

Classical Verification of Quantum Computations

Urmila Mahadev*

September 27, 2019

Abstract

We present the first protocol allowing a classical computer to interactively verify the result of an efficient quantum computation. We achieve this by constructing a *measurement protocol*, which enables a classical verifier to ensure that the quantum prover holds an n qubit quantum state, and correctly reports the results of measuring it in a basis of the verifier's choice. This is enforced based on the assumption that the learning with errors problem is computationally intractable for efficient quantum machines.

1 Introduction

We propose a solution to the open question of verifying quantum computations through purely classical means. The question is as follows: is it possible for an efficient classical verifier (a **BPP** machine) to verify the output of an efficient quantum prover (a **BQP** machine)? This question was first asked by Gottesman in 2004 ([Got04]) and has since been extensively studied ([GKK17]). Since it was unknown how to verify a quantum computation with a classical verifier, the question was relaxed in two different ways. The first relaxation asked whether a classical machine with access to a small quantum computer could verify the result of a quantum computation. This question was answered affirmatively by a series of works, beginning with [BFK08]/[FK17], [ABOE08]/[ABOEM17]. The second relaxation of the question asked whether a **BPP** verifier could verify the result of a **BQP** computation given access to two **BQP** machines. This question was also answered affirmatively in [RUV12]. However, the original question with a single classical verifier and a quantum prover has remained open.

Recently, there has been progress on a related question: it was shown that it is possible for a classical machine to outsource a quantum computation to a **BQP** server while ensuring that the **BQP** server could learn nothing about the computation ([Mah17]). It may seem like there is a straightforward link between the two questions: if blind delegation of quantum computation is possible, shouldn't it be possible to verify that the quantum computation is performed correctly? For example, the classical computer could delegate several quantum computations in a random order, half of which he knows the answer to. Upon receiving the answers from the prover, he could check that half are correct, and could therefore be convinced that the other half are also correct.

The link between blind and verifiable computation does exist with respect to classical computations, and relies on malleability properties of classical encryption schemes. In a blind computation, it may be possible for a prover to deviate from the honest strategy by mapping one valid encryption to another, without knowing the underlying value of either encryption. If an encryption scheme can be made non-malleable, this type of cheating can be prevented, thereby allowing the extension of a blind scheme to a verifiable scheme. The issue of malleability presents a significant obstacle in the context of quantum computations, due to the inherent randomness involved;

*Department of Computer Science, UC Berkeley, USA. Supported by Templeton Foundation Grant 52536, ARO Grant W911NF-12-1-0541, NSF Grant CCF-1410022 and MURI Grant FA9550-18-1-0161. Email: mahadev@cs.berkeley.edu.

all known blind quantum computation schemes (including the scheme given in [Mah17]) involve uniformly random outputs, even in the case of an honest prover.

Due to this issue, we chose to pursue an entirely different route to achieve classical verification of quantum computations. The core of our measurement protocol is a trapdoor claw-free function, a two-to-one function for which it is hard to find a pair of two preimages (i.e. a claw). Using this primitive, the classical verifier can force a quantum prover to encode a quantum state in a way that is unknown to the prover. By strengthening the claw-free property in two different ways, this encoding can be used to randomize any operator applied by the prover which is a deviation from the strategy of an honest prover, rendering the deviation of the prover essentially useless. We provide the definition and construction of an *extended trapdoor claw-free family* which satisfies the strengthened properties. This family is an extension of the family given in [BCM⁺18]. Like the construction in [BCM⁺18], our construction relies on the computational assumption that a BQP machine cannot solve the learning with errors problem with superpolynomial noise ratio ([Reg05]).

1.1 Related Work

The basic idea of using trapdoor claw-free functions to constrain the power of the prover in the context of randomness generation was introduced in [BCM⁺18]. This idea was further used in a computational setting to hide information from the prover (but not constrain the prover) in [Mah17]. What is new to our paper is that it constrains the prover in the context of carrying out a particular computation. This requires the verifier to exert a much greater degree of control over the prover than in randomness generation. To accomplish this goal, we develop a new protocol and proof techniques for strongly characterizing the state of an untrusted prover.

2 Overview

In this paper, we construct a *measurement protocol* between an efficient quantum prover and an efficient classical verifier. In our protocol, an honest prover constructs an n qubit quantum state ρ of his choice, and the verifier would like each qubit to be measured in either the standard basis or the Hadamard basis. The goal of the protocol is to guarantee that, even in the case of a dishonest prover, the measurement result $m \in \{0, 1\}^n$ obtained by the verifier is indeed the result of measuring some n qubit quantum state in the desired basis. One of the features of the measurement protocol is that the choice of basis is sent to the prover in an encrypted form, and the prover ends up with no information about the measurement basis chosen by the verifier.

Denote the choice of measurement basis by an n bit string $h = (h_1, \dots, h_n)$. For a prover \mathbb{P} and a measurement basis choice $h = (h_1, \dots, h_n)$, we define $D_{\mathbb{P},h}$ to be the resulting distribution over m . For an n qubit state ρ , we define $D_{\rho,h}$ to be the distribution obtained by measuring ρ in the basis corresponding to h . Our measurement protocol is complete, in the following sense: for all efficiently computable n qubit states ρ , there exists a prover \mathbb{P} such that $D_{\mathbb{P},h}$ is approximately equal to $D_{\rho,h}$ for all h . Moreover, \mathbb{P} is accepted by the verifier with all but negligible probability.

Our soundness notion for the measurement protocol is as follows. We will show that if the prover \mathbb{P} is accepted by the verifier with perfect probability, there exists an efficiently computable n qubit quantum state ρ underlying the distribution over m . More precisely, for all $h \in \{0, 1\}^n$, $D_{\rho,h}$ is computationally indistinguishable from $D_{\mathbb{P},h}$. This statement can easily be made robust: we will show that if a prover \mathbb{P} is accepted by the verifier on basis choice h with probability $1 - p_h$, there exists a prover \mathbb{P}' who is always accepted by the verifier and the statistical distance between $D_{\mathbb{P},h}$ and $D_{\mathbb{P}',h}$ is approximately $\sqrt{p_h}$ for all h . Given the strength of this soundness guarantee, it is straightforward to extend our measurement protocol to a protocol which can be used to verify membership in

all languages in BQP (see Section 2.5).

Our measurement protocol relies on two cryptographic primitives which give a BPP verifier some leverage over a BQP prover. We begin by describing these primitives in Section 2.1. Using these primitives, we can describe our measurement protocol in Section 2.2. In Sections 2.2 - 2.4, we show how the two cryptographic primitives can be used to guarantee that all provers must essentially be creating a state and measuring it in the basis chosen by the verifier. In Section 2.5, we show how to extend our measurement protocol to a verification protocol for all of BQP.

2.1 Cryptographic Primitives

2.1.1 Trapdoor Claw-Free Families

The first cryptographic primitive we will use is a function family $\mathcal{F} = \{f_{k,b} : \mathcal{X} \rightarrow \mathcal{Y}\}$ (for $b \in \{0,1\}$) called a *trapdoor claw-free* function family. For convenience, we will assume in this overview that $\mathcal{X} = \{0,1\}^w$. A trapdoor claw-free function family is a family of functions which are two to one (both $f_{k,0}(\cdot)$ and $f_{k,1}(\cdot)$ are injective and their images are equal) and for which it is computationally difficult to find a *claw*, i.e. a pair of points $(0, x_0)$ and $(1, x_1)$ which have the same image ($f_{k,0}(x_0) = f_{k,1}(x_1)$). Given y in the image of $f_{k,0}$ or $f_{k,1}$, the trapdoor t_k of the functions $f_{k,0}, f_{k,1}$ allows recovery of both preimages of y . The trapdoor claw-free family also satisfies two hardcore bit properties, which are stronger versions of the claw-free property: roughly, they state that it is computationally difficult to find a string d and the bit $d \cdot (x_0 \oplus x_1)$, where (x_0, x_1) form a claw. These two properties are specified as needed (in Claim 2.1 and Claim 2.2).

In this overview, we will assume the existence of the function family described above for simplicity. However, since we do not know how to construct such a family, in the rest of the paper we will instead rely on an approximate version of this family. We provide the definition of the function family we will use, which we call an extended trapdoor claw-free family, in Section 4, Definition 4.4. The construction of this family (from learning with errors) is given in Section A. Both the definition and construction are extensions of those given in [BCM⁺18].

We now describe a BQP process we call *state commitment*, which requires a function key k corresponding to functions $f_{k,0}, f_{k,1} \in \mathcal{F}$ (we assume that the functions $f_{k,0}, f_{k,1}$ can be computed given access to the function key k). The state commitment process is performed with respect to an arbitrary single qubit state $|\psi\rangle$:

$$|\psi\rangle = \sum_{b \in \{0,1\}} \alpha_b |b\rangle \quad (1)$$

The commitment process consists of two steps. First, the functions $f_{k,0}, f_{k,1}$ are applied in superposition, using $|\psi\rangle$ to determine whether to apply $f_{k,0}$ or $f_{k,1}$ and a uniform superposition over $x \in \mathcal{X}$ as the input to $f_{k,0}$ or $f_{k,1}$:

$$\frac{1}{\sqrt{|\mathcal{X}|}} \sum_{b \in \{0,1\}} \sum_{x \in \mathcal{X}} \alpha_b |b\rangle |x\rangle |f_{k,b}(x)\rangle \quad (2)$$

Second, the final register of the resulting state is measured, obtaining $y \in \mathcal{Y}$. At this point, the state is:

$$\sum_{b \in \{0,1\}} \alpha_b |b\rangle |x_{b,y}\rangle \quad (3)$$

where $x_{0,y}$ and $x_{1,y}$ are the two preimages of y . We will call the qubit containing b the *committed qubit*, the register containing $x_{b,y}$ the *preimage register* and the string y the *commitment string*. The crucial point here is that, due to the claw-free nature of the functions $f_{k,0}, f_{k,1}$, it is computationally difficult for a BQP machine to compute both

inverses $x_{0,y}$ and $x_{1,y}$ given only y . However, with access to the trapdoor t_k , both inverses can be computed from y . If we think of the state commitment process in an interactive setting, in which the verifier selects the function key and the trapdoor and the BQP prover performs the commitment process (sending the commitment y to the verifier), the BQP prover cannot compute both inverses, but the verifier can. This gives the verifier some leverage over the prover's state.

A key property of the committed state in (3) is that it allows a logical Hadamard measurement up to an X Pauli operator, which is performed as follows. First, a Hadamard transform is applied to both the committed qubit and preimage register of the state in (3):

$$\frac{1}{\sqrt{|\mathcal{X}|}} \sum_{d \in \mathcal{X}} X^{d \cdot (x_{0,y} \oplus x_{1,y})} H |\psi\rangle \otimes Z^{x_{0,y}} |d\rangle \quad (4)$$

The next step in applying the logical Hadamard measurement is to measure the second (preimage) register, obtaining $d \in \mathcal{X}$. The state at this point is:

$$X^{d \cdot (x_{0,y} \oplus x_{1,y})} H |\psi\rangle \quad (5)$$

To obtain the Hadamard measurement of $|\psi\rangle$, the operator $X^{d \cdot (x_{0,y} \oplus x_{1,y})}$ (which we call the decoding operator and requires the trapdoor) is first applied, followed by a standard basis measurement of $H |\psi\rangle$. Note that these two operations commute: it is equivalent to first perform a standard basis measurement of the state in (5) followed by applying the X decoding operator. The X decoding operator applied after measurement is simply the classical XOR operation.

We can again think of this logical Hadamard transform in the interactive setting, in which the BQP prover applies the Hadamard transform to obtain the state in (4) and then measures the committed qubit and preimage register, sending the measurement results $b' \in \{0, 1\}$ and $d \in \{0, 1\}^w$ to the verifier. The verifier decodes the measurement b' by XORing it with $d \cdot (x_{0,y} \oplus x_{1,y})$ (which can be computed using the trapdoor) to obtain the bit m , which the verifier stores as the result of the Hadamard basis measurement.

2.1.2 Trapdoor Injective Function Families

The second primitive is a function family $\mathcal{G} = \{g_{k,b} : \mathcal{X} \rightarrow \mathcal{Y}\}$ (for $b \in \{0, 1\}$) called a *trapdoor injective* function family. A trapdoor injective function family is a family of injective functions such that the images of $g_{k,0}(\cdot)$ and $g_{k,1}(\cdot)$ are disjoint. Given $y = g_{k,b}(x_{b,y})$, the trapdoor t_k of the functions $g_{k,0}, g_{k,1}$ allows recovery of $b, x_{b,y}$. We will also require that the trapdoor injective family is computationally indistinguishable from the trapdoor claw-free family: given a function key k , it must be computationally difficult to determine whether k belongs to an injective or claw-free family. As in the case of trapdoor claw-free families, we will assume the existence of the function family described above for the purpose of this overview, but in the rest of the paper we will rely on an approximate version of this function family. We define the approximate version in Section 4 (Definition 4.2) and construct it from learning with errors in Section A.3.

The state commitment process described in Section 2.1.1 can also be performed with a function key k corresponding to functions $g_{k,0}, g_{k,1} \in \mathcal{G}$. At the stage of (2), the following state has been created:

$$\frac{1}{\sqrt{|\mathcal{X}|}} \sum_{b \in \{0,1\}} \sum_{x \in \mathcal{X}} \alpha_b |b\rangle |x\rangle |g_{k,b}(x)\rangle \quad (6)$$

However, in this case when the last register is measured to obtain $y \in \mathcal{Y}$, the superposition over b collapses. This is because the images of $g_{k,0}(\cdot)$ and $g_{k,1}(\cdot)$ are disjoint. It follows that with probability $|\alpha_b|^2$, $y \in g_{k,b}(\cdot)$. In this case, the state after measurement of y is:

$$|b\rangle |x_{b,y}\rangle \quad (7)$$

where $x_{b,y}$ is the unique preimage of y .

If we think of this process in the interactive setting (in which the BQP prover performs the commitment and sends the verifier y), the verifier can use the trapdoor t_k to extract $(b, x_{b,y})$ from y . Therefore, the verifier can obtain b , which is the result of standard basis measurement of the prover's state, simply by asking for the commitment y .

2.2 Measurement Protocol

Given the two primitives described in Section 2.1, we can now describe our measurement protocol (formally given in Protocol 5.1). Before the protocol begins, the verifier will select the basis for which he would like a measurement result (either Hadamard or standard) for each of n qubits. We will represent this basis choice by a string $h \in \{0, 1\}^n$. If $h_i = 0$, it indicates that the standard basis was chosen for qubit i .

The protocol begins with the commitment round. For all i for which $h_i = 1$, the verifier samples a key k_i and a trapdoor t_{k_i} corresponding to a pair of trapdoor claw-free functions $f_{k_i,0}, f_{k_i,1} \in \mathcal{F}$. For all i for which $h_i = 0$, the verifier samples a key k_i and a trapdoor t_{k_i} corresponding to a pair of trapdoor injective functions $g_{k_i,0}, g_{k_i,1} \in \mathcal{G}$. The verifier sends the prover the function keys k_1, \dots, k_n . The honest prover first constructs an n qubit state of his choice and then commits to each qubit of the state using the corresponding function key, as described in Section 2.1. The verifier receives the measurement results $y_1, \dots, y_n \in \mathcal{Y}$ from the prover.

Once the verifier receives y_1, \dots, y_n , he chooses at random either to run a test round or a Hadamard round. In the case of a test round, the verifier requests standard basis measurements of the committed qubit and preimage register for all n qubits. He receives $b'_i \in \{0, 1\}$ and $x'_i \in \mathcal{X}$ from the prover and checks if the measurement is a preimage of y_i .

In the case of a Hadamard round, the verifier requests a Hadamard basis measurement of the committed qubit and preimage register for all i . The verifier receives $b'_i \in \{0, 1\}, d_i \in \mathcal{X}$ for all i . For all i for which $h_i = 0$, the verifier ignores the measurement results b'_i and d_i and uses the trapdoor of g_{k_i} to invert y_i . The verifier stores the first bit of the inverted value as the standard basis measurement result m_i . For all i for which $h_i = 1$, the verifier first decodes b'_i by XORing it with $d_i \cdot (x_{0,y_i} \oplus x_{1,y_i})$ (this can equivalently be thought of as applying the decoding operator $X^{d \cdot (x_{0,y_i} \oplus x_{1,y_i})}$ - see the end of Section 2.1.1). The verifier stores the result $m_i = b'_i \oplus d_i \cdot (x_{0,y_i} \oplus x_{1,y_i})$ as the Hadamard basis measurement result.

Completeness of our measurement protocol (as defined at the start of Section 2) follows immediately from the description of the state commitment process given in Sections 2.1.1 and 2.1.2.

2.3 Measurement Protocol Soundness

We now give an overview of our soundness guarantee: we describe how to show that for $n = 1$ and a prover \mathbb{P} who passes the test round perfectly, there exists a quantum state underlying the distribution over measurement results obtained by the verifier. The generalization to arbitrary n (given in Section 7) follows easily due to the fact that all n function keys are drawn independently. The generalization to provers \mathbb{P} who do not pass perfectly (also given in Section 7) is straightforward as well; it is done by conditioning \mathbb{P} on acceptance in a test round, thereby creating

an efficient prover who passes the test round perfectly as long as \mathbb{P} is accepted with non negligible probability. In this section, we begin by characterizing the behavior of a general prover. We then show that if this characterization satisfies a certain requirement, we can prove the existence of an underlying quantum state. In Section 2.4 (which is the crux of this paper), we show how to enforce this requirement on general provers.

2.3.1 Prover Behavior

The analysis of the measurement protocol is based on understanding and characterizing the prover's Hilbert space and operations. We will rely on the following principle behind interactive proofs between a BQP prover and a BPP verifier. A round of the protocol begins with the verifier's message and ends with the prover's message. A general prover is equivalent, from the verifier's perspective, to a prover who begins each round by applying an arbitrary unitary operator to his space and then behaves exactly the same as an honest prover, culminating in a measurement (the result of which is sent to the verifier). This principle implies that an arbitrary prover measures the same registers that an honest prover does in each round, which will be particularly useful in our protocol.

Let \mathbb{P}_0 be an honest prover in our measurement protocol and assume the unitary operator he applies in the commitment round is $U_{C,0}$, after which he measures the commitment string register in the standard basis. As described in Section 2.1, \mathbb{P}_0 has three designated registers: the register containing the committed qubit, the preimage register, and the commitment string register. Each message of \mathbb{P}_0 to the verifier is the result of the measurement of one of these registers.

It follows from the principle above that a general prover \mathbb{P} has the same designated registers as \mathbb{P}_0 and is characterized by 3 unitary operators: the unitary U_C applied in the commitment round, the unitary U_T applied in the test round, and the unitary U_H applied in the Hadamard round. We assume that both U_T and U_H do not act on the commitment string register, since it has already been measured; the measurement result could have been copied into the auxiliary space, on which U_T and U_H can act.

We now use the structure of our protocol to simplify the general prover one step further. There are only two possible messages of the verifier for the second round of our protocol: the message indicates either a test or Hadamard round. Due to this property, we can assume that the test round attack U_T is equal to the identity operator. To see this, we only need to make one observation: the attack U_T applied in the test round commutes with the measurement of the commitment string. Therefore, it could have been applied prior to reporting the commitment string.

It follows that the general prover \mathbb{P} described above is identical, from the perspective of the verifier, to a prover who applies the unitary $U_0 = U_T U_{C,0} U_C$ immediately prior to measuring the commitment string register and applies $U = U_H U_T^\dagger$ prior to performing Hadamard basis measurements of the committed qubit and preimage register in the Hadamard round. We will say that such a prover is *characterized* by (U_0, U) . For a formal statement and proof of the above argument, see Claim 5.4.

The characterization of all provers by two unitary attacks allows us to use the test round of the measurement protocol to enforce that the prover's state has a specific structure, which is derived from the cryptographic primitives in Section 2.1. Let \mathbb{P} be a prover who passes the test round perfectly. If $h = 1$, the state of \mathbb{P} at the start of either the test or the Hadamard round (i.e. immediately after reporting y) can be written as follows (the two preimages of y are $x_{0,y}, x_{1,y}$):

$$\sum_{b \in \{0,1\}} |b\rangle |x_{b,y}\rangle |\psi_{b,x_{b,y}}\rangle \quad (8)$$

where $|\psi_{b,x_{b,y}}\rangle$ contains all additional qubits held by the prover. This is because the verifier checks, in a test round, if he receives a valid pre-image from the prover. Since the prover simply measures the requested registers when asked by the verifier in a test round (i.e. he does not apply an attack in the test round), these registers must be in a superposition over the two preimages of the reported measurement result y .

If $h = 0$ and \mathbb{P} reports y , there is only one inverse of y . If we assume this inverse is $x_{b,y}$ (i.e. $g_{k,b}(x_{b,y}) = y$), the state of \mathbb{P} at the start of the test or Hadamard round can be written as follows, due to the same reasoning used in (8):

$$|b\rangle |x_{b,y}\rangle |\psi_{b,x_{b,y}}\rangle \quad (9)$$

This structure enforced by the test run is the key to proving the existence of an underlying quantum state, as we will see shortly.

2.3.2 Construction of Underlying Quantum State

We begin by using the characterization of general provers in Section 2.3.1 to define a single qubit state ρ corresponding a prover \mathbb{P} who is characterized by (U_0, U) . Recall that \mathbb{P} has a well defined committed qubit, which he measures when the verifier asks for the measurement of a committed qubit. Let ρ' be the state of the committed qubit prior to the prover's measurement in the Hadamard round in the case that $h = 1$. We can think of ρ' as encoded by the operator $Z^{d \cdot (x_0, y \oplus x_1, y)}$, which is determined by the prover's measurements d and y . This Z operator is derived from the verifier's X decoding operator applied in the measurement protocol; we have used a Z operator here since the Hadamard measurement has not yet been performed. The single qubit state ρ will be the result of applying the Z decoding operator to the committed qubit ρ' .

Define X -trivial operators to be those which commute with standard basis measurement of the committed qubit. We now show that if the prover's Hadamard round attack U is an X -trivial operator, the distribution $D_{\mathbb{P},h}$ obtained by the verifier in the measurement protocol is computationally indistinguishable from the distribution which is obtained by measuring ρ in basis specified by h .

Recall that $D_{\rho,h}$ is the distribution obtained by measuring ρ in the basis corresponding to h . By construction, $D_{\rho,1} = D_{\mathbb{P},1}$. If $h = 0$, there are two differences between the distribution $D_{\rho,h}$ and the distribution $D_{\mathbb{P},h}$. The first difference lies in the function sampling: in our measurement protocol, an injective function is sampled if $h = 0$, but in the state ρ , a claw-free function is sampled. The second difference comes from how the standard basis measurement is obtained: in $D_{\mathbb{P},h}$ the standard basis measurement is obtained from the commitment string y , but in $D_{\rho,h}$ the standard basis measurement is obtained by measuring ρ (the committed qubit) in the standard basis.

We can handle the first difference by making two key observations. First, the Z decoding operator has no effect if $h = 0$; in this case, the committed qubit will be measured in the standard basis immediately after application of Z in order to obtain $D_{\rho,h}$. Second, if the Z decoding operator is not applied, the trapdoor t_k is no longer needed to construct the distribution $D_{\rho,h}$. If $D_{\rho,h}$ is only dependent on the function key k (and not the trapdoor t_k), the function key k can be replaced with a function key which corresponds to a pair of trapdoor injective functions, rather than a pair of trapdoor claw-free functions, to obtain a computationally indistinguishable distribution. This is due to the computational indistinguishability between keys drawn from the trapdoor claw-free family \mathcal{F} and the trapdoor claw-free family \mathcal{G} .

Let ρ_0 be the committed qubit of the prover prior to measurement in the Hadamard round in the case that $h = 0$. Due to the argument above, the distribution $D_{\rho,0}$ is computationally indistinguishable from $D_{\rho_0,0}$. To address the second difference, we now show that measuring ρ_0 in the standard basis produces the same distribution obtained

from extracting the standard basis measurement from the commitment string y . First, note that measuring the committed qubit prior to application of U (i.e. at the start of the Hadamard round) results in the same measurement obtained from y ; as seen in (9), the value of the committed qubit is equal to the value m extracted from y , since the prover passes the test round perfectly. To complete our proof, recall that U is X -trivial with respect to the committed qubit, and therefore commutes with standard basis measurement of the committed qubit.

To recap, the argument above shows that there exists a quantum state underlying the distribution $D_{\mathbb{P},h}$ as long as the prover's attack operator in the Hadamard round is an X -trivial operator. For a formal statement and complete proof of this argument, see Claim 5.7.

2.4 Replacement of General Attack with an X -Trivial Attack

We can now proceed to the crux of our paper: assuming that $n = 1$ and \mathbb{P} passes the test round perfectly, we show that there exists a prover \mathbb{P}' such that $D_{\mathbb{P},h}$ is computationally indistinguishable from $D_{\mathbb{P}',h}$ for both h and \mathbb{P}' attacks with an X -trivial operator in the Hadamard round. By the argument in Section 2.3.2 and the triangle inequality, this implies that there exists a state ρ for which $D_{\mathbb{P},h}$ and $D_{\rho,h}$ are computationally indistinguishable, thereby proving our soundness guarantee.

Assume \mathbb{P} is characterized by (U_0, U) . Then \mathbb{P}' is characterized by $(U_0, \{U_x\}_{x \in \{0,1\}})$, where $\{U_x\}_{x \in \{0,1\}}$ is an X -trivial CPTP map:

$$U = \sum_{x,z \in \{0,1\}} X^x Z^z \otimes U_{xz} \quad (10)$$

$$U_x = \sum_{z \in \{0,1\}} Z^z \otimes U_{xz} \quad (11)$$

Observe that if $h = 0$, $D_{\mathbb{P},h} = D_{\mathbb{P}',h}$; this is simply because the standard basis measurement is obtained from the commitment y , which is measured prior to the Hadamard round attack U . This argument requires a bit more detail for $n > 1$ and is given formally in Claim 7.4. We proceed to describing how to replace the attack U in (10) with the CPTP map $\{U_x\}_{x \in \{0,1\}}$ in (11) in the case that the verifier chooses the Hadamard basis ($h = 1$). We will rely heavily on the structure of the prover's state, as written in (8).

The replacement of U with $\{U_x\}_{x \in \{0,1\}}$ will be done by using the Z Pauli twirl (Corollary 3.6). The Z Pauli twirl is a technique which allows the replacement of U with the CPTP map $\{U_x\}_{x \in \{0,1\}}$ by conjugating U by a random Z Pauli operator. More formally, Corollary 3.6 states that the following two CPTP maps are equivalent when followed by Hadamard basis measurement:

$$\left\{ \frac{1}{\sqrt{2}} (Z^r \otimes \mathcal{I}) U (Z^r \otimes \mathcal{I}) \right\}_{r \in \{0,1\}} \quad (12)$$

$$\{U_x\}_{x \in \{0,1\}} \quad (13)$$

To apply the Z Pauli twirl in this setting, it suffices to show that replacing the prover's attack U with the unitary attack $(Z \otimes \mathcal{I})U(Z \otimes \mathcal{I})$ results in a computationally indistinguishable distribution.

To prove this statement, we will rely on the fact that there is already computational randomness, due to the trapdoor claw-free function, which is hiding both the Z operator applied prior to U and the Z operator applied after. The computational randomness hiding the posterior Z operator comes from the verifier's decoding operator $X^{d \cdot (x_0, y \oplus x_1, y)}$ applied at the end of the measurement protocol (see Section 2.2); if this decoding operator is shifted

prior to the Hadamard transform on the committed qubit, it acts as a Z operator immediately after the attack U . The computational randomness hiding the anterior Z operator results from the format of the prover's state. Recall that, since the prover is perfect, we can assume the prover begins the Hadamard round with a state of the form in (8):

$$|\phi_y\rangle = \sum_{b \in \{0,1\}} |b\rangle |x_{b,y}\rangle |\psi_{b,x_{b,y}}\rangle \quad (14)$$

Applying a Hadamard transform to the preimage register of the state in (14) results in a Z operator acting on the committed qubit:

$$(\mathcal{I} \otimes H^{\otimes w} \otimes \mathcal{I}) \sum_{b \in \{0,1\}} |b, x_{b,y}\rangle |\psi_{b,x_{b,y}}\rangle = \frac{1}{\sqrt{|\mathcal{X}|}} \sum_{d \in \mathcal{X}, b \in \{0,1\}} Z^{d \cdot (x_{0,y} \oplus x_{1,y})} |b\rangle \otimes Z^{x_{0,y}} |d\rangle \otimes |\psi_{b,x_{b,y}}\rangle \quad (15)$$

In order to use these two sources of computational randomness to hide the difference between U and $(Z \otimes \mathcal{I})U(Z \otimes \mathcal{I})$, it must be the case that the bit $d \cdot (x_{0,y} \oplus x_{1,y})$ is computationally indistinguishable from a uniformly random bit. Formalizing this requirement is a bit tricky, since d is sampled from the state created by the prover. In the next section, we show how to prove computational indistinguishability between the distributions resulting from U and $(Z \otimes \mathcal{I})U(Z \otimes \mathcal{I})$. As part of this process, we formalize the computational randomness requirement regarding $d \cdot (x_{0,y} \oplus x_{1,y})$ as two different hardcore bit conditions for the function pair $f_{k,0}, f_{k,1}$.

2.4.1 Computational Indistinguishability of Phase Flip

Let \mathbb{P} be the prover characterized by (U_0, U) and let $\hat{\mathbb{P}}$ be the prover characterized by $(U_0, (Z \otimes \mathcal{I})U(Z \otimes \mathcal{I}))$. In this section, we will show that the distributions resulting from the two provers ($D_{\mathbb{P},h}$ and $D_{\hat{\mathbb{P}},h}$) are computationally indistinguishable for all h . For convenience, we will instead refer to these two distributions as mixed states; let σ_0 be the mixed state corresponding to $D_{\mathbb{P},h}$ and let σ_1 be the mixed state corresponding to $D_{\hat{\mathbb{P}},h}$, i.e.

$$\sigma_0 = \sum_{m \in \{0,1\}} D_{\mathbb{P},h}(m) |m\rangle \langle m| \quad (16)$$

To prove the computational indistinguishability of σ_0 and σ_1 , each state is split into two terms (for $r \in \{0,1\}$):

$$\sigma_r = \sigma_r^D + \sigma_r^C \quad (17)$$

By a straightforward application of the triangle inequality, we obtain that if σ_0 is computationally distinguishable from σ_1 , either σ_0^D and σ_1^D are computationally distinguishable or σ_0^C and σ_1^C are. Note that even if the terms are not quantum states, the notion of computational indistinguishability (Definition 3.4) is still well defined: to show that two terms, for example σ_0^C and σ_1^C , are computationally indistinguishable, we need to show (informally) that there does not exist an efficiently computable CPTP map \mathcal{S} such that the following expression is non negligible

$$|\text{Tr}(|0\rangle \langle 0| \otimes \mathcal{I}) \mathcal{S}(\sigma_0^C - \sigma_1^C)| \quad (18)$$

In more detail, the density matrices σ_0 and σ_1 are created by beginning with the state $|\phi_y\rangle$ in (14) and applying the operations of both the prover and verifier in the Hadamard round, followed by tracing out all but the first qubit. Therefore, to split σ_0 and σ_1 into two parts, we can equivalently split the density matrix of $|\phi_y\rangle$ into the following two parts, corresponding to the diagonal and cross terms:

$$\sum_{b \in \{0,1\}} |b\rangle \langle b| \otimes |x_{b,y}\rangle \langle x_{b,y}| \otimes |\psi_{b,x_{b,y}}\rangle \langle \psi_{b,x_{b,y}}| \quad (19)$$

$$\sum_{b \in \{0,1\}} |b\rangle \langle b \oplus 1| \otimes |x_{b,y}\rangle \langle x_{b \oplus 1,y}| \otimes |\psi_{b,x_{b,y}}\rangle \langle \psi_{b,x_{b \oplus 1,y}}| \quad (20)$$

Let σ_0^D and σ_1^D be the result of applying the operations of both the prover and the verifier in the Hadamard round to (19), followed by tracing out all but the first qubit. Recall the difference between σ_0^D and σ_1^D : in the latter, the prover's attack U is conjugated by $(Z \otimes \mathcal{I})$. Define σ_0^C and σ_1^C similarly, but replace (19) with (20). In the following two sections, we show that both pairs of terms are computationally indistinguishable.

2.4.1.1 Diagonal Terms In this section, we will show that if there exists a BQP attacker \mathcal{A}' who can distinguish between the terms σ_0^D and σ_1^D , then there exists a BQP attacker \mathcal{A} who can violate the following informal hardcore bit property of the function family \mathcal{F} (the formal statement is part of Definition 4.1):

Claim 2.1 *Assume $f_{k,0}$ and $f_{k,1}$ are sampled from a trapdoor claw-free family \mathcal{F} . Then there does not exist an attacker who, on input k , can produce $b \in \{0,1\}$, $x_b \in \mathcal{X}$, $d \in \{0,1\}^w \setminus \{0^w\}$ and $c \in \{0,1\}$ such that $c = d \cdot (x_0 \oplus x_1)$ where $f_{k,0}(x_0) = f_{k,1}(x_1)$.*

We first describe the state σ_0^D , which is created by beginning with the state in (19), in more detail. Note that the state in (19) can be efficiently created by following the prover's commitment process and then measuring the committed qubit and preimage register. To create σ_0^D , the attack U is applied to the state in (19), followed by Hadamard measurement of the committed qubit and preimage register and application of the verifier's X decoding operator. Finally, all qubits but the first are traced out. σ_1^D is almost the same as σ_0^D , except the attack U is replaced with the attack $(Z \otimes \mathcal{I})U(Z \otimes \mathcal{I})$. Note that the initial phase operator has no effect, since it acts on the diagonal state in (19). The final phase flip, once it is shifted past the Hadamard transform, is equivalent to flipping the decoding bit of the verifier; it follows that $\sigma_1^D = X\sigma_0^D X$.

We now construct the BQP attacker \mathcal{A} who will use \mathcal{A}' to violate Claim 2.1. Let σ_D be the state σ_0^D except for the verifier's decoding. Observe from the description in the previous paragraph that this state can be efficiently created, and as part of creating the state, the measurements $b, x_{b,y}$ and d are obtained. The attacker \mathcal{A} creates the state σ_D . For the string d obtained by \mathcal{A} , the decoding bit $d \cdot (x_{0,y} \oplus x_{1,y})$ determines which of the two states σ_0^D and σ_1^D \mathcal{A} has created; if $d \cdot (x_{0,y} \oplus x_{1,y}) = r$, \mathcal{A} has created σ_r^D . Now \mathcal{A} can run \mathcal{A}' on the resulting mixed state in order to learn $d \cdot (x_{0,y} \oplus x_{1,y})$. As a result, \mathcal{A} holds the following information: $b, x_{b,y}, d$, and $d \cdot (x_0 \oplus x_1)$, therefore violating Claim 2.1.

2.4.1.2 Cross Terms In this section, we will show that the cross terms σ_0^C and σ_1^C are computationally indistinguishable. Since the cross terms are not quantum states, we first show below that if there exists a CPTP map \mathcal{S} which distinguishes between σ_0^C and σ_1^C , i.e. if the following expression is non negligible:

$$|\text{Tr}(|0\rangle \langle 0| \otimes \mathcal{I})\mathcal{S}(\sigma_0^C - \sigma_1^C)| \quad (21)$$

then there exists an efficiently computable CPTP map \mathcal{S}' such that the CPTP map $\mathcal{S}\mathcal{S}'$ distinguishes between the quantum states $\hat{\sigma}_0$ and $\hat{\sigma}_1$, defined as follows. The density matrix $\hat{\sigma}_r$ corresponds to the following pure state (recall $|\phi_y\rangle$ from (14)):

$$(Z^r \otimes \mathcal{I})|\phi_y\rangle = (Z^r \otimes \mathcal{I})\left(\sum_{b \in \{0,1\}} |b\rangle |x_{b,y}\rangle |\psi_{b,x_{b,y}}\rangle\right) \quad (22)$$

To do this, it suffices to show that $\sigma_0^C - \sigma_1^C = \mathcal{S}'(\hat{\sigma}_0 - \hat{\sigma}_1)$. This equality is straightforward for two reasons. First, $\frac{1}{2}(\hat{\sigma}_0 - \hat{\sigma}_1)$ is equal to the cross term in (20). Second, both σ_0^C and σ_1^C also begin with (20), but followed by a CPTP map which is inefficient due to the verifier's decoding. To prove the existence of \mathcal{S}' , we show that taking the difference between σ_0^C and σ_1^C effectively removes the verifier's decoding, creating an efficient CPTP map \mathcal{S}' .

Finally, we will show that an attacker who can distinguish between $\hat{\sigma}_0$ and $\hat{\sigma}_1$ can violate the following informal hardcore bit property of the function family \mathcal{F} :

Claim 2.2 *Assume $f_{k,0}$ and $f_{k,1}$ are sampled from a trapdoor claw-free family \mathcal{F} . Then there exists $d \in \{0,1\}^w$ such that there does not exist an attacker who, on input k , can produce $c = d \cdot (x_0 \oplus x_1) \in \{0,1\}$ (where $f_{k,0}(x_0) = f_{k,1}(x_1)$).*

We begin by describing the cross term σ_0^C (which is not a quantum state) in more detail. σ_0^C is created by beginning with the expression in (20), copied here for reference:

$$\sum_{b \in \{0,1\}} |b\rangle \langle b \oplus 1| \otimes |x_{b,y}\rangle \langle x_{b \oplus 1,y}| \otimes |\psi_{b,x_{b,y}}\rangle \langle \psi_{b,x_{b \oplus 1,y}}| \quad (23)$$

then applying the attack U , followed by Hadamard measurement of the committed qubit and preimage register and application of the verifier's X decoding operator. Finally, all qubits but the first are traced out. σ_1^C is almost the same, except the attack U is replaced with the attack $(Z \otimes \mathcal{I})U(Z \otimes \mathcal{I})$. As in Section 2.4.1.1, the phase flip acting after U is equivalent to flipping the decoding operator of the verifier (i.e. applying an X operator to the matrix σ_0^C). The initial phase flip, which acts on the first qubit of (23), results in a phase of -1. Combining these two observations yields the following equality:

$$\sigma_1^C = -X\sigma_0^C X \quad (24)$$

Taking the difference between σ_0^C and σ_1^C results in a matrix which has a uniform X operator applied:

$$\sigma_0^C - \sigma_1^C = \sum_{r \in \{0,1\}} X^r \sigma_0^C X^r \quad (25)$$

Observe that the CPTP map applied to (23) to create σ_0^C is efficiently computable except for the verifier's X decoding operator. In (25), there is a uniform X operator acting on σ_0^C , effectively replacing the verifier's decoding operator. Let \mathcal{S}' be the resulting efficiently computable CPTP map. It follows immediately that $\sigma_0^C - \sigma_1^C = \mathcal{S}'(\hat{\sigma}_0 - \hat{\sigma}_1)$.

We now proceed to showing that an attacker \mathcal{A}' who can distinguish between $\hat{\sigma}_0$ and $\hat{\sigma}_1$ can be used to violate Claim 2.2. Since the state $\hat{\sigma}_r$ is the state $|\phi_y\rangle$ from (14) with the operator Z^r applied to the committed qubit, an attacker who can distinguish between $\hat{\sigma}_0$ and $\hat{\sigma}_1$ can distinguish whether or not a Z operator is applied to the committed qubit of $|\phi_y\rangle$. The following equality (which holds up to a global phase) shows that a Z operator on the preimage register is equivalent to a Z operator on the committed qubit:

$$(\mathcal{I} \otimes Z^d \otimes \mathcal{I}) \left(\sum_{b \in \{0,1\}} |b\rangle |x_{b,y}\rangle |\psi_{b,x_{b,y}}\rangle \right) = (Z^{d \cdot (x_{0,y} \oplus x_{1,y})} \otimes \mathcal{I}) \left(\sum_{b \in \{0,1\}} |b\rangle |x_{b,y}\rangle |\psi_{b,x_{b,y}}\rangle \right) \quad (26)$$

This equality, along with the attacker \mathcal{A}' , can be used to construct a BQP attacker \mathcal{A} who can determine $d \cdot (x_{0,y} \oplus x_{1,y})$ for an arbitrary fixed string d . \mathcal{A} first constructs $|\phi_y\rangle$ (this is simply the prover's state after reporting the commitment string y , so it can be constructed efficiently). Next, \mathcal{A} applies Z^d to the preimage register of $|\phi_y\rangle$. Due to the equality in (26), this is equivalent to instead applying $Z^{d \cdot (x_{0,y} \oplus x_{1,y})}$ to the committed qubit. By running the attacker \mathcal{A}' , \mathcal{A} can determine $d \cdot (x_{0,y} \oplus x_{1,y})$, therefore violating Claim 2.2.

2.5 Extension to Verification Protocol for BQP

Our goal is to verify that an instance $x \in L$ for a language $L \in \text{BQP}$. Recall that each instance can be converted into a local Hamiltonian H with the following property: if $x \in L$, H has ground state energy at most a and if

$x \notin L$, H has ground state energy at least b , where the gap $b - a$ is inverse polynomial. Therefore, to verify that an instance $x \in L$, a verifier with a quantum computer can simply ask the prover for the ground state and estimate the energy of the received state with respect to the Hamiltonian H . The soundness of such a protocol rests on the fact that if an instance $x \notin L$, all possible states sent by the prover will have energy $\geq b$.

To use such a verification procedure in our setting, we need to rely on one more fact: the Hamiltonian H can be written as a sum over terms which are each a product of X and Z operators [BL08]. Therefore, when the verifier is estimating the energy of a state sent by the prover, he only needs to perform Hadamard or standard basis measurements on each individual qubit. In [MF16], the authors formalize the resulting protocol and use it to build a protocol in which a verifier with access to a single qubit register can verify the result of a BQP computation. Their protocol achieves a completeness/ soundness gap which is negligibly close to 1 by performing polynomially many repetitions of the energy estimation process described above.

In [MF16], the prover sends single qubits to the verifier, who performs either Hadamard or standard basis measurements. To obtain a verification protocol for BQP, we simply replace this step of their protocol with our measurement protocol. Completeness and soundness follow, since our measurement protocol allows the verifier to collect standard and Hadamard basis measurements of a given state, and our soundness claim guarantees that the distribution over measurement results obtained by the verifier comes from the measurement of an underlying quantum state. The extension of our measurement protocol to a verification protocol is described in Section 8.

2.6 Paper Outline

As noted in Section 2.1.1, the trapdoor function families we use in the rest of the paper are not the ideal families used so far in the overview. We instead use approximations of these function families, which are defined in Section 4. The protocol as described used several properties of the ideal families. We take care to define our approximate families to make sure that they satisfy these required properties, at the expense of additional notation. At the start of Section 4, there is a summary of the differences between the approximate functions and the ideal functions (taken from [BCM⁺18]).

We begin with preliminaries in Section 3 and we present the definition of our extended trapdoor claw-free family in Section 4. Section 5 covers Sections 2.1 to 2.3 of the overview. In Section 6, we present the argument outlined in Section 2.4, in which we replace a general attack with an attack which commutes with standard basis measurement. In Section 7, we prove the soundness of the measurement protocol. Finally, the extension of the measurement protocol to a QPIP₀ (described in Section 2.5) is given in Section 8, providing the main result of our paper (stated formally as Theorem 8.6 in Section 8):

Theorem 2.3 *Assuming the existence of an extended trapdoor claw-free family as defined in Definition 4.4, BQP = QPIP₀.*

In Section A, we provide a learning with errors based construction of an extended trapdoor family, providing a proof of the following theorem (stated formally as Theorem A.6 in Section A):

Theorem 2.4 *Under the assumption that the learning with errors problem with superpolynomial noise ratio is computationally intractable for an efficient quantum machine, there exists an extended trapdoor claw-free family.*

3 Preliminaries

Parts of this section are taken from [ABOE08]/[ABOEM17] and [BCM⁺18].

3.1 Notation

For all $q \in \mathbb{N}$ we let \mathbb{Z}_q denote the ring of integers modulo q . We represent elements in \mathbb{Z}_q using numbers in the range $(-\frac{q}{2}, \frac{q}{2}] \cap \mathbb{Z}$. We denote by $[x]_q$ the unique integer y s.t. $y = x \pmod{q}$ and $y \in (-\frac{q}{2}, \frac{q}{2}]$. For $x \in \mathbb{Z}_q$ we define $|x| = |[x]_q|$. When considering an $s \in \{0, 1\}^n$ we sometimes also think of s as an element of \mathbb{Z}_q^n , in which case we write it as s .

We use the terminology of polynomially bounded, super-polynomial, and negligible functions. A function $n : \mathbb{N} \rightarrow \mathbb{R}_+$ is *polynomially bounded* if there exists a polynomial p such that $n(\lambda) \leq p(\lambda)$ for all $\lambda \in \mathbb{N}$. A function $n : \mathbb{N} \rightarrow \mathbb{R}_+$ is *negligible* (resp. *super-polynomial*) if for every polynomial p , $p(\lambda)n(\lambda) \rightarrow_{\lambda \rightarrow \infty} 0$ (resp. $n(\lambda)/p(\lambda) \rightarrow_{\lambda \rightarrow \infty} \infty$).

We generally use the letter D to denote a distribution over a finite domain X , and f for a density on X , i.e. a function $f : X \rightarrow [0, 1]$ such that $\sum_{x \in X} f(x) = 1$. We often use the distribution and its density interchangeably. We write U for the uniform distribution. We write $x \leftarrow D$ to indicate that x is sampled from distribution D , and $x \leftarrow_U X$ to indicate that x is sampled uniformly from the set X . We write \mathcal{D}_X for the set of all densities on X . For any $f \in \mathcal{D}_X$, $\text{SUPP}(f)$ denotes the support of f ,

$$\text{SUPP}(f) = \{x \in X \mid f(x) > 0\}.$$

For two densities f_1 and f_2 over the same finite domain X , the Hellinger distance between f_1 and f_2 is

$$H^2(f_1, f_2) = 1 - \sum_{x \in X} \sqrt{f_1(x)f_2(x)}. \quad (27)$$

and the total variation distance between f_1 and f_2 is:

$$\|f_1 - f_2\|_{TV} = \frac{1}{2} \sum_{x \in X} |f_1(x) - f_2(x)|. \quad (28)$$

The following lemma will be useful:

Lemma 3.1 *Let D_0, D_1 be distributions over a finite domain X . Let $X' \subseteq X$. Then:*

$$\left| \Pr_{x \leftarrow D_0} [x \in X'] - \Pr_{x \leftarrow D_1} [x \in X'] \right| \leq \|D_0 - D_1\|_{TV} \quad (29)$$

We require the following definition:

Definition 3.2 Computational Indistinguishability of Distributions *Two families of distributions $\{D_{0,\lambda}\}_{\lambda \in \mathbb{N}}$ and $\{D_{1,\lambda}\}_{\lambda \in \mathbb{N}}$ (indexed by the security parameter λ) are computationally indistinguishable if for all quantum polynomial-time attackers \mathcal{A} there exists a negligible function $\mu(\cdot)$ such that for all $\lambda \in \mathbb{N}$*

$$\left| \Pr_{x \leftarrow D_{0,\lambda}} [\mathcal{A}(x) = 0] - \Pr_{x \leftarrow D_{1,\lambda}} [\mathcal{A}(x) = 0] \right| \leq \mu(\lambda). \quad (30)$$

3.2 Quantum Computation Preliminaries

3.2.1 Quantum Operations

We will use the X, Y and Z Pauli operators: $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ and $Y = iXZ$. The n -qubits Pauli group consists of all elements of the form $P = P_1 \otimes P_2 \otimes \dots \otimes P_n$ where $P_i \in \{I, X, Y, Z\}$, 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 $P = P_1 \otimes P_2 \otimes \dots \otimes P_n$ but not the multiplicative factors. We will use the fact that Pauli operators anti commute: $ZX = -XZ$. 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 .

We will use completely positive trace preserving (CPTP) maps to represent general quantum operations. A CPTP map \mathcal{S} can be represented by its Kraus operators, $\{B_\tau\}_\tau$. The result of applying \mathcal{S} to a state ρ is:

$$\mathcal{S}(\rho) = \sum_{\tau} B_\tau \rho B_\tau^\dagger \quad (31)$$

We say that two CPTP maps \mathcal{S} and \mathcal{S}' are equal if, for all density matrices ρ , $\mathcal{S}(\rho) = \mathcal{S}'(\rho)$.

3.2.2 Trace Distance

For density matrices ρ, σ , the trace distance $\|\rho - \sigma\|_{tr}$ is equal to:

$$\|\rho - \sigma\|_{tr} = \frac{1}{2} \text{Tr}(\sqrt{(\rho - \sigma)^2}) \quad (32)$$

We will use the following fact ([Wik18]):

$$\|\rho - \sigma\|_{tr} = \max_P \text{Tr}(P(\rho - \sigma)) \quad (33)$$

where the maximization is carried over all projectors P . We will also use the fact that the trace distance is contractive under completely positive trace preserving maps ([Wik18]). The following lemma relates the Hellinger distance (as given in (27)) and the trace distance of superpositions:

Lemma 3.3 *Let X be a finite set and $f_1, f_2 \in \mathcal{D}_x$. Let*

$$|\psi_1\rangle = \sum_{x \in X} \sqrt{f_1(x)} |x\rangle \quad \text{and} \quad |\psi_2\rangle = \sum_{x \in X} \sqrt{f_2(x)} |x\rangle .$$

Then

$$\| |\psi_1\rangle \langle \psi_1| - |\psi_2\rangle \langle \psi_2| \|_{tr} = \sqrt{1 - (1 - H^2(f_1, f_2))^2} .$$

We require the following definition, which is analogous to Definition 3.2:

Definition 3.4 Computational Indistinguishability of Quantum States *Two families of density matrices $\{\rho_{0,\lambda}\}_{\lambda \in \mathbb{N}}$ and $\{\rho_{1,\lambda}\}_{\lambda \in \mathbb{N}}$ (indexed by the security parameter λ) are computationally indistinguishable if for all efficiently computable CPTP maps \mathcal{S} there exists a negligible function $\mu(\cdot)$ such that for all $\lambda \in \mathbb{N}$:*

$$\left| \text{Tr}(|0\rangle \langle 0| \otimes \mathcal{I}) \mathcal{S}(\rho_{0,\lambda} - \rho_{1,\lambda}) \right| \leq \mu(\lambda) . \quad (34)$$

3.2.3 Pauli Twirl

We call the conjugation of a unitary operator (or a CPTP map) by a random Pauli a Pauli twirl. The twirled version of a unitary U is the CPTP map $\{(X^x Z^z)^\dagger U (X^x Z^z)\}_{x,z}$. If the Pauli is a random X (or Z) Pauli operator, we call the conjugation an X (or Z) Pauli twirl. A Z Pauli twirl has the following effect:

Lemma 3.5 Z Pauli Twirl For a CPTP map with Kraus operators $\{B_\tau\}_\tau$, the following two CPTP maps are equal:

$$\left\{ \frac{1}{\sqrt{2}} (Z^r \otimes \mathcal{I}) B_\tau (Z^r \otimes \mathcal{I}) \right\}_{r \in \{0,1\}, \tau} = \{(X^x \otimes \mathcal{I}) B'_{x,\tau}\}_{x \in \{0,1\}, \tau} \quad (35)$$

where $B_\tau = \sum_{x,z \in \{0,1\}} X^x Z^z \otimes B_{xz\tau}$ and the CPTP map $\{B'_{x,\tau}\}_{x \in \{0,1\}, \tau}$ is equal to $B'_{x,\tau} = \sum_{z \in \{0,1\}} Z^z \otimes B_{xz\tau}$.

The following corollary follows from Lemma 3.5 and captures the effect of a Z Pauli twirl followed by Hadamard basis measurement (the proof of both the lemma and the corollary are provided in Section B):

Corollary 3.6 Z Pauli Twirl with Measurement For a CPTP map with Kraus operators $\{B_\tau\}_\tau$, the following two CPTP maps are equal:

$$\left\{ \frac{1}{\sqrt{2}} (|b\rangle \langle b| H Z^r \otimes \mathcal{I}) B_\tau (Z^r \otimes \mathcal{I}) \right\}_{b,r \in \{0,1\}, \tau} = \{(|b\rangle \langle b| H \otimes \mathcal{I}) B'_{x,\tau}\}_{b,x \in \{0,1\}, \tau} \quad (36)$$

where $B_\tau = \sum_{x,z \in \{0,1\}} X^x Z^z \otimes B_{xz\tau}$ and the CPTP map $\{B'_{x,\tau}\}_{x \in \{0,1\}, \tau}$ is equal to $B'_{x,\tau} = \sum_{z \in \{0,1\}} Z^z \otimes B_{xz\tau}$.

3.3 QPIP Background

A QPIP is defined as follows (this definition is taken from [ABOE08]/ [ABOEM17]):

Definition 3.7 A language \mathcal{L} is said to have a Quantum Prover Interactive Proof (QPPI $_\tau$) with completeness c and soundness s (where $c - s$ is at least 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 τ 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 τ qubits, on which the verifier can perform arbitrary quantum operations and which has access to a quantum channel which can transmit τ qubits. At any given time, the verifier is not allowed to possess more than τ 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 $_\tau$.

4 Function Definitions

4.1 Noisy Trapdoor Claw-Free Functions

This section is taken directly from [BCM⁺18]. Let λ be a security parameter, and \mathcal{X} and \mathcal{Y} be finite sets (depending on λ). For our purposes an ideal family of functions \mathcal{F} would have the following properties. For each public key k , there are two functions $\{f_{k,b} : \mathcal{X} \rightarrow \mathcal{Y}\}_{b \in \{0,1\}}$ that are both injective and have the same range (equivalently, $(b, x) \mapsto f_{k,b}(x)$ is 2-to-1), and are invertible given a suitable trapdoor t_k (i.e. t_k can be used to compute x given b and $y = f_{k,b}(x)$). Furthermore, the pair of functions should be claw-free: it must be hard for an attacker to find two pre-images $x_0, x_1 \in \mathcal{X}$ such that $f_{k,0}(x_0) = f_{k,1}(x_1)$. Finally, the functions should satisfy an adaptive hardcore bit property, which is a stronger form of the claw-free property: assuming for convenience that $\mathcal{X} = \{0,1\}^w$, we would like that it is computationally infeasible to simultaneously generate an $(b, x_b) \in \{0,1\} \times \mathcal{X}$ and a $d \in \{0,1\}^w \setminus \{0^w\}$ such that with non-negligible advantage over $\frac{1}{2}$ the equation $d \cdot (x_0 \oplus x_1) = 0$, where x_{1-b} is defined as the unique element such that $f_{k,1-b}(x_{1-b}) = f_{k,b}(x_b)$, holds.

Unfortunately, we do not know how to construct a function family that exactly satisfies all these requirements under standard cryptographic assumptions. Instead, we construct a family that satisfies slightly relaxed requirements, that we will show still suffices for our purposes, based on the hardness of the learning with errors (LWE) problem. The requirements are relaxed as follows. First, the range of the functions is no longer a set \mathcal{Y} ; instead, it is $\mathcal{D}_{\mathcal{Y}}$, the set of probability densities over \mathcal{Y} . That is, each function returns a density, rather than a point. The trapdoor injective pair property is then described in terms of the support of the output densities: these supports should either be identical, for a colliding pair, or be disjoint, in all other cases.

The consideration of functions that return densities gives rise to an additional requirement of efficiency: there should exist a quantum polynomial-time procedure that efficiently prepares a superposition over the range of the function, i.e. for any key k and $b \in \{0,1\}$, the procedure can prepare the state

$$\frac{1}{\sqrt{|\mathcal{X}|}} \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \sqrt{f_{k,b}(x)(y)} |x\rangle |y\rangle . \quad (37)$$

In our instantiation based on LWE, it is not possible to prepare (37) perfectly, but it is possible to create a superposition with coefficients $\sqrt{f'_{k,b}(x)}$ such that the resulting state is within negligible trace distance of (37). The density $f'_{k,b}(x)$ is required to satisfy two properties used in our protocol. First, it must be easy to check, without the trapdoor, if an element $y \in \mathcal{Y}$ lies in the support of $f'_{k,b}(x)$. Second, the inversion algorithm should operate correctly on all y in the support of $f'_{k,b}(x)$.

We slightly modify the adaptive hardcore bit requirement as well. Since the set \mathcal{X} may not be a subset of binary strings, we first assume the existence of an injective, efficiently invertible map $J : \mathcal{X} \rightarrow \{0,1\}^w$. Next, we only require the adaptive hardcore bit property to hold for a subset of all nonzero strings, instead of the set $\{0,1\}^w \setminus \{0^w\}$. Finally, membership in the appropriate set should be efficiently checkable, given access to the trapdoor.

A formal definition follows.

Definition 4.1 (NTCF Family) *Let λ be a security parameter. Let \mathcal{X} and \mathcal{Y} be finite sets. Let $\mathcal{K}_{\mathcal{F}}$ be a finite set of keys. A family of functions*

$$\mathcal{F} = \{f_{k,b} : \mathcal{X} \rightarrow \mathcal{D}_{\mathcal{Y}}\}_{k \in \mathcal{K}_{\mathcal{F}}, b \in \{0,1\}}$$

*is called a **noisy trapdoor claw-free (NTCF) family** if the following conditions hold:*

1. **Efficient Function Generation.** *There exists an efficient probabilistic algorithm $\text{GEN}_{\mathcal{F}}$ which generates a key $k \in \mathcal{K}_{\mathcal{F}}$ together with a trapdoor t_k :*

$$(k, t_k) \leftarrow \text{GEN}_{\mathcal{F}}(1^\lambda) .$$

2. **Trapdoor Injective Pair.** For all keys $k \in \mathcal{K}_{\mathcal{F}}$ the following conditions hold.

- (a) *Trapdoor:* For all $b \in \{0, 1\}$ and $x \neq x' \in \mathcal{X}$, $\text{SUPP}(f_{k,b}(x)) \cap \text{SUPP}(f_{k,b}(x')) = \emptyset$. Moreover, there exists an efficient deterministic algorithm $\text{INV}_{\mathcal{F}}$ such that for all $b \in \{0, 1\}$, $x \in \mathcal{X}$ and $y \in \text{SUPP}(f_{k,b}(x))$, $\text{INV}_{\mathcal{F}}(t_k, b, y) = x$.
- (b) *Injective pair:* There exists a perfect matching $\mathcal{R}_k \subseteq \mathcal{X} \times \mathcal{X}$ such that $f_{k,0}(x_0) = f_{k,1}(x_1)$ if and only if $(x_0, x_1) \in \mathcal{R}_k$.

3. **Efficient Range Superposition.** For all keys $k \in \mathcal{K}_{\mathcal{F}}$ and $b \in \{0, 1\}$ there exists a function $f'_{k,b} : \mathcal{X} \mapsto \mathcal{D}_{\mathcal{Y}}$ such that

- (a) For all $(x_0, x_1) \in \mathcal{R}_k$ and $y \in \text{SUPP}(f'_{k,b}(x_b))$, $\text{INV}_{\mathcal{F}}(t_k, b, y) = x_b$ and $\text{INV}_{\mathcal{F}}(t_k, b \oplus 1, y) = x_{b \oplus 1}$.
- (b) There exists an efficient deterministic procedure $\text{CHK}_{\mathcal{F}}$ that, on input k , $b \in \{0, 1\}$, $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, returns 1 if $y \in \text{SUPP}(f'_{k,b}(x))$ and 0 otherwise. Note that $\text{CHK}_{\mathcal{F}}$ is not provided the trapdoor t_k .
- (c) For every k and $b \in \{0, 1\}$,

$$\mathbb{E}_{x \leftarrow \mathcal{U}\mathcal{X}} [H^2(f_{k,b}(x), f'_{k,b}(x))] \leq \mu(\lambda),$$

for some negligible function $\mu(\cdot)$. Here H^2 is the Hellinger distance; see (27). Moreover, there exists an efficient procedure $\text{SAMP}_{\mathcal{F}}$ that on input k and $b \in \{0, 1\}$ prepares the state

$$\frac{1}{\sqrt{|\mathcal{X}|}} \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \sqrt{(f'_{k,b}(x))(y)} |x\rangle |y\rangle. \quad (38)$$

4. **Adaptive Hardcore Bit.** For all keys $k \in \mathcal{K}_{\mathcal{F}}$ the following conditions hold, for some integer w that is a polynomially bounded function of λ .

- (a) For all $b \in \{0, 1\}$ and $x \in \mathcal{X}$, there exists a set $G_{k,b,x} \subseteq \{0, 1\}^w$ such that $\Pr_{d \leftarrow \mathcal{U}\{0,1\}^w} [d \notin G_{k,b,x}]$ is negligible, and moreover there exists an efficient algorithm that checks for membership in $G_{k,b,x}$ given k, b, x and the trapdoor t_k .
- (b) There is an efficiently computable injection $J : \mathcal{X} \rightarrow \{0, 1\}^w$, such that J can be inverted efficiently on its range, and such that the following holds. If

$$\begin{aligned} H_k &= \{(b, x_b, d, d \cdot (J(x_0) \oplus J(x_1))) \mid b \in \{0, 1\}, (x_0, x_1) \in \mathcal{R}_k, d \in G_{k,0,x_0} \cap G_{k,1,x_1}\},^1 \\ \overline{H}_k &= \{(b, x_b, d, c) \mid (b, x, d, c \oplus 1) \in H_k\}, \end{aligned}$$

then for any quantum polynomial-time procedure \mathcal{A} there exists a negligible function $\mu(\cdot)$ such that

$$\left| \Pr_{(k,t_k) \leftarrow \text{GEN}_{\mathcal{F}}(1^\lambda)} [\mathcal{A}(k) \in H_k] - \Pr_{(k,t_k) \leftarrow \text{GEN}_{\mathcal{F}}(1^\lambda)} [\mathcal{A}(k) \in \overline{H}_k] \right| \leq \mu(\lambda). \quad (39)$$

4.2 Extended Trapdoor Claw-Free Functions

In this section, we define the extended trapdoor claw-free family we will use in this paper, which is a NTCF family (Definition 4.1) with two additional properties. In order to define an extended trapdoor claw-free family, we must first define a trapdoor injective family. A trapdoor injective family differs from a NTCF family in two ways: the function pairs have disjoint images (rather than perfectly overlapping images) and there is no adaptive hardcore bit condition.

¹Note that although both x_0 and x_1 are referred to to define the set H_k , only one of them, x_b , is explicitly specified in any 4-tuple that lies in H_k .

Definition 4.2 (Trapdoor Injective Function Family) Let λ be a security parameter. Let \mathcal{X} and \mathcal{Y} be finite sets. Let $\mathcal{K}_{\mathcal{G}}$ be a finite set of keys. A family of functions

$$\mathcal{G} = \{g_{k,b} : \mathcal{X} \rightarrow \mathcal{D}_{\mathcal{Y}}\}_{b \in \{0,1\}, k \in \mathcal{K}_{\mathcal{G}}}$$

is called a **trapdoor injective family** if the following conditions hold:

1. **Efficient Function Generation.** There exists an efficient probabilistic algorithm $GEN_{\mathcal{G}}$ which generates a key $k \in \mathcal{K}_{\mathcal{G}}$ together with a trapdoor t_k :

$$(k, t_k) \leftarrow GEN_{\mathcal{G}}(1^\lambda).$$

2. **Disjoint Trapdoor Injective Pair.** For all keys $k \in \mathcal{K}_{\mathcal{G}}$, for all $b, b' \in \{0, 1\}$ and $x, x' \in \mathcal{X}$, if $(b, x) \neq (b', x')$, $SUPP(g_{k,b}(x)) \cap SUPP(g_{k,b'}(x')) = \emptyset$. Moreover, there exists an efficient deterministic algorithm $INV_{\mathcal{F}}$ such that for all $b \in \{0, 1\}$, $x \in \mathcal{X}$ and $y \in SUPP(g_{k,b}(x))$, $INV_{\mathcal{G}}(t_k, y) = (b, x)$.

3. **Efficient Range Superposition.** For all keys $k \in \mathcal{K}_{\mathcal{G}}$ and $b \in \{0, 1\}$

- (a) There exists an efficient deterministic procedure $CHK_{\mathcal{G}}$ that, on input $k, b \in \{0, 1\}$, $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, outputs 1 if $y \in SUPP(g_{k,b}(x))$ and 0 otherwise. Note that $CHK_{\mathcal{G}}$ is not provided the trapdoor t_k .
- (b) There exists an efficient procedure $SAMP_{\mathcal{G}}$ that on input k and $b \in \{0, 1\}$ returns the state

$$\frac{1}{\sqrt{|\mathcal{X}|}} \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \sqrt{(g_{k,b}(x))(y)} |x\rangle |y\rangle. \quad (40)$$

Definition 4.3 (Injective Invariance) A noisy trapdoor claw-free family \mathcal{F} is **injective invariant** if there exists a trapdoor injective family \mathcal{G} such that:

1. The algorithms $CHK_{\mathcal{F}}$ and $SAMP_{\mathcal{F}}$ are the same as the algorithms $CHK_{\mathcal{G}}$ and $SAMP_{\mathcal{G}}$.
2. For all quantum polynomial-time procedures \mathcal{A} , there exists a negligible function $\mu(\cdot)$ such that

$$\left| \Pr_{(k,t_k) \leftarrow GEN_{\mathcal{F}}(1^\lambda)} [\mathcal{A}(k) = 0] - \Pr_{(k,t_k) \leftarrow GEN_{\mathcal{G}}(1^\lambda)} [\mathcal{A}(k) = 0] \right| \leq \mu(\lambda) \quad (41)$$

Definition 4.4 (Extended Trapdoor Claw-Free Family) A noisy trapdoor claw-free family \mathcal{F} is an **extended trapdoor claw-free family** if:

1. It is injective invariant.
2. For all $k \in \mathcal{K}_{\mathcal{F}}$ and $d \in \{0, 1\}^w$, let:

$$H'_{k,d} = \{d \cdot (J(x_0) \oplus J(x_1)) | (x_0, x_1) \in \mathcal{R}_k\} \quad (42)$$

For all quantum polynomial-time procedures \mathcal{A} , there exists a negligible function $\mu(\cdot)$ and a string $d \in \{0, 1\}^w$ such that

$$\left| \Pr_{(k,t_k) \leftarrow GEN_{\mathcal{F}}(1^\lambda)} [\mathcal{A}(k) \in H'_{k,d}] - \frac{1}{2} \right| \leq \mu(\lambda) \quad (43)$$

5 Measurement Protocol

We begin by introducing the state commitment process (described in Section 2.1) followed by the measurement protocol (given in Section 2.2). The presentation below is slightly more involved than the overview, since we are not using the perfect trapdoor claw-free/ injective families used in the overview; we are instead using the families given in Definitions 4.1, 4.2 and 4.4. After presenting the measurement protocol, we provide notation which will be used throughout the rest of the paper. Next, we prove completeness of our measurement protocol and characterize the behavior of general provers (as described in Section 2.3.1). This section ends with the construction of the quantum state underlying the measurement distribution obtained by the verifier (given in Section 2.3.2).

5.1 How to Commit Using a Noisy Trapdoor Claw-Free Family

Here we describe the process of state commitment with a NTCF function (as defined in Definition 4.1). The state commitment process requires a function key $k \in \mathcal{K}_{\mathcal{F}}$ (which corresponds to functions $f_{k,0}, f_{k,1} \in \mathcal{F}$) and is performed with respect to the first qubit of an arbitrary state $|\psi\rangle$:

$$|\psi\rangle = \sum_{b \in \{0,1\}} \alpha_b |b\rangle |\psi_b\rangle \quad (44)$$

The first step of the commitment process is to apply the $\text{SAMP}_{\mathcal{F}}$ procedure in superposition, with k and the first qubit containing b as input:

$$\frac{1}{\sqrt{|\mathcal{X}|}} \sum_{\substack{b \in \{0,1\} \\ x \in \mathcal{X}, y \in \mathcal{Y}}} \alpha_b \sqrt{f'_{k,b}(x)(y)} |b\rangle |x\rangle |\psi_b\rangle |y\rangle \quad (45)$$

By condition 3(c) of Definition 4.1 and Lemma 3.3 (45) is within negligible trace distance of the following state:

$$\frac{1}{\sqrt{|\mathcal{X}|}} \sum_{\substack{b \in \{0,1\} \\ x \in \mathcal{X}, y \in \mathcal{Y}}} \alpha_b \sqrt{f_{k,b}(x)(y)} |b\rangle |x\rangle |\psi_b\rangle |y\rangle \quad (46)$$

The second step of the commitment process is to measure the last register, obtaining the *commitment string* $y \in \mathcal{Y}$. Let $x_{b,y} = \text{INV}_{\mathcal{F}}(t_k, b, y)$ (t_k is the trapdoor corresponding to the key k). The remaining state at this point is within negligible trace distance of the following state

$$\sum_{b \in \{0,1\}} \alpha_b |b\rangle |x_{b,y}\rangle |\psi_b\rangle \quad (47)$$

The fact that the superposition collapses in this manner is due to both the trapdoor and injective pair conditions in Definition 4.1. The trapdoor condition implies that for each b , there can be at most one remaining element $x \in \mathcal{X}$ in the superposition after measuring y . The injective pair condition states that for all $x_{b,y} \in \mathcal{X}$, there exists exactly one $x_{b \oplus 1, y} \in \mathcal{X}$ such that $(x_{0,y}, x_{1,y}) \in \mathcal{R}_k$ (i.e. $f_{k,b}(x_b) = f_{k,b}(x_{b \oplus 1})$). Therefore, if $y \in \text{SUPP}(f_{k,b}(x_b))$, it follows that $y \in \text{SUPP}(f_{k,b \oplus 1}(x_{b \oplus 1}))$. We will call the first qubit of (47) the *committed qubit* and the second register (containing $x_{b,y}$) the *preimage register*.

5.1.1 Hadamard Measurement of a Committed State

The first step in measuring a committed state in the Hadamard basis is to apply the unitary U_J , which uses the injective map J defined in condition 4(b) of Definition 4.1:

$$U_J \left(\sum_{b \in \{0,1\}} \alpha_b |b\rangle |x_{b,y}\rangle |\psi_b\rangle |0\rangle^{e_J} \right) = \sum_{b \in \{0,1\}} \alpha_b |b\rangle |J(x_{b,y})\rangle |\psi_b\rangle |0\rangle^{e'_J} \quad (48)$$

where the number of auxiliary qubits (e_J and e'_J) is determined by the map J . The map U_J is unitary since J is both efficiently computable and efficiently invertible. The second step is to apply the Hadamard transform $H^{\otimes w+1}$ to the first two registers of the state in (48). The resulting state is:

$$\frac{1}{\sqrt{2^w}} \sum_{\substack{d \in \{0,1\}^w \\ b \in \{0,1\}}} \alpha_b X^{d \cdot (J(x_{0,y}) \oplus J(x_{1,y}))} H |b\rangle \otimes (-1)^{d \cdot J(x_{0,y})} |d\rangle \otimes |\psi_b\rangle \otimes |0\rangle^{e'_J} \quad (49)$$

The third step is measurement of the preimage register, obtaining a string $d \in \{0,1\}^w$ and resulting in the following state (recall the state $|\psi\rangle$ from (44)):

$$(X^{d \cdot (J(x_{0,y}) \oplus J(x_{1,y}))} H \otimes \mathcal{I}) |\psi\rangle |0\rangle^{e'_J} \quad (50)$$

The final step is measuring the committed qubit to obtain a bit b' . The Hadamard measurement result of the first qubit of $|\psi\rangle$ is $b' \oplus d \cdot (J(x_{0,y}) \oplus J(x_{1,y})) \in \{0,1\}$ ($x_{0,y}$ and $x_{1,y}$ can be recovered from y using the trapdoor t_k and the function $\text{INV}_{\mathcal{F}}$).

5.1.2 How to Commit Using a Trapdoor Injective Family

The commitment process described in Section 5.1 can also be performed using a key $k \in \mathcal{K}_{\mathcal{G}}$ corresponding to trapdoor injective functions $g_{k,0}, g_{k,1} \in \mathcal{G}$ (see Definition 4.2). Prior to measuring y (at the stage of (45)), the state is:

$$\frac{1}{\sqrt{|\mathcal{X}|}} \sum_{\substack{b \in \{0,1\} \\ x \in \mathcal{X}, y \in \mathcal{Y}}} \alpha_b \sqrt{g_{k,b}(x)(y)} |b\rangle |x\rangle |\psi_b\rangle |y\rangle \quad (51)$$

Now the last register is measured to obtain $y \in \mathcal{Y}$. Since the sets $\text{SUPP}(g_{k,b}(x))$ and $\text{SUPP}(g_{k,b'}(x'))$ are disjoint for all $(b, x) \neq (b', x')$ (see the trapdoor condition of Definition 4.2), $y \in \bigcup_{x \in \mathcal{X}} \text{SUPP}(g_{k,b}(x))$ with probability $|\alpha_b|^2$.

Let $(b, x_{b,y}) = \text{INV}_{\mathcal{G}}(t_k, y)$. The remaining state after measurement is:

$$|b\rangle |x_{b,y}\rangle |\psi_b\rangle \quad (52)$$

Therefore, measuring y acts as a standard basis measurement of the first qubit of the state $|\psi\rangle$ in (44). The standard basis measurement b can be obtained (with access to only the trapdoor t_k of $g_{k,0}, g_{k,1}$ and y) by running the function $\text{INV}_{\mathcal{G}}$.

5.2 Measurement Protocol

We now use the commitment process in Section 5.1 to construct our measurement protocol for n qubits, where n is polynomial in the security parameter λ . We require an extended trapdoor claw-free family \mathcal{F} (Definition 4.4) as well as its corresponding trapdoor injective family \mathcal{G} (Definition 4.2). The measurement protocol depends on a string $h \in \{0,1\}^n$, called the basis choice, which represents the basis for which the verifier would like measurement results of the n qubits; $h_i = 0$ indicates the standard basis and $h_i = 1$ indicates the Hadamard basis. We now provide the measurement protocol:

Protocol 5.1 Measurement Protocol (for $h \in \{0,1\}^n$)

1. The verifier performs the following operations for $1 \leq i \leq n$:

- (a) If the standard basis is chosen for qubit i (i.e. $h_i = 0$), the verifier runs $\text{GEN}_{\mathcal{G}}(1^\lambda)$ to produce a function key $k_i \in \mathcal{K}_{\mathcal{G}}$ and its corresponding trapdoor t_{k_i} .

- (b) If the Hadamard basis is chosen for qubit i ($h_i = 1$), the verifier runs $\text{GEN}_{\mathcal{F}}(1^\lambda)$ to produce a function key $k_i \in \mathcal{K}_{\mathcal{F}}$ and its corresponding trapdoor t_{k_i} .

Let $k' = (k_1, \dots, k_n)$. The verifier sends the function choice k' to the prover.

2. The verifier receives $y' = (y_1, \dots, y_n) \in \mathcal{Y}^n$ from the prover.
3. The verifier chooses at random to run a test round or a Hadamard round (each is chosen with probability $\frac{1}{2}$).
4. For a test round:
 - (a) The verifier asks the prover for standard basis measurements of committed qubit i and preimage register i , for $1 \leq i \leq n$.
 - (b) For $1 \leq i \leq n$, the verifier receives a bit b'_i and a string $x'_i \in \mathcal{X}$. For all i such that $h_i = 0$, the verifier rejects if $\text{CHK}_{\mathcal{G}}(k_i, b'_i, x'_i, y_i) = 0$. For all i such that $h_i = 1$, the verifier rejects if $\text{CHK}_{\mathcal{F}}(k_i, b'_i, x'_i, y_i) = 0$.
5. For a Hadamard round:
 - (a) The verifier asks the prover for Hadamard measurements of committed qubit i and preimage register i for $1 \leq i \leq n$.
 - (b) For $1 \leq i \leq n$, the verifier receives a bit b'_i and a string $d_i \in \{0, 1\}^w$.
 - (c) For qubits i for which $h_i = 0$, the results (b'_i, d_i) are ignored. The verifier computes

$$(m_i, x_{m_i, y_i}) = \text{INV}_{\mathcal{G}}(t_{k_i}, y_i) \quad (53)$$

If the inverse does not exist, the verifier stores a random bit as the measurement result and rejects. Otherwise the verifier stores m_i as the standard basis measurement result.

- (d) For qubits i for which $h_i = 1$, the verifier computes

$$x_{0, y_i} = \text{INV}_{\mathcal{F}}(t_{k_i}, 0, y_i) \quad (54)$$

$$x_{1, y_i} = \text{INV}_{\mathcal{F}}(t_{k_i}, 1, y_i) \quad (55)$$

If either of the inverses does not exist, the verifier stores a random bit as the measurement result and rejects. The verifier uses t_{k_i} to check if $d_i \in G_{k_i, 0, x_{0, y_i}} \cap G_{k_i, 1, x_{1, y_i}}$. If not, the verifier stores a random bit as the measurement result and rejects. Otherwise, the verifier stores $m_i = b'_i \oplus d_i \cdot (J(x_{0, y_i}) \oplus J(x_{1, y_i}))$ as the Hadamard basis measurement result.

5.2.1 Honest Prover

We now provide an honest prover's behavior in Protocol 5.1, assuming the prover would like to report measurement results of an n qubit state ρ :

Protocol 5.2 Honest Prover in Measurement Protocol (for an efficiently computable n qubit state ρ)

1. The prover creates the state ρ . Upon receipt of k' from the verifier, the prover commits to qubit i of ρ using k_i as described in Section 5.1. The prover reports the measurement results $y' = (y_1, \dots, y_n) \in \mathcal{Y}^n$ obtained from each commitment process to the verifier.
2. For a test round:

(a) The prover measures each of the n committed qubits and preimage registers in the standard basis, sending the verifier the resulting bit b'_i and string $x'_i \in \mathcal{X}$ for $1 \leq i \leq n$.

3. For a Hadamard round:

(a) The prover first applies the unitary U_J to all n preimage registers. The prover then measures each of the n committed qubits and preimage registers in the Hadamard basis, sending the verifier the resulting bit b'_i and string $d_i \in \{0, 1\}^w$ for $1 \leq i \leq n$.

5.3 Notation

We now introduce some notation (and provide reminders of previously used notation) which will be useful throughout the rest of the paper.

1. The string $h \in \{0, 1\}^n$ is called the *basis choice*; $h_i = 0$ indicates the standard basis and $h_i = 1$ indicates the Hadamard basis.
2. We will call k' (produced in step 1 of Protocol 5.1) the *function choice* of the verifier. Let $D_{V,h}$ be the distribution which it is sampled from (this is the distribution produced by $\text{GEN}_{\mathcal{F}}$ and $\text{GEN}_{\mathcal{G}}$).
3. A *perfect* prover is a prover who is always accepted by the verifier on the test round.
4. For a density matrix ρ on n qubits and a string $h \in \{0, 1\}^n$, let $D_{\rho,h}$ be the distribution over $\{0, 1\}^n$ which results from measuring all qubits of ρ in the basis specified by h .
5. For every prover \mathbb{P} and basis choice $h \in \{0, 1\}^n$, let $D_{\mathbb{P},h}$ be the distribution over measurement results $m \in \{0, 1\}^n$ obtained by the verifier when interacting with \mathbb{P} on basis choice h in a Hadamard round. Let $D_{\mathbb{P},h}^C$ be the same distribution, but conditioned on acceptance by the verifier (in a Hadamard round). Let $\sigma_{\mathbb{P},h}$ be the density matrix corresponding to the distribution $D_{\mathbb{P},h}$:

$$\sigma_{\mathbb{P},h} \stackrel{\text{def}}{=} \sum_{m \in \{0,1\}^n} D_{\mathbb{P},h}(m) |m\rangle \langle m| \quad (56)$$

We will frequently use the fact that for provers \mathbb{P} and \mathbb{P}' , $\sigma_{\mathbb{P},h}$ and $\sigma_{\mathbb{P}',h}$ are computationally indistinguishable if and only if $D_{\mathbb{P},h}$ and $D_{\mathbb{P}',h}$ are computationally indistinguishable, by definition of computational indistinguishability of distributions (Definition 3.2) and of density matrices (Definition 3.4). Also note that by definition of trace distance in (32) and total variation distance in (28):

$$\|\sigma_{\mathbb{P},h} - \sigma_{\mathbb{P}',h}\|_{tr} = \|D_{\mathbb{P},h} - D_{\mathbb{P}',h}\|_{TV} \quad (57)$$

6. As introduced in Section 5.1, a *committed qubit* is the qubit which is used to determine whether to apply $f_{k,0}$ or $f_{k,1}$ (or $g_{k,0}$ or $g_{k,1}$) and the *preimage register* is the register which contains the inverse after the measurement; i.e. in the following state from (47):

$$\sum_b \alpha_b |b\rangle |x_{b,y}\rangle |\psi_b\rangle \quad (58)$$

the first qubit is the committed qubit and the second register (containing $x_{b,y}$) is the preimage register. The *commitment string* is the string $y \in \mathcal{Y}$.

5.4 Completeness of Protocol 5.1

Claim 5.3 Completeness of Protocol 5.1 For all n qubit states ρ and for all basis choices $h \in \{0, 1\}^n$, the prover \mathbb{P} described in Protocol 5.2 is a perfect prover (\mathbb{P} is accepted by the verifier in a test round for basis choice h with perfect probability). There exists a negligible function μ such that in the Hadamard round for basis choice h , the verifier accepts \mathbb{P} with probability $\geq 1 - \mu$ and $\|D_{\mathbb{P},h}^C - D_{\rho,h}\|_{TV} \leq \mu$.

Proof of Claim 5.3: First, assume that the prover could produce the ideal states in the commitment procedure, as written in (46) for the Hadamard basis and (51) for the standard basis. Call such a prover \mathbb{P}' . The distribution over measurement results obtained by the verifier when interacting with \mathbb{P}' (prior to conditioning on acceptance) is equal to the distribution over measurement results obtained by measuring ρ in the basis specified by h , i.e.:

$$\|D_{\mathbb{P}',h} - D_{\rho,h}\|_{TV} = 0 \quad (59)$$

We now return to analyzing the prover \mathbb{P} given in Protocol 5.2. First note that \mathbb{P} is a perfect prover: when measured, the superpositions created by \mathbb{P} during the commitment process (in (45) and (51)) pass the CHK procedure perfectly, by definition (see Definition 4.1 and Definition 4.2). Moving on to the Hadamard round, \mathbb{P} is rejected by the verifier only if there exists an i such that the measurement result d_i is not in the set $G_{k_i,0,x_0,y_i} \cap G_{k_i,1,x_1,y_i}$ (and $h_i = 1$). The adaptive hardcore bit clause of Definition 4.1 (item 4(a)) implies that since d_i is sampled uniformly (it is the result of a Hadamard transform), there exists a negligible function μ_H such that the probability that the verifier rejects \mathbb{P} in the Hadamard round is at most μ_H :

$$\|D_{\mathbb{P},h} - D_{\mathbb{P}',h}^C\|_{TV} \leq \mu_H \quad (60)$$

Next, observe that the prover \mathbb{P} in Protocol 5.2 can produce the state in (51) (which is used by the prover \mathbb{P}'), but can only create a state within negligible trace distance of the state in (46). It follows that there exists a negligible function μ' such that:

$$\|D_{\mathbb{P},h} - D_{\mathbb{P}',h}\|_{TV} \leq \mu' \quad (61)$$

Using the triangle inequality, we obtain:

$$\|D_{\mathbb{P},h}^C - D_{\rho,h}\|_{TV} \leq \|D_{\mathbb{P},h}^C - D_{\mathbb{P},h}\|_{TV} + \|D_{\mathbb{P},h} - D_{\mathbb{P}',h}\|_{TV} + \|D_{\mathbb{P}',h} - D_{\rho,h}\|_{TV} \quad (62)$$

We complete the calculation by plugging in (59), (60) and (61):

$$\|D_{\mathbb{P},h}^C - D_{\rho,h}\|_{TV} \leq \mu_H + \mu' \quad (63)$$

To complete the proof of the claim, set $\mu = \mu_H + \mu'$. \square

5.5 Prover Behavior

We now give a claim which characterizes the behavior of a general prover in Protocol 5.1 (the overview of this claim and its proof are given in Section 2.3.1). The only difference between the following claim and the version given in the overview is the inclusion of the operator U_J (as defined in (48)):

Claim 5.4 Prover Behavior For all BQP provers \mathbb{P} in Protocol 5.1, there exist two efficiently computable unitary operators U_0, U and a prover \mathbb{P}' (described below) such that for all basis choices $h \in \{0, 1\}^n$, \mathbb{P} and \mathbb{P}' are accepted by the verifier with the same probability in a test round and the distribution over measurement results $D_{\mathbb{P},h}$ produced by the prover and verifier as a result of Protocol 5.1 is equal to the distribution $D_{\mathbb{P}',h}$ corresponding to the prover \mathbb{P}' . We say that \mathbb{P} is characterized by (U_0, U) .

1. \mathbb{P}' designates his first n qubits as committed qubits, the next n registers as preimage registers and the final n registers as commitment string registers. All other registers contain auxiliary space.
2. Upon receipt of the function choice k' from the verifier, the prover \mathbb{P}' applies U_0 to his initial state:

$$|0\rangle^{\otimes e} \otimes |k'\rangle \quad (64)$$

where e is determined by U_0, U and U_0 uses the last register (containing k') as a control register; i.e. there exists a unitary $U_{0,k'}$ such that

$$U_0(|0\rangle^{\otimes e} \otimes |k'\rangle) = U_{0,k'}(|0\rangle^{\otimes e}) \otimes |k'\rangle \quad (65)$$

3. \mathbb{P}' measures the commitment string registers to obtain $y' = (y_1, \dots, y_n) \in \mathcal{Y}^n$, which is sent to the verifier.
4. For a Hadamard round:
 - (a) \mathbb{P}' appends $e_J \cdot n$ auxiliary 0 qubits to his state and applies the unitary U_J to all n preimage registers, followed by application of the unitary U to his entire state.
 - (b) \mathbb{P}' measures the n committed qubits and preimage registers in the Hadamard basis. \mathbb{P}' sends the verifier the resulting bit b'_i and string $d_i \in \{0, 1\}^w$ for $1 \leq i \leq n$.
5. For a test round, \mathbb{P}' measures each of the n committed qubits and preimage registers in the standard basis, sending the verifier the resulting bit b'_i and string $x'_i \in \mathcal{X}$ for $1 \leq i \leq n$.

Notation 5.5 We will also frequently say that a prover \mathbb{P} is characterized by two CPTP maps $(\mathcal{S}_0, \mathcal{S})$. This means that for all basis choices $h \in \{0, 1\}^n$, \mathbb{P} and \mathbb{P}' are accepted with the same probability in a test round and $D_{\mathbb{P},h} = D_{\mathbb{P}',h}$, and the prover \mathbb{P}' follows steps 1 - 5 in Claim 5.4, but uses the CPTP maps $\mathcal{S}_0, \mathcal{S}$ rather than the unitary operators U_0, U .

Proof of Claim 5.4: We will follow the principle given in Section 2.3.1: a general prover is equivalent from the verifier's perspective to a prover \mathbb{P} who begins each round by applying an arbitrary unitary attack and then behaves honestly. The first implication of the principle is that \mathbb{P} measures the same registers as an honest prover; therefore, like the honest prover, \mathbb{P} designates the first n qubits as committed qubits, the next n registers as preimage registers, and the final n registers as commitment string registers. All other registers of \mathbb{P} contain the auxiliary space.

The second implication is that there exist unitary operators U', U_T and U_C such that \mathbb{P} acts as follows. \mathbb{P} begins with the initial state $|0\rangle^{\otimes e} \otimes |k'\rangle$ and then applies a unitary operator U' to his state, followed by standard basis measurement of the commitment string registers to obtain y' . If the verifier chooses a test round, the prover applies another unitary U_T followed by standard basis measurements of the committed qubit and preimage registers to obtain the requested measurement results. If the verifier chooses a Hadamard round, the prover first appends $e_J \cdot n$ auxiliary 0 qubits to his state. Next, the prover applies a unitary U_C to his state. He then applies the unitary U_J (see Section 5.1.1) to all n preimage registers. Finally, the prover measures all n committed qubits and preimage registers in the Hadamard basis to obtain the requested measurement results. We can assume both U_T and U_C do not act on the register containing y' . This is because y' could have been copied into the prover's auxiliary space prior to measurement, and U_T and U_C can instead act on this space. It follows that both U_T and U_C commute with the measurement of y' .

To obtain the attacks U_0 and U which characterize \mathbb{P} , we make two changes. First, we use the fact that U_T commutes with measurement of y' to shift it prior to the measurement. Due to this change, we also need to append

U_T^\dagger to the start of the Hadamard round attack. Our second change is to shift the unitary U_J so that it is prior to the Hadamard round attack; this can be done by conjugating the attack by U_J . It follows that if we let $U_0 = U_T U'$, $U = U_J^{\otimes n} U_C U_T^\dagger (U_J^{\otimes n})^\dagger$ and consider the prover \mathbb{P}' described in the statement of Claim 5.4 (with respect to U_0 and U), \mathbb{P} and \mathbb{P}' are accepted with the same probability in a test round and $D_{\mathbb{P},h} = D_{\mathbb{P}',h}$ for all basis choices h . \square

5.6 Construction of Underlying Quantum State

We require the following definition:

Definition 5.6 Trivial Prover A perfect prover \mathbb{P} in Protocol 5.1 characterized by (U_0, \mathcal{S}) (where U_0 is a unitary, \mathcal{S} is a CPTP map and both are efficiently computable) is called trivial if \mathcal{S} commutes with standard basis measurement on the first n qubits.

In this section, we prove the following claim (the overview of this claim is given in Section 2.3.2):

Claim 5.7 For all trivial provers \mathbb{P} , there exists an n qubit state ρ (which can be created using a BQP circuit) such that for all $h \in \{0, 1\}^n$, the distribution over measurement results $D_{\mathbb{P},h}$ produced in Protocol 5.1 with respect to \mathbb{P} for basis choice h is computationally indistinguishable from the distribution $D_{\rho,h}$ which results from measuring ρ in the basis determined by h .

Proof: For a unitary U_0 and CPTP map \mathcal{S} , let the prover \mathbb{P} be characterized by (U_0, \mathcal{S}) . The state ρ is constructed as follows:

Protocol 5.8 Construction of ρ corresponding to \mathbb{P}

1. For $1 \leq i \leq n$: sample $(k_i, t_{k_i}) \leftarrow \text{GEN}_{\mathcal{F}}(1^\lambda)$.
2. Follow steps 1-4(a) in Claim 5.4 (with respect to U_0, \mathcal{S}).
3. Measure all preimage registers in the Hadamard basis to obtain $d_1, \dots, d_n \in \{0, 1\}^w$.
4. For $1 \leq i \leq n$, use the trapdoor t_{k_i} to apply $Z^{d_i \cdot (x_{0,y_i} \oplus x_{1,y_i})}$ to the i^{th} committed qubit.
5. Trace out all qubits except n committed qubits.

We now argue that, for all $h \in \{0, 1\}^n$, $D_{\rho,h}$ is computationally indistinguishable from $D_{\mathbb{P},h}$. We will proceed through two families of hybrid states which are dependent on the basis choice h . In the first family $\{\rho_h^{(1)}\}_{h \in \{0,1\}^n}$, we simply remove the Z decoding operator (step 4 of Protocol 5.8) if $h_i = 0$. This also eliminates the need for the trapdoor t_{k_i} if $h_i = 0$:

Protocol 5.9 Construction of $\rho_h^{(1)}$ corresponding to \mathbb{P}

1. For $1 \leq i \leq n$: sample $(k_i, t_{k_i}) \leftarrow \text{GEN}_{\mathcal{F}}(1^\lambda)$. If $h_i = 0$, discard the trapdoor t_{k_i} .
2. Apply steps 2-3 of Protocol 5.8.
3. For $1 \leq i \leq n$, if $h_i = 1$, use the trapdoor t_{k_i} to apply $Z^{d_i \cdot (x_{0,y_i} \oplus x_{1,y_i})}$ to the i^{th} committed qubit.
4. Trace out all qubits except the n committed qubits.

The distributions $D_{\rho,h}$ and $D_{\rho_h^{(1)},h}$ differ only on i for which $h_i = 0$. To address this difference, note that if $h_i = 0$, the Z operator applied in step 4 of Protocol 5.8 has no effect on $D_{\rho,h}$: to obtain $D_{\rho,h}$ the i^{th} committed qubit is measured in the standard basis immediately after application of the Z operator. Therefore, $D_{\rho,h} = D_{\rho_h^{(1)},h}$ for all h .

Our next hybrid is:

Protocol 5.10 *Construction of $\rho_h^{(2)}$ corresponding to \mathbb{P}*

1. For $1 \leq i \leq n$: if $h_i = 1$, sample $(k_i, t_{k_i}) \leftarrow \text{GEN}_{\mathcal{F}}(1^\lambda)$. If $h_i = 0$, sample $(k_i, t_{k_i}) \leftarrow \text{GEN}_{\mathcal{G}}(1^\lambda)$ and discard the trapdoor t_{k_i} .
2. Apply steps 2-4 of Protocol 5.9.

The computational indistinguishability of $D_{\rho_h^{(1)},h}$ and $D_{\rho_h^{(2)},h}$ follows due to the injective invariance (Definition 4.3) of \mathcal{F} with respect to \mathcal{G} : as long as the trapdoor t_{k_i} is unknown, a key k_i sampled from $\mathcal{K}_{\mathcal{F}}$ is computationally indistinguishable from a key k_i sampled from $\mathcal{K}_{\mathcal{G}}$. We can apply this argument for all i such that $h_i = 0$ since the trapdoor t_{k_i} was discarded for all such i in Protocols 5.9 and 5.10.

We have so far shown that $D_{\rho,h}$ is computationally indistinguishable from $D_{\rho_h^{(2)},h}$ for all $h \in \{0,1\}^n$. To complete our proof, we now show that $D_{\rho_h^{(2)},h} = D_{\mathbb{P},h}$. The two distributions differ as follows: if $h_i = 0$, the distribution of the i^{th} bit of $D_{\mathbb{P},h}$ is obtained from the commitment string y_i (see step 5(c) of Protocol 5.1), but the distribution of the i^{th} bit of $D_{\rho_h^{(2)},h}$ is obtained from measuring the i^{th} committed qubit of $\rho_h^{(2)}$ in the standard basis.

To see that these two distributions are equal, we begin by observing that since the prover \mathbb{P} is perfect, if $h_i = 0$, measuring the i^{th} committed qubit prior to the attack \mathcal{S} (i.e. at the start of the Hadamard round) results in the same outcome as extracting the measurement outcome from y_i . To complete our proof, recall that since the prover is trivial, the attack \mathcal{S} commutes with standard basis measurement. \square

6 Replacement of General Attack with X-Trivial Attack for Hadamard Basis

In this section, we analyze Protocol 5.1 with a perfect prover (a prover who passes the test round of Protocol 5.1 with perfect probability). We will rely on notation introduced in Section 5.3. This section is dedicated to proving the following claim:

Claim 6.1 *General to X-Trivial Attack for Hadamard Basis* Let $1 \leq j \leq n$. Let $\mathcal{S} = \{B_\tau\}_\tau$ and $\mathcal{S}_j = \{B'_{j,x,\tau}\}_{x \in \{0,1\}, \tau}$ be CPTP maps written in terms of their Kraus operators:

$$B_\tau = \sum_{x,z \in \{0,1\}} X^x Z^z \otimes B_{jxz\tau} \quad (66)$$

$$B'_{j,x,\tau} = \sum_{z \in \{0,1\}} Z^z \otimes B_{jxz\tau} \quad (67)$$

where B_τ and $B'_{j,x,\tau}$ have been rearranged so that $X^x Z^z$ and Z^z act on the j^{th} qubit. For a unitary operator U_0 , let \mathbb{P} be a perfect prover characterized by (U_0, \mathcal{S}) (see Claim 5.4 and notation 5.5). Let \mathbb{P}_j be a perfect prover characterized by (U_0, \mathcal{S}_j) . If $h_j = 1$, $D_{\mathbb{P},h}$ and $D_{\mathbb{P}_j,h}$ are computationally indistinguishable.

The overview of the proof of Claim 6.1 is given in Section 2.4. Claim 6.1 is slightly more general than the statement in the overview: n does not have to be equal to 1, and we are proving that we can replace the attack \mathcal{S} with an attack which acts trivially on any one of the committed qubits j for which $h_j = 1$. We begin by writing out the state $\sigma_{\mathbb{P},h}$ which corresponds to the distribution $D_{\mathbb{P},h}$ (as defined in (56)). This requires some care, since we need to go through the steps of Protocol 5.1 in order to construct $\sigma_{\mathbb{P},h}$. Once we write down the state $\sigma_{\mathbb{P},h}$, we can proceed to proving computational indistinguishability between $\sigma_{\mathbb{P},h}$ and $\sigma_{\mathbb{P}_{j,h}}$. As written below (56), proving computational indistinguishability between $\sigma_{\mathbb{P},h}$ and $\sigma_{\mathbb{P}_{1,h}}$ is equivalent to proving indistinguishability between $D_{\mathbb{P},h}$ and $D_{\mathbb{P}_{1,h}}$.

Proof of Claim 6.1: We will assume for convenience that $j = 1$; the proof for all other values of j is equivalent. To analyze the state $\sigma_{\mathbb{P},h}$, we will assume that \mathbb{P} follows steps 1-4 in Claim 5.4. We can do this since \mathbb{P} is characterized by U_0, \mathcal{S} ; therefore, the state $\sigma_{\mathbb{P},h}$ can be obtained by following the steps in Claim 5.4.

We first provide some notation we will require. Let $k = k_1 \in \mathcal{K}_{\mathcal{F}}$ be the first function key received by the prover \mathbb{P} in Protocol 5.1. Throughout this proof, we will only be focusing on the first committed qubit (since $j = 1$). Therefore, for notational convenience, we will drop the subscript of 1 for values pertaining to the first committed qubit (i.e. the basis choice, function key, commitment string, etc.). Let $h_{>1} = (h_2, \dots, h_n)$, let $k_{>1} = (k_2, \dots, k_n)$ and define $t_{k>1}$ similarly. We will also require the following mixed state, which contains the distribution over all function keys and trapdoors except the first (as sampled by the verifier).

$$\sum_{k_{>1}} D_{\mathbb{V},h_{>1}} |k_{>1}\rangle \langle k_{>1}| \otimes |t_{k>1}\rangle \langle t_{k>1}| \quad (68)$$

This mixed state is required to create $\sigma_{\mathbb{P},h}$: the function keys are part of the prover's input and the trapdoors are used for the verifier's decoding. For convenience, let $|\phi_{k>1}\rangle$ be a purification of the above state; when analyzing the state $\sigma_{\mathbb{P},h}$, we can consider a purification since we will eventually be tracing out all but the committed qubits. For $b \in \{0, 1\}$, let $T_{k,b} = \bigcup_{x \in \mathcal{X}} \text{SUPP}(f_{k,b}(x))$ ($\text{SUPP}(f_{k,b}(x))$ is the support of the probability density function $f_{k,b}(x)$ - see Definition 4.1 for a reminder). Let $T_k = T_{k,0} \cup T_{k,1}$.

We begin by writing the state of \mathbb{P} after application of U_0 . Recall from Claim 5.4 that when the verifier requests test round measurement results from \mathbb{P} , \mathbb{P} simply measures the requested registers in the standard basis and sends the results to the verifier. Since \mathbb{P} is a perfect prover, it follows that the state of \mathbb{P} after applying U_0 must yield measurement results which pass the test round perfectly. The state of \mathbb{P} after applying U_0 can therefore be written as:

$$U_0 |0\rangle^{\otimes e} |k\rangle |\phi_{k>1}\rangle = \sum_{\substack{b \in \{0,1\} \\ y \in T_{k,b}}} |b, x_{b,y}\rangle |\psi_{b,y,k}\rangle |y\rangle \quad (69)$$

where $x_{b,y}$ is the output of $\text{INV}_{\mathcal{F}}(t_k, b, y)$. We have suppressed the dependence of $x_{b,y}$ on k for convenience. The state in (69) can be written in this format since if the prover returns $y \in \mathcal{Y}$ in the commitment stage (step 2 of Protocol 5.1), in the test round he must return $b \in \{0, 1\}$ and $x \in \mathcal{X}$ such that $\text{CHK}_{\mathcal{F}}(t_k, b, x, y) = 1$. Conditions 3(a) and 3(b) of Definition 4.1 imply that only $x_{b,y} = \text{INV}_{\mathcal{F}}(t_k, b, y)$ satisfies this condition. The auxiliary space represented by $|\psi_{b,y,k}\rangle$ includes the remaining $n - 1$ committed qubits, preimage registers and commitment strings as well as the state $|\phi_{k>1}\rangle$.

For convenience, we will instead write the state in (69) as follows:

$$U_0 |0\rangle^{\otimes e} |k\rangle = \sum_{\substack{b \in \{0,1\} \\ y \in T_k}} |b, x_{b,y}\rangle |\psi_{b,y,k}\rangle |y\rangle \quad (70)$$

The only change we have made is we have replaced the summation over $y \in T_{k,b}$ with the summation over $y \in T_k = T_{k,0} \cup T_{k,1}$. Note that the existence of two inverses of y ($x_{0,y}$ and $x_{1,y}$) is guaranteed since $y \in T_k$ - see Definition 4.1. For b, y for which $y \notin T_{k,b}$, let $|\psi_{b,y,k}\rangle = 0$.

After the prover measures y and sends it to the verifier, the state shared between the prover and verifier is:

$$\sum_{y \in T_k} \left(\sum_{b \in \{0,1\}} |b, x_{b,y}\rangle |\psi_{b,y,k}\rangle \right) \left(\sum_{b \in \{0,1\}} |b, x_{b,y}\rangle |\psi_{b,y,k}\rangle \right)^\dagger \otimes |y\rangle \langle y| \quad (71)$$

and the last register (containing y) is held by the verifier. Next, the prover applies the injective map U_J (see Section 5.1.1) to all n preimage registers along with auxiliary 0 qubits, which we assume have already been included in the extra space $|\psi_{b,y,k}\rangle$. At this point, the state shared between the prover and verifier is:

$$\rho_k = \sum_{y \in T_k} \rho_{y,k} \quad (72)$$

where

$$\rho_{y,k} = \left(\sum_{b \in \{0,1\}} |b, J(x_{b,y})\rangle |\psi'_{b,y,k}\rangle \right) \left(\sum_{b \in \{0,1\}} |b, J(x_{b,y})\rangle |\psi'_{b,y,k}\rangle \right)^\dagger \otimes |y\rangle \langle y| \quad (73)$$

The auxiliary space $|\psi_{b,y,k}\rangle$ has changed to $|\psi'_{b,y,k}\rangle$ to account for the fact that the commitment strings corresponding to indices $i > 1$ were measured and the unitary U_J was applied to the corresponding preimage registers in the auxiliary space (we are considering a purification of the auxiliary space for convenience).

The prover then applies his CPTP map $\mathcal{S} = \{B_\tau\}_\tau$ followed by Hadamard basis measurement of the first committed qubit and preimage register of the state in (72). The state shared between the prover and verifier at this point is:

$$\sum_{\substack{b' \in \{0,1\}, \tau \\ d \in \{0,1\}^w}} (|b'\rangle \langle b'| \otimes |d\rangle \langle d| \otimes \mathcal{I})(H^{\otimes l+1} \otimes \mathcal{I}) B_\tau \rho_k B_\tau^\dagger (H^{\otimes l+1} \otimes \mathcal{I})^\dagger (|b'\rangle \langle b'| \otimes |d\rangle \langle d| \otimes \mathcal{I})^\dagger \quad (74)$$

Next, if the measurement result $d \in G_{k,0,x_{0,y}} \cap G_{k,1,x_{1,y}}$, the verifier decodes the first qubit by applying the operator $X^{d \cdot (J(x_{0,y}) \oplus J(x_{1,y}))}$ (see step 5(d) of Protocol 5.1). If not, the verifier stores a random bit as his measurement result; we can equivalently assume the verifier decodes the first qubit by applying a random X operator. Note that the verifier's decoding (the application of the X operator) commutes with the prover's measurement of the first qubit. Therefore, the entire state, including the verifier's decoding, can be written as:

$$\sigma_{0,k} = \sum_{\substack{b', c \in \{0,1\}, \tau \\ d \in \{0,1\}^w, y \in R_{c,d,k}}} \delta_{d,y} (|b'\rangle \langle b'| X^c \otimes |d\rangle \langle d| \otimes \mathcal{I})(H^{\otimes l+1} \otimes \mathcal{I}) B_\tau \rho_{y,k} B_\tau^\dagger (H^{\otimes l+1} \otimes \mathcal{I})^\dagger (|b'\rangle \langle b'| X^c \otimes |d\rangle \langle d| \otimes \mathcal{I})^\dagger \quad (75)$$

where $\delta_{d,y} = \frac{1}{2}$ if $d \notin G_{k,0,x_{0,y}} \cap G_{k,1,x_{1,y}}$ and 1 if $d \in G_{k,0,x_{0,y}} \cap G_{k,1,x_{1,y}}$ and

$$R_{c,d,k} = \{y \in T_k | (d \in G_{k,0,x_{0,y}} \cap G_{k,1,x_{1,y}} \wedge d \cdot (J(x_{0,y}) \oplus J(x_{1,y})) = c) \vee (d \notin G_{k,0,x_{0,y}} \cap G_{k,1,x_{1,y}})\} \quad (76)$$

For ease of notation, we will instead write the state in (75) as:

$$\sigma_{0,k} = \sum_{\substack{b', c \in \{0,1\}, \tau \\ d \in \{0,1\}^w, y \in R_{c,d,k}}} \delta_{d,y} O_{b',c,d,\tau} \rho_{y,k} O_{b',c,d,\tau}^\dagger \quad (77)$$

where

$$O_{b',c,d,\tau} = (|b'\rangle\langle b'| X^c \otimes |d\rangle\langle d| \otimes \mathcal{I})(H^{\otimes l+1} \otimes \mathcal{I})B_\tau \quad (78)$$

Let $\mathcal{S}_{>1}$ be the CPTP map which contains all operations done on the remaining $n - 1$ committed qubits and preimage registers after application of the attack \mathcal{S} : $\mathcal{S}_{>1}$ consists of the Hadamard measurement of the remaining $n - 1$ committed qubits and preimage registers as well as the verifier decoding of those committed qubits. $\mathcal{S}_{>1}$ is independent of the function key k and trapdoor t_k ; it is only dependent on the remaining $n - 1$ function keys and trapdoors, which are drawn independently and included in the auxiliary space of $\sigma_{0,k}$. Given this, the state $\sigma_{\mathbb{P},h}$ is obtained by applying $\mathcal{S}_{>1}$, and then tracing out all but the first n qubits (the committed qubits):

$$\sigma_{\mathbb{P},h} = \text{Tr}_{>n}(\mathcal{S}_{>1}(\sum_{k \in \mathcal{K}_{\mathcal{F}}} D_{\mathbb{V},h}(k) \sigma_{0,k})) \quad (79)$$

where $D_{\mathbb{V},h}$ is the distribution over the set of function keys $\mathcal{K}_{\mathcal{F}}$ (since $h = 1$) produced by $\text{GEN}_{\mathcal{F}}$.

To prove the claim, we need to show that $\sigma_{\mathbb{P},h}$ is computationally indistinguishable from $\sigma_{\mathbb{P}_1,h}$. Let

$$\sigma_{\mathbb{P},h,E} = \sum_{k \in \mathcal{K}_{\mathcal{F}}} D_{\mathbb{V},h}(k) \sigma_{0,k} \quad (80)$$

We will instead prove the stronger statement that $\sigma_{\mathbb{P},h,E}$ is computationally indistinguishable from $\sigma_{\mathbb{P}_1,h,E}$ (to see why this is stronger, observe from (79) that $\sigma_{\mathbb{P},h}$ can be obtained from $\sigma_{\mathbb{P},h,E}$ by applying the efficiently computable superoperator $\mathcal{S}_{>1}$ and tracing out all but the first n qubits). Recall from the statement of the claim that \mathbb{P}_1 is characterized by (U_0, \mathcal{S}_1) . Since \mathcal{S}_1 is followed by Hadamard basis measurement, Corollary 3.6 implies that \mathbb{P}_1 is also characterized by $(U_0, \{\frac{1}{\sqrt{2}}(Z^r \otimes \mathcal{I})\mathcal{S}_1(Z^r \otimes \mathcal{I})\}_{r \in \{0,1\}})$. If we let $\hat{\mathbb{P}}_1$ be the prover characterized by $(U_0, (Z \otimes \mathcal{I})\mathcal{S}(Z \otimes \mathcal{I}))$, it follows by linearity that

$$\frac{1}{2}(\sigma_{\mathbb{P},h,E} + \sigma_{\hat{\mathbb{P}}_1,h,E}) = \sigma_{\mathbb{P}_1,h,E} \quad (81)$$

Therefore, to complete the proof of Claim 6.1, we can instead show that $\sigma_{\mathbb{P},h,E}$ is computationally indistinguishable from $\sigma_{\hat{\mathbb{P}}_1,h,E}$, which implies that $\sigma_{\mathbb{P},h,E}$ is computationally indistinguishable from $\sigma_{\mathbb{P}_1,h,E}$.

Computational Indistinguishability For convenience, let $\sigma_{\mathbb{P},h,E} = \sigma_0$ and $\sigma_{\hat{\mathbb{P}}_1,h,E} = \sigma_1$. We now prove that σ_0 and σ_1 are computationally indistinguishable; our proof follows the outline given in Section 2.4.1. As given in (80):

$$\sigma_r = \sum_{k \in \mathcal{K}_{\mathcal{F}}} D_{\mathbb{V},h}(k) \sigma_{r,k} \quad (82)$$

and using (77):

$$\sigma_{r,k} = \sum_{\substack{b',c \in \{0,1\}, \tau \\ d \in \{0,1\}^w, y \in R_{c,d,k}}} \delta_{d,y} O_{b',c \oplus r, d, \tau} (Z^r \otimes \mathcal{I}) \rho_{y,k} (Z^r \otimes \mathcal{I}) O_{b',c \oplus r, d, \tau}^\dagger \quad (83)$$

Note that, in the case that $r = 1$, the operator $(Z \otimes \mathcal{I})$ acting after B_τ was absorbed into $O_{b',c,d,\tau}$ to create $O_{b',c \oplus r, d, \tau}$. Recall from (73) that:

$$\rho_{y,k} = (\sum_{b \in \{0,1\}} |b, J(x_{b,y})\rangle |\psi'_{b,y,k}\rangle) (\sum_{b \in \{0,1\}} |b, J(x_{b,y})\rangle |\psi'_{b,y,k}\rangle)^\dagger \otimes |y\rangle \langle y| \quad (84)$$

We can break down the state $\rho_{y,k}$ into two components:

$$\rho_{y,k} = \rho_{y,k}^D + \rho_{y,k}^C \quad (85)$$

The components are as follows:

$$\rho_{y,k}^D = \sum_{b \in \{0,1\}} |b\rangle \langle b| \otimes |J(x_{b,y})\rangle \langle J(x_{b,y})| \otimes |\psi'_{b,y,k}\rangle \langle \psi'_{b,y,k}| \otimes |y\rangle \langle y| \quad (86)$$

$$\rho_{y,k}^C = \sum_{b \in \{0,1\}} |b\rangle \langle b \oplus 1| \otimes |J(x_{b,y})\rangle \langle J(x_{b \oplus 1,y})| \otimes |\psi'_{b,y,k}\rangle \langle \psi'_{b \oplus 1,y}| \otimes |y\rangle \langle y| \quad (87)$$

Since Z operators acting on the first qubit have no effect on (86) and add a phase of -1 to (87), we can rewrite (83) as:

$$\sigma_{r,k} = \sum_{\substack{b', c \in \{0,1\}, \tau \\ d \in \{0,1\}^w, y \in R_{c,d,k}}} \delta_{d,y} O_{b', c \oplus r, d, \tau} (Z^r \otimes \mathcal{I}) (\rho_{y,k}^D + \rho_{y,k}^C) (Z^r \otimes \mathcal{I}) O_{b', c \oplus r, d, \tau}^\dagger \quad (88)$$

$$= \sum_{\substack{b', c \in \{0,1\}, \tau \\ d \in \{0,1\}^w, y \in R_{c,d,k}}} \delta_{d,y} O_{b', c \oplus r, d, \tau} (\rho_{y,k}^D + (-1)^r \rho_{y,k}^C) O_{b', c \oplus r, d, \tau}^\dagger \quad (89)$$

To show that σ_0 and σ_1 are computationally indistinguishable, we reduce the problem to showing that the components corresponding to the diagonal and cross terms of the committed state $\rho_{y,k}$ are computationally indistinguishable:

Claim 6.2 *If σ_0 is computationally distinguishable from σ_1 , then one of the following must hold:*

1. Let

$$\sigma_r^D \stackrel{\text{def}}{=} \sum_{k \in \mathcal{K}_{\mathcal{F}}} D_{\mathbb{V},h}(k) \sigma_{r,k}^D \quad (90)$$

$$\sigma_{r,k}^D \stackrel{\text{def}}{=} \sum_{\substack{b', c \in \{0,1\}, \tau \\ d \in \{0,1\}^w, y \in R_{c,d,k}}} \delta_{d,y} O_{b', c \oplus r, d, \tau} (\rho_{y,k}^D) O_{b', c \oplus r, d, \tau}^\dagger \quad (91)$$

The density matrices σ_0^D and σ_1^D are computationally distinguishable.

2. Let

$$\hat{\sigma}_r \stackrel{\text{def}}{=} \sum_{k \in \mathcal{K}_{\mathcal{F}}} D_{\mathbb{V},h}(k) \hat{\sigma}_{r,k} \quad (92)$$

$$\hat{\sigma}_{r,k} \stackrel{\text{def}}{=} (Z^r \otimes \mathcal{I}) \left(\sum_{y \in T_k} \rho_{y,k} \right) (Z^r \otimes \mathcal{I}) \quad (93)$$

The density matrices $\hat{\sigma}_0$ and $\hat{\sigma}_1$ are computationally distinguishable.

The first pair of density matrices is equal to the terms of σ_0 and σ_1 resulting from the diagonal term of the committed state and the second pair represents the cross terms. The proof of Claim 6.2 is a simple application of the triangle inequality and is given in Section 6.3. We complete the proof of Claim 6.1 with the following two claims:

Claim 6.3 *If σ_0^D is computationally distinguishable from σ_1^D , then there exists a BQP attacker \mathcal{A} who can violate the hardcore bit clause of Definition 4.1.*

Claim 6.4 *If $\hat{\sigma}_0$ is computationally distinguishable from $\hat{\sigma}_1$, then there exists a BQP attacker \mathcal{A} who can violate the hardcore bit clause of Definition 4.4.*

We prove Claims 6.3 and 6.4 in the next two sections.

□

6.1 Indistinguishability of Diagonal Terms (Proof of Claim 6.3)

The overview of the following proof is given in Section 2.4.1.1.

Proof of Claim 6.3: Recall that we would like to show that the density matrices σ_0^D and σ_1^D are computationally indistinguishable. We will proceed by contradiction. Assume σ_0^D and σ_1^D are distinguishable using an efficiently computable CPTP map \mathcal{S} . It follows by the definition of computational indistinguishability (Definition 3.4) and by the expression for σ_r^D in (90) that the following expression is non negligible:

$$\left| \sum_{k \in \mathcal{K}_{\mathcal{F}}} D_{\mathbb{V},h}(k) \cdot \text{Tr}(|0\rangle\langle 0| \otimes \mathcal{I}) \mathcal{S} \left(\sum_{r \in \{0,1\}} (-1)^r \sigma_{r,k}^D \right) \right| \quad (94)$$

We will use the CPTP map \mathcal{S} to construct an attacker \mathcal{A} who violates the hardcore bit clause of Definition 4.1.

Let

$$R_{c,d,k}^D = \{y \in T_k \mid d \in G_{k,0,x_{0,y}} \cap G_{k,1,x_{1,y}} \wedge d \cdot (J(x_{0,y}) \oplus J(x_{1,y})) = c\} \quad (95)$$

We will require the unnormalized state $\tilde{\sigma}_{r,k}^D$, which is the state $\sigma_{r,k}^D$ from (91) conditioned on obtaining measurements y, d such that $d \in G_{k,0,x_{0,y}} \cap G_{k,1,x_{1,y}}$:

$$\tilde{\sigma}_{r,k}^D = \sum_{\substack{b', c \in \{0,1\}, \tau \\ d \in \{0,1\}^w, y \in R_{c,d,k}^D}} O_{b', c \oplus r, d, \tau} (\rho_{y,k}^D) O_{b', c \oplus r, d, \tau}^\dagger \quad (96)$$

Observe that for all $k \in \mathcal{K}_{\mathcal{F}}$

$$\sum_{r \in \{0,1\}} (-1)^r \sigma_{r,k}^D = \sum_{r \in \{0,1\}} (-1)^r \tilde{\sigma}_{r,k}^D \quad (97)$$

This is because $\sigma_{0,k}^D$ and $\sigma_{1,k}^D$ are identical when conditioned on $d \notin G_{k,0,x_{0,y}} \cap G_{k,1,x_{1,y}}$ (both have a uniform X decoding operator applied). It follows that the expression in (94) is equal to:

$$\left| \sum_{k \in \mathcal{K}_{\mathcal{F}}} D_{\mathbb{V},h}(k) \cdot \text{Tr}(|0\rangle\langle 0| \otimes \mathcal{I}) \mathcal{S} \left(\sum_{r \in \{0,1\}} (-1)^r \tilde{\sigma}_{r,k}^D \right) \right| \quad (98)$$

The attacker \mathcal{A} (on input $k \in \mathcal{K}_{\mathcal{F}}$) first constructs the following state from (72):

$$\sum_{y \in T_k} \rho_{y,k} \quad (99)$$

He does this exactly as the BQP prover \mathbb{P} would have: by applying the prover's initial operator U_0 to the input state and measuring the last register to obtain y . Then \mathcal{A} measures both the committed qubit and preimage register, obtaining $(b, x_{b,y})$. The resulting state is a summation over the state in (86):

$$\sum_{y \in T_k} \rho_{y,k}^D \quad (100)$$

\mathcal{A} now continues as \mathbb{P} would have: he applies the CPTP map $\{B_\tau\}_\tau$, followed by Hadamard measurement of the committed qubit and preimage register, obtaining results $b' \in \{0,1\}$ and $d \in \{0,1\}^w$. \mathcal{A} then chooses a bit c' at random, stores the bit c' in an auxiliary register, and applies $X^{c'}$ to the committed qubit (this operation commutes

with measurement of the committed qubit). The unnormalized state created by \mathcal{A} (conditioned on d, y such that $d \in G_{k,0,x_0,y} \cap G_{k,1,x_1,y}$) is equal to:

$$\frac{1}{2} \sum_{\substack{b',c' \in \{0,1\}, \tau, y \in T_k \\ d \in G_{k,0,x_0,y} \cap G_{k,1,x_1,y}}} O_{b',c',d,\tau}(\rho_{y,k}^D) O_{b',c',d,\tau}^\dagger \otimes |c'\rangle \langle c'| \quad (101)$$

We will partition the above state into components using the following projection (the set $R_{c,d,k}^D$ is defined in (95)):

$$P_{c,k}^D = \mathcal{I} \otimes \sum_{d \in \{0,1\}^w, y \in R_{c,d,k}^D} |d\rangle \langle d| \otimes \mathcal{I} \otimes |y\rangle \langle y| \quad (102)$$

The state of \mathcal{A} in (101) can now be written in terms of the state $\tilde{\sigma}_{r,k}^D$, as defined in (96):

$$= \frac{1}{2} \sum_{c,c' \in \{0,1\}} P_{c,k}^D \tilde{\sigma}_{c \oplus c',k}^D P_{c,k}^D \otimes |c'\rangle \langle c'| \quad (103)$$

Finally, \mathcal{A} applies the efficiently computable CPTP map \mathcal{S} (which is used to distinguish between σ_0^D and σ_1^D) to the state in (103) and measures the first qubit. If the result of the measurement is $r \in \{0, 1\}$, \mathcal{A} outputs $b, x_b, y, d, c' \oplus r$.

In order to violate the hardcore bit clause of Definition 4.1, \mathcal{A} must output $(b, x_b, d, d \cdot (J(x_0) \oplus J(x_1)))$ with non negligible advantage (over outputting $(b, x_b, d, d \cdot (J(x_0) \oplus J(x_1)) \oplus 1)$). More formally, we need to show that the following advantage of \mathcal{A} (taken from Definition 4.1) is non negligible:

$$\left| \Pr_{(k,t_k) \leftarrow \text{GEN}_{\mathcal{F}}(1^\lambda)} [\mathcal{A}(k) \in H_k] - \Pr_{(k,t_k) \leftarrow \text{GEN}_{\mathcal{F}}(1^\lambda)} [\mathcal{A}(k) \in \overline{H}_k] \right| \leq \mu \quad (104)$$

where

$$H_k = \{(b, x_b, d, d \cdot (J(x_0) \oplus J(x_1))) | b \in \{0, 1\}, (x_0, x_1) \in \mathcal{R}_k, d \in G_{k,0,x_0} \cap G_{k,1,x_1}\} \quad (105)$$

$$\overline{H}_k = \{(b, x_b, d, c) | (b, x_b, d, c \oplus 1) \in H_k\} \quad (106)$$

\mathcal{A} outputs a string in H_k if, on components $P_{c,k}^D \tilde{\sigma}_{0,k}^D P_{c,k}^D$ and $P_{c,k}^D \tilde{\sigma}_{1,k}^D P_{c,k}^D$, the final bit of \mathcal{A} 's output is c . This occurs as long as the distinguishing operator \mathcal{S} outputs r on components $P_{0,k}^D \tilde{\sigma}_{r,k}^D P_{0,k}^D$ and $P_{1,k}^D \tilde{\sigma}_{r,k}^D P_{1,k}^D$. It follows that the probability that \mathcal{A} outputs a string in H_k is equal to:

$$\Pr_{(k,t_k) \leftarrow \text{GEN}_{\mathcal{F}}(1^\lambda)} [\mathcal{A}(k) \in H_k] = \frac{1}{2} \sum_{\substack{k \in \mathcal{K}_{\mathcal{F}} \\ r \in \{0,1\}}} D_{\mathbb{V},h}(k) \cdot \text{Tr}(|r\rangle \langle r| \otimes \mathcal{I}) \mathcal{S} \left(\sum_{c \in \{0,1\}} P_{c,k}^D \tilde{\sigma}_{r,k}^D P_{c,k}^D \right) \quad (107)$$

$$= \frac{1}{2} \sum_{\substack{k \in \mathcal{K}_{\mathcal{F}} \\ r \in \{0,1\}}} D_{\mathbb{V},h}(k) \cdot \text{Tr}(|r\rangle \langle r| \otimes \mathcal{I}) \mathcal{S}(\tilde{\sigma}_{r,k}^D) \quad (108)$$

By similar reasoning,

$$\Pr_{(k,t_k) \leftarrow \text{GEN}_{\mathcal{F}}(1^\lambda)} [\mathcal{A}(k) \in \overline{H}_k] = \frac{1}{2} \sum_{\substack{k \in \mathcal{K}_{\mathcal{F}} \\ r \in \{0,1\}}} D_{\mathbb{V},h}(k) \cdot \text{Tr}(|r \oplus 1\rangle \langle r \oplus 1| \otimes \mathcal{I}) \mathcal{S}(\tilde{\sigma}_{r,k}^D) \quad (109)$$

By combining (107) and (109) and then using the equality in (97), we obtain that the advantage of \mathcal{A} in (104) is equal to:

$$\left| \sum_{k \in \mathcal{K}_{\mathcal{F}}} D_{\mathbb{V},h}(k) \cdot \text{Tr}(|0\rangle\langle 0| \otimes \mathcal{I}) \mathcal{S}(\tilde{\sigma}_{0,k}^D - \tilde{\sigma}_{1,k}^D) \right| = \left| \sum_{k \in \mathcal{K}_{\mathcal{F}}} D_{\mathbb{V},h}(k) \cdot \text{Tr}(|0\rangle\langle 0| \otimes \mathcal{I}) \mathcal{S}(\sigma_{0,k}^D - \sigma_{1,k}^D) \right| \quad (110)$$

The expression in (110) is non negligible, due to our initial assumption that the CPTP map \mathcal{S} can distinguish between σ_0^D and σ_1^D (see (94)).

□

6.2 Indistinguishability of Cross Terms (Proof of Claim 6.4)

The overview of the following proof is given in Section 2.4.1.2.

Proof of Claim 6.4: Recall that we would like to show that the density matrices $\hat{\sigma}_0$ and $\hat{\sigma}_1$ are computationally indistinguishable. We will proceed by contradiction. Assume the two matrices $\hat{\sigma}_0$ and $\hat{\sigma}_1$ are computationally distinguishable using the efficiently computable CPTP map \mathcal{S} . It follows by the definition of computational indistinguishability (Definition 3.4) and the expression for $\hat{\sigma}_r$ in (92) that the following expression is non negligible:

$$\left| \sum_{k \in \mathcal{K}_{\mathcal{F}}} D_{\mathbb{V},h}(k) \cdot \text{Tr}(|0\rangle\langle 0| \otimes \mathcal{I}) \mathcal{S}\left(\sum_{r \in \{0,1\}} (-1)^r \hat{\sigma}_{r,k}\right) \right| \quad (111)$$

We will use the CPTP map \mathcal{S} to construct a BQP attacker \mathcal{A} who will violate the hardcore bit clause of Definition 4.4.

Fix a string $d \in \{0,1\}^w$. The attacker \mathcal{A} (on input $k \in \mathcal{K}_{\mathcal{F}}$) first constructs the following state from (72):

$$\sum_{y \in T_k} \rho_{y,k} \quad (112)$$

He does this exactly as the BQP prover \mathbb{P} would have: by applying the prover's initial operator U_0 the to input state and measuring the last register to obtain y . Then \mathcal{A} applies Z^d to the preimage register. The resulting state is:

$$(\mathcal{I} \otimes Z^d \otimes \mathcal{I}) \left(\sum_{y \in T_k} \rho_{y,k} \right) (\mathcal{I} \otimes Z^d \otimes \mathcal{I}) = \sum_{y \in T_k} (Z^{d \cdot (J(x_{0,y}) \oplus J(x_{1,y}))} \otimes \mathcal{I}) \rho_{y,k} (Z^{d \cdot (J(x_{0,y}) \oplus J(x_{1,y}))} \otimes \mathcal{I}) \quad (113)$$

The equality is due to the format of the state $\rho_{y,k}$, as written in (73):

$$\rho_{y,k} = \left(\sum_{b \in \{0,1\}} |b, J(x_{b,y})\rangle |\psi'_{b,y,k}\rangle \right) \left(\sum_{b \in \{0,1\}} |b, J(x_{b,y})\rangle |\psi'_{b,y,k}\rangle \right)^\dagger \otimes |y\rangle\langle y| \quad (114)$$

If we let

$$R_{c,d,k}^C = \{y \in T_k \mid d \cdot (J(x_{0,y}) \oplus J(x_{1,y})) = c\} \quad (115)$$

the expression in (113) is equal to:

$$\sum_{c \in \{0,1\}, y \in R_{c,d,k}^C} (Z^c \otimes \mathcal{I}) \rho_{y,k} (Z^c \otimes \mathcal{I}) \quad (116)$$

Finally, \mathcal{A} chooses a random bit c' , applies $Z^{c'}$ to the committed qubit and stores c' in an auxiliary register. Continuing from (116), the state of \mathcal{A} at this point is equal to:

$$\frac{1}{2} \sum_{c, c' \in \{0,1\}, y \in R_{c,d,k}^C} (Z^{c \oplus c'} \otimes \mathcal{I}) \rho_{y,k} (Z^{c \oplus c'} \otimes \mathcal{I}) \otimes |c'\rangle\langle c'| \quad (117)$$

We partition the state in (117) into components using the following projection:

$$P_{c,k}^C = \mathcal{I} \otimes \sum_{y \in R_{c,d,k}^C} |y\rangle \langle y| \quad (118)$$

The state in (117) can now be written in terms of $\hat{\sigma}_{r,k}$, as defined in (93)

$$\hat{\sigma}_{r,k} = (Z^r \otimes \mathcal{I}) \left(\sum_{y \in T_k} \rho_{y,k} \right) (Z^r \otimes \mathcal{I}) \quad (119)$$

as follows:

$$= \frac{1}{2} \sum_{c,c' \in \{0,1\}} P_{c,k}^C (Z^{c \oplus c'} \otimes \mathcal{I}) \left(\sum_{y \in T_k} \rho_{y,k} \right) (Z^{c \oplus c'} \otimes \mathcal{I}) P_{c,k}^C \otimes |c'\rangle \langle c'| \quad (120)$$

$$= \frac{1}{2} \sum_{c,c' \in \{0,1\}} P_{c,k}^C (\hat{\sigma}_{c \oplus c',k}) P_{c,k}^C \otimes |c'\rangle \langle c'| \quad (121)$$

Finally, \mathcal{A} applies the CPTP map \mathcal{S} (which is used to distinguish between $\hat{\sigma}_0$ and $\hat{\sigma}_1$) to the state in (121) and measures the first qubit. If the result of the measurement is $r \in \{0,1\}$, \mathcal{A} outputs $c' \oplus r$.

In order to violate the hardcore bit clause of Definition 4.4, \mathcal{A} must guess the value of $d \cdot (J(x_{0,y}) \oplus J(x_{1,y}))$ with non negligible advantage. More formally, we need to show that the following advantage of \mathcal{A} (taken from Definition 4.4) is non negligible:

$$\left| \Pr_{(k,t_k) \leftarrow \text{GEN}_{\mathcal{F}}(1^\lambda)} [\mathcal{A}(k) \in H'_{k,d}] - \frac{1}{2} \right| \quad (122)$$

where

$$H'_{k,d} = \{d \cdot (J(x_0) \oplus J(x_1)) \mid (x_0, x_1) \in \mathcal{R}_k\} \quad (123)$$

The bit output by \mathcal{A} is in $H'_{k,d}$ if, on components $P_{c,k}^C \hat{\sigma}_{0,k} P_{c,k}^C$ and $P_{c,k}^C \hat{\sigma}_{1,k} P_{c,k}^C$, \mathcal{A} outputs c . This occurs as long as the distinguishing operator \mathcal{S} outputs r on components $P_{0,k}^C \hat{\sigma}_{r,k} P_{0,k}^C$ and $P_{1,k}^C \hat{\sigma}_{r,k} P_{1,k}^C$. It follows that the probability that \mathcal{A} outputs a value in $H'_{k,d}$ is equal to:

$$\Pr_{(k,t_k) \leftarrow \text{GEN}_{\mathcal{F}}(1^\lambda)} [\mathcal{A}(k, d) \in H'_{k,d}] = \frac{1}{2} \sum_{\substack{k \in \mathcal{K}_{\mathcal{F}} \\ r \in \{0,1\}}} D_{\mathbb{V},h}(k) \cdot \text{Tr}(|r\rangle \langle r| \otimes \mathcal{I}) \mathcal{S} \left(\sum_{c \in \{0,1\}} P_{c,k}^C \hat{\sigma}_{r,k} P_{c,k}^C \right) \quad (124)$$

$$= \frac{1}{2} \sum_{\substack{k \in \mathcal{K}_{\mathcal{F}} \\ r \in \{0,1\}}} D_{\mathbb{V},h}(k) \cdot \text{Tr}(|r\rangle \langle r| \otimes \mathcal{I}) \mathcal{S}(\hat{\sigma}_{r,k}) \quad (125)$$

$$= \frac{1}{2} \sum_{k \in \mathcal{K}_{\mathcal{F}}} D_{\mathbb{V},h}(k) \cdot \text{Tr}(|0\rangle \langle 0| \otimes \mathcal{I}) \mathcal{S}(\hat{\sigma}_{0,k} - \hat{\sigma}_{1,k}) + \frac{1}{2} \quad (126)$$

We can use (126) to write the advantage of \mathcal{A} from (122) as:

$$\frac{1}{2} \left| \sum_{k \in \mathcal{K}_{\mathcal{F}}} D_{\mathbb{V},h}(k) \cdot \text{Tr}(|0\rangle \langle 0| \otimes \mathcal{I}) \mathcal{S}(\hat{\sigma}_{0,k} - \hat{\sigma}_{1,k}) \right| \quad (127)$$

The expression in (127) is non negligible, due to our initial assumption that the CPTP map \mathcal{S} can distinguish between $\hat{\sigma}_0$ and $\hat{\sigma}_1$ (see (111)).

□

6.3 Proof of Claim 6.2

Proof: If σ_0 is computationally distinguishable from σ_1 , by Definition 3.4 and the expression for σ_r in (82) there exists an efficiently computable CPTP map \mathcal{S} for which the following expression is non negligible:

$$\left| \sum_{k \in \mathcal{K}_{\mathcal{F}}} D_{\mathbb{V},h}(k) \cdot \text{Tr}(|0\rangle\langle 0| \otimes \mathcal{I}) \mathcal{S} \left(\sum_{r \in \{0,1\}} (-1)^r \sigma_{r,k} \right) \right| \quad (128)$$

We use the expression for $\sigma_{r,k}$ from (89) and the matrix $\sigma_{r,k}^D$ from (91) to define the matrix $\sigma_{r,k}^C$:

$$\sigma_{r,k} = \sum_{\substack{b',c \in \{0,1\}, \tau \\ d \in \{0,1\}^w, y \in R_{c,d,k}}} \delta_{d,y} O_{b',c \oplus r,d,\tau} (\rho_{y,k}^D + (-1)^r \rho_{y,k}^C) O_{b',c \oplus r,d,\tau}^\dagger \quad (129)$$

$$= \sigma_{r,k}^D + \sigma_{r,k}^C \quad (130)$$

By combining the following equality

$$\sum_{r \in \{0,1\}} (-1)^r \sigma_{r,k} = \sum_{r \in \{0,1\}} (-1)^r \sigma_{r,k}^D + \sum_{r \in \{0,1\}} (-1)^r \sigma_{r,k}^C \quad (131)$$

with the triangle inequality, it follows that if the quantity in (128) is non negligible, one of the following two quantities must be non negligible:

$$\left| \sum_{k \in \mathcal{K}_{\mathcal{F}}} D_{\mathbb{V},h}(k) \cdot \text{Tr}(|0\rangle\langle 0| \otimes \mathcal{I}) \mathcal{S} \left(\sum_{r \in \{0,1\}} (-1)^r \sigma_{r,k}^D \right) \right| \quad (132)$$

$$\left| \sum_{k \in \mathcal{K}_{\mathcal{F}}} D_{\mathbb{V},h}(k) \cdot \text{Tr}(|0\rangle\langle 0| \otimes \mathcal{I}) \mathcal{S} \left(\sum_{r \in \{0,1\}} (-1)^r \sigma_{r,k}^C \right) \right| \quad (133)$$

If the quantity in (132) is non negligible, σ_0^D is computationally distinguishable from σ_1^D . To complete the proof of Claim 6.2, we will show that if the quantity in 133 is non negligible, $\hat{\sigma}_0$ is computationally distinguishable from $\hat{\sigma}_1$ ($\hat{\sigma}_r$ is defined in (93) and copied below in (140)). To do this, we show below that for the efficiently computable CPTP map $\mathcal{S}' = \{ \frac{1}{\sqrt{2}} O_{b',c,d,\tau} \}_{b',c,d} (O_{b',c,d,\tau}$ is introduced in (78)):

$$\sum_{r \in \{0,1\}} (-1)^r \sigma_{r,k}^C = \mathcal{S}' \left(\sum_{r \in \{0,1\}} (-1)^r \hat{\sigma}_{r,k} \right) \quad (134)$$

Therefore, if the quantity in (133) is non negligible, $\hat{\sigma}_0$ is computationally distinguishable from $\hat{\sigma}_1$ by using the CPTP map $\mathcal{S}\mathcal{S}'$.

We now prove (134), beginning with the expression for $\sigma_{r,k}^C$ in (130). First, we observe that

$$\sigma_{1,k}^C = -(X \otimes \mathcal{I}) \sigma_{0,k}^C (X \otimes \mathcal{I}) \quad (135)$$

Therefore:

$$\sigma_{0,k}^C - \sigma_{1,k}^C = \sum_{r \in \{0,1\}} (X^r \otimes \mathcal{I}) \sigma_{0,k}^C (X^r \otimes \mathcal{I}) \quad (136)$$

$$= \sum_{\substack{b',c,r \in \{0,1\}, \tau \\ d \in \{0,1\}^w, y \in R_{c,d,k}}} \delta_{d,y} O_{b',r,d,\tau} (\rho_{y,k}^C) O_{b',r,d,\tau}^\dagger \quad (137)$$

$$= \sum_{\substack{b',c \in \{0,1\}, \tau \\ d \in \{0,1\}^w, y \in T_k}} O_{b',c,d,\tau} (\rho_{y,k}^C) O_{b',c,d,\tau}^\dagger \quad (138)$$

The second inequality follows since $\delta_{d,y} = \frac{1}{2}$ if $d \notin G_{k,0,x_{0,y}} \cap G_{k,1,x_{1,y}}$ and 1 if $d \in G_{k,0,x_{0,y}} \cap G_{k,1,x_{1,y}}$ and (from (76))

$$R_{c,d,k} = \{y \in T_k | (d \in G_{k,0,x_{0,y}} \cap G_{k,1,x_{1,y}} \wedge d \cdot (J(x_{0,y}) \oplus J(x_{1,y})) = c) \vee (d \notin G_{k,0,x_{0,y}} \cap G_{k,1,x_{1,y}})\} \quad (139)$$

Recall the state $\hat{\sigma}_{r,k}$ from (93):

$$\hat{\sigma}_{r,k} = (Z^r \otimes \mathcal{I}) \left(\sum_{y \in T_k} \rho_{y,k} \right) (Z^r \otimes \mathcal{I}) \quad (140)$$

We use the equality $\rho_{y,k} = \rho_{y,k}^D + \rho_{y,k}^C$ from (85):

$$\hat{\sigma}_{0,k} - \hat{\sigma}_{1,k} = \sum_{y \in T_k} (\rho_{y,k}^D + \rho_{y,k}^C) - (Z \otimes \mathcal{I}) \sum_{y \in T_k} (\rho_{y,k}^D + \rho_{y,k}^C) (Z \otimes \mathcal{I}) \quad (141)$$

$$= 2 \sum_{y \in T_k} \rho_{y,k}^C \quad (142)$$

Plugging the equality in (142) into (138) yields (134). \square

7 Measurement Protocol Soundness

In this section, we prove soundness of the measurement protocol, as stated in the following claim:

Claim 7.1 Soundness of Protocol 5.1 *For a prover \mathbb{P} in Protocol 5.1, let $1 - p_{h,H}$ be the probability that the verifier accepts \mathbb{P} on basis choice h in the Hadamard round and $1 - p_{h,T}$ be the probability that the verifier accepts \mathbb{P} in the test round. There exists a state ρ , a prover \mathbb{P}' and a negligible function μ such that for all h , $\|D_{\mathbb{P},h} - D_{\mathbb{P}',h}\|_{TV} \leq p_{h,H} + \sqrt{p_{h,T}} + \mu$ and $D_{\mathbb{P}',h}$ is computationally indistinguishable from the distribution $D_{\rho,h}$ which results from measuring ρ in the basis determined by h .*

Throughout this section, we will use notation introduced in Section 5.3. To prove Claim 7.1, we will proceed as follows. We begin by transitioning from a general prover to a *perfect* prover, i.e. a prover who is always accepted in the test run by the verifier. We show in the following claim (which we prove in Section 7.1) that for all provers \mathbb{P} , there exists a perfect prover \mathbb{P}' such that the statistical distance between the distributions $D_{\mathbb{P},h}$ and $D_{\mathbb{P}',h}$ is a function of the probability of acceptance of the general prover.

Claim 7.2 *For a prover \mathbb{P} in Protocol 5.1, let $1 - p_{h,H}$ be the probability that the verifier accepts \mathbb{P} on basis choice h in the Hadamard round and $1 - p_{h,T}$ be the probability that the verifier accepts \mathbb{P} on basis choice h in the test round. There exists a perfect prover \mathbb{P}' and a negligible function μ such that for all h , $\|D_{\mathbb{P},h}^C - D_{\mathbb{P}',h}\|_{TV} \leq p_{h,H} + \sqrt{p_{h,T}} + \mu$.*

\mathbb{P}' is created by conditioning \mathbb{P} on acceptance in the test round. This argument is straightforward, but does have one delicate aspect: we need to ensure that the perfect prover is still efficient, even though we have conditioned on acceptance in the test run. This is taken care of by recalling that in the test round, the verifier computes the procedure $\text{CHK}_{\mathcal{F}}$ or $\text{CHK}_{\mathcal{G}}$ (see Protocol 5.1). By definition (Definitions 4.1 and 4.2), both of these procedures require only the function key and not the trapdoor, which implies that the procedures can be computed efficiently by the prover.

Next, we transition from a perfect prover to a trivial prover (as defined in Definition 5.6, a trivial prover's Hadamard round attack commutes with standard basis measurement on the n committed qubits):

Claim 7.3 For all perfect provers \mathbb{P} , there exists a trivial prover $\hat{\mathbb{P}}$ such that for all h , $D_{\mathbb{P},h}$ is computationally indistinguishable from $D_{\hat{\mathbb{P}},h}$.

We prove Claim 7.3 in Section 7.2 as follows. First, in Claim 7.4, we prove a statement analogous to Claim 6.1 for the standard basis: we show that for an index j such that $h_j = 0$, we can replace the prover's Hadamard round attack with an attack which acts X -trivially on the j^{th} committed qubit to obtain the same distribution over measurements. The proof of Claim 7.4 is quite straightforward, since the prover's Hadamard round attack has no effect on the standard basis measurement obtained by the verifier; this measurement is obtained from the commitment y_j . To prove Claim 7.3, we sequentially apply Claim 6.1 and Claim 7.4 to each of the n committed qubits, ending with an attack which commutes with standard basis measurement on all of the committed qubits.

Assuming Claim 7.2 and Claim 7.3, the proof of Claim 7.1 is straightforward:

Proof of Claim 7.1: We begin with a prover \mathbb{P} and apply Claim 7.2 to transition to a perfect prover \mathbb{P}' such that for all h , $\|D_{\mathbb{P},h}^C - D_{\mathbb{P}',h}\|_{TV} \leq p_{h,H} + \sqrt{p_{h,T}} + \mu$ for a negligible function μ . Combining Claim 7.3 and Claim 5.7 tells us that there exists a state ρ such that for all h , $D_{\mathbb{P}',h}$ is computationally indistinguishable from $D_{\rho,h}$. In more detail, Claim 7.3 shows that there exists a trivial prover $\hat{\mathbb{P}}$ such that for all h , $D_{\mathbb{P}',h}$ is computationally indistinguishable from $D_{\hat{\mathbb{P}},h}$. Next, Claim 5.7 shows that there exists a state ρ such that for all h , $D_{\hat{\mathbb{P}},h}$ is computationally indistinguishable from $D_{\rho,h}$. \square

7.1 General to Perfect Prover (Proof of Claim 7.2)

Proof: We begin by observing that by definition of total variation distance (see (28)):

$$\|D_{\mathbb{P},h}^C - D_{\mathbb{P},h}\|_{TV} = p_{h,H} \quad (143)$$

It follows by (57) that $\|\sigma_{\mathbb{P},h}^C - \sigma_{\mathbb{P},h}\|_{tr} = p_{h,H}$. In the rest of this proof, we will show that there exists a perfect prover \mathbb{P}' and a negligible function μ such that for all h for which $1 - p_{h,T}$ is non negligible, the trace distance between $\sigma_{\mathbb{P},h}$ and $\sigma_{\mathbb{P}',h}$ is $\leq \sqrt{p_{h,T}} + \mu$. For all h for which $1 - p_{h,T}$ is negligible, the trace distance bound of $\sqrt{p_{h,T}} + \mu$ is trivial and therefore satisfied. It follows by the triangle inequality that for all h , the trace distance between $\sigma_{\mathbb{P},h}^C$ and $\sigma_{\mathbb{P}',h}$ is $\leq p_{h,H} + \sqrt{p_{h,T}} + \mu$ (to complete the proof of the claim, recall from (57) that $\|\sigma_{\mathbb{P},h}^C - \sigma_{\mathbb{P}',h}\|_{tr} = \|D_{\mathbb{P},h}^C - D_{\mathbb{P}',h}\|_{TV}$).

Distance from perfect prover (between $\sigma_{\mathbb{P},h}$ and $\sigma_{\mathbb{P}',h}$) By Claim 5.4, there exist unitaries U_0, U such that the prover \mathbb{P} is characterized by (U_0, U) and the state $\sigma_{\mathbb{P},h}$ can be created by following steps 1-4 in the statement of the claim. The state after application of U_0 (step 1 of Claim 5.4) is:

$$\sigma_{\mathbb{P},h,0} = \sum_{k'} D_{\mathbb{V},h}(k') \cdot U_0(|0\rangle \langle 0|^{\otimes e} \otimes |k'\rangle \langle k'|) U_0^\dagger \quad (144)$$

The state above is a mixed state over the verifier's choice of the function key k' , which is sampled according to the distribution $D_{\mathbb{V},h}$ (see Section 5.3 for a notation reminder). To create $\sigma_{\mathbb{P},h}$ (i.e. the state resulting from the Hadamard round defined in (56)), the prover will measure the second to last register (obtaining $y' \in \mathcal{Y}^n$), apply his attack U , measure all committed qubits and preimage registers in the Hadamard basis, send the results to the verifier and discard all other qubits. The verifier will decode the appropriate registers and discard all other measurement results (as described in Protocol 5.1). Note that for all provers \mathbb{P}, \mathbb{P}' :

$$\|\sigma_{\mathbb{P},h} - \sigma_{\mathbb{P}',h}\|_{tr} \leq \|\sigma_{\mathbb{P},h,0} - \sigma_{\mathbb{P}',h,0}\|_{tr} \quad (145)$$

This is because the operators described above which are applied to $\sigma_{\mathbb{P},h,0}$ to create $\sigma_{\mathbb{P},h}$ represent a CPTP map.

We now construct a perfect prover \mathbb{P}' and provide an upper bound for $\|\sigma_{\mathbb{P},h,0} - \sigma_{\mathbb{P}',h,0}\|_{tr}$. We begin by partitioning the state $\sigma_{\mathbb{P},h,0}$ from (144) according to acceptance or rejection by the verifier in the test round:

$$\sigma_{\mathbb{P},h,0} = \sum_{k'} D_{\mathbb{V},h}(k') \cdot (|\psi_{ACC,k'}\rangle + |\psi_{REJ,k'}\rangle)(\langle\psi_{ACC,k'}| + \langle\psi_{REJ,k'}|) \otimes |k'\rangle\langle k'| \quad (146)$$

where

$$U_0 |0\rangle^{\otimes e} |k'\rangle = U_{0,k'}(|0\rangle^{\otimes e}) \otimes |k'\rangle = (|\psi_{ACC,k'}\rangle + |\psi_{REJ,k'}\rangle) \otimes |k'\rangle \quad (147)$$

The first equality in (147) is given in (65) in Claim 5.4, $|\psi_{ACC,k'}\rangle$ (resp. $|\psi_{REJ,k'}\rangle$) is the part of the state $U_{0,k'}(|0\rangle^{\otimes e})$ which will be accepted (resp. rejected) by the verifier in the test round for function choice k' , and $\langle\psi_{ACC,k'}|\psi_{REJ,k'}\rangle = 0$. Consider the following state:

$$\sigma_{perfect} = \frac{1}{1 - p_{h,T}} \sum_{k'} D_{\mathbb{V},h}(k') \cdot (|\psi_{ACC,k'}\rangle)(\langle\psi_{ACC,k'}|) \otimes |k'\rangle\langle k'| \quad (148)$$

The trace distance between $\sigma_{perfect}$ and $\sigma_{\mathbb{P},h,0}$ is at most $\sqrt{p_{h,T}}$. We show below that there exists a CPTP map \mathcal{S}_0 , a perfect prover \mathbb{P}' characterized by (\mathcal{S}_0, U) and a negligible function μ such that for all h for which $1 - p_{h,T}$ is non negligible, $\|\sigma_{\mathbb{P}',h,0} - \sigma_{perfect}\|_{tr} \leq \mu$. It follows by the triangle inequality that:

$$\|\sigma_{\mathbb{P},h,0} - \sigma_{\mathbb{P}',h,0}\|_{tr} \leq \|\sigma_{\mathbb{P},h,0} - \sigma_{perfect}\|_{tr} + \|\sigma_{perfect} - \sigma_{\mathbb{P}',h,0}\|_{tr} \quad (149)$$

$$\leq \sqrt{p_{h,T}} + \mu \quad (150)$$

It then follows by (145) that $\|\sigma_{\mathbb{P},h} - \sigma_{\mathbb{P}',h}\|_{tr} \leq \sqrt{p_{h,T}} + \mu$, which completes the proof of Claim 7.2.

We now describe the CPTP map \mathcal{S}_0 . To do so, we will require the unitary V , which consists of first applying U_0 , then applying the verifier's test round check (the algorithm $\text{CHK}_{\mathcal{F}}$) in superposition, and storing the result in an extra register. The result of applying V is:

$$V(\sum_{k'} D_{\mathbb{V},h}(k') \cdot |0\rangle\langle 0|^{\otimes e} \otimes |k'\rangle\langle k'|)V^\dagger \quad (151)$$

$$= \sum_{k'} D_{\mathbb{V},h}(k') \cdot (|1\rangle|\psi_{ACC,k'}\rangle + |0\rangle|\psi_{REJ,k'}\rangle)(|1\rangle|\psi_{ACC,k'}\rangle + |0\rangle|\psi_{REJ,k'}\rangle)^\dagger \otimes |k'\rangle\langle k'| \quad (152)$$

where the first register contains the result of the algorithm $\text{CHK}_{\mathcal{F}}$. If the first register is measured to obtain a bit b , the probability of obtaining $b = 1$ is $1 - p_{h,T}$. Therefore, for all h for which $1 - p_{h,T}$ is non negligible, there exists a polynomial n'_h such that if V is applied n'_h times to the input state in (151) the probability of never obtaining $b = 1$ over n'_h iterations is negligible.

Let $n' = \max_h n'_h$, where the maximum is taken over all h such that $1 - p_{h,T}$ is non negligible. The CPTP map \mathcal{S}_0 will apply the unitary V followed by measurement of the first qubit to the input $|0\rangle^{\otimes e} |k'\rangle$ at most n' times or until measuring $b = 1$. In the case that the measurement result $b = 1$ is never obtained, the CPTP map \mathcal{S}_0 applies a unitary operator which would be applied by an honest prover (described in Protocol 5.2). Since the prover in Protocol 5.2 is perfect (see Claim 5.3), it follows that the prover \mathbb{P}' characterized by (\mathcal{S}_0, U) is perfect.

Due to the choice of n' , it follows that there exists a negligible function μ such that for all h for which $1 - p_{h,T}$ is non negligible

$$\sigma_{\mathbb{P}',h,0} = \mathcal{S}_0(\sum_{k'} D_{\mathbb{V},h}(k') \cdot (|0\rangle\langle 0|^{\otimes e} \otimes |k'\rangle\langle k'|)) = (1 - \mu)\sigma_{perfect} + \mu\sigma_{fail} \quad (153)$$

where σ_{fail} is the state created in the case that all n' applications of V do not yield 1 as the measurement result. It follows by the convexity of trace distance that for all h for which $1 - p_{h,T}$ is non negligible the trace distance between $\sigma_{\mathbb{P}',h,0}$ in (153) and $\sigma_{perfect}$ is $\leq \mu$.

□

7.2 Perfect to Trivial Prover (Proof of Claim 7.3)

We begin by proving a claim which is analogous to Claim 6.1, but for the standard basis (i.e. if $h_j = 0$):

Claim 7.4 *Let $1 \leq j \leq n$. Let $\mathcal{S} = \{B_\tau\}_\tau$ and $\mathcal{S}_j = \{B'_{j,x,\tau}\}_{x \in \{0,1\}, \tau}$ be CPTP maps written in terms of their Kraus operators:*

$$B_\tau = \sum_{x,z \in \{0,1\}} X^x Z^z \otimes B_{jxz\tau} \quad (154)$$

$$B'_{j,x,\tau} = \sum_{z \in \{0,1\}} Z^z \otimes B_{jxz\tau} \quad (155)$$

where B_τ and $B'_{j,x,\tau}$ have been rearranged so that $X^x Z^z$ and Z^z act on the j^{th} qubit. For a unitary operator U_0 , let \mathbb{P} be a perfect prover characterized by (U_0, \mathcal{S}) (see Claim 5.4 and notation 5.5). Let \mathbb{P}_j be a perfect prover characterized by (U_0, \mathcal{S}_j) . If $h_j = 0$, $D_{\mathbb{P},h} = D_{\mathbb{P}_j,h}$.

Proof of Claim 7.4: For convenience, we will assume that $h_1 = 0$ and we will prove the claim for $j = 1$. The proof is identical for all other values of j . The key observation is that we can change the CPTP map \mathcal{S} to act in an arbitrary way on the first qubit (which is also the first committed qubit), as long as its action on the remaining qubits is unchanged. This is because the measurement of the first committed qubit will be ignored by the verifier; the verifier will obtain the standard basis measurement for the first qubit from the commitment y_1 . In other words, the first committed qubit is traced out after application of \mathcal{S} and standard basis measurement (it is not part of the distribution $D_{\mathbb{P},h}$).

To prove the claim, we will show that the distribution over measurement results remains the same if $\mathcal{S} = \{B_\tau\}_\tau$ is replaced with $\mathcal{S}_1 = \{B'_{1,x,\tau}\}_{x \in \{0,1\}, \tau}$. Our first step is to replace \mathcal{S} with $\mathcal{S}^{(1)} = \{\frac{1}{\sqrt{2}}(Z^r \otimes \mathcal{I})B_\tau\}_{r \in \{0,1\}, \tau}$. As described above, this change has no impact since the measurement of the first committed qubit is ignored. Therefore, if we let $\mathbb{P}^{(1)}$ be a perfect prover characterized by $(U_0, \mathcal{S}^{(1)})$, $D_{\mathbb{P},h} = D_{\mathbb{P}^{(1)},h}$. Next, we replace $\mathcal{S}^{(1)}$ with $\mathcal{S}^{(2)} = \{\frac{1}{\sqrt{2}}(Z^r \otimes \mathcal{I})B_\tau(Z^r \otimes \mathcal{I})\}_{r \in \{0,1\}, \tau}$. Observe that the added Z operator (acting prior to B_τ) has no effect on the state of a perfect prover: it acts on the first committed qubit, which must be in a standard basis state since $h_1 = 0$ (the function key k_1 corresponds to a pair of injective functions $g_{k,0}, g_{k,1}$ for which there is only one valid preimage for each commitment string y - see equation 9 for more details). It follows that if we let $\mathbb{P}^{(2)}$ be the prover characterized by $(U_0, \mathcal{S}^{(2)})$, $D_{\mathbb{P}^{(1)},h} = D_{\mathbb{P}^{(2)},h}$. Finally, since $\mathcal{S}^{(2)}$ is followed by Hadamard basis measurement, we can apply Corollary 3.6 to replace $\mathcal{S}^{(2)}$ with \mathcal{S}_1 , showing that $D_{\mathbb{P}^{(2)},h} = D_{\mathbb{P}_1,h}$ (therefore $D_{\mathbb{P},h} = D_{\mathbb{P}_1,h}$).

□

Using Claim 6.1 and Claim 7.4, we now prove Claim 7.3, which is copied here for reference:

Claim 7.3 *For all perfect provers \mathbb{P} , there exists a trivial prover $\hat{\mathbb{P}}$ such that for all h , $D_{\mathbb{P},h}$ is computationally indistinguishable from $D_{\hat{\mathbb{P}},h}$.*

Proof of Claim 7.3: Fix a basis choice $h \in \{0, 1\}^n$. Assume for convenience that for indices $1 \leq i \leq t$, $h_i = 1$ and for indices $t + 1 \leq i \leq n$, $h_i = 0$. We apply Claim 6.1 t times, beginning with the prover \mathbb{P} (who is characterized by (U_0, U)). Let $\hat{\mathbb{P}}_0 = \mathbb{P}$ and let $\hat{\mathbb{P}}_j$ be the prover characterized by $(U_0, \{U_x\}_{x \in \{0, 1\}^j})$:

$$U = \sum_{x, z \in \{0, 1\}^j} X^x Z^z \otimes U_{xz}^j \quad (156)$$

$$U_x = \sum_{z \in \{0, 1\}^j} Z^z \otimes U_{xz}^j \quad (157)$$

where $X^x Z^z$ acts on qubits $1, \dots, j$ (which are also committed qubits $1, \dots, j$). The i^{th} application of Claim 6.1 shows that $D_{\hat{\mathbb{P}}_{i-1}, h}$ is computationally indistinguishable from $D_{\hat{\mathbb{P}}_i, h}$. It follows by the triangle inequality that $D_{\mathbb{P}, h}$ is computationally indistinguishable from $D_{\hat{\mathbb{P}}_t, h}$. We complete the proof of Claim 7.3 by applying Claim 7.4 $n - t$ times, which tells us that $D_{\hat{\mathbb{P}}_t, h} = D_{\hat{\mathbb{P}}, h}$, where $\hat{\mathbb{P}} = \hat{\mathbb{P}}_n$ is a trivial prover characterized by $(U_0, \{U_x\}_{x \in \{0, 1\}^n})$. \square

8 Extension of Protocol 5.1 to a Verification Protocol for BQP

To extend Protocol 5.1 to a QPIP₀, we will use the QPIP₁ protocol in [MF16]. We begin by presenting this protocol.

8.1 Morimae-Fitzsimons Protocol

Most of this description is taken directly from [MF16]. Let L be a language in BQP. Since BQP is contained in QMA, for all $x \in L$, there exists a local Hamiltonian H such that

1. if $x \in L$, then the ground energy of H is $\leq a$
2. if $x \notin L$, then the ground energy of H is $\geq b$

where $b - a \geq \frac{1}{\text{poly}(|x|)}$. It is known that H can be a 2-local Hamiltonian with only X and Z operators ([BL08]).

Let us write the 2-local Hamiltonian as $H = \sum_S d_S S$, where d_S is a real number and S is a tensor product of Pauli operators, where only two operators are Z or X and others are \mathcal{I} . We define the rescaled Hamiltonian:

$$H' = \sum_S \pi_S P_S, \quad (158)$$

where $\pi_S = \frac{|d_S|}{\sum_S |d_S|} \geq 0$ and $P_S = \frac{\mathcal{I} + \text{sign}(d_S)S}{2}$.

We now present the protocol:

Protocol 8.1 [MF16] *This protocol is used to verify that an instance $x \in L$ for a language $L \in \text{BQP}$. Let H be the Hamiltonian which corresponds to x , and define H' as in 158.*

1. The verifier randomly chooses S with probability π_S .
2. The prover sends the verifier a state ρ , sending one qubit at a time.
3. The verifier measures S by performing single qubit measurements of only two qubits of ρ in the X or Z basis, discarding all other qubits of ρ without measuring them.

4. The verifier computes the product of the measurement results. If the verifier obtains the result $-\text{sign}(d_S)$ the verifier accepts.

An honest prover would simply send the ground state of H' .

Theorem 8.2 [MF16] *Protocol 8.1 is a QPIP₁ for all languages in BQP with completeness c and soundness s , where $c - s$ is inverse polynomial in $|x|$.*

Proof of Theorem 8.2: It was shown (in [MNS15]) that the acceptance probability of each round of Protocol 8.1 is

$$p_{\text{acc}} = 1 - \frac{1}{\sum_S 2^{|d_S|}} (\text{Tr}(H\rho) + \sum_S |d_S|) \quad (159)$$

which is

$$p_{\text{acc}} \geq \frac{1}{2} - \frac{a}{\sum_S 2^{|d_S|}} \quad (160)$$

when $x \in L$, and

$$p_{\text{acc}} \leq \frac{1}{2} - \frac{b}{\sum_S 2^{|d_S|}} \quad (161)$$

when $x \notin L$. Their difference is $\frac{1}{\text{poly}(|x|)}$. \square

We require the following version of Protocol 8.1 with an amplified completeness/ soundness gap:

Protocol 8.3 [MF16] *This protocol is used to verify that an instance $x \in L$ for a language $L \in \text{BQP}$. Let H be the Hamiltonian which corresponds to x , and define H' as in 158. Let k' be a polynomial in $|x|$.*

1. The verifier randomly chooses $S_1, \dots, S_{k'}$ independently, each with probability π_{S_i} .
2. The prover sends the verifier a state ρ' , sending one qubit at a time.
3. The verifier measures each S_i (for $1 \leq i \leq k'$) by performing single qubit measurements of only two qubits of ρ' in the X or Z basis, discarding all other qubits of ρ' without measuring them.
4. The verifier computes the product of the measurement results for each S_i . If the verifier obtains the result $-\text{sign}(d_{S_i})$ more than half of the time the verifier accepts.

In Protocol 8.3, an honest prover would simply send k' copies of the ground state of H' .

Theorem 8.4 [MF16] *Protocol 8.1 is a QPIP₁ for all languages in BQP with completeness $1 - \mu$ and soundness μ , where μ is negligible in the size of the instance.*

Proof: This theorem follows from Theorem 8.2; since $c - s$ is $\frac{1}{\text{poly}(|x|)}$ the verifier can distinguish the case where $x \in L$ from the case where $x \notin L$ with probability of error bounded to be exponentially small with only polynomially many repetitions. \square

8.2 Extending Protocol 5.1

We now present the extension of Protocol 5.1 to a QPIP₀. To do this, we use the structure of Protocol 8.3, but we replace the prover sending qubits to the verifier to perform the measurement (i.e. steps 2 and 3 of Protocol 8.3) with Protocol 5.1. Assume that in Protocol 8.3, n qubits are sent from the prover to the verifier. The QPIP protocol is as follows:

Protocol 8.5 Measurement Protocol QPIP This protocol is used to verify that an instance $x \in L$ for a language $L \in \text{BQP}$.

1. The verifier performs step 1 of Protocol 8.3, which partially defines a basis choice $h \in \{0,1\}^n$. For all undefined h_i , the verifier sets $h_i = 0$.
2. The prover and verifier run Protocol 5.1 on basis choice h . The verifier accepts or rejects as specified in Protocol 5.1.
3. In the case of a Hadamard round of Protocol 5.1, the verifier performs step 4 of Protocol 8.3 using the measurement results obtained from Protocol 5.1.

We now prove the following theorem, which is the main result of this paper (and was stated earlier as Theorem 2.3):

Theorem 8.6 Protocol 8.5 is a QPIP_0 for all languages in BQP with completeness negligibly close to 1 and soundness negligibly close to $\frac{3}{4}$.

We require the completeness and soundness guarantees of Protocol 5.1, copied below for reference. Both claims use notation from Section 5.3.

Claim 5.3 Completeness of Protocol 5.1 For all n qubit states ρ and for all basis choices $h \in \{0,1\}^n$, the prover \mathbb{P} described in Protocol 5.2 is a perfect prover (\mathbb{P} is accepted by the verifier in a test run for basis choice h with perfect probability). There exists a negligible function μ such that in the Hadamard round for basis choice h , the verifier accepts \mathbb{P} with probability $\geq 1 - \mu$ and $\left\| D_{\mathbb{P},h}^C - D_{\rho,h} \right\|_{TV} \leq \mu$.

Claim 7.1 Soundness of Protocol 5.1 For a prover \mathbb{P} in Protocol 5.1, let $1 - p_{h,H}$ be the probability that the verifier accepts \mathbb{P} on basis choice h in the Hadamard round and $1 - p_{h,T}$ be the probability that the verifier accepts \mathbb{P} in the test round. There exists a state ρ , a prover \mathbb{P}' and a negligible function μ such that for all h , $\left\| D_{\mathbb{P},h}^C - D_{\mathbb{P}',h} \right\|_{TV} \leq p_{h,H} + \sqrt{p_{h,T}} + \mu$ and $D_{\mathbb{P}',h}$ is computationally indistinguishable from the distribution $D_{\rho,h}$ which results from measuring ρ in the basis determined by h .

Proof of Theorem 8.6: We will require some notation. For all provers \mathbb{P} , let $E_{\mathbb{P},h}^H$ be the event that the verifier accepts in a Hadamard round of Protocol 5.1 with basis choice h while interacting with \mathbb{P} , let $E_{\mathbb{P},h}^T$ be the same event in a test round, and let $E_{\mathbb{P},h}$ be the event that the verifier accepts in step 3 of Protocol 8.5. Let v_h be the probability that the verifier chooses basis choice h in step 1 of Protocol 8.5. As a reminder, in step 3 of Protocol 8.5 (which is step 4 of Protocol 8.3), the verifier determines whether or not to accept by computing the product of relevant measurement results; the verifier's decision is a function of the basis choice h , the measurement result and the BQP instance, but we leave off the dependence on the BQP instance for convenience. For a distribution D over n bit strings and a basis choice h , let $\tilde{p}_h(D)$ be the probability that the verifier rejects an n bit string drawn from D for basis choice h in step 3 of Protocol 8.5.

Completeness Recall that in Protocol 8.3, an honest prover sends polynomially many copies of the ground state for the Hamiltonian corresponding to an instance $x \in L$ (where $L \in \text{BQP}$). Let the entire state sent by the prover be ρ , and assume it contains n qubits. To compute the completeness parameter of Protocol 8.5, we will consider the prover \mathbb{P} for the state ρ , as described in Protocol 5.2, and upper bound the probability that the verifier rejects in Protocol 8.5. To do this, we need to upper bound the probability that the verifier rejects in step 2 of Protocol 8.5

(i.e. in Protocol 5.1) or the verifier rejects in step 3 of Protocol 8.5:

$$1 - c = \frac{1}{2} \sum_{h \in \{0,1\}^n} v_h (\Pr[\bar{E}_{\mathbb{P},h}^T] + \Pr[\bar{E}_{\mathbb{P},h}^H \cup \bar{E}_{\mathbb{P},h}]) \quad (162)$$

$$\leq \frac{1}{2} \sum_{h \in \{0,1\}^n} v_h (\Pr[\bar{E}_{\mathbb{P},h}^H] + \Pr[\bar{E}_{\mathbb{P},h}]) \quad (163)$$

$$\leq \frac{1}{2} \mu + \frac{1}{2} \sum_{h \in \{0,1\}^n} v_h \Pr[\bar{E}_{\mathbb{P},h}] \quad (164)$$

The last two expressions follow due to Claim 5.3: we know that for all basis choices $h \in \{0,1\}^n$, the prover \mathbb{P} described in Protocol 5.2 is accepted by the verifier in a test round with perfect probability and is accepted in a Hadamard round with probability $\geq 1 - \mu$ for a negligible function μ . $\Pr[\bar{E}_{\mathbb{P},h}]$ is the probability that the verifier rejects the distribution $D_{\mathbb{P},h}^C$ for basis choice h in step 3 of Protocol 8.5:

$$\Pr[\bar{E}_{\mathbb{P},h}] = \tilde{p}_h(D_{\mathbb{P},h}^C) \quad (165)$$

Recall from Section 5.3 that for a density matrix on n qubits and basis choice h , we let $D_{\rho,h}$ be the distribution obtained by measuring ρ in the basis corresponding to h . It follows by Lemma 3.1 and Claim 5.3 that

$$\tilde{p}_h(D_{\mathbb{P},h}^C) - \tilde{p}_h(D_{\rho,h}) \leq \|D_{\mathbb{P},h}^C - D_{\rho,h}\|_{TV} \leq \mu \quad (166)$$

Due to the completeness parameter of Protocol 8.3 (see Section 8.1), there exists a negligible function μ_C such that:

$$\sum_h v_h \tilde{p}_h(D_{\rho,h}) \leq \mu_C \quad (167)$$

We now use (165), (166) and (167) to wrap up the calculation of the completeness parameter c (continuing from (164)):

$$1 - c \leq \frac{1}{2} \mu + \frac{1}{2} \sum_{h \in \{0,1\}^n} v_h \tilde{p}_h(D_{\mathbb{P},h}^C) \quad (168)$$

$$\leq \mu + \frac{1}{2} \mu_C \quad (169)$$

Therefore, the completeness parameter c is negligibly close to 1.

Soundness To compute the soundness parameter, we will fix an arbitrary prover \mathbb{P} and upper bound the probability that the verifier accepts in Protocol 8.5 for an instance $x \notin L$. To do so, we need to upper bound the probability that the verifier accepts in step 2 of Protocol 8.5 (i.e. in Protocol 5.1) and the verifier accepts in step 3 of Protocol 8.5. The intuition here is that as long as there exists a state ρ such that for all h , $D_{\mathbb{P},h}^C$ is close (computationally) to $D_{\rho,h}$, the soundness parameter should be close to the soundness parameter of Protocol 8.3, which is negligible. We will rely on Claim 7.1 (we also use the same notation used in Claim 7.1):

$$s = \sum_{h \in \{0,1\}^n} v_h \left(\frac{1}{2} \Pr[E_{\mathbb{P},h}^T] + \frac{1}{2} \Pr[E_{\mathbb{P},h}^H \cap E_{\mathbb{P},h}] \right) \quad (170)$$

$$= \sum_{h \in \{0,1\}^n} v_h \left(\frac{1}{2} (1 - p_{h,T}) + \frac{1}{2} \Pr[E_{\mathbb{P},h}^H] \Pr[E_{\mathbb{P},h} | E_{\mathbb{P},h}^H] \right) \quad (171)$$

$$= \sum_{h \in \{0,1\}^n} v_h \left(\frac{1}{2} (1 - p_{h,T}) + \frac{1}{2} (1 - p_{h,H}) (1 - \tilde{p}_h(D_{\mathbb{P},h}^C)) \right) \quad (172)$$

where the last equality follows because $\Pr[E_{\mathbb{P},h}|E_{\mathbb{P},h}^H]$ is the probability that the verifier accepts a string drawn from the distribution $D_{\mathbb{P},h}^C$ for basis choice h in step 3 of Protocol 8.5.

Claim 7.1 guarantees the existence of a state ρ , a prover \mathbb{P}' and a negligible function μ such that for all h , $\|D_{\mathbb{P},h}^C - D_{\mathbb{P}',h}\|_{TV} \leq p_{h,H} + \sqrt{p_{h,T}} + \mu$ and $D_{\mathbb{P}',h}$ is computationally indistinguishable from $D_{\rho,h}$. By Lemma 3.1 and Claim 7.1:

$$\tilde{p}_h(D_{\mathbb{P}',h}) - \tilde{p}_h(D_{\mathbb{P},h}^C) \leq \|D_{\mathbb{P},h}^C - D_{\mathbb{P}',h}\|_{TV} \leq p_{h,H} + \sqrt{p_{h,T}} + \mu \quad (173)$$

We now return to the calculation of the soundness parameter of Protocol 8.5 in (172):

$$s \leq \sum_{h \in \{0,1\}^n} v_h \left(\frac{1}{2}(1 - p_{h,T}) + \frac{1}{2}(1 - p_{h,H})(1 - \tilde{p}_h(D_{\mathbb{P}',h}) + p_{h,H} + \sqrt{p_{h,T}} + \mu) \right) \quad (174)$$

$$\begin{aligned} &\leq \frac{1}{2}\mu + \frac{1}{2} \sum_{h \in \{0,1\}^n} v_h(1 - p_{h,T} + (1 - p_{h,H})(p_{h,H} + \sqrt{p_{h,T}})) + \frac{1}{2} \sum_{h \in \{0,1\}^n} v_h(1 - \tilde{p}_h(D_{\mathbb{P}',h})) \\ &\leq \frac{1}{2}\mu + \frac{3}{4} + \frac{1}{2} \sum_{h \in \{0,1\}^n} v_h(1 - \tilde{p}_h(D_{\mathbb{P}',h})) \end{aligned} \quad (175)$$

Next, Claim 7.1 guarantees that for all h , $D_{\mathbb{P}',h}$ and $D_{\rho,h}$ are computationally indistinguishable. It follows that for all h :

$$\tilde{p}_h(D_{\rho,h}) - \tilde{p}_h(D_{\mathbb{P}',h}) \leq \mu_h \quad (176)$$

To see this implication, assume there did exist an $h \in \{0,1\}^n$ such that the difference in (176) was non negligible. Then $D_{\mathbb{P}',h}$ and $D_{\rho,h}$ could be distinguished by computing whether or not the verifier would reject for basis choice h in step 3 of Protocol 8.5, which is step 4 of Protocol 8.3. This is because the computational indistinguishability of $D_{\mathbb{P}',h}$ and $D_{\rho,h}$ holds even if h is known; the indistinguishability is due to the hardcore bit and injective invariance properties of the extended trapdoor claw-free family (Definition 4.4).

Due to the soundness parameter of Protocol 8.3, we also know that there exists a negligible function μ_S such that:

$$\sum_{h \in \{0,1\}^n} v_h(1 - \tilde{p}_h(D_{\rho,h})) \leq \mu_S \quad (177)$$

We return to calculating the soundness parameter of the QPIP (continuing from (175)):

$$s \leq \frac{1}{2}\mu + \frac{3}{4} + \frac{1}{2} \sum_{h \in \{0,1\}^n} v_h(\mu_h + 1 - \tilde{p}_h(D_{\rho,h})) \quad (178)$$

$$\leq \frac{1}{2}\mu + \frac{3}{4} + \frac{1}{2} \max_{h \in \{0,1\}^n} \mu_h + \frac{1}{2}\mu_S \quad (179)$$

Therefore, the soundness parameter of Protocol 8.5 is negligibly close to $\frac{3}{4}$. \square

9 Acknowledgments

Thanks to Dorit Aharonov, Zvika Brakerski, Zeph Landau, Umesh Vazirani and Thomas Vidick for many useful discussions.

References

- [ABOE08] Dorit Aharonov, Michael Ben-Or, and Elad Eban. Interactive Proofs For Quantum Computations. *Arxiv preprint arXiv:0810.5375*, 2008.
- [ABOEM17] Dorit Aharonov, Michael Ben-Or, Elad Eban, and Urmila Mahadev. Interactive Proofs for Quantum Computations. *Arxiv preprint 1704.04487*, 2017.
- [AKPW13] Joel Alwen, Stephan Krenn, Krzysztof Pietrzak, and Daniel Wichs. Learning with rounding, revisited: New reduction, properties and applications. Cryptology ePrint Archive, Report 2013/098, 2013. <https://eprint.iacr.org/2013/098>.
- [Ban93] Wojciech Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(1):625–635, 1993.
- [BCM⁺18] Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh Vazirani, and Thomas Vidick. Certifiable randomness from a single quantum device. *Arxiv preprint 1804.00640*, 2018.
- [BFK08] Anne Broadbent, Joseph F. Fitzsimons, and Elham Kashefi. Universal blind quantum computation. *Arxiv preprint arXiv:0807.4154*, 2008.
- [BL08] Jacob D. Biamonte and Peter J. Love. Realizable hamiltonians for universal adiabatic quantum computers. *Phys. Rev. A*, 78:012352, Jul 2008.
- [BLP⁺13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC’13, Palo Alto, CA, USA, June 1-4, 2013*, pages 575–584. ACM, 2013.
- [FK17] Joseph F. Fitzsimons and Elham Kashefi. Unconditionally verifiable blind quantum computation. *Phys. Rev. A*, 96:012303, Jul 2017.
- [GKK17] Alexandru Gheorghiu, Theodoros Kapourniotis, and Elham Kashefi. Verification of quantum computation: An overview of existing approaches. *Arxiv preprint 1709.06984*, 2017.
- [GKPV10] Shafi Goldwasser, Yael Kalai, Chris Peikert, and Vinod Vaikuntanathan. Robustness of the learning with errors assumption. *ICS*, pages 230–240, 2010.
- [Got04] Daniel Gottesman, 2004. As referenced in [<http://www.scottaaronson.com/blog/?p=284>; accessed 13-Apr-2017].
- [GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. Cryptology ePrint Archive, Report 2013/340, 2013. <http://eprint.iacr.org/2013/340>.
- [Mah17] Urmila Mahadev. Classical Homomorphic Encryption for Quantum Circuits. *Arxiv preprint arXiv:1708.02130*, 2017.
- [MF16] Tomoyuki Morimae and Joseph F. Fitzsimons. Post hoc verification with a single prover. *Arxiv preprint arXiv:1603.06046*, 2016.
- [MNS15] Tomoyuki Morimae, Daniel Nagaj, and Norbert Schuch. Quantum proofs can be verified using only single qubit measurements, 2015.

- [MP11] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. Cryptology ePrint Archive, Report 2011/501, 2011. <http://eprint.iacr.org/2011/501>.
- [Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *STOC*, pages 333–342, 2009.
- [PRS17] Chris Peikert, Oded Regev, and Noah Stephens-Davidowitz. Pseudorandomness of ring-lwe for any ring and modulus. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 461–473. ACM, 2017.
- [Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing, STOC '05*, pages 84–93, New York, NY, USA, 2005. ACM.
- [RUV12] B. Reichardt, F. Unger, and U. Vazirani. A classical leash for a quantum system. *Arxiv preprint arXiv:1209.0448*, 2012.
- [Wik18] Wikipedia. Trace distance — Wikipedia, the free encyclopedia, 2018. [https://en.wikipedia.org/wiki/Trace_distance; accessed 7-Jan-2018].

A Extended Trapdoor Claw-Free Family from LWE

A.1 Learning with Errors Preliminaries

This background section on the learning with errors problem is taken directly from [BCM⁺18]. For a positive real B and positive integers q , the truncated discrete Gaussian distribution over \mathbb{Z}_q with parameter B is supported on $\{x \in \mathbb{Z}_q : \|x\| \leq B\}$ and has density

$$D_{\mathbb{Z}_q, B}(x) = \frac{e^{-\frac{\pi\|x\|^2}{B^2}}}{\sum_{x \in \mathbb{Z}_q, \|x\| \leq B} e^{-\frac{\pi\|x\|^2}{B^2}}} . \quad (180)$$

We note that for any $B > 0$, the truncated and non-truncated distributions have statistical distance that is exponentially small in B [Ban93, Lemma 1.5]. For a positive integer m , the truncated discrete Gaussian distribution over \mathbb{Z}_q^m with parameter B is supported on $\{x \in \mathbb{Z}_q^m : \|x\| \leq B\sqrt{m}\}$ and has density

$$\forall x = (x_1, \dots, x_m) \in \mathbb{Z}_q^m, \quad D_{\mathbb{Z}_q^m, B}(x) = D_{\mathbb{Z}_q, B}(x_1) \cdots D_{\mathbb{Z}_q, B}(x_m) . \quad (181)$$

Lemma A.1 *Let B be a positive real number and q, m be positive integers. Let $\mathbf{e} \in \mathbb{Z}_q^m$. The Hellinger distance between the distribution $D = D_{\mathbb{Z}_q^m, B}$ and the shifted distribution $D + \mathbf{e}$ satisfies*

$$H^2(D, D + \mathbf{e}) \leq 1 - e^{-\frac{2\pi\sqrt{m}\|\mathbf{e}\|}{B}} , \quad (182)$$

and the statistical distance between the two distributions satisfies

$$\|D - (D + \mathbf{e})\|_{TV}^2 \leq 2 \left(1 - e^{-\frac{2\pi\sqrt{m}\|\mathbf{e}\|}{B}} \right) . \quad (183)$$

Definition A.2 For a security parameter λ , let $n, m, q \in \mathbb{N}$ be integer functions of λ . Let $\chi = \chi(\lambda)$ be a distribution over \mathbb{Z} . The $\text{LWE}_{n,m,q,\chi}$ problem is to distinguish between the distributions $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e} \pmod{q})$ and (\mathbf{A}, \mathbf{u}) , where \mathbf{A} is uniformly random in $\mathbb{Z}_q^{n \times m}$, \mathbf{s} is a uniformly random row vector in \mathbb{Z}_q^n , \mathbf{e} is a row vector drawn at random from the distribution χ^m , and \mathbf{u} is a uniformly random vector in \mathbb{Z}_q^m . Often we consider the hardness of solving LWE for any function m such that m is at most a polynomial in $n \log q$. This problem is denoted $\text{LWE}_{n,q,\chi}$. When we write that we make the $\text{LWE}_{n,q,\chi}$ assumption, our assumption is that no quantum polynomial-time procedure can solve the $\text{LWE}_{n,q,\chi}$ problem with more than a negligible advantage in λ .

As shown in [Reg05, PRS17], for any $\alpha > 0$ such that $\sigma = \alpha q \geq 2\sqrt{n}$ the $\text{LWE}_{n,q,D_{\mathbb{Z}_q,\sigma}}$ problem, where $D_{\mathbb{Z}_q,\sigma}$ is the discrete Gaussian distribution, is at least as hard as approximating the shortest independent vector problem (SIVP) to within a factor of $\gamma = \tilde{O}(n/\alpha)$ in worst case dimension n lattices. This is proven using a quantum reduction. Classical reductions (to a slightly different problem) exist as well [Pei09, BLP⁺13] but with somewhat worse parameters. The best known (classical or quantum) algorithm for these problems run in time $2^{\tilde{O}(n/\log \gamma)}$. For our construction we assume hardness of the problem against a quantum polynomial-time adversary in the case that γ is a super polynomial function in n . This is a commonly used assumption in cryptography (for e.g. homomorphic encryption schemes such as [GSW13]).

We use two additional properties of the LWE problem. The first is that it is possible to generate LWE samples $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$ such that there is a trapdoor allowing recovery of \mathbf{s} from the samples.

Theorem A.3 (Theorem 5.1 in [MP11]) Let $n, m \geq 1$ and $q \geq 2$ be such that $m = \Omega(n \log q)$. There is an efficient randomized algorithm $\text{GENTRAP}(1^n, 1^m, q)$ that returns a matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ and a trapdoor $t_{\mathbf{A}}$ such that the distribution of \mathbf{A} is negligibly (in n) close to the uniform distribution. Moreover, there is an efficient algorithm INVERT that, on input $\mathbf{A}, t_{\mathbf{A}}$ and $\mathbf{A}\mathbf{s} + \mathbf{e}$ where $\|\mathbf{e}\| \leq q/(C_T \sqrt{n \log q})$ and C_T is a universal constant, returns \mathbf{s} and \mathbf{e} with overwhelming probability over $(\mathbf{A}, t_{\mathbf{A}}) \leftarrow \text{GENTRAP}$.

The second property is the existence of a “lossy mode” for LWE. The following definition is Definition 3.1 in [AKPW13].

Definition A.4 Let $\chi = \chi(\lambda)$ be an efficiently sampleable distribution over \mathbb{Z}_q . Define a lossy sampler $\tilde{\mathbf{A}} \leftarrow \text{LOSSY}(1^n, 1^m, 1^\ell, q, \chi)$ by $\tilde{\mathbf{A}} = \mathbf{B}\mathbf{C} + \mathbf{F}$, where $\mathbf{B} \leftarrow_U \mathbb{Z}_q^{m \times \ell}$, $\mathbf{C} \leftarrow_U \mathbb{Z}_q^{\ell \times n}$, $\mathbf{F} \leftarrow \chi^{m \times n}$.

Theorem A.5 (Lemma 3.2 in [AKPW13]) Under the $\text{LWE}_{\ell,q,\chi}$ assumption, the distribution of $\tilde{\mathbf{A}} \leftarrow \text{LOSSY}(1^n, 1^m, 1^\ell, q, \chi)$ is computationally indistinguishable from $\mathbf{A} \leftarrow_U \mathbb{Z}_q^{m \times n}$.

A.2 Parameters

We will use the same parameters used in [BCM⁺18]. Let λ be the security parameter. All other parameters are functions of λ . Let $q \geq 2$ be a prime integer. Let $\ell, n, m, w \geq 1$ be polynomially bounded functions of λ and B_L, B_V, B_P be positive integers such that the following conditions hold:

1. $n = \Omega(\ell \log q)$ and $m = \Omega(n \log q)$,
2. $w = n \lceil \log q \rceil$,
3. $B_P = \frac{q}{2C_T \sqrt{mn \log q}}$, for C_T the universal constant in Theorem A.3, (184)
4. $2\sqrt{n} \leq B_L < B_V < B_P$,
5. The ratios $\frac{B_P}{B_V}$ and $\frac{B_V}{B_L}$ are both super-polynomial in λ .

Given a choice of parameters satisfying all conditions above, we describe the function family \mathcal{F}_{LWE} (taken from [BCM⁺18]). Let $\mathcal{X} = \mathbb{Z}_q^n$ and $\mathcal{Y} = \mathbb{Z}_q^m$. The key space is $\mathcal{K}_{\mathcal{F}_{\text{LWE}}} = \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$. For $b \in \{0, 1\}$, $x \in \mathcal{X}$ and key $k = (\mathbf{A}, \mathbf{A}s + \mathbf{e})$, the density $f_{k,b}(x)$ is defined as

$$\forall y \in \mathcal{Y}, \quad (f_{k,b}(x))(y) = D_{\mathbb{Z}_q^m, B_P}(y - \mathbf{A}x - b \cdot \mathbf{A}s), \quad (185)$$

where the definition of $D_{\mathbb{Z}_q^m, B_P}$ is given in (180). Note that $f_{k,b}$ is well-defined given k , as for our choice of parameters k uniquely specifies s . The following theorem was proven in [BCM⁺18]:

Theorem 22 [BCM⁺18] *For any choice of parameters satisfying the conditions (184), the function family \mathcal{F}_{LWE} is a noisy trapdoor claw-free family under the hardness assumption $\text{LWE}_{\ell, q, D_{\mathbb{Z}_q, B_L}}$.*

We will prove the following theorem:

Theorem A.6 *For any choice of parameters satisfying the conditions (184), the function family \mathcal{F}_{LWE} is an extended trapdoor claw-free family under the hardness assumption $\text{LWE}_{\ell, q, D_{\mathbb{Z}_q, B_L}}$.*

We begin by providing a trapdoor injective family and then use this family to show that \mathcal{F}_{LWE} is an extended trapdoor claw-free function family.

A.3 Trapdoor Injective Family from LWE

We now describe the trapdoor injective family \mathcal{G}_{LWE} . Let $\mathcal{X} = \mathbb{Z}_q^n$ and $\mathcal{Y} = \mathbb{Z}_q^m$. The key space is $\mathcal{K}_{\mathcal{G}_{\text{LWE}}} = \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$. For $b \in \{0, 1\}$, $x \in \mathcal{X}$ and key $k = (\mathbf{A}, \mathbf{u})$, the density $g_{k,b}(x)$ is defined as

$$\forall y \in \mathcal{Y}, \quad (g_{k,b}(x))(y) = D_{\mathbb{Z}_q^m, B_P}(y - \mathbf{A}x - b \cdot \mathbf{u}), \quad (186)$$

where the definition of $D_{\mathbb{Z}_q^m, B_P}$ is given in (180). The three properties required for a trapdoor injective family, as specified in Definition 4.2, are verified in the following subsections, providing a proof of the following theorem.

Theorem A.7 *For any choice of parameters satisfying the conditions (184), the function family \mathcal{G}_{LWE} is a trapdoor injective family under the hardness assumption $\text{LWE}_{\ell, q, D_{\mathbb{Z}_q, B_L}}$.*

A.3.1 Efficient Function Generation

$\text{GEN}_{\mathcal{G}_{\text{LWE}}}$ is defined as follows. First, the procedure samples a random $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, together with trapdoor information $t_{\mathbf{A}}$. This is done using the procedure $\text{GENTRAP}(1^n, 1^m, q)$ from Theorem A.3. The trapdoor allows the evaluation of an inversion algorithm INVERT that, on input \mathbf{A} , $t_{\mathbf{A}}$ and $b = \mathbf{A}s + \mathbf{e}$ returns s and \mathbf{e} as long as $\|\mathbf{e}\| \leq \frac{q}{C_T \sqrt{n \log q}}$. Moreover, the distribution on matrices \mathbf{A} returned by GENTRAP is negligibly close to the uniform distribution on $\mathbb{Z}_q^{m \times n}$.

Next, the sampling procedure selects $\mathbf{u} \in \mathbb{Z}_q^m$ uniformly at random. By using the trapdoor $t_{\mathbf{A}}$, the sampling procedure checks if there exist s, \mathbf{e} such that $\|\mathbf{e}\| \leq \frac{q}{C_T \sqrt{n \log q}}$ and $\mathbf{A}s + \mathbf{e} = \mathbf{u}$. If so, the sampling algorithm discards \mathbf{u} and samples \mathbf{u} again. Since \mathbf{u} is discarded with negligible probability, the distribution over \mathbf{u} is negligibly close to the uniform distribution. $\text{GEN}_{\mathcal{G}_{\text{LWE}}}$ returns $k = (\mathbf{A}, \mathbf{u})$ and $t_k = t_{\mathbf{A}}$.

A.3.2 Trapdoor Injective Functions

It follows from (186) and the definition of the distribution $D_{\mathbb{Z}_q^m, B_P}$ in (180) that for any key $k = (\mathbf{A}, \mathbf{u}) \in \mathcal{K}_{\mathcal{G}_{\text{LWE}}}$ and for all $x \in \mathcal{X}$,

$$\text{SUPP}(g_{k,0}(x)) = \{y = \mathbf{A}x + \mathbf{e}_0 \mid \|\mathbf{e}_0\| \leq B_P \sqrt{m}\}, \quad (187)$$

$$\text{SUPP}(g_{k,1}(x)) = \{y = \mathbf{A}x + \mathbf{e}_0 + \mathbf{u} \mid \|\mathbf{e}_0\| \leq B_P \sqrt{m}\}. \quad (188)$$

Since there do not exist \mathbf{s}, \mathbf{e} such that $\|\mathbf{e}\| \leq \frac{q}{C_T \sqrt{n \log q}}$ and $\mathbf{u} = \mathbf{A}\mathbf{s} + \mathbf{e}$, the intersection $\text{SUPP}(g_{k,0}(x)) \cap \text{SUPP}(g_{k,1}(x))$ is empty as long as

$$B_P \leq \frac{q}{2C_T \sqrt{mn \log q}}. \quad (189)$$

The procedure $\text{INV}_{\mathcal{G}_{\text{LWE}}}$ takes as input the trapdoor $t_{\mathbf{A}}$ and $y \in \mathcal{Y}$. It first runs the algorithm INVERT on input y . If INVERT outputs $\mathbf{s}_0, \mathbf{e}_0$ such that $y = \mathbf{A}\mathbf{s}_0 + \mathbf{e}_0$ and $\|\mathbf{e}_0\| \leq B_P \sqrt{m}$, the procedure $\text{INV}_{\mathcal{F}_{\text{LWE}}}$ outputs $(0, \mathbf{s}_0)$. Otherwise, it runs the algorithm INVERT on input $y - \mathbf{u}$ to obtain $\mathbf{s}_0, \mathbf{e}_0$ and outputs $(1, \mathbf{s}_0)$ if $y - \mathbf{u} = \mathbf{A}\mathbf{s}_0 + \mathbf{e}_0$. Using Theorem A.3, this procedure returns the unique correct outcome provided $y - b \cdot \mathbf{u} = \mathbf{A}\mathbf{s}_0 + \mathbf{e}_0$ for some \mathbf{e}_0 such that $\|\mathbf{e}_0\| \leq \frac{q}{C_T \sqrt{n \log q}}$. Due to (189), this condition is satisfied for all $y \in \text{SUPP}(f_{k,b}(x))$.

A.3.3 Efficient Range Superposition

The procedures $\text{CHK}_{\mathcal{G}_{\text{LWE}}}$ and $\text{SAMP}_{\mathcal{G}_{\text{LWE}}}$ are the same as the procedures $\text{CHK}_{\mathcal{F}_{\text{LWE}}}$ and $\text{SAMP}_{\mathcal{F}_{\text{LWE}}}$.

A.4 Injective Invariance

We now show that \mathcal{F}_{LWE} as given in (185) is injective invariant with respect to \mathcal{G}_{LWE} (see Definition 4.3). To show this, we need to show that for all BQP attackers \mathcal{A} , the distributions produced by $\text{GEN}_{\mathcal{F}_{\text{LWE}}}$ and $\text{GEN}_{\mathcal{G}_{\text{LWE}}}$ are computationally indistinguishable. This is equivalent to proving the hardness of LWE (as defined in Definition A.2) with a binary secret, as seen in the following lemma:

Lemma A.8 *Assume a choice of parameters satisfying the conditions (184). Assume the hardness assumption $\text{LWE}_{\ell, q, D_{\mathbb{Z}_q}, B_L}$ holds. Then the distributions*

$$\mathcal{D}_0 = ((\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e}) \leftarrow \text{GEN}_{\mathcal{F}_{\text{LWE}}}(1^\lambda)) \quad (190)$$

and

$$\mathcal{D}_1 = ((\mathbf{A}, \mathbf{u}) \leftarrow \text{GEN}_{\mathcal{G}_{\text{LWE}}}(1^\lambda)) \quad (191)$$

are computationally indistinguishable.

The hardness of LWE with a binary secret is well studied, and the above lemma is implied by several results, starting with [GKPV10] and improved in [BLP⁺13]. To be precise, it is also immediately implied by Theorem B.5 of [AKPW13].

A.5 Extended Trapdoor Claw-Free Family

We have already shown that \mathcal{F}_{LWE} is injective invariant. To show that \mathcal{F}_{LWE} is an extended trapdoor claw-free family, we now prove the second condition (the hardcore bit condition) of Definition 4.4. First recall (from [BCM⁺18]) that $\mathcal{X} = \mathbb{Z}_q^n$, $w = n \lceil \log q \rceil$ and $J : \mathcal{X} \rightarrow \{0, 1\}^w$ is the map such that $J(x)$ returns the binary representation of $x \in \mathcal{X}$. The key point, which we prove in Lemma A.10, is that the inner product appearing in the definition of $H'_{k,d}$ (in condition 2 of Definition 4.4) is equal to the inner product $\hat{d} \cdot s$ (for $\hat{d} \in \{0, 1\}^n$) if $d = J(\hat{d})$. We first show in Lemma A.9 that producing an inner product $\hat{d} \cdot s$ is computationally difficult given $\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e}$. Next, in Lemma A.10, we use Lemma A.9 to prove condition 2 of Definition 4.4.

Lemma A.9 *Assume a choice of parameters satisfying the conditions (184). Assume the hardness assumption $\text{LWE}_{\ell, q, D_{\mathbb{Z}_q}, B_L}$ holds. For all $\hat{d} \in \{0, 1\}^n \setminus \{0^n\}$, the distributions*

$$\mathcal{D}_0 = ((\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e}) \leftarrow \text{GEN}_{\mathcal{F}_{\text{LWE}}}(1^\lambda), \hat{d} \cdot s \bmod 2) \quad (192)$$

and

$$\mathcal{D}_1 = ((\mathbf{A}, \mathbf{A}s + \mathbf{e}) \leftarrow \text{GEN}_{\mathcal{F}_{\text{LWE}}}(1^\lambda), r), \quad (193)$$

where $r \leftarrow_{\mathcal{U}} \{0, 1\}$, are computationally indistinguishable.

Proof: This proof is a simpler version of the proof of Lemma 27 in [BCM⁺18]. We first transition from \mathcal{D}_0 to the following computationally indistinguishable distribution:

$$\mathcal{D}^{(1)} = (\mathbf{BC} + \mathbf{F}, \mathbf{BC}s + \mathbf{e}, \hat{d} \cdot s \bmod 2) \quad (194)$$

where $\tilde{\mathbf{A}} = \mathbf{BC} + \mathbf{F} \leftarrow \text{LOSSY}(1^n, 1^m, 1^\ell, q, D_{\mathbb{Z}_q, B_L})$ is sampled from a lossy sampler. The transition from \mathcal{D}_0 to $\mathcal{D}^{(1)}$ is the same as the transition from (47) to (50) in the proof of Lemma 27 in [BCM⁺18]: the first step is to replace \mathbf{A} with a lossy matrix $\tilde{\mathbf{A}}$ to obtain a computationally indistinguishable distribution and the second step is to remove the term $\mathbf{F}s$ from the lossy LWE sample $\tilde{\mathbf{A}}s + \mathbf{e}$. Next, we apply Lemma 23 ([BCM⁺18]) to $\mathcal{D}^{(1)}$ to replace $\hat{d} \cdot s \bmod 2$ with a uniformly random bit r , resulting in the following statistically indistinguishable distribution:

$$\mathcal{D}^{(2)} = (\mathbf{BC} + \mathbf{F}, \mathbf{BC}s + \mathbf{e}, r) \quad (195)$$

Computational indistinguishability between $\mathcal{D}^{(2)}$ and \mathcal{D}_1 follows similarly to between $\mathcal{D}^{(1)}$ and \mathcal{D}_0 . \square

We now show that the second condition of Definition 4.4 holds:

Lemma A.10 *Assume a choice of parameters satisfying the conditions (184). Assume the hardness assumption $\text{LWE}_{\ell, q, D_{\mathbb{Z}_q, B_L}}$ holds. Let $s \in \{0, 1\}^n$ and for $d \in \{0, 1\}^w$ let ²*

$$H'_{s,d} = \{d \cdot (J(x) \oplus J(x - (-1)^b \mathbf{s}) \mid x \in \mathcal{X}\}, \quad (196)$$

Then for all $\hat{d} \in \{0, 1\}^n$ and for any quantum polynomial-time procedure

$$\mathcal{A} : \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m \rightarrow \{0, 1\}$$

there exists a negligible function $\mu(\cdot)$ such that

$$\left| \Pr_{(\mathbf{A}, \mathbf{A}s + \mathbf{e}) \leftarrow \text{GEN}_{\mathcal{F}_{\text{LWE}}}(1^\lambda)} [\mathcal{A}(\mathbf{A}, \mathbf{A}s + \mathbf{e}) \in H'_{s, J(\hat{\mathbf{d}})}] - \frac{1}{2} \right| \leq \mu(\lambda). \quad (197)$$

Proof: This proof is very similar to the proof of Lemma 28 in [BCM⁺18]. The proof is by contradiction. Assume that there exists $\hat{d} \in \{0, 1\}^n$ and a quantum polynomial-time procedure \mathcal{A} such that the left-hand side of (197) is at least some non-negligible function $\eta(\lambda)$. We derive a contradiction by showing that for \hat{d} , the two distributions \mathcal{D}_0 and \mathcal{D}_1 in Lemma A.9 are computationally distinguishable, giving a contradiction.

Let $(\mathbf{A}, \mathbf{A}s + \mathbf{e}) \leftarrow \text{GEN}_{\mathcal{F}_{\text{LWE}}}(1^\lambda)$. To link \mathcal{A} to the distributions in Lemma A.9 we relate the inner product condition in (196) to the inner product $\hat{d} \cdot s$ appearing in (192). This is based on the following claim.

Claim A.11 *For all $b \in \{0, 1\}$, $x \in \mathcal{X}$, $\hat{d} \in \{0, 1\}^n$ and $s \in \{0, 1\}^n$ the following equality holds:*

$$J(\hat{\mathbf{d}}) \cdot (J(x) \oplus J(x - (-1)^b \mathbf{s})) = \hat{d} \cdot s. \quad (198)$$

²We write the set as $H'_{s,d}$ instead of $H'_{k,d}$ to emphasize the dependence on s .

Proof: We do the proof in case $n = 1$ and $w = \lceil \log q \rceil$, as the case of general n follows by linearity. In this case s is a single bit. If $s = 0$ then both sides of (198) evaluate to zero, so the equality holds trivially. If $s = 1$, then the least significant bit of $J(x) \oplus J(x - (-1)^b \mathbf{s})$ is s and the least significant bit of $J(\hat{\mathbf{d}}) = \hat{d}$. Since the remaining $w - 1$ bits of $J(\hat{\mathbf{d}})$ are 0, the claim follows. \square

To conclude we construct a distinguisher \mathcal{A}' between the distributions \mathcal{D}_0 and \mathcal{D}_1 in Lemma A.9. Consider two possible distinguishers, \mathcal{A}'_u for $u \in \{0, 1\}$. Given a sample $((\mathbf{A}, \mathbf{As} + \mathbf{e}), t)$, \mathcal{A}'_u computes $c = \mathcal{A}(\mathbf{A}, \mathbf{As} + \mathbf{e})$ and returns 0 if $c = t \oplus u$, and 1 otherwise. The sum of the advantages of \mathcal{A}'_0 and \mathcal{A}'_1 is:

$$\sum_{u \in \{0, 1\}} \left| \Pr_{((\mathbf{A}, \mathbf{As} + \mathbf{e}), \hat{d} \cdot s) \leftarrow \mathcal{D}_0} [\mathcal{A}'_u((\mathbf{A}, \mathbf{As} + \mathbf{e}), \hat{d} \cdot s) = 0] - \Pr_{((\mathbf{A}, \mathbf{As} + \mathbf{e}), r) \leftarrow \mathcal{D}_1} [\mathcal{A}'_u((\mathbf{A}, \mathbf{As} + \mathbf{e}), r) = 0] \right| \quad (199)$$

$$\begin{aligned} &= \sum_{u \in \{0, 1\}} \left| \Pr_{(\mathbf{A}, \mathbf{As} + \mathbf{e}) \leftarrow \text{GEN}_{\mathcal{F}_{\text{LWE}}}(1^\lambda)} [\mathcal{A}(\mathbf{A}, \mathbf{As} + \mathbf{e}) = \hat{d} \cdot s \oplus u] - \Pr_{((\mathbf{A}, \mathbf{As} + \mathbf{e}), r) \leftarrow \mathcal{D}_1} [\mathcal{A}(\mathbf{A}, \mathbf{As} + \mathbf{e}) = r \oplus u] \right| \\ &= \sum_{u \in \{0, 1\}} \left| \Pr_{(\mathbf{A}, \mathbf{As} + \mathbf{e}) \leftarrow \text{GEN}_{\mathcal{F}_{\text{LWE}}}(1^\lambda)} [\mathcal{A}(\mathbf{A}, \mathbf{As} + \mathbf{e}) = \hat{d} \cdot s \oplus u] - \frac{1}{2} \right| \end{aligned} \quad (200)$$

$$\begin{aligned} &\geq \left| \Pr_{(\mathbf{A}, \mathbf{As} + \mathbf{e}) \leftarrow \text{GEN}_{\mathcal{F}_{\text{LWE}}}(1^\lambda)} [\mathcal{A}(\mathbf{A}, \mathbf{As} + \mathbf{e}) = \hat{d} \cdot s] - \Pr_{(\mathbf{A}, \mathbf{As} + \mathbf{e}) \leftarrow \text{GEN}_{\mathcal{F}_{\text{LWE}}}(1^\lambda)} [\mathcal{A}(\mathbf{A}, \mathbf{As} + \mathbf{e}) = \hat{d} \cdot s \oplus 1] \right| \\ &\geq 2 \left| \Pr_{(\mathbf{A}, \mathbf{As} + \mathbf{e}) \leftarrow \text{GEN}_{\mathcal{F}_{\text{LWE}}}(1^\lambda)} [\mathcal{A}(\mathbf{A}, \mathbf{As} + \mathbf{e}) = \hat{d} \cdot s] - \frac{1}{2} \right| \end{aligned} \quad (201)$$

By Claim A.11, we can replace $\hat{d} \cdot s$ with $J(\hat{\mathbf{d}}) \cdot (J(x) \oplus J(x - (-1)^b \mathbf{s}))$ to obtain:

$$= 2 \left| \Pr_{(\mathbf{A}, \mathbf{As} + \mathbf{e}) \leftarrow \text{GEN}_{\mathcal{F}_{\text{LWE}}}(1^\lambda)} [\mathcal{A}(\mathbf{A}, \mathbf{As} + \mathbf{e}) \in H'_{s, J(\hat{\mathbf{d}})}] - \frac{1}{2} \right| \quad (202)$$

$$\geq 2\eta(\lambda) \quad (203)$$

Therefore, at least one of \mathcal{A}'_0 or \mathcal{A}'_1 must successfully distinguish between \mathcal{D}_0 and \mathcal{D}_1 with advantage at least η , a contradiction with the statement of Lemma A.9. \square

B Pauli Twirl

Here we prove the Z Pauli twirl lemma, copied below for reference:

Lemma 3.5 *For a CPTP map with Kraus operators $\{B_\tau\}_\tau$, the following two CPTP maps are equal:*

$$\left\{ \frac{1}{\sqrt{2}} (Z^r \otimes \mathcal{I}) B_\tau (Z^r \otimes \mathcal{I}) \right\}_{r \in \{0, 1\}, \tau} = \{(X^x \otimes \mathcal{I}) B'_{x, \tau}\}_{x \in \{0, 1\}, \tau} \quad (204)$$

where $B_\tau = \sum_{x, z \in \{0, 1\}} X^x Z^z \otimes B_{xz\tau}$ and the CPTP map $\{B'_{x, \tau}\}_{x \in \{0, 1\}, \tau}$ is equal to $B'_{x, \tau} = \sum_{z \in \{0, 1\}} Z^z \otimes B_{xz\tau}$.

Proof: To prove the lemma, we show that applying either of the two CPTP maps in equation (204) on an arbitrary density matrix ρ results in the same state. We begin with the CPTP map on the left of (204):

$$\frac{1}{2} \sum_{r \in \{0, 1\}, \tau} (Z^r \otimes \mathcal{I}) B_\tau (Z^r \otimes \mathcal{I}) \rho (Z^r \otimes \mathcal{I}) B_\tau^\dagger (Z^r \otimes \mathcal{I})^\dagger \quad (205)$$

Using the fact that $B_\tau = \sum_{x,z \in \{0,1\}} X^x Z^z \otimes B_{xz\tau}$, we can rewrite the expression from (205) as:

$$\frac{1}{2} \sum_{\substack{r \in \{0,1\}, \tau \\ x,z,x',z' \in \{0,1\}}} (Z^r X^x Z^z Z^r \otimes B_{xz\tau}) \rho(Z^r X^{x'} Z^{z'} Z^r \otimes B_{x'z'\tau})^\dagger \quad (206)$$

Next, we use the anti commutation properties of Pauli operators (by commuting Z^r with both X^x and $X^{x'}$) to obtain the following state:

$$\frac{1}{2} \sum_{\substack{r \in \{0,1\}, \tau \\ x,z,x',z' \in \{0,1\}}} (-1)^{r \cdot (x \oplus x')} (X^x Z^z \otimes B_{xz\tau}) \rho(X^{x'} Z^{z'} \otimes B_{x'z'\tau})^\dagger \quad (207)$$

At this point, we can sum over r to obtain $x = x'$, resulting in the following expression:

$$\sum_{x,z,z' \in \{0,1\}} (X^x Z^z \otimes B_{xz\tau}) \rho(X^x Z^{z'} \otimes B_{x'z'\tau})^\dagger \quad (208)$$

$$= \sum_{x \in \{0,1\}, \tau} (X^x \otimes \mathcal{I}) B'_{x,\tau} \rho((X^x \otimes \mathcal{I}) B'_{x,\tau})^\dagger \quad (209)$$

□

We now prove Corollary 3.6, which we copy below for reference:

Corollary 3.6 *For a CPTP map with Kraus operators $\{B_\tau\}$, the following two CPTP maps are equal:*

$$\left\{ \frac{1}{\sqrt{2}} (|b\rangle \langle b| H Z^r \otimes \mathcal{I}) B_\tau (Z^r \otimes \mathcal{I}) \right\}_{b,r \in \{0,1\}, \tau} = \{(|b\rangle \langle b| H \otimes \mathcal{I}) B'_{x,\tau}\}_{b,x \in \{0,1\}, \tau} \quad (210)$$

where $B_\tau = \sum_{x,z \in \{0,1\}} X^x Z^z \otimes B_{xz\tau}$ and the CPTP map $\{B'_{x,\tau}\}_{x \in \{0,1\}, \tau}$ is equal to $B'_{x,\tau} = \sum_{z \in \{0,1\}} Z^z \otimes B_{xz\tau}$.

Proof: We begin by applying Lemma 3.5 to obtain the following equality:

$$\left\{ \frac{1}{\sqrt{2}} (|b\rangle \langle b| H Z^r \otimes \mathcal{I}) B_\tau (Z^r \otimes \mathcal{I}) \right\}_{b,r \in \{0,1\}, \tau} = \{(|b\rangle \langle b| H X \otimes \mathcal{I}) B'_{x,\tau}\}_{b,x \in \{0,1\}, \tau} \quad (211)$$

To prove the corollary, we show that

$$\{(|b\rangle \langle b| H X \otimes \mathcal{I}) B'_{x,\tau}\}_{b,x \in \{0,1\}, \tau} = \{(|b\rangle \langle b| H \otimes \mathcal{I}) B'_{x,\tau}\}_{x \in \{0,1\}, \tau} \quad (212)$$

This is straightforward and only requires one intermediate step:

$$\{(|b\rangle \langle b| H X \otimes \mathcal{I}) B'_{x,\tau}\}_{b,x \in \{0,1\}, \tau} = \{(|b\rangle \langle b| Z H \otimes \mathcal{I}) B'_{x,\tau}\}_{x \in \{0,1\}, \tau} \quad (213)$$

The second CPTP map in (213) is equal to the second CPTP map in (212), since a Z operator applied prior to standard basis measurement has no effect, and can be replaced with the identity operator. □