Quantum cryptography with classical communication

Parallel remote state preparation for copy-protection, verification, and more

Alexandru Gheorghiu[1], Tony Metger[*2], and Alexander Poremba[3]

[1]Institute for Theoretical Studies, ETH Zürich, 8092 Zürich, Switzerland
[2]Institute for Theoretical Physics, ETH Zürich, 8093 Zürich, Switzerland
[3]Computing and Mathematical Sciences, California Institute of Technology, CA 91125, United States

Abstract

Quantum mechanical effects have enabled the construction of cryptographic primitives that are impossible classically. For example, quantum copy-protection allows for a program to be encoded in a quantum state in such a way that the program can be evaluated, but not copied. Many of these cryptographic primitives are two-party protocols, where one party, Bob, has full quantum computational capabilities, and the other party, Alice, is only required to send random BB84 states to Bob. In this work, we show how such protocols can *generically* be converted to ones where Alice is fully classical, assuming that Bob cannot efficiently solve the LWE problem. In particular, this means that all communication between (classical) Alice and (quantum) Bob is classical, yet they can still make use of cryptographic primitives that would be impossible if both parties were classical. We apply this conversion procedure to obtain quantum cryptographic protocols with classical communication for unclonable encryption, copy-protection, computing on encrypted data, and verifiable blind delegated computation.

The key technical ingredient for our result is a protocol for *classically-instructed parallel remote state preparation of BB84 states*. This is a multi-round protocol between (classical) Alice and (quantum polynomial-time) Bob that allows Alice to certify that Bob must have prepared n uniformly random BB84 states (up to a change of basis on his space). Furthermore, Alice knows which specific BB84 states Bob has prepared, while Bob himself does not. Hence, the situation at the end of this protocol is (almost) equivalent to one where Alice sent n random BB84 states to Bob. This allows us to replace the step of preparing and sending BB84 states in existing protocols by our remote-state preparation protocol in a generic and modular way.

---

[*]Corresponding author: tmetger@ethz.ch

Contents

1 Introduction

A central distinction between classical and quantum information is that a classical string can always be copied, but a quantum state cannot: the *no-cloning theorem* states that there cannot exist a procedure that produces the state $\rho \otimes \rho$ when given as input an arbitrary quantum state ρ [WZ82]. The first cryptographic protocols that made use of the no-cloning theorem were Wiesner's proposal to use quantum states as unforgeable banknotes [Wie83] and Bennett and Brassard's protocol for information-theoretically secure quantum key-distribution (the BB84 QKD protocol) [BB84]. These protocols rely on the idea of a *conjugate coding scheme*: classical information can be encoded into a quantum state in (at least) two incompatible bases, most commonly the standard basis $\{|0\rangle, |1\rangle\}$ and the Hadamard basis $\{|+\rangle, |-\rangle\}$, where $|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$. These four states are commonly referred to as *BB84 states*. If we encode a bit $b \in \{0, 1\}$ as either $|b\rangle$ or $|(-)^b\rangle = \frac{1}{\sqrt{2}}(|0\rangle + (-1)^b|1\rangle)$, then an adversary who does not know which basis we chose for the encoding cannot create a copy of this quantum state. Furthermore, if the adversary tries to measure the state, with probability $1/2$ they will choose the "wrong" measurement basis, which disturbs the state and means that the adversary's tampering can be detected.

There is an important conceptual difference between the BB84 protocol and Wiesner's quantum money scheme. The former addresses the problem of key-distribution, which is a task that can also be achieved classically under computational assumptions using public-key cryptography [DH76]. In contrast, Wiesner's quantum money scheme achieves a functionality which is entirely impossible classically, even under computational assumptions. Recently there has been renewed interest in this latter kind of application, i.e. to use BB84 states to construct quantum cryptographic primitives that have no classical analogue. Perhaps the most striking example of this is the idea of *quantum copy-protection* [Aar09]. Suppose that a vendor has created a piece of software (viewed as a function that maps some input to some output) and wants to allow a user to run it (i.e. to evaluate the function), while preventing the user from producing additional "pirated" copies of the original software. Clearly, this is impossible classically: any piece of software is specified by a string of symbols, which can easily be copied. Surprisingly, it has been shown that it is possible to encode certain narrow classes of functions in the form of a quantum state in such a way that a user can evaluate the function without being able to copy it [CMP20].

Copy-protection and many related protocols require only limited quantum capabilities from one party, e.g. the *vendor* in the case of copy-protection: they only need to prepare random BB84 states and send them to the other party (e.g. the *user* in copy-protection), who has full quantum computational capabilities. In particular, this requires a quantum channel between the two parties to send the BB84 states. The purpose of this paper is to show that such protocols, where one party's quantum operations are limited to preparing and sending random BB84 states, can be converted into protocols where that party is *fully classical*. This *dequantises* such protocols in the sense that all communication becomes classical. To achieve this, we need to construct a protocol between a *classical verifier* and a *computationally bounded quantum prover* that achieves the same outcome as if the verifier had prepared and sent random BB84 states to the prover. We call this task *classically-instructed parallel remote state preparation of BB84 states*, or *parallel RSP* for short. Our protocol builds on techniques introduced in [Mah18, BCM$^+$18, GV19] that allow the verifier to use post-quantum cryptography to constrain the actions of an untrusted (but computationally bounded) prover. Proving soundness for this parallel RSP protocol is the main technical result of our work. We then use this result to dequantise a number of cryptographic protocols, namely unclonable quantum encryption, quantum copy-protection, quantum computing on encrypted data and blind verification of quantum computation. As an additional contribution, for copy-protection we improve upon a previous result [CMP20] by removing the use of a quantum random oracle for a certain class of copy-protected functions.

1.1 Main results

We start by first describing the soundness guarantee achieved by our parallel RSP protocol. Intuitively, the goal of our protocol is to guarantee that the prover has prepared a quantum state of the form $H^{\theta_1}|v_1\rangle\langle v_1|H^{\theta_1} \otimes \ldots \otimes H^{\theta_n}|v_n\rangle\langle v_n|H^{\theta_n}$, where $\vec{v}, \vec{\theta} \in \{0,1\}^n$. Additionally, the prover should not have any information about $\vec{v}$ and $\vec{\theta}$ beyond what is contained in its BB84 states, while the verifier should know both $\vec{v}$ and $\vec{\theta}$. Our protocol achieves a guarantee of this kind assuming the quantum-intractability of the *Learning with Errors* (LWE) problem introduced by Regev [Reg09]. Our main result is the following (see Theorem 4.33 for the corresponding formal statement):

Theorem 1.1 (Informal). *There exists an interactive protocol between a classical verifier and a computationally bounded quantum prover such that the following holds assuming the quantum-intractability of LWE. For any number n of BB84 states and any basis choice $\vec{\theta} \in \{0,1\}^n$, the verifier and prover's final joint state σ_{WP} conditioned on the prover being accepted in the protocol satisfies*

$$p_{\text{success}} V \sigma_{WP} V^\dagger \approx_{1/\text{poly}(n)} \frac{1}{2^n} \sum_{\vec{v} \in \{0,1\}^n} |v\rangle\langle v|_W \otimes \left(H^{\theta_1}|v_1\rangle\langle v_1|H^{\theta_1} \otimes \ldots \otimes H^{\theta_n}|v_n\rangle\langle v_n|H^{\theta_n} \right) \otimes \alpha_{P'}.$$

Here, W is a system in the verifier's possession, P is in the prover's possession, V is an isometry from P to QP' that is independent of $\vec{v}$ and $\vec{\theta}$, $\alpha_{P'}$ is some additional (subnormalised) state independent of $\vec{v}$ and $\vec{\theta}$, and p_{success} is the prover's success probability in the protocol.

We make two remarks regarding this security guarantee. Firstly, the theorem makes a statement about the joint state of the verifier's system W and the prover's system P after applying an isometry V that *only acts on the prover's space*. This additional isometry is unavoidable: it represents the prover's freedom to use any basis of its choice on its space. Hence, we cannot guarantee that the prover prepares BB84 states (in the standard basis), only that it prepares BB84 states up to a change of basis. However, crucially this change of basis is *independent of which BB84 state was supposed to be prepared*, i.e., V is independent of $\vec{v}$ and $\vec{\theta}$. Put differently, the theorem guarantees that the prover prepares one of 4^n possible states whose relation to each other is the same as the relation between the 4^n BB84 states. This does not affect the utility of the prover's state for applications. In fact, this freedom also exists if the verifier sent n BB84 states to the prover via a quantum channel: the prover could apply an isometry V to these states immediately upon receipt, but the security of any application using the BB84 states is not impacted by this.

Secondly, the theorem holds *for any* basis choice $\vec{\theta}$, but *on average* over the values $\vec{v}$. In other words, in the protocol, the verifier gets to choose the bases at will, but the values will be uniformly random and cannot be chosen by the verifier. Furthermore, the only dependence on $\vec{v}$ and $\vec{\theta}$ in the prover's state is via the BB84 states. This means that the protocol forces the prover to prepare these states "blindly", i.e., the prover does not know which BB84 states were actually prepared. In contrast, the verifier does know, because they chose $\vec{\theta}$ and are in possession of the system W, which contains information about $\vec{v}$. This asymmetry of knowledge about the prover's state is the same as what is achieved by preparing and sending BB84 states through a quantum channel and is crucial for applications.

Having introduced our parallel RSP theorem, we can turn to its cryptographic applications. We consider various cryptographic primitives that have previously been defined and constructed in a setting where one party sends random BB84 states to the other. For each primitive, we give a formal definition of the "classical-client version" and show that this definition can be satisfied using our parallel RSP protocol as a building block. Since our parallel RSP protocols relies on the LWE assumption, so do the dequantised protocols we present here. Furthermore, Theorem 1.1 only guarantees the preparation of BB84 states

up to an inverse polynomial error, so as a result, the dequantised protocols only have inverse polynomial security (see Section 1.4 for a discussion of this point). Some of these primitives have previously been dequantised using an application-specific approach (and similarly relying on computational assumptions) [GV19, CCKW19, RS19, HMNY21, KNY21]; in contrast, our approach is generic and simply uses RSP to replace the sending of BB84 states. We give a short overview of the different applications and refer to Part II for details.

Unclonable quantum encryption. As a first application of our parallel RSP protocol, we consider the notion of *unclonable quantum encryption*. This cryptographic functionality was coined by Gottesman [Got02] and then formalised by Broadbent and Lord [BL19]. In a private-key unclonable quantum encryption scheme, a classical message is encrypted into a quantum state (the *quantum ciphertext*) with the following property: given only a single quantum ciphertext, it is impossible to create two states that can later both be decrypted with access to the private key. Broadbent and Lord gave an information-theoretic unclonable encryption scheme based on conjugate coding: a plaintext $\vec{m} \in \{0,1\}^n$ is encrypted with a randomly chosen secret key $k = (\vec{r}, \vec{\theta}) \xleftarrow{\$} \{0,1\}^n \times \{0,1\}^n$ into the quantum ciphertext given by

$$\mathsf{Enc}_k(\vec{m}) = \bigotimes_{i=1}^{n} H^{\theta_i} |r_i \oplus m_i\rangle\langle r_i \oplus m_i| H^{\theta_i}. \tag{1.1}$$

To decrypt using the secret key $k = (\vec{r}, \vec{\theta})$, one applies Hadamard operations $H^{\theta_1} \otimes \cdots \otimes H^{\theta_n}$ to the state $\mathsf{Enc}_k(\vec{m})$, measures in the computational basis with outcome $\vec{x} \in \{0,1\}^\lambda$, and outputs $\vec{m}' = \vec{x} \oplus \vec{r}$. The fact that this scheme is unclonable is a direct consequence of the *monogamy of entanglement* [BL19, TFKW13].

To dequantise this protocol, we consider a scenario in which a classical client $\mathcal{C}$ wishes to delegate an unclonable ciphertext to a quantum receiver $\mathcal{R}$. As a first step, $\mathcal{C}$ and $\mathcal{R}$ run our parallel RSP protocol to delegate a collection of random BB84 states of the form $H^{\theta_1} |v_1\rangle \otimes \cdots \otimes H^{\theta_n} |v_n\rangle$, where $\vec{v}, \vec{\theta} \in \{0,1\}^n$ are random strings known only to $\mathcal{C}$. Then, $\mathcal{C}$ can choose $\vec{r} \in \{0,1\}^n$ such that $\vec{v} = \vec{m} \oplus \vec{r}$ and set $\vec{k} = (\vec{r}, \vec{\theta})$ as the secret key. With this choice of key, the delegated parallel BB84 states are exactly the ciphertext $\mathsf{Enc}_k(\vec{m})$ in Equation (1.1). Because the final output state of the protocol is computationally indistinguishable from a tensor product of BB84 states (known to the client), we can follow a similar proof as in [BL19] to obtain a classical-client unclonable encryption scheme with inverse-polynomial security.

Quantum copy-protection. In quantum copy-protection (QCP), a vendor wishes to encode a program into a quantum state in a way that enables a recipient to run the program, but not to create functionally equivalent "pirated" copies. The notion of QCP was introduced by Aaronson [Aar09], who gave the first construction for unlearnable and efficiently computable functions in a strong quantum oracle model, which has since been improved to only requiring classical oracles [ALL+21]. Recent work [CMP20] has also provided the first construction of QCP for compute-and-compare programs in the quantum random oracle model (QROM) as well as a construction for multi-bit point functions in the QROM based on unclonable encryption with wrong-key detection (WKD), a property which enables the decryption procedure to recognise incorrect keys. In this work, we build on the latter and give a new QCP scheme that does not require the QROM assumption.

Our new QCP scheme for multi-bit point functions relies on pair-wise independent permutations [NR97], which exist unconditionally and have previously been used in the context of classical obfuscation of point functions [CKVW10]. The basic idea behind our QCP scheme is as follows. To encode a point function $P_{\vec{y}, \vec{m}}$, which is defined as returning $\vec{m}$ on input $\vec{y}$ and 0 otherwise, we sample a random pairwise independent

permutation π over $\{0,1\}^{4n}$ and a random string $\vec{r} \xleftarrow{\$} \{0,1\}^n$, let $\pi(\vec{y}) = (\vec{s}||\vec{\theta})$, and output the state

$$\rho = \mathsf{Enc}_{\pi(\vec{y})}(\vec{r}||\vec{m}) \otimes |\vec{r}\rangle\langle\vec{r}| \otimes |\pi\rangle\langle\pi|, \text{ where } \mathsf{Enc}_{\pi(\vec{y})}(\vec{r}||\vec{m}) = \bigotimes_{i=1}^{2n} H^{\theta_i}|s_i \oplus (r_i||m_i)\rangle\langle s_i \oplus (r_i||m_i)|H^{\theta_i}.$$

To evaluate the program on an input $\vec{x} \in \{0,1\}^{4n}$, we compute $\pi(\vec{x}) = (\vec{s}_x||\vec{\theta}_x)$ and apply the decryption procedure of our aforementioned unclonable encryption scheme using the key $(\vec{s}_x, \vec{\theta}_x)$. Note that it is possible to *coherently* check whether a particular input decrypts to a plaintext with pre-fix $\vec{r} \in \{0,1\}^n$. In other words, the WKD property allows us to check whether an input to the point function $P_{y,m}$ is valid without irreversibly destroying the state ρ in the process. Our second contribution is to show how to obtain a QCP scheme with a classical client through the use of our parallel RSP protocol for preparing random BB84 states, similar to our aforementioned classical-client unclonable encryption scheme . Our scheme enables a classical client to delegate a correct copy-protected program from the class of multi-bit point functions consisting of uniformly random marked inputs $\vec{y}$ and output strings $\vec{m}$ with inverse-polynomial security.

Quantum computing on encrypted data. Suppose a client wishes to perform some quantum computation, represented as the action of a quantum circuit C on an input state $|x\rangle$, with $x \in \{0,1\}^n$. For simplicity, we will assume the desired output is classical and corresponds to a computational basis measurement of $C|x\rangle$. The client only has limited quantum capability and therefore wishes to delegate the computation to a quantum server while ensuring the privacy of the input $|x\rangle$ and the output resulting from the measurement of $C|x\rangle$. Essentially, the client would like to send the server an encryption of the input and, after performing an interactive protocol, obtain an encryption of the output (which the client can decrypt, but the server cannot)[1]. This primitive is called quantum computing on encrypted data (QCED).

Many protocols for QCED with differing quantum requirements on the client have been developed (see [Fit17] for a survey). Here we will focus on the protocol of Broadbent [Bro15] which achieves QCED with a client that is only required to prepare BB84 states and send them to the server. This makes the protocol well-suited for dequantisation via our parallel RSP protocol. Before explaining this dequantisation, we (informally) define what a QCED protocol *with a classical client* should achieve. As before, the client's input is the string $x \in \{0,1\}^n$ and the goal is to obtain the outcome of measuring $C|x\rangle$ in the computational basis. In contrast to before, this must be achieved using only *classical* interaction with the quantum server. The requirement that the client's input must stay private is captured by the condition that after interacting with the client, it must be computationally intractable for the server to decide which one of two distinct inputs the client used.

Our QCED protocol with a classical client works as follows. The client first performs the parallel RSP protocol with the server, resulting in the preparation of BB84 states (or the client aborting). Provided the protocol succeeded, the client proceeds to run Broadbent's protocol [Bro15] as if the server had received those BB84 states via a quantum channel. The security proof is straightforward. First, we know that after performing RSP the server's state is computationally indistinguishable from a tensor product of BB84 states (known to the client). Furthermore, the interaction in [Bro15] preserves this computational indistinguishability. Hence, the server's state at the end of the protocol is indistinguishable from the state the server would have obtained by executing the protocol with random BB84 states and the security of our protocol follows from [Bro15].

---

[1]This also allows the client to hide the computation itself from the server by suitably encoding it as part of the input x and taking C to be a *universal circuit*. When the primary goal of the protocol is to hide the computation, it is referred to as a *blind quantum computing protocol* [AS06, BFK09].

Verifiable delegated blind quantum computation. The final application we consider is verifiable delegated blind quantum computation (VDBQC). VDBQC is an interactive protocol between two parties, in this case denoted as the verifier and the prover. The verifier delegates a computation to the prover and, in addition to ensuring input-output privacy as in QCED, the protocol also ensures that the probability for the verifier to accept an incorrect output is small. In other words, if the prover deviates from the protocol and does not perform the verifier's instructed computation, the verifier should be able to detect this and abort with high probability. As with QCED, a number of such protocols have been developed and we refer the reader to [GKK19] for a survey.

Here we focus on a protocol by Morimae [Mor18]. This protocol achieves verifiability by combining a protocol for blind quantum computation (or QCED) with the *history state construction*, which is a special encoding of a quantum circuit into a quantum state [KSVV02, KKR06]. In Morimae's protocol, for a given circuit C the verifier uses a QCED protocol to delegate to the prover the preparation of two such history states (one for C and one for the complement of C, where the output qubit is negated). The verifier then requests these states from the prover and proceeds to measure them in the computational or Hadamard basis. This allows the verifier to determine the output of the computation. The history state construction guarantees that malicious behavior on the prover's part would be detected by the verifier's measurement. Additionally, the use of a QCED protocol ensures that the prover is "blind", i.e. does not know which computation the verifier delegated.

To dequantise this protocol, we use our QCED protocol with a classical client to delegate the preparation of the two history states to the quantum prover. We then replace the verifier's measurements on this state by a *measurement protocol* due to Mahadev [Mah18], which allows the classical verifier to delegate these measurements to the prover in a way that forces the prover to report the correct outcomes. We thus obtain a VDBQC protocol with a classical verifier. Crucially, through the use of the classical client QCED protocol and Mahadev's measurement protocol, the prover is "computationally blind", i.e. unable to distinguish which computation the verifier has performed. In contrast, Mahadev's verification protocol [Mah18] does not have this property.[2]

1.2 Related work

A number recent of works starting with [BCM+18, Mah18] have developed techniques that allow a classical verifier to use post-quantum cryptography to force an untrusted (but computationally bounded) quantum prover to behave in a certain way. Here, we briefly describe these works and explain their relation to our parallel RSP protocol.

In a breakthrough result [Mah18], Mahadev introduced a protocol that allows a classical verifier to delegate a quantum computation to a quantum computer and be able to verify the correctness of the result. The key ingredient for this protocol is a *measurement protocol*, which allows the verifier to securely delegate single-qubit measurements in the standard or Hadamard basis to a quantum prover, assuming that the prover cannot break the LWE assumption. This can then be applied to so-called *prepare-and-measure protocols*: if one has a protocol that involves a quantum prover preparing and sending a quantum state to the verifier and the verifier performing single-qubit measurements on this state, one can use Mahadev's measurement protocol to delegate these quantum measurements to the prover itself. This yields a protocol in which the prover only sends classical measurement outcomes to the verifier, hence making the verifier classical.

---

[2]In [GV19], the authors also construct a blind verification protocol based on RSP. However, they approach the problem in a composable framework, which requires them to make an additional assumption on the prover (called the *measurement buffer* in [GV19]). In contrast, our protocol requires no extra assumptions on the prover. We describe the issue with the measurement buffer assumption in more detail in Section 1.2.

This measurement protocol is in many ways similar to what we seek to do in this paper: it removes the need for quantum communication between a fully quantum prover and a verifier with very limited quantum capabilities (only measuring single qubits in the computational or Hadamard basis). The difference to our work is that we are concerned with *prepare-and-send protocols*, in which the verifier sends random BB84 states to the prover instead of receiving them.

It turns out that replacing the quantum communication of prepare-and-send protocols requires significantly stronger control over the untrusted prover. At a high level, the reason is the following: for Mahadev's measurement protocol, it suffices to show that *there exists* a quantum state that is consistent with the distribution of measurement outcomes reported by the prover, in the sense that the measurement outcomes for different bases could have been obtained by measurements on (copies of) the same state. In contrast, if we want to replace the step of the verifier sending a physical quantum state to the prover, we need to show that the prover has *actually constructed* a certain quantum state, not just that such a quantum state exists mathematically.[3] We give a more detailed description of what it means to "actually construct" a quantum state in Section 1.3.

The first classical protocol that provably forced a quantum prover to prepare a certain quantum state was the single-qubit RSP protocol of [GV19] (see also [CCKW19] for a related result). This protocol essentially achieves our informal theorem as stated above for a single qubit, i.e. $n = 1$.[4] We give a brief outline of that protocol and its soundness proof in Section 1.3.

The main difficulty in going from [GV19] to our parallel RSP result is enforcing a tensor product structure on the prover's space: we would like to show that if we execute multiple instances of a single-qubit RSP protocol in parallel, a successful prover must treat each of these copies independently. Mathematically, this means that we need to be able to split the prover's a priori uncharacterised Hilbert space into a tensor product, where each tensor factor is supposed to correspond to one instance of the RSP protocol. This is a more demanding version of the classic question of parallel repetition: there, one is interested in showing that any prover's winning probability in the protocol decays in essentially the same way as it would for a prover who executes the instances independently. In contrast, we need to show that the prover really does execute the different instances independently in a physically meaningful sense. We call this stronger requirement *parallel rigidity*.[5] The question of parallel rigidity has been studied extensively in the literature on quantum self-testing [CN16, Col16, NV17, NV18], where one considers a setting of two non-communicating provers. Unfortunately, those techniques are not immediately transferable to the setting we consider here, namely a single computationally bounded prover.

Some progress towards the question of parallel rigidity for single computationally bounded provers was

---

[3]In fact, in [VZ21] it was shown that Mahadev's measurement protocol does ensure that the prover *knows* (in the sense of a proof of knowledge) the state it is measuring, not just that it exists mathematically. The notion of "knowing" a quantum state is quite subtle to define and we forego a detailed description here, but point out that this is weaker than showing that the prover actually constructed the state and (to the best of our knowledge) not sufficient to use Mahadev's protocol for prepare-and-measure scenarios.

[4]The protocol in [GV19] in fact works for a set of 10 states which includes the 4 BB84 states. Here, we only focus on the 4 BB84 states as this is the case we will deal with in our parallel RSP protocol.

[5]In [GV19], the authors show that their protocol has composable security. This may suggest that one can obtain a parallel rigidity statement simply by composing the protocol with itself in parallel. However, this is not the case because the composable security statement in [GV19] requires an additional assumption called a *measurement buffer*, which effectively acts as a trusted intermediary between the verifier and the prover. A parallel composition of the protocol in [GV19] would utilise a different measurement buffer for each instance, thereby forcing the prover to treat the different instances in a (largely) independent way. In particular, this means that one already assumes a tensor product structure with n separate qubits in the prover's space, whereas in our work enforcing this tensor product structure is the key technical challenge. For cryptographic applications, we do not want to place any such assumption on the prover and instead allow the prover to perform arbitrary global operations involving all instances. This is what our parallel RSP protocol achieves.

made in [MV20], which gives a protocol that allows a classical verifier to certify that a quantum prover must have prepared and measured a Bell state, i.e. an entangled 2-qubit quantum state. This has since been applied to device-independent quantum key distribution [MDCAF21] and oblivious transfer [BY21], and been extended to work for magic states [MTH$^+$21]. The protocol from [MV20] uses a 2-fold parallel repetition of [GV19] (with additional steps to allow for the certification of an entangled state, not just product states). As part of their soundness proof, [MV20] do show a kind of parallel rigidity result for 2 instances of the RSP protocol. However, their method does not generalise to an n-fold parallel repetition without an exponential decay in parameters. Hence, for our n-fold parallel rigidity proof, new techniques are needed. We give a more detailed comparison between our new parallel rigidity proof and the method in [MV20] at the end of Section 1.3.

In addition to this line of work focused on rigidity statements, application-specific dequantisations were already considered for private-key quantum money [RS19], certifiable deletion of quantum encryption [HMNY21] and secure software leasing [KNY21]. In all these cases the authors derived the desired security statement from properties of trapdoor claw-free functions, a cryptographic primitive which is also the basis of our RSP protocol. While this is less generic and modular than our approach and requires a new analysis for each application, it does have the advantage that one can obtain *negligible* security, whereas with RSP we obtain inverse polynomial security. We comment more on the possibility of negligible security from RSP-like primitives in Section 1.4.

1.3 Soundness proof for parallel RSP protocol

We give a short sketch of the soundness proof for our parallel RSP protocol. The full protocol is described as Protocol 3 (though our discussion in the introduction is restricted to Protocol 1) and its soundness proof is given in Section 4. We briefly explain the difference between Protocol 1 and Protocol 3: Protocol 1 is a protocol to *test* the prover, i.e. in this protocol the prover is asked to prepare *and measure* a quantum state, and the verifier runs checks on the prover's answer. The soundness statement for this protocol is a self-testing statement in the sense of [MV20], which characterises which states and measurements the prover used in the protocol. Although we do not spell this out in Section 4, it is easy to obtain an explicit self-testing statement from our proof. In contrast, Protocol 3 is a protocol for remote state preparation, so the prover is supposed to prepare, but not yet measure, a particular quantum state. Instead, this quantum state will be used for other applications. This means that we do not want to make a statement about how the prover measured its state, but rather what state remains in its quantum memory. The soundness of Protocol 3 follows from that of Protocol 1 via a statistical argument. In the following, we focus on Protocol 1. We do not explain the protocol and the cryptographic primitives underlying it in detail; instead, we give a very high-level description of the relevant part of the soundness proof of the RSP protocol from [GV19] and then explain our method for proving a parallel rigidity statement based on that result.

The main cryptographic primitive underlying the RSP protocol is a so-called extended noisy trapdoor claw-free function (ENTCF) family, which can be constructed assuming the quantum hardness of LWE [Reg09, Mah18]. An ENTCF family is a family of functions indexed by a set of keys $\mathcal{K}_0 \cup \mathcal{K}_1$. $\mathcal{K}_0$ and $\mathcal{K}_1$ are disjoint sets of keys with the property that given a $k \in \mathcal{K}_0 \cup \mathcal{K}_1$, it is computationally intractable to determine which set this key belongs to. See [Mah18, Section 4] for further details on ENTCF families.

In the RSP protocol from [GV19], for a given *basis choice* $\theta \in \{0, 1\}$ (where "0" corresponds to the computational and "1" to the Hadamard basis), the verifier samples a key $k \in \mathcal{K}_\theta$, alongside some trapdoor information t. The verifier sends k to the prover and keeps t private. The verifier and prover then interact classically; for us, the main point of interest is the last round of the protocol. Let us denote the protocol's transcript up to the last round by ts. Before the last round, the remaining quantum state of an *honest* prover

9

is the single-qubit state $H^\theta|v\rangle\langle v|H^\theta$ for $v \in \{0,1\}$. From the transcript and the trapdoor information, the verifier can compute v; in contrast, the prover, who does not know the trapdoor, cannot efficiently compute θ or v. In the last round, the verifier sends a random measurement basis $\theta' \in \{0,1\}$ to the prover, who returns $v' \in \{0,1\}$. In case $\theta' = \theta$, the verifier then checks whether $v' = v$. The honest prover would generate v' by measuring its remaining qubit $H^\theta|v\rangle\langle v|H^\theta$ in the basis θ' and therefore always pass the verifier's check.

We can model this last round of the protocol (with a potentially dishonest prover) as follows: at the start, the prover has a state $\sigma^{(\theta,v)}$, which it produced as a result of the previous rounds of the protocol. For an honest prover, $\sigma^{(\theta,v)} = H^\theta|v\rangle\langle v|H^\theta$. Of course, this state can depend on all of ts, but we only make the dependence on θ and v explicit. Upon receiving $\theta' \in \{0,1\}$ the prover measures a binary observable Z (if $\theta' = 0$) or X (if $\theta' = 1$) and returns the outcome v'. An honest prover would simply use the Pauli observables $Z = \sigma_Z$ and $X = \sigma_X$. The key step in the proof of [GV19] is to show that, due to the properties of ENTCF families, for any (potentially dishonest) prover that is accepted with high probability, the observables X and Z must anti-commute when acting on the prover's state. Then, Theorem 1.1 (for $n = 1$) follows from known results [MYS12, NV17, GH17].

For our parallel RSP protocol we run n independent copies of the protocol from [GV19] in parallel, except that the basis choice θ_i is the same for each copy, and likewise all θ_i' are the same.[6] The prover's state before the last round of each copy of the RSP protocol is now denoted by $\sigma^{(\theta,\vec{v})}$, where $\vec{v} \in \{0,1\}^n$ can be calculated by the verifier from the transcript ts by repeating the same calculation as above for each parallel copy. Generalising from the single-qubit case, given $\theta' \in \{0,1\}$ the prover performs a measurement to generate $\vec{v} \in \{0,1\}^n$, which we can describe by binary observables Z_i, X_i (for $\theta' = 0,1$ respectively) that correspond to the observable used to produce the i-th entry of $\vec{v}$. (For an honest prover, $\sigma^{(\theta,\vec{v})} = H^\theta|v_1\rangle\langle v_1|H^\theta \otimes \ldots \otimes H^\theta|v_n\rangle\langle v_n|H^\theta$ and Z_i is a Pauli-Z measurement on the i-th qubit.)

The main challenge in the proof is to establish that the prover must treat all of the parallel copies of the RSP protocol independently, i.e. to show that its (a priori uncharacterised) Hilbert space can be partitioned into n identical subspaces, one for each copy of the protocol. At first sight, it might look as though for this it suffices to show that X_i and Z_j (approximately) commute for all $i \neq j$. However, this is not the case because any such commutation statement can only be shown in a special *state-dependent distance* [Vid11], which does not allow us to combine individual commutation statements into the *global* statement that the Hilbert space factorises into n subspaces. Instead, we need to consider the family $\{Z(\vec{a})X(\vec{b})\}_{\vec{a},\vec{b}\in\{0,1\}^n}$ of 4^n binary observables, where $Z(\vec{a}) = Z_1^{a_1} \cdots Z_n^{a_n}$. We then have to show that $\{Z(\vec{a})X(\vec{b})\}$ form an approximate representation of the Pauli group [GH17, Vid20].[7] This means that when acting on the prover's (unknown) state $\sigma^{(\theta)}$ (where $\sigma^{(\theta)}$ is like $\sigma^{(\theta,\vec{v})}$, but averaged over all $\vec{v}$), the operators $\{Z(\vec{a})X(\vec{b})\}$ behave essentially like Pauli operators. Formally, this means showing that on average over $\vec{a}, \vec{b} \in \{0,1\}^n$,

$$\text{Tr}\left[Z(\vec{a})X(\vec{b})Z(\vec{a})X(\vec{b})\sigma^{(\theta)}\right] \approx (-1)^{\vec{a}\cdot\vec{b}}. \tag{1.2}$$

This is the appropriate generalisation of the statement that Z and X anti-commute in the single-qubit case. It is easy to check that Equation (1.2) holds when Z_i and X_i are the Pauli observables.

Our proof of Equation (1.2) has five main steps, which we briefly sketch here with references to the corresponding parts of the formal proof.

---

[6]The advantage of this is that a prover that succeeds with high probability on average over θ must also succeed with high probability for each θ individually. If we were to sample θ independently for each of the parallel copies we could not conclude that a prover succeeds with high probability for any particular choice of $\theta_1, \ldots, \theta_n$ as there are exponentially many such choices.

[7]When we say "Pauli group" we always mean the Pauli group modulo complex conjugation, which is also sometimes called the Heisenberg-Weyl group.

(1) Instead of working with the observables X_i, we define "inefficient observables" $\tilde{X}_i = (-1)^{v_i} X_i$, where v_i is the i-th bit of the verifier's string $\vec{v}$ (Definition 4.4). $\tilde{X}_i$ is not an observable that an efficient prover can implement because it depends on v_i, which requires the trapdoor information to be computed efficiently. Intuitively, while X_i describes the prover's answer, $\tilde{X}_i$ describes whether that answer is accepted by the verifier. This has the advantage that the state $\sigma^{(\theta=1)}$ (averaged over $\vec{v}$) of a successful prover is an approximate $+1$-eigenstate of $\tilde{X}_i$, but not of X_i (Lemma 4.10).

(2) We extend the family of states $\{\sigma^{(\theta)}\}_{\theta \in \{0,1\}}$ to a larger family of "counterfactual states" $\{\sigma^{(\vec{\theta})}\}_{\vec{\theta} \in \{0,1\}^n}$, which are defined as the states the prover would have prepared if the verifier had sent keys $k_i \in \mathcal{K}_{\theta_i}$. In Protocol 1 the basis choice is the same for all i, i.e. $\vec{\theta} = \vec{0}$ or $\theta = \vec{1}$, so for other choices of $\vec{\theta}$ these states are never actually prepared. However, they are still well-defined because for any prover in the actual protocol, we can fix that prover's operations (as a quantum circuit acting on a given input) and then consider what state those operations would produce if given keys with an arbitrary basis choice $\vec{\theta}$. The reason these counterfactual states are useful is that we can show that, as a consequence of the properties of ENTCF families, the states $\{\sigma^{(\vec{\theta})}\}_{\vec{\theta}}$ are computationally indistinguishable (Section 4.2).

(3) We now want to show various commutation and anti-commutation relations for the observables $Z(\vec{a})$ and $\tilde{X}(\vec{b})$ (Lemma 4.21 and Lemma 4.23). For example, we want to show that Z_i and $\tilde{X}_i$ anti-commute, but Z_i and $\tilde{X}_j$ commute (for $i \neq j$). To show these relations, we make use of the counterfactual states $\sigma^{(\vec{\theta})}$ in the following way: for any particular relation, we can pick a $\vec{\theta}$ that makes showing this relation especially convenient. For example, to show that Z_i and $\tilde{X}_j$ commute, we would choose a $\vec{\theta}$ with $\theta_i = 0$ and $\theta_j = 1$ since the verifier can check the outcomes of "Z-type observables" for $\theta = 0$ and "X-type observables" for $\theta = 1$. Using the properties of ENTCF families, we can argue that the prover's measurements on these counterfactual states still yield outcomes that would pass the verifier's checks for each choice of θ_i. Based on this, we can show the desired relations for a "convenient" choice of counterfactual state $\sigma^{(\vec{\theta})}$. Then, we can relate these statements back to the prover's actual states $\sigma^{(\theta)}$ using the computational indistinguishability of $\{\sigma^{(\vec{\theta})}\}$. This is somewhat delicate because $\tilde{X}_i$ are inefficient, see Lemma 4.13 for the precise statement.

(4) We can combine the various commutation and anti-commutation statements from the previous step to show that the observables $\{Z(\vec{a})\tilde{X}(\vec{b})\}$ behave like Pauli observables on $\sigma^{(\theta=1)}$, i.e. we show Equation (1.2) but with $\tilde{X}$ instead of X (Proposition 4.19). This step relies on the fact that $\sigma^{(\theta=1)}$ is an approximate $+1$-eigenstate of $\tilde{X}(\vec{b})$ for all $\vec{b}$; see the proof of Proposition 4.19 for details on how this simplifies the analysis.

(5) Since we now know that $\{Z(\vec{a})\tilde{X}(\vec{b})\}$ behave essentially like Pauli observables, we can define an explicit isometry $\tilde{V}$ which can be shown to map $\{Z(\vec{a})\tilde{X}(\vec{b})\}$ to the corresponding Pauli observables (Lemma 4.28). This means that we have good control over these inefficient observables, and we know how the inefficient and efficient observables are related. We can use this to define a modified isometry V that maps the efficient observables $\{Z(\vec{a})X(\vec{b})\}$ to the corresponding Pauli observables (Proposition 4.29). This is a stronger version of Equation (1.2) and, combined again with the verifier's checks in the protocol and properties of ENTCF families, can be used to show that the prover must have prepared BB84 states (Section 4.5).

We briefly comment on the relation between our soundness proof and that in [MV20]. At a high level, the soundness proof in [MV20] also shows a kind of "parallel rigidity" of two executions of a remote state preparation protocol. However, their proof proceeds quite differently from ours: they first show that observables "on the first qubit" anti-commute, which allows them to make a partial statement about the prover's state. This in turn can be used to extend the statement about the prover's observables to two-qubit

observables, which is finally used to prove a statement about the prover's two-qubit state. This qubit-by-qubit approach is extremely costly in terms of parameters due to switching back and forth between making partial statements about the observables and state, and cannot reasonably be extended to n qubits. In contrast, we can make a global statement about the prover's 4^n possible observables without first characterising parts of the prover's state. This allows us to prove a parallel rigidity statement for n qubits without an exponential degradation of parameters.

1.4 Discussion

We have shown how a classical verifier can certify a tensor product of BB84 states in the memory of a quantum prover, assuming the quantum-intractability of the LWE problem. Importantly, the prover does not know which BB84 states it has prepared, whereas the verifier does. Hence, the result at the end of the protocol is as if the verifier had sent random BB84 states to the prover. This allows us to dequantise a number of quantum cryptographic primitives, yielding a generic and modular way of translating these protocols to a setting where only classical communication is used.

Our work raises a number of interesting open problems. Firstly, while our RSP primitive is based on the hardness of LWE, we can ask whether it is possible to achieve this functionality from weaker computational assumptions. For instance, would it be possible to perform an RSP-like protocol assuming only the existence of quantum-secure one-way functions? This is of particular interest because recent results have shown that secure two-party computation can be achieved from one-way functions and quantum communication [BCKM21, GLSV21]. These results are based on the fact that an oblivious-transfer protocol can be implemented from one-way functions and quantum communication that consists of BB84 states. However, an RSP primitive like ours would allow one to generically dequantise that quantum communication. Hence if RSP (with sufficiently strong parameters) can be obtained from quantum-secure one-way functions, then secure two-party computation can also be obtained from those functions, together with classical communication. In light of earlier work [IR89, Rud91] we conjecture that this is impossible. Formalising this intuition could lead to a better understanding of the minimum assumptions required for performing RSP-like protocols.

Secondly, a more technical open problem concerns the parameters of our rigidity theorem, Theorem 1.1. As stated above, provided the prover accepts, the state the verifier certifies is $1/\operatorname{poly}(n)$-close to a tensor product of n BB84 states (up to an isometry). The $1/\operatorname{poly}(n)$ closeness means that the soundness error of our dequantised protocols also scales as $1/\operatorname{poly}(n)$. It would be desirable to achieve *negligible* soundness error, particularly when considering composable instances of these protocols. Doing this via RSP would require a version of the rigidity statement where the prover's state can be characterised up to negligible closeness, which seems unlikely. A more promising approach might be to circumvent an explicit RSP statement: the advantage of the RSP statement in our paper is that one can use it to dequantise existing protocols easily, but these existing protocols typically only use BB84 states because of their no-cloning properties. Therefore, instead of using an RSP protocol to prepare those states, one could instead try to show a "post-quantum cryptographic no-cloning property" directly that could plausibly be used to dequantise these protocols while preserving negligible soundness.

Finally, we mention that our derivation of the parameters in the rigidity theorem (Theorem 4.33) is likely not optimal and could be optimised to improve the efficiency of our protocol. The situation here is similar to that of parallel self-testing in the multi-prover setting, with the first works having round complexity that scaled as a high-degree polynomial [RUV13] and more recent works achieving quasilinear scaling [NV17, CGJV19]. It would be interesting to see whether ideas from these newer works are also applicable in the setting of parallel remote state preparation.

During the preparation of this manuscript, we became aware of related independent work by Fu, Wang, and Zhao, who also show a parallel rigidity statement in the setting of a single computationally bounded prover. More specifically, they are able to self-test (in the sense of [MV20]), i.e., certify the preparation and measurement of, two kinds of states: (i) n parallel BB84 states subject to the restriction that only one of the qubits may be in the Hadamard basis and the remaining $n - 1$ are in the computational basis, and (ii) parallel EPR pairs. They also present a dimension testing protocol based on their self-test. We refer to their paper [FWZ22] for more details and thank them for their cooperation in publishing our respective results at the same time.

Part I

Classically-instructed parallel remote state preparation of BB84 states

2 Preliminaries

We will follow the notation of [MV20], and also make frequent use of results from the preliminaries of that paper. We include a copy of the most frequently used notations and lemmas from that paper here for the reader's convenience.

2.1 Notation

A function $n : \mathbb{N} \to \mathbb{R}_+$ is called negligible if $\lim_{\lambda \to \infty} n(\lambda)p(\lambda) = 0$ for any polynomial p. We use $\mathrm{negl}(\lambda)$ to denote an arbitrary negligible function. We use the notation $x \xleftarrow{\$} \mathcal{S}$ to denote that x is being sampled from the set $\mathcal{S}$ uniformly at random. We denote the concatenation of two strings x and y as $x||y$.

We use $\mathcal{H}$ to denote an arbitrary finite-dimensional Hilbert space, and use indices to differentiate between distinct spaces. The map $\mathrm{Tr} : \mathcal{L}(\mathcal{H}) \to \mathbb{C}$ denotes the trace, and $\mathrm{Tr}_B : \mathcal{L}(\mathcal{H}_A \otimes \mathcal{H}_B) \to \mathcal{L}(\mathcal{H}_A)$ is the partial trace over subsystem B. $\mathrm{Pos}(\mathcal{H})$ denotes the set of positive semidefinite operators on $\mathcal{H}$, and $\mathcal{D}(\mathcal{H}) = \{A \in \mathrm{Pos}(\mathcal{H}) \mid \mathrm{Tr}[A] = 1\}$ is the set of density matrices on $\mathcal{H}$.

The single qubit Pauli operators are $\sigma_X = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ and $\sigma_Z = \left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$. The Hadamard basis states are written as $|(-)^b\rangle = \frac{1}{\sqrt{2}}(|0\rangle + (-1)^b|1\rangle)$.

An observable on $\mathcal{H}$ is a Hermitian linear operator on $\mathcal{H}$. A binary observable is an observable that only has eigenvalues $\in \{-1, 1\}$. For a binary observable O and $b \in \{0, 1\}$, we denote by $O^{(b)}$ the projector onto the $(-1)^b$-eigenspace of O. For any procedure which takes a quantum state as input and produces a bit (or more generally an integer) as output, e.g., by measuring the input state, we denote the probability distribution over outputs b on input state ψ by $\mathrm{Pr}[b \mid \psi]$.

Our main protocol Protocol 1 will be (almost) a parallel repetition of a sub-protocol. We make use of vector notation to denote tuples of items corresponding to the different copies of the sub-protocol. For example, if each of the n parallel sub-protocols requires a key k_i, we denote $\vec{k} = (k_1, \ldots, k_n)$. A function that takes as input a single value can be extended to input vectors in the obvious way: for example, if f takes as input a single key k, then we write $f(\vec{k})$ for the vector $(f(k_1), \ldots, f(k_n))$. We will also use $\vec{0}$ and $\vec{1}$ for the bitstrings consisting only of 0 and 1, respectively (and whose length will be clear from the context), and $\vec{1}^i \in \{0, 1\}^n$ for the bitstring whose i-th bit is 1 and whose remaining bits are 0.

2.2 Extended trapdoor claw-free functions

Our remote state preparation protocol is based on a cryptographic primitive called extended noisy trapdoor claw free function families (ENTCF families), which are defined in [Mah18, Section 4] and can be constructed from the Learning with Errors assumption [Reg09, BCM$^+$18] We use the same notation as in [Mah18, Section 4], with the exception that we write $\mathcal{K}_0$ instead of $\mathcal{K}_{\mathcal{G}}$ and $\mathcal{K}_1$ instead of $\mathcal{K}_{\mathcal{F}}$. In addition, we also define the following functions for convenience:

Definition 2.1 (Decoding maps, [MV20, Definition 2.1])**.**

(i) For a key $k \in \mathcal{K}_0 \cup \mathcal{K}_1$, an image $y \in \mathcal{Y}$, a bit $b \in \{0,1\}$, and a preimage $x \in \mathcal{X}$, we define $\text{CHK}(k,y,b,x)$ to return 1 if $y \in \text{supp}(f_{k,b}(x))$, and 0 otherwise. (This definition is as in [Mah18, Definition 4.1 and 4.2]).

(ii) For a key $k \in \mathcal{K}_0$ and a $y \in \mathcal{Y}$, we define $\hat{b}(k,y)$ by the condition $y \in \cup_x \text{supp}(f_{k,\hat{b}(k,y)}(x))$. (This is well-defined because $f_{k,1}$ and $f_{k,2}$ form an injective pair.)

(iii) For a key $k \in \mathcal{K}_0 \cup \mathcal{K}_1$ and a $y \in \mathcal{Y}$, we define $\hat{x}_b(k,y)$ by the condition $y \in \text{supp}(f_{k,b}(\hat{x}_b(k,y)))$, and $\hat{x}_b(k,y) = \perp$ if $y \notin \cup_x \text{supp}(f_{k,b}(x))$. For $k \in \mathcal{K}_0$, we also use the shorthand $\hat{x}(k,y) = \hat{x}_{\hat{b}(k,y)}(k,y)$.

(iv) For a key $k \in \mathcal{K}_1$, a $y \in \mathcal{Y}$, and a $d \in \{0,1\}^w$, we define $\hat{u}(k,y,d)$ by the condition $d \cdot (\hat{x}_0(k,y) \oplus \hat{x}_1(k,y)) = \hat{u}(k,y,d)$.

The above decoding maps applied to vector inputs are understood to act in an element-wise fashion. For example, for $\vec{k} \in \mathcal{K}_1^{\times n}, \vec{y} \in \mathcal{Y}^{\times n}$, and $\vec{d} \in \{0,1\}^{w \times n}$, we denote by $\hat{u}(\vec{k}, \vec{y}, \vec{d}) \in \{0,1\}^n$ the string defined by $\left(\hat{u}(\vec{k}, \vec{y}, \vec{d}) \right)_i = \hat{u}(k_i, y_i, d_i)$.

2.3 Computational efficiency and distinguishability

Our proof will make a computational assumption, meaning that we assume that a certain problem cannot be solved by an *efficient* adversary. To formally define what "efficient" means, one has to introduce a security parameter λ, which roughly speaking specifies the instance size of a problem. We usually leave this security parameter implicit, i.e. when we say a procedure is efficient, we mean that there is a family of procedures, one for each λ, whose complexity has the required scaling in λ. Suppressing the explicit λ-dependence, we can give the following definitions.

Definition 2.2 (QPT procedure). A procedure (with potentially quantum inputs and outputs and an implicit dependence on a security parameter λ) is called *quantum polynomial-time* (QPT) if there exists a (classical) polynomial-time Turing machine M that, on input 1^λ, outputs a description of a circuit (with a fixed gate set) that implements the procedure. We often use the term "efficient" to mean "implementable by a QPT procedure".

Definition 2.3 (Computational (in)distinguishability). We call two (families of) states $\psi, \psi' \in \mathcal{D}(\mathcal{H})$ *computationally indistinguishable up to* $O(\delta)$ if for an QPT procedure (called a *distinguisher*) that takes as input ψ or ψ' and produces an output bit b, we have

$$\Pr[b = 0|\psi] \approx_\delta \Pr[b = 0|\psi'] . \tag{2.1}$$

2.4 Distance measures

Definition 2.4 (Norms). Let $A \in \mathcal{L}(\mathcal{H})$ with singular values $\lambda_1, \dots, \lambda_n \geq 0$. Then, the trace norm is defined as

$$\|A\|_1 = \sum_i \lambda_i ,$$

and the spectral norm is

$$\|A\|_\infty = \max\{\lambda_i\} .$$

Definition 2.5 (Total variation distance). Let P, Q be probability distributions over a finite alphabet $\mathcal{X}$. Then, the total variation distance between P and Q is given by

$$\mathrm{TVD}[P, Q] = \frac{1}{2} \sum_{x \in \mathcal{X}} |P(x) - Q(x)|.$$

Definition 2.6 (Approximate equality, [MV20, Definition 2.8 and Definition 2.14]). We overload the symbol "$\approx$" in the following ways (leaving the dependence on the security parameter implicit in the quantities on the left):

(i) **Complex numbers:** For $a, b \in \mathbb{C}$ we define:

$$a \approx_\varepsilon b \iff |a - b| = O(\varepsilon) + \mathrm{negl}(\lambda). \tag{2.2}$$

(ii) **Operators:** For $A, B \in \mathcal{L}(\mathcal{H})$, we define:

$$A \approx_\varepsilon B \iff \|A - B\|_1^2 = O(\varepsilon) + \mathrm{negl}(\lambda). \tag{2.3}$$

(We will most frequently use this for (possibly subnormalised) quantum states $A, B \in \mathrm{Pos}(\mathcal{H})$.)

(iii) **Operators on a state:** For $A, B \in \mathcal{L}(\mathcal{H})$ and $\psi \in \mathrm{Pos}(\mathcal{H})$, we define:

$$A \approx_{\varepsilon, \psi} B \iff \mathrm{Tr}\Big[(A - B)^\dagger (A - B)\psi\Big] = O(\varepsilon) + \mathrm{negl}(\lambda). \tag{2.4}$$

(iv) **Computationally indistinguishable states:** For two (families of) states $\psi, \psi' \in \mathcal{D}(\mathcal{H})$ which are computationally indistinguishable up to δ (i.e., no efficient distinguisher has advantage exceeding δ in distinguishing ψ from ψ'), we write:

$$\psi \overset{c}{\approx}_\delta \psi'. \tag{2.5}$$

If we write $\approx_0$, we mean that the quantities are negligibly close. All asymptotic statements are understood to be in the limits $\varepsilon \to 0$ and $\lambda \to \infty$. It is easy to check that all the distance measures underlying the $\approx$-notation are indeed metrics and in particular satisfy the triangle inequality (see [MV20, Section 2.4] for details).

Lemma 2.7 ([MV20, Lemma 2.16]). *Let $\mathcal{H}_1, \mathcal{H}_2$ be Hilbert spaces with $\dim(\mathcal{H}_1) \leq \dim(\mathcal{H}_2)$ and $V : \mathcal{H}_1 \to \mathcal{H}_2$ an isometry. Let A and B be unitaries on $\mathcal{H}_1$ and $\mathcal{H}_2$, respectively, $\psi_1 \in \mathrm{Pos}(\mathcal{H}_1)$, $\psi_2 \in \mathrm{Pos}(\mathcal{H}_2)$, and $\varepsilon \geq 0$. Then:*

$$\mathrm{Tr}\Big[V^\dagger B^\dagger V A \psi_1\Big] \approx_\varepsilon \mathrm{Tr}[\psi_1] \implies V^\dagger B V \approx_{\varepsilon, \psi_1} A, \tag{2.6}$$

$$\mathrm{Tr}\Big[V A^\dagger V^\dagger B \psi_2\Big] \approx_\varepsilon \mathrm{Tr}[\psi_2] \implies V A V^\dagger \approx_{\varepsilon, \psi_2} B. \tag{2.7}$$

We will also frequently use this lemma in the simpler case where $V = \mathbb{1}$.

Remark 2.8. In [MV20, Lemma 2.16], this lemma is only stated for the case where A and B are binary observables (i.e. are Hermitian in addition to being unitary), but it is easy to check that the proof of [MV20, Lemma 2.16] also goes through in the more general case stated here.

Lemma 2.9 (Replacement lemma, [MV20, Lemma 2.21]). *Let $\psi \in \mathrm{Pos}(\mathcal{H})$, and $A, B, C \in \mathcal{L}(\mathcal{H})$. If $A \approx_{\varepsilon, \psi} B$ and $\|C\|_\infty = O(1)$, then*

$$\mathrm{Tr}[CA\psi] \approx_{\varepsilon^{1/2}} \mathrm{Tr}[CB\psi], \tag{2.8}$$

$$\mathrm{Tr}[AC\psi] \approx_{\varepsilon^{1/2}} \mathrm{Tr}[BC\psi]. \tag{2.9}$$

3 Protocol description and completeness

In this section, we introduce our protocol for classically instructed parallel remote state preparation. The formal description of the protocol is given as Protocol 3. On a high-level, Protocol 3 works as follows: the verifier first runs a number of *test rounds* (Protocol 1), where the prover is asked to measure its entire quantum state. These test rounds are used by the verifier to check whether the prover behaves as intended. Once the verifier is convinced of this, the verifier runs a preparation round (Protocol 2). Test and preparation rounds are indistinguishable from the point of view of the prover, except that unlike in a test round, in a preparation the prover is not asked to measure its final state. Instead, the final state can then be used in another protocol, as we will do in Part II.

The main guarantee achieved by our protocol is the following: if the prover passed the test rounds with high probability, then the verifier knows that (up to a change of basis) the prover must have prepared n parallel random BB84 states. The formal soundness theorem is given in Theorem 4.33. Before we turn to the soundness proof, we show completeness, i.e., that an honest prover can succeed in our protocol with overwhelming probability.

Proposition 3.1. *There exists an efficient quantum prover that is accepted in Protocol 3 with probability negligibly close to 1 in the security parameters λ (for parameter choices n at most polynomial in λ and δ at least inverse polynomial in λ). Furthermore, the final state of such a prover at the end of Protocol 3 is*

$$\bigotimes_{i \in [n]} H^{\theta_i} |v_i\rangle\langle v_i| H^{\theta_i} \,, \tag{3.1}$$

where $\vec{v}$ and $\vec{\theta}$ are the strings recorded by the verifier in Protocol 3.

Proof. We describe the honest behaviour. An honest prover behaves the same in each of the test rounds in step 2. of Protocol 3. Hence, to show that an honest prover succeeds in Protocol 3 with probability negligibly close to 1, it suffices to describe an honest strategy for Protocol 1 that succeeds with probability negligibly close to 1.[8] In Protocol 1, the prover receives n keys $k_1, \ldots, k_n$ and returns answers for each key k_i individually. Since the verifier's checks are independent for each i, we only need to describe an honest procedure for one key k_i that succeeds in the verifier's checks for that i with probability negligibly close to 1.

The honest behaviour for a single key k_i is essentially the same as in [GV19, Lemma 3.4] and [MV20, Proposition 3.1]. For the sake of completeness, we spell out the details, but note that most of the proof is simply copied from [MV20, Proposition 3.1].

Given a key $k_i \in \mathcal{K}_\theta$, the prover prepares the state

$$\frac{1}{\sqrt{2 \cdot |\mathcal{X}|}} \sum_{b \in \{0,1\}} \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \sqrt{f_{k_i,b}(x)(y)} |b\rangle |x\rangle |y\rangle \,.$$

Preparing this state can be efficiently done (up to negligible error) using the SAMP procedure from the definition of ENTCF families ([BCM⁺18, Definition 3.1] and [Mah18, Definition 4.2]). The prover then measures the "image register" (i.e., the register that store y) to obtain an image $y_i \in \mathcal{Y}$ and sends this back

---

[8]In fact, it would suffice to have an honest strategy for Protocol 1 that succeeds with probability $1 - \delta/2$. Then, as long as we choose $N = \Omega(\lambda^c)$ for some $c > 0$, a standard application of Hoeffding's inequality shows success in Protocol 3 with probability negligibly close to 1.

Protocol 1. Test round protocol.

Let $\lambda \in \mathbb{N}$ be the security parameter, $(\mathcal{F}, \mathcal{G})$ an ENTCF family, and $n = \mathrm{poly}(\lambda)$ the number of BB84 states that the verifier wishes to prepare.

1. The verifier selects a uniformly random basis $\theta \xleftarrow{\$} \{0, 1\}$, where 0 corresponds to the computational and 1 to the Hadamard basis.

2. The verifier samples keys and trapdoors $(k_1, t_{k_1}; \dots k_n, t_{k_n})$ by computing $(k_i, t_{k_i}) \leftarrow \mathrm{GEN}_{\mathcal{K}_\theta}(1^\lambda)$. The verifier then sends $(k_1, \dots k_n)$ to the prover (but keeps the trapdoors t_{k_i} private).

3. The verifier receives $(y_1, \dots, y_n) \in \mathcal{Y}^{\times n}$ from the prover.

4. The verifier selects a round type $\in \{\text{preimage round, Hadamard round}\}$ uniformly at random and sends the round type to the prover.

   (i) For a *preimage round*: The verifier receives $(b_1, x_1; \dots b_n, x_n)$ from the prover, with $b_i \in \{0, 1\}$ and $x_i \in \mathcal{X}$. The verifier sets $\mathtt{flag} \leftarrow \mathtt{fail_{Pre}}$ if $\mathrm{CHK}(k_i, y_i, b_i, x_i) = 0$.

   (ii) For a *Hadamard round*: The verifier receives $d_1, \dots d_n \in \{0, 1\}^w$ from the prover (for some w depending on the security parameter). The verifier sends $q = \theta$ to the prover, and receives answers $v_1, \dots v_n \in \{0, 1\}$. The verifier performs the following checks:

Case	Verifier's check
$q = \theta = 0$	Set $\mathtt{flag} \leftarrow \mathtt{fail_{Had}}$ if $\hat{b}(k_i, y_i) \neq v_i$ for some i.
$q = \theta = 1$	Set $\mathtt{flag} \leftarrow \mathtt{fail_{Had}}$ if $\hat{u}(k_i, y_i, d_i) \neq v_i$.

Note. We denote the "question" separately by q (even though here we always have $q = \theta$) because when the variant of this protocol in Protocol 2 is used in the context of another cyrptographic task, the verifier can also send questions q which are different from θ.

18

Protocol 2. Preparation round protocol.

Let $\lambda \in \mathbb{N}$ be the security parameter, $(\mathcal{F}, \mathcal{G})$ an ENTCF family, and $n = \mathrm{poly}(\lambda)$ the number of BB84 states that the verifier wishes to prepare.

1. The verifier selects bases $\vec{\theta} \xleftarrow{\$} \{0,1\}^n$, where 0 corresponds to the computational and 1 to the Hadamard basis.

2. The verifier samples keys and trapdoors $(k_1, t_{k_1}; \ldots k_n, t_{k_n})$ by computing $(k_i, t_{k_i}) \leftarrow \mathrm{GEN}_{\mathcal{K}_{\theta_i}}(1^\lambda)$. The verifier then sends $(k_1, \ldots k_n)$ to the prover (but keeps the trapdoors t_{k_i} private).

3. The verifier receives $y_1, \ldots y_n \in \mathcal{Y}$ from the prover.

4. The verifier sends "Hadamard round" to the prover as the round type.

5. The verifier receives $d_1, \ldots d_n \in \{0,1\}^w$ from the prover (for some w depending on the security parameter). The verifier computes a string $\vec{v}$ according to

$$v_i = \begin{cases} \hat{b}(k_i, y_i) & \text{if } \theta_i = 0\,, \\ \hat{u}(k_i, y_i, d_i) & \text{if } \theta_i = 1\,. \end{cases}$$

Protocol 3. Multi-round protocol for preparation of BB84 states.

Let $\lambda \in \mathbb{N}$ be the security parameter, $(\mathcal{F}, \mathcal{G})$ an ENTCF family, $n = \mathrm{poly}(\lambda)$ the number of BB84 states that the verifier wishes to prepare, $N = M^2$ the maximum number of test rounds (for $M \in \mathbb{N}$), and δ an error tolerance parameter. For $j \in [M]$ we denote by $B_j = \{(j-1)M+1, \ldots, jM\}$ the j-th "block" of M rounds.

1. The verifier (privately) samples $S \xleftarrow{\$} \{0, \ldots, M-1\}$ (the number of M-round blocks of test rounds that will be performed).

2. The verifier performs SM executions of Protocol 1 with the prover. The verifier aborts if *for any $j \in [S]$*, the fraction of rounds in B_j for which $\texttt{flag} = \texttt{fail}_{\texttt{Pre}}$ or $\texttt{flag} = \texttt{fail}_{\texttt{Had}}$ exceeds δ.

3. The verifier (privately) samples $R \xleftarrow{\$} [M]$ and executes Protocol 1 with the prover $R-1$ times. Then, the verifier executes Protocol 2 with the prover and records the basis choice $\vec{\theta}$ and the string $\vec{v}$ from that execution.

to the verifier. The post-measurement for each i is

$$\begin{cases} |\hat{b}(k_i, y_i)\rangle |\hat{x}(k_i, y_i)\rangle & \text{if } k_i \in \mathcal{K}_0, \\ \frac{1}{\sqrt{2}} \left(|0\rangle |\hat{x}_0(k_i, y_i)\rangle + |1\rangle |\hat{x}_1(k_i, y_i)\rangle \right) & \text{if } k_i \in \mathcal{K}_1. \end{cases} \tag{3.2}$$

If the verifier selects a preimage round, the prover measures both registers in the computational basis and returns the result. From the states in Equation (3.2) it is clear that the prover succeeds with probability negligibly close to 1 in the preimage round.

If the verifier selects a Hadamard round, the prover measures the "x-register" in the Hadamard basis to obtain d_i and returns this to the verifier. We introduce the shorthand $b_i = \hat{b}(k_i, y_i)$ and $u_i = \hat{u}(k_i, y_i, d_i)$. At this point, the prover's state for each i is (up to a global phase)

$$\begin{cases} |b_i\rangle & \text{if } k_i \in \mathcal{K}_0, \\ |(-)^{u_i}\rangle & \text{if } k_i \in \mathcal{K}_1. \end{cases} \tag{3.3}$$

The prover now receives a question $q = \theta$ and measures the remaining qubit in the computational basis if $q = 0$ and in the Hadamard basis if $q_i = 1$. Then it is clear from the expression for the prover's remaining qubit in Equation (3.3) that the prover will pass the verifier's check.

Having described the honest behaviour for a test round, we now turn our attention to showing Equation (3.1). In a preparation round, the prover receives keys $k_1, \ldots, k_n$ with $k_i \in \mathcal{K}_{\theta_i}$. For each key k_i, an honest prover performs the same steps as described above for a test round.[9] Then, Equation (3.1) follows immediately from Equation (3.3). □

4 Protocol soundness

The purpose of this section is to prove that a prover that succeeds in Protocol 3 must have prepared a product of BB84 states without knowing which BB84 states it has prepared, which is formally stated as Theorem 4.33. This means that the situation at the end of Protocol 3 is essentially identical to one in which the verifier has sent random BB84 states to the prover via a quantum channel. We will make use of this in Part II.

The structure of the proof is as follows: a prover in Protocol 3 is subjected to multiple test rounds, i.e., executions of Protocol 1, by the verifier. We can model the behaviour of an arbitrary prover in a single test round as a *device*, defined in Section 4.1. The bulk of the soundness proof is concerned with showing that any such device that has a high success probability in Protocol 1 must have prepared (and subsequently measured) a product of BB84 states (Proposition 4.32). For this, we first show that certain "inefficient observables" associated with a device approximately satisfy the same commutation and anti-commutation relations as Pauli observables (Proposition 4.19). This allows us to conclude that these inefficient observables can be mapped to Pauli observables by an isometry (Lemma 4.28).[10] Finally, since we now know a lot about the structure of these inefficient observables, we can relate them to the device's actual (efficient) observables in a way that allows us to conclude that the efficient observables can be mapped to Pauli operators, too, albeit by a slightly different isometry (Proposition 4.29).

---

[9]In fact, due to the injective invariance property of ENTCF functions, the prover cannot distinguish keys in $\mathcal{K}_0$ from those in $\mathcal{K}_1$. Hence, any (potentially dishonest) prover does not know whether it is in a test or preparation round, so it must perform the same actions in both. This will play a central role in our soundness proof in Section 4.

[10]This can be viewed as a specific instantiation of a general stability theorem for approximate group representations due to Gowers and Hatami [GH17].

4.1 Modelling devices in Protocol 1

Definition 4.1 (Devices). A device $D = (S, \Pi, M, P)$ is specified by the following:

(i) A set $S = \{\psi^{(\vec{\theta})}\}_{\vec{\theta} \in \{0,1\}^n}$ of states $\psi^{(\vec{\theta})} \in \mathcal{D}(\mathcal{H}_D \otimes \mathcal{H}_Y)$, where $\dim(\mathcal{H}_Y) = |\mathcal{Y}|^n$ and the states are classical on $\mathcal{H}_Y$:

$$\psi^{(\vec{\theta})} = \sum_{\vec{y} \in \mathcal{Y}^n} \psi_{\vec{y}}^{(\vec{\theta})} \otimes |\vec{y}\rangle\langle\vec{y}|_Y. \tag{4.1}$$

In the context of Protocol 1, $\psi^{(\vec{\theta})}$ is the prover's state after returning $\vec{y}$ for the case where the verifier makes basis choices $\vec{\theta}$.[11] Each $\psi^{(\vec{\theta})}$ also implicitly depends on the specific keys chosen by the verifier (not just the basis choice $\vec{\theta}$); all the statements we make hold on average over key choices (for a fixed basis choice $\vec{\theta}$).

(ii) A projective measurement Π on $\mathcal{H}_D \otimes \mathcal{H}_Y$:

$$\Pi = \left\{ \Pi^{(\vec{b},\vec{x})} = \sum_{\vec{y}} \Pi_{\vec{y}}^{(\vec{b},\vec{x})} \otimes |\vec{y}\rangle\langle\vec{y}|_Y \right\}_{\vec{b} \in \{0,1\}^n; \, \vec{x} \in \mathcal{X}^n}. \tag{4.2}$$

This is the measurement used by the prover to compute his answer $(\vec{b}, \vec{x})$ in the preimage challenge.

(iii) A projective measurement M on $\mathcal{H}_D \otimes \mathcal{H}_Y$:

$$M = \left\{ M^{(\vec{d})} = \sum_{\vec{y}} M_{\vec{y}}^{(\vec{d})} \otimes |\vec{y}\rangle\langle\vec{y}|_Y \right\}_{\vec{d} \in \{0,1\}^{w \times n}}. \tag{4.3}$$

This is the measurement used by the prover to compute his answer $\vec{d}$ in the Hadamard challenge. We use an additional Hilbert spaces $\mathcal{H}_R$ to record the outcomes of measuring M and write the post-measurement state after applying M to $\psi^{(\vec{\theta})}$ as

$$\sigma^{(\vec{\theta})} := \sum_{\vec{y},\vec{d}} M_{\vec{y}}^{(\vec{d})} \psi_{\vec{y}}^{(\vec{\theta})} M_{\vec{y}}^{(\vec{d})} \otimes |\vec{y}, \vec{d}\rangle\langle\vec{y}, \vec{d}|_{YR}. \tag{4.4}$$

(iv) A set $P = \{P_q\}$, where for each $q \in \{0, 1\}$, P_q is a projective measurement on $\mathcal{H}_D \otimes \mathcal{H}_Y \otimes \mathcal{H}_R$:

$$P_q = \left\{ P_q^{(\vec{v})} = \sum_{\vec{y},\vec{d}} P_{q,\vec{y},\vec{d}}^{(\vec{v})} \otimes |\vec{y}, \vec{d}\rangle\langle\vec{y}, \vec{d}|_{YR} \right\}_{\vec{v} \in \{0,1\}}. \tag{4.5}$$

In the context of Protocol 1, given question q, the prover will measure $\{P_q^{(\vec{v})}\}$ and return the outcome $\vec{v}$ as his answer.

---

[11] In Protocol 1, the only two basis choices are $\vec{\theta} = \vec{0}$ and $\vec{\theta} = \vec{1}$. However, $\psi^{(\vec{\theta})}$ is still well-defined as the state that the prover (who is defined in terms of the quantum circuits he runs on a given input) would prepare if given keys of basis choice $\vec{\theta}$, even though this never occurs in Protocol 1. The reason for including all basis choices $\vec{\theta}$ is that in Protocol 2, the verifier makes a uniformly random basis choice $\vec{\theta}$, prompting the prover to use the state $\psi^{(\vec{\theta})}$.

Definition 4.2 (Efficient devices). A device is called *efficient* if the states $\psi^{(\vec\theta)}$ can be prepared efficiently and the measurements Π, M, and P_{q_1,q_2} can be performed efficiently.

Remark 4.3. Throughout this paper, whenever we say that a procedure is "efficient" (e.g. in Definition 4.2, or in proofs when constructing adversaries that break a property of ENTCF families) we mean a quantum polynomial-time procedure *with advice*, i.e. in addition to the input, the procedure also has access to a quantum state (the "advice state") which only depends on the input length, but not the input itself. This advice state need not be efficiently preparable itself. The additional advice state plays no role in the single-round soundness analysis (i.e. the entire soundness proof up to Theorem 4.33) and hence we do not write it explicitly. The reason the advice state is required at all is that if we want to model a prover in Protocol 3 as a sequence of (single-round) devices of the kind in Definition 4.1, we need to account for the fact that the prover in round $i + 1$ has access to the final state after round i, which we can view as an advice state for the device modelling the $(i + 1)$-th round of the prover. We note that the same convention is (implicitly) used in [BCM$^+$18, GV19], and that as a result these works and our paper require that the LWE problem is hard to solve quantumly with quantum advice (see [BCM$^+$18, Definition 2.5] for the precise statement).

Definition 4.4 (Observables). For a device $D = (S, \Pi, M, P)$ with projective measurements as in Definition 4.1, we define the following binary observables:

$$Z_i = \sum_{\vec v}(-1)^{v_i}P_0^{(\vec v)}\,, \tag{4.6}$$

$$X_i = \sum_{\vec v}(-1)^{v_i}P_1^{(\vec v)}\,, \tag{4.7}$$

$$\tilde X_i = \sum_{\vec v,\vec y,\vec d}(-1)^{v_i+\hat u(k_i,y_i,d_i)}P_{1,\vec y,\vec d}^{(\vec v)}\otimes|\vec y,\vec d\rangle\langle\vec y,\vec d|_{YR}\,. \tag{4.8}$$

We further use the following notation for products of observables: for $\vec a \in \{0,1\}^n$, we define

$$Z(\vec a) := Z_1^{a_1}\dots Z_n^{a_n} = \sum_{\vec v}(-1)^{\vec a\cdot\vec v}P_0^{(\vec v)}\,, \tag{4.9}$$

and likewise for $X(\vec a)$ and $\tilde X(\vec a)$. Note that it directly follows from the definition that

$$\tilde X_{i,\vec y,\vec d} = (-1)^{\hat u(k_i,y_i,d_i)}X_{i,\vec y,\vec d}\quad\text{and}\quad \tilde X(\vec a)_{\vec y,\vec d} = (-1)^{\vec a\cdot\hat u(\vec k,\vec y,\vec d)}X(\vec a)_{\vec y,\vec d}. \tag{4.10}$$

Remark 4.5. It is easy to check that the operators defined above are indeed binary observables using that $\{P_{q,\vec y,\vec d}^{(\vec v)}\}_{\vec v}$ are projective measurements.

Definition 4.6 (Partial post-measurement states). For $k \in \mathcal K_0 \cup \mathcal K_1$ and $v \in \{0,1\}$ define the set $V_{k,v} \subseteq \mathcal Y \times \{0,1\}^w$ by the following condition:

$$(y,d) \in V_{k,v} \iff \begin{cases}\hat b(k,y) = v & \text{if } k \in \mathcal K_0\,,\\ \hat u(k,y,d) = v & \text{if } k \in \mathcal K_1\,.\end{cases} \tag{4.11}$$

Then for $\vec k, \vec\theta, \vec v$ define

$$\sigma^{(\vec\theta,\vec v)} = \sum_{y_1,d_1\in V_{k_1,v_1}}\cdots\sum_{y_n,d_n\in V_{k_n,v_n}}\sigma_{\vec y,\vec d}^{(\vec\theta)}\otimes|\vec y,\vec d\rangle\langle\vec y,\vec d|\,. \tag{4.12}$$

Further for $\vec a \in \{0,1\}^n$ we define

$$\sigma^{(\vec\theta,v,\vec a)} = \sum_{\vec v:\,\vec v\cdot\vec a=v}\sigma^{(\vec\theta,\vec v)}\,. \tag{4.13}$$

22

To get an intuition for this definition, consider the case $\theta = \vec{0}$. Then for any $\vec{a} \in \{0,1\}^n$, $\sigma^{(\vec{0},v,\vec{a})}$ is that part of the state $\sigma^{(\vec{0})}$ for which the honest device would receive outcome v when measuring the observable $Z(\vec{a})$.

Definition 4.7 (Success probabilities). For any device $D = (S, \Pi, M, P)$ we define $\gamma_P(D)$ as the device's failure probability in a preimage round, and $\gamma_H(D)$ as the failure probability in a Hadamard round in Protocol 1:

$$\gamma_P(D) = \Pr[\texttt{flag} = \texttt{fail}_{\texttt{Pre}} \mid \text{round type} = \text{preimage round}] , \tag{4.14}$$

$$\gamma_H(D) = \Pr[\texttt{flag} = \texttt{fail}_{\texttt{Had}} \mid \text{round type} = \text{Hadamard round}] . \tag{4.15}$$

The following lemma tells us that for any device that has a non-negligible failure probability in the preimage test, there is another device that is "close" to the original one in the sense that its measurements are the same as for the original device and its states only differ by $O(\gamma_P(D))$, but that has a negligible failure probability in the preimage test. A device with a negligible failure probability in the preimage test is called *perfect*. This means that for the soundness proof, it suffices to only consider perfect devices: for any arbitrary device, we can first switch to the corresponding perfect device at the cost of incurring an approximation error of $O(\gamma_P(D))$, and then apply our soundness proof to the perfect device. We omit the proof of Lemma 4.9 as it is essentially identical to that of [MV20, Lemma 4.13].[12]

Definition 4.8 (Perfect device). We call a device D *perfect* if $\gamma_P(D) = \mathrm{negl}(\lambda)$.

Lemma 4.9. *Let $D = (S, \Pi, M, P)$ be an efficient device with $\gamma_P(D) < 1$, where $S = \left\{\psi^{(\vec{\theta})}\right\}$. Then there exists an efficient perfect device $D' = (S', \Pi, M, P)$, which uses the same measurements Π, M, P and whose states $S' = \left\{\psi'^{(\vec{\theta})}\right\}$ satisfy for any $\vec{\theta} \in \{0,1\}^n$:*

$$\psi'^{(\vec{\theta})} \approx_{\gamma_P(D)} \psi^{(\vec{\theta})} . \tag{4.16}$$

The following lemma shows what outcomes a successful device must produce when measuring the observables from Definition 4.4 on the partial post-measurement states from Definition 4.6.

Lemma 4.10. *For any efficient device $D = (S, \Pi, M, P)$, string $\vec{a} \in \{0,1\}^n$, and bit $v \in \{0,1\}$:*

$$\mathrm{Tr}\left[Z(\vec{a})^{(v)}\sigma^{(\vec{0},v,\vec{a})}\right] \approx_{\gamma_H(D)} \mathrm{Tr}\left[\sigma^{(\vec{0},v,\vec{a})}\right] , \tag{4.17}$$

$$\mathrm{Tr}\left[X(\vec{a})^{(v)}\sigma^{(\vec{1},v,\vec{a})}\right] \approx_{\gamma_H(D)} \mathrm{Tr}\left[\sigma^{(\vec{1},v,\vec{a})}\right] , \tag{4.18}$$

$$\mathrm{Tr}\left[\tilde{X}(\vec{a})\sigma^{(\vec{1})}\right] \approx_{\gamma_H(D)} 1 . \tag{4.19}$$

Proof. We first prove Equation (4.17). Since the case $q = \theta = 0$ occurs with probability $1/2$ in the protocol in Protocol 1, the device's failure probability in this case can be at most $2\gamma_H(D)$. Furthermore, since the device only succeeds if $v_i = \hat{b}(k_i, y_i)$ for all $i \in [n]$ in the protocol, it is in particular the case that for any

---

[12]To avoid confusion, we point out that [MV20, Lemma 4.13] phrases the approximation in terms of trace distance, but Lemma 4.9 uses the $\approx$-notation for states, which corresponds to the square of the trace distance (see Definition 2.6). This is why we have $\gamma_P(D)$ in Lemma 4.9, not $\gamma_P(D)^{1/2}$ as in [MV20, Lemma 4.13].

$\vec{a} \in \{0,1\}^n$, $\vec{a} \cdot \hat{b}(\vec{k}, \vec{y}) = \vec{a} \cdot \vec{v}$ with probability at least $1 - 2\gamma_H(D)$. Now comparing the definition of $\sigma^{(\vec{0},v,\vec{a})}$ with the verifier's checks in the protocol, this means that

$$\sum_{v' \in \{0,1\}} \text{Tr}\left[Z(\vec{a})^{(v')} \sigma^{(\vec{0},v',\vec{a})}\right] \geq 1 - 2\gamma_H(D). \tag{4.20}$$

Since $Z(\vec{a})^{(v)}$ is a projector and $\sigma^{(\vec{0})} = \sigma^{(\vec{0},v,\vec{a})} + \sigma^{(\vec{0},v \oplus 1,\vec{a})}$ is a normalised state,

$$\text{Tr}\left[Z(\vec{a})^{(v \oplus 1)} \sigma^{(\vec{0},v \oplus 1,\vec{a})}\right] \leq \text{Tr}\left[\sigma^{(\vec{0},v \oplus 1,\vec{a})}\right] = 1 - \text{Tr}\left[\sigma^{(\vec{0},v,\vec{a})}\right]. \tag{4.21}$$

Inserting this into Equation (4.20), we obtain

$$1 - 2\gamma_H(D) \leq \text{Tr}\left[Z(\vec{a})^{(v)} \sigma^{(\vec{0},v,\vec{a})}\right] + 1 - \text{Tr}\left[\sigma^{(\vec{0},v,\vec{a})}\right], \tag{4.22}$$

which implies

$$\text{Tr}\left[Z(\vec{a})^{(v)} \sigma^{(\vec{0},v,\vec{a})}\right] \geq \text{Tr}\left[\sigma^{(\vec{0},v,\vec{a})}\right] - 2\gamma_H(D). \tag{4.23}$$

For the inequality in the other direction, we note that since $Z(\vec{a})^{(v)}$ is a projector, we immediately have $\text{Tr}\left[Z(\vec{a})^{(v)} \sigma^{(\vec{0},v,\vec{a})}\right] \leq \text{Tr}\left[\sigma^{(\vec{0},v,\vec{a})}\right]$, finishing the proof of Equation (4.17).

The proof of Equation (4.18) is completely analogous. Finally, we need to show Equation (4.19). For this, observe that by definition $\tilde{X}(\vec{a})$ returns outcome 0 if the device's answers satisfy $\vec{a} \cdot \vec{v} = \vec{a} \cdot \hat{u}(\vec{k}, \vec{y}, \vec{d})$. Since this is implied by the checks performed by the verifier in the protocol, we have

$$\text{Tr}\left[\tilde{X}^{(0)} \sigma^{(\vec{1})}\right] \approx_{\gamma_H(D)} 1. \tag{4.24}$$

Since $\tilde{X}^{(0)} + \tilde{X}^{(1)} = \mathbb{1}$, this implies $\text{Tr}\left[\tilde{X}^{(1)} \sigma^{(\vec{1})}\right] \approx_{\gamma_H(D)} 0$. The lemma follows because $\tilde{X} = \tilde{X}^{(0)} - \tilde{X}^{(1)}$. $\qquad\square$

4.2 Extending Lemma 4.10 to different basis choices

In [Mah18], it was shown that keys sampled from $\mathcal{K}_0$ and $\mathcal{K}_1$ (according to the corresponding sampling procedures, see [Mah18, Definition 4.3] for details) are computationally indistinguishable (up to negligible advantage). This property is called *injective invariance*. In this section, we will use this property to extend the statement of Lemma 4.10 to basis choices $\vec{\theta}$ other than $\vec{\theta} = \vec{0}$ and $\vec{\theta} = \vec{1}$. For this, we need a slight strengthening of injective invariance to multiple keys, which follows almost immediately from the original injective invariance property.

Lemma 4.11. *For any two $\vec{\theta}, \vec{\theta}' \in \{0,1\}^n$, no efficient distinguisher has non-negligible advantage in the following distinguishing game: the challenger samples a bit $b \xleftarrow{\$} \{0,1\}$. The challenger then samples keys $\vec{k}$ with $k_i \in \mathcal{K}_{\theta_i}$ if $b = 0$ and $k_i \in \mathcal{K}_{\theta_i'}$ if $b = 1$, and sends $\vec{k}$ to the distinguisher. The distinguisher provides a guess for the bit b.*

Proof. First suppose that $\vec{\theta}$ and $\vec{\theta}'$ only differ in one position $\theta_i \neq \theta_i'$. For convenience, we can assume that $\theta_i = 0$ and $\theta_i' = 1$ (and swap the labelling of $\vec{\theta}$ and $\vec{\theta}'$ if this is not the case). Consider an efficient distinguisher $\mathcal{D}$ for the distinguishing game in the lemma. From such a distinguisher $\mathcal{D}$ we can construct a

distinguisher $\mathcal{D}'$ for the original injective invariance game: $\mathcal{D}'$ is given a key $k_i \in \mathcal{K}_0 \cup \mathcal{K}_1$, samples $n - 1$ further keys $k_j \in \mathcal{K}_{\theta_j}$ for $j \in [n] \setminus \{i\}$, and runs $\mathcal{D}$ on the keys $\vec{k} = (k_1, \ldots, k_n)$. If $\mathcal{D}$ guesses $b = 0$, then $\mathcal{D}'$ guesses $k \in \mathcal{K}_0$, otherwise $\mathcal{D}'$ guesses $k \in \mathcal{K}_1$. From the construction it is clear that $\mathcal{D}'$ guesses correctly if and only if $\mathcal{D}$ guesses correctly. However, from the injective invariance property of the ENTCF family, no such $\mathcal{D}'$ can succeed with non-negligible advantage. Hence, $\mathcal{D}$ also cannot succeed with non-negligible advantage in the distinguishing game from the lemma.

We can extend this proof to any $\vec{\theta}, \vec{\theta}' \in \{0, 1\}^n$ by a simple hybrid argument: for any $\vec{\theta}, \vec{\theta}' \in \{0, 1\}^n$, we can find $\vec{\theta}^1, \ldots, \vec{\theta}^n$ with $\vec{\theta}^1 = \vec{\theta}$ and $\vec{\theta}^n = \vec{\theta}'$ such that $\vec{\theta}^i$ and $\vec{\theta}^{i+1}$ differ on at most one position. Applying the above argument in each step of the hybrid argument, the advantage in distinguishing $\vec{\theta}$ from $\vec{\theta}'$ can be at most $n \cdot \mathrm{negl}(\lambda)$, which is still negligible in λ since $n = \mathrm{poly}(\lambda)$. $\qquad\square$

Lemma 4.12. *Consider an efficient device $D = (S, \Pi, M, P)$. Then, for any $\vec{\theta}, \vec{\theta}'$:*

$$\sigma^{(\vec{\theta})} \overset{c}{\approx}_0 \sigma^{(\vec{\theta}')} . \tag{4.25}$$

Proof. Since the device D (and hence the preparation of the states $\sigma^{(\vec{\theta})}$) is efficient, this follows immediately from Lemma 4.11. $\qquad\square$

The fact that the states $\sigma^{(\vec{\theta})}$ are computationally indistinguishable allows us to extend relations between operators that hold on a particular $\sigma^{(\vec{\theta})}$ to other states $\sigma^{(\vec{\theta}')}$. This will be formalised by Lemma 4.13. The reason that the lemma does not just deal with efficient observables, but also observables which are efficient only with access to some of the trapdoors, is that the observables $\tilde{X}(\vec{a})$ (as defined in Definition 4.4) are only efficient if one has access to all trapdoors t_i for i s.t. $a_i = 1$. Hence, the following lemma will also be applicable to operator relations involving $\tilde{X}(\vec{a})$ (provided the constraint on $\vec{\theta}$ and $\vec{\theta}'$ stated in the lemma is satisfied).

Lemma 4.13. *Consider an efficient device $D = (S, \Pi, M, P)$ and a subset of indices $\mathcal{I} \subset [n]$. Suppose that for any keys $\vec{k} = (k_1, \ldots, k_n)$ with corresponding trapdoors $\vec{t} = (t_1, \ldots, t_n)$, A and B are observables which can be efficiently implemented with access to the trapdoors $\{t_i\}_{i \in \mathcal{I}}$. Then for any $\vec{\theta}, \vec{\theta}' \in \{0, 1\}^n$ such that $\theta_i = \theta_i'$ for all $i \in \mathcal{I}$,*

$$A \approx_{\varepsilon, \sigma^{(\vec{\theta})}} B \iff A \approx_{\varepsilon, \sigma^{(\vec{\theta}')}} B , \tag{4.26}$$

and

$$\mathrm{Tr}\left[A \sigma^{(\vec{\theta})} \right] \approx_0 \mathrm{Tr}\left[A \sigma^{(\vec{\theta}')} \right] . \tag{4.27}$$

Proof. We first prove Equation (4.26). Fix $\vec{\theta}, \vec{\theta}' \in \{0, 1\}^n$. By Definition 2.6, it suffices to show that

$$\mathrm{Tr}\left[(A - B)^\dagger (A - B) \sigma^{(\vec{\theta})} \right] \approx_0 \mathrm{Tr}\left[(A - B)^\dagger (A - B) \sigma^{(\vec{\theta}')} \right] . \tag{4.28}$$

Suppose for the sake of contradiction that there exists a positive non-negligible function $\mu(\lambda)$ such that

$$\mathrm{Tr}\left[(A - B)^\dagger (A - B) \sigma^{(\vec{\theta})} \right] - \mathrm{Tr}\left[(A - B)^\dagger (A - B) \sigma^{(\vec{\theta}')} \right] > \mu(\lambda) . \tag{4.29}$$

(For the other case where this difference is smaller than a negative non-negligible function the proof is analogous.) We want to show that this allows us to break the extended injective invariance property of

Lemma 4.11. For this, consider the injective invariance game of Lemma 4.11 on the indices $[n] \setminus \mathcal{I}$: a challenger chooses $b \xleftarrow{\$} \{0, 1\}$ samples a set of keys $\{k_i\}_{i \in [n] \setminus \mathcal{I}}$, where $k_i \in \mathcal{K}_{\theta_i}$ if $b = 0$ and $k_i \in \mathcal{K}_{\theta'_i}$ if $b = 1$. The keys $\{k_i\}_{i \in [n] \setminus \mathcal{I}}$ are sent to a distinguisher $\mathcal{D}$, who is asked to provide a guess for b.

The distinguisher $\mathcal{D}$ proceeds as follows: $\mathcal{D}$ samples keys $k_j \in \mathcal{K}_{\theta_j}$ for $j \in \mathcal{I}$ along with the corresponding trapdoors t_j. $\mathcal{D}$ then runs the device's operations on the keys $\vec{k} = (k_1, \ldots, k_n)$ and will obtain a state σ, with $\sigma = \sigma^{(\vec{\theta})}$ if $b = 0$ and $\sigma = \sigma^{(\vec{\theta}')}$ if $b = 1$.

By [MV20, Lemma 2.6], there exists an efficient procedure that given the state σ and the trapdoors $\{t_j\}_{j \in \mathcal{I}}$ (so that for this procedure, A and B are efficient binary observables) produces a bit b' with

$$\Pr[b' = 0 \mid \sigma] = \frac{1}{4} \mathrm{Tr}\left[(A - B)^\dagger (A - B)\sigma\right]. \tag{4.30}$$

$\mathcal{D}$ runs this procedure on the state σ it produced using the device's operations and outputs b' as its guess for b.

It is clear that $\mathcal{D}$ is efficient. Its distinguishing advantage is given by

$$\Pr[\mathcal{D} \text{ guesses } 0 \mid b = 0] - \Pr[\mathcal{D} \text{ guesses } 0 \mid b = 1] \tag{4.31}$$

$$= \Pr[b' = 0 \mid b = 0] - \Pr[b' = 0 \mid b = 1] \tag{4.32}$$

$$= \Pr\left[b' = 0 \mid \sigma^{(\vec{\theta})}\right] - \Pr\left[b' = 0 \mid \sigma^{(\vec{\theta}')}\right] \tag{4.33}$$

$$= \frac{1}{4} \mathrm{Tr}\left[(A - B)^\dagger (A - B)\sigma^{(\vec{\theta})}\right] - \frac{1}{4} \mathrm{Tr}\left[(A - B)^\dagger (A - B)\sigma^{(\vec{\theta}')}\right] \tag{4.34}$$

$$> \frac{1}{4} \mu(\lambda), \tag{4.35}$$

which is non-negligible by assumption, yielding the desired contradiction.

For Equation (4.27), the proof is analogous, except that instead of using the procedure from [MV20, Lemma 2.6] to estimate $\mathrm{Tr}\left[(A - B)^\dagger (A - B)\sigma\right]$, in this case the distinguisher can simply measure the observable A on the state it has produced by running the device's operations. $\square$

We can apply this lemma to extend Equation (4.19) to states other than $\sigma^{(\vec{1})}$:

Lemma 4.14. *For any efficient device $D = (S, \Pi, M, P)$ and strings $\vec{a}, \vec{\theta} \in \{0, 1\}^n$ such that $a_i = 1 \implies \theta_i = 1$:*

$$\mathrm{Tr}\left[\tilde{X}(\vec{a})\sigma^{(\vec{\theta})}\right] \approx_{\gamma_H(D)} 1. \tag{4.36}$$

Proof. We define $\mathcal{I} = \{i \in [n] \mid a_i = 1\}$. The observable $\tilde{X}(a)$ is efficient if one has access to the trapdoors $\{t_i\}_{i \in \mathcal{I}}$. Since further $\theta_i = 1$ for all $i \in \mathcal{I}$ by assumption, the lemma follows from Equation (4.19) by means of Lemma 4.13. $\square$

Using Lemma 2.7, we can immediately obtain the following corollary:

Corollary 4.15. *For any efficient device $D = (S, \Pi, M, P)$ and strings $\vec{a}, \vec{\theta} \in \{0, 1\}^n$ such that $a_i = 1 \implies \theta_i = 1$:*

$$\tilde{X}(\vec{a}) \approx_{\gamma_H(D), \sigma^{(\vec{\theta})}} \mathbb{1}. \tag{4.37}$$

We still need to extend Equation (4.17) and Equation (4.18) to different choices of θ. For this, we need to strengthen the statement of Lemma 4.12 and show that certain partial post-measurement states are also indistinguishable.

Lemma 4.16. *For an efficient device* $D = (S, \Pi, M, P)$ *and strings* $\vec{b}, \vec{\theta}^0, \vec{\theta}^1 \in \{0,1\}^n$ *such that* $b_i = 1 \implies \theta_i^0 = \theta_i^1$ *the following holds for any* $v \in \{0,1\}$:

$$\sigma^{(\vec{\theta}^0, v, \vec{b})} \approx_0^c \sigma^{(\vec{\theta}^1, v, \vec{b})}. \tag{4.38}$$

Proof. Fix v and $\vec{b}$. As a first step, we note that it suffices to show the lemma for the case where $\vec{\theta}^0$ and $\vec{\theta}^1$ only differ on one bit by the same argument as in the proof of Lemma 4.11.

Hence consider $\vec{\theta}^0, \vec{\theta}^1$ that only differ on one bit i s.t. $b_i = 0$. Without loss of generality, we can assume $\theta_i^0 = 0$ and $\theta_i^1 = 1$. Further consider an efficient distinguisher $\mathcal{D}$ that, given $\sigma^{(\vec{\theta}^c, v, \vec{b})}$, guesses $c \in \{0,1\}$. Let $\Delta^{\mathcal{D}}$ be the distinguishing advantage of $\mathcal{D}$.

Using $\mathcal{D}$, we construct the following adversary $\mathcal{A}$ for the injective invariance game. $\mathcal{A}$ is given a uniformly random key $k \in \mathcal{K}_0 \cup \mathcal{K}_1$ and asked to guess $\theta^* \in \{0,1\}$ such that $k \in \mathcal{K}_{\theta^*}$. $\mathcal{A}$ sets $k_i := k$ and samples keys $k_j \in \mathcal{K}_{\theta_j}$ for $j \neq i$ (along with the corresponding trapdoors t_j). Then, $\mathcal{A}$ performs the device's operations with these keys, obtaining $\vec{y}$ and $\vec{d}$ in the process. Since $\mathcal{A}$ knows t_j for all $j \in [n]$ for which $b_j = 1$, it can efficiently compute

$$v' = \sum_{j \text{ s.t. } \theta_j = 0} b_j \, \hat{b}(k_j, y_j) + \sum_{j \text{ s.t. } \theta_j = 1} b_j \, \hat{u}(k_j, y_j, d_j) \quad \mod 2. \tag{4.39}$$

If $v' \neq v$, $\mathcal{A}$ submits a uniformly random guess for θ^*. If $v' = v$, $\mathcal{A}$ runs the distinguisher $\mathcal{D}$ on the post-measurement state. $\mathcal{D}$ outputs $c' \in \{0,1\}$, and $\mathcal{A}$ guesses $\theta^* = c'$.

Let $\Delta^{\mathcal{A}}$ be the distinguishing advantage of $\mathcal{A}$. Since $\mathcal{D}$ is efficient, so is $\mathcal{A}$, so from the injective invariance property of ENTCF families we get that $\Delta^{\mathcal{A}} \leq \text{negl}(\lambda)$, so

$$\Pr[\mathcal{A} \text{ correct}] \leq \frac{1}{2} + \text{negl}(\lambda). \tag{4.40}$$

On the other hand,

$$\Pr[\mathcal{A} \text{ correct}] = \Pr[v = v'] \Pr[\mathcal{A} \text{ correct} \mid v = v'] + \Pr[v \neq v'] \Pr[\mathcal{A} \text{ correct} \mid v \neq v']. \tag{4.41}$$

In the case $v \neq v'$, $\mathcal{A}$ guesses uniformly at random , so $\Pr[\mathcal{A} \text{ correct} \mid v \neq v'] = 1/2$. In the case $v = v'$, by construction $\mathcal{A}$ guesses correctly if and only if $\mathcal{D}$ guesses correctly, so $\Pr[\mathcal{A} \text{ correct} \mid v = v'] = \frac{1}{2} + \Delta^{\mathcal{D}}$. Inserting this, we get

$$\Pr[\mathcal{A} \text{ correct}] = \Pr[v = v'] \left(\frac{1}{2} + \Delta^{\mathcal{D}} \right) + (1 - \Pr[v = v']) \frac{1}{2} \tag{4.42}$$

$$= \frac{1}{2} + \Pr[v = v'] \, \Delta^{\mathcal{D}}. \tag{4.43}$$

Now notice that

$$\Pr[v = v'] = \frac{1}{2} \text{Tr} \left[\sigma^{(\vec{\theta}^0, v, \vec{b})} \right] + \frac{1}{2} \text{Tr} \left[\sigma^{(\vec{\theta}^1, v, \vec{b})} \right] \tag{4.44}$$

$$\geq \frac{1}{2} \left\| \sigma^{(\vec{\theta}^0, v, \vec{b})} - \sigma^{(\vec{\theta}^1, v, \vec{b})} \right\|_1 \tag{4.45}$$

$$\geq \Delta^{\mathcal{D}}, \tag{4.46}$$

27

since the distinguishing advantage of any distinguisher is upper-bounded by the trace distance. Combining this with Equation (4.40), we get

$$\frac{1}{2} + \mathrm{negl}(\lambda) \geq \frac{1}{2} + \left(\Delta^{\mathcal{D}}\right)^2 . \tag{4.47}$$

Since the square root of a negligible function is again negligible, $\Delta^{\mathcal{D}} = \mathrm{negl}(\lambda)$ as desired. $\qquad \square$

Since $Z(\vec{a})$ and $X(\vec{b})$ are efficient observables, Lemma 4.16 immediately implies that we can extend Equation (4.17) and Equation (4.18) to different choices of $\vec{\theta}$:

Corollary 4.17. *Consider an efficient device $D = (S, \Pi, M, P)$ and a bit $v \in \{0,1\}$.*

(i) *For any $\vec{\theta}, \vec{a} \in \{0,1\}^n$ such that $a_i = 1 \implies \theta_i = 0$:*

$$\mathrm{Tr}\left[Z(\vec{a})^{(v)} \sigma^{(\vec{\theta},v,\vec{a})}\right] \approx_{\gamma_H(D)} \mathrm{Tr}\left[\sigma^{(\vec{\theta},v,\vec{a})}\right] . \tag{4.48}$$

(ii) *For any $\vec{\theta}, \vec{a} \in \{0,1\}^n$ such that $a_i = 1 \implies \theta_i = 1$:*

$$\mathrm{Tr}\left[X(\vec{a})^{(v)} \sigma^{(\vec{\theta},v,\vec{a})}\right] \approx_{\gamma_H(D)} \mathrm{Tr}\left[\sigma^{(\vec{\theta},v,\vec{a})}\right] . \tag{4.49}$$

Using [MV20, Lemma 2.19], this can also be restated as follows:

Corollary 4.18. *Consider an efficient device $D = (S, \Pi, M, P)$ and a bit $v \in \{0,1\}$.*

(i) *For any $\vec{\theta}, \vec{a} \in \{0,1\}^n$ such that $a_i = 1 \implies \theta_i = 0$:*

$$Z(\vec{a}) \approx_{\gamma_H(D), \sigma^{(\vec{\theta},v,\vec{a})}} (-1)^v \mathbb{1} . \tag{4.50}$$

(ii) *For any $\vec{\theta}, \vec{a} \in \{0,1\}^n$ such that $a_i = 1 \implies \theta_i = 1$:*

$$X(\vec{a}) \approx_{\gamma_H(D), \sigma^{(\vec{\theta},v,\vec{a})}} (-1)^v \mathbb{1} . \tag{4.51}$$

4.3 Pauli group relations for inefficient observables[13]

The goal of this section is to prove that the observables $\{\tilde{X}(\vec{a})\}_{\vec{a} \in \{0,1\}^n}$ and $\{Z(\vec{b})\}_{\vec{b} \in \{0,1\}^n}$ approximately satisfy the relations of the n-qubit Pauli observables. More formally, we will show the following proposition:

Proposition 4.19. *For any efficient perfect device $D = (S, \Pi, M, P)$ and any $\vec{a}, \vec{b} \in \{0,1\}^n$:*

$$\tilde{X}(\vec{a}) Z(\vec{b}) \approx_{n\gamma_H(D)^{1/4}, \sigma^{(\vec{1})}} (-1)^{\vec{a} \cdot \vec{b}} Z(\vec{b}) \tilde{X}(\vec{a}) . \tag{4.52}$$

The proof of this can be found at the end of this section. We first need to show a number of preparatory results.

---

[13] This section is based on unpublished work by Jeffrey Champion, John Wright, and the second author. We thank Jeffrey and John for allowing us to use these results here.

Lemma 4.20. *We define*

$$\Pi_i^{(b)} = \sum_{b_1,\dots,b_{i-1},b_{i+1},\dots,b_n} \sum_{\vec{x}} \Pi^{(\vec{b},\vec{x})}, \qquad (4.53)$$

where $\vec{b} = (b_1,\dots,b_{i-1},b,b_{i+1},\dots,b_n)$. For any efficient perfect device $D = (S,\Pi,M,P)$, the following holds for any i:

$$\sum_{b,\vec{d}} \left\| \left(M^{(\vec{d})}\Pi_i^{(b)} - Z_{i,\vec{d}}^{(b)} M^{(\vec{d})} \right) \left(\psi^{(\vec{1}^i)} \right)^{1/2} \right\|_2^2 \approx_{\gamma_H(D)} 0, \qquad (4.54)$$

where $\|\cdot\|_2$ denotes the Schatten 2-norm (also called Hilbert-Schmidt norm) and $\vec{1}^i \in \{0,1\}^n$ is the bitstring whose i-th bit is 1 and whose remaining bits are 0 (as defined in Section 2.1).

Proof. The proof is analogous to [MV20, Lemma 4.19], but we spell out the details for completeness. Writing out the definition of the 2-norm and multiplying out terms, we find that the l.h.s. of Equation (4.54) equals

$$\sum_{b,\vec{d}} \mathrm{Tr}\left[M^{(\vec{d})}\Pi_i^{(b)}\psi^{(\vec{1}^i)}\Pi_i^{(b)}M^{(\vec{d})} \right] + \sum_{b,\vec{d}} \mathrm{Tr}\left[Z_{i,\vec{d}}^{(b)}M^{(\vec{d})}\psi^{(\vec{1}^i)}M^{(\vec{d})}Z_{i,\vec{d}}^{(b)} \right]$$
$$- \sum_{b,\vec{d}} \mathrm{Tr}\left[M^{(\vec{d})}Z_{i,\vec{d}}^{(b)}M^{(\vec{d})} \left(\Pi_i^{(b)}\psi^{(\vec{1}^i)} + \psi^{(\vec{1}^i)}\Pi_i^{(b)} \right) \right] \quad (4.55)$$

Since $\{M^{(\vec{d})}\}_{\vec{d}}$, $\{\Pi_i^{(b)}\}_b$, and $\{Z_{i,\vec{d}}^{(b)}\}_b$ form projective measurements, the first two terms equal 1. For the third term, we note that since $\Pi_i^{(0)} + \Pi_i^{(1)} = \mathbb{1}$,

$$\Pi_i^{(b)}\psi^{(\vec{1}^i)} + \psi^{(\vec{1}^i)}\Pi_i^{(b)} = 2\Pi_i^{(b)}\psi^{(\vec{1}^i)}\Pi_i^{(b)} + \Pi_i^{(0)}\psi^{(\vec{1}^i)}\Pi_i^{(1)} + \Pi_i^{(1)}\psi^{(\vec{1}^i)}\Pi_i^{(0)}. \qquad (4.56)$$

Note that since $\{Z_{i,\vec{d}}^{(b)}\}_b$ and $\{M^{(\vec{d})}\}_{\vec{d}}$ are projective measurements, we have

$$\sum_{b,\vec{d}} \mathrm{Tr}\left[M^{(\vec{d})}Z_{i,\vec{d}}^{(b)}M^{(\vec{d})} \left(\Pi_i^{(0)}\psi^{(\vec{1}^i)}\Pi_i^{(1)} + \Pi_i^{(1)}\psi^{(\vec{1}^i)}\Pi_i^{(0)} \right) \right] = \mathrm{Tr}\left[\Pi_i^{(0)}\psi^{(\vec{1}^i)}\Pi_i^{(1)} + \Pi_i^{(1)}\psi^{(\vec{1}^i)}\Pi_i^{(0)} \right] = 0,$$

where the last equality holds because $\{\Pi_i^{(b)}\}_b$ are orthogonal projectors. Therefore, the third term in Equation (4.55) equals

$$2\sum_{b,\vec{d}} \mathrm{Tr}\left[Z_{i,\vec{d}}^{(b)}M^{(\vec{d})}\Pi_i^{(b)}\psi^{(\vec{1}^i)}\Pi_i^{(b)}M^{(\vec{d})} \right]. \qquad (4.57)$$

We now want to replace $\psi^{(\vec{1}^i)}$ by $\psi^{(\vec{0})}$ in the above expression. For this, observe that given any state ρ, we can efficiently estimate $\sum_{b,\vec{d}} \mathrm{Tr}\left[Z_{i,\vec{d}}^{(b)}M^{(\vec{d})}\Pi_i^{(b)}\psi^{(\vec{1}^i)}\Pi_i^{(b)}M^{(\vec{d})} \right]$ by measuring $\{\Pi_i^{(b)}\}_b$, $\{M^{(\vec{d})}\}_{\vec{d}}$, and $\{Z_{i,\vec{d}}^{(b)}\}_b$ in sequence and checking whether the Π_i- and Z_i-measurements yielded the same result. Therefore, since $\psi^{(\vec{1}^i)} \approx_0^c \psi^{(\vec{0})}$ by an argument analogous to Lemma 4.12, we have

$$\sum_{b,\vec{d}} \mathrm{Tr}\left[Z_{i,\vec{d}}^{(b)}M^{(\vec{d})}\Pi_i^{(b)}\psi^{(\vec{1}^i)}\Pi_i^{(b)}M^{(\vec{d})} \right] \approx_0 \sum_{b,\vec{d}} \mathrm{Tr}\left[Z_{i,\vec{d}}^{(b)}M^{(\vec{d})}\Pi_i^{(b)}\psi^{(\vec{0})}\Pi_i^{(b)}M^{(\vec{d})} \right]. \qquad (4.58)$$

29

This expression equals 1 if the Π_i- and Z_i-measurements yield the same result. Since in $\psi^{(\vec{0})}$ the basis choice for the i-th key is $\theta_i = 0$ and the device D is perfect, for any image y_i the Π_i-measurement will output the bit $\hat{b}(k_i, y_i)$ with probability negligibly close to 1. Hence, by Definition 4.6,

$$\sum_{\vec{d}} M^{(\vec{d})} \Pi_i^{(b)} \psi^{(\vec{0})} \Pi_i^{(b)} M^{(\vec{d})} \otimes |\vec{d}\rangle\langle\vec{d}| \approx_0 \sigma^{(\vec{0},b,\vec{1}^i)} . \tag{4.59}$$

With this and noting that $Z(\vec{1}^i) = Z_i$, we get from Lemma 4.10 that

$$\sum_{b,\vec{d}} \mathrm{Tr}\left[Z_{i,\vec{d}}^{(b)} M^{(\vec{d})} \Pi_i^{(b)} \psi^{(\vec{0})} \Pi_i^{(b)} M^{(\vec{d})} \right] \approx_0 \sum_{b} \mathrm{Tr}\left[Z_i^{(b)} \sigma^{(\vec{0},b,\vec{1}^i)} \right] \tag{4.60}$$

$$\approx_{\gamma_H(D)} \sum_{b} \mathrm{Tr}\left[\sigma^{(\vec{0},b,\vec{1}^i)} \right] \tag{4.61}$$

$$= \mathrm{Tr}\left[\sigma^{(\vec{0})} \right] \tag{4.62}$$

$$= 1, \tag{4.63}$$

concluding the proof. $\square$

Lemma 4.21. *For any efficient perfect device* $D = (S, \Pi, M, P)$, *the following holds for any* $i \in [n]$:

$$\mathrm{Tr}\left[Z_i \tilde{X}_i Z_i \sigma^{(\vec{1}^i)} \right] \approx_{\gamma_H(D)^{1/2}} -1 . \tag{4.64}$$

Proof. Using that Z_i is a binary observable, it is easy to check that

$$Z_i \tilde{X}_i Z_i = \left(2 \sum_{b} Z_i^{(b)} \tilde{X}_i Z_i^{(b)} \right) - \tilde{X}_i . \tag{4.65}$$

Since

$$\mathrm{Tr}\left[\tilde{X}_i \sigma^{(\vec{1}^i)} \right] \approx_{\gamma_H(D)} 1 \tag{4.66}$$

by Lemma 4.14, to show the lemma it suffices to show that

$$\sum_{b} \mathrm{Tr}\left[Z_i^{(b)} \tilde{X}_i Z_i^{(b)} \sigma^{(\vec{1}^i)} \right] \approx_{\gamma_H(D)^{1/2}} 0 . \tag{4.67}$$

Inserting the definition of $\sigma^{(\vec{1}^i)}$ from Equation (4.4), we get

$$\sum_{b} \mathrm{Tr}\left[Z_i^{(b)} \tilde{X}_i Z_i^{(b)} \sigma^{(\vec{1}^i)} \right] = \sum_{b,\vec{d}} \mathrm{Tr}\left[\tilde{X}_{i,\vec{d}} Z_{i,\vec{d}}^{(b)} M^{(\vec{d})} \psi^{(\vec{1}^i)} M^{(\vec{d})} Z_{i,\vec{d}}^{(b)} \right] \tag{4.68}$$

Using Lemma 4.20 and the Cauchy-Schwarz inequality, we can show that

$$\sum_{b,\vec{d}} \mathrm{Tr}\left[\tilde{X}_{i,\vec{d}} Z_{i,\vec{d}}^{(b)} M^{(\vec{d})} \psi^{(\vec{1}^i)} M^{(\vec{d})} Z_{i,\vec{d}}^{(b)} \right] \approx_{\gamma_H(D)^{1/2}} \sum_{b,\vec{d}} \mathrm{Tr}\left[\tilde{X}_{i,\vec{d}} M^{(\vec{d})} \Pi_i^{(b)} \psi^{(\vec{1}^i)} \Pi_i^{(b)} M^{(\vec{d})} \right] . \tag{4.69}$$

This is a standard step in state-dependent distance calculations, but we spell out the details for completeness in Lemma 4.22.

To prove the lemma, it now suffices to show that the r.h.s. of Equation (4.69) is negligible. For the sake of contradiction, suppose that there exists a non-negligible function $\mu(\lambda)$ such that

$$\sum_{b,\vec{d}} \text{Tr}\left[\tilde{X}_{i,\vec{d}} M^{(\vec{d})} \Pi_i^{(b)} \psi^{(\vec{1}^i)} \Pi_i^{(b)} M^{(\vec{d})}\right] > \mu(\lambda). \tag{4.70}$$

From this, we can construct the following adversary $\mathcal{A}$ for the adaptive hardcore bit property: $\mathcal{A}$ receives a key $k_i \in \mathcal{K}_1$ and samples keys $k_1, \ldots, k_{i-1}, k_{i+1}, k_n \in \mathcal{K}_0$. It then performs the device's operations with these keys to obtain $\psi^{(\vec{1}^i)}$. $\mathcal{A}$ now measures the projectors $\{\Pi_i^{(b,x)}\}_{b,x}$ and records the result (b, x), where

$$\Pi_i^{(b,x)} = \sum_{b_1,\ldots,b_{i-1},b_{i+1},\ldots,b_n} \sum_{x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n} \Pi^{(\vec{b},\vec{x})}. \tag{4.71}$$

$\mathcal{A}$ then measures $\{M^{(\vec{d})}\}_{\vec{d}}$ receiving an outcome $\vec{d}$, followed by $\{X_{i,\vec{d}}^{(c)}\}_c$, receiving an outcome c. $\mathcal{A}$ submits (b, x, d_i, c) as a guess for the adaptive hardcore bit property.

First note that since the device operations are efficient, $\mathcal{A}$ is efficient, too. To show that $\mathcal{A}$ breaks the adaptive hardcore bit property, we need to show that on average over $\vec{k}$, $\vec{y}$, and $\vec{d}$, $\hat{u}(k_i, y_i, d_i) = c$ with non-negligible advantage.

For this, we observe that since the device D is perfect, if the adversary measures (b, x) as described above, conditioned on receiving the bit b, the preimage x it obtains will be $x_b(k_i, y_i)$ with probability negligibly close to 1. As a consequence, by the gentle measurement lemma [Win99] the post-measurement state after measuring $\{\Pi_i^{(b,x)}\}_{b,x}$ is negligibly close to the post-measurement state after measuring $\{\Pi_i^{(b)}\}_b$ (i.e. it does not matter for the post-measurement state whether $\mathcal{A}$ also measures the preimage x or not):

$$\Pi_i^{(b,x)} \psi^{(\vec{1}^i)} \Pi_i^{(b,x)} \approx_0 \Pi_i^{(b)} \psi^{(\vec{1}^i)} \Pi_i^{(b)}. \tag{4.72}$$

In the next step, the adversary measures $\{M^{(\vec{d})}\}_{\vec{d}}$, followed by $\{X_{i,\vec{d}}^{(c)}\}_c$. The probability that $\mathcal{A}$ guesses correctly, i.e. that $c = \hat{u}(k_i, y_i, d_i)$, is then given by (up to negligible error from to the approximation $\sum_{b,x} \Pi_i^{(b,x)} \psi^{(\vec{1}^i)} \Pi_i^{(b,x)} \approx_0 \sum_b \Pi_i^{(b)} \psi^{(\vec{1}^i)} \Pi_i^{(b)}$)

$$\Pr[\mathcal{A} \text{ guesses correctly}] \approx_0 \sum_{b,\vec{y},\vec{d}} \text{Tr}\left[X_{i,\vec{y},\vec{d}}^{(\hat{u}(k_i,y_i,d_i))} M_{\vec{y}}^{(\vec{d})} \Pi_{i,\vec{y}}^{(b)} \psi_{\vec{y}}^{(\vec{1}^i)} \Pi_{i,\vec{y}}^{(b)} M_{\vec{y}}^{(\vec{d})}\right]. \tag{4.73}$$

Since $X_{i,\vec{y},\vec{d}}$ is a binary observable, we can express this as

$$\Pr[\mathcal{A} \text{ guesses correctly}] = \sum_{b,\vec{y},\vec{d}} \text{Tr}\left[\frac{1}{2}\left(\mathbb{1} + (-1)^{\hat{u}(k_i,y_i,d_i)} X_{i,\vec{y},\vec{d}}\right) M_{\vec{y}}^{(\vec{d})} \Pi_{i,\vec{y}}^{(b)} \psi_{\vec{y}}^{(\vec{1}^i)} \Pi_{i,\vec{y}}^{(b)} M_{\vec{y}}^{(\vec{d})}\right] \tag{4.74}$$

Using that $M_{\vec{y}}^{(\vec{d})}$ and $\Pi_{i,\vec{y}}^{(b)}$ form projective measurements:

$$= \frac{1}{2} + \sum_{b,\vec{y},\vec{d}} \text{Tr}\left[(-1)^{\hat{u}(k_i,y_i,d_i)} X_{i,\vec{y},\vec{d}} M_{\vec{y}}^{(\vec{d})} \Pi_{i,\vec{y}}^{(b)} \psi_{\vec{y}}^{(\vec{1}^i)} \Pi_{i,\vec{y}}^{(b)} M_{\vec{y}}^{(\vec{d})}\right] \tag{4.75}$$

By Definition 4.4, $(-1)^{\hat{u}(k_i, y_i, d_i)} X_{i, \vec{y}, \vec{d}} = \tilde{X}_{i, \vec{y}, \vec{d}}$:

$$= \frac{1}{2} + \sum_{b, \vec{y}, \vec{d}} \text{Tr}\left[\tilde{X}_{i, \vec{y}, \vec{d}} M_{\vec{y}}^{(\vec{d})} \Pi_{i, \vec{y}}^{(b)} \psi_{\vec{y}}^{(\bar{1}^i)} \Pi_{i, \vec{y}}^{(b)} M_{\vec{y}}^{(\vec{d})}\right] \tag{4.76}$$

$$= \frac{1}{2} + \sum_{b, \vec{d}} \text{Tr}\left[\tilde{X}_{i, \vec{d}} M^{(\vec{d})} \Pi_i^{(b)} \psi^{(\bar{1}^i)} \Pi_i^{(b)} M^{(\vec{d})}\right] . \tag{4.77}$$

Hence, Equation (4.70) would imply that the adversary has non-negligible advantage in guessing the adaptive hardcore bit, a contradiction. $\qquad \square$

Lemma 4.22. *With the setup and notation of Lemma 4.21, the following holds:*

$$\sum_{b, \vec{d}} \text{Tr}\left[\tilde{X}_{i, \vec{d}} Z_{i, \vec{d}}^{(b)} M^{(\vec{d})} \psi^{(\bar{1}^i)} M^{(\vec{d})} Z_{i, \vec{d}}^{(b)}\right] \approx_{\gamma_H(D)^{1/2}} \sum_{b, \vec{d}} \text{Tr}\left[\tilde{X}_{i, \vec{d}} M^{(\vec{d})} \Pi_i^{(b)} \psi^{(\bar{1}^i)} \Pi_i^{(b)} M^{(\vec{d})}\right] . \tag{4.78}$$

Proof. To show the lemma, there are two instances where we need to replace terms of the form $M^{(\vec{d})} Z_{i, \vec{d}}$ by $\Pi_i^{(b)} M^{(\vec{d})}$. As a first step, we will show that

$$\sum_{b, \vec{d}} \text{Tr}\left[\tilde{X}_{i, \vec{d}} Z_{i, \vec{d}} M^{(\vec{d})} \psi^{(\bar{1}^i)} \left(M^{(\vec{d})} Z_{i, \vec{d}} - \Pi_i^{(b)} M^{(\vec{d})}\right)\right] \approx_{\gamma_H(D)^{1/2}} 0, \tag{4.79}$$

with $\Pi_i^{(b)}$ defined as in Lemma 4.20. For this, we apply the triangle inequality and write the l.h.s. of Equation (4.79) as a Hilbert-Schmidt inner product. Then we can apply the Cauchy-Schwarz inequality twice:

$$\left|\sum_{b, \vec{d}} \text{Tr}\left[\tilde{X}_{i, \vec{d}} Z_{i, \vec{d}} M^{(\vec{d})} \psi^{(\bar{1}^i)} \left(M^{(\vec{d})} Z_{i, \vec{d}} - \Pi_i^{(b)} M^{(\vec{d})}\right)\right]\right| \tag{4.80}$$

$$\leq \sum_{b, \vec{d}} \left|\left\langle \left(Z_{i, \vec{d}} M^{(\vec{d})} - M^{(\vec{d})} \Pi_i^{(b)}\right) \left(\psi^{(\bar{1}^i)}\right)^{1/2}, \tilde{X}_{i, \vec{d}} Z_{i, \vec{d}} M^{(\vec{d})} \left(\psi^{(\bar{1}^i)}\right)^{1/2}\right\rangle\right| \tag{4.81}$$

$$\leq \sum_{b, \vec{d}} \left\|\tilde{X}_{i, \vec{d}} Z_{i, \vec{d}} M^{(\vec{d})} \left(\psi^{(\bar{1}^i)}\right)^{1/2}\right\|_2 \left\|\left(Z_{i, \vec{d}} M^{(\vec{d})} - M^{(\vec{d})} \Pi_i^{(b)}\right) \left(\psi^{(\bar{1}^i)}\right)^{1/2}\right\|_2 \tag{4.82}$$

$$\leq \left(\sum_{b, \vec{d}} \left\|\tilde{X}_{i, \vec{d}} Z_{i, \vec{d}} M^{(\vec{d})} \left(\psi^{(\bar{1}^i)}\right)^{1/2}\right\|_2^2\right)^{1/2} \left(\sum_{b, \vec{d}} \left\|\left(Z_{i, \vec{d}} M^{(\vec{d})} - M^{(\vec{d})} \Pi_i^{(b)}\right) \left(\psi^{(\bar{1}^i)}\right)^{1/2}\right\|_2^2\right)^{1/2} \tag{4.83}$$

Writing out the definition of the norm, it is easy to check that the first factor equals 1, and the second factor is bounded by Lemma 4.20:

$$\leq O(\gamma_H(D)^{1/2}) . \tag{4.84}$$

Repeating the above steps for the other $Z_{i, \vec{d}}^{(b)} M^{(\vec{d})}$-term, we obtain the desired statement. $\qquad \square$

Lemma 4.23. *For any efficient perfect device $D = (S, \Pi, M, P)$, any index $i \in [n]$, and any string $\vec{b} \in \{0, 1\}^n$, the following holds:*

$$\tilde{X}_i Z(\vec{b}) \approx_{\gamma_H(D)^{1/2}, \sigma^{(\bar{1})}} (-1)^{b_i} Z(\vec{b}) \tilde{X}_i . \tag{4.85}$$

Proof. Since $Z(\vec{b})$ is efficient, $\tilde{X}_i$ is efficient with access to the trapdoor t_i for the i-th key, and $\vec{1}$ and $\vec{1}^i$ agree on their i-th bit, by Lemma 4.13 it suffices to show that

$$\tilde{X}_i Z(\vec{b}) \approx_{\gamma_H(D)^{1/2}, \sigma^{(\vec{1}^i)}} (-1)^{b_i} Z(\vec{b}) \tilde{X}_i \,. \tag{4.86}$$

By Lemma 2.7, this is implied by

$$\mathrm{Tr}\left[\tilde{X}_i Z(\vec{b}) \tilde{X}_i Z(\vec{b}) \sigma^{(\vec{1}^i)} \right] \approx_{\gamma_H(D)^{1/2}} (-1)^{b_i} \,. \tag{4.87}$$

We can use Corollary 4.15 (noting that the i-th bit of $\vec{1}^i$ is 1) together with Lemma 2.9 to replace the leftmost $\tilde{X}_i$-operator by $\mathbb{1}$:

$$\mathrm{Tr}\left[\tilde{X}_i Z(\vec{b}) X_i Z(\vec{b}) \sigma^{(\vec{1}^i)} \right] \approx_{\gamma_H(D)^{1/2}} \mathrm{Tr}\left[Z(\vec{b}) \tilde{X}_i Z(\vec{b}) \sigma^{(\vec{1}^i)} \right] \,. \tag{4.88}$$

For the rest of the proof, we distinguish two cases:

Case $b_i = 0$. In this case, we split $\sigma^{(\vec{1}^i)} = \sum_{v \in \{0,1\}} \sigma^{(\vec{1}^i, v, \vec{b})}$:

$$\mathrm{Tr}\left[Z(\vec{b}) \tilde{X}_i Z(\vec{b}) \sigma^{(\vec{1}^i)} \right] = \sum_{v \in \{0,1\}} \mathrm{Tr}\left[Z(\vec{b}) \tilde{X}_i Z(\vec{b}) \sigma^{(\vec{1}^i, v, \vec{b})} \right] \tag{4.89}$$

Since $b_i = 0$ by assumption, the condition $b_i = 1 \implies \theta_i = 0$ (with $\vec{\theta} = \vec{1}^i$) in Corollary 4.18 is satisfied. We can therefore use that lemma together with Lemma 2.9 on each term in the sum to get:

$$\approx_{\gamma_H(D)^{1/2}} \sum_{v \in \{0,1\}} (-1)^v \, \mathrm{Tr}\left[Z(\vec{b}) \tilde{X}_i \sigma^{(\vec{1}^i, v, \vec{b})} \right] \tag{4.90}$$

$$\approx_{\gamma_H(D)^{1/2}} \sum_{v \in \{0,1\}} \mathrm{Tr}\left[\tilde{X}_i \sigma^{(\vec{1}^i, v, \vec{b})} \right] \tag{4.91}$$

$$= \mathrm{Tr}\left[\tilde{X}_i \sigma^{(\vec{1}^i)} \right] \tag{4.92}$$

Noting that $\tilde{X}_i = \tilde{X}(\vec{1}^i)$, we can use Corollary 4.15 and Lemma 2.9 to conclude:

$$\approx_{\gamma_H(D)^{1/2}} \mathrm{Tr}\left[\sigma^{(\vec{1}^i)} \right] \tag{4.93}$$

$$= 1 \,. \tag{4.94}$$

Case $b_i = 1$. We define $\vec{b}'$ by $b_i' = 0$ and $b_j' = b_j$ for $j \neq i$. Then, by Corollary 4.18 we have that $Z(\vec{b}') \approx_{\sigma^{(\vec{1}^i, v, \vec{b}')}} (-1)^v \mathbb{1}$. Furthermore, by definition of the Z-observables, $Z(\vec{b}) = Z_i Z(\vec{b}') = Z(\vec{b}') Z_i$. We can use this with Lemma 2.9 similarly to the previous case to obtain

$$\mathrm{Tr}\left[Z(\vec{b}) \tilde{X}_i Z(\vec{b}) \sigma^{(\vec{1}^i)} \right] = \sum_{v \in \{0,1\}} \mathrm{Tr}\left[Z(\vec{b}') Z_i \tilde{X}_i Z_i Z(\vec{b}') \sigma^{(\vec{1}^i, v, \vec{b})} \right] \tag{4.95}$$

$$\approx_{\gamma_H(D)^{1/2}} \sum_{v \in \{0,1\}} (-1)^v \mathrm{Tr}\left[Z(\vec{b}') Z_i \tilde{X}_i Z_i \sigma^{(\vec{1}^i, v, \vec{b})} \right] \tag{4.96}$$

$$\approx_{\gamma_H(D)^{1/2}} \sum_{v \in \{0,1\}} \mathrm{Tr}\left[Z_i \tilde{X}_i Z_i \sigma^{(\vec{1}^i, v, \vec{b})} \right] \tag{4.97}$$

$$= \mathrm{Tr}\left[Z_i \tilde{X}_i Z_i \sigma^{(\vec{1}^i)} \right] \tag{4.98}$$

$$\approx_{\gamma_H(D)^{1/2}} -1 \,, \tag{4.99}$$

where we used Lemma 4.21 in the last line. □

Proof of Proposition 4.19. Recall that our goal is to prove that

$$\tilde{X}(\vec{a})Z(\vec{b}) \approx_{n\gamma_H(D)^{1/4},\sigma^{(\vec{1})}} (-1)^{\vec{a}\cdot\vec{b}} Z(\vec{b})\tilde{X}(\vec{a}). \tag{4.100}$$

By Lemma 2.7, it suffices to show that

$$\text{Tr}\left[Z(\vec{b})\tilde{X}(\vec{a})Z(\vec{b})\tilde{X}(\vec{a})\sigma^{(\vec{1})}\right] \approx_{n\gamma_H(D)^{1/4}} (-1)^{\vec{a}\cdot\vec{b}}. \tag{4.101}$$

As a first step, we apply Corollary 4.15 to obtain

$$\text{Tr}\left[Z(\vec{b})\tilde{X}(\vec{a})Z(\vec{b})\tilde{X}(\vec{a})\sigma^{(\vec{1})}\right] \approx_{\gamma_H(D)^{1/2}} \text{Tr}\left[Z(\vec{b})\tilde{X}(\vec{a})Z(\vec{b})\sigma^{(\vec{1})}\right]. \tag{4.102}$$

We can write $\tilde{X}(\vec{a}) = \prod_{i=1}^{n} \tilde{X}_i^{a_i}$ (where $\tilde{X}_i^{a_i} = \mathbb{1}$ if $a_i = 0$):

$$\text{Tr}\left[Z(\vec{b})\tilde{X}(\vec{a})Z(\vec{b})\sigma^{(\vec{1})}\right] = \text{Tr}\left[Z(\vec{b})\left(\prod_{i=1}^{n-1} \tilde{X}_i^{a_i}\right) X_n^{a_n} Z(\vec{b})\sigma^{(\vec{1})}\right] \tag{4.103}$$

Applying Lemma 2.9 with Lemma 4.23 (if $a_n = 1$; if $a_n = 0$, the same step follows trivially and with equality):

$$\approx_{\gamma_H(D)^{1/4}} (-1)^{a_n b_n} \text{Tr}\left[Z(\vec{b})\left(\prod_{i=1}^{n-1} \tilde{X}_i^{a_i}\right) Z(\vec{b}) X_n^{a_n}\sigma^{(\vec{1})}\right] \tag{4.104}$$

Applying Lemma 2.9 with Corollary 4.15 (if $a_n = 1$; if $a_n = 0$, the same step follows trivially and with equality):

$$\approx_{\gamma_H(D)^{1/2}} (-1)^{a_n b_n} \text{Tr}\left[Z(\vec{b})\left(\prod_{i=1}^{n-1} \tilde{X}_i^{a_i}\right) Z(\vec{b})\sigma^{(\vec{1})}\right]. \tag{4.105}$$

Repeating the above steps for each of the remaining $\tilde{X}_i^{a_i}$, we find that

$$\text{Tr}\left[Z(\vec{b})\tilde{X}(\vec{a})Z(\vec{b})\sigma^{(\vec{1})}\right] \approx_{n\gamma_H(D)^{1/4}} \left(\prod_{i=1}^{n}(-1)^{a_i b_i}\right) \text{Tr}\left[Z(\vec{b})Z(\vec{b})\sigma^{(\vec{1})}\right] = (-1)^{\vec{a}\cdot\vec{b}} \tag{4.106}$$

as desired. The factor of n in the approximation arises because we incur an approximation error of $O(\gamma_H(D)^{1/4})$ for dealing with each of the n operators $\tilde{X}_i^{a_i}$, so by the triangle inequality the total approximation error is $O(n\gamma_H(D)^{1/4})$. □

4.4 Switching to efficient observables

Definition 4.24 (Rounding isometries). For a device D with associated Hilbert space $\mathcal{H}_D$ and $\vec{y} \in \mathcal{Y}^{\times n}$, $d \in \{0,1\}^{w\times n}$, we define the isometry $\tilde{V}_{y,d} : \mathcal{H}_D \to \mathcal{H}_D \otimes \mathcal{H}_A \otimes \mathcal{H}_Q$ by the following action on an arbitrary state $|\varphi\rangle_D$:

$$\tilde{V}_{\vec{y},\vec{d}}|\varphi\rangle_D = \underset{\vec{a},\vec{b}\in\{0,1\}^n}{\mathbb{E}} \left(\left(\tilde{X}(\vec{a})_{\vec{y},\vec{d}} Z(\vec{b})_{\vec{y},\vec{d}}\right)_D \otimes \left(\sigma_X(\vec{a})\sigma_Z(\vec{b})\right)_A\right) |\varphi\rangle_D \otimes \left(|\phi^{(0,0)}\rangle^{\otimes n}\right)_{AQ}, \tag{4.107}$$

34

where $|\phi^{(0,0)}\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$ denotes an EPR pair, and $\left(|\phi^{(0,0)}\rangle^{\otimes n}\right)_{AQ}$ is distributed between A and Q such that every EPR pair has one qubit in either system. We can combine the different $V_{y,d}$ into one isometry

$$\tilde{V} = \sum_{\vec{y},\vec{d}} \tilde{V}_{\vec{y},\vec{d}} \otimes |\vec{y},\vec{d}\rangle\langle\vec{y},\vec{d}| : \mathcal{H}_D \otimes \mathcal{H}_Y \otimes \mathcal{H}_R \to \mathcal{H}_D \otimes \mathcal{H}_Y \otimes \mathcal{H}_R \otimes \mathcal{H}_A \otimes \mathcal{H}_Q. \tag{4.108}$$

We similarly define

$$V_{\vec{y},\vec{d}}|\varphi\rangle_D = \mathop{\mathbb{E}}_{\vec{a},\vec{b}\in\{0,1\}^n} \left(\left(X(\vec{a})_{\vec{y},\vec{d}} Z(\vec{b})_{\vec{y},\vec{d}}\right)_D \otimes \left(\sigma_X(\vec{a})\sigma_Z(\vec{b})\right)_A\right) |\varphi\rangle_D \otimes \left(|\phi^{(0,0)}\rangle^{\otimes n}\right)_{AQ} \tag{4.109}$$

and

$$V = \sum_{\vec{y},\vec{d}} V_{\vec{y},\vec{d}} \otimes |\vec{y},\vec{d}\rangle\langle\vec{y},\vec{d}|. \tag{4.110}$$

Remark 4.25. The fact that $\tilde{V}$ and V as defined in Definition 4.24 are indeed isometries, i.e. that $\tilde{V}^\dagger\tilde{V} = \mathbb{1}$ and $V^\dagger V = \mathbb{1}$ follows straightforwardly from the fact that

$$\left(\langle\phi^{(0,0)}|^{\otimes n}\right)_{AQ} \left(\sigma_X(\vec{a}')\sigma_Z(\vec{b}')\right)_A^\dagger \left(\sigma_X(\vec{a})\sigma_Z(\vec{b})\right)_A \left(|\phi^{(0,0)}\rangle^{\otimes n}\right)_{AQ} = \delta_{\vec{a},\vec{a}'}\delta_{\vec{b},\vec{b}'}, \tag{4.111}$$

and $\tilde{X}(\vec{a})^2 = X(\vec{a})^2 = Z(\vec{b})^2 = \mathbb{1}$.

Remark 4.26. The isometry $\tilde{V}$ is inefficient since $\tilde{X}$ is an inefficient observable and is furthermore only defined for the basis choice $\vec{\theta} = \vec{1}$ since $\tilde{X}$ depends on the function $\hat{u}$ (see Definition 4.4). In contrast, since X and Z are both efficient observables, V is an efficient isometry and is well-defined for any basis choice.

The following lemma relates $\tilde{V}$ and V.

Lemma 4.27. *For any keys* $\vec{k} \in \mathcal{K}_1^n$:

$$V_{\vec{y},\vec{d}} = \sigma_Z(\hat{u}(\vec{k},\vec{y},\vec{d}))_A \otimes \sigma_Z(\hat{u}(\vec{k},\vec{y},\vec{d}))_Q \tilde{V}_{\vec{y},\vec{d}}. \tag{4.112}$$

Proof. For any state $|\varphi\rangle_D$, we have:

$$\sigma_Z(\hat{u}(\vec{k},\vec{y},\vec{d}))_A \otimes \sigma_Z(\hat{u}(\vec{k},\vec{y},\vec{d}))_Q \tilde{V}_{\vec{y},\vec{d}}|\varphi\rangle_D$$

$$= \mathop{\mathbb{E}}_{a,b\in\{0,1\}^n} \left(\tilde{X}(\vec{a})_{\vec{y},\vec{d}} Z(\vec{b})_{\vec{y},\vec{d}}\right)_D |\varphi\rangle_D \otimes \left[\left(\sigma_Z(\hat{u}(\vec{k},\vec{y},\vec{d}))\sigma_X(\vec{a})\sigma_Z(\vec{b})\right)_A \otimes \sigma_Z(\hat{u}(\vec{k},\vec{y},\vec{d}))_Q \left(|\phi^{(0,0)}\rangle^{\otimes n}\right)_{AQ}\right]$$

Repeatedly using that $(\sigma_Z)_A |\phi^{(0,0)}\rangle_{AQ} = (\sigma_Z)_Q |\phi^{(0,0)}\rangle_{AQ}$:

$$= \mathop{\mathbb{E}}_{a,b\in\{0,1\}^n} \left(\tilde{X}(\vec{a})_{\vec{y},\vec{d}} Z(\vec{b})_{\vec{y},\vec{d}}\right)_D |\varphi\rangle_D \otimes \left[\left(\sigma_Z(\hat{u}(\vec{k},\vec{y},\vec{d}))\sigma_X(\vec{a})\sigma_Z(\vec{b})\sigma_Z(\hat{u}(\vec{k},\vec{y},\vec{d}))\right)_A \left(|\phi^{(0,0)}\rangle^{\otimes n}\right)_{AQ}\right]$$

Since $\sigma_Z(\hat{u}(\vec{k},\vec{y},\vec{d}))\sigma_X(\vec{a})\sigma_Z(\vec{b})\sigma_Z(\hat{u}(\vec{k},\vec{y},\vec{d})) = (-1)^{a\cdot\hat{u}(\vec{k},\vec{y},\vec{d})}\sigma_X(\vec{a})\sigma_Z(\vec{b})$:

$$= \mathop{\mathbb{E}}_{a,b\in\{0,1\}^n} \left((-1)^{a\cdot u(\vec{k},\vec{y},\vec{d})}\tilde{X}(\vec{a})_{\vec{y},\vec{d}} Z(\vec{b})_{\vec{y},\vec{d}}\right)_D |\varphi\rangle_D \otimes \left[\left(\sigma_X(\vec{a})\sigma_Z(\vec{b})\right)_A \left(|\phi^{(0,0)}\rangle^{\otimes n}\right)_{AQ}\right]$$

Recalling from Definition 4.4 that $(-1)^{\vec{a}\cdot\hat{u}(\vec{k},\vec{y},\vec{d})}\tilde{X}(\vec{a})_{\vec{y},\vec{d}} = X(\vec{a})_{\vec{y},\vec{d}}$:

$$= \mathop{\mathbb{E}}_{a,b\in\{0,1\}^n} \left(X(\vec{a})_{\vec{y},\vec{d}} Z(\vec{b})_{\vec{y},\vec{d}} \right)_D |\varphi\rangle_D \otimes \left[\left(\sigma_X(\vec{a})\sigma_Z(\vec{b}) \right)_A \left(|\phi^{(0,0)}\rangle^{\otimes n} \right)_{AQ} \right]$$
$$= V|\varphi\rangle_D.$$

$\square$

Lemma 4.28. *For an efficient perfect device $D = (S, \Pi, M, P)$ and any $\vec{a}, \vec{b} \in \{0,1\}^n$ we have*

$$\mathrm{Tr}\left[\tilde{V}^\dagger \left(\sigma_X(\vec{a})\sigma_Z(\vec{b}) \right)_Q^\dagger \tilde{V}\tilde{X}(\vec{a})_{DYR} Z(\vec{b})_{DYR} \sigma_{DYR}^{(\vec{1})} \right] \approx_{n^{1/2}\gamma_H(D)^{1/8}} 1. \tag{4.113}$$

Proof. Inserting the definition of $\tilde{V}$:

$$\mathrm{Tr}\left[\tilde{V}^\dagger \left(\sigma_X(\vec{a})\sigma_Z(\vec{b}) \right)_Q^\dagger \tilde{V}\tilde{X}(\vec{a})_{DYR} Z(\vec{b})_{DYR} \sigma_{DYR}^{(\vec{1})} \right]$$

$$= \mathop{\mathbb{E}}_{\vec{a}',\vec{b}'} \mathrm{Tr}\left[\tilde{V}^\dagger \left(\tilde{X}(\vec{a}')Z(\vec{b}')\tilde{X}(\vec{a})Z(\vec{b})\sigma^{(\vec{1})} \right)_{DYR} \otimes \left(\sigma_X(\vec{a}')\sigma_Z(\vec{b}') \right)_A \otimes \left(\sigma_X(\vec{a})\sigma_Z(\vec{b}) \right)_Q^\dagger \left(|\phi^{(0,0)}\rangle^{\otimes n} \right)_{AQ} \right]$$

Using $\left(\sigma_X(\vec{a})\sigma_Z(\vec{b}) \right)_Q^\dagger \left(|\phi^{(0,0)}\rangle^{\otimes n} \right)_{AQ} = \left(\sigma_X(\vec{a})\sigma_Z(\vec{b}) \right)_A \left(|\phi^{(0,0)}\rangle^{\otimes n} \right)_{AQ}$ and the Pauli group relation $\sigma_X(\vec{a}')\sigma_Z(\vec{b}')\sigma_X(\vec{a})\sigma_Z(\vec{b}) = (-1)^{\vec{a}\cdot\vec{b}'}\sigma_X(\vec{a}+\vec{a}')\sigma_Z(\vec{b}+\vec{b}')$:

$$= \mathop{\mathbb{E}}_{\vec{a}',\vec{b}'} (-1)^{\vec{a}\cdot\vec{b}'} \mathrm{Tr}\left[\tilde{V}^\dagger \left(\tilde{X}(\vec{a}')Z(\vec{b}')\tilde{X}(\vec{a})Z(\vec{b})\sigma^{(\vec{1})} \right)_{DYR} \otimes \left(\sigma_X(\vec{a}+\vec{a}')\sigma_Z(\vec{b}+\vec{b}') \right)_A \left(|\phi^{(0,0)}\rangle^{\otimes n} \right)_{AQ} \right]$$

Using Proposition 4.19 with Lemma 2.9 to exchange the order of $\tilde{X}(\vec{a})$ and $Z(\vec{b})$ (which act directly on the state in the above expression), and then combining $Z(\vec{b})Z(\vec{b}') = Z(\vec{b}+\vec{b}')$:

$$\approx_{n^{1/2}\gamma_H(D)^{1/8}} \mathop{\mathbb{E}}_{\vec{a}',\vec{b}'} (-1)^{\vec{a}\cdot\vec{b}'+\vec{a}\cdot\vec{b}} \mathrm{Tr}\left[\tilde{V}^\dagger \left(\tilde{X}(\vec{a}')Z(\vec{b}+\vec{b}')\tilde{X}(\vec{a})\sigma^{(\vec{1})} \right)_{DYR} \otimes \left(\sigma_X(\vec{a}+\vec{a}')\sigma_Z(\vec{b}+\vec{b}') \right)_A \left(|\phi^{(0,0)}\rangle^{\otimes n} \right)_{AQ} \right]$$

We can now again use Proposition 4.19 with Lemma 2.9 to exchange the order of $Z(\vec{b}+\vec{b}')$ and $\tilde{X}(\vec{a})$ (and note that this will cancel the $(-1)^{\vec{a}\cdot\vec{b}'+\vec{a}\cdot\vec{b}}$ pre-factor in the above expression), and combine $X(\vec{a})X(\vec{a}') = X(\vec{a}+\vec{a}')$:

$$\approx_{n^{1/2}\gamma_H(D)^{1/8}} \mathop{\mathbb{E}}_{\vec{a}',\vec{b}'} \mathrm{Tr}\left[\tilde{V}^\dagger \left(\tilde{X}(\vec{a}+\vec{a}')Z(\vec{b}+\vec{b}')\sigma^{(\vec{1})} \right)_{DYR} \otimes \left(\sigma_X(\vec{a}+\vec{a}')\sigma_Z(\vec{b}+\vec{b}') \right)_A \left(|\phi^{(0,0)}\rangle^{\otimes n} \right)_{AQ} \right]$$

If we now shift the indices in the expectation $\vec{a}' \mapsto \vec{a}' - \vec{a}$ and $\vec{b}' \mapsto \vec{b}' - \vec{b}$, then this simplifies to:

$$= \mathrm{Tr}\left[\tilde{V}^\dagger \tilde{V}\sigma^{(\vec{1})} \right]$$
$$= 1.$$

$\square$

We can now combine Lemma 4.27 and Lemma 4.28 to show that the isometry V maps the observables $X(\vec{a})Z(\vec{b})$ to the corresponding Pauli observables.

Proposition 4.29. *For an efficient perfect device $D = (S, \Pi, M, P)$ and any $\vec{a}, \vec{b} \in \{0,1\}^n$ we have*

$$VX(\vec{a})Z(\vec{b})V^\dagger \approx_{n^{1/2}\gamma_H(D)^{1/8}, V\sigma^{(\vec{1})}V^\dagger} \left(\sigma_X(\vec{a})\sigma_Z(\vec{b})\right)_Q \otimes \mathbb{1}_{YRDA}. \tag{4.114}$$

Proof. By Lemma 2.7, it suffices to show that

$$\text{Tr}\left[\left(\sigma_X(\vec{a})\sigma_Z(\vec{b})\right)_Q^\dagger VX(\vec{a})Z(\vec{b})V^\dagger V\sigma^{(\vec{1})}V^\dagger\right] \approx_{n^{1/2}\gamma_H(D)^{1/8}} 1. \tag{4.115}$$

For this, we perform the following calculation. Using $V^\dagger V = \mathbb{1}$, cyclicity of the trace, and tracing over the registers Y and R:

$$\text{Tr}\left[\left(\sigma_X(\vec{a})\sigma_Z(\vec{b})\right)_Q^\dagger VX(\vec{a})Z(\vec{b})V^\dagger V\sigma^{(\vec{1})}V^\dagger\right]$$

$$= \sum_{\vec{y},\vec{d}} \text{Tr}\left[V_{\vec{y},\vec{d}}^\dagger \left(\sigma_X(\vec{a})\sigma_Z(\vec{b})\right)_Q^\dagger V_{\vec{y},\vec{d}} X(\vec{a})_{\vec{y},\vec{d}} Z(\vec{b})_{\vec{y},\vec{d}} \sigma_{\vec{y},\vec{d}}^{(\vec{1})}\right]$$

Using Lemma 4.27:

$$= \sum_{\vec{y},\vec{d}} \text{Tr}\left[\tilde{V}_{\vec{y},\vec{d}}^\dagger \sigma_Z(\hat{u}(\vec{k},\vec{y},\vec{d}))_A \otimes \sigma_Z(\hat{u}(\vec{k},\vec{y},\vec{d}))_Q \left(\sigma_X(\vec{a})\sigma_Z(\vec{b})\right)_Q^\dagger V_{\vec{y},\vec{d}} X(\vec{a})_{\vec{y},\vec{d}} Z(\vec{b})_{\vec{y},\vec{d}} \sigma_{\vec{y},\vec{d}}^{(\vec{1})}\right]$$

Exchanging the order of $\sigma_Z(\hat{u}(\vec{k},\vec{y},\vec{d}))_A \otimes \sigma_Z(\hat{u}(\vec{k},\vec{y},\vec{d}))_Q$ and $\left(\sigma_X(\vec{a})\sigma_Z(\vec{b})\right)_Q^\dagger$ (which produces a factor of $(-1)^{a \cdot \hat{u}(\vec{k},\vec{y},\vec{d})}$):

$$= \sum_{\vec{y},\vec{d}} (-1)^{\vec{a} \cdot \hat{u}(\vec{k},\vec{y},\vec{d})} \text{Tr}\left[\tilde{V}_{\vec{y},\vec{d}}^\dagger \left(\sigma_X(\vec{a})\sigma_Z(\vec{b})\right)_Q^\dagger \left[\sigma_Z(\hat{u}(\vec{k},\vec{y},\vec{d}))_A \otimes \sigma_Z(\hat{u}(\vec{k},\vec{y},\vec{d}))_Q V_{\vec{y},\vec{d}}\right] X(\vec{a})_{\vec{y},\vec{d}} Z(\vec{b})_{\vec{y},\vec{d}} \sigma_{\vec{y},\vec{d}}^{(\vec{1})}\right]$$

By Lemma 4.27, the expression in square brackets is simply $\tilde{V}_{\vec{y},\vec{d}}$. Additionally recalling from Definition 4.4 that $(-1)^{\vec{a} \cdot \hat{u}(\vec{k},\vec{y},\vec{d})} X(\vec{a})_{\vec{y},\vec{d}} = \tilde{X}(\vec{a})_{\vec{y},\vec{d}}$:

$$= \sum_{\vec{y},\vec{d}} \text{Tr}\left[\tilde{V}_{\vec{y},\vec{d}}^\dagger \left(\sigma_X(\vec{a})\sigma_Z(\vec{b})\right)_Q^\dagger \tilde{V}_{\vec{y},\vec{d}} \tilde{X}(\vec{a})_{\vec{y},\vec{d}} Z(\vec{b})_{\vec{y},\vec{d}} \sigma_{\vec{y},\vec{d}}^{(\vec{1})}\right]$$

Finally, we can re-introduce the systems Y and R and use Lemma 4.28 to obtain:

$$= \text{Tr}\left[\tilde{V}^\dagger \left(\sigma_X(\vec{a})\sigma_Z(\vec{b})\right)_Q^\dagger \tilde{V}\tilde{X}(\vec{a})Z(\vec{b})\sigma^{(\vec{1})}\right]$$

$$\approx_{n^{1/2}\gamma_H(D)^{1/8}} 1.$$

$\square$

4.5 Preparing BB84 states

Lemma 4.30. *For an efficient perfect device $D = (S, \Pi, M, P)$ and any $\vec{\theta} \in \{0,1\}^n$:*

(i) *If $\theta_i = 0$, then*

$$\sum_{\vec{v}} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\sigma_{Z,i}^{(v_i)}\right)_Q \approx_{n^{1/4}\gamma_H(D)^{1/16}, \sum_{\vec{v}'} |\vec{v}'\rangle\langle\vec{v}'| \otimes V\sigma^{(\vec{\theta},\vec{v}')}V^\dagger} \mathbb{1}. \tag{4.116}$$

(ii) *If $\theta_i = 1$, then*

$$\sum_{\vec{v}} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\sigma_{X,i}^{(v_i)}\right)_Q \approx_{n^{1/4}\gamma_H(D)^{1/16}, \sum_{\vec{v}'} |\vec{v}'\rangle\langle\vec{v}'| \otimes V\sigma^{(\vec{\theta},\vec{v}')}V^\dagger} \mathbb{1}. \tag{4.117}$$

Proof. We first prove the first statement. It is easy to check that $\sum_{\vec{v}} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\sigma_{Z,i}^{(v_i)}\right)_Q$ is a projector, so we can expand the definition of the state-dependent distance and compute:

$$\text{Tr}\left[\left(\sum_{\vec{v}} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\sigma_{Z,i}^{(v_i)}\right)_Q - \mathbb{1}\right)^\dagger \left(\sum_{\vec{v}} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\sigma_{Z,i}^{(v_i)}\right)_Q - \mathbb{1}\right) \sum_{\vec{v}'} |\vec{v}'\rangle\langle\vec{v}'| \otimes V\sigma^{(\vec{\theta},\vec{v}')}V^\dagger\right]$$

$$= \text{Tr}\left[\left(\mathbb{1} - \sum_{\vec{v}} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\sigma_{Z,i}^{(v_i)}\right)_Q\right) \sum_{\vec{v}'} |\vec{v}'\rangle\langle\vec{v}'| \otimes V\sigma^{(\vec{\theta},\vec{v}')}V^\dagger\right]$$

$$= 1 - \sum_{\vec{v}} \text{Tr}\left[\left(|\vec{v}\rangle\langle\vec{v}| \otimes \left(\sigma_{Z,i}^{(v_i)}\right)_Q\right) \sum_{\vec{v}'} |\vec{v}'\rangle\langle\vec{v}'| \otimes V\sigma^{(\vec{\theta},\vec{v}')}V^\dagger\right]$$

$$= 1 - \sum_{\vec{v}} \text{Tr}\left[\left(\sigma_{Z,i}^{(v_i)}\right)_Q V\sigma^{(\vec{\theta},\vec{v})}V^\dagger\right]$$

$$= 1 - \sum_{v_i \in \{0,1\}} \text{Tr}\left[\left(\sigma_{Z,i}^{(v_i)}\right)_Q V\sigma^{(\vec{\theta},v_i,\vec{1}^i)}V^\dagger\right],$$

where for the last line we used that $\sum_{v_1,\ldots,v_{i-1},v_{i+1},\ldots,v_n \in \{0,1\}} \sigma^{(\vec{\theta},\vec{v})} = \sigma^{(\vec{\theta},v_i,\vec{1}^i)}$. Therefore, to show the first part of the lemma, we need to show that

$$\sum_{v_i \in \{0,1\}} \text{Tr}\left[\left(\sigma_{Z,i}^{(v_i)}\right)_Q V\sigma^{(\vec{\theta},v_i,\vec{1}^i)}V^\dagger\right] \approx_{n^{1/4}\gamma_H(D)^{1/16}} 1. \tag{4.118}$$

For this, recall from Proposition 4.29 that we have

$$VZ_iV^\dagger \approx_{n^{1/2}\gamma_H(D)^{1/8}, V\sigma^{(\vec{1})}V^\dagger} (\sigma_{Z,i})_Q \otimes \mathbb{1}_{YRDA}. \tag{4.119}$$

Since V and Z_i are efficient, by Lemma 4.13 this implies that for any $\vec{\theta}$,

$$VZ_iV^\dagger \approx_{n^{1/2}\gamma_H(D)^{1/8}, V\sigma^{(\vec{\theta})}V^\dagger} (\sigma_{Z,i})_Q \otimes \mathbb{1}_{YRDA}. \tag{4.120}$$

(Compared to Lemma 4.13, here we have an additional isometry V applied to the state in the approximation. However, because V is efficient, it is easy to see that the proof of Lemma 4.13 still goes through.) Using [MV20, Lemma 2.18(ii)] and [MV20, Lemma 2.24], we get for any $v_i \in \{0,1\}$:

$$VZ_i^{(v_i)}V^\dagger \approx_{n^{1/2}\gamma_H(D)^{1/8}, V\sigma^{(\vec{\theta},v_i,\vec{1}^i)}V^\dagger} \left(\sigma_{Z,i}^{(v_i)}\right)_Q \otimes \mathbb{1}_{YRDA}. \tag{4.121}$$

Using the replacement lemma (Lemma 2.9), we obtain

$$\sum_{v_i \in \{0,1\}} \mathrm{Tr}\left[\left(\sigma_{Z,i}^{(v_i)}\right)_Q V \sigma^{(\vec{\theta},v_i,\vec{1}^i)} V^\dagger\right] \approx_{n^{1/4}\gamma_H(D)^{1/16}} \sum_{v_i \in \{0,1\}} \mathrm{Tr}\left[V Z_i^{(v_i)} V^\dagger V \sigma^{(\vec{\theta},v_i,\vec{1}^i)} V^\dagger\right] \tag{4.122}$$

$$= \sum_{v_i \in \{0,1\}} \mathrm{Tr}\left[Z_i^{(v_i)} \sigma^{(\vec{\theta},v_i,\vec{1}^i)}\right] \tag{4.123}$$

$$\approx_{\gamma_H(D)} 1\,, \tag{4.124}$$

where the last line follows from Equation (4.48) because $\theta_i = 0$. This finishes the proof of the first statement.

For the second statement, we can perform the same calculation, but use Equation (4.49) instead of Equation (4.48). $\qquad\square$

We are now in a position to show that on average over $\vec{v}$ and under the isometry V, an efficient perfect device must have prepared a product of BB84 states tensored with an additional state $\alpha^{(\vec{\theta},\vec{v})}$.

Lemma 4.31. *For an efficient perfect device $D = (S, \Pi, M, P)$ and any $\vec{\theta} \in \{0,1\}^n$, there exists a set of subnormalised states $\{\alpha^{(\vec{\theta},\vec{v})}\}_{\vec{v} \in \{0,1\}^n}$ such that*

$$\sum_{\vec{v} \in \{0,1\}^n} |\vec{v}\rangle\langle\vec{v}| \otimes V \sigma^{(\vec{\theta},\vec{v})} V^\dagger \approx_{n^{5/4}\gamma_H(D)^{1/16}} \sum_{\vec{v} \in \{0,1\}^n} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\bigotimes_i H^{\theta_i}|v_i\rangle\langle v_i|H^{\theta_i}\right)_Q \otimes \alpha^{(\vec{\theta},\vec{v})}\,. \tag{4.125}$$

Proof. We define the shorthand

$$M(\theta) = \begin{cases} Z & \text{if } \theta = 0\,, \\ X & \text{if } \theta = 1\,. \end{cases}$$

We can then apply Lemma 4.30 and [MV20, Lemma 2.22] twice to get

$$\sum_{\vec{v} \in \{0,1\}^n} |\vec{v}\rangle\langle\vec{v}| \otimes V \sigma^{(\vec{\theta},\vec{v})} V^\dagger$$

$$\approx_{n^{1/4}\gamma_H(D)^{1/16}} \left(\sum_{\vec{v}} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\sigma_{M(\theta_1),1}^{(v_1)}\right)_Q\right) \sum_{\vec{v} \in \{0,1\}^n} |\vec{v}\rangle\langle\vec{v}| \otimes V \sigma^{(\vec{\theta},\vec{v})} V^\dagger \left(\sum_{\vec{v}} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\sigma_{M(\theta_1),1}^{(v_1)}\right)_Q\right)$$

We can repeat this for the remaining indices $i = 2, \ldots, n$. Since we incur an approximation error of $n^{1/4}\gamma_H(D)^{1/16}$ for each of the n steps, the total approximation error will be $n^{5/4}\gamma_H(D)^{1/16}$, so we have

$$\sum_{\vec{v} \in \{0,1\}^n} |\vec{v}\rangle\langle\vec{v}| \otimes V \sigma^{(\vec{\theta},\vec{v})} V^\dagger$$

$$\approx_{n^{5/4}\gamma_H(D)^{1/16}} \left(\sum_{\vec{v}} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\sigma_{M(\theta_1),1}^{(v_1)}\right)_Q\right) \cdots \left(\sum_{\vec{v}} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\sigma_{M(\theta_n),n}^{(v_n)}\right)_Q\right) \sum_{\vec{v} \in \{0,1\}^n} |\vec{v}\rangle\langle\vec{v}| \otimes V \sigma^{(\vec{\theta},\vec{v})} V^\dagger$$

$$\left(\sum_{\vec{v}} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\sigma_{M(\theta_1),1}^{(v_1)}\right)_Q\right) \cdots \left(\sum_{\vec{v}} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\sigma_{M(\theta_n),n}^{(v_n)}\right)_Q\right)$$

$$= \sum_{\vec{v} \in \{0,1\}^n} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\prod_i \sigma_{M(\theta_i),i}^{(v_i)}\right)_Q V \sigma^{(\vec{\theta},\vec{v})} V^\dagger \left(\prod_i \sigma_{M(\theta_i),i}^{(v_i)}\right)_Q\,.$$

Now noting that $\prod_i \sigma_{M(\theta_i),i}^{(v_i)} = \bigotimes_i H^{\theta_i}|v_i\rangle\langle v_i|H^{\theta_i}$, we obtain

$$= \sum_{\vec{v}\in\{0,1\}^n} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\bigotimes_i H^{\theta_i}|v_i\rangle\langle v_i|H^{\theta_i}\right)_Q \otimes \left(\bigotimes_i \langle v_i|H^{\theta_i}\right)_Q V\sigma^{(\vec{\theta},\vec{v})}V^\dagger \left(\bigotimes_i H^{\theta_i}|v_i\rangle\right)_Q$$

$$= \sum_{\vec{v}\in\{0,1\}^n} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\bigotimes_i H^{\theta_i}|v_i\rangle\langle v_i|H^{\theta_i}\right)_Q$$

$$\otimes \mathrm{Tr}_Q\left[\left(\bigotimes_i H^{\theta_i}|v_i\rangle\langle v_i|H^{\theta_i}\right)_Q V\sigma^{(\vec{\theta},\vec{v})}V^\dagger \left(\bigotimes_i H^{\theta_i}|v_i\rangle\langle v_i|H^{\theta_i}\right)_Q\right]$$

Analogously to how we added the factors $\prod_i \sigma_{M(\theta_i),i}^{(v_i)}$ in a previous step, we can now replace the factors $\left(\bigotimes_i H^{\theta_i}|v_i\rangle\langle v_i|H^{\theta_i}\right)_Q$ inside the partial trace by identity, resulting in

$$\approx_{n^{1/4}\gamma_H(D)^{1/16}} \sum_{\vec{v}\in\{0,1\}^n} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\bigotimes_i H^{\theta_i}|v_i\rangle\langle v_i|H^{\theta_i}\right)_Q \otimes \mathrm{Tr}_Q\left[V\sigma^{(\vec{\theta},\vec{v})}V^\dagger\right].$$

We then obtain the desired statement by defining

$$\alpha^{(\vec{\theta},\vec{v})} := \mathrm{Tr}_Q\left[V\sigma^{(\vec{\theta},\vec{v})}V^\dagger\right]. \tag{4.126}$$

$\square$

The statement of Lemma 4.31 says that up to an isometry, the device's state has to be (information-theoretically) close to a product of BB84 states tensored with an auxiliary state $\alpha^{(\vec{\theta},\vec{v})}$. The drawback of this is that the auxiliary state depends on $\vec{\theta}$ and $\vec{v}$. For many applications, it is crucial that the only dependence on $\vec{\theta}$ and $\vec{v}$ in the prover's state is in the form of BB84 states, so we want that the auxiliary state be independent of $\vec{\theta}$ and $\vec{v}$. This holds in the sense that a computationally bounded device cannot learn anything about $\vec{\theta}$ and $\vec{v}$ from $\alpha^{(\vec{\theta},\vec{v})}$, as formalised in the following proposition.

Proposition 4.32. *For an efficient perfect device $D = (S, \Pi, M, P)$, there exists an efficiently preparable state α' such that for any $\vec{\theta} \in \{0,1\}^n$:*

$$\sum_{\vec{v}\in\{0,1\}^n} |\vec{v}\rangle\langle\vec{v}| \otimes V\sigma^{(\vec{\theta},\vec{v})}V^\dagger \overset{c}{\approx}_{n^{5/8}\gamma_H(D)^{1/32}} \frac{1}{2^n} \sum_{\vec{v}\in\{0,1\}^n} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\bigotimes_i H^{\theta_i}|v_i\rangle\langle v_i|H^{\theta_i}\right)_Q \otimes \alpha'. \tag{4.127}$$

Proof. Fix $\vec{\theta}$. For any $\vec{\theta}'$, we define

$$\alpha^{(\vec{\theta}')} = \sum_{\vec{v}} \alpha^{(\vec{\theta}',\vec{v})}, \tag{4.128}$$

where $\alpha^{(\vec{\theta}',\vec{v})}$ are the states in Lemma 4.31. From Lemma 4.12 we know that for any $\vec{\theta}', \vec{\theta}''$, $\sigma^{(\vec{\theta}')} \overset{c}{\approx}_0 \sigma^{(\vec{\theta}'')}$. Since V is an efficient isometry, it follows from this and Lemma 4.31 that

$$\alpha^{(\vec{\theta}')} \overset{c}{\approx}_{n^{5/4}\gamma_H(D)^{1/16}} \alpha^{(\vec{\theta}'')}. \tag{4.129}$$

40

We now set $\alpha^{(\vec{\theta}'')} = \alpha^{(\vec{0})} =: \alpha'$ (any other choice also works). From the proof of Lemma 4.31 we see that this state α' is efficiently preparable because the prover's operations are efficient, so the state α' (which is averaged over the possible answer's the prover can give, i.e. no post-selection is needed) can be efficiently prepared by executing the prover with basis choice $\vec{0}$, applying the (efficient) isometry V, and tracing out additional registers. Taking $\vec{\theta}' = \vec{\theta} \oplus \vec{1}$, we have

$$\frac{1}{2^n} \sum_{\vec{v} \in \{0,1\}^n} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\bigotimes_i H^{\theta_i} |v_i\rangle\langle v_i| H^{\theta_i} \right)_Q \otimes \alpha^{(\vec{\theta}')} \tag{4.130}$$

$$\overset{c}{\approx}_{n^{5/4}\gamma_H(D)^{1/16}} \frac{1}{2^n} \sum_{\vec{v} \in \{0,1\}^n} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\bigotimes_i H^{\theta_i} |v_i\rangle\langle v_i| H^{\theta_i} \right)_Q \otimes \alpha'. \tag{4.131}$$

From Lemma 4.31, we have

$$\sum_{\vec{v} \in \{0,1\}^n} |\vec{v}\rangle\langle\vec{v}| \otimes V\sigma^{(\vec{\theta},\vec{v})}V^\dagger \overset{c}{\approx}_{n^{5/4}\gamma_H(D)^{1/16}} \sum_{\vec{v} \in \{0,1\}^n} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\bigotimes_i H^{\theta_i} |v_i\rangle\langle v_i| H^{\theta_i} \right)_Q \otimes \alpha^{(\vec{\theta},\vec{v})}. \tag{4.132}$$

Comparing Equation (4.127), Equation (4.131), and Equation (4.132), we see that to show the lemma it suffices to show

$$\sum_{\vec{v} \in \{0,1\}^n} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\bigotimes_i H^{\theta_i} |v_i\rangle\langle v_i| H^{\theta_i} \right)_Q \otimes \alpha^{(\vec{\theta},\vec{v})}$$

$$\overset{c}{\approx}_{n^{5/4}\gamma_H(D)^{1/16}} \frac{1}{2^n} \sum_{\vec{v} \in \{0,1\}^n} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\bigotimes_i H^{\theta_i} |v_i\rangle\langle v_i| H^{\theta_i} \right)_Q \otimes \alpha^{(\vec{\theta}')}. \tag{4.133}$$

To show Equation (4.133), we first note that by tracing out the first system in the statement of Lemma 4.31, we get that there exists an $\varepsilon = O(n^{5/4}\gamma_H(D)^{1/16})$ such that for any $\vec{\theta}''$

$$\left\| \sum_{\vec{v} \in \{0,1\}^n} V\sigma^{(\vec{\theta}'',\vec{v})}V^\dagger - \sum_{\vec{v} \in \{0,1\}^n} \left(\bigotimes_i H^{\theta_i''} |v_i\rangle\langle v_i| H^{\theta_i''} \right)_Q \otimes \alpha^{(\vec{\theta}'',\vec{v})} \right\|_1^2 \le \varepsilon. \tag{4.134}$$

Supposing that Equation (4.133) does not hold, there exists an efficient measurement $\{\Lambda, \mathbb{1} - \Lambda\}$ such that

$$\mathrm{Tr}\left[\Lambda \left(\sum_{\vec{v} \in \{0,1\}^n} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\bigotimes_i H^{\theta_i} |v_i\rangle\langle v_i| H^{\theta_i} \right)_Q \otimes \alpha^{(\vec{\theta},\vec{v})} \right. \right.$$

$$\left. \left. - \frac{1}{2^n} \sum_{\vec{v} \in \{0,1\}^n} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\bigotimes_i H^{\theta_i} |v_i\rangle\langle v_i| H^{\theta_i} \right)_Q \otimes \alpha^{(\vec{\theta}')} \right) \right] \ge 2\varepsilon^{1/2} + \mu(\lambda) \tag{4.135}$$

for some non-negligible function $\mu(\lambda)$.

Since both states in the above expression are classical on the first two systems, without loss of generality we can assume that Λ is classical on the first two systems, too, i.e. we can write (remembering that $\vec{\theta}$ is fixed)

$$\Lambda = \sum_{\vec{v}} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\bigotimes_i H^{\theta_i} |v_i\rangle\langle v_i| H^{\theta_i} \right)_Q \otimes \Lambda_{\vec{v}}. \tag{4.136}$$

With this, we can rewrite Equation (4.135) in a more compact form:

$$\sum_{\vec{v}} \text{Tr}\left[\Lambda_{\vec{v}}\left(\alpha^{(\vec{\theta},\vec{v})} - \frac{1}{2^n}\alpha^{(\vec{\theta}')}\right)\right] \geq 2\,\varepsilon^{1/2} + \mu(\lambda).\tag{4.137}$$

Define

$$\Gamma = V^{\dagger}\left(\sum_{\vec{v}}\left(\bigotimes_i H^{\theta_i}|v_i\rangle\langle v_i|H^{\theta_i}\right)_Q \otimes \Lambda_{\vec{v}}\right) V.\tag{4.138}$$

Since $\{\Lambda, \mathbb{1} - \Lambda\}$ is an efficient measurement and V an efficient isometry, $\{\Gamma, \mathbb{1} - \Gamma\}$ is an efficient measurement, too. We claim that, assuming Equation (4.137) holds, we can use the measurement $\{\Gamma, \mathbb{1} - \Gamma\}$ to distinguish $\sigma^{(\vec{\theta})}$ from $\sigma^{(\vec{\theta}')}$ with advantage $\mu(\lambda)$, contradicting Lemma 4.12. To show this, we perform the following calculation:

$$\text{Tr}\left[\Gamma\left(\sigma^{(\vec{\theta})} - \sigma^{(\vec{\theta}')}\right)\right]$$

$$= \text{Tr}\left[\left(\sum_{\vec{v}}\left(\bigotimes_i H^{\theta_i}|v_i\rangle\langle v_i|H^{\theta_i}\right)_Q \otimes \Lambda_{\vec{v}}\right)\left(V\sigma^{(\vec{\theta})}V^{\dagger} - V\sigma^{(\vec{\theta}')}V^{\dagger}\right)\right]$$

Now using Equation (4.134) and [MV20, Lemma 2.21(ii)] (in a non-asymptotic form, which is easily seen to hold from the proof of [MV20, Lemma 2.21(ii)]):

$$\geq \text{Tr}\left[\left(\sum_{\vec{v}}\left(\bigotimes_i H^{\theta_i}|v_i\rangle\langle v_i|H^{\theta_i}\right)_Q \otimes \Lambda_{\vec{v}}\right)\right.$$
$$\left.\left(\sum_{\vec{v}'\in\{0,1\}^n}\left(\bigotimes_i H^{\theta_i}|v_i'\rangle\langle v_i'|H^{\theta_i}\right)_Q \otimes \alpha^{(\vec{\theta},\vec{v}')} - \sum_{\vec{v}'\in\{0,1\}^n}\left(\bigotimes_i H^{\theta_i'}|v_i'\rangle\langle v_i'|H^{\theta_i'}\right)_Q \otimes \alpha^{(\vec{\theta}',\vec{v}')}\right)\right] - 2\varepsilon^{1/2}$$

Since $\theta_i \neq \theta_i'$ for all i, we have that for all $\vec{v}$ and $\vec{v}'$, $|\langle\vec{v}'|(\otimes_i H^{\theta_i'}H^{\theta_i})|\vec{v}\rangle|^2 = 2^{-n}$. Then using $\sum_{\vec{v}'}\alpha^{(\vec{\theta}',\vec{v}')} = \alpha^{(\vec{\theta}')}$:

$$= \sum_{\vec{v}}\text{Tr}\left[\Lambda_{\vec{v}}\left(\alpha^{(\vec{\theta},\vec{v})} - \frac{1}{2^n}\alpha^{(\vec{\theta}')}\right)\right] - 2\varepsilon^{1/2}$$
$$\geq \mu(\lambda),$$

where the last inequality follows from Equation (4.137). This yields the desired contradiction and finishes the proof. $\qquad\square$

We are now ready to prove the main result of this part of the paper, namely that any efficient quantum prover that does not cause Protocol 3 to abort must have prepared a product of BB84 states (up to an isometry and an additional state α).

Theorem 4.33. *Suppose the verifier executes Protocol 3 with an efficient quantum prover with parameters n (number of BB84 states the verifier wishes to prepare), $N = M^2$ (maximum possible number of test rounds), and δ (error tolerance, i.e. proportion of test rounds that a prover is allowed to fail) that satisfy*

$N \geq C \frac{1}{n^{5/4}\delta^{1+1/32}}$ for a sufficiently large constant C. We denote by $\sigma_{SWDYR}^{(\vec{\theta})}$ the verifier and prover's joint final state at the end of Protocol 3, where $\vec{\theta}$ is the basis choice recorded by the verifier in Protocol 3, S is set to $|\perp\rangle\langle\perp|$ by the verifier if the protocol aborts and $|\top\rangle\langle\top|$ otherwise, W is the register in which the verifier records the string $\vec{v}$, and DYR are the prover's registers. Then, denoting the probability of success as $\Pr[\top]$ and expanding

$$\sigma_{SWDYR}^{(\vec{\theta})} = \Pr[\top]\,|\top\rangle\langle\top|_S \otimes \sigma_{WDYR|\top}^{(\vec{\theta})} + (1 - \Pr[\top])|\perp\rangle\langle\perp|_S \otimes \sigma_{WDYR|\perp}^{(\vec{\theta})}$$

there exists an efficiently preparable state α'_{DYRA} such that for any $\vec{\theta} \in \{0,1\}^n$ the following holds (with $V : \mathcal{H}_{DYR} \to \mathcal{H}_{DYRAQ}$ the efficient isometry defined in Definition 4.24):

$$\Pr[\top]\,V\sigma_{WDYR|\top}^{(\vec{\theta})}V^\dagger \overset{c}{\approx}_{n^{5/8}\delta^{1/64}} \Pr[\top]\frac{1}{2^n}\sum_{\vec{v}\in\{0,1\}^n}|\vec{v}\rangle\langle\vec{v}|_W \otimes \left(\bigotimes_i H^{\theta_i}|v_i\rangle\langle v_i|H^{\theta_i}\right)_Q \otimes \alpha'_{DYRA}. \quad (4.139)$$

Remark 4.34. Computational indistinguishability in the above theorem is with respect to a distinguisher that does not know the trapdoors for the keys used in the protocol. In particular, this includes the quantum prover with whom the verifier executed the protocol, i.e., the quantum prover itself cannot distinguish the joint state of the verifier's output $\vec{v}$ and its own final quantum state from the state on the r.h.s. of Equation (4.139) (up to the isometry V). Also note that on the prover's systems $QDYRA$ on the r.h.s. of Equation (4.139), the only dependence on $\vec{\theta}$ and $\vec{v}$ is in form of the BB84 states $H^{\theta_i}|v_i\rangle\langle v_i|H^{\theta_i}$, i.e. the state α is the same for all $\vec{\theta}$ and $\vec{v}$.

Proof. We will reduce this theorem to Proposition 4.32 in essentially the same manner as in [GV19, Theorem 3.17], but we spell out the details for completeness. Fix a prover strategy for Protocol 3. As a first step, we observe that we can model any such strategy as a sequence of devices $D_1, \ldots, D_N$, where D_i defines the prover's strategy for the i-th test round. Note that these devices need not be independent, i.e. device D_{i+1} can depend on the final state of device D_i after execution of the i-th test round. Since the prover is computationally efficient, so are the devices D_i (and the input to device D_i from device D_{i-1} is treated as an advice state as explained in Remark 4.3).

Having fixed such sequence of devices $D_1, \ldots, D_N$, we can now view an execution of Protocol 3 as a random experiment with sample space $\Omega = \{0, \ldots, M-1\} \times [M] \times \{0,1\}^n$. We denote by S and R the random variables that output the first and second component of an element of Ω, respectively, and by F_i the (bit-valued) random variable that outputs the i-th entry of the the third component. The probability measure on this space is induced by $\Pr[S = s, R = r, (F_1 = f_1, \ldots, F_n = f_n)] = \frac{1}{M^2}\Pr[F_1 = f_1, \ldots, F_n = f_n]$, where $\Pr[F_1 = f_1, \ldots, F_n = f_n]$ is given by the fixed sequence of devices $D_1, \ldots, D_N$ via the condition that $F_i = 1$ if the verifier sets `flag = fail`$_{\text{Pre}}$ or `flag = fail`$_{\text{Had}}$ when executing Protocol 1 with D_i, and $F_i = 0$ otherwise.

We also define random variables that denote whether this sequence of devices causes the verifier to abort on one of the M blocks of rounds in Protocol 3: for $j \in [M]$, we define

$$F_j^B = \begin{cases} 0 & \text{if } \frac{1}{M}\sum_{i \in B_j} F_i \leq \delta, \\ 1 & \text{else,} \end{cases}$$

where B_j is defined as in Protocol 3. We now denote by $\Omega' \subset \Omega$ the event that the verifier does not abort in Protocol 3, i.e., $\Pr[\Omega'] = \Pr[\top]$. By the definition of the abort condition in Protocol 3, $\omega =$

$(s, r, (f_1, \ldots, f_n)) \in \Omega'$ if and only if $F_j^B(\omega) = 0$ for all $j \leq s$. For any $\omega \in \Omega$ there can only exist one index $j^* \in [M]$ such that $F_{j^*}^B(\omega) = 1$ and $F_j^B(\omega) = 0$ for $j < j^*$. Therefore, since S is chosen uniformly at random by the verifier and F_i are independent of S (since S is unknown to the prover),

$$\Pr\left[F_{S+1}^B = 1 | \Omega'\right] = \frac{\Pr\left[F_{S+1}^B = 1 \wedge \Omega'\right]}{\Pr[\Omega']} \tag{4.140}$$

$$= \frac{\Pr\left[F_{S+1}^B = 1 \wedge F_S^B = 0 \wedge \cdots \wedge F_1^B = 0\right]}{\Pr[\Omega']} \tag{4.141}$$

$$\leq \frac{1}{M \Pr[\Omega']}. \tag{4.142}$$

Here, F_S^B is a random variable defined in the obvious way, i.e.,

$$F_S^B((s, r, (f_1, \ldots, f_n))) := F_s^B((s, r, (f_1, \ldots, f_n))) = \begin{cases} 0 & \text{if } \frac{1}{M} \sum_{i \in B_s} f_i \leq \delta, \\ 1 & \text{else.} \end{cases} \tag{4.143}$$

We call a pair (s, r) "bad" if

$$\Pr\left[F_{Ms+r} = 1 | \Omega' \wedge S = s \wedge R = r\right] \geq \sqrt{\delta}. \tag{4.144}$$

Then we have that for any s,

$$\Pr\left[(s, R) \text{ bad} | \Omega' \wedge S = s \wedge F_{s+1}^B = 0\right] \leq \sqrt{\delta}. \tag{4.145}$$

Here, we slightly abused notation and used "(s, R) bad" to denote the event consisting of all $\omega \in \Omega$ for which $(s, R(\omega))$ is bad. To see why the inequality holds, note that the event $\Omega' \wedge S = s \wedge F_{s+1}^B = 0$ is independent of R, so even if we condition on this event, R is still uniform. Then it follows from the definition of F_{s+1}^B that conditioned on $\Omega' \wedge S = s \wedge F_{s+1}^B = 0$, there can be at most a fraction $\sqrt{\delta}$ of choices of R such that (s, R) is bad. Since Equation (4.145) holds for any choice of $S = s$, we also have

$$\Pr\left[(S, R) \text{ good} | \Omega' \wedge F_{S+1}^B = 0\right] \geq 1 - \sqrt{\delta}, \tag{4.146}$$

where "(S, R) good" is defined analogously to before, i.e., it is the complement of the set of $\omega \in \Omega$ for which $(S(\omega), R(\omega))$ is bad. Combining Equation (4.142) and Equation (4.146), we then find that

$$\Pr\left[(S, R) \text{ good} | \Omega'\right] \geq \Pr\left[(S, R) \text{ good} \wedge F_{S+1}^B = 0 | \Omega'\right] \tag{4.147}$$

$$= \Pr\left[(S, R) \text{ good} | \Omega' \wedge F_{S+1}^B = 0\right] \Pr\left[F_{S+1}^B = 0 | \Omega'\right] \tag{4.148}$$

$$\geq (1 - \sqrt{\delta}) \left(1 - \frac{1}{M \Pr[\Omega']}\right) \tag{4.149}$$

$$\geq 1 - \sqrt{\delta} - \frac{1}{M \Pr[\Omega']}. \tag{4.150}$$

Note that if $\Pr[\Omega'] < O(n^{5/8} \delta^{1/64})$, Theorem 4.33 becomes trivial. We may therefore assume $\Pr[\Omega'] \geq O(n^{5/8} \delta^{1/64}) \geq O(1/\sqrt{\delta N})$ since we assumed $N = M^2 \geq \frac{C}{n^{5/4} \delta^{1+1/32}}$ for sufficiently large C. In that case we get that

$$\Pr\left[(S, R) \text{ good} | \Omega'\right] \geq 1 - O(\sqrt{\delta}). \tag{4.151}$$

44

If the verifier's choice of (S, R) in its run of Protocol 3 was "good", we get from the definition in Equation (4.144) that for the device $D = D_{MS+R}$ used by the prover in the run of Protocol 2 that occurs as step 3. in Protocol 3, we have $\gamma_H(D) \leq O(\sqrt{\delta})$. Hence, we can apply Proposition 4.32 and find that conditioned on the verifier making a "good" choice of (S, R), the state $\sigma^{(\vec{\theta})}_{|(S,R) \text{ good}}$ at the end of Protocol 3 satisfies

$$\sum_{\vec{v} \in \{0,1\}^n} |\vec{v}\rangle\langle\vec{v}| \otimes V\sigma^{(\vec{\theta},\vec{v})}_{|(S,R) \text{ good}} V^\dagger \overset{c}{\approx}_{n^{5/8}\delta^{1/64}} \frac{1}{2^n} \sum_{\vec{v} \in \{0,1\}^n} |\vec{v}\rangle\langle\vec{v}| \otimes \left(\bigotimes_i H^{\theta_i} |v_i\rangle\langle v_i| H^{\theta_i} \right)_Q \otimes \alpha'_{DYRA},$$

(4.152)

for an efficiently preparable state α'_{DYRA}. On the other hand, if the verifier's choice of (S, R) was "bad", then the computational distinguishability between the prover's final state and the state on the r.h.s. of Equation (4.152) is at most 1. Conditioned on the event Ω', Equation (4.151) tells us that the probability of the verifier choosing a bad (S, R) is at most $\sqrt{\delta}$. Hence, on average over the verifier's choice of (S, R) conditioned on Ω' (with the conditional state $\sigma^{(\vec{\theta})}_{|\Omega'} = \sigma^{(\vec{\theta})}_{|\top}$ as in the theorem statement) we get that

$$\sum_{\vec{v} \in \{0,1\}^n} |\vec{v}\rangle\langle\vec{v}|_W \otimes V\sigma^{(\vec{\theta},\vec{v})}_{DYR|\top} V^\dagger \overset{c}{\approx}_{n^{5/8}\delta^{1/64}+\sqrt{\delta}} \frac{1}{2^n} \sum_{\vec{v} \in \{0,1\}^n} |\vec{v}\rangle\langle\vec{v}|_W \otimes \left(\bigotimes_i H^{\theta_i} |v_i\rangle\langle v_i| H^{\theta_i} \right)_Q \otimes \alpha'_{DYRA}.$$

(4.153)

Noting that $\sqrt{\delta} \leq O(n^{5/8}\delta^{1/64})$, we obtain the desired statement. $\qquad\square$

While Theorem 4.33 does list explicit parameters, we emphasise that we have made no attempt at optimising these, so they can likely be improved. For the proof-of-principle applications in Part II, we typically want to choose a number of BB84 states n that we want to prepare and use parameters in the protocol that are polynomial in n. For that use case, we provide the following corollary, which is a simplified version of Theorem 4.33 in terms of asymptotically polynomial quantities.

Corollary 4.35. *For any $n \in \mathbb{N}$, there exist choices $N = \text{poly}(n)$ and $\delta = 1/\text{poly}(n)$ such that if the verifier executes Protocol 3 with this choice of parameters, the following holds. For any basis choice $\vec{\theta}$ by the verifier, denoting the verifier and prover's joint final state as*

$$\sigma^{(\vec{\theta})}_{SWDYR} = \Pr[\top] |\top\rangle\langle\top|_S \otimes \sigma^{(\vec{\theta})}_{WDYR|\top} + (1 - \Pr[\top])|\bot\rangle\langle\bot|_S \otimes \sigma^{(\vec{\theta})}_{WDYR|\bot}$$

as in Theorem 4.33, the state $\sigma^{(\vec{\theta})}_{WDYR|\top} = \sum_{\vec{v}} |\vec{v}\rangle\langle\vec{v}|_W \otimes \sigma^{(\vec{\theta},\vec{v})}_{DYR|\top}$ conditioned on acceptance satisfies

$$\Pr[\top] \, V\sigma^{(\vec{\theta})}_{WDYR|\top} V^\dagger \overset{c}{\approx}_{1/\text{poly}(n)} \Pr[\top] \frac{1}{2^n} \sum_{\vec{v} \in \{0,1\}^n} |\vec{v}\rangle\langle\vec{v}|_W \otimes \left(\bigotimes_i H^{\theta_i} |v_i\rangle\langle v_i| H^{\theta_i} \right)_Q \otimes \alpha'_B \qquad (4.154)$$

for an efficiently preparable state α_B with $B := DYRA$ and the isometry $V : DYR \to QB$ as in Theorem 4.33. If furthermore the prover's success probability is lower-bounded by an inverse polynomial, i.e., $\Pr[\top] \geq 1/\text{poly}(n)$, then we can choose $N = \text{poly}(n)$ and $\delta = 1/\text{poly}(n)$ such that an analogous statement also holds for the normalised conditional state:

$$V\sigma^{(\vec{\theta})}_{WDYR|\top} V^\dagger \overset{c}{\approx}_{1/\text{poly}(n)} \frac{1}{2^n} \sum_{\vec{v} \in \{0,1\}^n} |\vec{v}\rangle\langle\vec{v}|_W \otimes \left(\bigotimes_i H^{\theta_i} |v_i\rangle\langle v_i| H^{\theta_i} \right)_Q \otimes \alpha'_B. \qquad (4.155)$$

45

Proof. We can choose $\delta = 1/\operatorname{poly}(n)$ such that $n^{5/8}\delta^{1/64} = 1/\operatorname{poly}(n)$. Then, we can choose $N = O(1/(n^{5/4}\delta^{1+1/32})) = \operatorname{poly}(n)$ as in Theorem 4.33 and obtain Equation (4.154) directly from Theorem 4.33. To show Equation (4.155), we note that by renormalising (and noting that in the definition of computational indistinguishability, the distinguishing advantage for subnormalised states scales proportionally to the normalisation factor), we immediately get

$$V\sigma_{WDYR|\top}^{(\vec{\theta})}V^{\dagger} \overset{c}{\approx}_{1/(\Pr[\top]\operatorname{poly}(n))} \frac{1}{2^n} \sum_{\vec{v}\in\{0,1\}^n} |\vec{v}\rangle\langle\vec{v}|_V \otimes \left(\bigotimes_i H^{\theta_i}|v_i\rangle\langle v_i|H^{\theta_i}\right)_Q \otimes \alpha'_B. \tag{4.156}$$

To finish the proof, we observe that for any $\Pr[\top] \geq 1/\operatorname{poly}(n)$ we can choose a sufficiently small $\delta = 1/\operatorname{poly}(n)$ such that in the approximation error in Equation (4.156), $\Pr[\top]\operatorname{poly}(n) \geq \operatorname{poly}(n)$. $\qquad\square$

Part II
Applications

5 Unclonable quantum encryption

As a first application of our parallel RSP protocol, we consider the notion of *unclonable quantum encryption*. This cryptographic functionality was coined by Gottesman [Got02] and subsequently formalised by Broadbent and Lord [BL19]. Intuitively, the idea is to create a quantum ciphertext, corresponding to the encryption of a classical message, that cannot be duplicated. In other words, given the quantum ciphertext, no adversary can create two ciphertexts that decrypt to the same message as the original ciphertext. Let us formalise this intuition by introducing a few relevant definitions.

5.1 Quantum encryption of classical messages

A private-key quantum encryption of classical messages (QECM) scheme is a procedure that takes as input a key and a plaintext in the form of a classical bit string in a quantum register, and produces a ciphertext in the form of a quantum state. We formalise this notion in Definition 5.1.

Definition 5.1 (QECM scheme).
Let $\lambda \in \mathbb{N}$ be the security parameter. A quantum encryption of classical messages (QECM) scheme with key space $\mathcal{K}$ and plaintext space $\mathcal{M}$ is a triplet $\mathsf{QECM} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ consisting of QPT algorithms:

- $\mathsf{QECM.KeyGen}(1^\lambda) \to k$: takes as input the security parameter and outputs a key $k \in \mathcal{K}$.

- $\mathsf{QECM.Enc}(k, m) \to \rho$: takes as input a key k and a message $m \in \mathcal{M}$ and outputs $\rho \in \mathcal{D}(\mathcal{H}_A)$.

- $\mathsf{QECM.Dec}(k, \rho) \to \sigma$: takes as input a key k and a quantum ciphertext $\rho \in \mathcal{D}(\mathcal{H}_A)$, and outputs a plaintext in the form of a state $\sigma \in \mathcal{D}(\mathcal{H}_M)$, where $\mathcal{H}_M = \mathrm{span}\{|m\rangle : m \in \mathcal{M}\}$.

We use the notation $\mathsf{QECM.Enc}_k$ for the map $m \mapsto \mathsf{QECM.Enc}(k, m)$, and likewise for $\mathsf{QECM.Dec}_k$. The map $\mathsf{QECM.Enc}_k$ can be extended to quantum inputs $\sigma \in \mathcal{D}(\mathcal{H}_M)$ with $\mathcal{H}_M = \mathrm{span}\{|m\rangle : m \in \mathcal{M}\}$ via

$$\mathsf{Enc}_k(\sigma) = \sum_{m \in \mathcal{M}} \mathrm{Tr}[|m\rangle\langle m|\sigma] \, \mathsf{Enc}_k(|m\rangle\langle m|).$$

A QECM scheme is called *correct* if, for all $m \in \mathcal{M}$ and $k \in \mathrm{supp}\,\mathsf{QECM.KeyGen}(1^\lambda)$,

$$\mathrm{Tr}[|m\rangle\langle m|\mathsf{QECM.Dec}_k \circ \mathsf{QECM.Enc}_k(m)] \geq 1 - \mathrm{negl}(\lambda).$$

Note that we typically assume that the key space is given by $\mathcal{K} = \{0,1\}^\lambda$ and that the plaintext space $\mathcal{M}$ and ciphertext space consist of inputs of at most $\mathrm{poly}(\lambda)$ many bits and qubits, respectively.

We use the following notion of indistinguishable encryptions for QECM schemes which is inspired by the definition of Alagic et al. [ABF+16].

Definition 5.2 (Indistinguishable security). A QECM scheme $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ has *indistinguishable encryptions* (or is IND-secure) if, for every QPT adversary $\mathcal{A} = (\mathcal{M}, \mathcal{D})$,

$$\left| \mathrm{Pr}\left[\mathcal{D}\big((\mathsf{Enc}_k \otimes \mathsf{id}_E)\rho_{ME}\big) = 1 \right] - \mathrm{Pr}\left[\mathcal{D}\big((\mathsf{Enc}_k \otimes \mathsf{id}_E)(|0\rangle\langle 0|_M \otimes \rho_E)\big) = 1 \right] \right| \leq \mathrm{negl}(\lambda),$$

where we assume that $k \leftarrow \mathsf{KeyGen}(1^\lambda)$, $\rho_{ME} \leftarrow \mathcal{M}(1^\lambda)$ with $\rho_E = \mathrm{Tr}_M[\rho_{ME}]$, and where $|0\rangle\langle 0|_M$ is the all-0 string in the plaintext register M.

Informally, we say that a quantum encryption scheme is *unclonable* if no QPT adversary can produce two "copies" of a quantum ciphertext which can each be decrypted with access to the private key. Before we make the notion of unclonable ciphertexts more precise, let us first introduce the following definition of a cloning attack due to Broadbent and Lord [BL19].

Definition 5.3 (Cloning attack). Let $\lambda \in \mathbb{N}$ be a parameter. A cloning attack $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ against a QECM scheme $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ consists of the following QPT algorithms (which are parameterised by λ)

- (cloning map) $\mathcal{A} : \mathcal{D}(\mathcal{H}_A) \to \mathcal{D}(\mathcal{H}_B \otimes \mathcal{H}_C)$

- (1st decoder) $\mathcal{B} : \mathcal{K} \times \mathcal{D}(\mathcal{H}_B) \to \mathcal{D}(\mathcal{H}_M)$

- (2nd decoder) $\mathcal{C} : \mathcal{K} \times \mathcal{D}(\mathcal{H}_C) \to \mathcal{D}(\mathcal{H}_M)$

where $\mathcal{K}$ and $\mathcal{H}_A$ are defined by the scheme Σ, i.e. $\mathcal{K}$ is the set of keys and $\mathcal{H}_A$ is the ciphertext system.

We remark that we only consider *efficient* cloning attacks throughout this work. This is in contrast with the definition of Broadbent and Lord [BL19] who consider CPTP maps more generally.

Definition 5.4 (Cloning experiment). Let $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a QECM scheme and let $\lambda \in \mathbb{N}$ be the security parameter. We define the following security game, called the *cloning experiment*, which takes place between a challenger and an efficient adversary who executes a cloning attack $(\mathcal{A}, \mathcal{B}, \mathcal{C})$:

1. The challenger samples a key $k \xleftarrow{\$} \mathcal{K}$ and message $m \xleftarrow{\$} \mathcal{M}$, and sends $\rho_A \leftarrow \mathsf{Enc}_k(m)$ to $\mathcal{A}$.

2. $\mathcal{A}$ maps ρ_A into a bipartite state ρ_{BC} on systems BC, and sends ρ_{BC} to the challenger together with a pair of circuit descriptions of ensembles of efficient quantum algorithms $\{\mathcal{B}_\kappa\}_{\kappa \in \mathcal{K}}$ and $\{\mathcal{C}_\kappa\}_{\kappa \in \mathcal{K}}$.

3. The challenger runs $\mathcal{B}_k$ on system B and $\mathcal{C}_k$ on system C of ρ_{BC}, measures the output states in the computational basis to obtain outcomes m_B and m_C, and outputs 1 if $m = m_B = m_C$, and 0 otherwise.

We let the random variable $\mathsf{CloneExp}_\Sigma\big(1^\lambda, (\mathcal{A}, \mathcal{B}, \mathcal{C})\big)$ denote the output bit of the challenger.

Building on the cloning experiment in Definition 5.4, we then define unclonable security as follows.

Definition 5.5 (Unclonable Security). Let $\lambda \in \mathbb{N}$ be a security parameter. We say that a QECM scheme $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is $t(\lambda)$-unclonable secure if, for all cloning attacks $(\mathcal{A}, \mathcal{B}, \mathcal{C})$, it holds that

$$\Pr\left[\mathsf{CloneExp}_\Sigma\big(1^\lambda, (\mathcal{A}, \mathcal{B}, \mathcal{C})\big) = 1\right] \leq 2^{-\lambda + t(\lambda)} + \mathsf{negl}(\lambda).$$

5.2 Unclonable quantum encryption with a classical client

Let us now extend the notion of quantum encryption schemes of classical messages from Definition 5.1 to allow for interaction between a classical client and a quantum receiver. In this setting, the encryption algorithm is replaced by a an interactive protocol which enables the delegation of a ciphertext. Contrary to the standard notion of QECM schemes in Definition 5.1, our definition of QECM schemes with a classical client does not feature a key generation procedure; rather, we assume that the key is produced as a result of the interactive encryption protocol between the client and the receiver. Our definition is the following:

Definition 5.6 (QECM scheme with a classical client).
Let $\lambda \in \mathbb{N}$ be a security parameter, let $\mathcal{K}$ be the key space and let $\mathcal{M}$ be the plaintext space. A QECM scheme with a classical client (i.e., a QECM$_{\mathsf{CC}}$ scheme) consists of a pair QECM$_{\mathsf{CC}} = (\mathsf{Enc}, \mathsf{Dec})$, where

- $\mathsf{QECM}_{\mathsf{CC}}.\mathsf{Enc}(\mathcal{C}(1^\lambda, m), \mathcal{R}(1^\lambda)) \to (k, \rho)$ **or** $\bot$: this is an interactive protocol between a classical client $\mathcal{C}$ which takes as input the security parameter 1^λ and message $m \in \mathcal{M}$, and a quantum receiver $\mathcal{R}$ which takes as input the parameter 1^λ. The protocol takes place as follows:

  - $\mathcal{C}$ and $\mathcal{R}$ exchange classical messages only. Once the protocol is complete, $\mathcal{C}$ obtains a flag which is either $\top$ (accept) or $\bot$ (reject). Provided that the protocol is successful with the flag $\top$, $\mathcal{C}$ obtains an associated secret key $k \in \mathcal{K}$, and $\mathcal{R}$ is in possession of a state $\rho \in \mathcal{D}(\mathcal{H}_A)$ (which depends on k and m). Otherwise, $\mathcal{C}$ outputs $\bot$.

- $\mathsf{QECM}_{\mathsf{CC}}.\mathsf{Dec}(k, \rho)$: takes as input a secret key k, a state $\rho \in \mathcal{D}(\mathcal{H}_A)$ and outputs a plaintext in the form of a quantum state $\sigma \in \mathcal{D}(\mathcal{H}_M)$, where $\mathcal{H}_M = \mathrm{span}\{|m\rangle : m \in \mathcal{M}\}$.

We say that $\mathsf{QECM}_{\mathsf{CC}} = (\mathsf{Enc}, \mathsf{Dec})$ is correct if, for all plaintexts $m \in \mathcal{M}$, it holds that

$$\mathrm{Tr}[|m\rangle\langle m|\mathsf{QECM}_{\mathsf{CC}}.\mathsf{Dec}_k(\rho)] \geq 1 - \mathrm{negl}(\lambda),$$

where $(k, \rho) \leftarrow \mathsf{QECM}_{\mathsf{CC}}.\mathsf{Enc}(\mathcal{C}(1^\lambda, m), \mathcal{R}(1^\lambda))$ is the final result of the interactive protocol between the client and the receiver, provided that the protocol is successful.

Similar to Definition 5.3, we now define an analogous security experiment in the classical client setting.

Definition 5.7 (Cloning experiment with a classical client). Let $\Sigma = (\mathsf{Enc}, \mathsf{Dec})$ be a $\mathsf{QECM}_{\mathsf{CC}}$ scheme and let $\lambda \in \mathbb{N}$ be the security parameter. We define the security game (called the *cloning experiment with a classical client*) which takes place between a challenger and an adversary $(\mathcal{P}, \mathcal{A}, \mathcal{B}, \mathcal{C})$ consisting of an efficient interactive prover $\mathcal{P}$ and an efficient cloning attack $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ as follows:

1. The challenger samples a random message $m \xleftarrow{\$} \mathcal{M}$. To classically delegate a ciphertext according to the scheme Σ, the challenger takes the role of the classical client $\mathcal{C}$, whereas the prover $\mathcal{P}$ takes the role of the (possibly malicious) recipient in the interactive protocol specified by $\mathsf{QECM}_{\mathsf{CC}}.\mathsf{Enc}$. Provided that the protocol succeeds with $\mathtt{flag} = \top$, $\mathcal{C}$ obtains an associated secret key $k \in \mathcal{K}$, and $\mathcal{P}$ is in possession of a state $\rho \in \mathcal{D}(\mathcal{H}_A)$ (which depends on m and k, as well as any additional information collected during the protocol). Otherwise, if $\mathtt{flag} = \bot$, the adversary loses.

2. $\mathcal{A}$ maps ρ_A into a bipartite state ρ_{BC} on systems BC, and sends ρ_{BC} to the challenger together with a pair of classical circuit descriptions of efficient quantum algorithms $\{\mathcal{B}_\kappa\}_{\kappa \in \mathcal{K}}$ and $\{\mathcal{C}_\kappa\}_{\kappa \in \mathcal{K}}$.

3. The challenger applies the channel $(\mathcal{B}_k \otimes \mathcal{C}_k)$ to the state ρ_{BC}, measures in the standard basis with outcomes m_B and m_C, and outputs 1 if $m = m_B = m_C$ and $\mathtt{flag} = \top$, and 0 otherwise.

We let the random variable $\mathsf{CloneExp}_{\Sigma, \mathsf{CC}}(1^\lambda, (\mathcal{P}, \mathcal{A}, \mathcal{B}, \mathcal{C}))$ denote the output bit of the challenger.

Definition 5.8 (Unclonable Security with a Classical Client).
Let $\lambda \in \mathbb{N}$ be the security parameter. A $\mathsf{QECM}_{\mathsf{CC}}$ scheme $\Sigma = (\mathsf{Enc}, \mathsf{Dec})$ with message space $\{0, 1\}^\lambda$ is said to be $t(\lambda)$-unclonable secure if, for any (interactive) QPT adversary $(\mathcal{P}, \mathcal{A}, \mathcal{B}, \mathcal{C})$,

$$\Pr\left[\mathsf{CloneExp}_{\Sigma, \mathsf{CC}}(1^\lambda, (\mathcal{P}, \mathcal{A}, \mathcal{B}, \mathcal{C})) = 1\right] \leq 2^{-\lambda + t(\lambda)} + \mathrm{negl}(\lambda).$$

5.3 Conjugate coding encryption with a classical client

In this section, we give a quantum encryption scheme with a classical client based on the notion of *conjugate coding* – an idea first proposed by Wiesner [Wie83]. The main idea behind the construction is that it is possible to *encrypt* a plaintext $m \in \{0,1\}^\lambda$ in the form of a quantum ciphertext by first applying a classical one-time pad, and then making λ uniformly random choices of basis (either computational or Hadamard).

Let us first introduce the basic conjugate cording encryption scheme which was also recently considered by Broadbent and Lord [BL19] in the context of unclonable encryption.

Construction 1 (Conjugate coding scheme). Let $\lambda \in \mathbb{N}$ be the security parameter. The conjugate coding encryption scheme $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ consists of the following QPT algorithms:

$\Sigma.\mathsf{KeyGen}(1^\lambda)$: takes as input the parameter 1^λ and outputs a key $k = (\vec{r}, \vec{\theta}) \xleftarrow{\$} \{0,1\}^\lambda \times \{0,1\}^\lambda$.

$\Sigma.\mathsf{Enc}(k, m)$: takes as input a key $k = (\vec{r}, \vec{\theta})$ and a message $m \in \{0,1\}^\lambda$ and outputs the cipherterxt

$$\rho = \bigotimes_{i=1}^{\lambda} H^{\theta_i} |r_i \oplus m_i\rangle\langle r_i \oplus m_i| H^{\theta_i}.$$

$\Sigma.\mathsf{Dec}(k, \rho)$: takes as input a key $k = (\vec{r}, \vec{\theta})$ and decrypts a quantum ciphertext ρ as follows:

(i) apply $H^{\theta_1} \otimes \cdots \otimes H^{\theta_\lambda}$ to the state ρ and measure in the standard basis with outcome $\vec{x}$.

(ii) output the plaintext $\vec{m}' = \vec{x} \oplus \vec{r}$.

Notice that the conjugate coding scheme in Construction 1 is trivially correct and also IND-secure according to Definition 5.2. Moreover, the scheme has the following desirable property which was shown by Broadbent and Lord [BL19]. Namely, the quantum ciphertext is *unclonable* in the following sense:

Theorem 5.9 ([BL19], Theorem 15). *Let $\lambda \in \mathbb{N}$ be the security parameter. Then, the conjugate coding encryption scheme $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ defined in Construction 1 is $t(\lambda)$-unclonable secure, where $t(\lambda) = \lambda \log(1 + 1/\sqrt{2})$. In other words, for all cloning attacks $(\mathcal{A}, \mathcal{B}, \mathcal{C})$,*

$$\mathop{\mathbb{E}}_{\vec{m}} \mathop{\mathbb{E}}_{\vec{r}} \mathop{\mathbb{E}}_{\vec{\theta}} \mathrm{Tr}\left[(|\vec{m}\rangle\langle\vec{m}| \otimes |\vec{m}\rangle\langle\vec{m}|)(\mathcal{B}_{(\vec{r},\vec{\theta})} \otimes \mathcal{C}_{(\vec{r},\vec{\theta})}) \circ \mathcal{A} \circ \mathsf{Enc}_{(\vec{r},\vec{\theta})}(\vec{m}) \right] \leq \left(\frac{1}{2} + \frac{1}{2\sqrt{2}} \right)^\lambda.$$

We define our conjugate coding encrypytion scheme with a classical client in Protocol 4. Let us first show the correctness of our scheme.

Proposition 5.10 (Correctness). $\Sigma = (\mathsf{Enc}, \mathsf{Dec})$ *in Protocol 4 is a correct QECM$_{\mathsf{CC}}$ scheme.*

Proof. Let $\lambda \in \mathbb{N}$ and $n(\lambda) \in \mathbb{N}$ be parameters and let $\Sigma = (\mathsf{Enc}, \mathsf{Dec})$ be the QECM$_{\mathsf{CC}}$ scheme in Protocol 4. To prove correctness, we have to show that, for all $m \in \{0,1\}^n$,

$$\mathrm{Tr}[|m\rangle\langle m| \mathsf{Dec}_k(\sigma)] \geq 1 - \mathrm{negl}(\lambda), \tag{5.1}$$

where $(k, \sigma) \leftarrow \mathsf{Enc}(\mathcal{C}(1^\lambda, m), \mathcal{R}(1^\lambda))$ is the final result of the interactive protocol between the client $\mathcal{C}$ and the receiver $\mathcal{R}$. Note that in this scenario we can assume that the receiver $\mathcal{R}$ is honest throughout the protocol. According to Proposition 3.1, $\mathcal{R}$ is accepted in Protocol 3 with probability negligibly close to 1 in

---

Protocol 4. Conjugate Coding Encryption with a Classical Client

Let $\lambda \in \mathbb{N}$ and $n = n(\lambda) \in \mathbb{N}$ be parameters. Consider the scheme $\Sigma = (\mathsf{Enc}, \mathsf{Dec})$ defined by:

$\Sigma.\mathsf{Enc}(\mathcal{C}(1^\lambda, m), \mathcal{R}(1^\lambda)) \to (k, \sigma)$ **or** $\perp$: this is the following interactive protocol between a classical client $\mathcal{C}$ (which takes as input the security parameter 1^λ and a plaintext $m \in \{0,1\}^n$) and a quantum receiver $\mathcal{R}$ (which takes as input the parameter 1^λ):

(i) $\mathcal{C}$ and $\mathcal{R}$ run the parallel RSP protocol (Protocol 3) with parameters λ and n for a random $\vec{\theta} \xleftarrow{\$} \{0,1\}^n$. If the protocol is successful, $\mathcal{C}$ obtains $\vec{v} \xleftarrow{\$} \{0,1\}^n$ and $\mathcal{R}$ obtains an n-qubit state σ (which depends on $\vec{v}, \vec{\theta}$). Otherwise, if the protocol fails, $\mathcal{C}$ outputs $\perp$.

(ii) the client lets $k = (\vec{r}, \vec{\theta})$ be the secret key, where $\vec{r}$ satisfies $\vec{v} = \vec{m} \oplus \vec{r}$.

$\Sigma.\mathsf{Dec}(k, \rho) \to \vec{m}'$: takes as input the secret key $k = (\vec{r}, \vec{\theta})$ and a state ρ and decrypts as follows:

(i) apply $H^{\theta_1} \otimes \cdots \otimes H^{\theta_n}$ to the state ρ and measure in the standard basis with outcome $\vec{x}$.

(ii) output the plaintext $\vec{m}' = \vec{x} \oplus \vec{r}$.

---

the security parameter λ (for parameter choices n at most polynomial in λ and δ at least inverse polynomial in λ). Furthermore, provided that Protocol 3 is successful, the final state of $\mathcal{R}$ is given by

$$\sigma = \bigotimes_{i=1}^{n} H^{\theta_i} |v_i\rangle\langle v_i| H^{\theta_i}, \tag{5.2}$$

where $\vec{v}$ and $\vec{\theta}$ are the strings recorded by the client $\mathcal{C}$. Hence, the correctness property in Equation (5.1) of $\Sigma = (\mathsf{Enc}, \mathsf{Dec})$ in Protocol 4 immediately follows from the correctness of the conjugate coding scheme in Construction 1. This proves the claim. $\qquad\square$

Theorem 5.11 (Security). *Let $\lambda \in \mathbb{N}$ and $n(\lambda) \in \mathbb{N}$ be parameters. Then, there exist $\varepsilon(\lambda) = 1/\operatorname{poly}(\lambda)$ and $t(\lambda) = n(\lambda)\log(1 + 1/\sqrt{2})$ such that the QECM_{CC} scheme $\Sigma = (\mathsf{Enc}, \mathsf{Dec})$ defined in Protocol 4 is $\tilde{t}(\lambda)$-unclonable secure, where we let $\tilde{t}(\lambda) := t(\lambda) + \log\left(1 + 2^{n(\lambda) - t(\lambda) + \log\varepsilon(\lambda)}\right)$. In other words, for every efficient interactive quantum prover $\mathcal{P}$ and for all QPT cloning attacks $(\mathcal{A}, \mathcal{B}, \mathcal{C})$, it holds that*

$$\Pr\left[\mathsf{CloneExp}_{\Sigma, CC}(1^\lambda, (\mathcal{P}, \mathcal{A}, \mathcal{B}, \mathcal{C})) = 1\right] \leq 2^{-n(\lambda) + t(\lambda)} + \varepsilon(\lambda) + \mathsf{negl}(\lambda).$$

Proof. Let us analyze the success probability in the cloning experiment in Definition 5.4. Let $(\mathcal{P}, \mathcal{A}, \mathcal{B}, \mathcal{C})$ be an efficient adversary consisting of an efficient interactive prover $\mathcal{P}$ and a triplet of QPT algorithms $(\mathcal{A}, \mathcal{B}, \mathcal{C})$. Let $\sigma^{(\vec{\theta})}_{WDYR|\top}$ be the final state at the end of the interactive protocol conditioned on the event that the challenger outputs $\mathtt{flag} = \top$ as in Corollary 4.35. Recall that the challenger has access to system W whereas the prover $\mathcal{P}$ has access to systems DYR. Corollary 4.35 then considers the state $V\sigma^{(\vec{\theta})}_{WDYR|\top}V^\dagger$ for an efficient isometry $V : DYR \to QB$ with $B = DYRA$. We can argue that w.l.og. we can assume that the prover's state is in fact $V\sigma^{(\vec{\theta})}_{WDYR|\top}V^\dagger$, not $\sigma^{(\vec{\theta})}_{WDYR|\top}$. The reason is that the prover can recover the

latter from the former by considering the (efficient) unitary extension $U : QB \to QB$ of V, applying $U^\dagger$ to $V\sigma^{(\vec{\theta})}_{WDYR|\top}V^\dagger$, and then tracing out systems QA. Hence, slightly abusing notation, we define

$$\sigma^{(\vec{\theta})}_{WQB|\top} := V\sigma^{(\vec{\theta})}_{WDYR|\top}V^\dagger$$

and assume that the prover has prepared the state $\sigma^{(\vec{\theta})}_{WQB|\top}$. We can expand this state as a cq-state as in Corollary 4.35, and write

$$\sigma^{(\vec{\theta})}_{WQB|\top} = \sum_{\vec{v}\in\{0,1\}^n} |\vec{v}\rangle\langle\vec{v}|_W \otimes \sigma^{(\vec{v},\vec{\theta})}_{QB|\top}, \tag{5.3}$$

where $\sigma^{(\vec{v},\vec{\theta})}_{QB|\top}$ are subnormalised according to the probability of the verifier receiving outcome $\vec{v}$. In Protocol 5, the verifier makes the basis choice $\vec{\theta}$ uniformly at random, and in the cloning experiment in Definition 5.4 the verifier chooses a uniformly random message m. Therefore, the success probability in the cloning experiment can be expressed as

$$\Pr\left[\mathsf{CloneExp}_{\Sigma,CC}\left(1^\lambda, (\mathcal{P}, \mathcal{A}, \mathcal{B}, \mathcal{C})\right) = 1\right]$$
$$= \sum_{\vec{v}} \mathbb{E}_{\vec{m}} \mathbb{E}_{\vec{\theta}} \operatorname{Tr}\left[(|\vec{m}\rangle\langle\vec{m}| \otimes |\vec{m}\rangle\langle\vec{m}|)(\mathcal{B}_{(\vec{v}\oplus\vec{m},\vec{\theta})} \otimes \mathcal{C}_{(\vec{v}\oplus\vec{m},\vec{\theta})}) \circ \mathcal{A}\left(\sigma^{(\vec{v},\vec{\theta})}_{QB|\top}\right)\right]. \tag{5.4}$$

(Here, we sum over $\vec{v}$ because $\sigma^{(\vec{v},\vec{\theta})}_{QB|\top}$ is subnormalised. Alternatively, one can also renormalise these states and take an expectation. Both methods are equivalent since the expression in the trace scales linearly in the normalisation factor.) We can assume that the acceptance probability of $\mathcal{P}$ in the interactive protocol satisfies $\Pr[\top] \geq 1/\operatorname{poly}(\lambda)$ as otherwise, the success probability of the adversary $(\mathcal{P}, \mathcal{A}, \mathcal{B}, \mathcal{C})$ in the cloning experiment would decay faster than an inverse polynomial in λ, in which case Theorem 5.11 is trivially satisfied. Under this assumption, it follows from Corollary 4.35 that there exists $\varepsilon(\lambda) = 1/\operatorname{poly}(\lambda)$ such that for any $\vec{\theta} \in \{0,1\}^n$,

$$\sum_{\vec{v}\in\{0,1\}^n} |\vec{v}\rangle\langle\vec{v}|_W \otimes \sigma^{(\vec{v},\vec{\theta})}_{QB|\top} \overset{c}{\approx}_{\varepsilon(\lambda)} \frac{1}{2^n} \sum_{\vec{v}\in\{0,1\}^n} |\vec{v}\rangle\langle\vec{v}|_W \otimes \left(\bigotimes_i H^{\theta_i}|v_i\rangle\langle v_i|H^{\theta_i}\right)_Q \otimes \alpha'_B, \tag{5.5}$$

where α'_B is a (normalised) quantum state which is independent of $\vec{v}$ and $\vec{\theta}$ (and $\vec{m}$). From this it follows that for any m and $\vec{\theta}$,

$$\sum_{\vec{v}} \operatorname{Tr}\left[(|\vec{m}\rangle\langle\vec{m}| \otimes |\vec{m}\rangle\langle\vec{m}|)(\mathcal{B}_{(\vec{v}\oplus\vec{m},\vec{\theta})} \otimes \mathcal{C}_{(\vec{v}\oplus\vec{m},\vec{\theta})}) \circ \mathcal{A}\left(\sigma^{(\vec{v},\vec{\theta})}_{QB|\top}\right)\right]$$
$$\leq \frac{1}{2^n} \sum_{\vec{v}} \operatorname{Tr}\left[(|\vec{m}\rangle\langle\vec{m}| \otimes |\vec{m}\rangle\langle\vec{m}|)(\mathcal{B}_{(\vec{v}\oplus\vec{m},\vec{\theta})} \otimes \mathcal{C}_{(\vec{v}\oplus\vec{m},\vec{\theta})}) \circ \mathcal{A}\left(\left(\bigotimes_i H^{\theta_i}|v_i\rangle\langle v_i|H^{\theta_i}\right)_Q \otimes \alpha'_B\right)\right] + \varepsilon(\lambda).$$
$$\tag{5.6}$$

This is the case because the adversary $(\mathcal{P}, \mathcal{A}, \mathcal{B}, \mathcal{C})$ is efficient, so given the state on the l.h.s. or r.h.s. of Equation (5.5), one can efficiently estimate the quantity on the l.h.s. or r.h.s. of Equation (5.6), respectively.

Since α'_B is a fixed and efficiently preparable state, we can consider a modified cloning map $\widetilde{\mathcal{A}}$ that only acts on system Q (rather than on QB) and is defined by

$$\widetilde{\mathcal{A}}(\rho_Q) = \mathcal{A}\left(\rho_Q \otimes \alpha'_B\right)$$

Using Equation (5.6), we can bound the cloning advantage as follows:

$$\Pr\left[\mathsf{CloneExp}_{\Sigma,\mathsf{CC}}\left(1^\lambda, (\mathcal{P},\mathcal{A},\mathcal{B},\mathcal{C})\right) = 1\right]$$

$$\leq \underset{\vec{m}}{\mathbb{E}}\,\underset{\vec{v}}{\mathbb{E}}\,\underset{\vec{\theta}}{\mathbb{E}}\,\mathrm{Tr}\left[\left(|\vec{m}\rangle\langle\vec{m}| \otimes |\vec{m}\rangle\langle\vec{m}|\right)\left(\mathcal{B}_{(\vec{v}\oplus\vec{m},\vec{\theta})} \otimes \mathcal{C}_{(\vec{v}\oplus\vec{m},\vec{\theta})}\right) \circ \mathcal{A}\left(\left(\bigotimes_i H^{\theta_i}|v_i\rangle\langle v_i|H^{\theta_i}\right)_Q \otimes \alpha'_B\right)\right] + \varepsilon(\lambda)$$

$$= \underset{\vec{m}}{\mathbb{E}}\,\underset{\vec{r}}{\mathbb{E}}\,\underset{\vec{\theta}}{\mathbb{E}}\,\mathrm{Tr}\left[\left(|\vec{m}\rangle\langle\vec{m}| \otimes |\vec{m}\rangle\langle\vec{m}|\right)\left(\mathcal{B}_{(\vec{r},\vec{\theta})} \otimes \mathcal{C}_{(\vec{r},\vec{\theta})}\right) \circ \widetilde{\mathcal{A}}\left(\left(\bigotimes_i H^{\theta_i}|r_i \oplus m_i\rangle\langle r_i \oplus m_i|H^{\theta_i}\right)_Q\right)\right] + \varepsilon(\lambda)$$

We can now use the fact that Theorem 5.9 must hold for the cloning attack $(\widetilde{\mathcal{A}},\mathcal{B},\mathcal{C})$, and get:

$$\leq \left(\frac{1}{2} + \frac{1}{2\sqrt{2}}\right)^{n(\lambda)} + \varepsilon(\lambda)$$

$$= 2^{-n(\lambda)+\widetilde{t}(\lambda)},$$

where we defined $\widetilde{t}(\lambda) := t(\lambda) + \log\left(1 + 2^{n(\lambda)-t(\lambda)+\log\varepsilon(\lambda)}\right)$ and $t(\lambda) = n(\lambda)\log(1+1/\sqrt{2})$. $\qquad\square$

6 Quantum copy-protection

Our second application of our parallel RSP protocol for BB84 states is the task of quantum copy-protection – an idea first proposed by Aaronson [Aar09]. Here, we imagine that a vendor wishes to encode a program into a quantum state in a way that enables a recipient to run the program, but not to create functionally equivalent "pirated" copies. Let us begin by introducing a few relevant definitions.

Definition 6.1 (Quantum copy-protection scheme). Let $\mathscr{F} = \bigcup_{\lambda \in \mathbb{N}} \mathcal{F}_\lambda$ be a class of efficiently computable functions $f : \mathcal{X} \to \mathcal{Y}$ with domain $\mathcal{X}$ and range $\mathcal{Y}$. A quantum copy-protection (QCP) scheme for the class $\mathscr{F}$ is a pair of QPT algorithms $\mathsf{QCP} = (\mathsf{Protect}, \mathsf{Eval})$ defined as follows:

$\mathsf{QCP.Protect}(1^\lambda, d_f) \to \rho$: takes as input the security parameter 1^λ and a classical description d_f of a function $f \in \mathcal{F}_\lambda$, and outputs a (possibly mixed) quantum state ρ.

$\mathsf{QCP.Eval}(1^\lambda, \rho, x) \to \rho' \otimes |y\rangle\langle y|$: takes as input the security parameter 1^λ, a quantum state ρ and an input $x \in \mathcal{X}$, and outputs a bipartite state $\rho' \otimes |y\rangle\langle y|$ with $y \in \mathcal{Y}$.

Slightly abusing notation, we occasionally ignore the post-evaluation state ρ' and simply identify the output of the procedure $\mathsf{QCP.Eval}(1^\lambda, \rho, x)$ with a classical outcome denoted by $y \in \mathcal{Y}$.

We say that a QCP scheme is δ-correct if, for any $\lambda \in \mathbb{N}$, any $f \in \mathcal{F}_\lambda$, and any input $x \in \mathcal{X}$ to f:

$$\Pr\left[\mathsf{QCP.Eval}(1^\lambda, \rho, x) = f(x) \,:\, \rho \leftarrow \mathsf{QCP.Protect}(1^\lambda, d_f)\right] \geq 1 - \delta(\lambda).$$

Note that the probability above comes from the procedure QCP.Eval of the QCP scheme. If $\delta(\lambda) = \mathrm{negl}(\lambda)$, we simply call a copy-protection scheme *correct*. By the Gentle Measurement Lemma [Win99] it is easy to see that a δ-correct scheme is reusable in the following sense: after performing QCP.Eval to ρ it is possible to rewind the procedure to obtain a state that is within trace distance $\sqrt{\delta}$ of the original state ρ.

Informally, we say that a QCP scheme QCP $=$ (Protect, Eval) is *secure* if no QPT adversary can produce two "copies" of a copy-protected program $\rho \leftarrow$ QCP.Protect$(1^\lambda, d_f)$ that can both be used to evaluate f. We formalise the security of copy-protection schemes by means of the following security experiment.

Definition 6.2 (Piracy experiment). Let QCP $=$ (Protect, Eval) be a copy-protection scheme for a class of functions $\mathscr{F} = \bigcup_{\lambda \in \mathbb{N}} \mathcal{F}_\lambda$ with domain $\mathcal{X}$ and range $\mathcal{Y}$. Let $\mathcal{D}_{\mathscr{F}} = \{\mathcal{D}_{\mathcal{F}_\lambda}\}_{\lambda \in \mathbb{N}}$ be an ensemble of distributions over $\mathcal{F}_\lambda$ and let $\mathcal{D}_{\mathcal{X}} = \{\mathcal{D}_{\mathcal{X}}(f)\}_{f \in \mathcal{F}_\lambda}$ be an ensemble of distributions over function inputs $\mathcal{X}$. The security game (which we call the *piracy experiment*) takes place between a challenger and an adversary consisting of a triplet of QPT algorithms $(\mathcal{A}, \mathcal{B}, \mathcal{C})$:

1. The challenger samples $f \leftarrow \mathcal{D}_{\mathcal{F}_\lambda}$ and sends the program $\rho \leftarrow$ QCP.Protect$(1^\lambda, d_f)$ to $\mathcal{A}$.

2. $\mathcal{A}$ applies an efficient CPTP map to map ρ into a bipartite state ρ_{BC} on systems BC, and sends system B to $\mathcal{B}$ and system C to $\mathcal{C}$ (who are not allowed to communicate from this step onward).

3. The challenger samples a pair $(x_B, x_C) \leftarrow \mathcal{D}_{\mathcal{X}}(f) \times \mathcal{D}_{\mathcal{X}}(f)$, and sends x_B to $\mathcal{B}$ and x_C to $\mathcal{C}$.

4. $\mathcal{B}$ and $\mathcal{C}$ output values $y_B \in \mathcal{Y}$ and $y_C \in \mathcal{Y}$, respectively, and send them to the challenger. The challenger outputs 1, if $y_B = f(x_B)$ and $y_C = f(x_C)$ (i.e., the adversary has succeeded) and 0, otherwise (i.e., the adversary has failed).

We let the random variable $\mathsf{PiracyExp}^{\mathsf{QCP}}_{\mathcal{D}_{\mathscr{F}}, \mathcal{D}_{\mathcal{X}}}(1^\lambda, (\mathcal{A}, \mathcal{B}, \mathcal{C}))$ denote the output bit of the challenger.

Definition 6.3 (Secure quantum copy-protection). Let QCP $=$ (Protect, Eval) be a QCP scheme for a class of functions $\mathscr{F} = \bigcup_{\lambda \in \mathbb{N}} \mathcal{F}_\lambda$. Let $\mathcal{D}_{\mathscr{F}} = \{\mathcal{D}_{\mathcal{F}_\lambda}\}_{\lambda \in \mathbb{N}}$ be an ensemble of distributions over $\mathcal{F}_\lambda$ and let $\mathcal{D}_{\mathcal{X}} = \{\mathcal{D}_{\mathcal{X}}(f)\}_{f \in \mathcal{F}_\lambda}$ be an ensemble of distributions over $\mathcal{X}$. Then, QCP $=$ (Protect, Eval) is called $(\mathcal{D}_{\mathscr{F}}, \mathcal{D}_{\mathcal{X}}, \gamma)$-secure if, for any triplet of QPT algorithms $(\mathcal{A}, \mathcal{B}, \mathcal{C})$, it holds that

$$\Pr\left[\mathsf{PiracyExp}^{\mathsf{QCP}}_{\mathcal{D}_{\mathscr{F}}, \mathcal{D}_{\mathcal{X}}}(1^\lambda, (\mathcal{A}, \mathcal{B}, \mathcal{C})) = 1\right] \leq p^{\mathrm{triv}}_{\mathcal{D}_{\mathscr{F}}, \mathcal{D}_{\mathcal{X}}} + \gamma(\lambda),$$

where $p^{\mathrm{triv}}_{\mathcal{D}_{\mathscr{F}}, \mathcal{D}_{\mathcal{X}}}$ is the trivial winning probability that is always possible due to correctness: $\mathcal{A}$ forwards the original copy-protected program to one of the parties, say $\mathcal{B}$ (who then evaluates it to obtain the correct output), and the other party, say $\mathcal{C}$, has to guess at random [CMP20].

6.1 Quantum encryption with wrong-key detection

Recent work [CMP20] has introduced a QCP scheme for multi-bit point functions in the QROM based on unclonable encryption with *wrong-key detection* (WKD), a property which enables the decryption procedure to recognise incorrect keys. Contrary to [CMP20], our new QCP scheme for multi-bit point functions does not require the QROM assumption and instead relies on pair-wise independent permutations [NR97], which exist unconditionally and have previously been used in the context of classical obfuscation of point functions [CKVW10]. Let us first introduce some relevant definitions.

Definition 6.4 (Pair-wise independent permutation family). Let $n \in \mathbb{N}$ be an integer. A set of functions $\mathcal{P}_n$ is called a *pair-wise independent permutation family* over $\{0, 1\}^n$ if

- every $\pi \in \mathcal{P}_n$ is a permutation on $\{0,1\}^n$, and

- for any distinct $x_1, x_2 \in \{0,1\}^n$ and random function $\pi \xleftarrow{\$} \mathcal{P}_n$, the pair $(\pi(x_1), \pi(x_2))$ is uniformly distributed on the set of distinct pairs $S_n = \{(x,y) \in \{0,1\}^n \times \{0,1\}^n : x \neq y\}$.

To give an example, we consider the following simple construction of a pair-wise independent permutation family due to Naor and Reingold [NR97]. Let $n \in \mathbb{N}$ be an integer and let $\mathsf{GF}(2^n)$ be a finite field of order 2^n. For $a, b \in \mathsf{GF}(2^n)$, we consider the family of functions defined by

$$f_{(a,b)}(x) := a \cdot x + b, \qquad \forall x \in \mathsf{GF}(2^n).$$

Then, the following set of functions $\mathcal{P}_n$ forms a pair-wise independent permutation family:

$$\mathcal{P}_n = \left\{ f_{(a,b)} : \mathsf{GF}(2^n) \to \mathsf{GF}(2^n) \ : \ a, b \in \mathsf{GF}(2^n), a \neq 0 \right\}.$$

Because the finite field $\mathsf{GF}(2^n)$ of order 2^n can be identified with $\{0,1\}^n$, we conclude that the above construction yields a pairwise independent permutation family over the set of binary strings $\{0,1\}^n$.

We make the following simple observation about pair-wise independent permutation families which we later make use of in the section on quantum copy-protection with a classical client.

Lemma 6.5. *Let $n \in \mathbb{N}$ and let $\mathcal{P}_n$ be a pairwise independent permutation family. Then, the family of inverse permutations $\mathcal{P}_n^{-1} = \{\pi^{-1} : \pi \in \mathcal{P}_n\}$ is also a pairwise independent permutation family.*

Proof. Let us verify the two properties in Definition 6.4. Fix a pairwise independent permutation family $\mathcal{P}_n$. For every $\pi^{-1} \in \mathcal{P}_n^{-1}$, the function π^{-1} is clearly a permutation whenever π is a permutation. This proves the first property. For the second property, we fix distinct $x_1, x_2 \in \{0,1\}^n$ and a random permutation π^{-1} in the family $\mathcal{P}_n^{-1}$. This implies that π is also a random permutation with $\pi \circ \pi^{-1} = \mathsf{id}$, and

$$\left(\pi^{-1}(x_1), \pi^{-1}(x_2) \right) = \left((\pi \circ \pi^{-1} \circ \pi^{-1})(x_1), (\pi \circ \pi^{-1} \circ \pi^{-1})(x_2) \right)$$
$$= \left(\pi(y_1), \pi(y_2) \right),$$

where $y_1 := (\pi^{-1} \circ \pi^{-1})(x_1)$ and $y_2 := (\pi^{-1} \circ \pi^{-1})(x_2)$ are distinct values in $\{0,1\}^n$. Thus, by the assumption that $\mathcal{P}_n$ is a pairwise independent permutation family, the pair $(\pi(y_1), \pi(y_2))$ is uniform in $S_n = \{(x,y) \in \{0,1\}^n \times \{0,1\}^n : x \neq y\}$. This proves the claim. $\square$

We use the following definition of wrong-key detection for QECM schemes, inspired by [CMP20] who consider quantum secret-key encryption schemes, more generally.

Definition 6.6 (Wrong-key detection for QECM schemes). A QECM scheme $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ satisfies the wrong-key detection (WKD) property if, for every $k' \neq k \leftarrow \mathsf{KeyGen}(1^\lambda)$ and plaintext m,

$$\mathrm{Tr}[(\mathbb{1} - |\bot\rangle\langle\bot|)\mathsf{Dec}_{k'} \circ \mathsf{Enc}_k(m)] \leq \mathsf{negl}(\lambda).$$

Using the notion of pair-wise independent permutations, we can now state the following transformation which enables us to turn any QECM scheme into another QECM scheme with wrong-key detection.

Our construction is inspired by Canetti et al. [CKVW10] who previously introduced the same transformation in the context classical obfuscation of point functions.

Construction 2 (Generic transformation for WKD). Let $\lambda \in \mathbb{N}$ be the security parameter. Let $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a QECM scheme with key space $\mathcal{K} = \{0,1\}^\lambda$ and let $\mathcal{P}_\lambda$ be a pair-wise independent permutation family. Consider the following QECM scheme $\Sigma' = (\mathsf{KeyGen}', \mathsf{Enc}', \mathsf{Dec}')$ defined by:

$\mathsf{KeyGen}'(1^\lambda) \to k$: this is the same as $\mathsf{KeyGen}(1^\lambda)$.

$\mathsf{Enc}'(k, m) \to \mathsf{ct}$: takes as input a key $k \in \{0,1\}^\lambda$ and plaintext $m \in \{0,1\}^\lambda$ and proceeds as follows:

1. Sample a uniformly random string $r \xleftarrow{\$} \{0,1\}^\lambda$.
2. Sample a random pair-wise independent permutation $\pi \xleftarrow{\$} \mathcal{P}_\lambda$.
3. Output the ciphertext $\mathsf{ct} = \mathsf{Enc}_{\pi(k)}(r||m) \otimes |r\rangle\langle r| \otimes |\pi\rangle\langle\pi|$.

$\mathsf{Dec}'(k, \mathsf{ct}) \to m' \text{ or } \perp$: on input key $k \in \{0,1\}^n$ and ciphertext ct, decrypt as follows:

1. Parse the ciphertext as $\rho \otimes |r\rangle\langle r| \otimes |\pi\rangle\langle\pi| \leftarrow \mathsf{ct}$.
2. Compute $|r'\rangle\langle r'| \otimes |m'\rangle\langle m'| \leftarrow \mathsf{Dec}_{\pi(k)}(\rho)$.
3. Output m' if $r' = r$, and output $\perp$ otherwise.

Proposition 6.7. *Let* $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *be any* IND-*secure* QECM *scheme and let* $\Sigma' = (\mathsf{KeyGen}', \mathsf{Enc}', \mathsf{Dec}')$ *be the associated transformation in* [Construction 2](). *Then, the* QECM *scheme* Σ' *is correct,* IND-*secure and has the* WKD *property from* [Definition 6.6]().

Proof. The correctness of the scheme $\Sigma' = (\mathsf{KeyGen}', \mathsf{Enc}', \mathsf{Dec}')$ is immediate. Let us first show that Σ' is IND-secure according to [Definition 5.2](). To this end, we fix a QPT adversary $\mathcal{A} = (\mathcal{M}, \mathcal{D})$ and let $\rho_{ME} \leftarrow \mathcal{M}(1^\lambda)$. Let $k \leftarrow \mathsf{KeyGen}'(1^\lambda)$ and $\rho_E = \mathrm{Tr}_M[\rho_{ME}]$. In what follows, we let $r \xleftarrow{\$} \{0,1\}^n$ be a random string and let $\pi \xleftarrow{\$} \mathcal{P}_n$ be a pair-wise independent permutation. Note that, since π is a pair-wise independent permutation, $\pi(k)$ is uniform in $\{0,1\}^n$ and thus has the same distribution as a uniformly random string $k \xleftarrow{\$} \{0,1\}^n$. We can now verify the IND-security of Σ' as follows:

$$\left| \Pr\left[\mathcal{D}\big((\mathsf{Enc}'_k \otimes \mathsf{id}_E)\rho_{ME}\big) = 1\right] - \Pr\left[\mathcal{D}\big((\mathsf{Enc}'_k \otimes \mathsf{id}_E)(|0\rangle\langle 0|_M \otimes \rho_E)\big) = 1\right] \right|$$

$$= \left| \Pr\left[\mathcal{D}\Big(\sum_{m \in \mathcal{M}} \mathsf{Enc}_{\pi(k)}(r||m) \otimes |r\rangle\langle r| \otimes |\pi\rangle\langle\pi| \otimes \mathrm{Tr}_M\Big[(|r,m\rangle\langle r,m|_M \otimes \mathbb{1}_E)\rho_{ME}\Big]\Big) = 1\right] \right.$$

$$\left. - \Pr\left[\mathcal{D}\Big(\mathsf{Enc}_{\pi(k)}(r||0) \otimes |r\rangle\langle r| \otimes |\pi\rangle\langle\pi| \otimes \rho_E\Big) = 1\right] \right|$$

$$= \left| \Pr\left[\mathcal{D}\Big(\sum_{m \in \mathcal{M}} \mathsf{Enc}_k(r||m) \otimes |r\rangle\langle r| \otimes |\pi\rangle\langle\pi| \otimes \mathrm{Tr}_M\Big[(|r,m\rangle\langle r,m|_M \otimes \mathbb{1}_E)\rho_{ME}\Big]\Big) = 1\right] \right.$$

$$\left. - \Pr\left[\mathcal{D}\Big(\mathsf{Enc}_k(r||0) \otimes |r\rangle\langle r| \otimes |\pi\rangle\langle\pi| \otimes \rho_E\Big) = 1\right] \right|$$

$$= \left| \Pr\left[\mathcal{D}\Big(\sum_{m \in \mathcal{M}} \mathsf{Enc}_k(r||m) \otimes |r\rangle\langle r| \otimes \mathrm{Tr}_M\Big[(|r,m\rangle\langle r,m|_M \otimes \mathbb{1}_E)\rho_{ME}\Big]\Big) = 1\right] \right.$$

$$\left. - \Pr\left[\mathcal{D}\Big(\mathsf{Enc}_k(r||0) \otimes |r\rangle\langle r| \otimes \rho_E\Big) = 1\right] \right|$$

$$= \mathsf{negl}(\lambda).$$

The second to last inequality follows from the fact that π is completely independent of the ciphertext. In the last step, we used the assumption that the QECM scheme $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is IND-secure.

To verify the WKD property of Σ' we have to show that, for any $k' \neq k \leftarrow \mathsf{KeyGen}'(1^\lambda)$ and plaintext $m \in \{0,1\}^\lambda$,

$$\mathrm{Tr}\big[(\mathbb{1} - |\bot\rangle\langle\bot|)\mathsf{Dec}'_{k'} \circ \mathsf{Enc}'_k(m)\big] \leq \mathrm{negl}(\lambda).$$

By the assumption that π is a random pair-wise independent permutation, it follows for any pair of distinct keys $k' \neq k \leftarrow \mathsf{KeyGen}'(1^\lambda)$ and any plaintext $m \in \{0,1\}^\lambda$:

$$\mathrm{Tr}\big[(\mathbb{1} - |\bot\rangle\langle\bot|)\mathsf{Dec}'_{k'} \circ \mathsf{Enc}'_k(m)\big] = \Pr_{\pi \xleftarrow{\$} \mathcal{P}_\lambda, r \xleftarrow{\$} \{0,1\}^\lambda}\left[\mathsf{Dec}_{\pi(k')} \circ \mathsf{Enc}_{\pi(k)}(r||m) \text{ has pre-fix } r\right]$$

$$= \Pr_{r,k,k' \xleftarrow{\$} \{0,1\}^\lambda}\left[\mathsf{Dec}_{k'} \circ \mathsf{Enc}_k(r||m) \text{ has pre-fix } r\right] \leq \mathrm{negl}(\lambda).$$

Notice that the last inequality is a simple consequence of the IND-security of the QECM scheme given by $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$; namely, if the above expression were non-negligible with respect to Σ, one could construct an efficient adversary that could distinguish between encryptions of m as well as encryptions of the all-zero string, thereby violating the IND-security of Σ. This proves the claim. $\qquad\square$

Hybrid encryption with wrong-key detection. We conclude our section on quantum encryption with wrong-key detection with the following simple private-key *hybrid encryption* scheme, which we make use of in our section on QCP schemes with a classical client. Our hybrid encryption scheme combines a standard QECM scheme with the classical *one-time pad* given by $\mathsf{OTP} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$. Here, an encryption of a plaintext $\vec{m} \in \{0,1\}^\lambda$ via a key $\vec{k} \in \{0,1\}^\lambda$ is generated as $\mathsf{OTP}.\mathsf{Enc}_k(\vec{m}) = \vec{k} \oplus \vec{m}$.

Construction 3 (Hybrid encryption scheme). Let $\lambda \in \mathbb{N}$ be the security parameter. Let $\mathsf{QECM} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a QECM scheme and let $\mathsf{OTP} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be the classical one-time pad. We define the hybrid encryption scheme $\mathsf{HE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ as follows:

$\mathsf{HE}.\mathsf{KeyGen}(1^\lambda)$: this is the same procedure as $\mathsf{QECM}.\mathsf{KeyGen}(1^\lambda)$.

$\mathsf{HE}.\mathsf{Enc}(k, m)$: given as input a key k and a plaintext $m \in \{0,1\}^\lambda$, encrypt as follows:

    (i) sample a random string $r \xleftarrow{\$} \{0,1\}^\lambda$.

    (ii) output the ciphertext $(\mathsf{QECM}.\mathsf{Enc}_k(r), \mathsf{OTP}.\mathsf{Enc}_r(m))$.

$\mathsf{HE}.\mathsf{Dec}(k, c)$: given as input a key k and ciphertext $c = (c_0, c_1)$, decrypt as follows:

    (i) compute $r' = \mathsf{QECM}.\mathsf{Dec}_k(c_0)$

    (ii) output the plaintext $\mathsf{OTP}.\mathsf{Dec}_{r'}(c_1)$.

The following statement is a simple consequence of Proposition 6.7.

Corollary 6.8. *Let $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be an arbitrary QECM scheme and suppose that Σ is IND-secure. Further, let $\Sigma' = (\mathsf{KeyGen}', \mathsf{Enc}', \mathsf{Dec}')$ be the associated WKD transformation in Construction 2. Then, the hybrid encryption scheme $\mathsf{HE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ in Construction 3 instantiated with Σ' is correct, IND-secure and has the WKD property.*

6.2 Quantum copy-protection with a classical client

Let us now define the notion of quantum copy-protection with a classical client. This notion is similar to Definition 6.1 and enables a *classical* client to delegate a copy-protected program to a *quantum* receiver in a way that allows the receiver to execute the program, but prevents it from creating additional "pirated" copies.

Definition 6.9 (Quantum copy-protection scheme with a classical client). Let $\mathscr{F} = \bigcup_{\lambda \in \mathbb{N}} \mathcal{F}_\lambda$ be a class of efficiently computable functions $f : \mathcal{X} \to \mathcal{Y}$. A QCP scheme with a classical client (or, $\mathsf{QCP_{CC}}$ scheme) for the class $\mathscr{F}$ is a pair of efficient interactive quantum algorithms $\mathsf{QCP_{CC}} = (\mathsf{Protect}, \mathsf{Eval})$:

$\mathsf{QCP_{CC}}.\mathsf{Protect}(\mathcal{C}(1^\lambda, d_f), \mathcal{R}(1^\lambda)) \to \rho$ **or** $\bot$: this is an interactive protocol between a classical client $\mathcal{C}$ (who takes as input the parameter 1^λ and a classical description d_f of a function $f \in \mathcal{F}_\lambda$) and a quantum receiver $\mathcal{R}$ (who takes as input 1^λ). Provided that the protocol is successful with the flag $\top$, $\mathcal{R}$ is in possession of a quantum state ρ (which depends on f). Otherwise, $\mathcal{C}$ outputs $\bot$.

$\mathsf{QCP_{CC}}.\mathsf{Eval}(1^\lambda, \rho, x)$: takes as input the security parameter 1^λ, a quantum state ρ and an input $x \in \mathcal{X}$, and outputs a bipartite state $\rho' \otimes |y\rangle\langle y|$ for an outcome $y \in \mathcal{Y}$.

Similar to Definition 6.1, we occasionally ignore the post-evaluation state ρ' and simply identify the output of the procedure $\mathsf{QCP_{CC}}.\mathsf{Eval}(1^\lambda, \rho, x)$ with a classical outcome $y \in \mathcal{Y}$. We say that a $\mathsf{QCP_{CC}}$ scheme is δ-correct, if, for any $\lambda \in \mathbb{N}$, any $f \in \mathcal{F}_\lambda$, and any input $x \in \mathcal{X}$ to f:

$$\Pr\left[\mathsf{QCP_{CC}}.\mathsf{Eval}(1^\lambda, \rho, x) = f(x) \ : \ \rho \leftarrow \mathsf{QCP_{CC}}.\mathsf{Protect}(\mathcal{C}(1^\lambda, d_f,), R(1^\lambda))\right] \geq 1 - \delta(\lambda).$$

Let us now define a piracy experiment with a classical client, similar to Definition 6.2.

Definition 6.10 (Piracy experiment with a classical client). Let $\mathsf{QCP_{CC}} = (\mathsf{Protect}, \mathsf{Eval})$ be a $\mathsf{QCP_{CC}}$ scheme for a class of functions $\mathscr{F} = \bigcup_{\lambda \in \mathbb{N}} \mathcal{F}_\lambda$ with domain $\mathcal{X}$ and range $\mathcal{Y}$. Let $\mathcal{D}_\mathscr{F} = \{\mathcal{D}_{\mathcal{F}_\lambda}\}_{\lambda \in \mathbb{N}}$ be an ensemble of distributions over $\mathcal{F}_\lambda$ and let $\mathcal{D}_\mathcal{X} = \{\mathcal{D}_\mathcal{X}(f)\}_{f \in \mathcal{F}_\lambda}$ be an ensemble of distributions over $\mathcal{X}$. The security game (called *piracy experiment with a classical client*) takes place between a challenger and an adversary consisting of an efficient interactive quantum prover $\mathcal{P}$ and triplet of QPT algorithms $(\mathcal{A}, \mathcal{B}, \mathcal{C})$:

(i) The challenger samples a function $f \leftarrow \mathcal{D}_{\mathcal{F}_\lambda}$. To classically delegate a program for the functionality f according to $\mathsf{QCP_{CC}}$, the challenger takes the role of the classical client $\mathcal{C}$ and the prover $\mathcal{P}$ takes the role of the (possibly malicious) receiver $\mathcal{R}$ in the interactive protocol specified by $\mathsf{QCP_{CC}}.\mathsf{Protect}$. Provided that the protocol succeeds with $\mathtt{flag} = \top$, $\mathcal{P}$ is in possession of a quantum state ρ_A (which depends on f, as well as any additional information the adversary has collected throughout the protocol). Otherwise, if $\mathtt{flag} = \bot$, the adversary loses.

(ii) $\mathcal{A}$ applies an efficient CPTP map to map ρ_A into a bipartite state ρ_{BC} on systems BC, and sends system B to $\mathcal{B}$ and system C to $\mathcal{C}$ (who are not allowed to communicate from this step onward).

(iii) The challenger samples a pair $(x_B, x_C) \leftarrow \mathcal{D}_\mathcal{X}(f) \times \mathcal{D}_\mathcal{X}(f)$, and sends x_B to $\mathcal{B}$ and x_C to $\mathcal{C}$.

(iv) $\mathcal{B}$ and $\mathcal{C}$ output values $y_B \in \mathcal{Y}$ and $y_C \in \mathcal{Y}$, respectively, and send them to the challenger. The challenger outputs 1 if it holds that $y_B = f(x_B)$, $y_C = f(x_C)$ and $\mathtt{flag} = \top$ (i.e., the adversary has succeeded), and 0 otherwise (i.e., the adversary has failed).

We let the random variable $\mathsf{PiracyExp}_{\mathcal{D}_{\mathscr{F}},\mathcal{D}_{\mathcal{X}}}^{\mathsf{QCP_{CC}}}(1^\lambda, (\mathcal{P}, \mathcal{A}, \mathcal{B}, \mathcal{C}))$ denote the output bit of the challenger.

We formalise the notion of security with a classical client similar to Definition 6.3.

Definition 6.11 (Secure quantum copy-protection with a classical client). Let $\mathsf{QCP_{CC}} = (\mathsf{Protect}, \mathsf{Eval})$ be a $\mathsf{QCP_{CC}}$ scheme for a class of functions $\mathscr{F} = \bigcup_{\lambda \in \mathbb{N}} \mathcal{F}_\lambda$ with domain $\mathcal{X}$ and range $\mathcal{Y}$. Further, let $\mathcal{D}_{\mathscr{F}} = \{\mathcal{D}_{\mathcal{F}_\lambda}\}_{\lambda \in \mathbb{N}}$ be an ensemble of distributions over $\mathcal{F}_\lambda$ and let $\mathcal{D}_{\mathcal{X}} = \{\mathcal{D}_{\mathcal{X}}(f)\}_{f \in \mathcal{F}_\lambda}$ be an ensemble of distributions over $\mathcal{X}$. Then, $\mathsf{QCP_{CC}}$ is called $(\mathcal{D}_{\mathscr{F}}, \mathcal{D}_{\mathcal{X}}, \gamma)$-secure if, for any efficient interactive quantum prover $\mathcal{P}$ and any triplet of QPT algorithms $(\mathcal{A}, \mathcal{B}, \mathcal{C})$, it holds that

$$\Pr\left[\mathsf{PiracyExp}_{\mathcal{D}_{\mathscr{F}},\mathcal{D}_{\mathcal{X}}}^{\mathsf{QCP_{CC}}}(1^\lambda, (\mathcal{P}, \mathcal{A}, \mathcal{B}, \mathcal{C})) = 1\right] \leq p_{\mathcal{D}_{\mathscr{F}},\mathcal{D}_{\mathcal{X}}}^{\mathsf{triv}} + \gamma(\lambda).$$

6.3 Quantum copy-protection of multi-bit point functions with a classical client

In this section, we combine our parallel RSP protocol for BB84 states with a QECM scheme with wrong-key detection to construct a $\mathsf{QCP_{CC}}$ scheme for multi-bit point functions, which is described in Protocol 5. We begin with a formal definition. For integers $\mu \in \mathbb{N}$ and $\ell \in \mathbb{N}$, the class of multi-bit point functions $\mathscr{P}_{\mu,\ell}$ consists of functions $P_{y,m} \in \mathscr{P}_{\mu,\ell}$, where for a marked input $\vec{y} \in \{0,1\}^\mu$ and output $\vec{m} \in \{0,1\}^\ell$,

$$P_{y,m}(\vec{x}) = \begin{cases} \vec{m}, & \text{if } \vec{x} = \vec{y}, \\ \vec{0}^\ell, & \text{if } \vec{x} \neq \vec{y}. \end{cases}$$

Let $\lambda \in \mathbb{N}$ be the security parameter and let $\mu = 4\lambda$ and $\ell = 2\lambda$. Our construction in Protocol 5 enables a classical client to remotely prepare a copy-protected program for a multi-bit point function of the form $P_{y,m}$, where $y \xleftarrow{\$} \{0,1\}^\mu$ and $m \xleftarrow{\$} \{0,1\}^\ell$ are uniformly random. We first prove the *correctness* of Protocol 5.

Proposition 6.12 (Correctness). *Let $\lambda \in \mathbb{N}$ be the security parameter and let $\mu = 4\lambda$ and $\ell = 2\lambda$. Then, the scheme $\mathsf{QCP_{CC}} = (\mathsf{Protect}, \mathsf{Eval})$ in Protocol 5 is a correct $\mathsf{QCP_{CC}}$ scheme for the class $\mathscr{P}_{\mu,\ell}$.*

Proof. To prove correctness, we show that for any $\lambda \in \mathbb{N}$, $P_{y,m} \in \mathscr{P}_{\mu,\ell}$ and any input $\vec{x} \in \{0,1\}^{4\lambda}$ to $P_{y,m}$,

$$\Pr\left[\mathsf{QCP_{CC}}.\mathsf{Eval}(1^\lambda, \sigma, \vec{x}) \neq P_{y,m}(\vec{x}) : \sigma \leftarrow \mathsf{QCP_{CC}}.\mathsf{Protect}(\mathcal{C}(1^\lambda, d_{P_{y,m}}), \mathcal{R}(1^\lambda))\right] \leq \mathsf{negl}(\lambda).$$

Note that in this scenario we can assume that the receiver $\mathcal{R}$ is honest throughout Protocol 5. According to Proposition 3.1, $\mathcal{R}$ is accepted in the interactive protocol $\mathsf{QCP_{CC}}.\mathsf{Protect}(\mathcal{C}(1^\lambda, d_{P_{y,m}}), \mathcal{R}(1^\lambda))$ with probability negligibly close to 1 in the parameter λ (for parameter choices n at most polynomial in λ and δ at least inverse polynomial in λ). Provided that the protocol successful, the final state of $\mathcal{R}$ is given by

$$\sigma = \bigotimes_{i=1}^{2\lambda} H^{\theta_i}|v_i\rangle\langle v_i|H^{\theta_i} \otimes |\vec{r}\rangle\langle\vec{r}| \otimes |\pi\rangle\langle\pi| \otimes |\vec{u} \oplus \vec{m}\rangle\langle\vec{u} \oplus \vec{m}|. \tag{6.1}$$

where $\vec{v}$ and $\vec{\theta}$ are the strings recorded by the client $\mathcal{C}$, and where $\pi \xleftarrow{\$} \mathcal{P}_{4\lambda}$. We remark that $\pi(\vec{y}) = (\vec{s}\|\vec{\theta})$ with $\vec{s} = (\vec{s}_0\|\vec{s}_1)$, and that $\vec{r} = \vec{v}_0 \oplus \vec{s}_0$ and $\vec{u} = \vec{v}_1 \oplus \vec{s}_1$, for a uniformly random string $\vec{v} = (\vec{v}_0\|\vec{v}_1)$.

Let $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be the QECM scheme with WKD in Construction 2 instantiated with the conjugate coding encryption scheme in Construction 1. Further, let $\mathsf{HE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be the

Protocol 5. Quantum Copy-Protection of Multi-bit Point Functions with a Classical Client

Let $\lambda \in \mathbb{N}$ be the security parameter and let $\mu = 4\lambda$ and $\ell = 2\lambda$. Let $\mathscr{P}_{\mu,\ell}$ be the class of multi-bit point functions with input length μ and output length ℓ. We define the $\mathsf{QCP}_{\mathsf{CC}}$ scheme consisting of a pair of efficient (interactive) algorithms $\mathsf{QCP}_{\mathsf{CC}} = (\mathsf{Protect}, \mathsf{Eval})$ for the class $\mathscr{P}_{\mu,\ell}$ as follows:

- $\mathsf{QCP}_{\mathsf{CC}}.\mathsf{Protect}(\mathcal{C}(1^\lambda, d_{P_{y,m}}), \mathcal{R}(1^\lambda))$: this is an interactive protocol between a classical client $\mathcal{C}$ and a quantum receiver $\mathcal{R}$. The client $\mathcal{C}$ takes as input the security parameter 1^λ and a succinct classical description $d_{P_{y,m}}$ of a multi-bit point function $P_{y,m} \in \mathscr{P}_{\mu,\ell}$, whereas the receiver $\mathcal{R}$ takes as input the security parameter 1^λ. The protocol takes place as follows:

  (i) $\mathcal{C}$ samples a random pair-wise independent permutation $\pi \xleftarrow{\$} \mathcal{P}_{4\lambda}$ and lets $(\vec{s}||\vec{\theta}) \leftarrow \pi(\vec{y})$, where $\vec{s} \in \{0,1\}^{2\lambda}$ and $\vec{\theta} \in \{0,1\}^{2\lambda}$ are strings of length 2λ.

  (ii) The client $\mathcal{C}$ and the receiver $\mathcal{R}$ run the interactive RSP protocol for BB84 states (Protocol 3) with security parameter 1^λ, $n = 2\lambda$ and choice of basis $\vec{\theta} \in \{0,1\}^{2\lambda}$.

  (iii) Provided the protocol is successful, $\mathcal{C}$ obtains $\vec{v} \xleftarrow{\$} \{0,1\}^{2\lambda}$ and the receiver $\mathcal{R}$ obtains a 2λ-qubit state σ (which depends on $\vec{v}$ and $\vec{\theta}$). Otherwise, $\mathcal{C}$ outputs $\perp$.

  (iv) $\mathcal{C}$ parses $(\vec{s}_0||\vec{s}_1) \leftarrow \vec{s}$ with $\vec{s}_0, \vec{s}_1 \in \{0,1\}^\lambda$ and $(\vec{v}_0||\vec{v}_1) \leftarrow \vec{v}$ with $\vec{v}_0, \vec{v}_1 \in \{0,1\}^\lambda$, and then sends $(\vec{r}, \pi, \vec{u} \oplus \vec{m})$ to $\mathcal{R}$, where $\vec{r} = \vec{v}_0 \oplus \vec{s}_0$ and $\vec{u} = \vec{v}_1 \oplus \vec{s}_1$.

- $\mathsf{QCP}_{\mathsf{CC}}.\mathsf{Eval}(1^\lambda, \rho; x)$: takes as input 1^λ, an alleged program σ, and a string $x \in \{0,1\}^{4\lambda}$ (the input on which the program is to be evaluated), and runs the program as follows:

  (i) Parse the program as $\sigma \otimes |\vec{r}\rangle\langle\vec{r}| \otimes |\pi\rangle\langle\pi| \otimes |\vec{t}\rangle\langle\vec{t}| \leftarrow \rho$.

  (ii) Compute $(\vec{s}_x||\vec{\theta}_x) \leftarrow \pi(\vec{x})$ and parse $(\vec{s}_{x,0}||\vec{s}_{x,1}) \leftarrow \vec{s}_x$ with $\vec{s}_{x,0}, \vec{s}_{x,1} \in \{0,1\}^\lambda$.

  (iii) Append an ancillary qubit in the state $|0\rangle\langle0|$.

  (iv) Apply Hadamard operations $H^{\theta_{x,1}} \otimes \cdots \otimes H^{\theta_{x,2\lambda}}$ to the state ρ.

  (v) Coherently perform a two-outcome measurement of the resulting state to check whether the state has the pre-fix $\vec{r} \oplus \vec{s}_{x,0}$ or not, and store the resulting bit in the ancilla.

  (vi) If false, output 0^ℓ. Otherwise, if true, rewind the procedure and measure the state in the standard basis to obtain an outcome $\vec{w} = (\vec{w}_0||\vec{w}_1)$. Output $\vec{m}' = \vec{w}_1 \oplus \vec{s}_{x,1} \oplus \vec{t}$.

hybrid encryption scheme in Construction 3 instantiated with Σ. Notice that the state σ in Equation (6.1) corresponds precisely to a ciphertext generated by $\mathsf{HE.Enc}_{\vec{y}}(\vec{m})$ with randomness $\vec{u}$, where

$$\mathsf{HE.Enc}_{\vec{y}}(\vec{m}) = \Sigma.\mathsf{Enc}_{\vec{y}}(\vec{u}) \otimes |\mathsf{OTP.Enc}_{\vec{u}}(\vec{m})\rangle\langle\mathsf{OTP.Enc}_{\vec{u}}(\vec{m})|. \tag{6.2}$$

Let us now show the correctness of the program. We distinguish between two different types of inputs.

Suppose that $\vec{x} = \vec{y}$. Using that $\mathsf{HE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is a *correct* scheme, we get

$$\Pr\left[\mathsf{QCP}_{\mathsf{CC}}.\mathsf{Eval}(1^\lambda, \sigma, \vec{x}) \neq P_{y,m}(\vec{x}) \ : \ \sigma \leftarrow \mathsf{QCP}_{\mathsf{CC}}.\mathsf{Protect}(\mathcal{C}(1^\lambda, d_{P_{y,m}}), \mathcal{R}(1^\lambda))) \right]$$

$$= \Pr\left[\mathsf{HE.Dec}_{\vec{y}} \circ \left(\bigotimes_{i=1}^{2\lambda} H^{\theta_i}|v_i\rangle\langle v_i|H^{\theta_i} \otimes |\vec{r}\rangle\langle\vec{r}| \otimes |\pi\rangle\langle\pi| \otimes |\vec{u} \oplus \vec{m}\rangle\langle\vec{u} \oplus \vec{m}| \right) \neq \vec{m} \right]$$

$$= \Pr\left[\mathsf{HE.Dec}_{\vec{y}} \circ (\Sigma.\mathsf{Enc}_{\vec{y}}(\vec{u}) \otimes |\mathsf{OTP.Enc}_{\vec{u}}(\vec{m})\rangle\langle\mathsf{OTP.Enc}_{\vec{u}}(\vec{m})|) \neq \vec{m} \right]$$

$$= \Pr\left[\mathsf{HE.Dec}_{\vec{y}} \circ \mathsf{HE.Enc}_{\vec{y}}(\vec{m}) \neq \vec{m} \right]$$

$$\leq \mathsf{negl}(\lambda).$$

Finally, suppose that $\vec{x} \neq \vec{y}$. By Corollary 6.8, $\mathsf{HE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ has the WKD property. Thus,

$$\Pr\left[\mathsf{QCP}_{\mathsf{CC}}.\mathsf{Eval}(1^\lambda, \sigma, \vec{x}) \neq P_{y,m}(\vec{x}) \ : \ \sigma \leftarrow \mathsf{QCP}_{\mathsf{CC}}.\mathsf{Protect}(\mathcal{C}(1^\lambda, d_{P_{y,m}}), \mathcal{R}(1^\lambda))) \right]$$

$$= \Pr\left[\mathsf{HE.Dec}_{\vec{x}} \circ \left(\bigotimes_{i=1}^{2\lambda} H^{\theta_i}|v_i\rangle\langle v_i|H^{\theta_i} \otimes |\vec{r}\rangle\langle\vec{r}| \otimes |\pi\rangle\langle\pi| \otimes |\vec{u} \oplus \vec{m}\rangle\langle\vec{u} \oplus \vec{m}| \right) \neq \perp \right]$$

$$= \Pr\left[\mathsf{HE.Dec}_{\vec{x}} \circ (\Sigma.\mathsf{Enc}_{\vec{y}}(\vec{u}) \otimes |\mathsf{OTP.Enc}_{\vec{u}}(\vec{m})\rangle\langle\mathsf{OTP.Enc}_{\vec{u}}(\vec{m})|) \neq \perp \right]$$

$$= \Pr\left[\mathsf{HE.Dec}_{\vec{x}} \circ \mathsf{HE.Enc}_{\vec{y}}(\vec{m}) \neq \perp \right]$$

$$\leq \mathsf{negl}(\lambda).$$

This proves the claim. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\square$

Let us now prove the security of our $\mathsf{QCP}_{\mathsf{CC}}$ scheme Protocol 5 for the class of point functions $\mathscr{P}_{\mu,\ell}$ of the form $P_{y,m}$, where the marked strings $y \xleftarrow{\$} \{0,1\}^\mu$ and $m \xleftarrow{\$} \{0,1\}^\ell$ are chosen uniformly random.

Proposition 6.13 (Security). *Let $\lambda \in \mathbb{N}$ be the security parameter and let $\mu = 4\lambda$ and $\ell = 2\lambda$. Let $\mathscr{P}_{\mu,\ell}$ be the class of multi-bit point functions with input $\mathcal{X} = \{0,1\}^\mu$ output $\mathcal{Y} = \{0,1\}^\ell$. Let $\mathcal{D}_{\mathcal{X}}$ be an arbitrary ensemble of challenge distributions over $\mathcal{X}$ and let $\mathcal{D}_{\mathscr{P}_{\mu,\ell}}$ be the ensemble of multi-bit point functions that samples both the marked input and the marked output uniformly at random. Then, there exists $\gamma(\lambda) = 1/\mathsf{poly}(\lambda)$ such that the scheme $\mathsf{QCP}_{\mathsf{CC}} = (\mathsf{Protect}, \mathsf{Eval})$ in Protocol 5 is a $(\mathcal{D}_{\mathscr{P}_{\mu,\ell}}, \mathcal{D}_{\mathcal{X}}, \gamma)$-secure QCP_{CC} scheme for the class of multi-bit point functions $\mathscr{P}_{\mu,\ell}$. In other words, for any (interactive) QPT algorithms $(\mathcal{P}, \mathcal{A}, \mathcal{B}, \mathcal{C})$, it holds that*

$$\Pr\left[\mathsf{PiracyExp}^{\mathsf{CP}_{\mathsf{CC}}}_{\mathcal{D}_{\mathscr{P}_{\mu,\ell}}, \mathcal{D}_{\mathcal{X}}}(1^\lambda, (\mathcal{P}, \mathcal{A}, \mathcal{B}, \mathcal{C})) = 1 \right] \leq p^{\mathsf{triv}}_{\mathcal{D}_{\mathscr{P}_{\mu,\ell}}, \mathcal{D}_{\mathcal{X}}} + \gamma(\lambda).$$

Proof. Let $\mathrm{Adv} = (\mathcal{A}, \mathcal{B}, \mathcal{C})$ denote the adversary in the challenge phase of the piracy experiment $\mathrm{PirExp}^{\mathsf{CPcc}}_{\mathcal{D}_{\mathscr{P}_{\mu,\ell}}, \mathcal{D}_{\mathcal{X}}}$. We consider two cases, namely when $p^{\mathrm{triv}}_{\mathcal{D}_{\mathscr{P}_{\mu,\ell}}, \mathcal{D}_{\mathcal{X}}} = 1$ and when $p^{\mathrm{triv}}_{\mathcal{D}_{\mathscr{P}_{\mu,\ell}}, \mathcal{D}_{\mathcal{X}}} < 1$ (depending on the challenge distribution $\mathcal{D}_{\mathcal{X}}$). In the former case, the scheme is trivially secure by definition and we are done. Hence, we will assume that $p^{\mathrm{triv}}_{\mathcal{D}_{\mathscr{P}_{\mu,\ell}}, \mathcal{D}_{\mathcal{X}}} < 1$ for the remainder of the proof. Note that in this case, the distribution $\mathcal{D}_{\mathcal{X}}$ has non-zero weight on the marked input y of a random point function $P_{y,m}$.

Let $(x_B, x_C) \leftarrow \mathcal{D}_{\mathcal{X}}$ denote the inputs received by the algorithms $\mathcal{B}$ and $\mathcal{C}$ during the challenge phase, and let y_B and y_C be the outputs returned by $\mathcal{B}$ and $\mathcal{C}$, respectively. We can express the probability that Adv succeeds at the challenge phase (i.e., that $y_B = P_{y,m}(x_B)$ and $y_C = P_{y,m}(x_C)$) as follows:

$$\Pr[\mathrm{Adv}\ \mathrm{wins}] = \Pr[\mathrm{Adv}\ \mathrm{wins} \mid x_B \neq y \neq x_C] \cdot \Pr[x_B \neq y \neq x_C] + \Pr[\mathrm{Adv}\ \mathrm{wins} \mid x_B = y \neq x_C] \cdot \Pr[x_B = y \neq x_C]$$
$$+ \Pr[\mathrm{Adv}\ \mathrm{wins} \mid x_B \neq y = x_C] \cdot \Pr[x_B \neq y = x_C] + \Pr[\mathrm{Adv}\ \mathrm{wins} \mid x_B = y = x_C] \cdot \Pr[x_B = y = x_C].$$

Without loss of generality, we assume that $\Pr[x_B = y \neq x_C] \leq \Pr[x_B \neq y = x_C]$. Hence,

$$\Pr[\mathrm{Adv}\ \mathrm{wins}] \leq \Pr[\mathrm{Adv}\ \mathrm{wins} \mid x_B \neq y \neq x_C] \cdot \Pr[x_B \neq y \neq x_C]$$
$$+ \big(\Pr[\mathrm{Adv}\ \mathrm{wins} \mid x_B = y \neq x_C] + \Pr[\mathrm{Adv}\ \mathrm{wins} \mid x_B \neq y = x_C] \big) \cdot \Pr[x_B \neq y = x_C]$$
$$+ \Pr[\mathrm{Adv}\ \mathrm{wins} \mid x_B = y = x_C] \cdot \Pr[x_B = y = x_C]. \tag{6.3}$$

Let us now state the following simple inequality. By first applying the union bound and then using that $\mathcal{B}$ and $\mathcal{C}$ are non-signalling, we find that:

$$\Pr[\mathrm{Adv}\ \mathrm{wins} \mid x_B = y = x_C]$$
$$= \Pr[\mathcal{B}\ \mathrm{succeeds} \wedge \mathcal{C}\ \mathrm{succeeds} \mid x_B = y = x_C]$$
$$\geq \Pr[\mathcal{B}\ \mathrm{succeeds} \mid x_B = y = x_C] + \Pr[\mathcal{C}\ \mathrm{succeeds} \mid x_B = y = x_C] - 1$$
$$= \Pr[\mathcal{B}\ \mathrm{succeeds} \mid x_B = y \neq x_C] + \Pr[\mathcal{C}\ \mathrm{succeeds} \mid x_B \neq y = x_C] - 1$$
$$\geq \Pr[\mathrm{Adv}\ \mathrm{wins} \mid x_B = y \neq x_C] + \Pr[\mathrm{Adv}\ \mathrm{wins} \mid x_B \neq y = x_C] - 1. \tag{6.4}$$

Plugging this into Equation (6.3), we obtain the following upper bound:

$$\Pr[\mathrm{Adv}\ \mathrm{wins}] \leq \Pr[\mathrm{Adv}\ \mathrm{wins} \mid x_B \neq y \neq x_B] \cdot \Pr[x_B \neq y \neq x_C]$$
$$+ \big(1 + \Pr[\mathrm{Adv}\ \mathrm{wins} \mid x_B = y = x_C] \big) \cdot \Pr[x_B \neq y = x_C]$$
$$+ \Pr[\mathrm{Adv}\ \mathrm{wins} \mid x_B = y = x_C] \cdot \Pr[x_B = y = x_C]$$
$$\leq \Pr[x_B \neq y \neq x_C] + \Pr[x_B \neq y = x_C] + 2 \Pr[\mathrm{Adv}\ \mathrm{wins} \mid x_B = y = x_C]$$
$$= p^{\mathrm{triv}}_{\mathcal{D}_{\mathscr{P}_{\mu,\ell}}, \mathcal{D}_{\mathcal{X}}} + 2 \Pr[\mathrm{Adv}\ \mathrm{wins} \mid x_B = y = x_C]. \tag{6.5}$$

In the last line, we used the assumption that $\Pr[x_B = y \neq x_C] \leq \Pr[x_B \neq y = x_C]$ together with the following simple identity for the trivial guessing probability:

$$p^{\mathrm{triv}}_{\mathcal{D}_{\mathscr{P}_{\mu,\ell}}, \mathcal{D}_{\mathcal{X}}} = \Pr[x_B \neq y \neq x_C] + \max \big\{ \Pr[x_B \neq y = x_C], \Pr[x_B = y \neq x_C] \big\}.$$

We complete the proof by showing that $\Pr[\mathrm{Adv}\ \mathrm{wins} \mid x_B = y = x_C] \leq \varepsilon(\lambda) + \mathrm{negl}(\lambda)$, for some function $\varepsilon(\lambda) = 1/\mathrm{poly}(\lambda)$. This implies that

$$\Pr[\mathrm{Adv}\ \mathrm{wins}] \leq p^{\mathrm{triv}}_{\mathcal{D}_{\mathscr{P}_{\mu,\ell}}, \mathcal{D}_{\mathcal{X}}} + \varepsilon(\lambda) + \mathrm{negl}(\lambda). \tag{6.6}$$

We will now use the adversary $(\mathcal{P}, \mathcal{A}, \mathcal{B}, \mathcal{C})$ against the scheme $\mathsf{QCP}_{\mathsf{CC}}$ who receives the challenge pair consisting of $x_B = y$ and $x_C = y$ to construct an adversary $(\mathcal{P}', \mathcal{A}', \mathcal{B}', \mathcal{C}')$ against the unclonable security of the classical-client QECM scheme $\Sigma = (\mathsf{Enc}, \mathsf{Dec})$ in Protocol 4.

The adversary $(\mathcal{P}', \mathcal{A}', \mathcal{B}', \mathcal{C}')$ against the unclonable security of Σ proceeds as follows:

(i) $\mathcal{P}'$ runs the prover $\mathcal{P}$ during the execution of Protocol 4, resulting in a state $\sigma_{Q_R B|\top}^{(\vec{v}_R, \vec{\theta}_R)}$ in systems $Q_R B$ conditioned on the event that protocol is successful with flag $\top$ and that the classical system W takes the random value $\vec{v}_R \xleftarrow{\$} \{0,1\}^\lambda$, where $\vec{v}_R = \vec{s}_R \oplus \vec{u}$ for some $\vec{u} \xleftarrow{\$} \{0,1\}^\lambda$.

(ii) $\mathcal{A}'$ samples a random string $\vec{r} \xleftarrow{\$} \{0,1\}^\lambda$, a random pairwise-independent permutation $\pi \xleftarrow{\$} \mathcal{P}_{4\lambda}$, and random strings $\vec{s}_L, \vec{\theta}_L, \vec{t} \xleftarrow{\$} \{0,1\}^\lambda$, lets $\vec{v}_L = \vec{s}_L \oplus \vec{r}$ and runs $\mathcal{A}$ on the quantum state

$$\bigotimes_i H^{\theta_{L,i}} |v_{L,i}\rangle\langle v_{L,i}|_{Q_L} H^{\theta_{L,i}} \otimes \sigma_{Q_R B|\top}^{(\vec{v}_R, \vec{\theta}_R)} \otimes |\vec{r}\rangle\langle\vec{r}| \otimes |\pi\rangle\langle\pi| \otimes |\vec{t}\rangle\langle\vec{t}|.$$

and sends the outcome to the challenger, together with the circuit descriptions

$$\left\{ \mathcal{B}'_{(\vec{s}_R, \vec{\theta}_R)} \right\}_{\vec{s}_R, \vec{\theta}_R \in \{0,1\}^\lambda} \quad \text{and} \quad \left\{ \mathcal{C}'_{(\vec{s}_R, \vec{\theta}_R)} \right\}_{\vec{s}_R, \vec{\theta}_R \in \{0,1\}^\lambda}$$

which are defined as the following procedures:

- on input $(\vec{s}_R, \vec{\theta}_R)$, $\mathcal{B}'_{(\vec{s}_R, \vec{\theta}_R)}$ runs $\mathcal{B}_{\pi^{-1}(\vec{s}_L||\vec{s}_R||\vec{\theta}_L||\vec{\theta}_R)}$ to obtain an outcome $\vec{m}$, and outputs $\vec{m} \oplus \vec{t}$.
- on input $(\vec{s}_R, \vec{\theta}_R)$, $\mathcal{C}'_{(\vec{s}_R, \vec{\theta}_R)}$ runs $\mathcal{C}_{\pi^{-1}(\vec{s}_L||\vec{s}_R||\vec{\theta}_L||\vec{\theta}_R)}$ to obtain an outcome $\vec{m}$, and outputs $\vec{m} \oplus \vec{t}$.

Let $\sigma_{|\top} \leftarrow \mathsf{QCP}_{\mathsf{CC}}.\mathsf{Protect}(\mathcal{C}(1^\lambda, d_{P_{y,m}}), \mathcal{R}(1^\lambda))$ denote the final state of the interactive protocol conditioned on the event that challenger outputs the flag $\bot$. Because $\pi \xleftarrow{\$} \mathcal{P}_{4\lambda}$ is a random pairwise-independent permutation, so is its inverse π^{-1} (by Lemma 6.5). Hence, the string $\pi^{-1}(\vec{s}_L||\vec{s}_R||\vec{\theta}_L||\vec{\theta}_R)$ has the same distribution as a uniformly random string $\vec{y} \xleftarrow{\$} \{0,1\}^{4\lambda}$. Notice also that by definition, we have that $\vec{t} = \vec{u} \oplus \vec{m}$, for some $\vec{m} \xleftarrow{\$} \{0,1\}^\lambda$. Thus, bounding the success probability of the adversary $(\mathcal{P}, \mathcal{A}, \mathcal{B}, \mathcal{C})$ against $\mathsf{QCP}_{\mathsf{CC}}$ who receives the challenge pair consisting of $x_B = y$ and $x_C = y$ in terms of the the adversary $(\mathcal{P}', \mathcal{A}', \mathcal{B}', \mathcal{C}')$ against Σ, we find

$$\Pr[\mathsf{Adv \ wins} \mid x_B = y = x_C]$$
$$= \mathbb{E}_{\vec{y}} \mathrm{Tr}\left[(|\vec{m}\rangle\langle\vec{m}| \otimes |\vec{m}\rangle\langle\vec{m}|)(\mathcal{B}_{\vec{y}} \otimes \mathcal{C}_{\vec{y}}) \circ \mathcal{A} \circ \sigma_{|\top}\right]$$
$$= \mathbb{E}_{\vec{u}} \mathbb{E}_{\vec{v}_R} \mathbb{E}_{\vec{\theta}_R} \mathrm{Tr}\left[(|\vec{u}\rangle\langle\vec{u}| \otimes |\vec{u}\rangle\langle\vec{u}|)(\mathcal{B}'_{(\vec{v}_R \oplus \vec{u}, \vec{\theta}_R)} \otimes \mathcal{C}'_{(\vec{v}_R \oplus \vec{u}, \vec{\theta}_R)}) \circ \mathcal{A}' \circ \sigma_{Q_R B|\top}^{(\vec{v}_R, \vec{\theta}_R)}\right]$$
$$= \Pr\left[\mathsf{CloneExp}_{\mathsf{CC}, \Sigma}(1^\lambda, (\mathcal{P}', \mathcal{A}', \mathcal{B}', \mathcal{C}')) = 1\right]$$
$$\leq \left(\frac{1}{2} + \frac{1}{2\sqrt{2}}\right)^{2\lambda} + \varepsilon(\lambda),$$

for some function $\varepsilon(\lambda) = 1/\mathrm{poly}(\lambda)$ according to Theorem 5.11. Putting everything together, we find that there exists $\gamma = 1/\mathrm{poly}(\lambda)$ such that the advantage in the piracy experiment is at most

$$\Pr\left[\mathsf{PiracyExp}_{\mathcal{D}_{\mathscr{P}_{\mu,\ell}}, \mathcal{D}_{\mathcal{X}}}^{\mathsf{CP_{CC}}}(1^\lambda, (\mathcal{P}, \mathcal{A}, \mathcal{B}, \mathcal{C})) = 1\right] \leq p_{\mathcal{D}_{\mathscr{P}_{\mu,\ell}}, \mathcal{D}_{\mathcal{X}}}^{\mathsf{triv}} + \gamma(\lambda).$$

$\square$

7 Quantum computing on encrypted data

The next application of parallel RSP we consider is quantum computing on encrypted data. At a high level, this is a protocol between a weakly-quantum client, $\mathcal{C}$, and a quantum server, $\mathcal{S}$. By "weakly-quantum" we mean that the client can perform some quantum operations (preparing, storing and applying unitaries on quantum states) but does not have the ability to evaluate a generic quantum circuit, C. For this reason, it will interact with $\mathcal{S}$ in order to delegate the application of C on some input quantum state, $\rho_{\mathcal{C}}$. Before doing so, however, the client will encrypt $\rho_{\mathcal{C}}$, yielding a quantum ciphertext ct. In addition, it will also generate a quantum state, π, which will aid in evaluating the circuit C on the encrypted input, but does not require full quantum capabilities to prepare. Both ct and π are sent to the server and the two parties exchange classical (or quantum) messages for a number of rounds that is polynomial in the size of the input. At the end of the interaction, the client obtains a state ct^*, which, ideally, is an encryption of $C\rho_{\mathcal{C}}C^\dagger$. Decrypting this state, should result in the client's desired output. The fundamental property of such a scheme is that, throughout the protocol, the server learns nothing about the client's input state, $\rho_{\mathcal{C}}$ (or the output state). This should be true even if the server deviates from the client's instructions and behaves maliciously.

Formalizing this intuition, we have the following:

Definition 7.1 (Quantum Computing on Encrypted Data). A protocol for quantum computing on encrypted data is a tuple of QPT algorithms $\mathsf{QCED} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Evaluate}, \mathsf{Dec})$, where

$\mathsf{QCED.Setup}(C, 1^n) \to (\mathsf{sk} = (\mathsf{sk}_{in}, \mathsf{sk}_{comp}), \pi)$: takes as input a quantum circuit C, acting on n qubits, and outputs two classical keys $\mathsf{sk}_{in}, \mathsf{sk}_{comp} \in \{0,1\}^{\mathrm{poly}(n)}$, jointly denoted as sk, and a quantum state π, on $\mathrm{poly}(n)$ qubits.

$\mathsf{QCED.Enc}(\mathsf{sk}, \rho_{\mathcal{C}}) \to \mathsf{ct}$: takes as input a key sk and an n-qubit state $\rho_{\mathcal{C}}$ and outputs an n-qubit state, ct, representing an encrypted version of $\rho_{\mathcal{C}}$.

$\mathsf{QCED.Evaluate}(\mathcal{C}(C, \mathsf{sk}, \mathsf{ct}, \pi), \mathcal{S}(\rho_{\mathcal{S}})) \to (\mathsf{sk}^*, \mathsf{ct}^*, \tau_{\mathcal{S}})$: is an interactive protocol between a client, $\mathcal{C}$, and a server, $\mathcal{S}$. The client's input consists of the circuit C, a classical key sk, and two quantum states ct and π. The server's input is the quantum state[14] $\rho_{\mathcal{S}}$. At the end of the interaction, the client obtains an output key sk^* and the n-qubit quantum state ct^*. The server obtains the quantum state $\tau_{\mathcal{S}}$.

$\mathsf{QCED.Dec}(\mathsf{sk}^*, \mathsf{ct}^*) \to \tau_{\mathcal{C}}$: takes as input a key, sk^* and a state ct^* and either outputs an n-qubit state $\tau_{\mathcal{C}}$ or sets $\tau_{\mathcal{C}}$ to an abort symbol $|\bot\rangle\langle\bot|$.

We now define the following properties associated with a QCED protocol:

(i) *δ-correctness:* for every quantum circuit C and every n-qubit client input $\rho_{\mathcal{C}}$, provided the server follows the instructions of the interactive protocol, QCED.Evaluate, we have that

$$\|\mathsf{QCED.Dec}(\mathsf{sk}^*, \mathsf{ct}^*) - C\rho_{\mathcal{C}}C^\dagger\|_1 \le \delta(n)$$

where,

$$(\mathsf{sk}^*, \mathsf{ct}^*) \leftarrow \mathsf{QCED.Evaluate}\left(\mathcal{C}(C, \mathsf{sk}, \mathsf{ct}, \pi), \mathcal{S}\left(|0\rangle\langle0|^{\otimes \mathrm{poly}(n)}\right)\right)$$

$$\mathsf{ct} \leftarrow \mathsf{QCED.Enc}(\mathsf{sk}, \rho_{\mathcal{C}})$$

$$(\mathsf{sk}, \pi) \leftarrow \mathsf{QCED.Setup}(C, 1^n)$$

---

[14]While in what follows we take $\rho_{\mathcal{S}}$ to be $|0\rangle\langle0|^{\otimes \mathrm{poly}(n)}$, it should be noted that $\rho_{\mathcal{S}}$ can in principle be any state that depends only on n. It can be thought of as a *quantum advice* state of the prover (in the sense of [NY04]), so that the overall security of the protocol holds against non-uniform adversaries.

(ii) *ε-security:* for each quantum circuit C and each QPT server $\mathcal{S}$, there exists a QPT algorithm known as a simulator, denoted Sim, such that for every n-qubit client input $\rho_{\mathcal{C}}$ and every server input $\rho_{\mathcal{S}}$ it is the case that

$$\|\mathsf{Sim}(\rho_{\mathcal{S}}) - \mathsf{QCED.Evaluate}(\mathcal{C}(C,\mathsf{sk},\mathsf{ct},\pi),\mathcal{S}(\rho_{\mathcal{S}}))\|_1 \leq \varepsilon(n)$$

with

$$\mathsf{ct} \leftarrow \mathsf{QCED.Enc}(\mathsf{sk},\rho_{\mathcal{C}})$$

$$(\mathsf{sk},\pi) \leftarrow \mathsf{QCED.Setup}(C,1^n)$$

where Sim and QCED.Evaluate are CPTP maps of the form $\rho_{\mathcal{S}} \mapsto (\mathsf{ct}^*, \tau_{\mathcal{S}})$, mapping states from the server's input register to its output space (which comprises the quantum ciphertext returned to the client, ct^*, and any side information the server retains, $\tau_{\mathcal{S}}$)[15].

A number of QCED protocols have been proposed [AS06, BFK09, Fit17]. Most of these have as a primary objective to not only hide the input to the computation C, but to also hide the computation itself. For this reason, such protocols are referred to as *blind quantum computing* protocols. Of course, it should be clear that a QCED protocol, as defined above, also allows for hiding of the computation C. This is done by delegating to the server a universal circuit C_U and with the encrypted input consisting of not just the client's state $\rho_{\mathcal{C}}$ but also a description of the circuit C, to be applied on $\rho_{\mathcal{C}}$.

It should also be noted that the definition of a QCED can be easily satisfied by a protocol in which the client performs the quantum computation itself and never interacts with the server. Naturally, this requires the client to have the capability of performing universal quantum computations. Instead, the QCED protocols that have been proposed are such that the client only needs to prepare or measure single qubit states and otherwise has no quantum capabilities. Here we focus on one such protocol from [Bro15]. For a formal description of the protocol, we refer to [Bro15]. For our purposes, the important feature of this protocol that we use is that the client encrypts its input using a *quantum one-time pad* (QOTP) and the state π it generates is a tensor product of BB84 states. More formally:

Theorem 7.2 ([Bro15]). *There exists a (δ,ε)-QCED protocol, with $\delta(n) = \varepsilon(n) = 0$, for which:*

1. $\mathsf{QCED.Setup}(C,1^n) \rightarrow ((\mathsf{sk}_{in},\mathsf{sk}_{comp}),\pi)$ *with*

$$\mathsf{sk}_{in} = (\vec{a},\vec{b}), \ \ \vec{a},\vec{b} \xleftarrow{\$} \{0,1\}^n$$

$$\mathsf{sk}_{comp} = (\vec{v},\vec{\theta}), \ \ \vec{v},\vec{\theta} \xleftarrow{\$} \{0,1\}^{|C|}$$

$$\pi = \bigotimes_{i=1}^{|C|} H^{\theta_i}|v_i\rangle\langle v_i|H^{\theta_i}.$$

where $|C|$ denotes the size, or number of gates[16], in $|C|$.

2. $\mathsf{QCED.Enc}(((\vec{a},\vec{b}),\mathsf{sk}_{comp}),\rho_{\mathcal{C}}) \rightarrow \mathsf{ct}$ *with*

$$\mathsf{ct} = \sigma_X(\vec{a})\sigma_Z(\vec{b}) \ \rho_{\mathcal{C}} \ \sigma_Z(\vec{b})\sigma_X(\vec{a}).$$

---

[15]Note that the states ct^* and $\tau_{\mathcal{S}}$ can be entangled.

[16]Strictly speaking, in the protocol of [Bro15], the length of sk_{comp} and the number of qubits in π are equal to the number of T gates in the circuit C.

3. QCED.Dec$(\text{sk}^*, \text{ct}^*) \rightarrow \sigma_C$ with

$$\sigma_C = \sigma_X(\vec{a}^*)\sigma_Z(\vec{b}^*) \; \text{ct}^* \; \sigma_Z(\vec{b}^*)\sigma_X(\vec{a}^*),$$

whenever $\text{sk}^* = (((\vec{a}^*, \vec{b}^*), \text{sk}^*_{comp})$ *and otherwise the client sets* $\sigma_C = |\bot\rangle\langle\bot|$.

For the special case in which the client's input (the state ρ_C) is entirely classical, note that it needs to prepare only BB84 states and the quantum one-time pad becomes a classical one-time pad. This will be the case of interest for us when switching to a completely classical client in the next subsection.

The security achieved by schemes for QCED is known as *simulation security*, owing its name to the existence of the simulator, Sim, which reproduces the behavior of $\mathcal{S}$. This notion of security is equivalent to *composable security* [Can01, MR11], allowing for the protocol to be composed in sequence or in parallel with other protocols.

Let us also consider another version of security that is standard in the cryptography literature, namely that of indistinguishability under chosen-plaintext-attacks:

Definition 7.3 ((Adaptive) ε-IND-CPA-security of QCED)**.** Let QCED $=$ (Setup, Enc, Evaluate, Dec) be a protocol for quantum computing on encrypted data. For some quantum circuit C acting on n qubits, consider the following security experiment between a computationally unbounded adversary $\mathcal{A}$ and a referee, $\mathcal{R}$:

1. $\mathcal{A}$ generates two n-qubit quantum states ρ_0 and ρ_1. It sends these states to the referee.

2. The referee chooses $b \xleftarrow{\$} \{0,1\}$ and keeps ρ_b. It then performs QCED.Setup$(C, 1^n) \rightarrow (\text{sk}, \pi)$ and QCED.Enc$(\text{sk}, \rho_b) \rightarrow \text{ct}$.

3. The referee and $\mathcal{A}$ engage in the interactive protocol defined by QCED.Evaluate$(\mathcal{R}, \mathcal{A})$.

4. The adversary $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$.

It should be the case that

$$\Pr[b' = b] \leq \frac{1}{2} + \varepsilon(n) \tag{7.1}$$

where $\varepsilon(n)$ is referred to as the adversary's advantage in the security experiment.

Note that in the above definition the adversary is assumed to be computationally unbounded. When we transition to the setting of poly-time-bounded adversaries, we can simply modify this definition so as to consider only QPT adversaries $\mathcal{A}$. It should be clear that simulation security is the stronger notion of security, in the sense that simulation security implies IND-CPA-security, while the converse is not true. However, for the purpose of designing quantum protocols that use only classical communication, we will use IND-CPA-security, as in the previous schemes. The reason for this stems from the difficulties in achieving simulation security in the setting where the client and server interact classically, as explained in [GV19, BCC$^+$20]. However, we expect that with ideas from [GV19] and the additional assumption of a measurement buffer introduced in that paper, one could strengthen our security proof to satisfy the simulation based criterion.

7.1 Quantum computing on encrypted data with a classical client

We start by defining a QCED protocol that only involves classical communication between the client and the server (and where, additionally, the client is entirely classical).

Definition 7.4 (Quantum Computing on Encrypted Data with a Classical Client). A protocol for quantum computing on encrypted data with a classical client is a tuple of polynomial-time classical and quantum algorithms $\mathsf{QCED} = (\mathsf{Setup}, \mathsf{StatePrep}, \mathsf{Evaluate}, \mathsf{Dec})$, where

$\mathsf{QCED}_{\mathsf{CC}}.\mathsf{Setup}(C, 1^n) \rightarrow (\mathsf{sk}, \mathsf{pk})$: takes as input a quantum circuit C, acting on n qubits and with $|C| = \mathrm{poly}(n)$, and outputs two classical keys $\mathsf{sk}, \mathsf{pk} \in \{0,1\}^{\mathrm{poly}(n)}$, referred to as the secret and public key, respectively.

$\mathsf{QCED}_{\mathsf{CC}}.\mathsf{StatePrep}(\mathcal{C}(\mathsf{sk}, \mathsf{pk}), \mathcal{S}(\mathsf{pk})) \rightarrow (\tau, \sigma_{\mathcal{S}})$: is an interactive protocol between a PPT client, $\mathcal{C}$, and a QPT server $\mathcal{S}$. The client's input consists of a secret key sk and a public key pk. The server's input is the public key pk. At the end of the interaction both parties either obtain the classical transcript $\tau \in \{0,1\}^{\mathrm{poly}(n)}$ and the server also obtains the quantum state $\sigma_{\mathcal{S}}$ or the client aborts.

$\mathsf{QCED}_{\mathsf{CC}}.\mathsf{Evaluate}(\mathcal{C}(C, m_{\mathcal{C}}, \mathsf{sk}, \mathsf{pk}, \tau), \mathcal{S}(\mathsf{pk}, \tau, \sigma_{\mathcal{S}})) \rightarrow (\mathsf{sk}^*, \mathsf{ct}^*, \gamma_{\mathcal{S}})$: is an interactive protocol between a PPT client, $\mathcal{C}$, and a QPT server, $\mathcal{S}$. The client's input consists of the circuit C, an input for that circuit denoted as the string $m_{\mathcal{C}} \in \{0,1\}^n$, the secret key sk, the public key pk and the classical transcript τ. The server's input is the quantum state $\sigma_{\mathcal{S}}$ as well as the public key pk and the transcript τ. At the end of the interaction, the client obtains an output key $\mathsf{sk}^* \in \{0,1\}^{\mathrm{poly}(n)}$ and a classical ciphertext $\mathsf{ct}^* \in \{0,1\}^{\mathrm{poly}(n)}$. The server obtains the quantum state $\gamma_{\mathcal{S}}$.

$\mathsf{QCED}_{\mathsf{CC}}.\mathsf{Dec}(\mathsf{sk}^*, \mathsf{ct}^*) \rightarrow r$: takes as input a key, sk^* and ciphertext ct^* and outputs the string $r \in \{0,1\}^{\mathrm{poly}(n)}$. If the string ct^* is not of the appropriate size, the client aborts.

We now define the following properties associated with a $\mathsf{QCED}_{\mathsf{CC}}$ protocol:

(i) *δ-correctness:* for every quantum circuit C, with $|C| = \mathrm{poly}(n)$, and every input $m_{\mathcal{C}} \in \{0,1\}^n$, provided the server follows the instructions of the interactive protocol, $\mathsf{QCED}_{\mathsf{CC}}.\mathsf{Evaluate}$, we have that
$$\mathsf{TVD}[\mathsf{QCED}_{\mathsf{CC}}.\mathsf{Dec}(\mathsf{sk}^*, \mathsf{ct}^*), M(C|m_{\mathcal{C}}\rangle\langle m_{\mathcal{C}}|C^{\dagger})] \leq \delta(n)$$
where $M(\cdot)$ denotes the process of measuring a quantum state in the computational basis and where,

$$(\mathsf{sk}^*, \mathsf{ct}^*) \leftarrow \mathsf{QCED}_{\mathsf{CC}}.\mathsf{Evaluate}(\mathcal{C}(C, m_{\mathcal{C}}, \mathsf{sk}, \mathsf{pk}, \tau), \mathcal{S}(\mathsf{pk}, \tau, \sigma_{\mathcal{S}})),$$
$$(\tau, \sigma_{\mathcal{S}}) \leftarrow \mathsf{QCED}_{\mathsf{CC}}.\mathsf{StatePrep}(\mathcal{C}(\mathsf{sk}, \mathsf{pk}), \mathcal{S}(\mathsf{pk})),$$
$$(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{QCED}_{\mathsf{CC}}.\mathsf{Setup}(C, 1^n).$$

(ii) *(Adaptive) ε-IND-CPA security:* For a quantum circuit C acting on n qubits, consider the following security experiment between a QPT adversary $\mathcal{A}$ and a referee, $\mathcal{R}$:

1. $\mathcal{A}$ generates two n-bit strings m_0 and m_1. It sends these to the referee.

2. The referee chooses $b \xleftarrow{\$} \{0,1\}$ and keeps m_b. It then performs $\mathsf{QCED}_{\mathsf{CC}}.\mathsf{Setup}(C, 1^n, 1^\lambda) \rightarrow (\mathsf{sk}, \mathsf{pk})$.

3. The referee and the adversary engage in the protocol $\mathsf{QCED}_{\mathsf{CC}}.\mathsf{StatePrep}(\mathcal{R}(\mathsf{sk}, \mathsf{pk}), \mathcal{A}(\mathsf{pk})) \rightarrow (\tau, \sigma_{\mathcal{A}})$.

4. Provided the referee did not abort in step 3, the referee and $\mathcal{A}$ engage in the interactive protocol defined by $\mathsf{QCED}_{\mathsf{CC}}.\mathsf{Evaluate}(\mathcal{R}(C, m_b, \mathsf{sk}, \mathsf{pk}, \tau), \mathcal{A}(\mathsf{pk}, \tau, \sigma_{\mathcal{A}}))$.

5. The adversary $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$.

It should be the case that

$$|\Pr[b' = 0 \wedge \mathcal{R} \text{ did not abort}|b = 0] - \Pr[b' = 0 \wedge \mathcal{R} \text{ did not abort}|b = 1]| \leq \varepsilon(n) \qquad (7.2)$$

where $\varepsilon(n)$ is referred to as the adversary's advantage in the security experiment.

Note that unlike Definition 7.3, the security definition for $\mathsf{QCED_{CC}}$ includes a condition about whether the referee (or client) aborted in the protocol or not. This is because the state preparation subroutine, $\mathsf{QCED_{CC}.StatePrep}$, that is meant to mimic the sending of quantum states, can fail if the adversary (or server) behaves maliciously. As such, the security definition in this case states that the adversary cannot both convince the referee to accept in $\mathsf{QCED_{CC}.StatePrep}$ and learn something about the referee's chosen input.

We now turn our attention to constructing a protocol that satisfies Definition 7.4. At a high level, the idea is to start from the protocol from [Bro15] described in Theorem 7.2 for the special case of classical inputs m_C and replace the step of sending random BB84 states with our parallel RSP protocol (Protocol 3) We give a formal description of this protocol in Protocol 6. The following theorem asserts that Protocol 6 indeed satisfies the conditions in Definition 7.4.

Protocol 6. Protocol for Quantum Computing on Encrypted Data with a Classical Client

Let C be a circuit acting on n qubits. Additionally, take $\delta = 1/\mathrm{poly}(n)$ to be some error tolerance parameter. Define the following algorithms.

$\mathsf{QCED_{CC}.Setup}(C, 1^n) \rightarrow (\mathsf{sk}, \mathsf{pk})$: Run the procedure to sample keys and trapdoors $(k_1, t_{k_1}; \ldots k_M, t_{k_M})$, as in Protocol 1 and Protocol 2, $M = \mathrm{poly}(n)$ is the total amount of keys used, as determined by Protocol 3, with error tolerance δ. In addition, take $\mathsf{sk}_{in} \xleftarrow{\$} \{0,1\}^n$. Set $\mathsf{pk} = \{k_i\}_i$, $\mathsf{sk} = \{\{t_{k_i}\}_i, \mathsf{sk}_{in}\}$.

$\mathsf{QCED_{CC}.StatePrep}(\mathcal{C}(\mathsf{sk}, \mathsf{pk}), \mathcal{S}(\mathsf{pk})) \rightarrow (\tau, \sigma_\mathcal{S})$: The client runs Protocol 3 with security parameter $\lambda = \mathrm{poly}(n)$ with the server. Here, τ will denote the transcript of this protocol and $\sigma_\mathcal{S}$ will be the server's state upon completing the protocol (in the case where the client accepts).

$\mathsf{QCED_{CC}.Evaluate}(\mathcal{C}(C, m_\mathcal{C}, \mathsf{sk}, \mathsf{pk}, \tau), \mathcal{S}(\mathsf{pk}, \tau, \sigma_\mathcal{S})) \rightarrow (\mathsf{sk}^*, \mathsf{ct}^*, \gamma_\mathcal{S})$: The client and server perform $\mathsf{QCED.Evaluate}$ from Broadbent's protocol [Bro15], mentioned in the previous section. Specifically, the client uses sk and τ to determine the BB84 states the server should have (upon completing $\mathsf{QCED_{CC}.StatePrep}$). The key sk_{in} is used as a one-time pad for the input $m_\mathcal{C}$. The client then performs the classical interaction from Broadbent's protocol with the server, as if the BB84 states determined by sk and τ were sent via a quantum channel.

$\mathsf{QCED_{CC}.Dec}(\mathsf{sk}^*, \mathsf{ct}^*) \rightarrow r$: Same as the $\mathsf{QCED.Dec}$ operation in Broadbent's protocol, but restricted to classical outputs.

Theorem 7.5. *Protocol 6 is a protocol for Quantum Computing on Encrypted Data with a Classical Client having* $\mathrm{negl}(n)$ *correctness and* $1/\mathrm{poly}(n)$ *adaptive* IND-CPA *security.*

Proof. Correctness follows from the completeness of Protocol 3 (see Proposition 3.1) together with the correctness of Broadbent's QCED protocol.

To show IND-CPA security, let us consider the security experiment in Definition 7.4. The adversary (or the server) chooses two inputs, m_0 and m_1 and sends them to the referee (or the client). The referee picks a random bit $b \in \{0, 1\}$ and then selects m_b. This is followed by $\mathsf{QCED_{CC}}$.Setup and the interactive protocol $\mathsf{QCED_{CC}}$.StatePrep between the referee and the adversary. As follows from Corollary 4.35 we have that the shared state between the two is:

$$\sigma_{SWDYR}^{(\vec{\theta})} = \Pr[\top] |\top\rangle\langle\top|_S \otimes \sigma_{WDYR|\top}^{(\vec{\theta})} + (1 - \Pr[\top])|\bot\rangle\langle\bot|_S \otimes \sigma_{WDYR|\bot}^{(\vec{\theta})}$$

where we recall that S is the register denoting acceptance or rejection in the protocol, W is a register of the referee and DYR are registers of the adversary. As in the proof of Theorem 5.11, we can assume w.l.o.g. that conditioned on acceptance, the prover in fact has prepared the state $V\sigma_{WDYR|\top}^{(\vec{\theta})}V^\dagger$, not $\sigma_{WDYR|\top}^{(\vec{\theta})}$, where V is the isometry from Corollary 4.35. Slightly abusing notation, we denote this state as

$$\sigma_{WQB|\top}^{(\vec{\theta})} := V\sigma_{WDYR|\top}^{(\vec{\theta})}V^\dagger,$$

with $B := DYRA$ and $V : DYR \to QB$ as in Corollary 4.35. Then, Corollary 4.35 states that for any $\vec{\theta}$,

$$\Pr[\top]\,\sigma_{WQB|\top}^{(\vec{\theta})} \overset{c}{\approx}_{1/\operatorname{poly}(n)} \Pr[\top]\,\frac{1}{2^n} \sum_{\vec{v} \in \{0,1\}^n} |\vec{v}\rangle\langle\vec{v}|_W \otimes \left(\bigotimes_i H^{\theta_i}|v_i\rangle\langle v_i|H^{\theta_i}\right)_Q \otimes \alpha_B'$$

for some efficiently preparable state α_B'

Next, conditioned on having accepted in $\mathsf{QCED_{CC}}$.StatePrep, the referee $\mathcal{R}$ and the adversary $\mathcal{A}$ engage in the protocol $\mathsf{QCED_{CC}}$.Evaluate. As mentioned, this procedure is identical to the one in Broadbent's protocol, except for the fact that the referee, in this case, does not send BB84 states to the adversary, instead assuming those states were prepared via $\mathsf{QCED_{CC}}$.StatePrep. Upon completing this step, the adversary will output the bit b'. We are interested in the adversary's advantage in guessing the referee's chosen input. We proceed via an argument by contradiction. Suppose the adversary guesses the referee's input with advantage greater than $\varepsilon(n)$, i.e.

$$\left|\Pr\left[b' = 0 \wedge \mathcal{R} \text{ did not abort}|b = 0\right] - \Pr\left[b' = 0 \wedge \mathcal{R} \text{ did not abort}|b = 1\right]\right| > \varepsilon(n). \quad (7.3)$$

We will consider two cases. First, suppose the referee's probability of not aborting is less than $\varepsilon(n)$, or, using the notation from Corollary 4.35, $\Pr[\top] < \varepsilon(n)$. Then, using the fact that the referee aborting in $\mathsf{QCED_{CC}}$.StatePrep is independent of the choice of b, we immediately find a contradiction to Equation (7.3).

In what follows, we will assume $\Pr[\top] \geq \varepsilon(n)$ and take $\varepsilon(n) = 1/\operatorname{poly}(n)$. In this case, we get from Corollary 4.35 that

$$\sigma_{WQB|\top}^{(\vec{\theta})} \overset{c}{\approx}_{1/\operatorname{poly}(n)} \frac{1}{2^n} \sum_{\vec{v} \in \{0,1\}^n} |\vec{v}\rangle\langle\vec{v}|_W \otimes \left(\bigotimes_i H^{\theta_i}|v_i\rangle\langle v_i|H^{\theta_i}\right)_Q \otimes \alpha_B', \quad (7.4)$$

where α_B' is an efficiently preparable (normalised) quantum state.

We would now like to construct a new adversary, $\mathcal{A}'$, that breaks the security of Broadbent's QCED protocol. First, suppose $\mathcal{A}'$ interacts with the referee in the security experiment of Definition 7.3, up to step

3.[17] At this point, the adversary has received random BB84 states from the referee. After adding to this the (fixed and efficiently preparable) state α'_B, adversary $\mathcal{A}'$ has a state that is $1/\text{poly}(n)$ computationally indistinguishable from $\sigma^{(\vec{\theta})}_{WQB|\top}$. Now, $\mathcal{A}'$ runs the original adversary $\mathcal{A}$. We know that $\mathcal{A}$ achieves advantage greater than $\varepsilon(n)$ in the security experiment when run on the state $\sigma^{(\vec{\theta})}_{WQB|\top}$. Since $\mathcal{A}$ is computationally efficient, it follows that its distinguishing advantage can change by at most $1/\text{poly}(n)$ when run on the state prepared by $\mathcal{A}'$. By choosing appropriate parameters for the execution of Protocol 3, this $1/\text{poly}(n)$ error can be made sufficiently small, e.g. less than $\varepsilon(n)/2$. This immediately implies that $\mathcal{A}'$ obtains advantage greater than $\varepsilon(n)/2$ in the security experiment of Definition 7.3, which contradicts the adaptive IND-CPA security of Broadbent's QCED protocol. It follows that $\mathcal{A}$'s advantage must be upper bounded by $\varepsilon(n)$, concluding the proof. $\square$

8 Verifiable delegated blind quantum computation

Our final application of parallel RSP is verifiable delegated blind quantum computation (VDBQC). We start with an overview of VDBQC and then show how one can obtain such a protocol which uses only classical communication. Much like quantum computing on encrypted data, VDBQC protocols are interactive protocols between two parties that are here referred to as *verifier* and *prover*. The verifier delegates some quantum computation, in the form of a quantum circuit C, to a QPT prover. The "blindness" of VDBQC protocols refers to the fact that the verifier's input should remain hidden from the prover[18], just like in QCED protocols. In addition to this condition, it must also be the case that the verifier accepts with high probability when the output it obtains after interacting with the prover is the correct one (corresponding to having performed the computation C), and otherwise it rejects with high probability (this is called *verifiability*).

Formalizing this intuition, we have the following:

Definition 8.1 (Verifiable Delegated Blind Quantum Computing). A protocol for verifiable delegated blind quantum computing is a tuple of QPT algorithms VDBQC = (Setup, Enc, Evaluate, Dec), where

VDBQC.Setup$(C, 1^n) \to (\text{sk}, \pi)$: takes as input a quantum circuit C, acting on n qubits, and outputs a classical key $\text{sk} \in \{0,1\}^{\text{poly}(n)}$, and a quantum state π, on $\text{poly}(n)$ qubits.

VDBQC.Enc$(\text{sk}, \rho_\mathcal{V}) \to \text{ct}$: takes as input a key sk and an n-qubit state $\rho_\mathcal{V}$ and outputs an n-qubit state, ct, representing an encrypted version of $\rho_\mathcal{V}$.

VDBQC.Evaluate$(\mathcal{V}(C, \text{sk}, \text{ct}, \pi), \mathcal{P}(\rho_\mathcal{P})) \to (\text{sk}^*, \text{ct}^*, \tau_\mathcal{P})$: is an interactive protocol between a verifier, $\mathcal{V}$, and a prover, $\mathcal{P}$. The verifier's input consists of the circuit C, a classical key sk, and two quantum states ct and π. The prover's input is the quantum state $\rho_\mathcal{P}$. At the end of the interaction, the verifier obtains an output key sk^* and a quantum state ct^*. The prover obtains the quantum state $\tau_\mathcal{P}$.

VDBQC.Dec$(\text{sk}^*, \text{ct}^*) \to \tau_\mathcal{V}$: takes as input a key sk^* and an n-qubit state ct^* and outputs an $poly(n)$-qubit state $\tau_\mathcal{V}$ such that $\tau_\mathcal{V} = \sigma \otimes \phi$, and either $\text{ct}^* \leftarrow \text{VDBQC.Enc}(\text{sk}^*, \sigma)$ and $\phi = |\top\rangle\langle\top|$ or σ is an arbitrary state and $\phi = |\bot\rangle\langle\bot|$.

---

[17]We are assuming that possible inputs of that security experiment are the classical strings m_0 and m_1 in order to match the experiment where the two interact classically.

[18]As mentioned in Section 7, the original definition of blind quantum computing [AS06, BFK09] refers to hiding both the input to the circuit and the circuit itself from the prover. However, this is equivalent to keeping only the input hidden, as the verifier can always delegate some universal circuit C_U, which takes as input a description of a circuit C and an input string x.

We now define the following properties associated with a VDBQC protocol:

(i) *δ-correctness:* for every quantum circuit C and every n-qubit verifier input $\rho_{\mathcal{V}}$, provided the prover follows the instructions of the interactive protocol, VDBQC.Evaluate, we have that

$$\|\text{VDBQC.Dec}(\text{sk}^*, \text{ct}^*) - C\rho_{\mathcal{V}}C^\dagger \otimes |\top\rangle\langle\top|\|_1 \le \delta(n),$$

where,

$$(\text{sk}^*, \text{ct}^*) \leftarrow \text{VDBQC.Evaluate}\left(\mathcal{V}(C, \text{sk}, \text{ct}, \pi), \mathcal{P}\left(|0\rangle\langle0|^{\otimes \text{poly}(n)}\right)\right)$$
$$\text{ct} \leftarrow \text{VDBQC.Enc}(\text{sk}, \rho_{\mathcal{V}})$$
$$(\text{sk}, \pi) \leftarrow \text{VDBQC.Setup}(C, 1^n)$$

(ii) *ε-blind-verifiability:* for each quantum circuit C and each QPT prover $\mathcal{P}$, there exists a QPT algorithm known as a simulator, denoted Sim, such that for every n-qubit verifier input $\rho_{\mathcal{V}}$ and every prover input $\rho_{\mathcal{P}}$ it is the case that

$$\|\text{Sim}(\rho_{\mathcal{P}}) - \text{VDBQC.Evaluate}(\mathcal{V}(C, \text{sk}, \text{ct}, \pi), \mathcal{P}(\rho_{\mathcal{P}}))\|_1 \le \varepsilon(n)$$

and there exist $0 \le p \le 1$ and a state ρ_{inc} on $\text{poly}(n)$ qubits such that

$$\|\tau_{\mathcal{V}} - \rho_{ver}\|_1 \le \varepsilon(n)$$

with

$$\rho_{ver} = pC\rho_{\mathcal{V}}C^\dagger \otimes |\top\rangle\langle\top| + (1-p)\rho_{inc} \otimes |\bot\rangle\langle\bot|,$$
$$\tau_{\mathcal{V}} \leftarrow \text{VDBQC.Dec}(\text{sk}^*, \text{ct}^*)$$
$$(\text{sk}^*, \text{ct}^*, \tau_{\mathcal{P}}) \leftarrow \text{VDBQC.Evaluate}\left(\mathcal{V}(C, \text{sk}, \text{ct}, \pi), \mathcal{P}(\rho_{\mathcal{P}})\right)$$
$$\text{ct} \leftarrow \text{VDBQC.Enc}(\text{sk}, \rho_{\mathcal{V}})$$
$$(\text{sk}, \pi) \leftarrow \text{QCED.Setup}(C, 1^n)$$

where Sim and VDBQC.Evaluate are CPTP maps of the form $\rho_{\mathcal{P}} \mapsto (\text{ct}^*, \tau_{\mathcal{P}})$, mapping states from the prover's input register to its output space (which comprises the quantum ciphertext returned to the verifier, ct^*, and any side information the prover retains, $\tau_{\mathcal{P}}$).

There are a number of protocols that satisfy the above definition and we refer the reader to [GKK19] for an overview. We should also note that the above definition does not apply to all VDBQC protocols in the literature. In a bit more detail, using terminology from [GKK19], the above definition generally refers to single-prover prepare-and-send protocols,[19] where the verifier sends quantum states to the prover. This is precisely the setting we are interested in. Specifically, we would like a VDBQC protocol in which the verifier has to prepare and send only BB84 states to the prover. As far as we are aware, there is no protocol that achieves this directly. However, it can be achieved indirectly using a protocol of Morimae [Mor18]. To explain, we need to give a high-level description of that protocol. We start by first explaining the *history state construction* [KSVV02, KKR06].

---

[19]The definition also applies for single-prover receive-and-measure protocols [GKK19].

8.1 History state construction and post-hoc verification

The history state construction, introduced in [KSVV02, KKR06], was used to show that the *local Hamiltonian problem* is complete for the complexity class QMA (the quantum analogue of NP). For our purposes, we forego the definition of QMA and instead explain how the history state construction can be used to verify general quantum computations.

Given a circuit $C = U_T U_{T-1}...U_1$ acting on n qubits and an input $x \in \{0,1\}^n$, where $T = |C|$ and U_i denotes the i'th gate in C (and with $U_0 = \text{id}$), consider the state

$$|\psi_{C,x}\rangle = \frac{1}{\sqrt{T+1}} \sum_{t=0}^{T} |t\rangle U_t U_{t-1}...U_0 |x\rangle. \tag{8.1}$$

This is referred to as the history state of C on input x. Importantly, $|\psi_{C,x}\rangle$ is the ground state of a k-local Hamiltonian denoted $H_{C,x}$, with $k = O(1)$. Whenever C accepts x (i.e. measuring the first qubit of $C|x\rangle$ yields outcome 1 with probability greater than 2/3) the state $|\psi_{C,x}\rangle$ has low energy with respect to $H_{C,x}$ otherwise it has high energy. More precisely, the gap between accepting and rejecting instances (referred to as the *promise gap*) is at least $1/\text{poly}(n)$. It should also be noted that $|\psi_{C,x}\rangle$ can be prepared efficiently, with a circuit of size $\text{poly}(|C|, |x|)$ applied to a $|00...0\rangle$ state.

This yields a simple protocol for verifying quantum computations: a quantum prover is instructed to prepare $|\psi_{C,x}\rangle$ and send the state to the verifier, who will then measure it with one of the local terms of $H_{C,x}$. The local term will be chosen with a probability proportional to its "weight" in $H_{C,x}$. In other words, it should be the case that the expectation of the verifier's measurement is $\langle\psi_{C,x}|H_{C,x}|\psi_{C,x}\rangle$. This provides an estimate for the state's energy which can be used to decide acceptance or rejection. Thus, if C accepts x, the history state (when measured according to $H_{C,x}$) will have low energy. In contrast, whenever C rejects x *all* states will have high energy with respect to $H_{C,x}$ so that a malicious prover cannot convince the verifier to accept. It should also be noted that one can construct local Hamiltonians that are 2-local and for which the non-trivial components of the Hamiltonian terms are either Pauli σ_X or σ_Z [BL08].

The protocol we sketched above is referred to as *post-hoc* verification of quantum computation and was introduced in [MF16, FHM18]. We can summarise this result with the following theorem.

Theorem 8.2 ([FHM18]). *Let C be a T-gate quantum circuit acting on n qubits such that $C = U_T U_{T-1}...U_1$ where $T = |C|$ and U_i denotes the i'th gate in C (and where we take $U_0 = \text{id}$). Suppose it is the case that for any input $x \in \{0,1\}^n$ either $|\langle 0|C|x\rangle|^2 \geq 2/3$ or $|\langle 0|C|x\rangle|^2 \leq 1/3$. Then, there exists a collection of n-qubit observables $\{O_i\}_{i\leq m}$, with $m = \text{poly}(n)$, of the form $O_i = \bigotimes_{j=1}^{n} P_j$, with $P_j \in \{\sigma_X, \sigma_Z, \text{id}\}$ (and with at most two P_j's not equal to id), a probability distribution $p : [m] \to [0,1]$, and values $\alpha, \beta > 0$ with $\beta - \alpha > 1/\text{poly}(n)$ (referred to as the* promise gap*), such that*

- *if $|\langle 0|C|x\rangle|^2 \geq 2/3$ then $\sum_{i=1}^{m} p(i)\langle\psi_{C,x}|O_i|\psi_{C,x}\rangle \leq \alpha$,*

- *if $|\langle 0|C|x\rangle|^2 \leq 1/3$ then for all states $|\psi'\rangle$, $\sum_{i=1}^{m} p(i)\langle\psi'|O_i|\psi'\rangle \geq \beta$,*

where

$$|\psi_{C,x}\rangle = \frac{1}{\sqrt{T+1}} \sum_{t=0}^{T} |t\rangle U_t U_{t-1}...U_0 |x\rangle. \tag{8.2}$$

Here the observables O_i are the local terms of the Hamiltonian $H_{C,x}$, whose groundstate is the history state $|\psi_{C,x}\rangle$. The probability distribution $p(i)$ is simply the weight of each local term in the Hamiltonian, suitably normalised. In the post-hoc protocol, the verifier samples an observable according to that distribution and measures the state received from the prover with that observable.

Remark 8.3 (Instance independence). An important observation about the post-hoc verification protocol, first made in [ACGH20], is that the sampling of an observable O_i and the measurement performed by the verifier can be made independent of each other. This is done as follows. The verifier will measure each qubit of the state received from the prover uniformly at random with either the σ_X or σ_Z observable and then sample an observable O_i according to the distribution p. Assuming the underlying Hamiltonian is 2-local, let $1 \leq j, k \leq n$ denote the indices of the qubits acted upon by non-trivial operators in O_i (i.e. operators not equal to id). If these operators match the observables the verifier used when measuring qubits i and j (which happens with probability $1/4$), the verifier keeps the measurement outcome, discarding it otherwise (and rejecting). As shown in [ACGH20], the effect of this is to reduce the promise gap by a constant factor of $1/4$, so that overall the promise gap is still $1/\operatorname{poly}(n)$.

8.2 Morimae's verification protocol

The post-hoc protocol described in the previous subsection is a simple protocol for verifying general quantum computations. Note, however, that it is not a VDBQC protocol according to our definition. This is because the protocol is not blind—the prover is told both the computation C and the input x so as to prepare the history state.

While the post-hoc protocol does not satisfy the blindness property, Morimae recently proposed a modified version of the protocol that is blind [Mor18]. This main idea of Morimae's protocol is to instruct the prover to prepare the history state using a QCED protocol.[20] In fact, the verifier delegates to the prover the preparation of the state $|\psi_{C,x}\rangle$ as well as $|\psi_{\bar{C},x}\rangle$, where $\bar{C}$ is the same as C except the output qubit is negated (that is, an additional σ_X gate is added to the output qubit of C). As the states are prepared via a QCED scheme, they will be encrypted and the prover learns nothing about the input x or the circuit C. The prover then sends these states to the verifier who proceeds to decrypt and measure them using the observables from Theorem 8.2. In this way, the verifier is able certify the outcome of the computation C on input x while also keeping C and x hidden from the prover. We can also see that if the QCED scheme that is used in the first part of the protocol is Broadbent's scheme [Bro15], mentioned in the previous section, the verifier will only be required to send BB84 states to the prover. Moreover, in that scheme, the encryption of the states is done via a quantum one-time pad. As this consists of the application of Pauli σ_X and σ_Z operations, these can be absorbed into the observables that the verifier measures, so that the overall protocol requires the verifier to send BB84 states and measure qubits in the standard and Hadamard bases (or with the observables σ_Z, σ_X, respectively).

Our goal is to dequantise this protocol. As we can see, however, this requires one to not only dequantise the sending of quantum states, but also the process of performing σ_X and σ_Z measurements. For this, we make use of the measurement subprotocol from Mahadev's verification protocol [Mah18]. This is described in the next subsection.

8.3 Mahadev's measurement protocol

In [Mah18], Mahadev introduced the first protocol allowing a classical verifier to delegate and verify arbitrary quantum computations to a quantum prover. At the core of that protocol is a primitive for classically instructing a quantum prover to perform measurements in the standard and Hadamard bases. Importantly,

---

[20]Strictly speaking, the preparation should be done with a blind quantum computing protocol, i.e. a protocol that explicitly hides the computation from the prover. However, as mentioned previously, a QCED protocol also yields blind quantum computation by having the QCED computation be a universal circuit, U, with the property that that $U|C, x\rangle = C|x\rangle \otimes |00...0\rangle$, for some quantum circuit C and input x.

this is done in a "verifiable way", so that if the prover does not perform the correct measurements, the verifier can detect this and abort. Combined with the idea of post-hoc verification, we can see how this yields a protocol for verifying general quantum computations: simply instruct the prover to measure observables from Theorem 8.2 on the history state of the computation.

For our purposes, we are only interested in the measurement subprotocol. For a detailed description of that protocol, we refer the reader to [Mah18]. At a high level, the protocol works in a similar way to our RSP protocol (which was directly inspired by Mahadev's result): the verifier uses an ENTCF family (based on the quantum intractability of LWE) to delegate the chosen measurement to the prover. As a result of interacting with the prover, the verifier obtains a purported measurement outcome and can decide whether to accept or reject. This decision is performed using the ENTCF family's trapdoor, which allows the verifier to check whether the prover performed the measurements as instructed or not. The main result of this protocol is summarised in the following two theorems, taken from [Mah18, Sections 5.4, 7], that capture the completeness and soundness of the protocol:

Theorem 8.4 (Completeness of measurement protocol [Mah18]). *For all n-qubit states ρ and all basis choices $h \in \{0,1\}^n$ (where $h_i = 0$ denotes measurement of qubit i in the standard basis and $h_i = 1$ denotes measurement in the Hadamard basis), an honest QPT prover, $\mathcal{P}$, is accepted by the verifier in the measurement protocol with probability $1 - \mathrm{negl}(n)$. Additionally, it is the case that*

$$\mathrm{TVD}[D_{\mathcal{P},h}, D_{\rho,h}] \leq \mathrm{negl}(n), \tag{8.3}$$

where $D_{\mathcal{P},h}$ is the probability distribution resulting from the verifier having delegated the measurement h to $\mathcal{P}$ and having accepted in the protocol; $D_{\rho,h}$ is the distribution resulting from measuring the state ρ in the bases determined by h.

Theorem 8.5 (Soundness of measurement protocol [Mah18]). *Assuming the quantum intractability of LWE, for a QPT prover $\mathcal{P}$, let $1 - p$ be the probability that the verifier accepts the prover in the measurement protocol.[21] Then, there exists an n-qubit state ρ and a prover $\mathcal{P}'$ such that for all basis choices h,*

$$\mathrm{TVD}[D_{\mathcal{P},h}, D_{\mathcal{P}',h}] \leq p + \sqrt{p} + \mathrm{negl}(n), \tag{8.4}$$

and with $D_{\mathcal{P}',h}$ being computationally indistinguishable from $D_{\rho,h}$.

8.4 Verifiable delegated blind quantum computation with a classical verifier

Let us start by defining VDBQC with a classical verifier. Throughout this section we are considering VDBQC protocols for decision problems, i.e. problems where the output is a single classical bit. In other words, for some computation C and input x, we are only interested in the result of measuring the first qubit of $C|x\rangle$ in the standard basis.

Definition 8.6 (Verifiable Delegated Blind Quantum Computing with a Classical Verifier). A protocol for verifiable delegated blind quantum computing (of decision problems) with a classical verifier is a tuple of QPT algorithms $\mathsf{VDBQC}_{\mathsf{CV}} = (\mathsf{StatePrep}, \mathsf{Evaluate}, \mathsf{Dec})$, where

---

[21]Strictly speaking, the measurement protocol has two types of rounds (a test round and a Hadamard round) and one should consider the probabilities of acceptance in each of these rounds. See [Mah18, Section 7] for more details. Here we take p to be the maximum of the two probabilities.

$\mathsf{VDBQC_{CV}}.\mathsf{StatePrep}(\mathcal{V}(C,1^n),\mathcal{P}(\rho_\mathcal{P})) \rightarrow (\mathsf{sk},\mathsf{pk},\tau,\sigma_\mathcal{P})$: is an interactive protocol between a PPT verifier, $\mathcal{V}$, and a QPT prover $\mathcal{P}$. The verifier's input consists of a description of a circuit C acting on n qubits. The prover has as input some $\mathrm{poly}(n)$-qubit quantum state $\rho_\mathcal{P}$. The protocol either aborts or succeeds[22]. If the protocol succeeds the verifier obtains as an output a secret key $\mathsf{sk} \in \{0,1\}^{\mathrm{poly}(n)}$ and the prover obtains a $\mathrm{poly}(n)$-qubit quantum state $\sigma_\mathcal{P}$. Additionally, they both obtain a public key $\mathsf{pk} \in \{0,1\}^{\mathrm{poly}(n)}$ and a classical transcript $\tau \in \{0,1\}^{\mathrm{poly}(n)}$.

$\mathsf{VDBQC_{CV}}.\mathsf{Evaluate}(\mathcal{V}(C,m_\mathcal{V},\mathsf{sk},\mathsf{pk},\tau),\mathcal{P}(\mathsf{pk},\tau,\sigma_\mathcal{P})) \rightarrow (\mathsf{sk}^*,\mathsf{ct}^*,\gamma_\mathcal{P})$: is an interactive protocol between a PPT verifier, $\mathcal{V}$, and a QPT prover, $\mathcal{P}$. The verifier's input consists of the circuit C, an input for that circuit denoted as the string $m_\mathcal{V} \in \{0,1\}^n$, the secret key sk, the public key pk and the classical transcript τ. The prover's input is the quantum state $\sigma_\mathcal{P}$ as well as the public key pk and the transcript τ. At the end of the interaction, the verifier obtains an output key $\mathsf{sk}^* \in \{0,1\}^{\mathrm{poly}(n)}$ and a classical ciphertext $\mathsf{ct}^* \in \{0,1\}^{\mathrm{poly}(n)}$. The prover obtains the quantum state $\gamma_\mathcal{P}$.

$\mathsf{VDBQC_{CV}}.\mathsf{Dec}(\mathsf{sk}^*,\mathsf{ct}^*) \rightarrow r$: takes as input a key, sk^*, and a ciphertext ct^* and either outputs the bit $r \in \{0,1\}$ or the protocol aborts.

We now define the following properties associated with a $\mathsf{VDBQC_{CV}}$ protocol:

(i) *δ-correctness:* for every quantum circuit C, with $|C| = \mathrm{poly}(n)$, and every input $m_\mathcal{V} \in \{0,1\}^n$, provided the prover follows the instructions of the interactive protocol, $\mathsf{VDBQC_{CV}}.\mathsf{Evaluate}$, we have that
$$\Pr[r = M(C|m_\mathcal{V}\rangle\langle m_\mathcal{V}|C^\dagger) \wedge \top] \geq 1 - \delta(n)$$
where $M(\cdot)$ denotes the process of measuring the first qubit of a quantum state in the standard basis and outputting that result, $\bot$ denotes the event of the protocol aborting and where,

$$r \leftarrow \mathsf{VDBQC_{CV}}.\mathsf{Dec}(\mathsf{sk}^*,\mathsf{ct}^*)$$
$$(\mathsf{sk}^*,\mathsf{ct}^*,\gamma_\mathcal{P}) \leftarrow \mathsf{VDBQC_{CV}}.\mathsf{Evaluate}(\mathcal{V}(C,m_\mathcal{V},\mathsf{sk},\mathsf{pk},\tau),\mathcal{P}(\mathsf{pk},\tau,\sigma_\mathcal{P}))$$
$$(\mathsf{sk},\mathsf{pk},\tau,\sigma_\mathcal{P}) \leftarrow \mathsf{VDBQC_{CV}}.\mathsf{StatePrep}(\mathcal{V}(C,1^n),\mathcal{P}(|0\rangle\langle 0|^{\otimes\mathrm{poly}(n)}))$$

(ii) *ε-blind-verifiability:* the protocol satisfies the ε-IND-CPA security condition of Definition 7.4 and in addition, for each quantum circuit C, each verifier input $m_\mathcal{V}$, each QPT prover $\mathcal{P}$ and each state $\rho_\mathcal{P}$ it is the case that
$$\Pr[r \neq M(C|m_\mathcal{V}\rangle\langle m_\mathcal{V}|C^\dagger) \wedge \top] \leq \varepsilon(n)$$
where $M(\cdot)$ denotes the process of measuring the first qubit of a quantum state in the standard basis and outputting that result, $\top$ denotes the event of the protocol *not aborting* and where,

$$r \leftarrow \mathsf{VDBQC_{CV}}.\mathsf{Dec}(\mathsf{sk}^*,\mathsf{ct}^*)$$
$$(\mathsf{sk}^*,\mathsf{ct}^*,\gamma_\mathcal{P}) \leftarrow \mathsf{VDBQC_{CV}}.\mathsf{Evaluate}(\mathcal{V}(C,m_\mathcal{V},\mathsf{sk},\mathsf{pk},\tau),\mathcal{P}(\mathsf{pk},\tau,\sigma_\mathcal{P}))$$
$$(\mathsf{sk},\mathsf{pk},\tau,\sigma_\mathcal{P}) \leftarrow \mathsf{VDBQC_{CV}}.\mathsf{StatePrep}(\mathcal{V}(C,1^n),\mathcal{P}(\rho_\mathcal{P}))$$

---

[22] Here we are assuming that there is a global abort flag for the entire protocol.

To obtain a protocol satisfying the above definition, for some computation C on input x, we follow the ideas in Morimae's protocol and have the verifier first perform a $\mathsf{QCED_{CC}}$ protocol with the prover, instructing it to prepare the states $|\psi_{C,x}\rangle$ and $|\psi_{\bar{C},x}\rangle$, as described in Section 8.2. The verifier must then instruct the prover to perform the measurement of local terms from the Hamiltonian $H_{C,x}$. We do this by making use of Mahadev's measurement protocol from Section 8.3. An important point to note here is that because the states $|\psi_{C,x}\rangle$ and $|\psi_{\bar{C},x}\rangle$ were prepared via the $\mathsf{VDBQC_{CV}}$ protocol, they will be encrypted and so it's not a priori clear whether the subsequent measurement is useful for the verifier. But this will indeed be the case, as the encryption of the states is given a by a quantum one-time pad (that is known to the verifier) which can be absorbed into the measurements by suitably flipping measurement outcomes. The protocol is described more formally in Protocol 7 and the following theorem shows that it satisfies Definition 8.6

Protocol 7. Protocol for Verifiable Delegated Blind Quantum Computation with a Classical Verifier

Let C be a circuit acting on n qubits. Additionally, take $\delta = 1/\operatorname{poly}(n)$ to be some error tolerance parameter. Define the following algorithms.

$\mathsf{VDBQC_{CV}.StatePrep}(\mathcal{V}(C,1^n),\mathcal{P}(\rho_{\mathcal{P}})) \to (\mathsf{sk},\mathsf{pk},\tau,\sigma_{\mathcal{P}})$: The verifier runs Protocol 3 with the prover, taking the security parameter $\lambda = \operatorname{poly}(n)$. Here, pk and sk denote the public and secret keys generated in that protocol, τ denotes the transcript of the protocol and $\sigma_{\mathcal{P}}$ is the prover's state upon completing the protocol (in the case where the verifier accepts).

$\mathsf{VDBQC_{CV}.Evaluate}(\mathcal{V}(C,m_{\mathcal{V}},\mathsf{sk},\mathsf{pk},\tau),\mathcal{P}(\mathsf{pk},\tau,\sigma_{\mathcal{P}})) \to (\mathsf{sk}^*,\mathsf{ct}^*,\gamma_{\mathcal{P}})$: The verifier first performs the $\mathsf{QCED_{CC}.Evaluate}(\mathcal{V}(C',m_{\mathcal{V}},\{\mathsf{sk},\mathsf{sk}_{in}\},\mathsf{pk},\tau),\mathcal{P}(\mathsf{pk},\tau,\sigma_{\mathcal{P}}))$ subprotocol, where the circuit C' is one that prepares the history states $|\psi_{C,m_{\mathcal{V}}}\rangle$ and $|\psi_{\bar{C},m_{\mathcal{V}}}\rangle$ and where $\mathsf{sk}_{in} \xleftarrow{\$} \{0,1\}^{\operatorname{poly}(n)}$ (see Protocol 6 for details). It then performs Mahadev's measurement protocol, instructing the prover to measure each qubit of the (encrypted) history states in either the standard basis or the Hadamard basis, chosen uniformly at random. The string ct^* is set to be the transcript of that protocol. The verifier also samples a local term from each of the Hamiltonians $H_{C,m_{\mathcal{V}}}$ and $H_{\bar{C},m_{\mathcal{V}}}$ with probability proportional to the weight of that term (as in the post-hoc protocol mentioned in Section 8.1). The string sk^* is set to the choices of these local terms, together with the strings sk and sk_{in}.

$\mathsf{VDBQC_{CV}.Dec}(\mathsf{sk}^*,\mathsf{ct}^*) \to r$: The verifier uses sk^* to decrypt the results in ct^*, corresponding to the measurements of local terms of the Hamiltonians $H_{C,m_{\mathcal{V}}}$ and $H_{\bar{C},m_{\mathcal{V}}}$ on the states $|\psi_{C,m_{\mathcal{V}}}\rangle$ and $|\psi_{\bar{C},m_{\mathcal{V}}}\rangle$[23]. If the tests of the Mahadev's measurement protocol failed, the verifier aborts. Otherwise, if, according to the measurements, $|\psi_{C,m_{\mathcal{V}}}\rangle$ has low energy with respect to $H_{C,m_{\mathcal{V}}}$ and $|\psi_{\bar{C},m_{\mathcal{V}}}\rangle$ has high energy with respect to $H_{\bar{C},m_{\mathcal{V}}}$[24], it sets $r=0$ and accepts. If $|\psi_{C,m_{\mathcal{V}}}\rangle$ has high energy with respect to $H_{C,m_{\mathcal{V}}}$ and $|\psi_{\bar{C},m_{\mathcal{V}}}\rangle$ has low energy with respect to $H_{\bar{C},m_{\mathcal{V}}}$, it sets $r=1$ and accepts. Otherwise, it aborts.

Theorem 8.7. *Protocol 7 is a protocol for Verifiable Delegated Blind Quantum Computation with a Classical Verifier having* $\operatorname{negl}(n)$ *correctness and* $1/\operatorname{poly}(n)$ *blind-verifiability.*

Proof. We start by showing the protocol's correctness. The prover will first engage in the $\mathsf{QCED_{CC}}$ protocol with the verifier, resulting in the preparation of BB84 states in its memory, followed by the application of the unitary for preparing the (encrypted) history states $|\psi_{C,m_\mathcal{V}}\rangle$ and $|\psi_{\bar{C},m_\mathcal{V}}\rangle$. As explained in Section 7, these states are encrypted via a quantum one-time pad known to the verifier. The verifier then uses Mahadev's measurement protocol to instruct the prover to measure the states according to randomly chosen terms from the Hamiltonians $H_{C,x}$ and $H_{\bar{C},x}$, respectively (using the instance-independent idea from Remark 8.3). Again, as the prover is assumed to behave correctly, this will result in the correct measurements being performed, as follows from Theorem 8.4. Recall that the terms of the Hamiltonian comprise only Pauli σ_X and σ_Z operators (and so the measurements performed via Mahadev's protocol will also be a measurements of σ_X or σ_Z observables). Since the history states have a quantum one-time pad applied, we can "absorb" the pad into the measurement by suitably flipping the measurement outcomes. More precisely, if a qubit has a σ_X (σ_Z, respectively) applied in the one-time pad and the qubit is measured with the σ_Z (σ_X, respectively) observable, then the outcome should be flipped. Otherwise, the outcome is not flipped. Once the verifier has decrypted the measurement outcomes in this way, it will then test the energy condition to estimate the energies of the two states relative to the Hamiltonians $H_{C,x}$ and $H_{\bar{C},x}$. The correctness of the history state construction (Theorem 8.2) ensures that the verifier obtains the correct outcome with high probability[25].

We would now like to show that the protocol also satisfies $1/\operatorname{poly}(n)$ blind-verifiability. The blindness condition, which is identical to that of $\mathsf{QCED_{CC}}$, is clearly satisfied up to the point where the verifier performs Mahadev's measurement protocol. This is because until that point, the protocol is simply an application of $\mathsf{QCED_{CC}}$ for a specific computation (that of preparing the two history states). Using the instance-independent version of the history state construction (see Remark 8.3), Mahadev's measurement protocol also satisfies the IND-CPA condition. This is due to the fact that the instructions the verifier sends to the prover in the measurement protocol are independent of the input.

Verifiability follows from the soundness of the measurement protocol (Theorem 8.5) together with the properties of the history state construction (Theorem 8.2). The proof is identical to the proof of Theorem 8.6 from [Mah18], which we will not restate here. It follows that

$$\left|\Pr[r \neq M(C|m_\mathcal{V}\rangle\langle m_\mathcal{V}|C^\dagger) \wedge \top] - 1/\operatorname{poly}(n)\right| \leq \operatorname{negl}(n),$$

which then implies that

$$\Pr[r \neq M(C|m_\mathcal{V}\rangle\langle m_\mathcal{V}|C^\dagger) \wedge \top] \leq 1/\operatorname{poly}(n) + \operatorname{negl}(n).$$

This yields the desired condition of $1/\operatorname{poly}(n)$ verifiability that, when combined with the blindness condition (i.e. the IND-CPA security) yields $1/\operatorname{poly}(n)$ blind-verifiability, concluding the proof. $\qquad\square$

References

[Aar09]   Scott Aaronson. Quantum copy-protection and quantum money. *2009 24th Annual IEEE Conference on Computational Complexity*, Jul 2009.

[ABF⁺16]   Gorjan Alagic, Anne Broadbent, Bill Fefferman, Tommaso Gagliardoni, Christian Schaffner, and Michael St. Jules. Computational security of quantum encryption. *Information Theoretic Security*, page 47–71, 2016.

---

[25]We are assuming here that the verifier and prover are performing the parallel-repeated version of these protocols, as described in [ACGH20], hence why correctness is negligible.

[ACGH20] Gorjan Alagic, Andrew M Childs, Alex B Grilo, and Shih-Han Hung. Non-interactive classical verification of quantum computation. In *Theory of Cryptography Conference*, pages 153–180. Springer, 2020.

[ALL$^+$21] Scott Aaronson, Jiahui Liu, Qipeng Liu, Mark Zhandry, and Ruizhe Zhang. New approaches for quantum copy-protection. In *Annual International Cryptology Conference*, pages 526–555. Springer, 2021.

[AS06] Pablo Arrighi and Louis Salvail. Blind quantum computation. *International Journal of Quantum Information*, 4(05):883–898, 2006.

[BB84] C. H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*, pages 8, vol. 175, 1984.

[BCC$^+$20] Christian Badertscher, Alexandru Cojocaru, Léo Colisson, Elham Kashefi, Dominik Leichtle, Atul Mantri, and Petros Wallden. Security limitations of classical-client delegated quantum computing. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 667–696. Springer, 2020.

[BCKM21] James Bartusek, Andrea Coladangelo, Dakshita Khurana, and Fermi Ma. One-way functions imply secure computation in a quantum world. In *Annual International Cryptology Conference*, pages 467–496. Springer, 2021.

[BCM$^+$18] Z. Brakerski, P. Christiano, U. Mahadev, U. Vazirani, and T. Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. *IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 320–331, 2018.

[BFK09] Anne Broadbent, Joseph Fitzsimons, and Elham Kashefi. Universal blind quantum computation. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 517–526. IEEE, 2009.

[BL08] Jacob D Biamonte and Peter J Love. Realizable hamiltonians for universal adiabatic quantum computers. *Physical Review A*, 78(1):012352, 2008.

[BL19] Anne Broadbent and Sébastien Lord. Uncloneable quantum encryption via random oracles. *IACR Cryptol. ePrint Arch.*, page 257, 2019.

[Bro15] Anne Broadbent. Delegating private quantum computations. *Canadian Journal of Physics*, 93(9):941–946, 2015.

[BY21] Anne Broadbent and Peter Yuen. Device-independent oblivious transfer from the bounded-quantum-storage-model and computational assumptions. *arXiv preprint arXiv:2111.08595*, 2021.

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 136–145. IEEE, 2001.

[CCKW19] Alexandru Cojocaru, Léo Colisson, Elham Kashefi, and Petros Wallden. QFactory: Classically-instructed remote secret qubits preparation. *Advances in Cryptology - ASI-ACRYPT 2019, Lecture Notes in Computer Science, Springer*, pages 615–645, 2019.

[CGJV19] Andrea Coladangelo, Alex B Grilo, Stacey Jeffery, and Thomas Vidick. Verifier-on-a-leash: new schemes for verifiable delegated quantum computation, with quasilinear resources. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 247–277. Springer, 2019.

[CKVW10] Ran Canetti, Yael Tauman Kalai, Mayank Varia, and Daniel Wichs. On symmetric encryption and point obfuscation. In *Theory of Cryptography Conference*, pages 52–71. Springer, 2010.

[CMP20] Andrea Coladangelo, Christian Majenz, and Alexander Poremba. Quantum copy-protection of compute-and-compare programs in the quantum random oracle model. *arXiv preprint arXiv:2009.13865*, 2020.

[CN16] Matthew Coudron and Anand Natarajan. The parallel-repeated magic square game is rigid. *arXiv preprint arXiv:1609.06306*, 2016.

[Col16] Andrea W Coladangelo. Parallel self-testing of (tilted) EPR pairs via copies of (tilted) CHSH. *arXiv preprint arXiv:1609.03687*, 2016.

[DH76] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[FHM18] Joseph F Fitzsimons, Michal Hajdušek, and Tomoyuki Morimae. Post hoc verification of quantum computation. *Physical review letters*, 120(4):040501, 2018.

[Fit17] Joseph F Fitzsimons. Private quantum computation: an introduction to blind quantum computing and related protocols. *npj Quantum Information*, 3(1):1–11, 2017.

[FWZ22] Honghao Fu, Daochen Wang, and Qi Zhao. Computational self-testing of multi-qubit states and measurements, 2022. Manuscript forthcoming.

[GH17] William Timothy Gowers and Omid Hatami. Inverse and stability theorems for approximate representations of finite groups. *Sbornik: Mathematics*, 208(12):1784, 2017.

[GKK19] Alexandru Gheorghiu, Theodoros Kapourniotis, and Elham Kashefi. Verification of quantum computation: An overview of existing approaches. *Theory of computing systems*, 63(4):715–808, 2019.

[GLSV21] Alex B Grilo, Huijia Lin, Fang Song, and Vinod Vaikuntanathan. Oblivious transfer is in miniqcrypt. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 531–561. Springer, 2021.

[Got02] Daniel Gottesman. Uncloneable encryption. *arXiv preprint quant-ph/0210062*, 2002.

[GV19] Alexandru Gheorghiu and Thomas Vidick. Computationally-secure and composable remote state preparation. *IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1024–1033, 2019.

[HMNY21]   Taiga Hiroka, Tomoyuki Morimae, Ryo Nishimaki, and Takashi Yamakawa. Quantum encryption with certified deletion, revisited: Public key, attribute-based, and classical communication. *arXiv preprint arXiv:2105.05393*, 2021.

[IR89]   Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 44–61, 1989.

[KKR06]   Julia Kempe, Alexei Kitaev, and Oded Regev. The complexity of the local hamiltonian problem. *Siam journal on computing*, 35(5):1070–1097, 2006.

[KNY21]   Fuyuki Kitagawa, Ryo Nishimaki, and Takashi Yamakawa. Secure software leasing from standard assumptions. In *Theory of Cryptography Conference*, pages 31–61. Springer, 2021.

[KSVV02]   Alexei Yu Kitaev, Alexander Shen, Mikhail N Vyalyi, and Mikhail N Vyalyi. *Classical and quantum computation*. Number 47. American Mathematical Soc., 2002.

[Mah18]   Urmila Mahadev. Classical verification of quantum computations. *IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 259–267, 2018.

[MDCAF21]   Tony Metger, Yfke Dulek, Andrea Coladangelo, and Rotem Arnon-Friedman. Device-independent quantum key distribution from computational assumptions. *New Journal of Physics*, 23(12):123021, 2021.

[MF16]   Tomoyuki Morimae and Joseph F Fitzsimons. Post hoc verification with a single prover. *arXiv preprint arXiv:1603.06046*, 2016.

[Mor18]   Tomoyuki Morimae. Blind quantum computing can always be made verifiable. *arXiv preprint arXiv:1803.06624*, 2018.

[MR11]   Ueli Maurer and Renato Renner. Abstract cryptography. In *In Innovations in Computer Science*. Citeseer, 2011.

[MTH^{+}21]   Akihiro Mizutani, Yuki Takeuchi, Ryo Hiromasa, Yusuke Aikawa, and Seiichiro Tani. Computational self-testing for entangled magic states. *arXiv preprint arXiv:2111.02700*, 2021.

[MV20]   Tony Metger and Thomas Vidick. Self-testing of a single quantum device under computational assumptions. *arXiv preprint arXiv:2001.09161*, 2020.

[MYS12]   M McKague, T H Yang, and V Scarani. Robust self-testing of the singlet. *Journal of Physics A: Mathematical and Theoretical*, 45(45):455304, oct 2012.

[NR97]   Moni Naor and Omer Reingold. On the construction of pseudo-random permutations: Luby-Rackoff revisited (extended abstract). In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, STOC '97, page 189–199, New York, NY, USA, 1997. Association for Computing Machinery.

[NV17]   Anand Natarajan and Thomas Vidick. A quantum linearity test for robustly verifying entanglement. *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 1003–1015, 2017.

[NV18]   Anand Natarajan and Thomas Vidick. Low-degree testing for quantum states, and a quantum entangled games PCP for QMA. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 731–742. IEEE, 2018.

[NY04]   Harumichi Nishimura and Tomoyuki Yamakami. Polynomial time quantum computation with advice. *Information Processing Letters*, 90(4):195–204, 2004.

[Reg09]   Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), 2009.

[RS19]   Roy Radian and Sattath. Semi-quantum money. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, AFT '19, page 132–146, New York, NY, USA, 2019. Association for Computing Machinery.

[Rud91]   Steven Rudich. The use of interaction in public cryptosystems. In *Annual International Cryptology Conference*, pages 242–251. Springer, 1991.

[RUV13]   Ben W Reichardt, Falk Unger, and Umesh Vazirani. Classical command of quantum systems. *Nature*, 496(7446):456–460, 2013.

[TFKW13]   Marco Tomamichel, Serge Fehr, Jedrzej Kaniewski, and Stephanie Wehner. A monogamy-of-entanglement game with applications to device-independent quantum cryptography. *New Journal of Physics*, 15(10):103002, Oct 2013.

[Vid11]   Thomas Vidick. *The complexity of entangled games*. PhD thesis, UC Berkeley, 2011.

[Vid20]   Thomas Vidick. Course FSMP, Fall'20: Interactions with quantum devices. http://users.cms.caltech.edu/~vidick/teaching/fsmp/fsmp.pdf, 2020.

[VZ21]   Thomas Vidick and Tina Zhang. Classical proofs of quantum knowledge. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 630–660. Springer, 2021.

[Wie83]   Stephen Wiesner. Conjugate coding. *SIGACT News*, 15(1):78–88, jan 1983.

[Win99]   A. Winter. Coding theorem and strong converse for quantum channels. *IEEE Transactions on Information Theory*, 45(7):2481–2485, 1999.

[WZ82]   William K Wootters and Wojciech H Zurek. A single quantum cannot be cloned. *Nature*, 299(5886):802–803, 1982.