma$. The verifier chooses n authentication keys $k_1, \ldots, k_n \in \mathcal{K}_{e+1}$, where $e = \lceil \log \frac{1}{\epsilon} \rceil$. The verifier authenticates the input qubits of the circuit one by one using the Clifford QAS ; that is qubit j is authenticated using operation C_{k_j} on $e+1$ qubits. The verifier sends the authenticated qubits to the prover $\mathbb{P}$. In round i (for $1 \leq i \leq N$), the verifier asks $\mathbb{P}$ to return the qubits on which U_i will act. The verifier decodes these qubits by applying the inverse Clifford operator. The verifier then applies U_i, authenticates the resulting qubits with new authentication keys and the same (unmeasured) auxiliary qubits and sends the qubits to $\mathbb{P}$. In round $N+1$, the prover sends the verifier the first authenticatd qubit, which the verifier decodes and rejects if the auxiliary qubits are not valid. The verifier then measures the decoded qubit (which contains the result of the circuit) and accepts or rejects accordingly. In any case that the verifier does not get the correct number of qubits during the protocol he aborts.*

Theorem 1.5 *For $0 < \epsilon < 1$ and $\gamma < 1 - \epsilon$, Protocol 4.1 is a $\mathsf{QPIP}_{O(\log(\frac{1}{\epsilon}))}$ with completeness $1 - \gamma$ and soundness $\gamma + \epsilon$ for $\mathsf{Q\text{-}CIRCUIT}_\gamma$.*

The quantum communication is linear in the number of gates. For $\epsilon = \frac{1}{2}$, we get $e = 1$, and so the verifier uses a register of 4 qubits (2 per gate). In fact 3 is enough, since each of the authenticated qubits can be decoded (or encoded and sent) on its own before a new authenticated qubit is handled.

4.1 Overall Proof of Theorem 1.5

Let us first analyze completeness:

Claim 4.1 (Completeness) *For any $\gamma > 0$, Protocol 4.1 is a QPIP protocol with completeness $1 - \gamma$ for Q-CIRCUIT.*

Proof: To prove completeness, we assume the prover is honest and we will show that if $x \in L$, the verifier accepts with probability $\geq 1 - \gamma$. Since the prover is honest, the state at all times is indeed the correctly authenticated state of the circuit. Thus, the output qubit is indeed an authentication of a (possibly mixed) state, which if measured, outputs 1 with probability $\geq 1 - \gamma$ (the error in the circuit is $\leq \gamma$). The decoding of the output block received by the verifier will thus result in accept with probability $\geq 1 - \gamma$. $\square$

To prove that Protocol 4.1 has soundness $\gamma + \epsilon$, we will first observe that each round is essentially a run of the Clifford authentication scheme (the only difference is that the auxiliary qubits are not measured and checked by the verifier in each round). Let round i be the round in which the verifier applies U_i. Each round can be seen as an authentication protocol on the 1 or 2 authenticated qubits requested by V; the rest of the qubits are independently authenticated and can therefore be thought of as the extra space of the verifier. Therefore, we can apply the main claim involved in the soundness proof of the authentication scheme (Claim 3.1) in each round to just the qubits requested by V.

We will use the claim below to prove soundness, and then we will prove the claim by induction:

Claim 4.2 (Clifford QPIP State Evolution) *The state held by the prover at the start of round i (before sending qubits for U_i to the verifier) can be written as:*

$$O^i(C_k \rho_i C_k^\dagger \otimes \rho_{E_i})(O^i)^\dagger$$

where $C_k = C_{k_1} \otimes \cdots \otimes C_{k_n}$, O^i is a unitary, $\rho_i = (U_{i-1} \cdots U_1)\rho(U_{i-1} \cdots U_1)^\dagger$ (ρ is the initial density n qubit matrix), and ρ_{E_i} represents the prover's extra space.

This is the claim, which, as mentioned in Section 1.4 of the introduction, uses the strong properties of Clifford decoherence (Lemma 3.2) to show that the attack of the prover can be passed through all rounds to the end of the protocol. Using this claim, we will prove soundness:

Claim 4.3 (Soundness) *For any $\epsilon, \gamma > 0$, Protocol 4.1 is a QPIP protocol with soundness $\gamma + \epsilon$ for Q-CIRCUIT.*

Proof: Assume Claim 4.2 holds. Before the prover sends the verifier the first authenticated qubit in round $N+1$ (the final round), Claim 4.2 implies that his state is:

$$O^{N+1}(C_k \rho_{N+1} C_k^\dagger \otimes \rho_{E_{N+1}})(O^{N+1})^\dagger$$

Now we can average over all of the authentication keys except k_1, since they will not be used for decoding. After averaging, the prover's state can be written as:

$$\frac{1}{|\mathfrak{C}_{e+1}|^{n-1}} \sum_{k_2,\ldots,k_n} O^{N+1}(C_k \rho_{N+1} C_k^\dagger \otimes \rho_{E_{N+1}})(O^{N+1})^\dagger$$

Here we require the following lemma, which states that random Clifford operators acting on a state turn it into a maximally mixed state:

Lemma 4.4 (Clifford Mixing) *For a matrix ρ on spaces $A \otimes B$, where A is the space of n qubits and $n \in \mathbb{N}$*

$$\frac{1}{|\mathfrak{C}_n|} \sum_{C \in \mathfrak{C}_n} (C \otimes \mathcal{I}_B)\rho(C \otimes \mathcal{I}_B)^\dagger = \frac{1}{2^n} \mathcal{I}_A \otimes Tr_A(\rho)$$

The proof of this lemma follows from a similar lemma which uses Pauli operators instead of Clifford operators:

Lemma 4.5 (Pauli Mixing) *For a matrix ρ on two spaces A, B*

$$\frac{1}{\mathbb{P}_n} \sum_{P \in \mathbb{P}_n} (P \otimes \mathcal{I}_B)\rho(P \otimes \mathcal{I}_B)^\dagger = \frac{1}{q^n} \mathcal{I}_A \otimes Tr_A(\rho)$$

The proofs of both Lemma 4.4 and Lemma 4.5 can be found in Appendix B. By applying Lemma 4.4,we can see that the averaging changes the state in the final round to:

$$O^{N+1}(C_{k_1} \mathrm{Tr}_A(\rho_{N+1}) C_{k_1}^\dagger \otimes \rho_{AE_{N+1}})(O^{N+1})^\dagger \tag{59}$$

where

$$\rho_{AE_{N+1}} = \frac{1}{2^{(n-1)(e+1)}} \mathcal{I}_A \otimes \rho_{E_{N+1}} \tag{60}$$

and A represents the space of all computational qubits other than the first. Now we proceed to write down the state after the verifier's decoding. Namely, the verifier will decode by applying $C_{k_1}^\dagger$, and then we can average over k_1, obtaining:

$$\frac{1}{|\mathfrak{C}_{e+1}|} \sum_{k_1 \in \mathcal{K}_{e+1}} (C_{k_1}^\dagger \otimes \mathcal{I}_{AE_{N+1}}) O^{N+1}(C_{k_1} \mathrm{Tr}_A(\rho_{N+1}) C_{k_1}^\dagger \otimes \rho_{AE_{N+1}})(O^{N+1})^\dagger (C_{k_1} \otimes \mathcal{I}_{AE_{N+1}}) \tag{61}$$

Now we can directly apply Claim 3.1 to obtain the state after decoding. This is done by replacing ρ in the statement of Claim 3.1 with $\mathrm{Tr}_A(\rho_{N+1})$ and replacing ρ_E in the statement with $\rho_{AE_{N+1}}$. We would like to bound the projection of the state on $\Pi_0^{|1\rangle}$:

$$\mathrm{Tr}\Big(\Pi_0^{|1\rangle}\big(s\mathrm{Tr}_A(\rho_{N+1}) + \frac{1-s}{4^m - 1} \sum_{Q \in \mathbb{P}_m \setminus \{\mathcal{I}\}} Q(\mathrm{Tr}_A(\rho_{N+1}))Q^\dagger\big)\Big) \tag{62}$$

$$= s\text{Tr}(\Pi_0^{|1\rangle}\text{Tr}_A(\rho_{N+1})) + \frac{1-s}{4^m-1}\sum_{Q\in\mathbb{P}_m\setminus\{\mathcal{I}\}}\text{Tr}(\Pi_0^{|1\rangle}Q(\text{Tr}_A(\rho_{N+1}))Q^\dagger) \tag{63}$$

Due to the circuit error γ, $\text{Tr}(\Pi_0^{|1\rangle}\rho_{N+1}) \leq \gamma$. The rest of the upper bound is exactly as in the proof of Theorem 1.4 (see the explanation linking equations 43 and 44), and the final upper bound is:

$$s\text{Tr}(\Pi_0^{|1\rangle}\rho_{N+1}) + \frac{1-s}{4^m-1}\sum_{Q\in\mathbb{P}_m\setminus\{\mathcal{I}\}}\text{Tr}(\Pi_0^{|1\rangle}Q\rho_{N+1}Q^\dagger) \leq s\gamma + \frac{1-s}{2^e} \tag{64}$$

$$\leq \gamma + \frac{1}{2^e} \tag{65}$$

□

4.2 Proof of Claim 4.2

We now proceed to the proof of the claim. We will require the following lemma, which we will prove after completing the current proof:

Lemma 4.6 (Unitary Commutation) *For all unitaries U acting on A, a space of k qubits (for $k \in \mathbb{N}$), and density matrices ρ acting on $A \otimes B$, we have:*

$$\sum_{Q\neq\mathcal{I}\in\mathbb{P}_k}(UQ\otimes\mathcal{I})\rho(UQ\otimes\mathcal{I})^\dagger = \sum_{Q\neq\mathcal{I}\in\mathbb{P}_k}(QU\otimes\mathcal{I})\rho(QU\otimes\mathcal{I})^\dagger$$

We prove Claim 4.2 by induction. The base case is clear. For the inductive step, we assume the claim holds for round i and show that it holds for round $i+1$. When the verifier requests the qubits needed for U_i, the prover sends back register $\mathcal{R}_i$, which contains the authenticated qudits required to apply U_i. Assume $\mathcal{R}_i$ is the first register of the state written below and that it contains 2 authenticated qudits. Then the prover sends back $\mathcal{R}_i$ from the state as given in the inductive step:

$$O^i(C_k\rho_i C_k^\dagger \otimes \rho_{E_i})(O^i)^\dagger \tag{66}$$

We now write the state after the verifier decodes register $\mathcal{R}_i$ (and after averaging over the Clifford keys for register $\mathcal{R}_i$):

$$\frac{1}{|\mathfrak{C}_{e+1}|}\sum_{k_1,k_2\in\mathcal{K}_{e+1}}(C_{k_1}\otimes C_{k_2}\otimes\mathcal{I})^\dagger O^i(C_k\rho_i C_k^\dagger\otimes\rho_{E_i})(O^i)^\dagger(C_{k_1}\otimes C_{k_2}\otimes\mathcal{I}) \tag{67}$$

Next, we can decompose O^i as $\sum_{P\in\mathbb{P}_{|\mathcal{R}_i|}} P\otimes O_P^i$, where P acts on register $\mathcal{R}_i$ and O_P^i acts on all other qubits (i.e. the remaining computational qubits as well as the extra space of the prover). Applying Lemma 3.2, we can write the state as:

$$(\mathcal{I}\otimes O_{\mathcal{I}}^i)C_{k'}(\rho_i\otimes\rho_{E_i})C_{k'}^\dagger(\mathcal{I}\otimes O_{\mathcal{I}}^i)^\dagger + \frac{1}{|\mathbb{P}_{|\mathcal{R}_i|}|-1}\sum_{P,Q\neq I\in\mathbb{P}_{|\mathcal{R}_i|}}(Q\otimes O_P^i)C_{k'}(\rho_i\otimes\rho_{E_i})C_{k'}^\dagger(Q\otimes O_P^i)^\dagger \tag{68}$$

where $C_{k'} = \mathcal{I}\otimes C_{k_3}\otimes\cdots\otimes C_{k_n}\otimes\mathcal{I}_E$. The verifier then applies the gate U_i and authenticates register $\mathcal{R}_i$ with fresh keys $C_{\hat{k}_1}\otimes C_{\hat{k}_2}$. Let $U_i' = (C_{\hat{k}_1}\otimes C_{\hat{k}_2})U_i$; this is the operation applied to the decoded state (given in equation 68) by the verifier. First observe the action of this operation on the first term of the decoded state; it is now:

$$(\mathcal{I}\otimes O_{\mathcal{I}}^i)C_{\hat{k}}(\rho_{i+1}\otimes\rho_{E_i})C_{\hat{k}}^\dagger(\mathcal{I}\otimes O_{\mathcal{I}}^i)^\dagger \tag{69}$$

where $C_{\hat{k}} = (C_{\hat{k}_1} \otimes C_{\hat{k}_2} \otimes \mathcal{I}) C_{k'}$. To determine what happens to the second term of the decoded state in equation 68 after the verifier applies U_i', we will apply the unitary commutation lemma, Lemma 4.6. We first write the second term (after application of U_i') in a way which makes it easier to apply Lemma 4.6:

$$\frac{1}{|\mathbb{P}_{|\mathcal{R}_i|}| - 1} \sum_{P,Q \neq I \in \mathbb{P}_{|\mathcal{R}_i|}} (\mathcal{I} \otimes O_P^i)(U_i' Q \otimes \mathcal{I}) C_{k'} (\rho_i \otimes \rho_{E_i}) C_{k'}^\dagger (U_i' Q \otimes \mathcal{I})^\dagger (\mathcal{I} \otimes O_P^i)^\dagger \tag{70}$$

Now we apply Lemma 4.6, obtaining that the above expression is equal to:

$$\frac{1}{|\mathbb{P}_{|\mathcal{R}_i|}| - 1} \sum_{P,Q \neq I \in \mathbb{P}_{|\mathcal{R}_i|}} (\mathcal{I} \otimes O_P^i)(Q U_i' \otimes \mathcal{I}) C_{k'} (\rho_i \otimes \rho_{E_i}) C_{k'}^\dagger (Q U_i' \otimes \mathcal{I})^\dagger (\mathcal{I} \otimes O_P^i)^\dagger \tag{71}$$

$$= \frac{1}{|\mathbb{P}_{|\mathcal{R}_i|}| - 1} \sum_{P,Q \neq I \in \mathbb{P}_{|\mathcal{R}_i|}} (Q \otimes O_P^i)(U_i' \otimes \mathcal{I}) C_{k'} (\rho_i \otimes \rho_{E_i}) C_{k'}^\dagger (U_i' \otimes \mathcal{I})^\dagger (Q \otimes O_P^i)^\dagger \tag{72}$$

$$= \frac{1}{|\mathbb{P}_{|\mathcal{R}_i|}| - 1} \sum_{P,Q \neq I \in \mathbb{P}_{|\mathcal{R}_i|}} (Q \otimes O_P^i) C_{\hat{k}} (\rho_{i+1} \otimes \rho_{E_i}) C_{\hat{k}}^\dagger (Q \otimes O_P^i)^\dagger \tag{73}$$

It follows that the entire state is:

$$(\mathcal{I} \otimes O_{\mathcal{I}}^i) C_{\hat{k}} (\rho_{i+1} \otimes \rho_{E_i}) C_{\hat{k}}^\dagger (\mathcal{I} \otimes O_{\mathcal{I}}^i)^\dagger + \frac{1}{|\mathbb{P}_{|\mathcal{R}_i|}| - 1} \sum_{P,Q \neq I \in \mathbb{P}_{|\mathcal{R}_i|}} (Q \otimes O_P^i) C_{\hat{k}} (\rho_{i+1} \otimes \rho_{E_i}) C_{\hat{k}}^\dagger (Q \otimes O_P^i)^\dagger \tag{74}$$

We require one last observation: as it stands, the above state consists of a superoperator acting on $C_{\hat{k}}(\rho_{i+1} \otimes \rho_{E_i}) C_{\hat{k}}^\dagger$. To see that the above operation is a superoperator, note that it was obtained by conjugating a unitary by Clifford operators, and then averaging over the Clifford operators. By expanding the extra space from ρ_{E_i} to $\rho_{E_{i+1}}$, we can instead assume we have a unitary O^{i+1} acting on $C_{\hat{k}}(\rho_{i+1} \otimes \rho_{E_{i+1}}) C_{\hat{k}}^\dagger$.

4.2.1 Proof of Lemma 4.6 (Unitary Commutation)

We obtain the following equality from Lemma 4.5:

$$\frac{1}{|\mathbb{P}_k|} \sum_{Q \neq \mathcal{I} \in \mathbb{P}_k} (Q \otimes \mathcal{I}) \rho (Q \otimes \mathcal{I})^\dagger = \frac{1}{2^k} \cdot \mathcal{I} \otimes \mathrm{Tr}_A(\rho) - \frac{1}{|\mathbb{P}_k|} \rho$$

where ρ is a matrix on spaces A, B and Q acts on A. We have:

$$\frac{1}{|\mathbb{P}_k|} \sum_{Q \neq \mathcal{I} \in \mathbb{P}_k} (UQ \otimes \mathcal{I}) \rho (UQ \otimes \mathcal{I})^\dagger = (U \otimes \mathcal{I})(\frac{1}{2^k} \cdot \mathcal{I} \otimes \mathrm{Tr}_A(\rho) - \frac{1}{|\mathbb{P}_k|} \rho)(U \otimes \mathcal{I})^\dagger \tag{75}$$

$$= \frac{1}{2^k} \cdot \mathcal{I} \otimes \mathrm{Tr}_A(\rho) - \frac{1}{|\mathbb{P}_k|} (U \otimes \mathcal{I}) \rho (U \otimes \mathcal{I})^\dagger \tag{76}$$

$$= \frac{1}{2^k} \cdot \mathcal{I} \otimes \mathrm{Tr}_A((U \otimes \mathcal{I}) \rho (U \otimes \mathcal{I})^\dagger) - \frac{1}{|\mathbb{P}_k|} (U \otimes \mathcal{I}) \rho (U \otimes \mathcal{I})^\dagger \tag{77}$$

$$= \frac{1}{|\mathbb{P}_k|} \sum_{Q \neq \mathcal{I} \in \mathbb{P}_k} (QU \otimes \mathcal{I}) \rho (QU \otimes \mathcal{I})^\dagger \tag{78}$$

This concludes the proof.

5 Signed Polynomial Code Authentication Scheme

Let $\mathcal{K}_m$ be the set of pairs of Pauli and sign operators which will be used for authentication; i.e. $\mathcal{K}_m = \{(k,z,x)|k \in \{-1,1\}^m, x,z \in F_q^m\}$.

Protocol 5.1 ***Polynomial Authentication protocol*** *: Alice wishes to send the state $|\psi\rangle$ of dimension q. She chooses a security parameter d, a code length $m = 2d+1$, and selects $k' = (k,z,x) \in \mathcal{K}_m$ at random.*

- ***Encoding -*** *$A_{k'}$: Alice applies E_k to $|\psi\rangle \otimes |0\rangle^{\otimes m-1}$ to encode $|\psi\rangle$ using the signed quantum polynomial code of polynomial degree d (see Definition 2.7). She then applies the Pauli $Z^z X^x$ defined by $x,z \in F_q^m$ (i.e., for $j \in \{1,..,m\}$ she applies $Z^{z_j}X^{x_j}$ on the j'th qubit).*
- ***Decoding -*** *$B_{k'}$: Bob applies the inverse of $A_{k'}$; he applies $(Z^z X^x)^\dagger$ followed by $E_k^\dagger$. Bob measures the $m-1$ auxiliary registers and declares the state valid if they are all 0, otherwise Bob aborts.*

Theorem 1.6 *The polynomial authentication scheme as described in Protocol 5.1 is a QAS with security $\epsilon = 2^{-d}$.*

5.1 The Overall Proof of Theorem 1.6

The completeness of this protocol is trivial. We proceed to prove the security of the protocol. As in the proof of Theorem 1.4, we first show that any intervention made by the adversary can be broken down into a distribution over generalized Pauli interventions. This is given by the Pauli decoherence lemma, Lemma 5.1, which is the weaker analogue of the Clifford decoherence lemma (Lemma 3.2). We then state and prove the sign key security lemma (Lemma 5.2), which states that non identity Pauli interventions by the adversary on states authenticated with the signed polynomial code are detected with high probability. We note that the main difference between the polynomial and Clifford QAS proofs is the ease in which we can prove the security of each QAS against Eve's non trivial Pauli interventions. In the Clifford case, this is easy because a random Pauli will change the auxiliary 0 states with high probability (see the explanation between equation 43 and equation 44 in the proof of Theorem 1.4). In the polynomial case, the sign key security lemma (Lemma 5.2) requires quite a few technical details.

*P*roof of **Theorem 1.6**: We denote $\rho = |\psi\rangle\langle\psi| \otimes |0\rangle\langle 0|^{\otimes e}$ the state of Alice prior to encoding. Let U be the attack made by Eve on the joint system, including the message space M and Eve's environment E. Bob's state prior to measuring but after applying the decoding operators is:

$$\rho_{Bob} = \frac{1}{2^m|\mathbb{P}_m|}\mathrm{Tr}_E\Big(\sum_{\substack{Q\in\mathbb{P}_m \\ k\in\{-1,1\}^m}} (QE_k\otimes\mathcal{I}_E)^\dagger U\Big((QE_k\otimes\mathcal{I}_E)\rho\otimes\rho_E(QE_k\otimes\mathcal{I}_E)^\dagger\Big)U^\dagger(QE_k\otimes\mathcal{I}_E)\Big)^\dagger \quad (79)$$

At this point we require Lemma 5.1 (it is analogous to Lemma 3.2), which allows us to reduce general adversary interventions to adversary interventions which are generalized Pauli operators. The lemma states that a random Pauli conjugating an operator has the effect of decohering (removing the cross terms) of the operator (we will prove this lemma in Section 5.2):

Lemma 5.1 (Pauli Decoherence) *Let ρ be a matrix on $m' > m$ qudits and let $U = \sum_{P\in\mathbb{P}_m} P\otimes U_P$ be a matrix acting on ρ. Then*

$$\frac{1}{|\mathbb{P}_m|}\sum_{Q\in\mathbb{P}_m}(Q\otimes\mathcal{I})^\dagger U(Q\otimes\mathcal{I})\rho(Q\otimes\mathcal{I})^\dagger U^\dagger(Q\otimes\mathcal{I}) = \sum_{P\in\mathbb{P}_m}(P\otimes U_P)\rho(P\otimes U_P)^\dagger$$

We decompose the attack U made by Eve to $U = \sum_{P\in\mathbb{P}_m} P \otimes U_P$ (where P acts on the space M and U_P acts on the space E) and then apply the lemma to equation 79 by replacing ρ in the lemma with $E_k\rho E_k^\dagger$, obtaining:

$$\begin{aligned}\rho_{Bob} &= \frac{1}{2^m}\mathrm{Tr}_E\Big(\sum_{P\in\mathbb{P}_m,k\in\{-1,1\}^m}(E_k^\dagger PE_k\otimes U_P)\rho\otimes\rho_E(E_k^\dagger PE_k\otimes U_P)^\dagger\Big) &(80)\\ &= \frac{1}{2^m}\sum_{P\in\mathbb{P}_m,k\in\{-1,1\}^m}E_k^\dagger PE_k\rho E_k^\dagger P^\dagger E_k\cdot\mathrm{Tr}(U_P\rho_E U_P^\dagger) &(81)\end{aligned}$$

We set $\alpha_P = \mathrm{Tr}\big(U_P^\dagger U_P\rho_E\big)$. Bob's state is now:

$$\cdots = \alpha_{\mathcal{I}}\cdot\rho + \frac{1}{2^m}\sum_{P\in\mathbb{P}_m\setminus\{\mathcal{I}\},k\in\{-1,1\}^m}\alpha_P\cdot E_k^\dagger PE_k\rho E_k^\dagger P^\dagger E_k \quad (82)$$

Recall that we are interested in the projection of Bob's state onto $\Pi_0^{|\psi\rangle}$ (defined in equation 4), which can now be written as:

$$\mathrm{Tr}(\Pi_0^{|\psi\rangle}\rho_{Bob}) = \alpha_{\mathcal{I}}\mathrm{Tr}(\Pi_0^{|\psi\rangle}\rho) + \frac{1}{2^m}\sum_{P\in\mathbb{P}_m\setminus\{\mathcal{I}\},k\in\{-1,1\}^m}\alpha_P\mathrm{Tr}(\Pi_0^{|\psi\rangle}E_k^\dagger PE_k\rho E_k^\dagger P^\dagger E_k) \quad (83)$$

$$= \frac{1}{2^m}\sum_{P\in\mathbb{P}_m\setminus\{\mathcal{I}\},k\in\{-1,1\}^m}\alpha_P\mathrm{Tr}(\Pi_0^{|\psi\rangle}E_k^\dagger PE_k\rho E_k^\dagger P^\dagger E_k) \quad (84)$$

Notice that each term in the above sum represents a generalized Pauli attack on the signed polynomial code. We now provide a lemma which states that the signed polynomial code allows detection of adversary interventions which are generalized Pauli operators (we will prove this lemma in Section 5.3):

Lemma 5.2 (Sign Key Security) *The signed polynomial code is $\frac{1}{2^{m-1}}$-secure against (generalized) Pauli attacks. More formally, for a density matrix $\rho = |\psi\rangle\langle\psi|\otimes|0\rangle\langle 0|^{\otimes m-1}$ and a generalized Pauli operator $P\in\mathbb{P}_m\setminus\{\mathcal{I}\}$:*

$$\frac{1}{2^m}\sum_{k\in\{-1,1\}^m}Tr(\Pi_0^{|\psi\rangle}E_k^\dagger PE_k\rho E_k^\dagger P^\dagger E_k)\leq\frac{1}{2^{m-1}} \quad (85)$$

We can now use the bound from Lemma 5.2 on each term in the sum in equation 84 to obtain:

$$\begin{aligned}\mathrm{Tr}(\Pi_0^{|\psi\rangle}\rho_{Bob}) &= \frac{1}{2^m}\sum_{P\in\mathbb{P}_m\setminus\{\mathcal{I}\},k\in\{-1,1\}^m}\alpha_P\mathrm{Tr}(\Pi_0^{|\psi\rangle}E_k^\dagger PE_k\rho E_k^\dagger P^\dagger E_k)) &(86)\\ &\leq \frac{1}{2^{m-1}}\sum_{P\in\mathbb{P}_m\setminus\{\mathcal{I}\}}\alpha_P &(87)\\ &= \frac{1-\alpha_I}{2^{m-1}} &(88)\\ &\leq \frac{1}{2^{m-1}} &(89)\end{aligned}$$

where the equality follows due to Lemma 3.3, which provides the following equality:

$$\sum_{P\in\mathbb{P}_m}\alpha_P = 1 \quad (90)$$

□

Similarly to the random Clifford authentication scheme, interventions that are very close to $\mathcal{I}$ are even more likely to keep the state in the space defined by $P_1^{|\psi\rangle}$.

We notice that in this scheme a q-dimensional system is encoded into a system of dimension $q^m = q^{2d+1}$ and achieves security of $\frac{1}{2^{m-1}}$. The Clifford QAS encodes a 2-dimensional system into a system of dimension 2^{1+e} and achieves security of 2^{-e}. The polynomial scheme is somewhat worse in parameters (since q must be at least 5), but still with exponentially good security.

To encode several registers, one can independently authenticate each register as in the Clifford case, but in fact we will see that we can use the same sign key k for all registers, while still maintaining security. This property will be extremely useful in applying gates as part of the polynomial QPIP protocol and we will use it in Section 6. For more details on how gates are applied on top of the signed polynomial code, see Section 2.5.1.

5.2 Proof of Lemma 5.1

We will require the following lemma:

Lemma 5.3 *For any two generalized Pauli operators P and Q*

$$Q^\dagger P Q \rho Q^\dagger P^\dagger Q = P \rho P^\dagger$$

*P*roof of **Lemma 5.3**: From the observation about generalized Pauli operators in Sec. 2 we know that for any two generalized Pauli operators P, Q $PQ = \beta QP$ where β is some phase (of magnitude 1) dependent on P and Q.

$$Q^\dagger P Q \rho Q^\dagger P^\dagger Q = Q^\dagger (\beta Q P) \rho (\beta^* P^\dagger Q^\dagger) Q = P \rho P^\dagger \tag{91}$$

□

We can now proceed to the proof:

*P*roof of **Lemma 5.1**: We start with:

$$\frac{1}{|\mathbb{P}_m|} \sum_{Q,P,P' \in \mathbb{P}_m} (Q \otimes \mathcal{I})^\dagger (P \otimes U_P)(Q \otimes \mathcal{I}) \rho (Q \otimes \mathcal{I})^\dagger (P' \otimes U_{P'})^\dagger (Q \otimes \mathcal{I})$$

We regroup elements to write the above expression as

$$\ldots = \frac{1}{\mathbb{P}_m} \sum_{Q,P,P' \in \mathbb{P}_m} (\mathcal{I} \otimes U_P)(Q^\dagger P Q \otimes \mathcal{I}) \rho (Q^\dagger P' Q \otimes \mathcal{I})^\dagger (\mathcal{I} \otimes U_{P'})^\dagger \tag{92}$$

We use Lemma 3.5 and are left only with $P = P'$

$$\ldots = \frac{1}{\mathbb{P}_m} \sum_{P,Q \in \mathbb{P}_m} (Q^\dagger P Q \otimes U_P) \rho (Q^\dagger P Q \otimes U_P)^\dagger \tag{93}$$

Now we use Lemma 5.3 :

$$\ldots = \sum_{P \in \mathbb{P}_m} (P \otimes U_P) \rho (P \otimes U_P)^\dagger \tag{94}$$

□

5.3 Proof of Lemma 5.2

In this section, we will prove Lemma 5.2 (security against Pauli attacks due to the sign key).

Proof: Our goal is to show for a density matrix $\rho = |\psi\rangle\langle\psi| \otimes |0\rangle\langle 0|^{\otimes m-1}$ and a generalized Pauli operator $P \in \mathbb{P}_m \setminus \{\mathcal{I}\}$:

$$\frac{1}{2^m} \sum_{k\in\{-1,1\}^m} \text{Tr}(\Pi_0^{|\psi\rangle} E_k^\dagger P E_k \rho E_k^\dagger P^\dagger E_k) \leq \frac{1}{2^{m-1}} \tag{95}$$

Throughout this proof, we will ignore phases which come about from Cliffords conjugating Pauli operators or moving Pauli operators past each other. This is due to the format of ρ_B; whenever we manipulate P to obtain a phase ω_q^a, we obtain ω_q^{-a} by manipulating $P^\dagger$ in the same manner. We first need to develop tools to understand how a generalized Pauli attack affects a signed polynomial state. This is done in the following subsection, after which we will return to the proof of Lemma 5.2.

5.3.1 k-Correlated Pauli Operators

We begin with definitions and their corresponding properties, and then proceed to analyze how generalized Pauli operators affect a signed polynomial state.

Definitions and Properties We now define what a correlated Pauli operator is:

Definition 5.1 *For a sign key $k \in \{-1,1\}^m$, we will call a non identity Pauli operator Q k-correlated if there exist one qudit states $|\psi\rangle$ and $|\phi\rangle$:*

$$QE_k |\psi\rangle \otimes |0\rangle^{m-1} = E_k |\phi\rangle \otimes |0\rangle^{m-1} \tag{96}$$

In other words, Q maps a state encoded with the signed polynomial code to another state with the same encoding and therefore cannot be detected. We will show that a non identity generalized Pauli operator Q can be k-correlated for at most 2 sign keys k according to the above definition. We will then show that for all sign keys k, all non identity Pauli operators Q which are *not* k-correlated can be written as a product of a k-correlated Pauli operator Q_k and a non identity, uncorrelated operator $\hat{Q}_k$ of a specific form. $\hat{Q}_k$ will always be detected by B's decoding procedure, as it will change the auxiliary states. This implies that a non identity Pauli operator will be caught with probability $\frac{1}{2^{m-1}}$ (it will be caught for all but at most two sign keys for which it is k-correlated).

Next, we describe what a k-correlated Pauli X operator looks like. We will require the following fact.

Fact 5.1 *For $k, \hat{k} \in \{-1,1\}^m$ (where $k \neq \hat{k}$), there exist polynomials f, g of degree at most d such that*

$$(k_1 f(\alpha_1), \ldots, k_m f(\alpha_m)) = (\hat{k}_1 g(\alpha_1), \ldots, \hat{k}_m g(\alpha_m)) \tag{97}$$

only if $k = -\hat{k}$.

Proof: There must either be at least $d+1$ indices on which k and $\hat{k}$ agree or at least $d+1$ indices on which they differ. First consider the case in which there are at least $d+1$ indices where they agree. Since the values of k and $\hat{k}$ at these indices uniquely define f and g, f and g must be equal. It follows that $k_i = \hat{k}_i$ for all i. If we instead consider the case when k and $\hat{k}$ differ on at least $d+1$ indices, we obtain that f is equal to $-g$ and therefore $k_i = -\hat{k}_i$. □

Claim 5.4 *A non identity Pauli operator $X = X^x$ is k-correlated if and only if it has the following form:*

$$X^x = \beta X^{k_1 f(\alpha_1)} \otimes \cdots \otimes X^{k_m f(\alpha_m)}$$

where β is a phase with $|\beta|^2 = 1$, f is a polynomial of degree at most d. The Pauli operator X^x can be k-correlated for at most 2 sign keys k.

Proof: It follows by Definition 5.1 that

$$X^{k_1 f(\alpha_1)} \otimes \cdots \otimes X^{k_m f(\alpha_m)} \tag{98}$$

is k-correlated. We will now show that if X^x is k-correlated it must have the form above. An X Pauli operator can only be k-correlated if it adds a low degree polynomial signed with k to the encoded state it is acting on. Therefore, if it is k-correlated, it must equal

$$X^{k_1 f(\alpha_1)} \otimes \cdots \otimes X^{k_m f(\alpha_m)} \tag{99}$$

for a polynomial f of degree at most d. We now show that a non identity Pauli operator X^x can be k-correlated for at most 2 sign keys. Assume now that X is also k'-correlated. By the argument above, it follows that it must equal

$$X^{k'_1 g(\alpha_1)} \otimes \cdots \otimes X^{k'_m g(\alpha_m)} \tag{100}$$

for a polynomial g of degree at most d. However, Fact 5.1 implies that either $k = k'$ or $k = -k'$. □

Next, we describe what a k-correlated Pauli Z operator looks like:

Claim 5.5 *A non identity Pauli operator $Z = Z^z$ is k-correlated if and only if it has the following form:*

$$Z^z = \beta Z^{c_1 k_1 f(\alpha_1)} \otimes \cdots \otimes Z^{c_m k_m f(\alpha_m)}$$

where β is a phase with $|\beta|^2 = 1$, f is a polynomial of degree at most d and c_i is the interpolation coefficient defined in Lemma 2.3 with the following property:

$$\sum_{1 \le i \le m} c_i f(\alpha_i) = f(0)$$

The Pauli operator Z^z can be k-correlated for at most 2 sign keys k.

Proof: We first show that if Z^z is k-correlated it must also have the form above. Assume that Z^z for $z \in F_q^m$ is k-correlated. Then it follows from Definition 5.6 that there exist one qudit states $|\psi\rangle$ and $|\phi\rangle$ such that:

$$Z^z E_k |\psi\rangle \otimes |0\rangle^{m-1} = E_k |\phi\rangle \otimes |0\rangle^{m-1} \tag{101}$$

Now if we apply a logical Fourier operator (described in Claim 2.4), the left hand side of the above equation becomes:

$$\begin{aligned} \tilde{F} Z^z E_k |\psi\rangle \otimes |0\rangle^{m-1} &= \tilde{F} Z^z \tilde{F}^\dagger \tilde{F} E_k |\psi\rangle \otimes |0\rangle^{m-1} && (102) \\ &= \tilde{F} Z^z \tilde{F}^\dagger E_k F |\psi\rangle \otimes |0\rangle^{m-1} && (103) \end{aligned}$$

where the last equality follows since $\tilde{F}$ is a logical operator (which is also proven in Claim 2.4). The right hand side of equation 101 becomes:

$$\tilde{F} E_k |\phi\rangle \otimes |0\rangle^{m-1} = E_k F |\phi\rangle \otimes |0\rangle^{m-1} \tag{104}$$

Then (by setting the right and left hand side of the equations equal to each other) we have:

$$\tilde{F} Z^z \tilde{F}^\dagger E_k F |\psi\rangle \otimes |0\rangle^{m-1} = E_k F |\phi\rangle \otimes |0\rangle^{m-1} \tag{105}$$

Due to the conjugation properties of $\tilde{F}$ (for more details about the conjugation behavior of $\tilde{F}$, see the Fourier description in Section 2.4), we have (where α is a phase)

$$\tilde{F} Z^z \tilde{F}^\dagger = \alpha X^{-c_1^{-1} z_1} \otimes \cdots \otimes X^{-c_m^{-1} z_m} \tag{106}$$

and due to equation 105 we can see that the above operator is k-correlated. By Claim 5.4, the X Pauli operator in equation 106 can be k-correlated for at most 2 sign keys k. It follows that the Pauli operator Z^z can be k-correlated for at most 2 sign keys. Finally, if Z^z is indeed k-correlated, we can combine the fact that the X operator in equation 106 is k-correlated and Claim 5.4, to write Z^z as:

$$Z^{c_1 k_1 f(\alpha_1)} \otimes \cdots \otimes Z^{c_m k_m f(\alpha_m)} \tag{107}$$

for a polynomial f of degree at most d. We now need to show the opposite direction: if Z^z can be written as

$$Z^{c_1 k_1 f(\alpha_1)} \otimes \cdots \otimes Z^{c_m k_m f(\alpha_m)} \tag{108}$$

it is k-correlated. To see this, we can obtain the following equality from equation 106

$$\tilde{F} Z^z \tilde{F}^\dagger = \alpha X^{-k_1 f(\alpha_1)} \otimes \cdots \otimes X^{-k_m f(\alpha_m)} \tag{109}$$

Since this is a correlated X operator, it follows by Definition 5.1 that Z^z is a correlated Z operator. □

We can extend the claims for X and Z Pauli operators to general Pauli operators:

Claim 5.6 *A non identity Pauli operator can be k-correlated for at most 2 sign keys k.*

Proof: Consider a non identity Pauli operator $Z^z X^x$. In order for it to be correlated, X^x must add a low degree signed polynomial to a state encoded by E_k which it is acting on. This means that X^x is k-correlated. It follows by Claim 5.4 that if $x \neq 0$, the Pauli operator $Z^z X^x$ can be k-correlated for at most 2 sign keys k. If $x = 0$, Claim 5.5 implies that the Pauli operator $Z^z X^x$ can be k-correlated for at most 2 sign keys. □

Correlation Properties of Generalized Pauli Operators Now that we have defined k-correlation, we can see how a generalized Pauli operator will behave on a signed polynomial state. We begin by showing that for a fixed sign key k, a Pauli operator Q can be broken down into a product of a k-correlated Pauli operator and an uncorrelated Pauli operator:

Claim 5.7 *Let k be a sign key and $Q \in \mathbb{P}_m \setminus \{\mathcal{I}\}$ be an uncorrelated Pauli operator. Then $Q = Z^z X^x$ can be written as*

$$Q = \hat{Q}_k Q_k \tag{110}$$

where Q_k is k-correlated and $\hat{Q}_k$ is uncorrelated (and in particular, non identity) and can be written (up to a phase) as:

$$\mathcal{I} \otimes Z^{\hat{z}_2} \otimes \cdots Z^{\hat{z}_{d+1}} \otimes X^{\hat{x}_{d+2}} \cdots \otimes X^{\hat{x}_m} \tag{111}$$

where if $z = 0$, $(\hat{z}_2, \ldots, \hat{z}_{d+1}) = 0^d$ and if $x = 0$, $(\hat{x}_{d+2}, \ldots, \hat{x}_m) = 0^d$.

Proof: Observe that a signed low degree polynomial is determined by $d+1$ points. For a given sign key k and $d+1$ points $y_{i_1}, \ldots, y_{i_{d+1}} \in F_q$, where $i_1, \ldots, i_{d+1} \in \{1, \ldots, m\}$, let

$$s_k(y_{i_1}, \ldots, y_{i_{d+1}}) = (k_1 f(\alpha_1), \ldots, k_m f(\alpha_m)) \in F_q^m \tag{112}$$

be the signed polynomial that is obtained by interpolating the $d+1$ points $y_{i_1}, \ldots, y_{i_{d+1}}$. Let:

$$[s'_k(y_{i_1}, \ldots, y_{i_{d+1}})]_i = c_i \cdot [s_k(y_{i_1}, \ldots, y_{i_{d+1}})]_i \tag{113}$$

For $Q = Z^z X^x = Z^{z_1} X^{x_1} \otimes \cdots \otimes Z^{z_m} X^{x_m}$, we claim that

$$Q_k = (Z^{s'_k(c_1^{-1} z_1, c_{d+2}^{-1} z_{d+2}, \ldots, c_m^{-1} z_m)})(X^{s_k(x_1, \ldots, x_{d+1})}) \tag{114}$$

is k-correlated. Claims 5.4 and 5.5 imply that both the Z and X operators of Q_k are k-correlated. It follows by the definition of a k-correlated operator (Definition 5.1) that the product of two k-correlated operators (Q_k in this case) is k-correlated. Now we define $\hat{Q}_k$ such that:

$$\hat{Q}_k = QQ_k^\dagger \tag{115}$$

It can be readily checked, using the definition of Q_k in equation 114, that $\hat{Q}_k$ is of the following form (up to a phase):

$$\hat{Q}_k \equiv \mathcal{I} \otimes Z^{\hat{z}_2} \otimes \cdots Z^{\hat{z}_{d+1}} \otimes X^{\hat{x}_{d+2}} \cdots \otimes X^{\hat{x}_m} \tag{116}$$

Observe that $\hat{Q}_k$ as written in the above equation is uncorrelated. If it was k-correlated, then Q would be a product of two k-correlated operators (Q_k and $\hat{Q}_k$) which implies that Q is also k-correlated, which contradicts our starting assumption that Q is uncorrelated. Observe also that the last line of the claim (about the implication of $z = 0$ or $x = 0$) follows immediately from the definitions of Q_k and $\hat{Q}_k$. □

Observe that an operator of the form of $\hat{Q}_k$ is always detected, as it will change the auxiliary qudits:

Claim 5.8 *For all one qudit states $|\psi\rangle$ and $|\phi\rangle$ and a fixed sign key $k \in \{-1, 1\}^m$, an uncorrelated operator $\hat{Q}_k$ of the form described in Claim 5.7 in equation 111 satisfies the following equation:*

$$Tr(\Pi_0^{|\psi\rangle} E_k^\dagger \hat{Q}_k E_k \left|\phi\right\rangle \left\langle\phi\right| \otimes \left|0\right\rangle \left\langle 0\right|^{\otimes m-1} E_k^\dagger \hat{Q}_k^\dagger E_k) = 0 \tag{117}$$

Proof: We claim the following equality holds up to a phase:

$$E_k^\dagger \hat{Q}_k E_k = \mathcal{I} \otimes X^{\hat{z}_2} \otimes \cdots X^{\hat{z}_{d+1}} \otimes X^{\hat{x}_{d+2}} \cdots \otimes X^{\hat{x}_m} \tag{118}$$

Recall (from Section 2.5.2) that

$$E_k = D_k(\mathcal{I} \otimes F^{\otimes d} \otimes \mathcal{I}) \tag{119}$$

The conjugation behavior of $E_k^\dagger$ can be determined by looking at the conjugation properties of Clifford operators (see Section 2.4). As a brief description, recall from Section 2.5.2 that $E_k^\dagger$ consists of the interpolation circuit ($D_k^\dagger$), which is a series of inverse controlled sum operations and an inverse multiplication operator on the first register (see Claim 2.6). The final operation in $E_k^\dagger$ is an inverse Fourier transform ($(\mathcal{I} \otimes F^{\otimes d} \otimes \mathcal{I})^\dagger$). Using the conjugation properties given in equations 21, 23, and 22 in Section 2.4, we obtain the following equalities. Inverse sum operations have the following conjugation behavior (up to a phase):

$$SUM^\dagger(Z^{z_1}X^{x_1} \otimes Z^{z_2}X^{x_2})SUM = Z^{z_1+z_2}X^{x_1} \otimes Z^{z_2}X^{x_2-x_1} \tag{120}$$

where the *SUM* operator above is controlled by the left register. The inverse multiplication operation, $M_r^\dagger$ (for $r \neq 0$), has the following conjugation behavior (up to a phase):

$$M_r^\dagger(Z^z X^x)M_r = Z^{rz}X^{r^{-1}x} \tag{121}$$

Fourier operations have the following conjugation behavior (up to a phase):

$$F^\dagger Z^z X^x F = X^z Z^{-x} \tag{122}$$

The inverse sum operations in $D_k^\dagger$ have no effect on $\hat{Q}_k$, since the target registers (registers $1, d+2, \ldots, m$) never have a non zero Z coefficient in equation 118 and the control registers (registers $1, \ldots, d+1$) never have a non zero X coefficient in equation 118. In other words, the coefficients x_1 and z_2 in equation 120 will be 0. The multiplication operation (which is in between the inverse sum operations) similarly has no effect on $\hat{Q}_k$; it is acting on the first register, for which both the Z and X coefficient are 0 (since it is $\mathcal{I}$). The inverse Fourier operation flips the Z operators of registers $2, \ldots, d+1$ to X operators.

Now, returning to equation 118, since $\hat{Q}_k$ was not equal to the identity, it will make at least one of the auxiliary qudits nonzero (recall that the auxiliary qudits are contained in registers $2, \ldots, m$). □

5.3.2 Proof of Lemma 5.2

We now use the concepts we developed in the above section to prove Lemma 5.2. To summarize, what we have shown is that for each sign key k, a generalized non identity Pauli operator Q can be broken down into a product of a k-correlated operator Q_k and an uncorrelated operator $\hat{Q}_k$ (Claim 5.7). The uncorrelated operator $\hat{Q}_k$ will always be detected (Claim 5.8) and will only be non identity for at most 2 sign keys k (Claim 5.6). Therefore, Q can only preserve a signed polynomial state for at most 2 sign keys k.

Recall that we would like to upper bound the following expression:

$$\frac{1}{2^m}\sum_{k\in\{-1,1\}^m}\mathrm{Tr}(\Pi_0^{|\psi\rangle}E_k^\dagger P E_k \rho E_k^\dagger P^\dagger E_k) \tag{123}$$

By Claim 5.6, P can be k-correlated for at most 2 sign keys k. Consider one k in the above sum for which P is not k-correlated. We can now apply Claim 5.7, to obtain that the term is equal to

$$\begin{aligned}
\cdots &= \mathrm{Tr}(\Pi_0^{|\psi\rangle}E_k^\dagger \hat{P}_k P_k E_k |\psi\rangle\langle\psi| \otimes |0\rangle\langle 0|^{\otimes m-1} E_k^\dagger(\hat{P}_k P_k)^\dagger E_k) && (124)\\
&= \mathrm{Tr}(\Pi_0^{|\psi\rangle}E_k^\dagger \hat{P}_k E_k |\psi_{P_k}\rangle\langle\psi_{P_k}| \otimes |0\rangle\langle 0|^{\otimes m-1} E_k^\dagger \hat{P}_k^\dagger E_k) && (125)\\
&= 0 && (126)
\end{aligned}$$

where the second equality follows from the fact that P_k is a k-correlated operator and the final equality follows from Claim 5.8. Then we only obtain a non zero expression when P is k-correlated. It follows that

$$\frac{1}{2^m}\sum_{k\in\{-1,1\}^m}\mathrm{Tr}(\Pi_0^{|\psi\rangle}E_k^\dagger P E_k \rho E_k^\dagger P^\dagger E_k) \leq \frac{1}{2^{m-1}} \tag{127}$$

□

6 Quantum Interactive Proofs with Polynomial Authentication

In this section, we give a QPIP for Q-CIRCUIT (providing another proof for Theorem 1.1) using the signed polynomial encoding from the previous section. The key advantage of this protocol is that the prover can perform the gates on top of the encoding without knowing the encoding itself. This means that the prover does not need to hand back the qudits to the verifier in order for the verifier to perform the gates; the prover can perform them on his own. This also means that only one way quantum communication is required (the verifier only needs to send qudits at the start of the protocol, and the rest of the communication is classical).

The key disadvantage of this protocol is the relative difficulty of proving soundness in comparison to the Clifford QPIP protocol (Theorem 1.5). This difficulty arises due to the difference between Lemma 5.1 (Pauli decoherence) and Lemma 3.2 (Clifford decoherence). The strength of Clifford decoherence allows us to prove Claim 4.2 (the Clifford state evolution claim), which states that the prover's state throughout the protocol is the correct authenticated state (i.e. the state with the gates applied as requested by the verifier) with an attack independent of the authentication acting on top of it. Essentially, this is because Claim 4.2 uses the unitary commutation lemma, Lemma 4.6, and the Clifford decoherence lemma, Lemma 3.2, to change any logical attack (an attack acting *inside* the authentication) to an attack *outside* of the authentication (which no longer preserves the authenticated state). Claim 4.2 then allows us to reduce the soundness of the Clifford QPIP to the security of the Clifford QAS.

We cannot use Pauli decoherence (Lemma 5.1) to prove a claim analogous to Claim 4.2 in the polynomial case for the following reason. Lemma 5.1 shows that averaging over the Pauli conjugations of an operator removes cross terms, thereby mapping the operator to a convex sum over Pauli operators. Lemma 3.2 shows that averaging over

the Clifford conjugations of an operator not only maps the operator to a convex sum over Pauli operators, but goes one step further to map each non identity Pauli operator to a unifrom mixture over all non identity Pauli operators. This uniform mixture is crucial to the proof of Claim 4.2; the key part of the proof is the application of the unitary commutation lemma (Lemma 4.6) to the mixture, which allows us to shift the prover's attacks to the end of the protocol.

Since we do not have a claim analogous to Claim 4.2 for the polynomial QPIP, we instead have to monitor how the authenticated state changes throughout the protocol, as a function of the prover's deviation. At a high level, we do this by partitioning the Hilbert space of the prover according to the interaction transcript (as done in [FK12]). In each partition, the transcript is fixed at the start and then the measurement results of the state are projected onto the fixed transcript to enforce consistency. This method is formalized in Claim 6.1, which describes the state shared by the verifier and prover throughout the protocol.

Now since each partition has a fixed interaction transcript, we can shift the prover's attack to the end of the protocol (his attack no longer determines the interaction transcript). After shifting the prover's attack, we can analyze each partition using the same main ideas we used to prove security of the polynomial QAS (Pauli decoherence from Lemma 5.1 and sign key security from Lemma 5.2).

We begin by discussing how to apply gates on top of the signed polynomial authentication (Section 6.1). We then describe the protocol, introduce necessary notation and assumptions and conclude with proving the soundness and completeness of the protocol.

6.1 Application of Quantum Gates

We will describe how the prover performs a set of universal gates (consisting of the Fourier transform and Toffoli gate) on authenticated qubits by applying only Clifford operators which do not require knowledge of the Pauli or sign keys. The prover does this by using classical communication with the verifier and authenticated Toffoli states sent by the verifier. As described in Section 2.3.1, if given an authenticated Toffoli state, a Toffoli gate can be applied using logical Pauli, *SUM* and Fourier operations, along with measurement. We now describe how to apply these operations on authenticated states, which will complete our description of how the prover performs a universal set of gates.

6.1.1 Pauli Operations

To apply Pauli X and Z operations, the verifier only needs to update his Pauli keys and the prover does not need to do anything. Recall from Section 2.5.1 that the logical $\widetilde{X}_k$ operator consists of an application of $X^{k_1}\otimes,\ldots\otimes X^{k_m}$ where $k\in\{-1,1\}^m$ is the sign key. We claim that this operation can be applied to the authenticated state by the verifier simply changing his Pauli key from (x,z) to $(x-k,z)$. This is because:

$$\begin{aligned} P_{x,z}\left|S_a^k\right\rangle &= P_{x-k,z}P^\dagger_{x-k,z}P_{x,z}\left|S_a^k\right\rangle && (128)\\ &= P_{x-k,z}X^{-(x-k)}Z^{-z}Z^zX^x\left|S_a^k\right\rangle && (129)\\ &= P_{x-k,z}(X^{k_1}\otimes,\ldots\otimes X^{k_m})\left|S_a^k\right\rangle && (130)\\ &= P_{x-k,z}\widetilde{X}_k\left|S_a^k\right\rangle && (131)\end{aligned}$$

The Z operator is performed in the same manner as the X operator; all that is needed is a change of the Pauli key. We recall that $\widetilde{Z}_k=Z^{c_1k_1}\otimes\ldots\otimes Z^{c_mk_m}$. We define the vector $\mathbf{t}$ to be $t_i=c_ik_i$. From the same argument as above, it holds that the change of keys must be $(x,z)\to(x,z-\mathbf{t})$.

6.1.2 Fourier and *SUM* Operations

To apply Fourier and *SUM* operations, the verifier needs to update his Pauli keys and the prover needs to apply the corresponding logical gate. For the *SUM* gate, the prover applies the logical *SUM* gate ($\widetilde{SUM}$ as given in Section 2.5.1) and the verifier updates his pair of keys (for $x_A, z_A, x_B, z_B \in F_q^m$) from $(x_A, z_A), (x_B, z_B)$ to $(x_A, z_A - z_B)$ and $(x_B + x_A, z_B)$ where A is the control register and B is the target register. This is because the logical *SUM* operator is applied on top of the Pauli keys, and must be shifted past. The update operations of the verifier essentially perform this shift:

$$\widetilde{SUM}(Z^{z_A}X^{x_A} \otimes Z^{z_B}Z^{x_B}) \left|S_a^k\right\rangle \left|S_b^k\right\rangle = \widetilde{SUM}(Z^{z_A}X^{x_A} \otimes Z^{z_B}Z^{x_B})\widetilde{SUM}^\dagger \widetilde{SUM} \left|S_a^k\right\rangle \left|S_b^k\right\rangle \quad (132)$$

$$= (Z^{z_A - z_B}X^{x_A} \otimes Z^{z_B}X^{x_A + x_B})\widetilde{SUM} \left|S_a^k\right\rangle \left|S_b^k\right\rangle \quad (133)$$

where the last equality is up to a global phase and follows due to the conjugation properties given in Section 2.4.

The Fourier gate is applied in a similar way; the prover applies the logical Fourier transform $\widetilde{F}$ given in Section 2.5.1 (Claim 2.4) and the verifier updates his keys according to the conjugation behavior of $\widetilde{F}$, which we can determine from Section 2.4. The following equality is up to a global phase:

$$\widetilde{F}(Z^z X^x)\widetilde{F}^\dagger = Z^{c_1 x_1}X^{-c_1^{-1}z_1} \otimes \ldots \otimes Z^{c_m x_m}X^{-c_m^{-1}z_m} \quad (134)$$

Therefore, for each register i, the verifier must change the key from (x_i, z_i) to $(-c_i^{-1}z_i, c_i x_i)$.

6.1.3 Measurement

The prover measures the encoded state in the standard basis and sends the resulting string in F_q^m to the verifier. The verifier first removes the entire Pauli key. Note that we are assuming a classical verifier can remove the Z portion of the Pauli key; this is because the Pauli key is acting on a measured string, and phase gates have no effect on standard basis strings. Therefore, applying a Z operator is the same as not applying it. We choose to assume the verifier does apply it because it simplifies the soundness proof of the protocol (specifically, it comes up in the proof of Claim 6.3). The verifier then applies $D_k^\dagger$ (see Section 2.5.2), obtaining a string $\delta \in F_q^m$. If the prover requires the decoded measurement result, the verifier sends the prover the first coordinate of δ (which should contain the value of the polynomial at 0). If the last d coordinates of δ are not 0, the verifier records the measurement as invalid and aborts at the end of the protocol.

Observe that the verifier is not applying $E_k^\dagger$ (the full decoding circuit). It turns out that this is actually enough for the interactive protocol, since we only need to be able to catch attack operators involving Pauli X deviations. Attack operators involving Z deviations will not change measurement results. We will see below (in Corollary 6.9) that applying $D_k^\dagger$ and checking the appropriate auxiliary qudits allows the verifier to catch Pauli X deviations.

6.1.4 Conversion to Logical Circuit

Now that we have described how to apply gates, we can describe how to convert a quantum circuit on n qubits consisting of gates from the above universal set, $U = U_N \cdots U_1$, into a logical circuit acting on authenticated states. Assume U contains L Toffoli gates. Then

$$U = A_L T_L \cdots A_1 T_1 A_0 \quad (135)$$

where A_i is a Clifford circuit. To apply U to authenticated states, we instead apply

$$\tilde{A}_L \tilde{T}_L \cdots \tilde{A}_1 \tilde{T}_1 \tilde{A}_0 \quad (136)$$

where $\tilde{A}_L$ denotes a logical operation, as described above. Each $\tilde{T}_i$ involves Clifford entanglement operations (which we will denote by $\tilde{B}_i$) with a new magic state followed by a measurement, the results of which are sent to the verifier. Assume that the i^{th} measurement result decodes to $\beta_i \in F_q^3$. Then $\tilde{T}_i$ consists of $\tilde{B}_i$, followed by measurement, followed by correction $\tilde{C}_{\beta_i}$ (which is the logical version of C_{β_i} - see Section 2.3.1 for a reminder of how the Toffoli gate is applied). Now combine the Clifford entangling operators with the preceding Clifford operators in the circuit:

$$\tilde{Q}_i = \tilde{B}_{i+1}\tilde{A}_i \tag{137}$$

where $B_{L+1} = \mathcal{I}$. Then to apply U to authenticated states, we first apply $\tilde{Q}_0$. Then for $1 \leq i \leq L$, we measure, obtaining $\beta_i \in F_q^3$, and then apply $\tilde{Q}_i\tilde{C}_{\beta_i}$.

Properties of Toffoli Gate by Teleportation In order to prove soundness of the polynomial QPIP, we will need to better understand the result of applying a circuit using Toffoli states (as described immediately above in Section 6.1.4). More specifically, this understanding will come in to play when we are analyzing the behavior of Pauli attacks on the state at the end of the protocol (this is done in Claim 6.4 and Claim 6.5). In this section, we will not work with logical operators and authenticated qudits, but with unauthenticated qudits. However, the analysis can immediately be extended to authenticated qudits. To begin, assume the measurement results $\beta_i \in F_q^3$ of each Toffoli gate are fixed beforehand. Then the circuit which will be applied (as described above) is:

$$Q_L C_{\beta_L} \cdots Q_1 C_{\beta_1} Q_0 \tag{138}$$

We will now provide a fact (used in Claim 6.4 and Claim 6.5) which characterizes what the state looks like (including measurement results) after applying the circuit in equation 138 on $n + 3L$ qudits (the circuit acts on n input qudits initially in state $|\phi\rangle$ and L Toffoli states of 3 qudits each):

Fact 6.1 *For a string $\beta = (\beta_1, \ldots, \beta_L) \in F_q^{3L}$, where $\beta_i \in F_q^3$, the result of applying*

$$Q_L C_{\beta_L} \cdots Q_1 C_{\beta_1} Q_0 \tag{139}$$

to

$$|\phi\rangle \, (\frac{1}{q} \sum_{a,b \in F_q} |a, b, ab\rangle)^{\otimes L} \tag{140}$$

is

$$\frac{1}{\sqrt{q^{3L}}} \sum_{l \in F_q^{3L}} |l\rangle \, |\psi\rangle_{\beta,l} \tag{141}$$

where $|\phi\rangle$ is a state on n qudits and $|\psi\rangle_{\beta,l}$ is a state on n qudits which equals $U|\phi\rangle$ if $\beta = l$.

Before proving the fact, observe that if we project the first register containing l onto β, we obtain the state $U|\psi\rangle$. This makes sense; if the measurement results obtained are the same ones we fixed for the Clifford corrections, then each Toffoli gate is applied as intended. Moreover, note that without this projection, each $l \in F_q^{3L}$ is equally probable. We now prove the fact.

*P*roof of **Fact6.1**: First consider what happens if we would like to apply one Toffoli (as described above) to a 3 qudit state $|\psi\rangle$ using a magic state. After the Clifford operations entangling $|\psi\rangle$ and the magic state, but preceding the measurement (i.e. at the stage of equation 17), the state is:

$$\begin{aligned} \frac{1}{\sqrt{q^3}} \sum_{a,b,l \in F_q} \omega^{-le} \, |a, b, ab + e, c - a, d - b, l\rangle &= \frac{1}{\sqrt{q^3}} \sum_{x,y,z \in F_q} \omega^{-ze} \, |c - x, d - y, (c - x)(d - y) + e\rangle \, |x, y, z\rangle \\ &= \frac{1}{\sqrt{q^3}} \sum_{x,y,z \in F_q} ((T(X^x \otimes X^y \otimes Z^z)T^\dagger)^\dagger T \, |\psi\rangle) \, |x, y, z\rangle \end{aligned} \tag{142}$$

We can write the state in this format because we know that when the measurement result is x, y, z, the operation $T(X^x \otimes X^y \otimes Z^z)T^\dagger$ corrects the state to $T|\psi\rangle$. Let's swap the first 3 registers with the last 3 registers (to be consistent with how the state is written in Fact 6.1). The state is now:

$$\frac{1}{\sqrt{q^3}} \sum_{x,y,z \in F_q} |x, y, z\rangle \, (T(X^x \otimes X^y \otimes Z^z)T^\dagger)^\dagger T|\psi\rangle \tag{143}$$

Now let's assume the correction operator applied is $T(X^a \otimes X^b \otimes Z^c)T^\dagger$. The result is:

$$\frac{1}{\sqrt{q^3}} \sum_{x,y,z \in F_q} |x, y, z\rangle \otimes |\psi\rangle_{(a,b,c),(x,y,z)} \tag{144}$$

where

$$|\psi\rangle_{(a,b,c),(x,y,z)} = T(X^{a-x} \otimes X^{b-y} \otimes Z^{c-z})|\psi\rangle \tag{145}$$

Observe that

$$|\psi\rangle_{(a,b,c),(a,b,c)} = T|\psi\rangle \tag{146}$$

In other words, when the correction β matches the measurement result l, we get the desired result: the application of a Toffoli gate to the initial state. To prove the fact, the above analysis is simply applied each time the Toffoli protocol is performed. □

6.2 Protocol Description

Protocol 6.1 ***Polynomial based Interactive Proof for Q-CIRCUIT*** *Fix a security parameter ϵ. Given is a quantum circuit on n qubits consisting of gates from the above universal set, $U = U_N \cdots U_1$, which can be converted to a logical circuit on authenticated qudits as in Section 6.1.4. We assume there are L Toffoli gates. We assume the circuit U has error probability $\leq \gamma$. The verifier sets $m = \lceil \log \frac{1}{\epsilon} \rceil + 1$, $d = \frac{m-1}{2}$ and uses 3 registers of m qudits each, where each qudit is of dimensionality $q > m$. The verifier uses the polynomial QAS with security parameter d to authenticate n input qudits and L Toffoli states and sends the authenticated states to the prover. The verifier uses the same sign key (but independent Pauli keys) for each state. This is done sequentially using $3m$ qudits at a time. Round 0 consists of the prover and verifier performing the Clifford gates $\tilde{Q}_0$. At the start of round i, for $1 \leq i \leq L$, the prover and verifier perform the measurement (as described in Section 6.1.3) on the $3m$ qudits as required for the i^{th} Toffoli gate. The verifier sends the prover the decoded measurement result, and then they jointly perform the Clifford corrections required to complete the Toffoli gate and the Clifford circuit $\tilde{Q}_i$. In round $L + 1$ (the final round), the verifier and prover perform the measurement of the first authenticated qudit (the verifier does not provide the prover with the decoded measurement result). The verifier aborts if the measurement results from any round were stored as invalid (see Section 6.1.3). If he does not abort, he accepts or rejects according to the final decoded measurement outcome.*

Theorem 1.7 *For $0 < \epsilon < 1$ and $\gamma < 1 - \epsilon$, Protocol 6.1 is a $\mathsf{QPIP}_{O(\log(\frac{1}{\epsilon}))}$ protocol with completeness $1 - \gamma$ and soundness $\gamma + \epsilon$ for $\mathsf{Q\text{-}CIRCUIT}_\gamma$.*

This theorem implies a second proof for Theorem 1.1. The size of the verifier's register is naively $3m$, but using the same idea as in the Clifford case, $m + 2$ suffice. As a reminder, the idea is to send qudits as they are encoded. For the Toffoli state, the verifier begins with 3 qudits, encodes the first one (using $m + 2$ registers at this point), sends the first encoded qudit to the prover, and continues. With $\epsilon = 1/2$, $m = 3$ (because $m = 2d + 1$) giving a register size of 5 qudits of dimension 5 (since $q > m$). Before we provide the proof of the theorem, we introduce some necessary notation and make several observations about the protocol described above.

6.3 Assumptions

- **The prover's messages are quantum states.** Note that although in the protocol the prover sends the verifier classical strings which the verifier then decodes, we can instead assume that the prover sends the verifier qudits, then the verifier decodes and finally measures. This is because the verifier's decoding operations (which consist of removing Pauli keys and applying $D_k^\dagger$, as described in Section 6.1.3) commute with standard basis measurement. In other words, if you consider an m qudit density matrix ρ, the following equality holds:

$$\sum_{j \in F_q^m} D_k^\dagger (Z^z X^x)^\dagger \left|j\right\rangle \left\langle j\right| \rho \left|j\right\rangle \left\langle j\right| Z^z X^x D_k = \sum_{j \in F_q^m} \left|j\right\rangle \left\langle j\right| D_k^\dagger (Z^z X^x)^\dagger \rho Z^z X^x D_k \left|j\right\rangle \left\langle j\right| \tag{147}$$

- **The prover's deviation can be delayed until the end of each round.** In round i (for $i \geq 1$), we can assume without loss of generality that the prover measures, sends the results to the verifier, receives the decoded measurement results $g(\delta_i)$ from the verifier, and then applies a unitary $\hat{V}_{g(\delta_i)}$ to the authenticated qudits and his extra space. Anything the prover does before the measurement can be shifted to the previous round. $\hat{V}_{g(\delta_i)}$ can be written as

$$\hat{V}_{g(\delta_i)} = \hat{V}_{g(\delta_i)} (\tilde{Q}_i \tilde{C}_{\beta_i})^\dagger \tilde{Q}_i \tilde{C}_{\beta_i} = V_{g(\delta_i)} \tilde{Q}_i \tilde{C}_{\beta_i} \tag{148}$$

  In other words, we can assume that the prover measures, applies the unitaries requested by the verifier in round i ($\tilde{Q}_i \tilde{C}_{\beta_i}$) and then applies a unitary attack $V_{g(\delta_i)}$. Using similar reasoning, in round 0, we can assume the prover first applies $\tilde{Q}_0$ as requested and then applies a unitary attack.

6.4 Notation

Now we provide some of the notation that will be used in the proof. Please see the notation tables in Appendix E for all notations together, which hopefully will help in reading this part of the paper, since it is quite heavy on notation. First, throughout this protocol, we will refer to Z and $\mathcal{I}$ Pauli operators as trivial and all other Pauli operators (Pauli operators containing the X operator) as non trivial. This is because, as noted in Section 6.1.3, trivial Pauli operators cannot change measurement results and non trivial Pauli operators can. Therefore, we only need to ensure that the verifier can detect non trivial Pauli operators.

For strings $b_1, \ldots, b_j \in F_q^m$, let

$$g((b_1, \ldots, b_j)) = (b_1(1), \ldots, b_j(1)) \tag{149}$$

where $b_i(1) \in F_q$ is the first value of the string. This is referred to as the decoded value of a measurement result, as it is the value the verifier will return to the prover after decoding the prover's measurement result.

If the prover decides to deviate from the protocol, he can apply unitary operators to both the qudits sent by the verifier and his environment. We call the register corresponding to his environment $\mathcal{E}$. For convenience, we also label the other registers of the quantum state shared between the prover and verifier as follows. Note that at the start of round 0, the total number of qudits sent to the prover by the verifier is $m' = 3mL + mn$. In every following round, except round $L+1$, $3m$ qudits (which are to be measured) are sent to the verifier (m qudits to be measured are sent in round $L+1$). For $i \geq 1$, we call the register containing all authenticated qudits left at the prover's hands at the beginning of round i (equivalently at the end of round $i-1$) register $\mathcal{P}_i$ (it holds $m(3(L-i+1)+n)$ qudits). We call the register containing all the qudits sent to the verifier in rounds $1, \ldots, i-1$ register $\mathcal{V}_i$ (it holds $3m(i-1)$ qudits and is held by the verifier). We also add another register to the verifier's space (which we call the key register): at the start of round i, it contains the state $\tau_i(z, x, k)$, which carries the memory of the sign key k and

the Pauli keys $x, z \in F_q^{|\mathcal{P}_i|}$ for those qudits still held by the prover (qudits in register $\mathcal{P}_i$). More precisely:

$$\tau_i(z,x,k) = |z\rangle\langle z| \otimes |x\rangle\langle x| \otimes |k\rangle\langle k|$$

Note that we are assuming the verifier no longer keeps record of the Pauli keys for qudits which were already sent to him by the prover; after the verifier uses these Pauli keys to decode, he traces them out of the key register.

Given this notation, we can now provide Figure 1 as an illustration of Protocol 6.1.

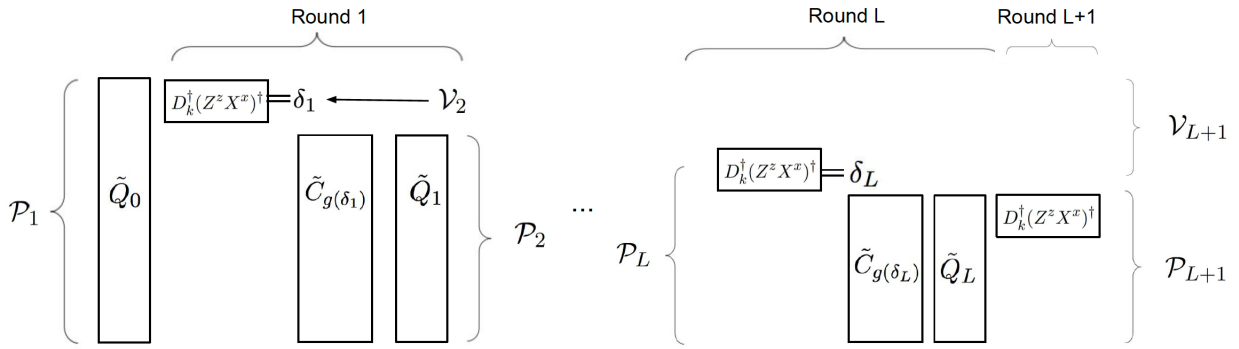

Figure 1: This figure illustrates the gates an honest prover would apply during Protocol 6.1, and which qubits are in which register during different rounds. In the figure, we are assuming that the measurement result in round i (after the verifier removes the Pauli key and decodes with $D_k^\dagger$) is $\delta_i \in F_q^{3m}$. To simplify the illustration, we have left out the detail that the decoding Pauli keys $(Z^z X^x)^{-1}$ will be different for each register, and that the final decoding (at the start of round $L+1$) acts on m qudits, while the previous i decodings act on $3m$ qudits.

6.5 Overall Proof of Theorem 1.7

Proof: The completeness is trivial, similarly to the Clifford case (see Theorem 1.5). To prove soundness, recall that we begin with a Q-CIRCUIT instance, U, which takes as input $|y\rangle^{\otimes n}$ (where y is a classical n bit string), and for soundness we would like to show that if the first qudit of $U|y\rangle^{\otimes n}$ is 0 with probability $1-\gamma$, the verifier will either abort or not accept the final decoded measurement result with probability $\geq 1-(\gamma+\epsilon)$, which gives the soundness parameter of $\gamma+\epsilon$. To do this, we will characterize how the prover's state evolves throughout the protocol.

When each qudit is sent to the verifier at the start of round i as part of the application of the Toffoli gate, the verifier will apply the inverse of the appropriate Pauli keys, interpolate with operator $D_k^\dagger$ (see Definition 2.8), and measure the $3m$ received qudits. Let the result of this measurement be $\delta_i \in F_q^{3m}$. We thus denote the effect of the measurement with this result by the projection $|\delta_i\rangle\langle\delta_i|$ conjugating the density matrix (for $\delta_i \in F_q^{3m}$). Of course we will sum over all the different values of δ_i. Next (in all rounds except round $L+1$), the verifier will send the prover the decoded measurement results $g(\delta_i)$ so the prover will be able to apply the Clifford correction $C_{g(\delta_i)}$, as written in equation 19. The verifier will then instruct the prover to apply the next set of Clifford gates $\tilde{Q}_i$ in the circuit. Since the verifier sent the prover $g(\delta_i)$, the prover's next attack can be dependent on this value.

We now provide the claim characterizing the state shared between the prover and the verifier, as a summation over all of the measurement results from previous rounds ($\Delta_{i-1} = (\delta_1, \ldots, \delta_{i-1})$):

Claim 6.1 (Polynomial QPIP State Evolution) *For $1 \leq i \leq L+1$, the state shared by the prover and the verifier at the start of round i can be written as:*

$$\frac{1}{2^m|\mathbb{P}_{m'}|} \sum_{\substack{z,x\in F_q^{|\mathcal{P}_i|} \\ k\in\{-1,1\}^m}} \tau_i(z,x,k) \otimes \sum_{\Delta_{i-1},z_1,x_1\in F_q^{|\mathcal{V}_i|}} W^i_{\Delta_{i-1},\hat{z},\hat{x},k}(\rho^k_{g(\Delta_{i-1})} \otimes \rho_{\mathcal{E}}) W^{i\dagger}_{\Delta_{i-1},\hat{z},\hat{x},k} \tag{150}$$

where $m' = 3mL + mn$,

$$\hat{z} = (z_1, z), \hat{x} = (x_1, x), \tag{151}$$

$$W^i_{\Delta_{i-1},\hat{z},\hat{x},k} = (|\Delta_{i-1}\rangle\langle\Delta_{i-1}| (D^\dagger_k)^{\otimes|M_i|}(Z^{z_1}X^{x_1})^\dagger_{\mathcal{V}_i} \otimes \mathcal{I}_{\mathcal{P}_i,E}) U_{g(\Delta_{i-1})}((Z^{z_1}X^{x_1})_{\mathcal{V}_i} \otimes (Z^zX^x)_{\mathcal{P}_i} \otimes \mathcal{I}_{\mathcal{E}}) \tag{152}$$

where $U_{g(\Delta_{i-1})}$ is a unitary operator dependent on $g(\Delta_i{-}1)$ and

$$\rho^k_{g(\Delta_{i-1})} = (\tilde{Q}_{i-1}\tilde{C}_{g(\delta_{i-1})} \cdots \tilde{Q}_1\tilde{C}_{g(\delta_1)}\tilde{Q}_0)\rho^k((\tilde{Q}_{i-1}\tilde{C}_{g(\delta_{i-1})} \cdots \tilde{Q}_1\tilde{C}_{g(\delta_1)}\tilde{Q}_0)^\dagger \tag{153}$$

for $\Delta_{i-1} = (\delta_1, \ldots, \delta_{i-1}) \in F_q^{|\mathcal{V}_i|}$, where ρ is the initial state on $3L + n$ qubits (consisting of L Toffoli states and an n qudit input state), ρ^k indicates authentication as described in equation 25, and $\rho_{\mathcal{E}}$ is the initial state of the prover's environment.

The projection $|\Delta_{i-1}\rangle\langle\Delta_{i-1}|$ denotes the verifier's measurement (it acts on register $\mathcal{V}_i$), part of which has been sent back to the prover in the form of $g(\Delta_{i-1})$ (hence the dependence of U and ρ^k on $g(\Delta_{i-1})$).

As a brief aside, recall that (as mentioned at the start of Section 6) one key difference between the Clifford and polynomial protocols is that the authenticated state throughout the polynomial protocol is not necessarily the correct authenticated state (i.e. the authentication of the state which would result by applying the Q-CIRCUIT instance U). This can be seen by observing the form of $\rho^k_{g(\Delta_{i-1})}$. Note that if the projection $|\Delta_{i-1}\rangle\langle\Delta_{i-1}|$ acted directly on the state, it would indeed be the correct state. However, the projection acts after the attack $U_{g(\Delta_{i-1})}$, which implies that if $U_{g(\Delta_{i-1})}$ acts non trivially on register $\mathcal{V}_i$, $\rho^k_{g(\Delta_{i-1})}$ will not necessarily be the correct authenticated state.

Before we proceed with the proof of soundness, we will write down the state at the start of round 1 as an example of how Claim 6.1 works. At the start of round 1, the state shared between the verifier and the prover is:

$$\frac{1}{2^m|\mathbb{P}_{m'}|} \sum_{\substack{z,x\in F_q^{|R_1|} \\ k\in\{-1,1\}^m}} \tau_1(z,x,k) \otimes V_0(Z^zX^x\tilde{Q}_0 \otimes \mathcal{I}_{\mathcal{E}})\rho^k \otimes \rho_{\mathcal{E}}(Z^zX^x\tilde{Q}_0 \otimes \mathcal{I}_{\mathcal{E}})^\dagger V_0^\dagger \tag{154}$$

where ρ^k is the initial state of the qudits sent to the prover and V_0 is the unitary attack of the prover applied at the end of round 0. Note that as the prover and verifier performed the Clifford operator $\tilde{Q}_0$, the verifier updated his initial Pauli keys to account for this operator (as described in Section 6.1). This is why the Pauli operator Z^zX^x acts after $\tilde{Q}_0$ on ρ^k in equation 154. As you can see, Claim 6.1 holds for $i = 1$.

We now proceed with the proof of soundness. Claim 6.1 implies that at the start of the final round, round $L+1$, the joint state of the prover's registers, $\mathcal{P}_{L+1}$ and the environment $\mathcal{E}$, and the verifier's registers, $\mathcal{V}_{L+1}$ and the key register containing the sign key and Pauli keys of qudits in $\mathcal{P}_{L+1}$, is:

$$\frac{1}{2^m|\mathbb{P}_{m'}|} \sum_{\substack{z,x\in F_q^{|\mathcal{P}_{L+1}|} \\ k\in\{-1,1\}^m}} \tau_{L+1}(z,x,k) \otimes \sum_{\Delta_L,z_1,x_1\in F_q^{|\mathcal{V}_{L+1}|}} W^{L+1}_{\Delta_L,\hat{z},\hat{x},k}(\rho^k_{g(\Delta_L)} \otimes \rho_{\mathcal{E}})(W^{L+1}_{\Delta_L,\hat{z},\hat{x}})^\dagger \tag{155}$$

As in previous rounds, the verifier decodes the final authenticated qudit sent by the prover, with both the Pauli and sign key. Let $\mathcal{F}$ denote the register containing the final authenticated qudit. Let $\mathcal{P}_{final}$ denote the register of the remaining authenticated qudits (this contains all qudits in $\mathcal{P}_{L+1}$ except those in register $\mathcal{F}$). Let $\mathcal{V}_{final} = \mathcal{F} \cup \mathcal{V}_{L+1}$ be the register containing all qudits sent to the verifier during the protocol.

Corollary 6.2 *The state shared between the prover and verifier after the decoding of register $\mathcal{F}$ is:*

$$\rho_{L+1} \stackrel{\text{def}}{=} \frac{1}{2^m|\mathbb{P}_{m'}|} \sum_{\substack{z,x\in F_q^{|\mathcal{P}_{final}|} \\ k\in\{-1,1\}^m}} \tau_{final}(z,x,k) \otimes \sum_{\substack{\Delta_L\in F_q^{|\mathcal{V}_{L+1}|} \\ z_1,x_1\in F_q^{|\mathcal{V}_{final}|}}} V_{\Delta_L,\hat{z},\hat{x},k}(\rho^k_{g(\Delta_L)}\otimes\rho_{\mathcal{E}})V^\dagger_{\Delta_L,\hat{z},\hat{x},k} \tag{156}$$

where $\hat{z}=(z_1,z)$, $\hat{x}=(x_1,x)$ and

$$V_{\Delta_L,\hat{z},\hat{x},k} = ((|\Delta_L\rangle\langle\Delta_L|\otimes\mathcal{I}_{\mathcal{F}})(D_k^\dagger)^{\otimes|\mathcal{V}_{final}|}(Z^{z_1}X^{x_1})^\dagger_{\mathcal{V}_{final}}\otimes\mathcal{I}_{\mathcal{P}_{final},\mathcal{E}})U_{g(\Delta_L)}((Z^{z_1}X^{x_1})_{\mathcal{V}_{final}}\otimes(Z^zX^x)_{\mathcal{P}_{final}}\otimes\mathcal{I}_{\mathcal{E}})$$

Proof: The only change between this state and equation 155 is the decoding of register F (by applying the inverse Pauli key and the signed polynomial decoding $D_k^\dagger$). Observe that $W^{L+1}_{\Delta_L,\hat{z},\hat{x}}$ and $V_{\Delta_L,\hat{z},\hat{x},k}$ differ only to the left of $U_{g(\delta)}$; nothing changes to the right. This is because in the state above, we are averaging over all Pauli operators acting on registers $\mathcal{V}_{final}$ and $\mathcal{P}_{final}$ and in equation 155 we are averaging over all Pauli operators acting on registers $\mathcal{V}_{L+1}$ and $\mathcal{P}_{L+1}$. To see that this is the same, observe that

$$\mathcal{V}_{final}\cup\mathcal{P}_{final} = \mathcal{V}_{L+1}\cup\mathcal{P}_{L+1} \tag{157}$$

To the left of $U_{g(\delta)}$, one additional register (F) is decoded first by the corresponding Pauli keys (which is reflected by the replacement of $(Z^{z_1}X^{x_1})^\dagger_{\mathcal{V}_{L+1}}$ with $(Z^{z_1}X^{x_1})^\dagger_{\mathcal{V}_{final}}$) and then by $D_k^\dagger$. The projection does not change (as indicated by $\mathcal{I}_{\mathcal{F}}$ in the projection) as we are only decoding register $\mathcal{F}$. □

Note that the verifier only holds the first key register (containing $\tau_{final}(z,x,k)$) and register $\mathcal{V}_{final}$. Recall that our goal is to show that for the Q-CIRCUIT instance U, if the first qudit of $U|0\rangle^{\otimes n}$ is 0 with probability $1-\gamma$, the verifier will either abort or not accept the final decoded measurement result with probability $\geq 1-(\gamma+\epsilon)$. For this purpose, we define the following projection on $\mathcal{V}_{final}=\mathcal{F}\cup\mathcal{V}_{L+1}$:

$$\hat{\Pi}_0 \stackrel{\text{def}}{=} (\mathcal{I}^{\otimes d+1}\otimes|0\rangle\langle 0|^{\otimes d})^{\otimes 3L}_{\mathcal{V}_{L+1}}\otimes(|1\rangle\langle 1|\otimes\mathcal{I}^{\otimes d}\otimes|0\rangle\langle 0|^{\otimes d})_{\mathcal{F}} \stackrel{\text{def}}{=} (\hat{\Pi}_0)_{\mathcal{V}_{L+1}}\otimes(\hat{\Pi}_0)_{\mathcal{F}} \tag{158}$$

The first term in the above projection describes the space of valid measurement results (i.e. strings which interpolate to low degree polynomials). The second term describes the space of a final qudit which is accepted and decodes to 1. We would like to show that

$$\text{Tr}(\hat{\Pi}_0\rho_{L+1}|_{\mathcal{V}_{final}}) \leq \gamma+\epsilon \tag{159}$$

In other words, if the decoded measurement result of the final qudit does not yield 1, the verifier rejects or aborts with high probability. Each block of m qudits in the register $\mathcal{V}_{L+1}$ is projected onto $\mathcal{I}^{\otimes d+1}\otimes|0\rangle\langle 0|^{\otimes d}$ as we are not looking for a specific decoded value in the measurement registers; we are only checking that the measurement results are valid.

Observe that in order to bound soundness, we only need to look at ρ_{L+1} on $\mathcal{V}_{final}$; the key register containing $\tau_{final}(z,x,k)$ was unnecessary. This is because the keys z,x acting on $\mathcal{P}_{final}$ will not be used; that register is never sent to the verifier. Also, the verifier no longer needs to remember the sign key, since it has already been used to decode the qudits in $\mathcal{V}_{final}$. Therefore, the verifier can trace out the first register containing $\tau_{final}(z,x,k)$.

Before continuing to prove equation 159, we can simplify $\rho_{L+1}|_{\mathcal{V}_{final}}$:

Claim 6.3 (Final State) *$\rho_{L+1}|_{\mathcal{V}_{final}}$ is equal to*

$$\frac{1}{2^m}\sum_{k\in\{-1,1\}^m}\sigma_k = \frac{1}{2^m}\sum_{k\in\{-1,1\}^m}\sum_{P\in\mathbb{P}_{|\mathcal{V}_{final}|}}\sigma_k^P \tag{160}$$

where

$$\sigma_k^P = \sum_{\Delta_L \in F_q^{|\mathcal{V}_{L+1}|}} \alpha_{P,g(\Delta_L)} \cdot ((|\Delta_L\rangle \langle \Delta_L| \otimes \mathcal{I}_{\mathcal{F}})(D_k^\dagger)^{\otimes |\mathcal{V}_{final}|} P) \sigma_{g(\Delta_L)}^k ((|\Delta_L\rangle \langle \Delta_L| \otimes \mathcal{I}_{\mathcal{F}})(D_k^\dagger)^{\otimes |\mathcal{V}_{final}|} P)^\dagger \quad (161)$$

and $Tr_{\mathcal{P}_{final}}(\rho_{g(\Delta_L)}^k) = \sigma_{g(\Delta_L)}^k$,

$$\alpha_{P,g(\Delta_L)} = \frac{1}{q^{|\mathcal{P}_{final}|}} Tr(U_{g(\Delta_L)}^P (\mathcal{I}_{\mathcal{P}_{final}} \otimes \rho_{\mathcal{E}})(U_{g(\Delta_L)}^P)^\dagger) \quad (162)$$

and

$$U_{g(\Delta_L)} = \sum_{P \in \mathbb{P}_{|\mathcal{V}_{final}|}} P \otimes U_{g(\Delta_L)}^P \quad (163)$$

Starting from the form of ρ_{L+1} in equation 156 in Corollary 6.2, we show this claim by first summing over z, x (this can be done since $\tau_{final}(z, x, k)$ is traced out), which has the effect of mixing register $\mathcal{P}_{final}$, as shown in the Pauli mixing lemma, Lemma 4.5 (which is analogous to Lemma 4.4 and also proven in Appendix B). Next, we can use z_1, x_1 to decohere (or remove all cross terms of) the part of $U_{g(\Delta_L)}$ acting on register $\mathcal{V}_{final}$ (by applying Lemma 5.1).

Now let's return to our goal of proving equation 159. With the above state simplification, we are now proving:

$$\frac{1}{2^m} \text{Tr}(\hat{\Pi}_0 (\sum_k \sigma_k) = \sum_{P \in \mathbb{P}_{\mathcal{V}_{final}}} \frac{1}{2^m} \text{Tr}(\hat{\Pi}_0 \sum_k \sigma_k^P) \leq \gamma + \epsilon \quad (164)$$

We first consider terms σ_k^P for which P is trivial (i.e. P consists only of Z and $\mathcal{I}$ operators). To prove the following claim, we first observe that trivial Pauli operators have no effect on measurement results, since they commute with the verifier's decoding process (application of $D_k^\dagger$ and the inverse Pauli keys). Given this observation, we can see that the prover's decoded final answer will be 0 with probability $1 - \gamma$ (as it should be), and therefore we can upper bound the projection of the state onto $\hat{\Pi}_0$:

Claim 6.4 (Trivial Deviation) *For trivial P,*

$$\frac{1}{2^m} Tr(\hat{\Pi}_0 \sum_k \sigma_k^P) \leq \frac{\gamma}{q^{3L}} \sum_{a \in F_q^{3L}} \alpha_{P,a}$$

Next, we consider terms σ_k^P for which P is non trivial. By using Lemma 5.2, which implies that P can produce a non zero trace (after the state is projected onto $\hat{\Pi}_0$) for at most 2 values of k, we show:

Claim 6.5 (Nontrivial Deviation) *For non trivial P,*

$$\frac{1}{2^m} Tr(\hat{\Pi}_0 \sum_k \sigma_k^P) \leq \frac{1}{q^{3L} 2^{m-1}} \sum_{a \in F_q^{3L}} \alpha_{P,a}$$

By combining both claims, we obtain:

$$\begin{aligned}
\frac{1}{2^m} \text{Tr}(\hat{\Pi}_0 \sum_k \sigma_k) &= \frac{1}{2^m} \sum_{P \in \mathbb{P}_{\mathcal{V}_{final}}} \text{Tr}(\hat{\Pi}_0 \sum_k \sigma_k^P) && (165) \\
&\leq \max(\gamma, \frac{1}{2^{m-1}}) \frac{1}{q^{3L}} \sum_{a \in F_q^{3L}} (\sum_{P \in \mathbb{P}_{\mathcal{V}_{final}}} \alpha_{P,a}) && (166) \\
&= \max(\gamma, \frac{1}{2^{m-1}}) && (167) \\
&\leq \gamma + \frac{1}{2^{m-1}} && (168)
\end{aligned}$$

The final equality follows because:

$$\sum_{P \in \mathbb{P}_{\mathcal{V}_{final}}} \alpha_{P,a} = 1$$

by Lemma 3.3. □

6.6 Proof of Claim 6.1 (Polynomial QPIP State Evolution)

Proof: We will prove this claim by induction. The base case (round 1) is proven already in Section 6.5, equation 154. We assume the claim holds in round i and show that it holds in round $i+1$. By the inductive hypothesis, we have the state shared by the prover and verifier in round i is:

$$\frac{1}{2^m|\mathbb{P}_{m'}|} \sum_{\substack{z,x \in F_q^{|\mathcal{P}_i|} \\ k \in \{-1,1\}^m}} \tau_i(z,x,k) \otimes \sum_{\Delta_{i-1},z_1,x_1 \in F_q^{|\mathcal{V}_i|}} W^i_{\Delta_{i-1},\hat{z},\hat{x},k} (\rho^k_{g(\Delta_{i-1})} \otimes \rho_{\mathcal{E}}) W^{i\dagger}_{\Delta_{i-1},\hat{z},\hat{x},k} \tag{169}$$

where $\hat{z} = (z_1, z)$, $\hat{x} = (x_1, x)$ and

$$W^i_{\Delta_{i-1},\hat{z},\hat{x},k} = (|\Delta_{i-1}\rangle \langle\Delta_{i-1}| (D^\dagger_k)^{\otimes|\mathcal{V}_i|} (Z^{z_1}X^{x_1})^\dagger_{\mathcal{V}_i} \otimes \mathcal{I}_{\mathcal{P}_i,E}) U_{g(\Delta_{i-1})} ((Z^{z_1}X^{x_1})_{\mathcal{V}_i} \otimes (Z^z X^x)_{\mathcal{P}_i} \otimes \mathcal{I}_{\mathcal{E}})$$

Recall that the verifier holds register $\mathcal{V}_i$ and registers $\mathcal{P}_i$ and E are held by the prover. When the prover measures and sends the verifier his measurement results, the verifier decodes them with both the Pauli keys and the sign key (as in equation 147) to obtain $\delta_i \in F_q^{3m}$. The shared state at this point is:

$$\frac{1}{2^m|\mathbb{P}_{m'}|} \sum_{\substack{z,x \in F_q^{|R_{i+1}|} \\ k \in \{-1,1\}^m}} \tau_{i+1}(z,x,k) \otimes \sum_{\Delta_i,z_1,x_1 \in F_q^{|M_{i+1}|}} T_{\Delta_i,\hat{z},\hat{x},k} (\rho^k_{g(\Delta_{i-1})} \otimes \rho_{\mathcal{E}}) T^\dagger_{\Delta_i,\hat{z},\hat{x},k} \tag{170}$$

where

$$T_{\Delta_i,\hat{z},\hat{x},k} = (|\Delta_i\rangle \langle\Delta_i| (D^\dagger_k)^{\otimes|M_{i+1}|} (Z^{z_1}X^{x_1})^\dagger_{M_{i+1}} \otimes \mathcal{I}_{R_{i+1},E}) U_{g(\Delta_{i-1})} ((Z^{z_1}X^{x_1})_{M_{i+1}} \otimes (Z^z X^x)_{R_{i+1}} \otimes \mathcal{I}_{\mathcal{E}})$$

Note that $\Delta_i = (\delta_i, \Delta_{i-1})$, where $\delta_i \in F_q^{3m}$ (δ_i is the measurement result obtained in round i) and $\Delta_{i-1} \in F_q^{|\mathcal{V}_i|}$ (measurement results from previous rounds). The key difference here is that we have taken $3m$ qudits from register R_i and added them to register $\mathcal{V}_i$ to create registers R_{i+1} and M_{i+1}. We have also removed the Pauli keys corresponding to the newly measured qudits from the first register; the verifier traces out these keys after decoding as he no longer needs them.

The remainder of round i consists of the prover and verifier performing the Clifford gate $\tilde{Q}_i \tilde{C}_{g(\delta_i)}$. To show that the shared state in round $i+1$ is of the form described in Claim 6.1, we need to replace $\rho^k_{g(\Delta_{i-1})}$ with $\rho^k_{g(\Delta_i)}$ in equation 170. This can be done by determing how the application of $\tilde{Q}_i \tilde{C}_{g(\delta_i)}$ changes the state. Recall that:

$$\begin{aligned} \rho^k_{g(\Delta_i)} &= (\tilde{Q}_i \tilde{C}_{g(\delta_i)} \cdots \tilde{Q}_1 \tilde{C}_{g(\delta_1)} \tilde{Q}_0) \rho^k ((\tilde{Q}_i \tilde{C}_{g(\delta_i)} \cdots \tilde{Q}_1 \tilde{C}_{g(\delta_1)} \tilde{Q}_0)^\dagger & (171) \\ &= \tilde{Q}_i \tilde{C}_{g(\delta_i)} \rho^k_{g(\Delta_{i-1})} (\tilde{Q}_i \tilde{C}_{g(\delta_i)})^\dagger & (172) \end{aligned}$$

In order to replace $\rho^k_{g(\Delta_{i-1})}$ with $\rho^k_{g(\Delta_i)}$, we need to commute $\tilde{Q}_i \tilde{C}_{g(\delta_i)}$ past $T_{\Delta_i,\hat{z},\hat{x},k}$. First observe that $\tilde{Q}_i \tilde{C}_{g(\delta_i)}$ operates on the register held by the prover, R_{i+1}, and therefore commutes with operators acting on register M_{i+1}. However, it does not commute with $U_{g(\Delta_{i-1})}$. To take care of this issue, observe that:

$$\tilde{Q}_i \tilde{C}_{g(\delta_i)} U_{g(\Delta_{i-1})} = (\tilde{Q}_i \tilde{C}_{g(\delta_i)}) U_{g(\Delta_{i-1})} (\tilde{Q}_i \tilde{C}_{g(\delta_i)})^\dagger (\tilde{Q}_i \tilde{C}_{g(\delta_i)}) \tag{173}$$

Now the rightmost part of the above expression, $\tilde{Q}_i\tilde{C}_{g(\delta_i)}$, is acting on the Pauli operator $(Z^zX^x)_{R_{i+1}}$ which is acting on $\rho^k_{g(\Delta_{i-1})}$. If the verifier updates his Pauli keys for register R_{i+1} (as is part of the protocol for performing a Clifford operation, described in Section 6.1), $\tilde{Q}_i\tilde{C}_{g(\delta_i)}$ can be commuted past the Pauli operator:

$$\tilde{Q}_i\tilde{C}_{g(\delta_i)}Z^zX^x = \tilde{Q}_i\tilde{C}_{g(\delta_i)}Z^zX^x(\tilde{Q}_i\tilde{C}_{g(\delta_i)})^\dagger\tilde{Q}_i\tilde{C}_{g(\delta_i)} \tag{174}$$

As described in more detail in Section 6.1, applying the Clifford operator $\tilde{Q}_i\tilde{C}_{g(\delta_i)}$ involves both the prover applying the operator to the authenticated states and the verifier updating his Pauli keys from Z^zX^x to $\tilde{Q}_i\tilde{C}_{g(\delta_i)}Z^zX^x(\tilde{Q}_i\tilde{C}_{g(\delta_i)})^\dagger$ (this is a Pauli since $\tilde{Q}_i\tilde{C}_{g(\delta_i)}$ is a Clifford).

Now $\tilde{Q}_i\tilde{C}_{g(\delta_i)}$, is acting directly on $\rho^k_{g(\Delta_{i-1})}$ so we have:

$$\rho^k_{g(\Delta_i)} = \tilde{Q}_i\tilde{C}_{g(\delta_i)}\rho^k_{g(\Delta_{i-1})}\tilde{C}^\dagger_{g(\delta_i)}\tilde{Q}^\dagger_i \tag{175}$$

Note that this is still a state encoded with the signed polynomial code (hence the superscript k), since the Clifford operators are logical operators on the signed polynomial encoding. Finally, the prover can apply another attack $V_{g(\Delta_i)}$. Note that this attack acts only on the registers held by the prover (R_{i+1} and E) and therefore can be shifted past operators acting on register M_{i+1} in $T_{\Delta_i,\hat{z},\hat{x},k}$. We now set:

$$U_{g(\Delta_i)} = V_{g(\Delta_i)}\tilde{Q}_i\tilde{C}_{g(\delta_i)}U_{g(\Delta_{i-1})}\tilde{C}^\dagger_{g(\delta_i)}\tilde{Q}^\dagger_i \tag{176}$$

The prover's state at the end of the round is then:

$$\frac{1}{2^m|\mathbb{P}_{m'}|}\sum_{\substack{z,x\in F_q^{|R_{i+1}|}\\ k\in\{-1,1\}^m}}\tau_{i+1}(z,x,k)\otimes\sum_{\Delta_i,z_1,x_1\in F_q^{|M_{i+1}|}}W^{i+1}_{\Delta_i,\hat{z},\hat{x},k}(\rho^k_{g(\Delta_i)}\otimes\rho_{\mathcal{E}})(W^{i+1}_{\Delta_i,\hat{z},\hat{x},k})^\dagger \tag{177}$$

where

$$W^{i+1}_{\Delta_i,\hat{z},\hat{x},k} = (|\Delta_i\rangle\langle\Delta_i|(D^\dagger_k)^{\otimes|M_{i+1}|}(Z^{z_1}X^{x_1})^\dagger\otimes\mathcal{I}_{R_{i+1},E})U_{g(\Delta_i)}((Z^{z_1}X^{x_1})_{M_{i+1}}\otimes(Z^zX^x)_{R_{i+1}}\otimes\mathcal{I}_{\mathcal{E}})$$

□

6.7 Proof of Claim 6.3 (Final State)

Proof: Recall that we start with the state given in equation 156 in Corollary 6.2:

$$\rho_{L+1}\stackrel{\text{def}}{=}\frac{1}{2^m|\mathbb{P}_{m'}|}\sum_{\substack{z,x\in F_q^{|\mathcal{P}_{final}|}\\ k\in\{-1,1\}^m}}\tau_{final}(z,x,k)\otimes\sum_{\substack{\Delta_L\in F_q^{|\mathcal{V}_{L+1}|}\\ z_1,x_1\in F_q^{|\mathcal{V}_{final}|}}}V_{\Delta_L,\hat{z},\hat{x},k}(\rho^k_{g(\Delta_L)}\otimes\rho_{\mathcal{E}})V^\dagger_{\Delta_L,\hat{z},\hat{x},k} \tag{178}$$

where $\hat{z}=(z_1,z)$, $\hat{x}=(x_1,x)$ and

$$V_{\Delta_L,\hat{z},\hat{x},k} = ((|\Delta_L\rangle\langle\Delta_L|\otimes\mathcal{I}_{\mathcal{F}})(D^\dagger_k)^{\otimes|\mathcal{V}_{final}|}(Z^{z_1}X^{x_1})^\dagger_{\mathcal{V}_{final}}\otimes\mathcal{I}_{\mathcal{P}_{final},\mathcal{E}})U_{g(\Delta_L)}((Z^{z_1}X^{x_1})_{\mathcal{V}_{final}}\otimes(Z^zX^x)_{\mathcal{P}_{final}}\otimes\mathcal{I}_{\mathcal{E}})$$

Our goal is to determine the form of the state after tracing out the first register (the key register) and registers $\mathcal{P}_{final}$ and E. We begin by tracing out the key register, which allows us to sum over $z,x\in F_q^{|\mathcal{P}_{final}|}$. We are also allowed to sum over k, but we will keep k fixed while we simplify the state. The state can then be written as:

$$\sum_{z,x\in F_q^{|\mathcal{P}_{final}|}}\sum_{\substack{\Delta_L\in F_q^{|\mathcal{V}_{L+1}|}\\ z_1,x_1\in F_q^{|\mathcal{V}_{final}|}}}V_{\Delta_L,\hat{z},\hat{x},k}(\rho^k_{g(\Delta_L)}\otimes\rho_{\mathcal{E}})V^\dagger_{\Delta_L,\hat{z},\hat{x},k} \tag{179}$$

By Lemma 4.5, this has the effect of mixing register $\mathcal{P}_{final}$. The state is now:

$$\frac{1}{q^{|\mathcal{P}_{final}|}} \sum_{\substack{\Delta_L \in F_q^{|\mathcal{V}_{L+1}|} \\ z_1, x_1 \in F_q^{|\mathcal{V}_{final}|}}} V'_{\Delta_L, z_1, x_1, k} (\sigma^k_{g(\Delta_L)} \otimes \mathcal{I}_{\mathcal{P}_{final}} \otimes \rho_{\mathcal{E}}) V'^{\dagger}_{\Delta_L, z_1, x_1, k} \tag{180}$$

where

$$V'_{\Delta_L, z_1, x_1, k} = ((|\Delta_L\rangle\langle\Delta_L| \otimes \mathcal{I}_{\mathcal{F}})(D_k^{\dagger})^{\otimes |\mathcal{V}_{final}|}(Z^{z_1}X^{x_1})^{\dagger} \otimes \mathcal{I}_{\mathcal{P}_{final},\mathcal{E}}) U_{g(\Delta_L)} (Z^{z_1}X^{x_1} \otimes \mathcal{I}_{\mathcal{P}_{final},\mathcal{E}})$$

and

$$\mathrm{Tr}_{\mathcal{P}_{final}}(\rho^k_{g(\Delta_L)}) = \sigma^k_{g(\Delta_L)}$$

Next, we observe that the Pauli encoding/decoding of $Z^{z_1}X^{x_1}$ on register $\mathcal{V}_{final}$ has the effect of decohering (removing cross terms of) the part of $U_{g(\Delta_L)}$ that is acting on $\mathcal{V}_{final}$, as shown in Lemma 5.1.

Applying the lemma with:

$$U = U_{g(\Delta_L)} = \sum_{P \in \mathbb{P}_{|\mathcal{V}_{final}|}} P \otimes U^P_{g(\Delta_L)} \tag{181}$$

we can simplify the prover's decoded state to:

$$\sum_{\substack{\Delta_L \in F_q^{|\mathcal{V}_{L+1}|} \\ P \in \mathbb{P}_{|\mathcal{V}_{final}|}}} ((|\Delta_L\rangle\langle\Delta_L| \otimes \mathcal{I}_{\mathcal{F}})(D_k^{\dagger})^{\otimes |\mathcal{V}_{final}|} P) \sigma^k_{g(\Delta_L)} ((|\Delta_L\rangle\langle\Delta_L| \otimes \mathcal{I}_{\mathcal{F}}) D_k^{\otimes |\mathcal{V}_{final}|} P)^{\dagger} \otimes U_P^{g(\Delta_L)} (\mathcal{I}_{\mathcal{P}_{final}} \otimes \rho_{\mathcal{E}}) (U_P^{g(\Delta_L)})^{\dagger} \tag{182}$$

where the above state also has a factor of $\frac{1}{q^{|\mathcal{P}_{final}|}}$.

We trace out registers $\mathcal{P}_{final}$ and E since the verifier will not look at these registers:

$$\sigma_k \quad = \quad \sum_{\substack{\Delta_L \in F_q^{|\mathcal{V}_{L+1}|} \\ P \in \mathbb{P}_{|\mathcal{V}_{final}|}}} \alpha_{P,\Delta_L} \cdot ((|\Delta_L\rangle\langle\Delta_L| \otimes \mathcal{I}_{\mathcal{F}}) D_k^{\otimes |\mathcal{V}_{final}|} P) \sigma^k_{g(\Delta_L)} ((|\Delta_L\rangle \ \langle\Delta_L| \otimes \mathcal{I}_{\mathcal{F}}) D_k^{\otimes |\mathcal{V}_{final}|} P)^{\dagger}$$

where $\alpha_{P,\Delta_L} = \frac{1}{q^{|\mathcal{P}_{final}|}} \mathrm{Tr}(U_P^{g(\Delta_L)} (\mathcal{I}_{\mathcal{P}_{final}} \otimes \rho_{\mathcal{E}}) (U_P^{g(\Delta_L)})^{\dagger})$. □

6.8 Proofs of Claim 6.4 and Claim 6.5 (Trivial and Nontrivial Deviation)

6.8.1 Necessary Claims

For both proofs, we require the three following claims. The first is regarding the state $\sigma^k_a = \mathrm{Tr}_{\mathcal{P}_{final}}(\rho^k_a)$ (for $a \in F_q^{3L}$), where ρ^k_a is defined in Claim 6.1 (in equation 153) and σ^k_a is first defined in Claim 6.3. The claim below considers the unauthenticated version of the state (σ_a). In other words, if σ'_a is σ_a with $m-1$ auxiliary 0 qudits appended to each individual qudit, then:

$$\sigma^k_a = E_k \sigma'_a E_k^{\dagger} \tag{183}$$

This equality follows from the definition of the encoding circuit (Definition 2.7).

Claim 6.6 *For $a \in F_q^{3L}$ and σ_a as defined in Claim 6.3 we claim that*

$$\sigma_a = \frac{1}{q^{3L}} \sum_{l, l' \in F_q^{3L}} |l\rangle\langle l'| \otimes Tr_{\mathcal{P}_{final}}(|\psi\rangle_{a,l}\langle\psi|_{a,l'}) \tag{184}$$

where $|\psi\rangle_{a,a} = U|0\rangle^{\otimes n}$.

The second claim involves conjugation properties of the encoding circuit E_k (see Definition 2.7) with respect to trivial Pauli operators:

Claim 6.7 *For a trivial Pauli operator $P \in \mathbb{P}_{|\mathcal{V}_{final}|}$,*

$$(\Pi_{G_a} \otimes (\hat{\Pi}_0)_{\mathcal{F}})(E_k^\dagger)^{\otimes|\mathcal{V}_{final}|} P E_k^{\otimes|\mathcal{V}_{final}|} = (E_k^\dagger)^{\otimes|\mathcal{V}_{final}|} P E_k^{\otimes|\mathcal{V}_{final}|}(\Pi_{G_a} \otimes (\hat{\Pi}_0)_{\mathcal{F}}) \tag{185}$$

where $(\hat{\Pi}_0)_{\mathcal{F}}$ is defined in equation 158 as

$$(\hat{\Pi}_0)_{\mathcal{F}} = |1\rangle\langle 1| \otimes \mathcal{I}^{\otimes d} |0\rangle\langle 0|^{\otimes d} \tag{186}$$

and for $a = (a(1), \ldots, a(3L)) \in F_q^{3L}$

$$\Pi_{G_a} = (|a(1)\rangle\langle a(1)| \otimes \mathcal{I}^{\otimes d} \otimes |0\rangle\langle 0|^{\otimes d}) \otimes \cdots \otimes (|a(3L)\rangle\langle a(3L)| \otimes \mathcal{I}^{\otimes d} \otimes |0\rangle\langle 0|^{\otimes d}) \tag{187}$$

The final claim simplifies the expression for $\text{Tr}(\hat{\Pi}_0 \sigma_k^P)$:

Claim 6.8 *For all $P = Z^z X^x \in \mathbb{P}_{|\mathcal{V}_{final}|}$, and for σ_k^P as defined in Claim 6.3 (equation 161),*

$$Tr(\hat{\Pi}_0 \sigma_k^P) = \sum_{a \in F_q^{3L}} \alpha_{P,a} \cdot Tr((\Pi_{G_a} \otimes (\hat{\Pi}_0)_{\mathcal{F}})(E_k^\dagger)^{\otimes|\mathcal{V}_{final}|} X^x \sigma_a^k (X^x)^\dagger (E_k)^{\otimes|\mathcal{V}_{final}|}) \tag{188}$$

where $(\hat{\Pi}_0)_{\mathcal{F}}$ is defined in equation 158 as

$$(\hat{\Pi}_0)_{\mathcal{F}} = |1\rangle\langle 1| \otimes \mathcal{I}^{\otimes d} |0\rangle\langle 0|^{\otimes d} \tag{189}$$

and for $a = (a(1), \ldots, a(3L)) \in F_q^{3L}$

$$\Pi_{G_a} = (|a(1)\rangle\langle a(1)| \otimes \mathcal{I}^{\otimes d} \otimes |0\rangle\langle 0|^{\otimes d}) \otimes \cdots \otimes (|a(3L)\rangle\langle a(3L)| \otimes \mathcal{I}^{\otimes d} \otimes |0\rangle\langle 0|^{\otimes d}) \tag{190}$$

We now proceed to proving Claims 6.4 and 6.5, and then we prove the claims listed above.

6.8.2 Proof of Claim 6.4 (Trivial Deviation)

*P*roof of **Claim 6.4**: Our goal in this proof is to show

$$\frac{1}{2^m}\text{Tr}(\hat{\Pi}_0 \sum_k \sigma_k^P) \leq \frac{\gamma}{q^{3L}} \sum_{a \in F_q^{3L}} \alpha_{P,a} \tag{191}$$

for a trivial Pauli operator P acting on $\mathcal{V}_{final}$. We will show that for all k,

$$\text{Tr}(\hat{\Pi}_0 \sigma_k^P) \leq \frac{\gamma}{q^{3L}} \sum_{a \in F_q^{3L}} \alpha_{P,a} \tag{192}$$

By Claim 6.8 (and by the fact that P is a trivial Pauli operator and therefore has no X operator), we have

$$\text{Tr}(\hat{\Pi}_0 \sigma_k^P) = \sum_{a \in F_q^{3L}} \alpha_{P,a} \cdot \text{Tr}((\Pi_{G_a} \otimes (\hat{\Pi}_0)_{\mathcal{F}})(E_k^\dagger)^{\otimes|\mathcal{V}_{final}|} \sigma_a^k (E_k)^{\otimes|\mathcal{V}_{final}|}) \tag{193}$$

where (as defined in equation 187) for $a = (a(1), \ldots, a(3L)) \in F_q^{3L}$

$$\Pi_{G_a} = (|a(1)\rangle\langle a(1)| \otimes \mathcal{I}^{\otimes d} \otimes |0\rangle\langle 0|^{\otimes d}) \otimes \cdots \otimes (|a(3L)\rangle\langle a(3L)| \otimes \mathcal{I}^{\otimes d} \otimes |0\rangle\langle 0|^{\otimes d}) \tag{194}$$

and (as defined in equation 158):

$$\hat{\Pi}_0 = (\mathcal{I}^{\otimes d+1} \otimes |0\rangle\langle 0|^{\otimes d})^{\otimes 3L}_{\mathcal{V}_{L+1}} \otimes (|1\rangle\langle 1| \otimes \mathcal{I}^{\otimes d} \otimes |0\rangle\langle 0|^{\otimes d})_{\mathcal{F}} = (\hat{\Pi}_0)_{\mathcal{V}_{L+1}} \otimes (\hat{\Pi}_0)_{\mathcal{F}} \tag{195}$$

We note that σ_a^k is the density matrix σ_a encoded with the signed polynomial code; i.e. if σ'_a is the density matrix σ_a with $m-1$ auxiliary 0 qudits appended to each individual qudit, we have:

$$\sigma_a^k = E_k^{\otimes|\mathcal{V}_{final}|}\sigma'_a(E_k^{\otimes|\mathcal{V}_{final}|})^\dagger \tag{196}$$

It follows that

$$\mathrm{Tr}(\hat{\Pi}_0\sigma_k^P) \quad = \quad \sum_{a\in F_q^{3L}} \alpha_{P,a}\cdot \mathrm{Tr}((\Pi_{G_a}\otimes(\hat{\Pi}_0)_{\mathcal{F}})\sigma'_a) \tag{197}$$

Observe that the projection $\Pi_{G_a}\otimes(\hat{\Pi}_0)_{\mathcal{F}}$ does not alter the auxiliary 0 qudits; it acts as $\mathcal{I}^{\otimes d}\otimes|0\rangle\langle 0|^{\otimes d}$ on each set of $m-1$ auxiliary qudits. Therefore, we can trace out all of the auxiliary qubits (and also remove the corresponding operators from the projections). Tracing out the auxiliary qudits from σ'_a simply results in σ_a. Π_{G_a} can be replaced by $|a\rangle\langle a|$, and $(\hat{\Pi}_0)_{\mathcal{F}}$ can be replaced by $|1\rangle\langle 1|$. The resulting expression is:

$$\mathrm{Tr}(\hat{\Pi}_0\sigma_k^P) = \sum_{a\in F_q^{3L}} \alpha_{P,a}\cdot \mathrm{Tr}(|a1\rangle\langle a1|\,\sigma_a) \tag{198}$$

Claim 6.6 implies that the state σ_a can be written as

$$\sigma_a = \frac{1}{q^{3L}}\sum_{l,l'\in F_q^{3L}} |l\rangle\langle l'| \otimes \mathrm{Tr}_{\mathcal{P}_{final}}(|\psi\rangle_{a,l}\langle\psi|_{a,l'}) \tag{199}$$

where $|\psi\rangle_{a,a} = U|0\rangle^{\otimes n}$. Given this information about σ_a, we can continue:

$$\begin{aligned}
\mathrm{Tr}(\hat{\Pi}_0\sigma_k^P) \quad &= \quad \sum_{a\in F_q^{3L}} \alpha_{P,a}\cdot \mathrm{Tr}(|a1\rangle\langle a1|\,\sigma_a) && (200)\\
&= \quad \frac{1}{q^{3L}}\sum_{a,l,l'\in F_q^{3L}} \alpha_{P,a}\cdot \mathrm{Tr}(|a\rangle\langle a|\,|l\rangle\langle l'|)\mathrm{Tr}(|1\rangle\langle 1|\,(\mathrm{Tr}_{\mathcal{P}_{final}}(|\psi\rangle_{a,l}\langle\psi|_{a,l'}))) && (201)\\
&= \quad \frac{1}{q^{3L}}\sum_{a\in F_q^{3L}} \alpha_{P,a}\cdot \mathrm{Tr}(|1\rangle\langle 1|\,(\mathrm{Tr}_{\mathcal{P}_{final}}(U|0\rangle\langle 0|^{\otimes n}U^\dagger))) && (202)\\
&\leq \quad \frac{\gamma}{q^{3L}}\sum_{a\in F_q^{3L}} \alpha_{P,a} && (203)
\end{aligned}$$

The last inequality follows because measuring $U|0\rangle$ results in 0 with probability $\geq 1-\gamma$ and 1 otherwise.

Remark 6.1 *Note that if $U|0\rangle$ resulted in 0 with probability exactly $1-\gamma$, the last inequality would be replaced by an equality (which would replace the inequality in the statement of the claim with an equality). Also note that if in equation 202 the projection $|1\rangle\langle 1|$ was replaced by $\mathcal{I}$, the next line would be the same, except γ would be replaced by 1 and the inequality by an equality. These two facts will be useful in the proof of Claim 8.2, which is required for the proof of the polynomial version of Corollary 1.11 (Corollary 1.12)*

□

6.8.3 Proof of Claim 6.5 (Nontrivial Deviation)

We now prove Claim 6.5.

Proof of **Claim 6.5**: Our goal in this proof is to show that for a non trivial Pauli operator $P = Z^z X^x \in \mathbb{P}_{|\mathcal{V}_{final}|}$,

$$\frac{1}{2^m}\text{Tr}(\hat{\Pi}_0 \sum_k \sigma_k^P) \leq \frac{1}{q^{3L}2^{m-1}} \sum_{a \in F_q^{3L}} \alpha_{P,a} \tag{204}$$

To do this, we will show that at most 2 terms in the above sum over k can be non zero; each of those terms can be at most

$$\frac{1}{2^m q^{3L}} \sum_{a \in F_q^{3L}} \alpha_{P,a} \tag{205}$$

The claim follows. We begin by using Claim 6.8 to write:

$$\text{Tr}(\hat{\Pi}_0 \sigma_k^P) = \sum_{a \in F_q^{3L}} \alpha_{P,a} \cdot \text{Tr}((\Pi_{G_a} \otimes (\hat{\Pi}_0)_{\mathcal{F}})(E_k^\dagger)^{\otimes |\mathcal{V}_{final}|} X^x \sigma_a^k (X^x)^\dagger (E_k)^{\otimes |\mathcal{V}_{final}|}) \tag{206}$$

where (as defined in equation 187) for $a = (a(1), \ldots, a(3L)) \in F_q^{3L}$

$$\Pi_{G_a} = (|a(1)\rangle \langle a(1)| \otimes \mathcal{I}^{\otimes d} \otimes |0\rangle \langle 0|^{\otimes d}) \otimes \cdots \otimes (|a(3L)\rangle \langle a(3L)| \otimes \mathcal{I}^{\otimes d} \otimes |0\rangle \langle 0|^{\otimes d}) \tag{207}$$

and (as defined in equation 158):

$$\hat{\Pi}_0 = (\mathcal{I}^{\otimes d+1} \otimes |0\rangle \langle 0|^{\otimes d})_{\mathcal{V}_{L+1}}^{\otimes 3L} \otimes (|1\rangle \langle 1| \otimes \mathcal{I}^{\otimes d} \otimes |0\rangle \langle 0|^{\otimes d})_{\mathcal{F}} = (\hat{\Pi}_0)_{\mathcal{V}_{L+1}} \otimes (\hat{\Pi}_0)_{\mathcal{F}} \tag{208}$$

Note that the projection $\Pi_{G_a} \otimes (\hat{\Pi}_0)_{\mathcal{F}}$ includes the projection of each block of m qudits onto $\mathcal{I}^{\otimes d+1} \otimes |0\rangle \langle 0|^{\otimes d}$; it can be written as:

$$\Pi_{G_a} \otimes (\hat{\Pi}_0)_{\mathcal{F}} = \hat{\Pi}_0^L \hat{\Pi}_0^A \tag{209}$$

where

$$\hat{\Pi}_0^A = (\mathcal{I}^{\otimes d+1} \otimes |0\rangle \langle 0|^{\otimes d})^{\otimes |\mathcal{V}_{final}|} \tag{210}$$

Intuitively, this implies that the non trivial Pauli operator P must preserve the authenticated state (up to trivial operators) on every block of m qudits in order end up in the subspace defined by $\hat{\Pi}_0^A$. Using similar reasoning as used in the proof of Lemma 5.2, we should be able to say that P can only do this for at most 2 sign keys at a time. This intuition is formalized in Corollary 6.9, which follows from Lemma 5.2 and is proven immediately after this proof:

Corollary 6.9 *For a non trivial Pauli operator $X^x \in \mathbb{P}_{tm}$ and a density matrix σ on m qudits, there exist at most 2 sign keys $k \in \{-1, 1\}^m$ (which are the same regardless of σ) for which the following expression*

$$(\mathcal{I}^{\otimes d+1} \otimes |0\rangle \langle 0|^{\otimes d})^{\otimes tm} (E_k^\dagger)^{\otimes t} X^x \sigma^k (X^x)^\dagger E_k^{\otimes t} (\mathcal{I}^{\otimes d+1} \otimes |0\rangle \langle 0|^{\otimes d})^{\otimes tm} \tag{211}$$

can be non zero. If $X^x = X^{x_1} \otimes \cdots \otimes X^{x_t}$ for $x_i \in F_q^m$ and the expression above is non zero, X^{x_i} must be k-correlated for all i.

This corollary implies that $\text{Tr}(\hat{\Pi}_0^A \sigma_k^P)$ (where $\hat{\Pi}_0^A$ is defined in equation 210) is non zero for at most 2 sign keys. Fix one sign key for which $\text{Tr}(\hat{\Pi}_0^A \sigma_k^P)$ is non zero. We will now simplify the expression in equation 206 for this fixed sign key k.

$$\text{Tr}(\hat{\Pi}_0 \sigma_k^P) \quad = \quad \sum_{a \in F_q^{3L}} \alpha_{P,a} \cdot \text{Tr}((\Pi_{G_a} \otimes (\hat{\Pi}_0)_{\mathcal{F}})(E_k^\dagger)^{\otimes |\mathcal{V}_{final}|} X^x \sigma_a^k (X^x)^\dagger (E_k)^{\otimes |\mathcal{V}_{final}|}) \tag{212}$$

Now due to Corollary 6.9, we know that X^x is k-correlated (see Definition 5.4). Because X^x maps an authenticated state to a different authenticated state, it is by definition equal to a logical Pauli operator $\tilde{X}^{x_k}$, so it maps σ_a^k to $\sigma_{a,x_k}^k = (X^{x_k}\sigma_a(X^{x_k})^\dagger)^k$:

$$\begin{aligned} \dots &= \sum_{a\in F_q^{3L}} \alpha_{P,a}\cdot \mathrm{Tr}((\Pi_{G_a}\otimes(\hat{\Pi}_0)_{\mathcal{F}})(E_k^\dagger)^{\otimes|\mathcal{V}_{final}|}\sigma_{a,x_k}^k(E_k)^{\otimes|\mathcal{V}_{final}|}) \quad (213)\\ &= \sum_{a\in F_q^{3L}} \alpha_{P,a}\cdot \mathrm{Tr}((\Pi_{G_a}\otimes(\hat{\Pi}_0)_{\mathcal{F}}))\sigma'_{a,x_k}(\Pi_{G_a}\otimes\mathcal{I}_{\mathcal{F}})) \quad (214)\end{aligned}$$

where the equality follows because σ_{a,x_k}^k is the density matrix σ_{a,x_k} encoded with the signed polynomial code, and σ'_{a,x_k} is the density matrix σ_{a,x_k} with $m-1$ auxiliary 0 qudits appended to each qudit of σ_{a,x_k}. Note that the projection $\Pi_{G_a}\otimes(\hat{\Pi}_0)_{\mathcal{F}}$ does not alter the auxiliary 0 qudits, as it acts on each set of $m-1$ 0 qudits as $\mathcal{I}^{\otimes d}\otimes|0\rangle\langle 0|^{\otimes d}$. Then we can trace out all the auxiliary 0 qudits and also remove them from the projection $\Pi_{G_a}\otimes(\hat{\Pi}_0)_{\mathcal{F}}$. The projection is now simply $|a\rangle\langle a|\otimes|1\rangle\langle 1|$ and we have:

$$\mathrm{Tr}(\hat{\Pi}_0\sigma_k^P) = \frac{1}{2^m}\sum_{a\in F_q^{3L}}\alpha_{P,a}\cdot\mathrm{Tr}((|a1\rangle\langle a1|)X^{x_k}\sigma_a(X^{x_k})^\dagger) \quad (215)$$

Now we use Claim 6.6 to write σ_a as

$$\sigma_a = \frac{1}{q^{3L}}\sum_{l,l'\in F_q^{3L}}|l\rangle\langle l'|\otimes\mathrm{Tr}_{\mathcal{P}_{final}}(|\psi\rangle_{a,l}\langle\psi|_{a,l'}) \quad (216)$$

where $|\psi\rangle_{a,a} = U|0\rangle^{\otimes n}$. Let $\sigma_{a,l,l'} = \mathrm{Tr}_{\mathcal{P}_{final}}(|\psi\rangle_{a,l}\langle\psi|_{a,l'})$ and let $X^{x_k} = X^{x_k}_{\mathcal{V}_{L+1}}\otimes X^{x_k}_{\mathcal{F}}$. Now we have:

$$\begin{aligned}\mathrm{Tr}(\hat{\Pi}_0\sigma_k^P) &= \frac{1}{q^{3L}}\sum_{a,l,l'\in F_q^{3L}}\alpha_{P,a}\cdot\mathrm{Tr}((|a1\rangle\langle a1|)X^{x_k}(|l\rangle\langle l'|\otimes\sigma_{a,l,l'})(X^{x_k})^\dagger) \quad (217)\\ &= \frac{1}{q^{3L}}\sum_{a,l,l'\in F_q^{3L}}\alpha_{P,a}\cdot\mathrm{Tr}(|a\rangle\langle a|(X^{x_k}_{\mathcal{V}_{L+1}}|l\rangle\langle l'|(X^{x_k}_{\mathcal{V}_{L+1}})^\dagger))\mathrm{Tr}(|1\rangle\langle 1|X^{x_k}_{\mathcal{F}}\sigma_{a,l,l'}(X^{x_k}_{\mathcal{F}})^\dagger))\\ &= \frac{1}{q^{3L}}\sum_{a\in F_q^{3L}}\alpha_{P,a}\cdot\mathrm{Tr}(|1\rangle\langle 1|X^{x_k}_{\mathcal{F}}\sigma_{a,a-x_k,a-x_k}(X^{x_k}_{\mathcal{F}})^\dagger)) \quad (218)\\ &\le \frac{1}{q^{3L}}\sum_{a\in F_q^{3L}}\alpha_{P,a} \quad (219)\end{aligned}$$

where we have obtained the second to last equality because $l=l'=a-x_k$ in order for $\mathrm{Tr}(|a\rangle\langle a|(X^{x_k}|l\rangle\langle l'|(X^{x_k})^\dagger))$ to be 1 (otherwise the trace will be 0). The last equality is obtained because $\sigma_{a,a-x_k,a-x_k}$ is a density matrix.

Remark 6.2 *Note that the final inequality would still hold if the projection $|1\rangle\langle 1|$ was replaced by $\mathcal{I}$; this fact will be useful in the proof of Claim 8.3, which is required for the proof of the polynomial version of Corollary 1.11 (Corollary 1.12).*

□

*P*roof of **Corollary6.9**: Let $X^x = X^{x_1}\otimes\cdots\otimes X^{x_t}$. We show that if there exists an i for which X^{x_i} is not k-correlated,

$$(\mathcal{I}^{\otimes d+1}\otimes|0\rangle\langle 0|^{\otimes d})^{\otimes tm}(E_k^\dagger)^{\otimes t}X^x\sigma^k(X^x)^\dagger E_k^{\otimes t}(\mathcal{I}^{\otimes d+1}\otimes|0\rangle\langle 0|^{\otimes d})^{\otimes tm} \quad (220)$$

$$= (\mathcal{I}^{\otimes d+1} \otimes |0\rangle\langle 0|^{\otimes d})^{\otimes tm}((E_k^\dagger)^{\otimes t} X^x E_k^{\otimes t})\sigma((E_k^\dagger)^{\otimes t} X^x E_k^{\otimes t})^\dagger(\mathcal{I}^{\otimes d+1} \otimes |0\rangle\langle 0|^{\otimes d})^{\otimes tm} = 0 \tag{221}$$

where the equality follows due to the fact that σ^k is the density matrix σ encoded with the signed polynomial code. It follows by Claim 5.4 that the expression can be non zero for at most 2 sign keys, which completes the proof. Note that it is implied that the choice of k for which the expression is non zero is independent of σ; it is dependent only on whether X^x is k-correlated or not. Assume there exists an i for which X^{x_i} is not k-correlated. By Claim 5.7 we can break down X^{x_i} into a product of a k-correlated operator Q_k and an uncorrelated operator $\hat{Q}_k$. By equation 116, we know that $\hat{Q}_k$ can be written as

$$\hat{Q}_k = \mathcal{I}^{\otimes d+1} \otimes X^{\hat{x}_{id+2}} \otimes \cdots \otimes X^{\hat{x}_{im}} \tag{222}$$

where $(\hat{x}_{id+2}, \ldots, \hat{x}_{im}) \neq 0^d$ since X^{x_i} is not k-correlated. Now we can refer to equation 118 to write:

$$E_k^\dagger \hat{Q}_k E_k = \mathcal{I}^{\otimes d+1} \otimes X^{\hat{x}_{id+2}} \otimes \cdots \otimes X^{\hat{x}_{im}} \tag{223}$$

Returning to the expression above (equation 221), but just the leftmost part of the expression which operates on the i^{th} register of σ, we have:

$$\begin{aligned}(\mathcal{I}^{\otimes d+1} \otimes |0\rangle\langle 0|^{\otimes d})E_k^\dagger X^{x_i} E_k &= (\mathcal{I}^{\otimes d+1} \otimes |0\rangle\langle 0|^{\otimes d})E_k^\dagger \hat{Q}_k Q_k E_k &(224)\\ &= (\mathcal{I}^{\otimes d+1} \otimes |0\rangle\langle 0|^{\otimes d})E_k^\dagger \hat{Q}_k E_k E_k^\dagger Q_k E_k &(225)\end{aligned}$$

Plugging in the expression for $E_k^\dagger \hat{Q}_k E_k$ from equation 223, we have

$$\ldots = (\mathcal{I}^{\otimes d+1} \otimes |0\rangle\langle 0|^{\otimes d})(\mathcal{I}^{\otimes d+1} \otimes X^{\hat{x}_{id+2}} \otimes \cdots \otimes X^{\hat{x}_{im}})E_k^\dagger Q_k E_k \tag{226}$$

Observe that Q_k is k-correlated; by definition, it preserves a state authenticated with a sign key. Then the rightmost part of equation 226 is simply $E_k^\dagger$ acting on an authenticated state. When $E_k^\dagger$ acts on an arbitrary authenticated density matrix, it performs the following map:

$$E_k^\dagger \sigma^k = \sigma \otimes |0\rangle\langle 0|^{m-1} \tag{227}$$

It follows that the above expression (equation 226) contains the inner product below:

$$\left\langle 0^d \right| (X^{\hat{x}_{id+2}} \otimes \cdots \otimes X^{\hat{x}_{im}}) \left| 0^d \right\rangle \tag{228}$$

which must be equal to 0 since $(\hat{x}_{id+2}, \ldots, \hat{x}_{im}) \neq 0^d$. □

6.8.4 Proofs of Necessary Claims

We begin with Claim 6.6.

Proof of **Claim 6.6**: Recall that we are considering the unauthenticated state σ_a (defined in Claim 6.3), where $\sigma_a = \text{Tr}_{\mathcal{P}_{final}}(\rho_a)$, for $a = (a_1, \ldots, a_L)$ ($a_i \in F_q^3$) and, as defined in Claim 6.1 (in equation 153),

$$\rho_a = (Q_L C_{a_L} \cdots Q_1 C_{a_1} Q_0)\rho(Q_L C_{a_L} \cdots Q_1 C_{a_1} Q_0)^\dagger \tag{229}$$

Note that ρ_a is a pure state and can be written as $|\psi_a\rangle\langle\psi_a|$ (since ρ consists of n 0 qudits and L resource states). Fact 6.1 states that $|\psi_a\rangle$ can be written as:

$$|\psi_a\rangle = \frac{1}{\sqrt{q^{3L}}} \sum_{l \in F_q^{3L}} |l\rangle |\psi\rangle_{a,l} \tag{230}$$

where $|\psi\rangle_{a,a} = U|0\rangle^{\otimes n}$ (recall that U is the Q-CIRCUIT instance which the prover was asked to apply). It follows that the state σ_a can be written as

$$\sigma_a = \mathrm{Tr}_{\mathcal{P}_{final}}(|\psi_a\rangle\langle\psi_a|) = \frac{1}{q^{3L}} \sum_{l,l' \in F_q^{3L}} |l\rangle\langle l'| \otimes \mathrm{Tr}_{\mathcal{P}_{final}}(|\psi\rangle_{a,l}\langle\psi|_{a,l'}) \tag{231}$$

□ We proceed to proving Claim 6.7.

Proof of **Claim 6.7**: Recall that we would like to prove

$$(\Pi_{G_a} \otimes (\hat{\Pi}_0)_{\mathcal{F}})(E_k^\dagger)^{\otimes|\mathcal{V}_{final}|} P E_k^{\otimes|\mathcal{V}_{final}|} = (E_k^\dagger)^{\otimes|\mathcal{V}_{final}|} P E_k^{\otimes|\mathcal{V}_{final}|} (\Pi_{G_a} \otimes (\hat{\Pi}_0)_{\mathcal{F}}) \tag{232}$$

for a trivial Pauli operator $P \in \mathbb{P}_{|\mathcal{V}_{final}|}$, where Π_{G_a} is defined in equation 187 and $(\hat{\Pi}_0)_{\mathcal{F}}$ is defined in 158. In other words, we want to show that the Pauli operator $(E_k^\dagger)^{\otimes|\mathcal{V}_{final}|} P E_k^{\otimes|\mathcal{V}_{final}|}$ commutes with the projection $\Pi_{G_a} \otimes (\hat{\Pi}_0)_{\mathcal{F}}$ when P is trivial. Observe that

$$(E_k^\dagger)^{\otimes|\mathcal{V}_{final}|} P E_k^{\otimes|\mathcal{V}_{final}|} \tag{233}$$

must be trivial in the registers $1, d+2, \ldots, m$; only registers $d+2, \ldots, m$ can be non trivial. This follows from the definition of E_k (see Section 2.5.2). In more detail, $E_k^\dagger$ consists of *SUM* and multiplication operations (which compose $D_k^\dagger$) followed by Fourier operations. As shown in Section 2.4 (in equations 21 and 23), the *SUM* and multiplication operators in $D_k^\dagger$ map trivial operators to trivial operators by conjugation. The Fourier operators, which flip Z and X operators, act only on registers $2, \ldots, d+1$, so only these registers can be mapped to non trivial operators.

Note that trivial operators commute with standard basis projections, and $\Pi_{G_a} \otimes (\hat{\Pi}_0)_{\mathcal{F}}$ acts with standard basis projections on registers $1, d+2, \ldots, m$ for each block of m registers. Since $\Pi_{G_a} \otimes (\hat{\Pi}_0)_{\mathcal{F}}$ acts as $\mathcal{I}$ on registers $2, \ldots, d+1$ for each block of m registers, the non trivial portion of $(E_k^\dagger)^{\otimes|\mathcal{V}_{final}|} P E_k^{\otimes|\mathcal{V}_{final}|}$ also commutes with $\Pi_{G_a} \otimes (\hat{\Pi}_0)_{\mathcal{F}}$. □

Finally, we prove Claim 6.8.

Proof of **Claim 6.8**: Recall that our goal is to prove the following equality for all $P = Z^z X^x \in \mathbb{P}_{|\mathcal{V}_{final}|}$:

$$\mathrm{Tr}(\hat{\Pi}_0 \sigma_k^P) = \sum_{a \in F_q^{3L}} \alpha_{P,a} \cdot \mathrm{Tr}((\Pi_{G_a} \otimes (\hat{\Pi}_0)_{\mathcal{F}})(E_k^\dagger)^{\otimes|\mathcal{V}_{final}|} X^x \sigma_a^k (X^x)^\dagger (E_k)^{\otimes|\mathcal{V}_{final}|}) \tag{234}$$

where σ_k^P was defined as follows in Claim 6.3:

$$\sigma_k^P = \sum_{\Delta_L \in F_q^{|\mathcal{V}_{L+1}|}} \alpha_{P,g(\Delta_L)} \cdot ((|\Delta_L\rangle\langle\Delta_L| \otimes \mathcal{I}_{\mathcal{F}})(D_k^\dagger)^{\otimes|\mathcal{V}_{final}|} P) \sigma_{g(\Delta_L)}^k ((|\Delta_L\rangle\langle\Delta_L| \otimes \mathcal{I}_{\mathcal{F}})(D_k^\dagger)^{\otimes|\mathcal{V}_{final}|} P)^\dagger \tag{235}$$

and the projection $\hat{\Pi}_0$ acting on $\mathcal{V}_{final} = \mathcal{V}_{L+1} \cup F$ was defined in equation 158 as follows:

$$\hat{\Pi}_0 = (\mathcal{I}^{\otimes d+1} \otimes |0\rangle\langle 0|^{\otimes d})_{\mathcal{V}_{L+1}}^{\otimes 3L} \otimes (|1\rangle\langle 1| \otimes \mathcal{I}^{\otimes d} \otimes |0\rangle\langle 0|^{\otimes d})_{\mathcal{F}} = (\hat{\Pi}_0)_{\mathcal{V}_{L+1}} \otimes (\hat{\Pi}_0)_{\mathcal{F}} \tag{236}$$

Before continuing, note that the projection $\hat{\Pi}_0$ makes it unnecessary to sum over all Δ_L in the expression for σ_k; we can instead sum over a subset of Δ_L, and split the sum according to the value $a \in F_q^{3L}$ of $g(\Delta_L)$:

$$\mathrm{Tr}(\hat{\Pi}_0 \sigma_k^P) = \mathrm{Tr}(\hat{\Pi}_0 \sum_{\substack{a \in F_q^{3L} \\ \Delta_L \in G_a}} \alpha_{P,a} \cdot ((|\Delta_L\rangle\langle\Delta_L| \otimes \mathcal{I}_{\mathcal{F}})(D_k^\dagger)^{\otimes|\mathcal{V}_{final}|} P) \sigma_a^k ((|\Delta_L\rangle\langle\Delta_L| \otimes \mathcal{I}_{\mathcal{F}})(D_k^\dagger)^{\otimes|\mathcal{V}_{final}|} P)^\dagger) \tag{237}$$

where

$$G_a \stackrel{\text{def}}{=} \{((s_1, 0^d), \ldots, (s_{3L}, 0^d))|s_1, \ldots, s_{3L} \in F_q^{d+1}, g(s_1, \ldots, s_{3L}) = a\} \tag{238}$$

Instead of summing over all $\Delta_L \in F_q^{|\mathcal{V}_{L+1}|}$, we have restricted to $\Delta_L \in \cup_a G_a$. This is because Δ_L must equal $((s_1, 0^d), \ldots, (s_{3L}, 0^d))$ to give a non zero trace when projected onto $\hat{\Pi}_0$. Next, since $|\Delta_L\rangle \langle\Delta_L| \otimes \mathcal{I}_{\mathcal{F}}$ commutes with $\hat{\Pi}_0$, we can remove $|\Delta_L\rangle \langle\Delta_L| \otimes \mathcal{I}_{\mathcal{F}}$ from the right hand side of equation 237 (due to the cyclic nature of trace), obtaining:

$$\begin{aligned} \text{Tr}(\hat{\Pi}_0 \sigma_k^P) &= \text{Tr}(\hat{\Pi}_0 \sum_{\substack{a \in F_q^{3L} \\ \Delta_L \in G_a}} \alpha_{P,a} \cdot ((|\Delta_L\rangle \langle\Delta_L| \otimes \mathcal{I}_{\mathcal{F}})(D_k^\dagger)^{\otimes |\mathcal{V}_{final}|} P) \sigma_a^k ((D_k^\dagger)^{\otimes |\mathcal{V}_{final}|} P)^\dagger) & (239) \\ &= \text{Tr}(\hat{\Pi}_0 \sum_{a \in F_q^{3L}} \alpha_{P,a} \cdot ((\Pi_{G_a} \otimes \mathcal{I}_{\mathcal{F}})(D_k^\dagger)^{\otimes |\mathcal{V}_{final}|} P) \sigma_a^k ((D_k^\dagger)^{\otimes |\mathcal{V}_{final}|} P)^\dagger) & (240) \end{aligned}$$

where for $a = (a(1), \ldots, a(3L)) \in F_q^{3L}$

$$\Pi_{G_a} = \sum_{\Delta_L \in G_a} |\Delta_L\rangle \langle\Delta_L| = (|a(1)\rangle \langle a(1)| \otimes \mathcal{I}^{\otimes d} \otimes |0\rangle \langle 0|^{\otimes d}) \otimes \cdots \otimes (|a(3L)\rangle \langle a(3L)| \otimes \mathcal{I}^{\otimes d} \otimes |0\rangle \langle 0|^{\otimes d}) \tag{241}$$

We can further simplify to:

$$\text{Tr}(\hat{\Pi}_0 \sigma_k^P) = \sum_{a \in F_q^{3L}} \alpha_{P,a} \cdot \text{Tr}((\Pi_{G_a} \otimes (\hat{\Pi}_0)_{\mathcal{F}})(D_k^\dagger)^{\otimes |\mathcal{V}_{final}|} P \sigma_a^k P^\dagger (D_k)^{\otimes |\mathcal{V}_{final}|}) \tag{242}$$

because

$$(\hat{\Pi}_0)_{\mathcal{V}_{L+1}} \Pi_{G_a} = \Pi_{G_a} \tag{243}$$

Recall (from 31) that

$$D_k^\dagger = (\mathcal{I} \otimes F^{\otimes d} \otimes \mathcal{I}^{\otimes d}) E_k^\dagger \tag{244}$$

It follows that:

$$\begin{aligned} (\Pi_{G_a} \otimes (\hat{\Pi}_0)_{\mathcal{F}})(D_k^\dagger)^{\otimes |\mathcal{V}_{final}|} &= (\Pi_{G_a} \otimes (\hat{\Pi}_0)_{\mathcal{F}})((\mathcal{I} \otimes F^{\otimes d} \otimes \mathcal{I}^{\otimes d})^\dagger)^{\otimes |\mathcal{V}_{final}|} (E_k^\dagger)^{\otimes |\mathcal{V}_{final}|} & (245) \\ &= ((\mathcal{I} \otimes F^{\otimes d} \otimes \mathcal{I}^{\otimes d})^\dagger)^{\otimes |\mathcal{V}_{final}|} (\Pi_{G_a} \otimes (\hat{\Pi}_0)_{\mathcal{F}})(E_k^\dagger)^{\otimes |\mathcal{V}_{final}|} & (246) \end{aligned}$$

The commutation in the final equality occurs because of the structure of $(\Pi_{G_a} \otimes (\hat{\Pi}_0)_{\mathcal{F}})$; for each set of m registers, it acts as identity on registers $2, \ldots, d+1$ in the set. Plugging in the equality obtained above, we obtain:

$$\text{Tr}(\hat{\Pi}_0 \sigma_k^P) = \sum_{a \in F_q^{3L}} \alpha_{P,a} \cdot \text{Tr}(((\Pi_{G_a} \otimes (\hat{\Pi}_0)_{\mathcal{F}})(E_k^\dagger)^{\otimes |\mathcal{V}_{final}|} P) \sigma_a^k ((E_k^\dagger)^{\otimes |\mathcal{V}_{final}|} P)^\dagger) \tag{247}$$

Note that we have removed the terms corresponding to $((\mathcal{I} \otimes F^{\otimes d} \otimes \mathcal{I}^{\otimes d})^\dagger)^{\otimes |\mathcal{V}_{final}|}$; these terms canceled due to the cyclic nature of trace, since they were present on both ends of the above expression.

To complete the claim, we need to show that $P = Z^z X^x$ can be replaced by X^x. To begin, observe that in the expression for σ_k^P, we can replace

$$(E_k^\dagger)^{\otimes |\mathcal{V}_{final}|} P \tag{248}$$

with

$$(E_k^\dagger)^{\otimes |\mathcal{V}_{final}|} Z^z E_k^{\otimes |\mathcal{V}_{final}|} (E_k^\dagger)^{\otimes |\mathcal{V}_{final}|} X^x \tag{249}$$

By Claim 6.7, we know that the Pauli operator $(E_k^\dagger)^{\otimes |\mathcal{V}_{final}|} Z^z E_k^{\otimes |\mathcal{V}_{final}|}$ can be commuted past the projection $\Pi_{G_a} \otimes (\hat{\Pi}_0)_{\mathcal{F}}$. Note that since we have now pulled the Pauli operator

$$(E_k^\dagger)^{\otimes |\mathcal{V}_{final}|} Z^z E_k^{\otimes |\mathcal{V}_{final}|} \tag{250}$$

past the projection, we can remove it from the expression, due to the cyclic nature of trace. □

7 Blind QPIP

In this section, we will prove the following theorem:
Theorem 1.8 *Theorem 1.1 holds also in a blind setting, namely, the prover does not get any information regarding the function being computed and its input.*

To begin, we define blindness:

Definition 7.1 *[AS06, BFK08, Chi01] Secure blind quantum computation is a process where a server computes a function for a client and the following properties hold:*

- ***Blindness**: The prover gets no information beyond an upper bound on the size of the circuit. Formally, in a blind computation scheme for a set of circuits $\mathfrak{C}_n$ which take as input strings in $\{0,1\}^n$, the prover's reduced density matrix is identical for every $C \in \mathfrak{C}_n$ and input $x \in \{0,1\}^n$.*
- ***Security**: Completeness and soundness hold the same way as in QAS (Definition 2.1).*

We use the QPIP protocols for Q-CIRCUIT in order to provide a blind QPIP for any language in BQP. To do this, we require a universal circuit. Roughly, a universal circuit acts on input bits and control bits. The control bits can be thought of as a description of a circuit that should be applied to the input bits. Universal circuits can be formally defined as follows:

Definition 7.2 *For a circuit U acting on n qubits, let $c(U) \in \{0,1\}^k$ be the canonical (classical) description of U. The universal circuit $\mathfrak{U}_{n,k}$ acts in the following way:*

$$\mathfrak{U}_{n,k} |\phi\rangle \otimes |c(U)\rangle \longrightarrow U |\phi\rangle |c(U)\rangle \tag{251}$$

Constructing such a circuit is an easy exercise. We would like the universal circuit to simulate any circuit made of Toffoli and Hadamard gates on n qubits - it is well known that such circuits are quantum universal. Assume there is an upper bound of m gates in the circuit. The universal circuit is split up into m layers. Each layer i consists of every possible gate on the n input qubits, and each such gate is controlled by 1 qubit. Only 1 of these control qubits will be set to 1, based on which is the i^{th} gate applied in U.

To perform a blind computation, the verifier will compute, with the prover's help, the result of the universal circuit acting on input and control bits. It follows that to prove blindness, we need to show that the input to the universal circuit is hidden from the prover. To do this, we show that the prover's density matrix in both the Clifford and polynomial schemes remains independent of the input (to the universal circuit) throughout the computation. We begin with the Clifford scheme.
Theorem 1.9 (Blindness of the Clifford based QPIP) *The state of the prover in the Clifford based QPIP (Protocol 4.1) is independent of the input to the circuit which is being computed throughout the protocol.*

*P*roof of **Theorem 1.9**: We do not need to consider the prover's extra space, since that contains no information about the input. We need only consider the qubits sent to the prover by the verifier. Whenever the prover receives a state from the verifier (at the beginning of the protocol and during the protocol), the verifier has chosen new, independent keys to encode the state using the Clifford QAS. Therefore, each density matrix sent to the prover by the verifier is the maximally mixed state (by Lemma 4.4) regardless of the input. This remains true throughout the protocol, since when the prover sends a register to the verifier, the verifier returns a completely mixed state, independent of the remaining registers. It follows that the prover's state (other than his extra space) is always described by the completely mixed state. □

We now consider the polynomial QPIP.

Theorem 1.10 (Blindness of the Polynomial Based QPIP) *The state of the prover in the polynomial based QPIP (Protocol 6.1) remains independent of the input to the circuit which is being computed throughout the protocol.*

We remark that the proof of this fact turns out to be rather cumbersome, because we are relying on the randomness provided by the measurement results to prove blindness. In fact, the proof can be greatly simplified by adding additional randomness to the Toffoli states (intuitively, this adds a one time pad to the decoded measurement results sent to the prover). However, it is interesting to note that this additional randomness is not needed for blindness, and that the randomness of the measurement results is indeed enough.

*P*roof of **Theorem 1.10**: This case is more complicated, due to the classical interaction in each round. Without the classical interaction, the prover's initial state is just the maximally mixed state, due to the Pauli keys (by Lemma 4.5), so blindness follows easily in this case. Returning to the case in which there is classical interaction, we need to show that the joint quantum state and classical information of the prover are independent of the input to the computation. Recall that each message sent back by the verifier is a decoded measurement result which is required in order to apply the Toffoli gate. We will argue that due to the way the Toffoli gate is applied (see Section 2.3.1 and Fact 6.1), the decoded measurement results are uniformly random regardless of the input. This implies that revealing the decoded measurement result leaks no information about the input. We now proceed to prove this formally.

We will show that at the start of the final round (round $L+1$) the prover's state (which includes the classical messages from the verifier) is independent of the input. As given in Claim 6.1 and equation 155 the joint state of the prover's registers, $\mathcal{P}_{L+1}$, the environment $\mathcal{E}$, the verifier's registers, $\mathcal{V}_{L+1}$ and the key register containing the sign key and Pauli keys of qudits in $\mathcal{P}_{L+1}$, is:

$$\frac{1}{2^m|\mathbb{P}_{m'}|}\sum_{\substack{z,x\in F_q^{|\mathcal{P}_{L+1}|}\\ k\in\{-1,1\}^m}}\tau_{L+1}(z,x,k)\otimes\sum_{\Delta_L,z_1,x_1\in F_q^{|\mathcal{V}_{L+1}|}}W^{L+1}_{\Delta_L,\hat{z},\hat{x},k}(\rho^k_{g(\Delta_L)}\otimes\rho_{\mathcal{E}})(W^{L+1}_{\Delta_L,\hat{z},\hat{x}})^\dagger \tag{252}$$

where

$$\hat{z}=(z_1,z),\hat{x}=(x_1,x), \tag{253}$$

$$W^{L+1}_{\Delta_L,\hat{z},\hat{x},k}=(|\Delta_L\rangle\langle\Delta_L|(D_k^\dagger)^{\otimes|\mathcal{V}_{L+1}|}(Z^{z_1}X^{x_1})^\dagger_{\mathcal{V}_{L+1}}\otimes\mathcal{I}_{\mathcal{P}_{L+1},E})U_{g(\Delta_L)}((Z^{z_1}X^{x_1})_{\mathcal{V}_{L+1}}\otimes(Z^zX^x)_{\mathcal{P}_{L+1}}\otimes\mathcal{I}_{\mathcal{E}}) \tag{254}$$

where $U_{g(\Delta_L)}$ is a unitary operator dependent on $g(\Delta_L)$, $\rho_{\mathcal{E}}$ is the initial state of the prover's environment and

$$\rho^k_{g(\Delta_L)}=(\tilde{Q}_L\tilde{C}_{g(\delta_L)}\cdots\tilde{Q}_1\tilde{C}_{g(\delta_1)}\tilde{Q}_0)\rho^k(\tilde{Q}_L\tilde{C}_{g(\delta_L)}\cdots\tilde{Q}_1\tilde{C}_{g(\delta_1)}\tilde{Q}_0)^\dagger \tag{255}$$

for $\Delta_L=(\delta_1,\ldots,\delta_L)\in F_q^{|\mathcal{V}_{L+1}|}$ and where ρ^k is the initial density matrix, containing an authentication of the input state on n qudits and authentications of L Toffoli states on 3 qudits each.

First, since we are only considering the prover's state, we can trace out the verifier's registers $\mathcal{V}_{L+1}$ and the first register containing the keys. This gives the following state:

$$\frac{1}{2^m|\mathbb{P}_{m'}|}\sum_{\substack{z,x\in F_q^{|\mathcal{P}_{L+1}|}\\ k\in\{-1,1\}^m}}\sum_{\Delta_L,z_1,x_1\in F_q^{|\mathcal{V}_{L+1}|}}\mathrm{Tr}_{\mathcal{V}_{L+1}}(W^{L+1}_{\Delta_L,\hat{z},\hat{x},k}(\rho^k_{g(\Delta_L)}\otimes\rho_{\mathcal{E}})(W^{L+1}_{\Delta_L,\hat{z},\hat{x}})^\dagger) \tag{256}$$

Next, the sum over $z, x \in F_q^{|\mathcal{P}_{L+1}|}$ allows us to use the Pauli mixing lemma (Lemma 4.5) to replace $\rho^k_{g(\Delta_L)}$ with

$$\mathrm{Tr}_{\mathcal{P}_{L+1}}(\rho^k_{g(\Delta_L)}) \otimes \frac{1}{q^{|\mathcal{P}_{L+1}|}} \mathcal{I}^{\otimes |\mathcal{P}_{L+1}|} \tag{257}$$

We can now rewrite the prover's state as:

$$\frac{1}{2^m |\mathbb{P}_{|\mathcal{V}_{L+1}|}|} \sum_{\substack{k \in \{-1,1\}^m \\ \Delta_L, z_1, x_1 \in F_q^{|\mathcal{V}_{L+1}|}}} \mathrm{Tr}_{\mathcal{V}_{L+1}}(W'_{\Delta_L, z_1, x_1, k}(\sigma_{g(\Delta_L),k})(W'_{\Delta_L, z_1, x_1, k})^\dagger) \tag{258}$$

where

$$\sigma_{g(\Delta_L),k} = \mathrm{Tr}_{\mathcal{P}_{L+1}}(\rho^k_{g(\Delta_L)}) \otimes \frac{1}{q^{|\mathcal{P}_{L+1}|}} \mathcal{I}^{\otimes |\mathcal{P}_{L+1}|} \otimes \rho_{\mathcal{E}} \tag{259}$$

$$W'_{\Delta_L, z_1, x_1, k} = (|\Delta_L\rangle \langle \Delta_L| (D_k^\dagger)^{\otimes |\mathcal{V}_{L+1}|} (Z^{z_1} X^{x_1})^\dagger_{\mathcal{V}_{L+1}} \otimes \mathcal{I}_{\mathcal{P}_{L+1}, E}) U_{g(\Delta_L)} ((Z^{z_1} X^{x_1})_{\mathcal{V}_{L+1}} \otimes \mathcal{I}_{\mathcal{P}_{L+1}} \otimes \mathcal{I}_{\mathcal{E}}) \tag{260}$$

Since $\rho^k_{g(\Delta_L)}$ is a pure state, it can be written as $\left|\psi_{g(\Delta_L)}\right\rangle^k \left\langle\psi_{g(\Delta_L)}\right|^k$. Recall from Fact 6.1 that the unauthenticated state $\left|\psi_{g(\Delta_L)}\right\rangle$ can be written as

$$\left|\psi_{g(\Delta_L)}\right\rangle = \frac{1}{\sqrt{q^{3L}}} \sum_{l \in F_q^{3L}} |l\rangle |\psi\rangle_{g(\Delta_L),l} \tag{261}$$

Given this form of the unauthenticated state, we revert back to analyzing the authenticated state, which is:

$$\rho^k_{g(\Delta_L)} = (\left|\psi_{g(\Delta_L)}\right\rangle \left\langle\psi_{g(\Delta_L)}\right|)^k = \frac{1}{q^{3L}} \sum_{l, l' \in F_q^{3L}} (|l\rangle \left\langle l'\right|)^k \otimes (|\psi\rangle_{g(\Delta_L),l} \langle\psi|_{g(\Delta_L),l'})^k \tag{262}$$

Since $\mathcal{V}_{L+1}$ corresponds to the first register in the above sum (containing the authentication of l) and $\mathcal{P}_{L+1}$ corresponds to the register containing the authentication of $|\psi\rangle_{g(\Delta_L),l}$, we have:

$$\mathrm{Tr}_{\mathcal{P}_{L+1}}(\rho^k_{g(\Delta_L)}) = \frac{1}{q^{3L}} \sum_{l, l' \in F_q^{3L}} \mathrm{Tr}(|\psi\rangle_{g(\Delta_L),l} \langle\psi|_{g(\Delta_L),l'})^k \cdot (|l\rangle \left\langle l'\right|)^k \tag{263}$$

In the following claim (which we prove after this proof), we show that once we plug in the above expression into the prover's state as given in Equation 258, the state is only non zero when $l = l'$. To see this, observe that summing over $z_1, x_1 \in F_q^{|\mathcal{V}_{L+1}|}$ results in decohering (removing cross terms of) the part of $U_{g(\Delta_L)}$ acting on register $\mathcal{V}_{L+1}$ by the Pauli decoherence lemma (Lemma 5.1). Then due to the standard basis projection onto $|\Delta_L\rangle \langle\Delta_L|$, the state will be zero unless $l = l'$.

Claim 7.1 *The following expression (from equation 258), which represents the prover's state at the start of round* $L+1$

$$\frac{1}{2^m |\mathbb{P}_{|\mathcal{V}_{L+1}|}|} \sum_{\substack{k \in \{-1,1\}^m \\ \Delta_L, z_1, x_1 \in F_q^{|\mathcal{V}_{L+1}|}}} Tr_{\mathcal{V}_{L+1}}(W'_{\Delta_L, z_1, x_1, k}(\sigma_{g(\Delta_L),k})(W'_{\Delta_L, z_1, x_1, k})^\dagger) \tag{264}$$

is equal to

$$\frac{1}{q^{3L} 2^m} \sum_{\substack{k \in \{-1,1\}^m \\ P \in \mathbb{P}_{|\mathcal{V}_{L+1}|} \\ \Delta_L \in F_q^{|\mathcal{V}_{L+1}|}, l \in F_q^{3L}}} \mu_{\Delta_L, l, P, k} \cdot U_{P, g(\Delta_L)} (\frac{1}{q^{|\mathcal{P}_{L+1}|}} \mathcal{I}^{\otimes |\mathcal{P}_{L+1}|} \otimes \rho_{\mathcal{E}}) U^\dagger_{P, g(\Delta_L)} \tag{265}$$

where

$$U_{g(\Delta_L)} = \sum_{P \in \mathbb{P}_{|\mathcal{V}_{L+1}|}} P \otimes U_{P,g(\Delta_L)} \tag{266}$$

and

$$\mu_{\Delta_L,l,P,k} = Tr(|\Delta_L\rangle\langle\Delta_L| (D_k^\dagger)^{\otimes|\mathcal{V}_{L+1}|} P(|l\rangle\langle l|)^k P^\dagger (D_k)^{\otimes|\mathcal{V}_{L+1}|}) \tag{267}$$

The above state is the same regardless of the input density matrix ρ^k; therefore, we have shown blindness for the polynomial scheme. □

We proceed to the proof of the claim.

*P*roof of **Claim 7.1**: We begin with

$$\frac{1}{2^m|\mathbb{P}_{|\mathcal{V}_{L+1}|}|} \sum_{\substack{k\in\{-1,1\}^m \\ \Delta_L,z_1,x_1\in F_q^{|\mathcal{V}_{L+1}|}}} \mathrm{Tr}_{\mathcal{V}_{L+1}}(W'_{\Delta_L,z_1,x_1,k}(\sigma_{g(\Delta_L),k})(W'_{\Delta_L,z_1,x_1,k})^\dagger) \tag{268}$$

where

$$\sigma_{g(\Delta_L),k} = \mathrm{Tr}_{\mathcal{P}_{L+1}}(\rho^k_{g(\Delta_L)}) \otimes \frac{1}{q^{|\mathcal{P}_{L+1}|}} \mathcal{I}^{\otimes|\mathcal{P}_{L+1}|} \otimes \rho_{\mathcal{E}} \tag{269}$$

and

$$W'_{\Delta_L,z_1,x_1,k} = (|\Delta_L\rangle\langle\Delta_L| (D_k^\dagger)^{\otimes|\mathcal{V}_{L+1}|}(Z^{z_1}X^{x_1})^\dagger_{\mathcal{V}_{L+1}} \otimes \mathcal{I}_{\mathcal{P}_{L+1},E}) U_{g(\Delta_L)}((Z^{z_1}X^{x_1})_{\mathcal{V}_{L+1}} \otimes \mathcal{I}_{\mathcal{P}_{L+1}} \otimes \mathcal{I}_{\mathcal{E}}) \tag{270}$$

We can now apply the Pauli decoherence lemma (Lemma 5.1) with the decomposition of $U_{g(\Delta_L)}$ as given in equation 266 (and with $Z^{z_1}X^{x_1}$ playing the role of Q in the lemma) to simplify the state in equation 268 to:

$$\frac{1}{2^m} \sum_{\substack{k\in\{-1,1\}^m,\Delta_L\in F_q^{|\mathcal{V}_{L+1}|} \\ P\in\mathbb{P}_{|\mathcal{V}_{L+1}|}}} \mathrm{Tr}_{\mathcal{V}_{L+1}}(W''_{\Delta_L,P,k}(\sigma_{g(\Delta_L),k})(W''_{\Delta_L,P,k})^\dagger) \tag{271}$$

where

$$W''_{\Delta_L,P,k} = |\Delta_L\rangle\langle\Delta_L| (D_k^\dagger)^{\otimes|\mathcal{V}_{L+1}|} P \otimes U_{P,g(\Delta_L)} \tag{272}$$

Now we can plug in the expression for $\mathrm{Tr}_{\mathcal{P}_{L+1}}(\rho^k_{g(\Delta_L)})$ from equation 263 to obtain:

$$\sigma_{g(\Delta_L),k} = \frac{1}{q^{3L}} \sum_{l,l'\in F_q^{3L}} \mathrm{Tr}(|\psi\rangle_{g(\Delta_L),l}\langle\psi|_{g(\Delta_L),l'})^k \cdot (|l\rangle\langle l'|)^k \otimes \frac{1}{q^{|\mathcal{P}_{L+1}|}} \mathcal{I}^{\otimes|\mathcal{P}_{L+1}|} \otimes \rho_{\mathcal{E}} \tag{273}$$

Plugging this in to equation 271 we obtain:

$$\frac{1}{q^{3L}2^m} \sum_{\substack{k\in\{-1,1\}^m \\ P\in\mathbb{P}_{|\mathcal{V}_{L+1}|} \\ \Delta_L\in F_q^{|\mathcal{V}_{L+1}|},l,l'\in F_q^{3L}}} \mu_{\Delta_L,l,l',P,k} \mathrm{Tr}((|\psi\rangle_{g(\Delta_L),l}\langle\psi|_{g(\Delta_L),l'})^k) \cdot U_{P,g(\Delta_L)}(\frac{1}{q^{|\mathcal{P}_{L+1}|}} \mathcal{I}^{\otimes|\mathcal{P}_{L+1}|} \otimes \rho_{\mathcal{E}}) U^\dagger_{P,g(\Delta_L)} \tag{274}$$

where

$$\mu_{\Delta_L,l,l',P,k} = \mathrm{Tr}(|\Delta_L\rangle\langle\Delta_L| (D_k^\dagger)^{\otimes|\mathcal{V}_{L+1}|} P(|l\rangle\langle l'|)^k P^\dagger (D_k)^{\otimes|\mathcal{V}_{L+1}|}) \tag{275}$$

Observe that $\mu_{\Delta_L,l,l',P,k}$ is only non zero when $l = l'$. This follows due to two observations. First, note that

$$(D_k)^{\otimes|\mathcal{V}_{L+1}|} |\Delta_L\rangle \langle\Delta_L| (D_k^\dagger)^{\otimes|\mathcal{V}_{L+1}|} \tag{276}$$

is a standard basis projection, because $|\Delta_L\rangle \langle\Delta_L|$ is a standard basis projection, and D_k consists only of sum and multiplication operations (see Claim 2.6). Next, note that if we have a standard basis projection S acting on a matrix $P|\psi\rangle\langle\psi'|P^\dagger$ in register $\mathcal{V}_{L+1}$, where $|\psi\rangle = \sum_i \alpha_i |i\rangle$ and $|\psi'\rangle = \sum_i \beta_i |i\rangle$, we need not consider the cross terms of $|\psi\rangle\langle\psi'|$:

$$\begin{aligned} \text{Tr}(SP|\psi\rangle\langle\psi'|P^\dagger) &= \sum_{i,j} \alpha_i \beta_j^* \text{Tr}(SP|i\rangle\langle j|P^\dagger) & (277)\\ &= \sum_i \alpha_i \beta_i^* \text{Tr}(SP|i\rangle\langle i|P^\dagger) & (278)\end{aligned}$$

Since the authenticated states $(|l\rangle\langle l'|)^k$ consists only of cross terms unless $l = l'$, this implies that l must equal l' in order for $\mu_{\Delta_L,l,l',P,k}$ to be non zero. Therefore, we can now write equation 274 as

$$\frac{1}{q^{3L}2^m} \sum_{\substack{k\in\{-1,1\}^m \\ P\in\mathbb{P}_{|\mathcal{V}_{L+1}|} \\ \Delta_L\in F_q^{|\mathcal{V}_{L+1}|}, l\in F_q^{3L}}} \mu_{\Delta_L,l,P,k} \text{Tr}((|\psi\rangle_{g(\Delta_L),l}\langle\psi|_{g(\Delta_L),l})^k) \cdot U_{P,g(\Delta_L)}(\frac{1}{q^{|\mathcal{P}_{L+1}|}}\mathcal{I}^{\otimes|\mathcal{P}_{L+1}|} \otimes \rho_\mathcal{E}) U_{P,g(\Delta_L)}^\dagger \tag{279}$$

Since $\text{Tr}((|\psi\rangle_{g(\Delta_L),l}\langle\psi|_{g(\Delta_L),l})^k) = 1$, the above expression is equal to

$$\frac{1}{q^{3L}2^m} \sum_{\substack{k\in\{-1,1\}^m \\ P\in\mathbb{P}_{|\mathcal{V}_{L+1}|} \\ \Delta_L\in F_q^{|\mathcal{V}_{L+1}|}, l\in F_q^{3L}}} \mu_{\Delta_L,l,P,k} \cdot U_{P,g(\Delta_L)}(\frac{1}{q^{|\mathcal{P}_{L+1}|}}\mathcal{I}^{\otimes|\mathcal{P}_{L+1}|} \otimes \rho_\mathcal{E}) U_{P,g(\Delta_L)}^\dagger \tag{280}$$

□

8 Interpretation of Results

In this section, we prove Corollary 1.11.

8.1 Clifford QPIP

Corollary 8.1 *For the Clifford QPIP protocol (Protocol 4.1) with security parameter ϵ (where $\epsilon = \frac{1}{2^e}$ by definition), if the verifier does not abort with probability $\geq \beta$ then the trace distance between the final density matrix conditioned on the verifier's acceptance and that of the correct state is at most $\frac{\epsilon}{\beta}$*

*P*roof of **Corollary 8.1**: The final state of the protocol before the verifier's cheat detection can be written as (see Eq. 62):

$$s\text{Tr}_A(\rho_{N+1}) + \frac{1-s}{4^m-1} \sum_{Q\in\mathbb{P}_m\setminus\{\mathcal{I}\}} Q(\text{Tr}_A(\rho_{N+1}))Q^\dagger \tag{281}$$

where s represents the weight of the prover's attack on the identity, A is the space of all computational qubits other than the first, and ρ_{N+1} is the correct final state of the protocol:

$$\rho_{N+1} = (U_N \cdots U_1)\rho(U_N \cdots U_1)^\dagger \tag{282}$$

where ρ is equal to the initial density matrix. Note that ρ_{N+1} includes the auxiliary 0 states, but the circuit does not act on the auxiliary 0 states (so it includes $\mathcal{I}$ operators which we have not included for ease of notation). We can instead write

$$\rho'_{N+1} = (U_N \cdots U_1)\rho'(U_N \cdots U_1)^\dagger \tag{283}$$

where ρ' is the input state ρ without the auxiliary 0's. Then

$$\mathrm{Tr}_A(\rho_{N+1}) = \mathrm{Tr}_{A'}(\rho'_{N+1}) \otimes |0\rangle \langle 0|^{\otimes e} \stackrel{\mathrm{def}}{=} \rho_C \otimes |0\rangle \langle 0|^{\otimes e} \tag{284}$$

where A' is the space of all compuational qubits other than the first (but excluding the auxiliary 0's).

We now rewrite the state from equation Eq. 281:

$$s\rho_C \otimes |0\rangle \langle 0|^{\otimes e} + \frac{1-s}{4^m - 1} \sum_{Q_1 \otimes Q_2 \in \mathbb{P}_m \setminus \{\mathcal{I}\}} Q_1 \rho_C Q_1^\dagger \otimes Q_2 |0\rangle \langle 0|^{\otimes e} Q_2^\dagger \tag{285}$$

Let $V \subset \mathbb{P}_m \setminus \{\mathcal{I}\}$ be the set of Pauli operators which pass the cheat detection procedure (i.e. preserve the auxiliary 0 states). Assume the verifier declares the computation valid with probability β. After he declares the computation valid (and we trace out the auxiliary 0 states), his state is:

$$\sigma = \frac{1}{\beta}(s\rho_C + \frac{1-s}{4^m - 1} \sum_{Q_1 \otimes Q_2 \in V} Q_1 \rho_C Q_1^\dagger) \tag{286}$$

Then the trace distance to the correct state ρ_C is:

$$\begin{aligned} T(\sigma, \rho_C) &\leq \frac{1}{\beta}(sT(\rho_C, \rho_C) + \frac{1-s}{4^m - 1} \sum_{Q_1 \otimes Q_2 \in V} T(Q_1 \rho_C Q_1^\dagger, \rho_C)) && (287) \\ &\leq \frac{1}{\beta} \cdot \frac{|V|}{4^m - 1} && (288) \\ &\leq \frac{\epsilon}{\beta} && (289) \end{aligned}$$

where the first inequality follows by convexity of trace distance and the final inequality follows from the same argument used to prove the security of the Cliford QAS- more specifically, see the explanation preceding equation 44. □

8.2 Polynomial QPIP

We now continue to the polynomial QPIP. In this setting, we are concerned with the trace distance between density matrices *after* measurement. This is because the verifier in the polynomial QPIP protocol only performs a classical verification circuit; therefore, he cannot detect phase attacks on the correct state. Once the state is measured, phase attacks have no effect on the state. For this purpose, let σ_M represent a density matrix σ on one qudit after measurement:

$$\sigma_M = \sum_{i \in F_q} |i\rangle \langle i| \sigma |i\rangle \langle i| \tag{290}$$

We will require the following fact:

Fact 8.1 *For all density matrices ρ, σ on 1 qudit,*

$$T(\sigma_M, \rho_M) \leq T(\sigma, \rho) \tag{291}$$

We require a bit more notation before stating the corollary. Recall that the register $\mathcal{V}_{final}$, which is the register of containing the $3mL + m$ qudits held by the verifier at the end of the protocol, is equal to $\mathcal{V}_{L+1} \cup \mathcal{F}$. We will only be interested in the first qudit of register $\mathcal{F}$; this is the qudit which contains the final result of the circuit. For this purpose, we introduce the following notation. For a density matrix σ on $\mathcal{V}_{final}$, let $\sigma' = \mathrm{Tr}_{2,\ldots,m}(\mathrm{Tr}_{\mathcal{V}_{L+1}}(\sigma))$.

Corollary 1.12 *For the polynomial QPIP protocol (Protocol 6.1) with security parameter ϵ (where $\epsilon = \frac{1}{2^{m-1}}$ by definition), assume the verifier aborts with probability at most $1 - \beta$. Then the trace distance between the final measured density matrix conditioned on the verifier's acceptance (σ'_M) and that of the correct measured state (ρ_C) is at most $\frac{2\epsilon}{\beta}$.*

*P*roof of **Corollary1.12**: Recall the final state on register $\mathcal{V}_{final}$ held by the verifier (before the verifier checks for errors, but after he decodes) from Claim 6.3:

$$\rho_{L+1}|\mathcal{V}_{final} = \frac{1}{2^m} \sum_{k \in \{-1,1\}^m} \sigma_k = \frac{1}{2^m} \sum_{k \in \{-1,1\}^m} \sum_{P \in \mathbb{P}_{|\mathcal{V}_{final}|}} \sigma_k^P \tag{292}$$

where

$$\sigma_k^P = \sum_{\Delta_L \in F_q^{|\mathcal{V}_{L+1}|}} \alpha_{P,g(\Delta_L)} \cdot ((|\Delta_L\rangle \langle \Delta_L| \otimes \mathcal{I}_{\mathcal{F}})(D_k^\dagger)^{\otimes |\mathcal{V}_{final}|} P) \sigma^k_{g(\Delta_L)} ((|\Delta_L\rangle \langle \Delta_L| \otimes \mathcal{I}_{\mathcal{F}})(D_k^\dagger)^{\otimes |\mathcal{V}_{final}|} P)^\dagger \tag{293}$$

and $\mathrm{Tr}_{\mathcal{P}_{final}}(\rho^k_{g(\Delta_L)}) = \sigma^k_{g(\Delta_L)}$,

$$\alpha_{P,g(\Delta_L)} = \frac{1}{q^{|\mathcal{P}_{final}|}} \mathrm{Tr}(U^P_{g(\Delta_L)} (\mathcal{I}_{\mathcal{P}_{final}} \otimes \rho_{\mathcal{E}}) (U^P_{g(\Delta_L)})^\dagger) \tag{294}$$

and

$$U_{g(\Delta_L)} = \sum_{P \in \mathbb{P}_{|\mathcal{V}_{final}|}} P \otimes U^P_{g(\Delta_L)} \tag{295}$$

We begin by conditioning on the verifier's acceptance, by applying the projection

$$(\hat{\Pi}_0)_{\mathcal{V}_{final}} \stackrel{\mathrm{def}}{=} (\mathcal{I}^{\otimes d+1} \otimes |0\rangle \langle 0|^{\otimes d})^{\otimes 3L+1} \tag{296}$$

on the state and then re-normalizing. The projection above represents the verifier's test of checking that the last d qudits of each block of m qudits are 0 (for a reminder of why this is the test and how the protocol works, see Protocol 6.1). The resulting state after conditioning on acceptance (where β is the probability of acceptance) is

$$\sigma = \frac{1}{\beta} (\hat{\Pi}_0)_{\mathcal{V}_{final}} (\rho_{L+1}|\mathcal{V}_{final}) (\hat{\Pi}_0)_{\mathcal{V}_{final}} \tag{297}$$

$$= \frac{1}{2^m \beta} \sum_{k \in \{-1,1\}^m} \sum_{P \in \mathbb{P}_{|\mathcal{V}_{final}|}} \gamma_k^P \hat{\sigma}_k^P \tag{298}$$

where

$$\hat{\sigma}_k^P = \frac{1}{\gamma_k^P} (\hat{\Pi}_0)_{\mathcal{V}_{final}} (\sigma_k^P) (\hat{\Pi}_0)_{\mathcal{V}_{final}} \tag{299}$$

and

$$\gamma_k^P = \mathrm{Tr}((\hat{\Pi}_0)_{\mathcal{V}_{final}} (\sigma_k^P)) \tag{300}$$

Our goal is to show that the trace distance $T(\sigma'_M, \rho_C)$ between ρ_C and σ'_M is at most $\frac{2\epsilon}{\beta}$. Note that the expression in equation 298 is a convex sum over density matrices; this is because σ_k^P is an unnormalized density matrix. To see this, observe that σ_k^P (written in equation 293) is a sum over terms of the following form: there is a density matrix ($\sigma^k_{g(\Delta_L)}$), followed by a unitary operation $(D_k^\dagger)^{\otimes|\mathcal{V}_{final}|}P$, followed by a projection. Each term has a non negative coefficient ($\alpha_{P,g(\Delta_L)}$ from equation 294). Therefore, by convexity of trace distance, we can upper bound the trace distance as follows:

$$\begin{aligned} T(\sigma'_M, \rho_C) &\leq \frac{1}{2^m\beta} \sum_{k\in\{-1,1\}^m} \sum_{P\in\mathbb{P}_{|\mathcal{V}_{final}|}} \gamma_k^P T((\hat{\sigma}_k^P)'_M, \rho_C) && (301)\\ &= \frac{1}{\beta} \sum_{P\in\mathbb{P}_{|\mathcal{V}_{final}|}} \frac{1}{2^m} \sum_{k\in\{-1,1\}^m} \gamma_k^P T((\hat{\sigma}_k^P)'_M, \rho_C) && (302)\end{aligned}$$

Next, we will require two claims (which we prove immediately after the current proof). The first claim shows that when P is a trivial Pauli operator, it preserves the correct final state on the first qudit of register $\mathcal{F}$:

Claim 8.2 *For all trivial $P \in \mathbb{P}_{|\mathcal{V}_{final}|}$ (i.e. P consisting of only Z and $\mathcal{I}$ operators):*

$$\frac{1}{2^m} \sum_{k\in\{-1,1\}^m} \gamma_k^P T((\hat{\sigma}_k^P)'_M, \rho_C) = 0 \tag{303}$$

The next claim shows that if P is a non trivial Pauli operator, the trace distance will still be small; intuitively, this is because the state with attack operator P can only pass the verifier's test for 2 (out of 2^m) sign keys:

Claim 8.3 *For all non trivial $P \in \mathbb{P}_{|\mathcal{V}_{final}|}$:*

$$\frac{1}{2^m} \sum_{k\in\{-1,1\}^m} \gamma_k^P T((\hat{\sigma}_k^P)'_M, \rho_C) \leq \frac{1}{2^{m-1}q^{3L}} \sum_{a\in F_q^{3L}} \alpha_{P,a} \tag{304}$$

Given the two claims, we can further simplify the bound in equation 301 by first using Claim 8.2 to remove trivial Pauli operators from the expression (let $P \in \mathbb{P}^T_{|\mathcal{V}_{final}|}$ be the set of all non trivial Pauli operators):

$$\begin{aligned} T(\sigma'_M, \rho_C) &\leq \frac{1}{\beta} \sum_{P\in\mathbb{P}^{NT}_{|\mathcal{V}_{final}|}} \frac{1}{2^m} \sum_{k\in\{-1,1\}^m} \gamma_k^P T((\hat{\sigma}_k^P)'_M, \rho_C) && (305)\\ &\leq \frac{1}{2^{m-1}q^{3L}\beta} \sum_{P\in\mathbb{P}^{NT}_{|\mathcal{V}_{final}|}} \sum_{a\in F_q^{3L}} \alpha_{P,a} && (306)\\ &= \frac{1}{2^{m-1}q^{3L}\beta} \sum_{a\in F_q^{3L}} \sum_{P\in\mathbb{P}^{NT}_{|\mathcal{V}_{final}|}} \alpha_{P,a} && (307)\\ & && (308)\end{aligned}$$

The second inequality follows from Claim 8.3. Next, we can use Lemma 3.3, which provides the following equality:

$$\sum_{P\in\mathbb{P}_{|\mathcal{V}_{final}|}} \alpha_{P,a} = 1 \tag{309}$$

Continuing with the upper bound, we obtain:

$$T(\sigma'_M, \rho_C) \leq \frac{1}{2^{m-1}q^{3L}\beta} \sum_{a \in F_q^{3L}} 1 \tag{310}$$

$$= \frac{1}{2^{m-1}\beta} \tag{311}$$

which completes the proof. □

8.2.1 Proof of Claim 8.2

*P*roof of **Claim 8.2**: We would like to show that for all trivial $P \in \mathbb{P}_{|\mathcal{V}_{final}|}$ (i.e. P consisting of only Z and $\mathcal{I}$ operators):

$$\frac{1}{2^m} \sum_{k \in \{-1,1\}^m} \gamma_k^P T((\hat{\sigma}_k^P)'_M, \rho_C) = 0 \tag{312}$$

To do so, we need to show that for all k,

$$\text{Tr}(|1\rangle\langle 1| (\hat{\sigma}_k^P)') = \text{Tr}(|1\rangle\langle 1| \rho_C) \stackrel{\text{def}}{=} p_1 \tag{313}$$

where p_1 is the probability that the correct state outputs 1 when measured. The reason the above statement is equivalent to equation 312 is because we are considering the measured density matrices; therefore, we need only prove that both density matrices obtain 1 with the same probability to prove that the trace distance is 0.

Recall from equation 299 that:

$$\hat{\sigma}_k^P = \frac{1}{\gamma_k^P}(\hat{\Pi}_0)_{\mathcal{V}_{final}}(\sigma_k^P)(\hat{\Pi}_0)_{\mathcal{V}_{final}} \tag{314}$$

Plugging this in to equation 313, we obtain:

$$\text{Tr}(|1\rangle\langle 1| (\hat{\sigma}_k^P)') = \frac{1}{\gamma_k^P}\text{Tr}((\mathcal{I}^{\otimes|\mathcal{V}_{final}|} \otimes |1\rangle\langle 1| \otimes \mathcal{I}^{\otimes m-1})(\hat{\Pi}_0)_{\mathcal{V}_{final}}(\sigma_k^P)(\hat{\Pi}_0)_{\mathcal{V}_{final}}) \tag{315}$$

$$= \frac{1}{\gamma_k^P}\text{Tr}((\mathcal{I}^{\otimes|\mathcal{V}_{final}|} \otimes |1\rangle\langle 1| \otimes \mathcal{I}^{\otimes m-1})(\hat{\Pi}_0)_{\mathcal{V}_{final}}(\sigma_k^P)) \tag{316}$$

$$= \frac{1}{\gamma_k^P}\text{Tr}(\hat{\Pi}_0\sigma_k^P) \tag{317}$$

where the second equality follows because $(\mathcal{I}^{\otimes|\mathcal{V}_{final}|} \otimes |1\rangle\langle 1| \otimes \mathcal{I}^{\otimes m-1})$ commutes with $(\hat{\Pi}_0)_{\mathcal{V}_{final}}$ and due to the cyclic nature of trace. The third equality follows due to the following equality:

$$\hat{\Pi}_0 = (\mathcal{I}^{\otimes|\mathcal{V}_{final}|} \otimes |1\rangle\langle 1| \otimes \mathcal{I}^{\otimes m-1})(\hat{\Pi}_0)_{\mathcal{V}_{final}} \tag{318}$$

where we recall (from equation 158) that

$$\hat{\Pi}_0 = (\mathcal{I}^{\otimes d+1} \otimes |0\rangle\langle 0|^{\otimes d})^{\otimes 3L}_{\mathcal{V}_{L+1}} \otimes (|1\rangle\langle 1| \otimes \mathcal{I}^{\otimes d} \otimes |0\rangle\langle 0|^{\otimes d})_{\mathcal{F}} \stackrel{\text{def}}{=} (\hat{\Pi}_0)_{\mathcal{V}_{L+1}} \otimes (\hat{\Pi}_0)_{\mathcal{F}} \tag{319}$$

and (from equation 296) that

$$(\hat{\Pi}_0)_{\mathcal{V}_{final}} = (\mathcal{I}^{\otimes d+1} \otimes |0\rangle\langle 0|^{\otimes d})^{\otimes 3L+1} \tag{320}$$

Next, Claim 6.4 gives the following inequality:

$$\frac{1}{\gamma_k^P}\text{Tr}(\hat{\Pi}_0\sigma_k^P) = \frac{1}{\gamma_k^P}\frac{p_1}{q^{3L}}\sum_{a\in F_q^{3L}}\alpha_{P,a} \tag{321}$$

Note that Claim 6.4 is not stated in this format. The first difference is that instead of p_1, the claim has γ; this is simply the probability that the circuit applied by the honest prover outputs 1, which is p_1 in this case. The second is that the statement of the claim also has a summation over k. However, in the proof of the claim, that summation is not used at all; the claim is shown for individual elements of the sum over k, and the sum over k is only used in the statement. Finally, Claim 6.4 has an inequality rather than an equality; see Remark 6.1 for why it is okay to use equality in this setting.

Finally, we claim that $\gamma_k^P = \frac{1}{q^{3L}}\sum_{a\in F_q^{3L}}\alpha_{P,a}$. Given this claim, we have

$$\begin{aligned}
\text{Tr}(|1\rangle\langle 1|(\hat{\sigma}_k^P)') &= \frac{1}{\gamma_k^P}\text{Tr}(\hat{\Pi}_0\sigma_k^P) && (322)\\
&= \frac{1}{\gamma_k^P}\frac{p_1}{q^{3L}}\sum_{a\in F_q^{3L}}\alpha_{P,a} && (323)\\
&= p_1 && (324)
\end{aligned}$$

which completes the proof, since we have proven the equality in equation 313.To see why γ_k^P satisfies the above equality, recall from equation 300 that

$$\gamma_k^P = \text{Tr}((\hat{\Pi}_0)_{\mathcal{V}_{final}}(\sigma_k^P)) \tag{325}$$

We can again apply Claim 6.4 here; the two differences are that we are using $(\hat{\Pi}_0)_{\mathcal{V}_{final}}$ rather than $\hat{\Pi}_0$ and that we require equality rather than inequality. The difference between these two projections is that the latter projects the first qudit of $\mathcal{F}$ onto $|1\rangle\langle 1|$. See Remark 6.1 for why the claim still applies, but with γ in the statement of the claim replaced by 1 and the inequality replaced by equality, which gives

$$\gamma_k^P = \text{Tr}((\hat{\Pi}_0)_{\mathcal{V}_{final}}(\sigma_k^P)) = \frac{1}{q^{3L}}\sum_{a\in F_q^{3L}}\alpha_{P,a} \tag{326}$$

□

8.2.2 Proof of Claim 8.3

Proof of **Claim 8.3**: We would like to show that for all non trivial $P \in \mathbb{P}_{|\mathcal{V}_{final}|}$:

$$\frac{1}{2^m}\sum_{k\in\{-1,1\}^m}\gamma_k^P T((\hat{\sigma}_k^P)'_M, \rho_C) \leq \frac{1}{2^{m-1}q^{3L}}\sum_{a\in F_q^{3L}}\alpha_{P,a} \tag{327}$$

First, we use Fact 8.1 and then upper bound the trace distance by 1 to obtain:

$$\begin{aligned}
\frac{1}{2^m}\sum_{k\in\{-1,1\}^m}\gamma_k^P T((\hat{\sigma}_k^P)'_M, \rho_C) &\leq \frac{1}{2^m}\sum_{k\in\{-1,1\}^m}\gamma_k^P T((\hat{\sigma}_k^P)', \rho_C) && (328)\\
&\leq \frac{1}{2^m}\sum_{k\in\{-1,1\}^m}\gamma_k^P && (329)
\end{aligned}$$

Next, note that by the definition of γ_k^P in equation 300

$$\frac{1}{2^m}\sum_{k\in\{-1,1\}^m}\gamma_k^P=\frac{1}{2^m}\sum_{k\in\{-1,1\}^m}\mathrm{Tr}((\hat{\Pi}_0)_{\mathcal{V}_{final}}\sigma_k^P)\leq\frac{1}{2^{m-1}q^{3L}}\sum_{a\in F_q^{3L}}\alpha_{P,a} \tag{330}$$

where the final inequality follows from Claim 6.5. Note that in Claim 6.5 the projection is $\hat{\Pi}_0$ rather than $(\hat{\Pi}_0)_{\mathcal{V}_{final}}$. Recall (from equation 158) that

$$\hat{\Pi}_0=(\mathcal{I}^{\otimes d+1}\otimes|0\rangle\langle 0|^{\otimes d})_{\mathcal{V}_{L+1}}^{\otimes 3L}\otimes(|1\rangle\langle 1|\otimes\mathcal{I}^{\otimes d}\otimes|0\rangle\langle 0|^{\otimes d})_{\mathcal{F}}\stackrel{\mathrm{def}}{=}(\hat{\Pi}_0)_{\mathcal{V}_{L+1}}\otimes(\hat{\Pi}_0)_{\mathcal{F}} \tag{331}$$

and (from equation 296)

$$(\hat{\Pi}_0)_{\mathcal{V}_{final}}=(\mathcal{I}^{\otimes d+1}\otimes|0\rangle\langle 0|^{\otimes d})^{\otimes 3L+1} \tag{332}$$

The difference between the two is that the first qudit of register $\mathcal{F}$ is projected onto $|1\rangle\langle 1|$ in $\hat{\Pi}_0$ and onto $\mathcal{I}$ in $(\hat{\Pi}_0)_{\mathcal{V}_{final}}$. However, the proof of Claim 6.5 holds if $\hat{\Pi}_0$ is replaced with $(\hat{\Pi}_0)_{\mathcal{V}_{final}}$; see Remark 6.2. □

9 Acknowledgements

D.A. thanks Oded Goldreich, Madhu sudan and Guy Rothblum for exciting and inspiring conversations that eventually led to this work. E.E. thanks Avinatan Hassidim for stimulating and refining ideas, particularly about fault tolerance. We also thank Gil Kalai, David DiVincenzo and Ari Mizel, for stimulating questions and clarifications, and Daniel Gottesman for many helpful ideas and remarks, and in particular, for his help in proving Theorem 1.4.

References

[AA06] D. Aharonov and I. Arad. The BQP-hardness of approximating the Jones Polynomial. *Arxiv preprint quant-ph/0605181*, 2006.

[Aar09] S. Aaronson. BQP and the Polynomial Hierarchy. *Arxiv preprint quant-ph/0910.4698*, 2009.

[AAV13] D. Aharonov, I. Arad, and T. Vidick. The quantum PCP conjecture. *ACM SIGACT News*, 44:47–49, 2013.

[AB09] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge Univ. Press, 2009.

[ABO97] D. Aharonov and M. Ben-Or. Fault-tolerant quantum computation with constant error. *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 176–188, 1997.

[ABOE08] D. Aharonov, M. Ben-Or, and E. Eban. Interactive Proofs For Quantum Computations. *Arxiv preprint arXiv:0810.5375*, 2008.

[ABW08] A. Ambainis, J. Bouda, and A. Winter. Tamper-resistant encryption of quantum information. *Arxiv preprint arXiv:0808.0353*, 2008.

[AE07] A. Ambainis and J. Emerson. Quantum t-designs: t-wise independence in the quantum world. *Arxiv preprint quant-ph/0701126*, 2007.

[AFK87] M. Abadi, J. Feigenbaum, and J. Kilian. On hiding information from an oracle. In *Proceedings of the nineteenth annual ACM conference on Theory of computing*, pages 195–203. ACM New York, NY, USA, 1987.

[AJL06] D. Aharonov, V. Jones, and Z. Landau. A polynomial quantum algorithm for approximating the Jones polynomial. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 427–436. ACM New York, NY, USA, 2006.

[AS06] P. Arrighi and L. Salvail. Blind Quantum Computation. *International Journal of Quantum Information*, 4(5):883–898, 2006.

[ASMZ17] D. Aharonov, F. Song, U. Mahadev, and J. Zhengfeng. Blind or verifiable fault tolerant delegated quantum computation. *In progress*, 2017.

[AV12] D. Aharonov and U. Vazirani. Is Quantum Mechanics Falsifiable? A computational perspective on the foundations of Quantum Mechanics. *Arxiv preprint arXiv:1206.3686*, 2012.

[BCG+02] H. Barnum, C. Crépeau, D. Gottesman, A. Smith, and A. Tapp. Authentication of Quantum Messages. *Proceedings of the 43rd Symposium on Foundations of Computer Science*, pages 449–458, 2002.

[BFK08] A. Broadbent, J. Fitzsimons, and E. Kashefi. Universal blind quantum computation. *Arxiv preprint arXiv:0807.4154*, 2008.

[BFKW13] S. Barz, J.F. Fitzsimons, E. Kashefi, and P. Walther. Demonstration of measurement-only blind quantum computing. *Nature Physics 9, 727*, 2013.

[BFLW09] M. Bordewich, M. Freedman, L. Lovász, and D. Welsh. Approximate Counting and Quantum Computation. *Arxiv preprint 0908.2122*, 2009.

[BGS12] A. Broadbent, G. Gutoski, and D. Stebila. Quantum one-time programs. *Arxiv preprint arXiv:1211.1080*, 2012.

[BJ14] A. Broadbent and S. Jeffery. Quantum homomorphic encryption for circuits of low T-gate complexity. *Arxiv preprint arXiv:1412.8766*, 2014.

[BK05] S. Bravyi and A. Kitaev. Universal quantum computation with ideal Clifford gates and noisy ancillas. *Physical Review A*, 71(2):22316, 2005.

[BKB+12] S. Barz, E. Kashefi, A. Broadbent, J. F. Fitzsimons, A. Zeilinger, and P. Walther. Demonstration of blind quantum computing. *Science 335, 303*, 2012.

[BOCG+06] M. Ben-Or, C. Crépeau, D. Gottesman, A. Hassidim, and A. Smith. Secure Multiparty Quantum Computation with (Only) a Strict Honest Majority. *Foundations of Computer Science, 2006. FOCS'05. 47th Annual IEEE Symposium on*, pages 249–260, 2006.

[Bro15] A. Broadbent. How to Verify a Quantum Computation. *Arxiv preprint arXiv:1509.09180*, 2015.

[Chi01] A.M. Childs. Secure assisted quantum computation. *Arxiv preprint quant-ph/0111046*, 2001.

[DLT02] D. DiVincenzo, D.W. Leung, and B.M. Terhal. Quantum data hiding. *IEEE Trans. Inf. Th., 48(3):580599*, 2002.

[DSS16] Y. Dulek, C. Schaffner, and F. Speelman. Quantum homomorphic encryption for polynomial-sized circuits. *Arxiv preprint arXiv:1603.09717*, 2016.

[FK12] J. Fitzsimons and E. Kashefi. Unconditionally verifiable blind computation. *Arxiv preprint arXiv:1203.5217*, 2012.

[FKLW01] M. Freedman, A. Kitaev, M. Larsen, and Z. Wang. Topological Quantum Computation. *Arxiv preprint quant-ph/0101025*, 2001.

[GKW15] A. Gheorghiu, E. Kashefi, and P. Wallden. Robustness and device independence of verifiable blind quantum computing. *New Journal of Physics 17, 083040*, 2015.

[GMR85] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems. In *Proceedings of the seventeenth annual ACM symposium on Theory of computing*, pages 291–304. ACM New York, NY, USA, 1985.

[Got04] Daniel Gottesman, 2004. As referenced in [`http://www.scottaaronson.com/blog/?p=284`; accessed 13-Apr-2017].

[GRB+16] C. Greganti, MC. Roehsner, s. Barz, T. Morimae, and P. Walther. Demonstration of measurement-only blind quantum computing. *New Journal of Physics 18, 013020*, 2016.

[HH16] M. Hayashi and M. Hajdušek. Self-guaranteed measurement-based quantum computation. *Arxiv preprint arXiv:1603.02195*, 2016.

[HM15] M. Hayashi and T. Morimae. Verifiable Measurement-Only Blind Quantum Computating with Stabilizer Testing. *Physical Review Letters 115*, 2015.

[HPDF15] M. Hajdušek, C. Pérez-Delgado, and J. Fitzsimons. Device-Independent Verifiable Blind Quantum Computation. *Arxiv preprint arXiv:1502.02563*, 2015.

[KSV02] A.Y. Kitaev, A. Shen, and M.N. Vyalyi. *Classical and Quantum Computation*. American Mathematical Society, 2002.

[Mck13] M. Mckague. Interactive proofs for BQP via self-tested graph states. *Arxiv preprint arXiv:1309.5675*, 2013.

[MF16] T. Morimae and J. Fitzsimons. Post hoc verification with a single prover. *Arxiv preprint arXiv:1603.06046*, 2016.

[Mor14] T. Morimae. Verification for measurement-only blind computation. *Physical Review A 89*, 2014.

[Roo03] A. Roodman. Blind Analysis in Particle Physics. In *Statistical Problems in Particle Physics, Astrophysics, and Cosmology, Proceedings of the PHYSTAT 2003 Conference held 8-11 September, 2003 at the Stanford Linear Accelerator Center. SLAC eConf C030908.* `http://www.slac.stanford.edu/econf/C030908.`*, p. 166*, 2003.

[RUV12] B. Reichardt, F. Unger, and U. Vazirani. A classical leash for a quantum system. *Arxiv preprint arXiv:1209.0448*, 2012.

[Sho96] P. Shor. Fault-tolerant quantum computation. *Arxiv preprint quant-ph/9605011*, 1996.

[Sho97] PW Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM journal on computing(Print)*, 26(5):1484–1509, 1997.

[TFMI16] Y. Takeuchi, K. Fujii, T. Morimae, and N. Imoto. Practically verifiable blind quantum computation with acceptance rate amplification. *Arxiv preprint arXiv:1607.01568*, 2016.

[Vaz07] Umesh Vazirani, 2007. Talk given in a conference in Japan.

[Wat03] J. Watrous. PSPACE has constant-round quantum interactive proof systems. *Theoretical Computer Science*, 292(3):575–588, 2003.

[Wik08] Wikipedia. Blind experiment — Wikipedia, the free encyclopedia, 2008. [`https://en.wikipedia.org/wiki/Blinded_experiment`; accessed 20-Oct-2008].

[Yaa08] Jonathan Yaari. Preprint: *Interactive Proofs as a Theory of Confirmation*. PhD thesis, The Hebrew University of Jerusalem, 2008.

[YPDF14] L. Yu, C. Perez-Delgado, and J. Fitzsimons. Limitations on information theoretically secure quantum homomorphic encryption. *Arxiv preprint arXiv:1406.2456*, 2014.

A A Symmetric Definition of QPIP

Here we provide the definition of $\mathsf{QPIP}_\kappa^{sym}$ and then prove Corollary 1.3 and Corollary 1.13. We begin with the definition of $\mathsf{QPIP}_\kappa^{sym}$:

Definition A.1 *A language $\mathcal{L}$ is in the class symmetric quantum prover interactive proof ($\mathsf{QPIP}_\kappa^{sym}$) with completeness c and soundness s (where $c - s$ is constant) if there exists an interactive protocol with the following properties:*

- *The prover $\mathbb{P}$ and verifier $\mathbb{V}$ are exactly the same as in the definition of QPIP_κ (Definition 1.1). Namely, a BQP machine and quantum-classical hybrid machine for the prover and verifier respectively.*
- *Communication is identical to the QPIP_κ definition.*
- *The verifier has three possible outcomes: **1**, **0**, and **ABORT**:*
  - ***1**: The verifier is convinced that $x \in \mathcal{L}$.*
  - ***0**: The verifier is convinced that $x \notin \mathcal{L}$.*
  - ***ABORT**: The verifier caught the prover cheating.*
- ***Completeness**: $\forall x \in \{0,1\}^*$, after interacting with $\mathbb{P}$, the verifier's outcome is correct with high probability:*

$$\Pr_r([\mathbb{V},\mathbb{P}](x,r) = \mathbb{1}_{\mathcal{L}}) \geq c$$

  where $\mathbb{1}_{\mathcal{L}}$ is the indicator function of $\mathcal{L}$, r represents the randomness used by the verifier, and $[\mathbb{V},\mathbb{P}](x,r)$ is the verifier's outcome after using randomness r and interacting with $\mathbb{P}$ on input x.
- ***Soundness**: For all provers $\mathbb{P}'$ (with the same description as $\mathbb{P}$) and for* **all** *$x \in \{0,1\}^*$, the verifier is mistaken with bounded probability, that is:*

$$\Pr_r([\mathbb{V},\mathbb{P}](x,r) = 1 - \mathbb{1}_{\mathcal{L}}) \leq s$$

We now prove Corollary 1.3:

*P*roof of **Corollary1.3**: We will prove that $\mathsf{QPIP}_c = \mathsf{QPIP}_c^{sym}$. It follows from the definitions of QPIP_c (Definition 1.1) and QPIP_c^{sym} (Definition A.1) that $\mathsf{QPIP}_c \subseteq \mathsf{QPIP}_c^{sym}$. We obtain the other direction by showing that for any language $\mathcal{L}$, if $\mathcal{L}$ is in QPIP_c then $\mathcal{L}, \mathcal{L}^c \in \mathsf{QPIP}_c^{sym}$. First note that if $\mathcal{L}$ is in QPIP_c, then so is $\mathcal{L}^c$, since BQP is closed under complement and $\mathsf{BQP} = \mathsf{QPIP}_c$ by Theorem 1.2. Let $\mathbb{V}_{\mathcal{L}}, \mathbb{P}_{\mathcal{L}}$ denote the QPIP_c verifier and prover for the language $\mathcal{L}$. By the assumption, there exists such a pair for both $\mathcal{L}$ and $\mathcal{L}^c$. We define the pair $\widetilde{\mathbb{P}}$ and $\widetilde{\mathbb{V}}$ to be QPIP_c^{sym} verifier and prover in the following way: on the first round the prover $\widetilde{\mathbb{P}}$ sends to $\widetilde{\mathbb{V}}$ "yes" if $x \in \mathcal{L}$ and "no" otherwise. Now, both $\widetilde{\mathbb{P}}$ and $\widetilde{\mathbb{V}}$ behave according to $\mathbb{V}_{\mathcal{L}}, \mathbb{P}_{\mathcal{L}}$ if "yes" was sent or according to $\mathbb{V}_{\mathcal{L}^c}, \mathbb{P}_{\mathcal{L}^c}$ otherwise. Soundness and completeness follow immediately from the definition.

□

Finally, we prove Corollary 1.13:

*P*roof of **Corollary1.13**: This corollary uses QPIP^{sym} rather than QPIP (recall from Corollary 1.3 that $\mathsf{QPIP}^{sym} = \mathsf{BQP}$). First note that if we run QPIP^{sym} (either the polynomial based or Clifford based protocol) on an instance x drawn from D and the verifier does not abort with probability β, the probability that the verifier outputs the incorrect answer is at most $\frac{2\epsilon}{\beta} + \gamma$, by Corollary 1.11 and Corollary 1.12. We need to amplify this probability so that the verifier outputs the incorrect answer with probability which is at most inverse exponential in n. If we can do so, the corollary follows since any BPP machine would err with non-negligible probability on

instances drawn from D, by assumption. Therefore, the prover cannot be simulated by a BPP machine.

To amplify the probability of outputting an incorrect answer, we run QPIP^{sym} polynomially many times (in n) and take the output to be the majority of the output values, ignoring runs on which the verifier aborted. Since β is constant, if we repeat the protocol polynomially many times, we expect to collect polynomially many output values (we fail to do so with probability p which is inverse exponential n). Since each output is correct with probability $1 - (\frac{2\epsilon}{\beta} + \gamma) > \frac{1}{2}$, by taking the majority of these output values, we can reduce the error of the output to be p', which is inverse exponential in n. The overall probability of error is then $(1-p)p' + p$, which is inverse exponential in n. □

B Clifford and Pauli Operators

Here are some useful lemmas about Clifford/Pauli operators. We first prove the Pauli mixing lemma:
*P*roof of **Lemma 4.5 (Pauli Mixing)**: First, we write ρ as:

$$\sum_{ij} |i\rangle \langle j|_A \otimes \rho_{ij}$$

It follows that:

$$\mathrm{Tr}_A(\rho) = \sum_i \rho_{ii}$$

Next, observe that:

$$\sum_{P \in \mathbb{P}_n} P |i\rangle \langle j| P^\dagger = \sum_{zx} Z^z X^x |i\rangle \langle j| (Z^z X^x)^\dagger \quad (333)$$

$$= \sum_{zx} \omega_q^{z(i-j)} X^x |i\rangle \langle j| (X^x)^\dagger \quad (334)$$

This expression is 0 if $i \neq j$. If $i = j$, we obtain $q^n \mathcal{I}$. Plugging in this observation to the above expression, we have:

$$\frac{1}{|\mathbb{P}_n|} \sum_{P \in \mathbb{P}_n} (P \otimes \mathcal{I}_B)\rho(P \otimes \mathcal{I}_B)^\dagger = \frac{1}{|\mathbb{P}_n|} \sum_{ij} \sum_{P \in \mathbb{P}_n} P |i\rangle \langle j|_A P^\dagger \otimes \rho_{ij} \quad (335)$$

$$= \frac{1}{|\mathbb{P}_n|} \sum_{i} \sum_{P \in \mathbb{P}_n} P |i\rangle \langle i|_A P^\dagger \otimes \rho_{ij} \quad (336)$$

$$= \frac{q^n}{|\mathbb{P}_n|} \mathcal{I} \otimes \sum_i \rho_{ii} \quad (337)$$

$$= \frac{1}{q^n} \mathcal{I} \otimes \mathrm{Tr}_A(\rho) \quad (338)$$

□

Now we prove the Clifford mixing lemma:
*P*roof of **Lemma 4.4 (Clifford Mixing)**: To prove this lemma, we observe that applying a random Clifford includes applying a random Pauli, and the lemma then follows from Lemma 4.5. In more detail, we have the following equality for all $Q \in \mathbb{P}_n$:

$$\frac{1}{|\mathfrak{C}_n|} \sum_{C \in \mathfrak{C}_n} (C \otimes \mathcal{I}_B)\rho(C \otimes \mathcal{I}_B)^\dagger = \frac{1}{|\mathfrak{C}_n|} \sum_{C \in \mathfrak{C}_n} (CQ \otimes \mathcal{I}_B)\rho(CQ \otimes \mathcal{I}_B)^\dagger \quad (339)$$

Now we have:

$$\frac{1}{|\mathfrak{C}_n|}\sum_{C\in\mathfrak{C}_n}(C\otimes\mathcal{I}_B)\rho(C\otimes\mathcal{I}_B)^\dagger \;=\; \frac{1}{|\mathfrak{C}_n||\mathbb{P}_n|}\sum_{C\in\mathfrak{C}_n}\sum_{Q\in\mathbb{P}_n}(CQ\otimes\mathcal{I}_B)\rho(CQ\otimes\mathcal{I}_B)^\dagger \tag{340}$$

Regrouping terms, we have

$$\ldots \;=\; \frac{1}{|\mathfrak{C}_n|}\sum_{C\in\mathfrak{C}_n}(C\otimes\mathcal{I}_B)(\frac{1}{|\mathbb{P}_n|}\sum_{Q\in\mathbb{P}_n}(Q\otimes\mathcal{I}_B)\rho(Q\otimes\mathcal{I}_B)^\dagger)(C\otimes\mathcal{I}_B)^\dagger \tag{341}$$

By Lemma 4.5 (with $q=2$) the above expression is equal to:

$$\ldots \;=\; \frac{1}{|\mathfrak{C}_n|2^n}\sum_{C\in\mathfrak{C}_n}(C\otimes\mathcal{I}_B)(\mathcal{I}\otimes\mathrm{Tr}_A(\rho))(C\otimes\mathcal{I}_B)^\dagger \tag{342}$$

$$=\; \frac{1}{2^n}\mathcal{I}\otimes\mathrm{Tr}_A(\rho) \tag{343}$$

□

C Clifford Technical Details

Here we prove Lemma 3.4, Lemma 3.5 and Lemma 3.6, which were used to prove Lemma 3.2 (and Lemma 3.5 is also used to prove Lemma 5.1).

Lemma 3.4 (Pauli Partitioning by Cliffords) *For every $P,Q\in\mathbb{P}_m\backslash\{\mathcal{I}\}$ it holds that : $\left|\left\{C\in\mathfrak{C}_m|C^\dagger PC=Q\right\}\right| = \frac{|\mathfrak{C}_m|}{|\mathbb{P}_m|-1}=\frac{|\mathfrak{C}_m|}{4^m-1}$.*

Proof of **Lemma 3.4**: We first claim that for every $Q,P\in\mathbb{P}_m\setminus\mathcal{I}$ there exists $D\in\mathfrak{C}_m$ such that $D^\dagger PD=Q$. We will prove this claim by induction. Specifically, we show that starting from any non identity Pauli operator one can, using conjunction by Clifford group operator reach the Pauli operator $X\otimes\mathcal{I}^{\otimes m-1}$.

We first notice that the swap operation is in $\mathfrak{C}_2$ since it holds that:

$$SWAP_{k,k+1} \;=\; CNOT_{k\to(k+1)}CNOT_{(k+1)\to k}CNOT_{k\to(k+1)} \tag{344}$$

Furthermore, we recall that $K^\dagger(XZ)K\;\propto\;X$ and $H^\dagger ZH\;=\;X$. Therefore, any non identity Pauli $P=P_1\otimes\ldots\otimes P_m$ can be transformed using $SWAP, H$ and K to the form: $X^{\otimes k}\otimes\mathcal{I}^{\otimes m-k}$ (up to a phase and for some $k\geq 1$). To conclude we use:

$$CNOT^\dagger_{1\to2}(X_1\otimes X_2)CNOT_{1\to2} \;=\; X\otimes\mathcal{I} \tag{345}$$

which reduces the number of X operations at hand. Applying this sufficiently many times results in reaching the desired form. Since this holds for any non-identity Pauli operators: P,Q we know there are $C,D\in\mathfrak{C}_m$ such that:

$$X\otimes\mathcal{I}^{\otimes m-1} \;=\; C^\dagger PC=D^\dagger QD \tag{346}$$

$$\Rightarrow\; DC^\dagger PCD^\dagger=Q \tag{347}$$

therefore $CD^\dagger$ is the operator we looked for.

Given $P'\in\mathbb{P}_m\setminus\mathcal{I}$, define $A_{P',Q}$ as follows $A_{P',Q}\stackrel{\text{def}}{=}\left\{C\in\mathfrak{C}_m|C^\dagger P'C=Q\right\}$. We will show that $|A_{P',Q}|$ is independent of P'. Now fix $P'\in\mathbb{P}_m\setminus\mathcal{I}$ and let $D\in\mathfrak{C}_m$ be one of the operators for which the following equality holds: $D^\dagger P'D=Q$. Then it holds that for all $Q'\in\mathbb{P}_m\setminus\mathcal{I}$, $D^\dagger C\in A_{Q',Q}\iff C\in A_{P',Q'}$. Therefore

$|A_{Q',Q}| = |A_{P',Q'}|$ for all non identity P', Q' and Q. Using the fact that $|A_{P',Q'}| = |A_{Q',P'}|$, it follows that $|A_{P',Q'}|$ is independent of Q' and P'.

Now note that the sets $\{A_{P',Q'} : \forall P'\}$ form a partition of $\mathfrak{C}_m$. These sets clearly do not intersect. Observe that for each $C \in \mathfrak{C}_m$, there exists $P' \in \mathbb{P}_m \setminus \mathcal{I}$ such that $P' = CQ'C^\dagger$. Since all the sets in the partition have the same size, we obtain:

$$|\mathfrak{C}_m| = \sum_{P' \in \mathbb{P}_m \setminus \mathcal{I}} \left|A_{P',Q}\right| = \quad (4^m - 1)\,|A_{P,Q}| \tag{348}$$

which concludes the proof. □

Lemma 3.5 (Pauli Twirl) *Let $P \neq P'$ be generalized Pauli operators. For any density matrix ρ' on $m' > m$ qubits it holds that*

$$\sum_{Q \in \mathbb{P}_m} (Q^\dagger P Q \otimes \mathcal{I}) \rho' (Q^\dagger (P')^\dagger Q \otimes \mathcal{I}) = 0$$

*P*roof of **Lemma 3.5**: Let $P \neq P'$ be generalized Pauli operator $P = X^a Z^b$ and $P' = X^{a'} Z^{b'}$.

$$\sum_{Q \in \mathbb{P}_m} (Q^\dagger P Q \otimes \mathcal{I}) \rho' (Q^\dagger P'^\dagger Q \otimes \mathcal{I}) \quad = \quad \sum_{d,c=0}^{q-1} ((X^c Z^d)^\dagger X^a Z^b (X^c Z^d) \otimes \mathcal{I}) \rho' ((X^c Z^d)^\dagger (X^{a'} Z^{b'})^\dagger (X^c Z^d) \otimes \mathcal{I})$$

We use the fact that $Z^d X^c = \omega_q^{dc} X^c Z^d$ (see Definition 2.2) and some algebra:

$$\ldots \quad = \quad \sum_{d,c=0}^{q-1} \omega_q^{d(a-a')+c(b-b')} (X^a Z^b \otimes \mathcal{I}) \rho' (Z^{-b'} X^{-a'} \otimes \mathcal{I}) \tag{349}$$

$$= \quad (X^a Z^b \otimes \mathcal{I}) \rho' (Z^{-b'} X^{-a'} \otimes \mathcal{I}) \sum_{c=0}^{q-1} \omega_q^{c(b-b')} \sum_{d=0}^{q-1} \omega_q^{d(a-a')} \tag{350}$$

To conclude the proof we recall that $a \neq a'$ or $b \neq b'$, hence one of the above sums vanishes.

□

Lemma 3.6 (Clifford Twirl) *Let $P \neq P'$ be Pauli operators. For any density matrix ρ' on $m' > m$ qubits it holds that*

$$\sum_{C \in \mathfrak{C}_m} (C^\dagger P C \otimes \mathcal{I}) \rho' (C^\dagger (P')^\dagger C \otimes \mathcal{I}) = 0$$

*P*roof of **Lemma 3.6**: Notice that applying a random Clifford operator "includes" the application of a random Pauli:

$$\sum_{c \in \mathfrak{C}_m} (C^\dagger P C \otimes \mathcal{I}) \rho' (C^\dagger (P')^\dagger C \otimes \mathcal{I}) \quad = \quad \sum_{c \in \mathfrak{C}_m} ((CQ)^\dagger P (CQ) \otimes \mathcal{I}) \rho' ((CQ)^\dagger (P')^\dagger (CQ) \otimes \mathcal{I}) \tag{351}$$

Equality holds for any $Q \in \mathfrak{C}_n$ since it is nothing but a change of order of summation.

$$\ldots \quad = \quad \sum_{Q \in \mathbb{P}_m} \frac{1}{|\mathbb{P}_m|} \sum_{c \in \mathfrak{C}_m} ((CQ)^\dagger P (CQ) \otimes \mathcal{I}) \rho' ((CQ)^\dagger (P')^\dagger (CQ) \otimes \mathcal{I}) \tag{352}$$

$$= \quad \sum_{c \in \mathfrak{C}_m} \frac{1}{|\mathbb{P}_m|} \sum_{Q \in \mathbb{P}_m} ((CQ)^\dagger P (CQ) \otimes \mathcal{I}) \rho' ((CQ)^\dagger (P')^\dagger (CQ) \otimes \mathcal{I}) \tag{353}$$

$$= \quad \sum_{c \in \mathfrak{C}_m} \frac{1}{|\mathbb{P}_m|} \sum_{Q \in \mathbb{P}_m} (Q^\dagger (C^\dagger P C) Q \otimes \mathcal{I}) \rho' (Q^\dagger (C^\dagger (P')^\dagger C) Q \otimes \mathcal{I}) \tag{354}$$

By Lemma 3.5 (with $q = 2$), we know that this expression is 0 if $C^\dagger PC \neq C^\dagger P'C$. □

D Logical Gates on Signed Polynomial Codes

In this section we prove that the logical operators given in Section 2.5.1 behave as claimed. We first prove that the logical X operator is correct.

*P*roof of **Claim 2.1**: We can easily verify that applying $X^{k_1 x} \otimes \cdots \otimes X^{k_m x}$ is the logical $\widetilde{X}_k^x$ operation:

$$\begin{aligned}\widetilde{X}_k^x \left|S_a^k\right\rangle =&(X^{k_1 x} \otimes \cdots \otimes X^{k_m x})\frac{1}{\sqrt{q^d}} \sum_{f:def(f)\leq d, f(0)=a} |k_1 f(\alpha_1), \ldots, k_m f(\alpha_m)\rangle \\ =&\frac{1}{\sqrt{q^d}} \sum_{f:def(f)\leq d, f(0)=a} |k_1(f(\alpha_1)+x), \ldots, k_m(f(\alpha_m)+x)\rangle\end{aligned} \tag{355}$$

Setting $f'(\alpha) = f(\alpha) + x$:

$$\begin{aligned}\ldots =&\frac{1}{\sqrt{q^d}} \sum_{f':deg(f')\leq d, f'(0)=a+1} \left|k_1 f'(\alpha_1), \ldots, k_m f'(\alpha_m)\right\rangle \\ =&\left|S_{a+x}^k\right\rangle\end{aligned} \tag{356}$$

□

We now prove that the logical *SUM* operator is correct.

*P*roof of **Claim 2.2**:

$$\begin{aligned}\widetilde{SUM}\,|S_a\rangle\,|S_b\rangle =& (SUM)^{\otimes m}\frac{1}{q^d} \sum_{f(0)=a} |k_1 f(\alpha_1), \ldots, k_m f(\alpha_m)\rangle \sum_{h(0)=b} |k_1 h(\alpha_1), \ldots, k_m h(\alpha_m)\rangle \\ =&\frac{1}{q^d} \sum_{f(0)=a, h(0)=b} |k_1 f(\alpha_1), \ldots, k_m f(\alpha_m)\rangle\,|k_1(h(\alpha_1)+f(\alpha_1)), \ldots, k_m(h(\alpha_m)+f(\alpha_m))\rangle\end{aligned} \tag{357}$$

We set $g(\alpha) = f(\alpha) + h(\alpha)$

$$\begin{aligned}\ldots =&\frac{1}{q^d} \sum_{f(0)=a, g(0)=a+b} |k_1 f(\alpha_1), \ldots, k_m f(\alpha_m)\rangle\,|k_1 g(\alpha_1), \ldots, k_m g(\alpha_m)\rangle \\ =&\left|S_a^k\right\rangle \left|S_{a+b}^k\right\rangle\end{aligned} \tag{358}$$

□

We proceed to the proof of the lemma needed for the logical Fourier transform.

*P*roof of **Lemma 2.3**: A polynomial p of degree $\leq m-1$ is completely determined by it's values in the points α_i. We write p as in the form of the Lagrange interpolation polynomial: $f(x) = \sum_i \prod_{j\neq i} \frac{x-\alpha_j}{\alpha_i-\alpha_j} f(\alpha_j)$. Therefore, we set $c_i = \prod_{j\neq i} \frac{-\alpha_j}{\alpha_i-\alpha_j}$ and notice that it is independent of p, and the claim follows. □

We continue to the proof of the logical Fourier transform.

*P*roof of **Claim 2.4**: We denote $|kf\rangle = |k_1 f(\alpha_1), \ldots, k_m f(\alpha_m)\rangle$

$$F_{c_1} \otimes F_{c_2} \ldots \otimes F_{c_m} \left|S_a^k\right\rangle = q^{-d/2} F_{c_1} \otimes F_{c_2} \otimes \ldots \otimes F_{c_m} \sum_{f:def(f)\leq d, f(0)=a} |kf\rangle \tag{359}$$

$$= q^{-d/2} q^{-m/2} \sum_{f:def(f)\leq d, f(0)=a} \sum_{b_1,\ldots,b_m} \omega_q^{\sum_i c_i k_i f(\alpha_i) b_i} |b_1, \ldots, b_m\rangle \tag{360}$$

We think of the b_i's as defining a signed polynomial g of degree $\leq m-1$ that is $k_i g(\alpha_i) = b_i$ and split the sum according to $g(0)$:

$$\begin{aligned} \ldots &= q^{-(m+d)/2} \sum_{\substack{f:deg(f)\leq d \\ f(0)=a}} \sum_{b} \sum_{\substack{g:deg(g)\leq m-1 \\ g(0)=b}} \omega_q^{\sum_i c_i k_i f(\alpha_i) k_i g(\alpha_i)} |kg\rangle \quad (361) \\ &= q^{-(m+d)/2} \sum_{\substack{f:deg(f)\leq d \\ f(0)=a}} \sum_{b} \sum_{\substack{g:deg(g)\leq m-1 \\ g(0)=b}} \omega_q^{\sum_i c_i f(\alpha_i) g(\alpha_i)} |kg\rangle \quad (362) \end{aligned}$$

We temporarily restrict our view to polynomials g with degree at most $m-d-1$ and therefore the polynomial fg has degree at most $m-1$. We use Lemma 2.3 on fg:

$$\sum_{i=1}^{m} c_i (fg)(\alpha_i) \quad = fg(0) \quad = ab \quad (363)$$

Going back to Eq. 362:

$$q^{-(m+d)/2} \sum_{f,g} \sum_{b\in F_q} \omega_q^{\sum_i c_i (fg)(\alpha_i)} |kg\rangle \quad = \quad q^{-(m+d)/2} \sum_{b\in F_q} \sum_{f,g} \omega_q^{ab} |kg\rangle \quad (364)$$

Where the summation is over all f, g such that $f(0) = a$ and $g(0) = b$ while the degrees of f and g are at most d and $m-d-1$ respectively.

The sum does not depend on f and there are exactly q^d polynomials f in the sum, therefore, we can write the expression as :

$$\begin{aligned} \ldots = \quad & q^{-(m+d)/2} \sum_{b\in F_q} q^d \sum_{g} \omega_q^{ab} |kg\rangle \\ = \quad & \frac{1}{\sqrt{q}} \sum_{b\in F_q} \omega_q^{ab} \frac{1}{\sqrt{q^{m-d-1}}} \sum_{g:deg(g)\leq m-d-1, g(0)=b} |kg\rangle \quad (365) \\ = \quad & \frac{1}{\sqrt{q}} \sum_{b\in F_q} \omega_q^{ab} \left|\widetilde{S_b^k}\right\rangle \end{aligned}$$

Since the above expression has norm 1, if follows that the coefficients that we temporally ignored at Eq. 361 all vanish. □

Finally, we prove that the logical Z operator is correct.

*P*roof of **Claim 2.5**:

$$\begin{aligned} \widetilde{Z}_k^z \left|S_a^k\right\rangle &= (Z^{k_1 c_1 z} \otimes \cdots \otimes Z^{k_m c_m z}) \frac{1}{\sqrt{q^d}} \sum_{f:deg(f)\leq d, f(0)=a} |k_1 f(\alpha_1), \ldots, k_m f(\alpha_m)\rangle \quad (366) \\ &= \frac{1}{\sqrt{q^d}} \sum_{f:deg(f)\leq d, f(0)=a} \omega_q^{\sum_i k_i c_i z k_i f(\alpha_i)} |k_1 f(\alpha_1), \ldots, k_m f(\alpha_m)\rangle \quad (367) \\ &= \frac{1}{\sqrt{q^d}} \sum_{f:deg(f)\leq d, f(0)=a} \omega_q^{z \sum_i c_i f(\alpha_i)} |k_1 f(\alpha_1), \ldots, k_m f(\alpha_m)\rangle \quad (368) \\ &= \frac{1}{\sqrt{q^d}} \sum_{f:deg(f)\leq d, f(0)=a} \omega_q^{z f(0)} |k_1 f(\alpha_1), \ldots, k_m f(\alpha_m)\rangle \quad (369) \\ &= \omega_q^{za} \left|S_a^k\right\rangle \quad (370) \end{aligned}$$

□

E Notation Tables

We begin with notation used for both the Clifford and polynomial protocols, and then proceed to notation used only in the polynomial protocol (beginning with the sign key k, Definition 2.6).

Notation	Reference	Explanation
γ	Protocol 4.1, 6.1	Error of circuit which is being applied; used in both protocols
ϵ	Protocol 3.1, 5.1	Security parameter
Π_0, Π_1	Definition 2.1	Projections used to define security of a QAS
L	Protocol 6.1	Number of Toffoli gates in the circuit
n	Protocol 4.1,6.1	Number of qubits in the circuit which is being applied
N	Definition 1.2	Number of gates in the circuit which is being applied
E	Theorem 1.4,1.6,1.5	Eve's environment
$\mathcal{E}$	Section 6.4	Prover's environment register in QPIP protocol
k	Definition 2.6	Sign key for signed polynomial code; used as superscript for encoded states
$\tilde{U}$	Section 2.5.1	Logical version of a gate U for the signed polynomial code
m, d	Definition 2.6	Length and degree of polynomial code
E_k, D_k	Definition 2.7, 2.8	Encoding circuit for signed polynomial code ($E_k = D_k(F^{\otimes d} \otimes \mathcal{I})$)
$\mathbb{P}_m$	Definition 2.3	Group of generalized Pauli operators
g	Equation 149	g takes as input strings in F_q^m and returns the first coordinate of each string
ρ^k	Definition 2.6	Initial state in polynomial QPIPcontaining n authenticated 0 states and L authenticated magic states
$\mathcal{P}_i$	Section 6.4	Register containing prover's $3m(L-i+1)+mn$ qubits at the end of round $i-1$
$\mathcal{V}_i$	Section 6.4	Register containing the $3m(i-1)$ qubits which have been sent to the verifier in rounds $1, \ldots, i-1$
$\mathcal{F}$	After equation 155	Register of m qudits containing the final authenticated qudit given to the verifier in the final round
$\mathcal{P}_{final}$	After equation 155	Register of authenticated qudits remaining with the prover at the end of the protocol ($\mathcal{P}_{L+1} = \mathcal{P}_{final} \cup \mathcal{F}$)
$\mathcal{V}_{final}$	After equation 155	Register containing all $m(3L+1)$ qudits sent to the verifier during the protocol ($\mathcal{V}_{final} = \mathcal{F} \cup \mathcal{V}_{L+1}$)
$\tau_i(z, x, k)$	Section 6.4	Keys held by verifier at the start of round i ($z, x \in F_q^{\lvert\mathcal{P}_i\rvert}$)
δ_i	Section 6.4	Measurement result in F_q^{3m} of prover in round i
Δ_i	Claim 6.1	Measurement results (composing a string in F_q^{3mi}) from rounds 1 to i
$\tilde{Q}_i$	Section 6.1.4	Logical Clifford operators applied in round i
$\tilde{C}_{\beta_i}$	Section 6.1.4	Clifford correction operators for Toffoli gate i if measurement result is $\beta_i \in F_q^3$
$\rho^k_{g(\Delta_{i-1})}$	Claim 6.1	The state on $mn + 3mL$ qudits resulting from applying operations requested in rounds $1, \ldots, i-1$
$l \in F_q^{3L}$	Fact 6.1	Used in the sum over all possible teleportation measurement results
$\beta \in F_q^{3L}$	Fact 6.1	Used to denote one fixed measurement result
$\hat{\Pi}_0$	Equation 158	Used to denote the accepting subspace on $3mL+m$ qudits in the polynomial QPIP
Π_{G_a}	Equation 187	Projection (on $3mL$ qudits) onto a valid, decoded measurement result $a \in F_q^{3L}$