Innovations in Computer Science 2010

Interactive Proofs For Quantum Computations

Dorit Aharonov Michael Ben-Or Elad Eban
School of Computer Science, The Hebrew University of Jerusalem, Israel
doria@cs.huji.ac.il benor@cs.huji.ac.il elade@cs.huji.ac.il

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. Is it possible to experimentally test that these systems perform as they should, if we cannot efficiently compute predictions for their behavior? Vazirani has asked [21]: 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 [13] 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 few qubits. Our main theorem can be roughly stated as: "Any language in BQP has a QPIP, and moreover, a fault tolerant one" (providing a partial answer to a challenge posted in [1]). 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, is not fault tolerant. Our second protocol uses polynomial codes QAS due to Ben-Or, Crépeau, Gottesman, Hassidim, and Smith [8], combined with quantum fault tolerance and secure multiparty quantum computation techniques. A slight modification of our constructions makes the protocol "blind": the quantum computation and input remain unknown to the prover.

Keywords: Interactive Proofs; Authentication

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 Vazirani to ask [21]: 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.

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. Consider for example, a company called *Q-Wave* which is trying to convince a certain potential customer that the system it had managed to build is in fact a quantum computer of 100 qubits. How can the customer, who cannot make predictions of the outcomes of the computations made by the machine, test that the machine is indeed a quantum computer which does what it is claimed to do? 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 [19, 24].

Also, 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 costumer to trust the outcome provided by an untrusted company? And even if the company is honest, can the costumer detect innocent errors in such a computation?

As Vazirani points out [21], a partial answer to these questions is already given in the form of Shor's algorithm. 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 multiplying the factors and comparing to the input).

This, however, does not fully address the issues raised above. Let us take for example the case of the company trying to convince a costumer that the system it is trying to sell is indeed 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 100 qubits can be easily performed by today's classical technology. How about delegated quantum computations? Here too it is unclear how the ability to verify the outcome of Shor's algorithm can help in the context of a costumer who wishes to be convinced of correctness of computation of other problems in BQP and in particular of BQP complete problems (e.g., [4, 16, 25]). As for experimental results, it is difficult to rigorously state what exactly does the ability to apply Shor's algorithm successfully imply. There is a fundamental difference between being convinced of the ability to factor, and testing universal quantum evolution; it may even be the case that factoring is in BPP while BQP is not.

We thus pose the following main question (this question was also asked by Gottesman [1]): Can one be convinced of the correctness of the computation of *any* polynomial quantum circuit? Does a similar statement to the one above regarding Shor's algorithm, apply for universal quantum computation? A different way to pose this question is: can one be convinced of the "correctness" of the quantum mechanical description of any quantum experiment that can be conducted in a reasonable amount of time in the laboratory, even though one cannot compute any predictions for the outcomes of this experiment?

In this paper we address the above fundamental question in a rigorous way. 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 it indeed computes what it should.

1.2 Quantum Prover Interactive Proofs (QPIP)

Interactive proof systems, defined by Goldwasser, Micali and Rackoff [13], 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 $\mathcal{P}$) and a BPP verifier ($\mathcal{V}$) such that for any $x \in \mathcal{L}$, $\mathcal{P}$ convinces $\mathcal{V}$ of the fact that $x \in \mathcal{L}$ with probability $\geq \frac{2}{3}$ (completeness). Otherwise, when $x \notin \mathcal{L}$ any prover fails to convince $\mathcal{V}$ with probability higher than $\frac{1}{3}$ (soundness).

Shor's factoring algorithm [20] 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). Recall that quantum interactive proofs which were studied previously in the literature (e.g., [23]) had an unbounded quantum prover and a BQP *verifier*.

Clearly, such an interactive proof between a BQP prover and a BPP verifier exists for any problem inside NP whose witness can be found in BQP. However, it is widely believed that BQP is not contained in NP (and in fact not even in the polynomial hierarchy). The main idea of the paper is to generalize the above interactive proof point of view of Shor's algorithm, and show that with this generalization, a verifier can be convinced of the result of *any* polynomial quantum circuit.

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. Indeed Gottesman [1] posed his question in this model. Unfortunately we do not know how to prove the results in this model. We therefore strengthen it slightly and allow the verifier additional access to a constant number of qubits and to a quantum channel. This verifier can be viewed as modeling our current computational abilities.

Definition 1.1 *Quantum Prover Interactive Proof (QPIP) is an interactive proof system with the following properties:*

- *The prover is computationally restricted to* BQP.
- *The verifier is a hybrid quantum-classical machine. Its classical part is a* BPP *machine. The quantum part is a register of c qubits (for some constant c), on which the prover can perform arbitrary quantum operations. At any given time, the verifier is not allowed to possess more than c qubits. The interaction between the quantum and classical parts 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 are two communication channels: one quantum and one classical.*

The completeness and soundness conditions are identical to the IP *conditions.*

Abusing notation, we denote the class of languages for which such a proof exists also by QPIP.

1.3 Main Results

Definition 1.2 *The promise problem Q-CIRCUIT consists of a quantum circuit made of a sequence of gates, $U = U_T \ldots U_1$, acting on n input bits. The task is to distinguish between two cases:*

$$\begin{aligned} \textit{YES} &: \|((|0\rangle\langle 0| \otimes \mathcal{I}_{n-1})\, U\, |\bar{0}\rangle\|^2 \geq \tfrac{2}{3} \\ \textit{NO} &: \|((|0\rangle\langle 0| \otimes \mathcal{I}_{n-1})\, U\, |\bar{0}\rangle\|^2 \leq \tfrac{1}{3} \end{aligned}$$

Q-CIRCUIT is a BQP complete problem. This remains true for any soundness and completeness parameters $0 < s, c < 1$, if $c - s > \frac{1}{Poly(n)}$. Our first result is:

Theorem 1.1 *The language Q-CIRCUIT has a QPIP.*

The proof is quite simple once the right definitions are set, given the notion of quantum authentication schemes (QAS). We explain the idea later in the introduction.

Since Q-CIRCUIT is BQP complete, and QPIP is trivially inside BQP, we have as an immediate corollary:

Corollary 1.2 BQP $=$ *QPIP.*

Thus, a BQP the prover can convince the verifier of any language he can compute. We remark that our definition of QPIP 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 extend the result to enable the verifier to be convinced of *correctness* of the prover's outcome (in both 0 and 1 cases) using simple arguments based on the fact that BQP is closed under complement (see Section 7).

Importantly, the above results apply in a setting in which the physical systems are subjected to noise:

Theorem 1.3 *Theorem 1.1 holds also when the quantum communication and computation devices are subjected to the usual local noise model assumed in quantum fault tolerance settings.*

Thus, our results apply also in physically realistic setting; the prover, the verifier and the communication channel can all be noisy, as long as the noise satisfies the standard restrictions of quantum fault tolerance. The proof of 1.3, unlike that of 1.1, requires some technical effort.

In the works [6, 11] 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 gives

Theorem 1.4 *Theorem 1.3 holds also in a blind setting, namely, the prover does not get any information regarding the function being computed, and its input, beyond an upper bound on the size of the circuit.*

An analogous result for NP-hard problems was shown already in the late 80's to be impossible unless the polynomial hierarchy collapses [2].

1.4 Proofs Overview (and a new Quantum Authentication Scheme)

Our main tool is quantum authentication schemes (QAS) [7]. Roughly, a QAS allows two parties to communicate in the following way: Alice sends an encoded quantum state to Bob. The scheme is secure if in case the state had been altered, Bob's chances of declaring valid a wrong state are small. The basic idea of our QPIP protocols is to have the prover hold, at time step t, the authenticated version of the state $U_t \cdots U_1 |0\rangle$. We need to describe how the prover and the verifier can together update this authenticated state, without the prover knowing the authentication key, and using only a constant number of qubits at the verifier's end.

The basic QPIP. Our starting point is a very simple QPIP protocol which works with any QAS (under the mild condition that it maintains its security when it is applied in parallel on several registers). Unfortunately this scheme is not fault tolerant, but it serves well to explain how QAS and QPIP are related. The idea is simple: use the prover as an untrusted storage device. To do this, the verifier asks the prover for the authenticated qubits on which he would like to apply the next gate. He then decodes those qubits, applies the gate, encodes them back and sends them to the prover. The proof of security is quite straight forward given the security of the QAS.

For concreteness, we apply this scheme using a new simple and efficient QAS, based on random Clifford group operations (it is reminiscent of Clifford based quantum 2-designs [12]). To encode a state of m qubits, tensor the state with d qubits in the state $|0\rangle$, and apply a random Clifford operator on the $m+d$ qubits. To prove the security proof of this QAS we use similar tools to that of the proof of [12] of quantum unitary $2-$designs, to show that any attack of the prover is mapped by the random Clifford operator to random Pauli operators. We then show that those are detected with high probability. This QAS might be interesting in its own right due to its simplicity, and since it is extremely efficient: using d extra qubits we get security 2^{-d} and moreover this security is independent of the dimension of the Hilbert space being authenticated (unlike in other QASs, e.g., the second QAS we use in this paper).

For fault tolerance, it seems necessary that the prover

will be able to apply quantum gates on the state by himself. Due to the lack of structure of the authenticated states in the general QAS we do not know how to do this without revealing the authentication key. We thus focus on one specific QAS which does exhibit enough structure to allow for fault tolerance. As a side benefit, this QPIP also involves just one round of quantum communication, compared to the basic QPIP in which all rounds are quantum.

Polynomial codes based QAS and its QPIP We recall the QAS due to Ben-Or, Crépeau, Gottesman, Hassidim and Smith [8]. This QAS is based on signed quantum polynomial codes, which are quantum polynomial codes [3] of degree at most d multiplied by some random sign (1 or -1) at every coordinate (this is called the sign key) and a random Pauli at every coordinate (the Pauli key).

First, we present a security proof for this QAS; this corrects an error in the security proof of the original paper [8]. To do this we first prove that no Pauli attack can fool more than a small fraction of the sign keys, and thus the sign key suffices in order to protect the code from any Pauli attack. 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 (this is the missing link in the proof of [8]). Indeed, if one omits the Pauli key from this QAS one gets an authentication scheme which is secure against Pauli attacks but *not* against general attacks. We proceed by showing, (with some similarity to the Clifford based QAS proof), that the random Pauli key effectively translates the prover'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 note that the security parameter d is different and is in fact slightly worse (though in an unimportant way) than the d parameter in the Clifford based QAS (see Sec. 2.4 for more precise statements).

Due to its algebraic structure, the signed polynomial code QAS allows applying gates without knowing the key. This was used in [8] for secure multiparty quantum computation; here we use it to allow the prover to perform gates without knowing the authentication key.

The QPIP protocol goes as follows. The prover selects an authenticated code, which encodes one qudit into m qudits; m is a small constant. The verifier authenticated the inputs to the circuit, as well as the magic states required to perform Toffoli gates (called Toffoli states), as described in [8, 18]. With those authenticated states at hand, the prover can perform universal computation using only Clifford group operations and measurements (universality was proved for qubits in [9], and the extension to higher dimensions was used in [8]). The prover sends the verifier results of measurements and the verifier sends information given those results, which enables the prover to continue the computation. The communication is thus classical except for the first round.

Fault Tolerance One is tempted to try and apply known fault tolerance techniques (e.g., [3]) to achieve robustness to local noise. However, a problem arises when attempting to apply those techniques directly: since all states sent to the prover must be authenticated, the verifier needs to send the prover polynomially many authenticated qubits every time step, so that the prover can perform error corrections on all qubits simultaneously, as is required for fault tolerance. However, the verifier's quantum register contains only a constant number of qubits.

We bypass this problem as follows. At the first stage of the protocol, the verifier authenticates as many zero states and Toffoli states as would be required during the entire protocol, and sends those to the prover. This is done sequentially, state by state, so that the verifier only uses a constant quantum register. As soon as the prover receives a qudit, he protects it using his own concatenated error correcting code; the result is that the effective error in any of the authenticated states is also a 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. Thus, polynomially many such authenticated states can be passed to the prover in sequence.

To proceed, we notice that the purity of the states can now be amplified using *purification* [9]. Indeed, the prover cannot perform purification on his own since the purification compares authenticated qubits and the prover does not know the authentication code. However, the verifier can help the prover using classical communication (For the purification of the Toffoli states, we use the methods of [8]). This way the prover can reduce the effective error on his encoded authenticated states; in fact, with polylogarithmic work the error can be reduced all the way to inverse polynomial. Effectively, the input at the prover's hands is now error free with very high probability; The prover and verifier can now perform the polynomial-code based QPIP on this input, to apply the desired circuit. During the application of the protocol the prover performs his gates fault tolerantly with respect to his error correcting codes, and constantly correct errors; in other words, he follows the standard quantum fault tolerance scheme on his side, with respect to his codes.

Blind Quantum Computation To achieve Theorem 1.4, 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, 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.

1.5 Interpretations of the Results

The corollaries below clarify how the QPIP protocols designed here are used to address the motivating questions from Sec. 1.1.

A natural question is how close the quantum state is to the correct state, in case the verifier accepted. As usual, to guarantee closeness to the correct state at the end, we need to require that the probability to abort is small:

Corollary 1.5 *For a QPIP protocol with security δ, if the verifier's probability to abort is $\leq 1-\gamma$ then the trace distance between the final density matrix and that of the correct state is at most $\frac{2\delta}{\gamma}$.*

We would also like to interpret the results as a guarantee that a prover that passed the test (for "hard" quantum circuits) cannot be in BPP, assuming BQP $\neq$ BPP. To make this formal, we need a somewhat stronger assumption which is still widely believed: we assume that there is a language $L \in$ BQP and a polynomial time samplable distribution D on which any BPP machine errs with non negligible probability regarding membership in L (e.g. the standard cryptographic assumptions about the hardness of Factoring or Discrete Log). Given this, the following corollary can be proven using Corollary 1.5.

Corollary 1.6 *Fix such a language L. If the verifier interacts with a given prover using the QPIP for L, and the probability for abort is $1-\gamma$ for some positive γ, (where probability is taken also over an input sampled according to $\mathcal{D}$), then the QPIP cannot be simulated in* BPP.

One might wonder whether it is possible to somehow get convinced not only of the fact that the computation that was performed by the prover is indeed the desired one, but also that the prover must have had access to some quantum computer, or at least to some quantum memory. We prove:

Claim 1.7 *There exists a language $\mathcal{L} \in$ BQP such that even if the prover in our QPIP is replaced by one with unbounded classical computational power, but only a constant number of qubits, the prover will not be able to convince the verifier to accept: $\mathcal{V}$ in this case aborts the computation with high probability.*

In fact, the proof of this claim does not require the machinery of QPIP, it is based on ideas that appeared already in the study bounded storage quantum models [22]. We thus see yet another setting in which quantum mechanics cannot be simulated by classical systems, regardless of how computationally powerful they are; In addition to bounded storage models, this property emerges also in other contexts, e.g., the *EPR* experiment.

We note that all our proofs are based on the assumption that the system obeys the mathematical model of quantum mechanics. It is thus assumed in all proofs that Nature behaves in a way which *can* be described in this framework. Note that this does not implicitly assume that the system is fully quantum mechanical (something which we might want to check using these protocols): Classical systems or decohered quantum systems can also be described in this mathematical model. However, this assumption does put restrictions on the possible models for which our security proofs would hold, and in particular, it requires the system to be linear and have some tensor product structure. Of course, one cannot hope to prove security results without *any* assumption on the mathematical model that describes the physical system.

Finally, we remark that in the study of IP, a natural question is whether a prover can prove any language it can compute; The answer is known to be positive for PSPACE, NP and #P provers, but is still open for coNP, SZK and PH [5]. Our results imply that a BQP prover can prove the entire class of BQP (albeit to a verifier who is not entirely BPP).

1.6 Related Work and Open Questions

The two questions regarding the cryptographic angle were asked by Childs in [11], and by Arrighi and Salvail in [6], who proposed schemes to deal with such scenarios. However [11] do not deal with a cheating prover, and [6] deals only with a restricted set of functions which they refer to as random verifiable.

Broadbent, Fitzsimons, and Kashefi [10] have proven related results independently. Using measurement based quantum computation, they construct a protocol for universal blind quantum computation, and moreover, the verifier's register consists of a single qubit. These results imply similar implications to ours, though they are not presented using the language of QPIP and do not discuss the more foundational motivations and implications.

Yi-Kai Liu suggested an alternative way to implement the basic QPIP protocol, based on Kitaev's circuit to Hamiltonian construction [15], where the prover prepares the history state of the circuit and sends the qubits of this state to the verifier one by one. The verifier decides randomly on one local term in the Hamiltonian, keeps only the qubits in this term (Two qubits suffice using [14]) and measures the energy of that term on those qubits. This is repeated $poly(n)$ times to amplify security. Unfortunately, it is not clear how to make this scheme fault tolerant.

An important and intriguing open question is whether it is possible to make the verifier completely classical. This

would have interesting fundamental implications regarding the ability of a completely classical system to learn and test a quantum system.

Another interesting (perhaps related?) open question is to study the model we have presented here of QPIP, with more than one prover. Possibly, multiprover QPIP might be strong enough even when restricted to classical communication. In [10] this was shown assuming that the two provers are entangled (but other than that cannot communicate).

This work also raises some questions in the philosophy of science. In particular, it suggests the possibility of formalizing and studying the interaction between physicists and Nature, using computational complexity notions. Following discussions with us at preliminary stages of this work, Jonathan Yaari is currently studying "interactive proofs with Nature" from the philosophy of science aspect [26].

Paper Organization We start with notations and background regarding the notions that appear in this paper, in Sec. 2. Here we also present the polynomial codes based QAS . Sec. 3 provides the basic protocol together with the Clifford QAS. Sec. 4 we describe the QPIP based on signed polynomial codes, and prove its fault tolerance; Sec. 5 generalizes the result to the blind setting. Sec. 6 proves the corollaries regarding how the results can be interpreted; Sec. 7 defines the symmetric version of QPIP and proves its equivalence to Definition 1.1. We provide a complete proof of security of the polynomial-codes-based QAS in the appendix.

2 Background

2.1 Pauli and Clifford Group

Let $\mathbb{P}_n$ denote the n-qubits Pauli group. $P = P_1 \otimes P_2 \otimes \ldots \otimes P_n$ were $P_i \in \{\mathcal{I}, X, Y, Z\}$.

Definition 2.1 *Generalized Pauli operator over F_q:* $X|a\rangle = |(a+1) \mod q\rangle$, $Z|a\rangle = \omega_q^a |a\rangle$, $Y = XZ$, *where $\omega_q = e^{2\pi i/q}$ is the primitive q-root of the unity.*

We note that $ZX = \omega_q XZ$. 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 2.2 *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 denote the set of all unitary matrices over a vector space A as $\mathbb{U}(A)$. The Pauli group $\mathcal{P}_n$ is a basis to the matrices acting on n-qubits. In particular, we can write any matrix $U \in \mathbb{U}(A \otimes B)$ for A the space of n qubits, as $\sum_{P \in \mathcal{P}_n} P \otimes U_P$ with U_P some matrix on B.

Let $\mathfrak{C}_n$ denote the n-qubit Clifford group. Recall that it is a finite subgroup of $\mathbb{U}(2^n)$ 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 *A random element from the Clifford group on n qubits can be sampled efficiently by choosing a string k of $poly(n)$ length 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.*

2.2 Polynomial Quantum Error Correction Codes

Definition 2.3 *Polynomial error correction code [3]. Given m, d, q and $\{\alpha_i\}_1^m$ where α_i are distinct non zero values from F_q, the encoding of $a \in F_q$ is $|S_a\rangle$*

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

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 [3].

We recall from [3] how to apply several useful gates on states encoded by those codes, in a *transversal* manner. The simplest gate is the generalized X:

$$\widetilde{X}|S_a\rangle = X^{\otimes m}|S_a\rangle = \left|S_{(a+1)}\right\rangle \quad (2)$$

Similarly for logical controlled-SUM (denoted *SUM*) which is the generalization of CNOT, it is easy to check that:

$$\widetilde{SUM}|S_a\rangle|S_b\rangle = SUM^{\otimes m}|S_a\rangle|S_b\rangle = |S_a\rangle|S_{a+b}\rangle \quad (3)$$

The logical Fourier transform requires a little more work. Recall the definition of the generalized Fourier transform in F_q:

$$W_r|a\rangle \stackrel{\text{def}}{=} \frac{1}{\sqrt{q}} \sum_b \omega_q^{rab}|b\rangle \quad (4)$$

The logical Fourier operator $\widetilde{F}$ can be applied using the interpolation coefficients c_i for degree $m-1$ polynomials:

$$\widetilde{W}|S_a\rangle \stackrel{\text{def}}{=} q^{-m/2} \sum_b \omega_q^{ab}|S_b\rangle =$$

$$W_{c_1} \otimes W_{c_2} \otimes \ldots \otimes W_{c_m}|S_a\rangle .$$

The Z operator can be seen as a product of Fourier and X operators, and thus can also be applied. More directly, it can be seen that

$$\widetilde{Z}\left|S_a\right\rangle = \left|(-1)^a S_a\right\rangle = Z^{c_1 k_1} \otimes \ldots \otimes Z^{c_m k_m} \left|S_a\right\rangle \quad (5)$$

To complete this to a universal set of gates, we need to add measurements, and Toffoli states. This is explained further in Section 4.1, in the relevant context.

2.3 Signed Polynomial Codes

Definition 2.4 ([8]) *The signed polynomial code with respect to a string $k \in \{\pm 1\}^m$ (denoted $\mathcal{C}_k$) is defined by:*

$$\left|S_a^k\right\rangle \stackrel{\text{def}}{=} \frac{1}{q^{d/2}} \sum_{\substack{f:deg(f)\leq d \\ f(0)=a}} \left|k_1 f(\alpha_1), \ldots, k_m f(\alpha_m)\right\rangle \quad (6)$$

Once again, we use $m = 2d+1$. The code can detect d errors. Also, $\mathcal{C}_k$ is self dual [8], namely, the code subspace is equal to the dual code subspace.

2.4 Quantum Authentication

Definition 2.5 *(adapted from Barnum et. al. [7]). A quantum authentication scheme (QAS) 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 an m-qubit message system M and a key $k \in \mathcal{K}$ and outputs a transmitted system T of $m+d$ qubits.*
- *$\mathcal{B}$ takes as input the (possibly altered) transmitted system T' and a classical key $k \in \mathcal{K}$ and outputs two systems: a m-qubit message state M, and a single qubit V which indicate whether the state is considered valid or erroneous. The basis states of V are called $|VAL\rangle, |ABR\rangle$. For a fixed k we denote the corresponding super-operators by A_k and B_k.*

Given a pure state $|\psi\rangle$, consider the following test on the joint system M, V : output a 1 if the first m qubits are in state $|\psi\rangle$ or if the last qubit is in state $|ABR\rangle$, otherwise, output 0. The corresponding projections are:

$$\begin{aligned} P_1^{|\psi\rangle} &= |\psi\rangle\langle\psi| \otimes I_V + \\ &\quad (I_M - |\psi\rangle\langle\psi|) \otimes |ABR\rangle\langle ABR| \\ P_0^{|\psi\rangle} &= (I_M - |\psi\rangle\langle\psi|) \otimes |VAL\rangle\langle VAL| \end{aligned}$$

The scheme is secure if for all possible input states $|\psi\rangle$ and for all possible interventions by the adversary, the expected fidelity of $\mathcal{B}$'s output to the space defined by $P_1^{|\psi\rangle}$ is high:

Definition 2.6 *A QAS is secure with error ϵ if for every state $|\psi\rangle$ it holds:*

- *Completeness: For all keys $k \in \mathcal{K}$: $B_k(A_k(|\psi\rangle\langle\psi|)) = |\psi\rangle\langle\psi| \otimes |VAL\rangle\langle VAL|$*
- *Soundness: For any super-operator Λ (representing a possible intervention by the adversary), if ρ_B is defined by defined by $\rho_B = \frac{1}{|\mathcal{K}|}\sum_k B_k\big(\Lambda(A_k(|\psi\rangle\langle\psi|))\big)$, then: $Tr(P_1^{|\psi\rangle}\rho_B) \geq 1-\epsilon$.*

2.5 QAS based on Signed Quantum Polynomial Codes

Protocol 2.1 ***Signed Polynomial Codes Authentication protocol*** *(due to [8]): Alice wishes to send the state $|\psi\rangle$ of dimension q. She chooses a security parameter d, and a code length $m = 2d+1$.* ***Encoding:*** *Alice randomly selects a pair of keys: a sign key $k \in \{\pm 1\}^m$ and a Pauli key (x,z) with $x, z \in {F_q}^m$. She encodes $|\psi\rangle$ using the signed quantum polynomial code $\mathcal{C}_k$ of polynomial degree d (see Definition 2.4). She then applies the Pauli $P_{(x,z)}$ (i.e., for $j \in \{1,..,m\}$ she applies $Z^{z_j}X^{x_j}$ on the j'th qubit).* ***Decoding*** *Bob applies the inverse of $P_{(x,z)}$, and performs the error detection procedure of the code $\mathcal{C}_k$. He aborts if any error is found and declares the message valid otherwise.*

The completeness of this protocol is trivial, while soundness requires work:

Theorem 2.2 *The polynomial authentication scheme is secure against general attacks with security 2^{-d}*

We provide a complete proof in the appendix. We notice that in this scheme a q-dimensional system is encoded into a system of dimension $q^m = q^{2d+1}$. It is easy to see that the scheme is secure when applied in parallel:

Theorem 2.3 *The polynomial based QAS applied in parallel (with the same sign key for all registers), and with degree d polynomial, has the same security as the individual QAS, that is: 2^{-d}.*

3 Basic QPIP

3.1 The QPIP Protocol

Protocol 3.1 ***Basic QPIP for Q-CIRCUIT****: Fix a security parameter ϵ. Given is a quantum circuit consisting of two-qubit gates, $U = U_T \ldots U_1$, with error probability reduced to $\leq \delta$. The verifier authenticates the input qubits of the circuit one by one using* ***any*** *QAS with security parameter d set such that the QAS is ϵ secure. That is, every qubit is authenticated using a constant number of auxiliary qubits, which are all sent to $\mathcal{P}$. For each $i = 1$ to m, the verifier asks the prover for the authenticated qubits on which he would like to apply the gate U_i, decodes them, aborts if any error is found, applies the gate, authenticates the resulting qubits using a new pair of authentication keys, and sends*

the encoded qubits back to $\mathcal{P}$. Finally, the verifier asks $\mathcal{P}$ to send the output authenticated qubit, decodes and aborts if any error is found; otherwise, measures the decoded qubit and accepts or rejects accordingly. In any case that $\mathcal{V}$ does not get the correct number of qubits he aborts.

We now prove Theorem 1.1 by proving:

Theorem 3.1 *For any $\epsilon, \delta > 0$ Protocol 3.1 is a QPIP protocol with completeness $1-\delta$ and soundness $\delta+\epsilon$ for Q-CIRCUIT.*

Proof: If the prover is honest, the verifier will declare valid with certainty. Since the error in the circuit is $\leq \delta$, $(1-\delta)$ completeness follows. For soundness, we observe that for the verifier to accept if x is not in the language, means that he has not aborted, and also, answers YES. Let us denote by P_{bad} the projection on this subspace (*Valid* on the first qubit, *Accept* on the second). To bound the probability of this event, we observe that the correct state at any given step is a state which is authenticated by the QAS applied in parallel.

3.2 Clifford based QAS

We now present the new Clifford based QAS, which is essentially adding zero states and applying a Clifford-twirl, namely, conjugation by a random Clifford operator.

3.2.1 The Clifford Based QAS: the Protocol

Protocol 3.2 ***Clifford based QAS**: Given is a state $|\psi\rangle$ on m qubits and $d \in \mathbb{N}$ a security parameter. We denote $n = m+d$. The set of keys $\mathcal{K}$ consists of succinct descriptions of Clifford operations on n qubits (following Fact 2.1). We denote by $C = C_k$ the operator specified by a key $k \in \mathcal{K}$. **Encoding -** A_k: Alice applies C_k on the state $|\psi\rangle \otimes |0\rangle^{\otimes d}$. **Decoding -** B_k: Bob applies $C_k^\dagger$ to the received state. Bob measures the auxiliary registers and declares the state valid if they are all 0, otherwise Bob aborts.*

3.2.2 Security

Theorem 3.2 *The Clifford scheme applied to $n = m+d$ qubits is a QAS with security 2^{-d}, where d is the number of qubits added to a message on m qubits.*

The proof relies on the fact that a Clifford twirl takes any operator to a mixture of Paulies (which is uniform on the non-identity ones), as stated in the following lemma.

Lemma 3.3 *(Adapted from [12] Corollary 4.) For an arbitrary trace-preserving completely positive map $\Lambda(\rho) = \sum_k \Lambda_k \rho \Lambda_k^\dagger$, applying a Clifford-twirl to Λ is equivalent to applying a Haar-twirl to Λ, that is for any density matrix ρ on n qubits:*

$$\begin{aligned} &\int dU \sum_k U^\dagger \Lambda_k U \rho U^\dagger \Lambda_k^\dagger U \\ &= \sum_{C \in \mathfrak{C}_n} \sum_k C^\dagger \Lambda_k C \rho C^\dagger \Lambda_k^\dagger C \\ &= s\rho + \frac{(1-s)}{4^n - 1} \sum_{Q \in \mathbb{P}_n, Q \neq \mathcal{I}} Q \rho Q^\dagger \end{aligned} \tag{7}$$

where $s = \sum_k |Tr(\Lambda_k)|^2$.

We now prove the security of the Clifford QAS given Lemma 3.3

*P*roof of **Theorem 3.2**: From Lemma 3.3 we know what Bob's state after Eve's intervention is and we would like to bound its projection on $P_1^{|\psi\rangle}$:

$$\begin{aligned} &\mathrm{Tr}\Big(P_1^{|\psi\rangle}\big(s\rho + \frac{1-s}{4^n-1} \sum_{Q \in \mathbb{P}_n \setminus \{\mathcal{I}\}} Q\rho Q^\dagger\big)\Big) \\ &= s\mathrm{Tr}(P_1^{|\psi\rangle}\rho) + \frac{1-s}{4^n-1} \sum_{\substack{Q \in \mathbb{P}_n \\ Q \neq \mathcal{I}}} \mathrm{Tr}(P_1^{|\psi\rangle} Q\rho Q^\dagger) \end{aligned} \tag{8}$$

By definition of $P_1^{|\psi\rangle}$ we see that $\mathrm{Tr}(P_1^{|\psi\rangle}\rho) = 1$. On the other hand: $\mathrm{Tr}(P_1^{|\psi\rangle} Q\rho Q^\dagger) = 1$ when Q does not flip any of the auxiliary qubit, and vanishes otherwise. The Pauli operators that do not flip auxiliary qubits can be written as $Q' \otimes Q''$ where $Q' \in \mathbb{P}_m$ and $Q'' \in \{\mathcal{I}, Z\}^{\otimes d}$. It follows that the total number of such operators is exactly $4^m 2^d$. Omitting the identity $\mathcal{I}_n$ we are left with $4^m 2^d - 1$ operators which are undetected by our scheme. We return to Eq. 8:

$$\begin{aligned} \ldots &\geq s + (1-s)\,(1 - \frac{4^m 2^d - 1}{4^n - 1}) \\ &\geq s + (1-s)\,(1 - \frac{4^m 2^d}{4^{m+d}}) \\ &= 1 - \frac{1-s}{2^d} \end{aligned} \tag{9}$$

The security follows from the fact that $s \geq 0$, and hence the projection is bounded by $1 - \frac{1}{2^d}$.

3.2.3 Applying the Clifford based QAS in Parallel

Here we make rigorous the definition and correctness of applying the a QAS on entangled registers in parallel. We consider a QAS (for instance the Clifford QAS) which authenticates m qubits. Given r blocks of m qubits each, we can apply the QAS separately on each one of the r blocks. $\mathcal{B}$ declares the state valid if all of the r registers are valid according to the original QAS. We call this applying the *QAS in parallel*. The completeness of the concatenated protocol is trivial, for any QAS. For soundness we prove the following theorem for the Clifford QAS.

Theorem 3.4 *Applying the Clifford QAS in parallel has the security of the individual Clifford with security parameter d, QAS, that is 2^{-d}. This holds regardless of the number of blocks (r) that are authenticated.*

Proof: From Lemma 3.3, we know that any attack by Eve on an authenticated register is equivalent to an effect of mixing operator $\mathcal{M}_s$:

$$\mathcal{M}_s(\rho) = s\rho + \frac{(1-s)}{4^n-1} \sum_{\substack{Q\in\mathbb{P}_n \\ Q\neq\mathcal{I}}} Q\rho Q^\dagger \tag{10}$$

on the unencoded message space. We denote $\mathcal{M}_s(\rho) = \widetilde{\rho}$, we first deal with the case were $r = 2$

We are interested of the effect of a twirl from $\mathcal{C}_n \otimes \mathcal{C}_n$ on a state $\rho_1 \otimes \rho_2$. It is fairly straight forward to notice (using the same ideas as [12]) that the twirl on any Λ results in the transformation:

$$\begin{aligned} \rho_1 \otimes \rho_2 \Rightarrow & s(\rho_1 \otimes \rho_2) + h(\widetilde{\rho}_1 \otimes \rho_2) \\ & + r(\rho_1 \otimes \widetilde{\rho}_2) + t(\widetilde{\rho}_1 \otimes \widetilde{\rho}_2) \end{aligned} \tag{11}$$

for some scalars s, q, r, t which depend on Λ.

Bob does not abort, if both individual Clifford QAS are valid that is: $P_0^{\rho_1\otimes\rho_2} = P_0^{\rho_1} \otimes P_0^{\rho_2}$. From the security of the individual QAS we know that $\mathrm{Tr}((P_0^{\rho_i})B(\widetilde{\rho}_i)) < 2^{-d}$ where B is Bob's cheat detecting procedure. Using this observations on Eq. 11:

$$\begin{aligned} &\mathrm{Tr}\Big(P_0^{\rho_1\otimes\rho_2} B \left(s\left(\rho_1 \otimes \rho_2\right) + h\left(\widetilde{\rho}_1 \otimes \rho_2\right) \right. \\ &\quad \left. + r\left(\rho_1 \otimes \widetilde{\rho}_2\right) + t\left(\widetilde{\rho}_1 \otimes \widetilde{\rho}_2\right)\right) \Big) \\ &= s \cdot 0 + q2^{-d} + r2^{-d} + t(2^{-d})^2 \\ &\leq (1-s)2^{-d} \end{aligned} \tag{12}$$

Where the inequality holds since $s + q + r + t = 1$.
The claim for $r > 2$ follows the exact same lines and therefore is omitted.

3.2.4 Analysis and Parameters

Using the Clifford QAS, the classical communication is linear in the number of gates. For $\epsilon = \frac{1}{2}$, we get $d = 1$, and so the verifier uses a register of 4 qubits. 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 Polynomial Codes Authentication Based QPIP

The above described QPIP is not fault tolerant. We therefore defer now to a different type of QPIP in which the prover can perform the quantum gates (with the help of classical communication with the verifier) despite the fact that he does not know the authentication key. To this end we use the QAS based on quantum signed polynomial codes, described in Section 2.5, and show how the prover can apply Clifford operations and measurements on the states. Those operations, augmented with the prover having access to authenticated Toffoli states, form a universal set of gates [8].

4.1 Secure Application of Quantum Gates

We have seen in Sec. 2.3 how to perform operations on states encoded by a polynomial code. In this section we present a way (described in [8]) for the prover to apply these operations on a signed shifted Polynomial error correcting code, without knowing the sign nor the Pauli keys. Hence, this can be done without compromising the security of the authentication scheme.

The main idea is that the prover applies the transitive operations assuming no keys, since he doesn't know the correct authentication key. This will have the wrong effect on the state, but the point is that the verifier can correct for this by updating his key. Once this update is performed, the prover's action has the desired effect on the state.

We will first show the simple and elegant fact that if the verifier wants a (generalized) Pauli applied to the state, he does not need to ask the prover to do anything. The only thing the verifier must do is change his Pauli keys. Then we show how to perform other operations such as *SUM*, Fourier and Measurement.

- **Pauli** X: The logical $\widetilde{X}$ operator consists of an application of $X^{k_1}\otimes, \ldots \otimes X^{k_m}$ where $\mathbf{k}$ is the sign key. We claim that the change $(x, z) \to (x - \mathbf{k}, z)$ will in fact change the interpretation the verifier assigns to the state in the desired way.

$$\begin{aligned} P_{x,z}\left|S_a^k\right\rangle &= P_{x-k,z}P^\dagger_{x-k,z}P_{x,z}\left|S_a^k\right\rangle \\ &= P_{x-k,z}X^{-(x-k)}Z^{-z}Z^zX^x\left|S_a^k\right\rangle \\ &= P_{x-k,z}(X^{k_1}\otimes, \ldots \otimes X^{k_m})\left|S_a^k\right\rangle \\ &= P_{x-k,z}\widetilde{X}\left|S_a^k\right\rangle \end{aligned} \tag{13}$$

- **Pauli** Z: Similarly to the X operator, all that is needed is a change of the Pauli key. We recall that $\widetilde{Z} = Z^{c_1k_1}\otimes\ldots\otimes Z^{c_mk_m}$. 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 $(x, z) \to (x, z - \mathbf{t})$.
- **Controlled-Sum**: In order to remotely apply the *SUM* operation the prover perform transversely Controlled-Sum (*SUM*) from register A to register B on the authenticated states; as if the code was not shifted by the Pauli masking. However, a change in the Pauli keys is needed for the operation to have the

desired effect. It is easy to check that:

$$\begin{aligned} &SUM(Z^{z_A}X^{x_A} \otimes Z^{z_B}X^{x_B}) \\ =&(Z^{z_A-z_B}X^{x_A} \otimes Z^{z_B}X^{x_B+x_A})SUM \end{aligned} \tag{14}$$

Which implies that the same hold for the logical operation $\widetilde{SUM}$ and the Pauli shift $P_{(x,z)}$:

$$\begin{aligned} &\widetilde{SUM}\left(P_{(x_A,z_A)} \otimes P_{(x_B,z_B)}\right) \\ =&\left(P_{(x_A,z_A-z_B)} \otimes P_{(x_B+x_A,z_B)}\right)\widetilde{SUM} \end{aligned} \tag{15}$$

Hence, the verifier must change the pair of keys $(x_A, z_A), (x_B, z_B)$ to $(x_A, z_A - z_B)$ and $(x_B + x_A, z_B)$, for the *SUM* to have the desired affect on the state.

- **Fourier**: The prover performs Fourier transversely on the authenticated state. We recall that the Fourier operation swaps the roles of the X and Z Pauli operator. $WX^xW^\dagger = Z^x$ and $WZ^zW^\dagger = X^{-z}$. This is true for each register separately and hence:

$$\begin{aligned} &\widetilde{W} \cdot Z^{z_1}X^{x_1} \otimes \ldots \otimes Z^{z_m}X^{x_m} \\ =&X^{-z_1}Z^{x_1} \otimes \ldots \otimes X^{-z_m}Z^{x_m} \cdot \widetilde{W} \\ \simeq&Z^{x_1}X^{-z_1} \otimes \ldots \otimes Z^{x_m}X^{-z_m} \cdot \widetilde{W} \end{aligned} \tag{16}$$

  Where the last equality is up to a global phase.
  Therefore the verifier must change the key (x, z) to $(-z, x)$.
- **Measurement in the standard basis**: The prover measures the encoded state in the standard basis, and sends the result to the verifier. Using the x part of Pauli key, and the knowledge of k, the verifier interpolates the polynomial according to values in the received set of points. If the polynomial is indeed a polynomial of low degree (which is always the case if the prover is honest) the verifier sends the encoded value to the prover. Otherwise, the prover is caught cheating and the verifier aborts.
- **Toffoli**: The (generalized) Toffoli gate is applied using Clifford group operations on the Toffoli state $\frac{1}{q}\sum_{a,b}|a, b, ab\rangle$ ([8, 18]). Changes to the keys are made with respect to the actual operations that were performed.

4.2 Polynomial Based QPIP .

Protocol 4.1 ***Polynomial based Interactive Proof for Q-CIRCUIT*** *Fix a security parameter ϵ. Given is a quantum circuit on n qubits consisting gates from the above universal set, $U = U_T \ldots U_1$. We assume the circuit has error probability $\leq \delta$. The verifier sets $d = \lceil \log \frac{1}{\epsilon} \rceil$ and uses 3 registers of $m = 2d + 1$ qudits each, where each qudit is of dimensionality $q > m$. The verifier uses concatenated polynomial QAS with security parameter d to authenticate n input qudits and the necessary number of Toffoli states. This is done sequentially using $3m$ qudits at a time. Then, the prover and verifier perform the gates of the circuit as described above. Finally, if the final measurement does not yield an authenticated answer, the verifier* ***aborts****, otherwise, he accepts or rejects according to the measurement outcome.*

Theorem 4.1 *Protocol 4.1 is a QPIP protocol with completeness $1 - \delta$ and soundness $\delta + \epsilon$ for Q-CIRCUIT.*

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. With $\epsilon = 1/2$, this gives a register of 5 qutrits.

Proof: The completeness is trivial, similarly to the basic QPIP case. To prove the soundness of the protocol we first prove the following lemma, originally stated in [8]:

Lemma 4.2 *At any stage of the protocol the verifier's set of keys, k and $\{(x, z)_i\}_1^n$ are distributed uniformly and independently.*

Proof: Before any gate is applied the claim holds. All that needs to be done it to check that all changes keep this desired property.

The sign key k does not change during the protocol so in this case the claim is trivial. At every step at most two pairs of Pauli keys change. Let us review the possible changes (see Section 4.1) and verify that the claim holds:

- Changes from the Pauli operators and Fourier transform induces shift, swap or negation changes to the keys; all of them preserve the uniform independent distribution trivially.
- The *SUM* operation involves two set of keys $(x_A, z_A), (x_B, z_B)$ which change to $(x_A, z_A - z_B)$ and $(x_B + x_A, z_B)$. The sum $x_B + x_A$, is $\mod q$ hence it is distributed uniformly, in addition it is not hard to see that is independent of x_A. The same holds for $z_A - z_B$ and z_B.
- When the prover measures in the standard basis an authenticated qubit the outcome of the measurement is distributed uniformly at random in F_q^m. Specifically, the outcome does not depend on the sign key or the information that is authenticated. Therefore, even when the prover has the interpretation of his measurement outcome, he does not gain any information about the sign key k or the Pauli keys of other registers.

This implies that at any stage of the computation the set of authentication keys that the verifier holds is uniform and independent of the communication it had so far with the prover. Therefore, if the prover is honest, the state that the prover holds at any given moment should be the correct

state of the quantum circuit, authenticated by the QAS with respect to the current set of keys. The rest of the argument of the proof follows that of the proof of Theorem 1.1.

4.3 Fault Tolerant QPIP

We now prove Theorem 1.3, which implies our main result, namely a fault tolerant QPIP for BQP.

Proof: Our proof combines several known fault-tolerant quantum computation techniques. However, fault tolerance does not follow straight-forwardly, since care must be given to the fact that the verifier is the only one who can authenticate qubits, while he cannot authenticate many qubits in parallel.

The proof can be divided into three stages.

In the first stage, the prover receives authenticated qudits from the verifier, one by one. Each qudit is authenticated on m qudits. The prover ignores the authentication structure and begins encoding each qudit out of the m qudits separately using a concatenated error correction code, with total length which is polylogarithmic (in n- the number of input qudits and also in m - which is a constant), as is required for the fault tolerance scheme in [3]. From the work of [3, 17] (and others) we know that this encoding can be done in a fault tolerant way, such that if the error probability was less than some threshold η, then the encoded qudit is faulty (namely, has an effective error) with probability at most η', where η' is a constant that depends on η and other parameters of the encoding scheme, but not on n. We denote this concatenated encoding procedure by $\bar{S}$.

Since each authenticated qudit sent to the prover is encoded using a constant number (m) of qudits, it follows that with a constant probability, η'' all these qudits are effectively correctly authenticated. In other words, the encoding of $\left|S_a^k\right\rangle$, $(\bar{S}\otimes\ldots\otimes\bar{S})\left|S_a^k\right\rangle$, has no effective faults with probability η''.

Once a qudit has been encoded by the prover, he can keep applying error corrections on that qudit, and thus, can keep its effective error below some constant for a polynomially long time. Polynomially many authenticated qudits are sent this way to the prover.

In the second stage a purification procedure is performed on the authenticated messages, which are now protected from noise by the prover's concatenated error correction code. Since the purification is of the *authenticated* qudits, it is done according to instructions from the verifier. As explained in Section 4.1 the verifier can also interpret measurements outcomes for the prover, which are needed for the purification procedure. We need to purify both input qubits which are without loss of generality $|0\rangle$, and Toffoli states. Any standard purification procedure (for example, that of [8]) would work for the $|0\rangle$ states. In order to purify the Toffoli states we use the purification described in [8]. The purification procedure uses polylogarithmically many qubits in order to provide a total error of at most $\frac{\Delta}{poly(nT)}$, where T is the number of gates in the circuit U that will be computed by the prover. This means (using the union bound) that with probability at most Δ all purified states are effectively correct.

Finally, having with high probability effectively correct input states, the polynomial QPIP (Protocol 4.1) is executed. To do this the prover uses fault tolerant techniques on his side, (with respect to his QECC) to apply any gate he needs to apply. Moreover, the prover keeps correcting his state with respect to his QECC, as in the standard fault tolerant schemes. At the end, a logical measurement of the output bit of the computation is executed by the prover. The result is sent to the verifier who subsequently interprets it according to his secret key.

The soundness of the this fault tolerant QPIP is the same as that of the standard QPIP. In fact, in this scheme, the verifier ignores the prover's overhead of encoding the input in an error correcting code, and performing encoded operations. The verifier can be thought as performing Protocol 4.1 for a purification circuit followed by the circuit he is interested in computing. Therefore, the security proof of Theorem 4.1 proves in fact that applying the purification and computation circuits, has the same soundness parameter as the standard QPIP.

Regarding completeness, the fact that the prover's computation is noisy changes the error probability only very slightly. There is a probability Δ that one of the input authenticated states is effectively incorrect; Once they are all correct, the fault tolerance proof implies that they remain correct the entire computation with all but an inverse polynomial probability. Therefore, if the standard QPIP protocol has completeness $1-\delta-\epsilon$ the completeness of this scheme is bounded by $1-\delta-\epsilon-2\Delta$.

5 Blind QPIP

Definition 5.1 *[6, 10, 11] 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 function $\mathfrak{F}_n$ the prover's reduced density matrix is identical for every $f \in \mathfrak{F}_n$.*
- ***Security**: Completeness and soundness hold the same way as in QAS (Definition 2.5).*

We would now like to prove Theorem 1.4, namely, that the protocols described so far can be made blind. We use the simple observation that the input is completely hidden from the prover. This holds since in both QASs presented

the density matrix that describes the prover's state does not depend on the input to the circuit. Specifically, due to the randomized selection of an authentication, the prover's state is the completely mixed state. We also use the notion of 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. Constructions of such universal circuits are left as an easy exercise to the reader.

Having mentioned the above observations, a blind computation protocol is not hard to devise. The verifier will, regardless of the input, compute, with the prover's help, the result of the universal circuit acting on input and control bits.

For completeness we first formally define a universal circuit:

Definition 5.2 *The universal circuit $\mathfrak{U}_{n,k}$ acts in the following way:*

$$\mathfrak{U}_{n,k} |\phi\rangle \otimes |c(U)\rangle \longrightarrow U |\phi\rangle |c(U)\rangle \tag{17}$$

Where $c(U)$ is the canonical (classical) description of the circuit U.

*P*roof of **Theorem 1.4**: We prove that both the Clifford based QPIP and the Polynomial QPIP can be used to create a blind computation protocol. We claim that the state of the prover through the protocols is described by the completely mixed state. This is true in the Polynomial scheme since the Pauli randomization does exactly that. Averaging over all possible Pauli keys, it is easy to check that the state of the prover is described by $\mathcal{I}/2^m$. Furthermore, the prover gains no information regarding the Pauli key during the protocol, therefore, the description of the state does not change during the protocol as claimed.

Since the above holds for any initial state, it follows that the prover has no information about the initial, intermediate or final state of the system.

To see that the same argument holds for the Clifford QAS, it suffices to notice that applying a random Clifford operator "includes" the application of a random Pauli:

$$\frac{1}{|\mathfrak{C}_n|}\sum_{c\in\mathfrak{C}_n} C\rho C^\dagger = \frac{1}{|\mathfrak{C}_n|}\sum_{c\in\mathfrak{C}_n} (CQ)\rho(CQ)^\dagger \tag{18}$$

Equality holds for any $Q \in \mathfrak{C}_n$ since it is nothing but a change of order of summation.

$$\begin{aligned}
\ldots &= \sum_{Q\in\mathbb{P}_n}\frac{1}{|\mathbb{P}_n|}\frac{1}{|\mathfrak{C}_n|}\sum_{c\in\mathfrak{C}_n} C(Q\rho Q^\dagger)C^\dagger \\
&= \frac{1}{|\mathfrak{C}_n|}\sum_{c\in\mathfrak{C}_n}\frac{1}{|\mathbb{P}_n|}\sum_{Q\in\mathbb{P}_n} C(Q\rho Q^\dagger)C^\dagger \\
&= \frac{1}{|\mathfrak{C}_n|}\sum_{c\in\mathfrak{C}_n} C\Big(\frac{1}{|\mathbb{P}_n|}\sum_{Q\in\mathbb{P}_n}\left(Q\rho Q^\dagger\right)\Big)C^\dagger \\
&= \frac{1}{|\mathfrak{C}_n|}\sum_{c\in\mathfrak{C}_n} C\left(\mathcal{I}/2^m\right)C^\dagger \\
&= \mathcal{I}/2^m
\end{aligned} \tag{19}$$

6 Interpretation of Results

We will now prove the various corollaries, having to do with how the results can be interpreted. We start with the proof of Corollary 1.5 which provides a guarantee on the closeness of the final output state to the correct state, given that the verifier did not abort.

*P*roof of **Corollary 1.5**: Let us first deal with the Clifford based QPIP. We assume that the soundness of the scheme is δ and that the prover applies a strategy on which the verifier does not abort with probability γ. The final state of the protocol before the verifier's cheat detection can be written as (see Eq. 7):

$$s\rho_c + \frac{(1-s)}{4^n-1}\sum_{Q\in\mathbb{P}_n\setminus\{\mathcal{I}\}}\left(Q\rho_c Q^\dagger\right) \tag{20}$$

Where ρ_c is the correct final state of the protocol. After the verifier applies the cheat detection procedure $\mathcal{B}$ (which checks that the control registers are indeed in the $|0\rangle$ state):

$$\begin{aligned}
&s\rho_c \otimes |VAL\rangle\langle VAL| + \\
&\alpha_{rej}\rho_{rej} \otimes |ABR\rangle\langle ABR| + \\
&\alpha_{bad}\rho_{bad} \otimes |VAL\rangle\langle VAL|
\end{aligned} \tag{21}$$

Assume the verifier declares the computation valid, then his state is:

$$\frac{s\rho_c + \alpha_{bad}\rho_{bad}}{1-\alpha_{rej}} \otimes |VAL\rangle\langle VAL| \tag{22}$$

then the trace distance to the correct state ρ_c is bounded by:

$$1 - \frac{s}{1-\alpha_{rej}} + \frac{\alpha_{bad}}{1-\alpha_{rej}} = \frac{2\alpha_{bad}}{1-\alpha_{rej}} \le \frac{2\delta}{\gamma} \tag{23}$$

Were the inequality follows from the security of the QPIP protocol: $\alpha_{bad} \le \delta$, and the fact that the non-aborting probability γ is equal to $\alpha_{bad} + s$.

The proof that the Polynomial based QPIP has the same property follows the exact same lines.

Next we prove Corollary 1.6, which states that if the QPIP system passes sufficiently difficult tests, is cannot be simulated efficiently by a BPP machine.

Proof of **Corollary 1.6** :: *(sketch)* We assume there is a language L which is ϵ hard to compute with respect to some samplable distribution $\mathcal{D} = \mathcal{D}_l$ (where l is the size of the input). Namely for any classical algorithm A it holds that

$$\Pr_{x\sim\mathcal{D}}(A(x) = \mathcal{I}_L(x)) \leq 1 - \epsilon \tag{24}$$

where $\mathcal{I}_L$ is the indicator function of L. (we need only the inequality to hold for all $|x| > l_0$ for some $l_0 \in \mathbb{N}$). Alternatively, if we consider the distributions induced by $\mathcal{D}$ on pairs of inputs and answers: $\{0,1\}^l \times \{0,1\}$ (we denote for example $\widetilde{A} \sim (x, A(x))$) we can say that $\left\langle \widetilde{A}, \widetilde{\mathcal{I}_L} \right\rangle_{TVD} \geq 2\epsilon$, where TVD denotes the total variation distance.

Let us fix some interaction between the verifier an a prover (not neccesarly honest) in which the verifier with probability at least $1-\gamma$ the verifier does not abort. Let us consider the *quantum* algorithm Q_L which computes $\mathcal{I}_L$ with probability $> 1-\delta$. By the same argument as before we have $\left\langle \widetilde{\mathcal{I}_L}, \widetilde{Q_L} \right\rangle_{TVD} < 2\delta$. Now finally, if we assume that A can simulate the interaction between the prover and the verifier which computes Q_L, then we know from Corollary 1.5 that $\left\langle \widetilde{A}, \widetilde{Q_L} \right\rangle_{TVD} < \frac{2\delta}{\gamma}$, since A can always answer according to the probability distribution resulting from the interaction, that is the density matrix, which is in turn close to Q_L.

Some simple algebra shows that A is $2\frac{\delta(1+\gamma)}{\gamma}$ close to $\mathcal{I}_L$ therefore we deduce that if $\epsilon > \delta\frac{1+\gamma}{\gamma}$ then no such algorithm A can exist.

Finally, we sketch the proof of Claim 1.7:

Proof of **Claim 1.7**: We first notice that our security proofs regarding a cheating prover do not assume any computational restriction on the prover. We would like to show that a prover with only a constant number of qubits (but computationally unbounded) can't convince the verifier of even a true statement.

Let us consider the following challenge. The verifier selects a random string $r \in \{0,1\}^n$, and prepares the state $|r\rangle$. At each location the verifier applies a Hadamard with probability $\frac{1}{2}$ and in either case he authenticates the resulting qubit, using the polynomial codes QAS and sends it to the prover. After sending all the qubits to the prover the verifier asks the prover to measure all qubits in the correct basis. The prover passes the challenge if all measurement outcomes are correctly authenticated and are compatible with the states that were initially sent by the verifier. To transform this challenge into a language in BQP one can think of the language $f(x_1, ...x_n, i_1, ..., i_{log(n)}) = x_{i_1,...,i_{log(n)}}$, namely the language in which the input is a string of bits followed by $log(n)$ bits that provide an index i out of n; the output is x_i. The fact that a bounded memory prover can pass the challenge with only a negligible probability better than a guess follows from bounded storage results.

7 Symmetric Definition of QPIP

The definitions and results presented so far are asymmetric. They refer to a setting where the provers wishes to convince the verifier solely of YES instances (of problems in BQP). This asymmetry does not seem relevant to our motivations. We provide a symmetric definition of quantum prover interactive proofs, and show that the two definitions are equivalent. Essentially, this follows from the trivial observation that the class BQP is closed under complement, that is, $\mathcal{L} \in \mathsf{BQP} \iff \mathcal{L}^c \in \mathsf{BQP}$. We first provide a symmetric definition for QPIP, and then prove the equivalence.

Definition 7.1 *A language $\mathcal{L}$ is in the class symmetric quantum prover interactive proof (QPIP^{sym}) if there exists an interactive protocol with the following properties:*

- *The prover and verifier computational power, and communication, is exactly the same as in the definition of QPIP (Definition 1.1).*
- *The verifier has three possible outcomes: **YES**, **NO**, and **ABORT**:*
  - ***YES**: The verifier is convinced that $x \in \mathcal{L}$.*
  - ***NO**: The verifier is convinced that $x \notin \mathcal{L}$.*
  - ***ABORT**: The verifier caught the prover cheating.*
- *Completeness: There exists a prover $\mathcal{P}$ such that $\forall x \in \{0,1\}^*$ the verifier is correct with high probability:*

$$\Pr\left([\mathcal{V}, \mathcal{P}](x, r) = \mathbb{1}_{\mathcal{L}}\right) \geq \frac{2}{3}$$

  where $\mathbb{1}_{\mathcal{L}}$ is the indicator function of $\mathcal{L}$.
- *Soundness: For any prover $\mathcal{P}'$ and for* **any** *$x \in \{0,1\}^*$, the verifier is mistaken with bounded probability, that is:*

$$\Pr\left([\mathcal{V}, \mathcal{P}](x, r) = 1 - \mathbb{1}_{\mathcal{L}}\right) \leq \frac{1}{3}$$

Theorem 7.1 *For any language $\mathcal{L}$: If $\mathcal{L}, \mathcal{L}^c$ are both in QPIP then $\mathcal{L}, \mathcal{L}^c \in \mathsf{QPIP}^{sym}$*

Proof: Let $\mathcal{V}_{\mathcal{L}}, \mathcal{P}_{\mathcal{L}}$ denote the QPIP 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 $\widetilde{\mathcal{P}}$ and $\widetilde{\mathcal{V}}$ to be QPIP^{sym} verifier and prover in the following way: On the first round the prover $\widetilde{\mathcal{P}}$ sends to $\widetilde{\mathcal{V}}$ "yes" if $x \in \mathcal{L}$

and "no" otherwise. Now, both $\widetilde{\mathcal{P}}$ and $\widetilde{\mathcal{V}}$ behave according to $\mathcal{V}_{\mathcal{L}}, \mathcal{P}_{\mathcal{L}}$ if "yes" was sent or according to $\mathcal{V}_{\mathcal{L}^c}, \mathcal{P}_{\mathcal{L}^c}$ otherwise. Soundness and completeness follows immediately from the definition.

Since BQP is closed under complement, we get:

Corollary 7.2 $\mathsf{BQP} = \mathit{QPIP}^{sym}$

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 3.2.

References

[1] S. Aaronson. *Shtetl-Optimized: The Aaronson 25.00 Prize*. http://scottaaronson.com/blog/?p=284, 2007. [Online; accessed 13-Feb-2009].

[2] 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.

[3] 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.

[4] 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.

[5] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. to appear: http://www.cs.princeton.edu/theory/complexity.

[6] P. Arrighi and L. Salvail. Blind Quantum Computation. *International Journal of Quantum Information*, 4(5):883–898, 2006.

[7] 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.

[8] 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.

[9] S. Bravyi and A. Kitaev. Universal quantum computation with ideal Clifford gates and noisy ancillas. *Physical Review A*, 71(2):22316, 2005.

[10] A. Broadbent, J. Fitzsimons, and E. Kashefi. Universal blind quantum computation. *Arxiv preprint arXiv:0807.4154*, 2008.

[11] A. Childs. Secure assisted quantum computation. *Arxiv preprint quant-ph/0111046*, 2001.

[12] C. Dankert, R. Cleve, J. Emerson, and E. Livine. Exact and Approximate Unitary 2-Designs: Constructions and Applications. *Arxiv preprint quant-ph/0606161*, 2006.

[13] 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.

[14] J. Kempe, A. Kitaev, and O. Regev. The Complexity of the Local Hamiltonian Problem. *SIAM JOURNAL ON COMPUTING*, 35(5):1070, 2006.

[15] A. Kitaev, A. Shen, and M. Vyalyi. *Classical and Quantum Computation*. American Mathematical Society, 2002.

[16] E. Knill and R. Laflamme. Quantum computing and quadratically signed weight enumerators. *Information Processing Letters*, 79(4):173–179, 2001.

[17] E. Knill, R. Laflamme, and W. Zurek. Resilient Quantum Computation. *Science*, 279(5349):342, 1998.

[18] M. Nielsen, I. Chuang, and L. Grover. *Quantum Computation and Quantum Information*. Cambridge Univ. Press, 2000.

[19] 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.

[20] P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM journal on computing(Print)*, 26(5):1484–1509, 1997.

[21] U. Vazirani. Computational constraints on scientific theories: insights from quantum computing. http://www.cs.caltech.edu/~schulman/Workshops/CS-Lens-2/cs-lens-2.html, 2007.

[22] J. Watrous. Space-Bounded Quantum Complexity. *Journal of Computer and System Sciences*, 59(2):281–326, 1999.

[23] J. Watrous. PSPACE has constant-round quantum interactive proof systems. *Theoretical Computer Science*, 292(3):575–588, 2003.

[24] Wikipedia. Blind experiment — Wikipedia, the free encyclopedia, 2008. [Online; accessed 20-Oct-2008].

[25] P. Wocjan and S. Zhang. Several natural BQP-complete problems arXiv: quant-ph/0606179. 2006.

[26] J. Yaari. Preprint: *Interactive Proofs as a Theory of Confirmation*. PhD thesis, The Hebrew University of Jerusalem, 2008.

A Security Proof of Polynomial QAS

A.1 Security Against Pauli Attacks

Lemma A.1 *The polynomial QAS is secure against (generalized) Pauli attacks, that is, in the case where the adver-*

sary applies a Pauli operator. In this case the projection of Bob's state on the space spanned by $P_1 |\psi\rangle$ is at least $1 - 2^{-d}$.

Proof: Let us consider the effect of a Pauli Q operator on the signed polynomial code $\mathcal{C}_k$. We first show that with probability $1 - 2^{-d}$ over the sign key k, the effect of Q is detected by the error detection procedure.

Let $Q_x \neq \mathcal{I}$ be a Pauli operator $Q_x = X^{x_i} \otimes \ldots \otimes X^{x_m}$ where $x \in F_q^m$. The effect of Q_x on the code is an addition of x_i to the $i'th$ qubit. This addition passes the error detection step only if coincides with the values of a **signed** polynomial of degree at most d. We consider two cases depending on the weight of x:

- If $|x| \leq d$: let us denote by g the polynomial that satisfies $\forall_i k_i g(\alpha_i) = x_i$, since $Q_x \neq \mathcal{I}$ we know that $g \neq 0$. then g has at least $m - d$ zeros. Since g is nonzero it must have degree at of least: $m - d = d + 1$. Such an attack will be detected with certainty by the error detection procedure.
- Otherwise, assume without loss of generality that $x_i \neq 0$ for $i \leq |x|$. There is exactly one polynomial f of degree at most d such that $\forall_{i \geq d+1}\ k_i f(\alpha_i) = x_i$. For the attack of Eve to be undetected x must agree with f on the remaining coordinates as well:

$$\Pr_k(\forall_{i \leq d}\ x_i = k_i f(\alpha_i)) = \prod_{i=1}^{d} \Pr_k(k_i = x_i^{-1} f(\alpha_i)) \quad (25)$$

  Equality holds since: k_i are independent, $k_i = k_i^{-1}$ and $x_i \neq 0$ for $i \leq d$. Since $k_i = c$ with probability at most half we conclude that the probability that Eve's attack is undetected is at most 2^{-d}.

Now that we have proved the claim for operators of the form Q_x, we handle the general case. Pauli Z are mapped in the dual code to X operators. Since the signed polynomial code is self dual, Q_z attacks will be caught with probability $1 - 2^{-d}$ as well. To conclude the proof we notice that detection Q_x attacks do not depend on the existence Q_z attacks, therefore, a non identity operator $Q_{x,z} = P_z P_x$ will be detected with the correct probability since either x or z must be non trivial.

What remains is to notice that the Pauli randomization $P_{x,z}$ simply shifts any attack Q on the authenticated message to a different Pauli. That is the effect on the signed polynomial code is $P_{x,z}^\dagger Q P_{x,z}$. We conclude that any Pauli operator acting on the polynomial QAS is detected with a probability of at least $1 - 2^{-d}$ as claimed.

A.2 Security Against General Attacks

We start with a simple lemma about pauli operators.

Lemma A.2 *Let $P \neq P'$ be generalized Pauli operators. Then: $\sum_{Q \in \mathbb{P}_m} Q^\dagger P Q \rho Q^\dagger P'^\dagger Q = 0$*

*P*roof of **Lemma A.2**: Let $P \neq P'$ be generalized Pauli operator $P = X^a Z^b$ and $P' = X^{a'} Z^{b'}$. We use the fact that $Z^d X^c = \omega_q^{dc} X^c Z^d$ and some algebra:

$$\begin{aligned} &\sum_{Q \in \mathbb{P}_1} Q^\dagger P Q \rho Q^\dagger P'^\dagger Q \\ &= \sum_{d,c=0}^{q-1} \omega_q^{d(a-a')+c(b-b')} X^a Z^b \rho Z^{-b'} X^{-a'} \\ &= X^a Z^b \rho Z^{-b'} X^{-a'} \sum_{c=0}^{q-1} \omega_q^{c(b-b')} \sum_{d=0}^{q-1} \omega_q^{d(a-a')} \end{aligned} \quad (26)$$

To conclude the proof we recall that $a \neq a'$ or $b \neq b'$, hence one of the above sums vanishes. The claim for system in higher dimension follows immediately.

In addition we need one more simple lemma:

Lemma A.3 *For any generalized Pauli operator P:*

$$\frac{1}{|\mathbb{P}_m|} \sum_{Q \in \mathbb{P}_m} Q^\dagger P Q \rho Q^\dagger P^\dagger Q = P \rho P^\dagger \quad (27)$$

*P*roof of **Lemma A.3**: From the observation about generalized Pauli operators in Sec. 2 we know that for any two generalized Pauli operators P, Q $PQ = \alpha QP$ where α is some phase dependent on P and Q.

$$\begin{aligned} &\frac{1}{|\mathbb{P}_m|} \sum_{Q \in \mathbb{P}_m} Q^\dagger P Q \rho Q^\dagger P^\dagger Q \\ &= \frac{1}{|\mathbb{P}_m|} \sum_{Q \in \mathbb{P}_m} Q^\dagger (\alpha Q P) \rho (\alpha^* P^\dagger Q^\dagger) Q \\ &= \frac{1}{|\mathbb{P}_m|} \sum_{Q \in \mathbb{P}_m} \alpha P \rho \alpha^* P^\dagger \\ &= P \rho P^\dagger \end{aligned} \quad (28)$$

*P*roof of **Theorem 2.2**: In order to prove this theorem we essentially analyze the effect of the (generalized) Pauli-twirl on signed polynomial code word. We will show that for any attack the effect is as a mixture of Pauli operators which is in turn detected with hight probability as shown in Lemma A.1. For clarity, we omit the normalization factor $|\mathbb{P}_m|$. In addition we denote $\widetilde{Q} = Q \otimes \mathcal{I}_E$. We start by decomposing any attack $V \in \mathbb{U}(M \otimes E)$ made by Eve to $V = \sum_{P \in \mathbb{P}_m} P \otimes U_P$. Bob's state prior to applying the

error detection procedure is:

$$\begin{aligned}\rho_{Bob} &= \mathrm{Tr}_E\Big(\sum_{Q\in\mathbb{P}_m}\widetilde{Q}^\dagger V\left(\widetilde{Q}\rho\otimes\rho_E\widetilde{Q}^\dagger\right)V^\dagger\widetilde{Q}\Big)\\ &= \mathrm{Tr}_E\Big(\sum_{P,P'\in\mathbb{P}_m}\sum_{Q\in\mathbb{P}_m}\widetilde{Q}^\dagger P\otimes U_P \\ &\qquad \left(\widetilde{Q}\rho\otimes\rho_E\widetilde{Q}^\dagger\right)P'\otimes U_{P'}^\dagger\widetilde{Q}\Big)\end{aligned} \tag{29}$$

Writing $\widetilde{Q}$ explicitly and regrouping elements operating on M and on E we have:

$$\begin{aligned}\ldots =&\mathrm{Tr}_E\Big(\sum_{P,P'\in\mathbb{P}_m}\sum_{Q\in\mathbb{P}_m}\left(Q^\dagger PQ\rho Q^\dagger P'Q\right)\otimes \\ &U_P\rho_E U_{P'}^\dagger\Big) \\ =&\sum_{P,P',Q\in\mathbb{P}_m}\mathrm{Tr}\Big(U_P\rho_E U_{P'}^\dagger\Big)\cdot\left(Q^\dagger PQ\rho Q^\dagger P'Q\right)\end{aligned} \tag{30}$$

We use Lemma A.2 and are left only with $P=P'$

$$\ldots=\sum_{P,Q\in\mathbb{P}_m}\mathrm{Tr}\Big(U_P\rho_E U_P^\dagger\Big)\cdot\left(Q^\dagger PQ\rho Q^\dagger PQ\right) \tag{31}$$

Now we use Lemma A.3 :

$$\ldots=\sum_{P\in\mathbb{P}_m}\mathrm{Tr}\Big(U_P^\dagger U_P\rho_E\Big)\cdot|\mathbb{P}_m|P\rho P^\dagger \tag{32}$$

We set $\alpha_P=\mathrm{Tr}\big(U_P^\dagger U_P\rho_E\big)$ and we rewrite Bob's state after normalization:

$$\alpha_{\mathcal{I}}\cdot\rho+\sum_{P\in\mathbb{P}_m\setminus\{\mathcal{I}\}}\alpha_P\cdot P\rho P^\dagger \tag{33}$$

Recall that we are interested in the projection of ρ_{Bob} on the subspace spanned by the operator $P_1^{|\psi\rangle}$.

$$\begin{aligned}&\mathrm{Tr}\Big(P_1^{|\psi\rangle}\big(\alpha_{\mathcal{I}}\cdot\rho+\sum_{P\in\mathbb{P}_m\setminus\{\mathcal{I}\}}\alpha_P\cdot P\rho P^\dagger\big)\Big)\\ &=\alpha_{\mathcal{I}}+\sum_{P\in\mathbb{P}_m\setminus\{\mathcal{I}\}}\alpha_P\mathrm{Tr}\Big(P_1^{|\psi\rangle}P\rho P^\dagger\Big)\end{aligned} \tag{34}$$

We use the bound from Lemma A.1:

$$\begin{aligned}\ldots\geq\quad&\alpha_{\mathcal{I}}+\sum_{P\in\mathbb{P}_m\setminus\{\mathcal{I}\}}\alpha_P\left(1-2^{-d}\right)\\ =\quad&(1-\frac{1-\alpha_{\mathcal{I}}}{2^d})\end{aligned} \tag{35}$$

Which concludes the proof. Similarly to the random Clifford authentication scheme, the further Eve's intervention is closer to the identity, that is – Eve does almost nothing, then the projection on the good subspace is closer to 1.

A.3 Polynomial QAS Applied in Parallel

When authenticating multiple registers, it may seem at first glance that Eve has the advantage of being able to tamper with the state by applying some transformation on the entire space. In the Clifford authentication protocol applied in p, the intervention of Eve is broken down to individual attacks on each register by the fact random Clifford operators are applied to each register independently.

The main idea for the concatenated polynomial authentication is to use an independent Pauli key (x,z) for each register, while maintaining the sign key k equal between registers. This idea will suffice to brake up the attack of Eve to a sequence of attacks on each register separately.

Protocol A.1 ***Concatenated polynomial Authentication protocol:***
Alice wishes to send a state $|\psi\rangle\in(\mathcal{C}^q)^{\otimes r}$ that is r q-dimensional systems. For a security parameter d, set $m=2d+1$. Alice randomly selects a single sign key $k\in\{\pm1\}^m$, furthermore, Alice selects r independent Pauli keys (x_j,z_j).

To encode $|\psi\rangle$ Alice encodes each q-dimensional system using the signed polynomial code specified by k. Additionally, Alice shifts the j'th encoded message by $P_{(x_j,y_j)}$.

Bob decodes each message separately, if all messages are correctly authenticated Bob declares as valid the concatenated message, otherwise Bob aborts.

We now prove Theorem 2.3.

*P*roof of **Theorem 2.3**: We notice that all the reasoning in Theorem 2.2 till Eq. 35 holds in this case as well. So we have that the projection on the good subspace $P_1^{|\psi\rangle}$ is equal to:

$$\alpha_{\mathcal{I}}+\sum_{P\in\mathbb{P}_{r\cdot m}\setminus\{\mathcal{I}\}}\alpha_P\mathrm{Tr}\left(P_1^{|\psi\rangle}P\rho P^\dagger\right) \tag{36}$$

We start by writing $\mathrm{Tr}(P_1^{|\psi\rangle}P\rho P^\dagger)=1-\mathrm{Tr}(P_0^{|\psi\rangle}P\rho P^\dagger)$. We recall that P here is a Pauli operator from the group $\mathbb{P}_{m\cdot r}$ so we write: $P=P_{(1)}\otimes\ldots\otimes P_{(r)}$.

Lemma A.4 *The probability for Bob to be fooled by the application of $P\neq\mathcal{I}$ is at most 2^{-d}.*

Proof: For $P\rho P^\dagger$ to be in $P_0^{|\phi\rangle}$ it must be the case that for all j such that $P_{(j)}\neq\mathcal{I}$ Eve escapes detection (Bob does not abort although the register is "corrupted"). We note that Bob declares as valid the remaining registers (where $P_{(j)}=\mathcal{I}$) with certainty. We assume without loss of generality that $P_{(1)}\neq\mathcal{I}$, we write the probability that Bob

is fooled:

$$\begin{aligned}
&\Pr\left(\textit{Bob is fooled by } P\right)\\
=&\Pr\left(\forall_{j:P_{(j)}\neq\mathcal{I}}\textit{Bob is fooled by } P_{(j)}\right)\\
\leq&\Pr\left(\textit{Bob is fooled by } P_{(1)}\right)\\
\leq&2^{-d}
\end{aligned} \tag{37}$$

Where the last inequality holds by Lemma A.1.

Plugging this result into Eq. 35 we have:

$$\begin{aligned}
\ldots=&\alpha_{\mathcal{I}}+\sum_{P\in\mathbb{P}_{r\cdot m}\setminus\{\mathcal{I}\}}\alpha_P\left(1-\mathrm{Tr}(P_0^{|\psi\rangle}P\rho P^\dagger)\right)\\
\geq&\alpha_{\mathcal{I}}+\sum_{P\in\mathbb{P}_{r\cdot m}\setminus\{\mathcal{I}\}}\alpha_P\left(1-2^{-d}\right)\\
=&\left(1-\frac{1-\alpha_{\mathcal{I}}}{2^d}\right)
\end{aligned} \tag{38}$$

Which concludes the proof.