Obfuscation of Pseudo-Deterministic Quantum Circuits

James Bartusek[*⋆], Fuyuki Kitagawa[‡], Ryo Nishimaki[‡], Takashi Yamakawa[‡]

[⋆]UC Berkeley, USA
bartusek.james@gmail.com
[‡]NTT Social Informatics Laboratories, Tokyo, Japan
{fuyuki.kitagawa.yh,ryo.nishimaki.zk,takashi.yamakawa.ga}@hco.ntt.co.jp

Abstract

We show how to obfuscate pseudo-deterministic quantum circuits in the classical oracle model, assuming the quantum hardness of learning with errors. Given the classical description of a quantum circuit Q, our obfuscator outputs a quantum state $|\widetilde{Q}\rangle$ that can be used to evaluate Q repeatedly on arbitrary inputs.

Instantiating the classical oracle using any candidate post-quantum indistinguishability obfuscator gives us the first candidate construction of indistinguishability obfuscation for all polynomial-size pseudo-deterministic quantum circuits. In particular, our scheme is the first candidate obfuscator for a class of circuits that is powerful enough to implement Shor's algorithm (SICOMP 1997).

Our approach follows Bartusek and Malavolta (ITCS 2022), who obfuscate *null* quantum circuits by obfuscating the verifier of an appropriate classical verification of quantum computation (CVQC) scheme. We go beyond null circuits by constructing a publicly-verifiable CVQC scheme for quantum *partitioning* circuits, which can be used to verify the evaluation procedure of Mahadev's quantum fully-homomorphic encryption scheme (FOCS 2018). We achieve this by upgrading the one-time secure scheme of Bartusek (TCC 2021) to a fully reusable scheme, via a publicly-decodable *Pauli functional commitment*, which we formally define and construct in this work. This commitment scheme, which satisfies a notion of binding against committers that can access the receiver's standard and Hadamard basis decoding functionalities, is constructed by building on techniques of Amos, Georgiou, Kiayias, and Zhandry (STOC 2020) introduced in the context of equivocal but collision-resistant hash functions.

[*]Part of this work was done while visiting NTT Social Informatics Laboratories for an internship.

Contents

1 Introduction

A program obfuscator is a "one-way compiler" that renders code unintelligible without harming its functionality. This concept dates back to the beginning of modern cryptography [DH76], and has since attracted much interest as a tool for protecting software against reverse-engineering, intellectual property theft, and piracy. While the theoretical foundations of program obfuscation were laid in 2001 [BGI$^+$12],[1] it was not until 2013 [GGH$^+$16][2] that researchers developed a proposal for obfuscating general-purpose (classical) computation. This first candidate sparked a massive research effort that has both established program obfuscation as a "central hub" [SW21] of cryptography with countless applications, and has resulted in obfuscation schemes based on well-founded cryptographic assumptions [JLS21].

Meanwhile, the concepts of quantum information and quantum computation have had a profound impact on computer science, with stunning applications such as unconditionally secure key agreement [BB84] and efficient integer factorization [Sho97a], not to mention the promise of major advances in chemistry and physics. As the field of quantum information science matures, researchers have investigated fundamental questions pertaining to information privacy and information integrity. This has resulted in a remarkable series of feasibility results for securing quantum information and computation, e.g. encryption [AMTDW00], authentication [BCG$^+$02], zero-knowledge [BJSW20], secure multi-party computation [CGS02, DNS10, DGH$^+$20], and delegation of computation [Chi05, ABOEM18, BFK09, RUV13, Mah18, Mah22]. However, despite these efforts, the feasibility of *quantum obfuscation* has remained elusive, and the following question has remained largely open.

Is it possible to obfuscate quantum computation?

Prior research has focused its efforts on definitional work [AF16], impossibility results [AF16, AL21, ABDS21], and limited classes of quantum computation [AJJ14, BK21, BM22]. The best feasibility results we had prior to this work were for obfuscating quantum circuits with logarithmically many non-Clifford gates [BK21] and for obfuscating "null" quantum circuits that always output zero [BM22]. Neither of these classes comes close to a notion of "general-purpose" quantum computation, and thus the feasibility of quantum obfuscation as a tool for quantum software protection has remained wide open.

Results. We consider the class of pseudo-deterministic quantum circuits, which are quantum circuits that take a classical input and produce a fixed classical output for each input with overwhelming probability. Essentially, these circuits compute a classical truth table, and can decide any language in (non-promise) BQP. This class captures Shor's algorithm [Sho97a], which is arguably the quintessential algorithm for demonstrating the power of quantum computation over classical computation.

Our main result is the following. In the *classical oracle model*, the evaluator (and adversary) are given oracle access to an efficiently computable classical functionality prepared by the obfuscator.

Theorem 1.1. *Assuming the quantum hardness of Learning with Errors (QLWE), there exists a VBB obfuscator for any polynomial-size pseudo-deterministic quantum circuit Q in the classical oracle model, where the obfuscated program is a quantum state $|\widetilde{Q}\rangle$.*

[1]The preliminary version appeared in CRYPTO 2001.
[2]The preliminary version appeared in FOCS 2013.

On the classical oracle model. The classical oracle model idealizes the notion of obfuscation for classical circuits, much like the random oracle model [BR95] idealizes a cryptographic hash function and the generic group model [Sho97b] idealizes a cryptographic group. Such idealized primitives are typically not realizable in the real world, and a classical oracle is no exception. Indeed, virtual black-box (VBB) obfuscation comes for free in the classical oracle model, and it is known [BGI$^+$12] that there exist (contrived) examples of circuits that provably cannot be VBB obfuscated (even with quantum information [AL21, ABDS21]). However, despite these contrived counterexamples, there is by now a fairly long history of establishing the feasibility of novel quantum-cryptographic primitives in the classical oracle model [AC12, BS16, AGKZ20, ALL$^+$21, BM22], and our result fits into this line of work.

Moreover, [BGI$^+$12] also defined a weaker notion of obfuscation called *indistinguishability obfuscation*, which only requires that the obfuscations of two functionally equivalent programs are computationally indistinguishable, and which was subsequently shown by [SW21] and many follow-up works to be extremely powerful. Our main result is a construction of obfuscation for pseudo-deterministic quantum circuits from obfuscation of classical circuits (plus QLWE), and in order to prove security, we treat the classical obfuscation as implementing a black-box. However, one can interpret this result as *heuristic* evidence that our construction gives indistinguishability obfuscation for pseudo-deterministic quantum circuits when the classical obfuscator is instantiated with a candidate post-quantum indistinguishability obfuscation scheme [BGMZ18, CVW18, BDGM22, GP21, WW21, DQV$^+$21].

One can also appreciate our result in an oraclized world. Here, we show that, assuming QLWE,[3] it is possible to simulate access to a "BQP oracle" with just a P oracle. More precisely, the BQP oracle we implement can decide *languages* in BQP, as opposed to more general promise problems.[4]

Building blocks. To obtain our main result of quantum obfuscation, we construct the following intermediate primitives that may be of independent interest.

Publicly-decodable Pauli functional commitments.

We formally define the notion of a *Pauli functional commitment*, which has appeared implicitly in many recent works, e.g. [BCM$^+$21, Mah22, Vid20]. These are bit commitment schemes that, when used in superposition to commit to a qubit, support opening the qubit to a measurement in either the standard or the Hadamard basis. While the only prior construction [BCM$^+$21, Mah22] of such commitments supports publicly-decodable standard basis measurements, security is completely compromised if the committer obtains access to the receiver's *Hadamard* basis decoding functionality.

In this work, we describe a novel construction of Pauli functional commitments where security holds even if the committer obtains access to *both* the receiver's standard and Hadamard basis decoding functionalities, and we argue security in the classical oracle model. Our construction is inspired by and unifies two lines of work: privately-decodable Pauli functional commitments

[3]In fact, it may be possible to remove the QLWE assumption entirely from our construction by showing that quantum fully-homomorphic encryption and dual-mode randomized trapdoor claw-free hash functions can be built from classical VBB obfuscation. We leave an exploration of this to future work.

[4]This is because circuits for deciding promise problems are technically not pseudo-deterministic. There exist inputs that are neither yes or no instances, and thus are not guaranteed to produce a pseudo-deterministic output.

[BCM+21, Mah22], and collision-resistant but equivocal hash functions [ARU14, AGKZ20].

Publicly-verifiable quantum fully-homomorphic encryption.

Given the recent progress in constructing quantum homomorphic encryption schemes [BJ15, DSS18, Mah18], a natural question is whether the homomorphic evaluation procedure for these schemes can be *verified* and with what resources.

The first work to address this question was [ADSS17], who showed how to make the scheme of [DSS18] verifiable. Unfortunately, their verifier requires secret parameters, including the decryption key of the homomorphic encryption scheme. In a recent work, [Bar21] showed how to obtain verifiable quantum FHE based on the scheme of [Mah18], with the following properties. The verifier is completely classical and doesn't require the decryption key of the FHE scheme, though it does require additional secret verification parameters.

In this work, we obtain the first feasibility result for *publicly-verifiable* quantum fully-homomorphic encryption. Our protocol supports the classical verification of pseudo-deterministic quantum computation over the underlying plaintexts, and is proven sound in the classical oracle model. We also remark that the public parameters for our scheme are *quantum* (while the verification is classical), and it is an interesting question for future work to see whether these public parameters (and, in turn, our obfuscated program) can be made completely classical.

Applications. Program obfuscation has direct applications to software protection, and our results indicate that such protections may be possible to achieve in the context of quantum software. Obfuscated programs intuitively cannot be reverse-engineered, meaning that we can now protect any intellectual property or other secret information contained in the implementation of the quantum program.

In the classical and post-quantum settings, obfuscation has also been identified as a useful tool for digital watermarking [BGI+12, CHN+18, KN22], which allows for embedding an unremovable "mark" into a program, and acts as a deterrent against software piracy. Quantum information potentially allows for much *stronger* forms of protection against piracy, enabling computation to be encoded into a quantum state that provably cannot be copied [Aar09]. However, the scope of such "copy-protection" schemes has so far been limited to classical functionalities [CMP20, ALL+21, CLLZ21, AK21, AKL+22, KN23]. In Section 6.2, we sketch how our obfuscation scheme results in a candidate for copy-protection of (unlearnable) *quantum* programs, following the construction of [ALL+21].

Another common application of obfuscation in the classical setting is to advanced forms of encryption, such as functional encryption [GGH+16]. In Section 6.3, we sketch an application of our construction to functional encryption for *quantum functionalities*.

That being said, we stress that the main focus of our work is on the construction of quantum obfuscation, and we leave a more in-depth exploration of applications to future work.

Open problems. Our work raises many interesting questions on the topic of quantum obfuscation. One immediate question is whether it is possible to obfuscate *all* quantum circuits with classical input and output, extending our result for pseudo-deterministic circuits. That is, can circuits that output an arbitrary distribution over classical strings be obfuscated? As explained in Section 2, we follow the approach of [BM22] who consider obfuscating the verifier of an appropriate classical verification of computation protocol [Mah22]. Unfortunately, it is not known how to classically verify general quantum sampling circuits, at least with negligible soundness (the work of [CLLW22] provides a solution with weaker soundness). This appears to be one barrier for extending our approach to all quantum circuits with classical input and output.

One can also wonder about the possibility of obfuscating general quantum operations over quantum registers. That is, while our scheme is able to obfuscate quantum computation, it still only implements a "classical" language, albeit one whose truth table may only be (known to be) computable with a quantum circuit. Thus, this leaves open the feasibility (or impossibility) of implementing *quantum* oracles, and we consider this to be a very interesting question to understand in future work.

Finally, we mention two natural open questions regarding our construction itself. First, is it possible to remove the quantum states from our construction and obtain a *classical* obfuscated program? Next, can we improve on the heuristic nature of our security proof, and obtain indistinguishability obfuscation for pseudo-deterministic quantum circuits from the assumption of indistinguishability obfuscation for classical circuits?

2 Technical Overview

In this overview, we will describe how to obfuscate any pseudo-deterministic quantum circuit Q, where pseudo-deterministic means that for each input x there exists an output y such that $\Pr[Q(x) \to y] = 1 - \text{negl}$. That is, we describe a compiler that given the classical description of Q, produces an obfuscated program $|\widetilde{Q}\rangle$ that reveals as little as possible about the description of Q while preserving the functionality of Q. Throughout this overview, we will treat such circuits as fully deterministic, associating a well-defined bit $y := Q(x)$ to each input x, which has a negligible effect on our arguments.

2.1 Our approach: Verifying quantum partitioning circuits

Fully-homomorphic encryption. A natural approach to obfuscation involves the notion of *fully-homomorphic encryption* (FHE), which allows for encoding data x into a ciphertext $\mathsf{Enc}(x)$ so that anyone holding $\mathsf{Enc}(x)$ and a function f can produce a ciphertext $\mathsf{Enc}(f(x))$. Indeed, given an FHE scheme that supports the evaluation of *quantum functionalities* [Mah18], one could release an encryption $\mathsf{Enc}(Q)$ of the description of Q. Then, any evaluator with an input x can obtain $\mathsf{Enc}(Q(x))$ by running an appropriate evaluation procedure.

This comes close to a working obfuscation scheme, except that the evaluator obtains $\mathsf{Enc}(Q(x))$ rather than the output $Q(x)$ in the clear. To fix this, we cannot simply release the FHE secret key sk, allowing the evaluator to decrypt $\mathsf{Enc}(Q(x))$ and learn $Q(x)$, because this would *also* allow the evaluator to decrypt $\mathsf{Enc}(Q)$ and learn the description of Q. Instead, we could release a carefully "broken" secret key that *only* allows decryption of ciphertexts $\mathsf{Enc}(Q(x))$ that encrypt an honestly evaluated output $Q(x)$.

Reducing to classical obfuscation. But how can we obtain such a carefully broken key? One attempt would be to release an obfuscation of the following program C, which has the secret key sk and the ciphertext $\mathsf{Enc}(Q)$ hard-coded,

$$C[\mathsf{sk}, \mathsf{Enc}(Q)](x, \mathsf{ct}) : \quad \text{if } \mathsf{Eval}[\mathsf{Enc}(Q)](x) \to \mathsf{ct}, \text{ output } \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}), \text{ and otherwise output } \perp,$$

where $\mathsf{Eval}[\mathsf{Enc}(Q)](\cdot)$ is the FHE evaluation circuit that on input x outputs $\mathsf{Enc}(Q(x))$. However, we don't know how to obfuscate C since $\mathsf{Eval}[\mathsf{Enc}(Q)](\cdot)$ is a quantum circuit.

Instead, building on observations by [BM22], we could hope to construct an argument system with a *classical* verifier V that satisfies the following properties.

- For any x, one can compute a ciphertext ct and a proof π such that $V(\mathsf{Enc}(Q), x, \mathsf{ct}, \pi) = 1$.
- It is hard to find (x, ct, π) such that $V(\mathsf{Enc}(Q), x, \mathsf{ct}, \pi) = 1$ and $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \neq Q(x)$.

If such a system existed, we could instead obfuscate the following *classical* program

$$\widetilde{C}[\mathsf{sk}, \mathsf{Enc}(Q)](x, \mathsf{ct}, \pi): \text{ if } V(\mathsf{Enc}(Q), x, \mathsf{ct}, \pi) \to 1, \text{ output } \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}), \text{ and otherwise output } \bot.$$

Crucially, this approach follows the "verify-then-decrypt" paradigm, where the output ciphertext is *first* verified to be honest, and only then decrypted using sk. A procedure that first decrypts and then verifies may not be secure since the adversary could submit dishonest ciphertexts to learn information about sk.

Classical verification of quantum computation and its limitations. Thus, it suffices to construct a classically-verifiable argument system for the class of quantum circuits $\mathsf{Eval}[\mathsf{Enc}(Q)]$ that take

$$\mathsf{Eval}[\mathsf{Enc}(Q)](x) \to \mathsf{Enc}(Q(x)),$$

where Enc is a quantum fully-homomorphic encryption (QFHE) scheme and Q is a deterministic quantum circuit.

As mentioned earlier, [Mah22] did construct a protocol for classical verification of quantum computation. Unfortunately, there are two major problems with using [Mah22]'s scheme for this application.

- **Sampling circuits.** [Mah22]'s scheme only supports verification of (pseudo)-deterministic quantum circuits. However, the evaluation procedure of known QFHE schemes [Mah18, Bra18] is inherently *randomized*, even if the underlying computation is deterministic, meaning that the circuit that we would like to verify actually produces a *sample* $\mathsf{Enc}(Q(x))$ from a classical distribution over ciphertexts.[5]
- **Public verifiability.** Note that the evaluator will have (obfuscated) access to the verification function, which means that it can repeatedly query the verifier with proofs of its choice. If soundness holds even when verification is *public*, then the evaluator cannot break soundness using access to this oracle. However, [Mah22]'s scheme is privately-verifiable, and can be broken given repeated access to the verifier.

Towards solving the first problem, [CLLW22] presented a scheme for classical verification of sampling circuits, though only with inverse polynomial soundness error. While interesting on its own, this renders the scheme difficult to use for our application, since a polynomial-time evaluator can eventually break soundness and thus break security of the obfuscation scheme. It appears that improving upon their result to obtain negligible soundness for classical verification of quantum sampling circuits is difficult, and could be considered a major open problem.

[5]While this distribution is only supported on ciphertexts that encrypt the correct output bit $Q(x)$, the random coins used for the output ciphertext will vary.

Quantum partitioning circuits. Instead, we relax our goal. We observe that if Q is deterministic, then we don't need the full power of verification of sampling circuits to verify the sampling of $\mathsf{Enc}(Q(x))$. Indeed, we can *partition* the output space of $\mathsf{Eval}[\mathsf{Enc}(Q)](\cdot)$ into ciphertexts ct_0 that decrypt to 0 and ciphertexts ct_1 that decrypt to 1. Thus, each input x outputs a sample from one of these two sets. That is, we can define a classical predicate $P \coloneqq \mathsf{Dec}(\mathsf{sk}, \cdot)$ such that $P(\mathsf{Eval}[\mathsf{Enc}(Q)](\cdot))$ is (pseudo)-deterministic.

Thus, we say that Q is a *quantum partitioning circuit* if there exists a predicate P such that $P(Q(\cdot))$ is pseudo-deterministic, and we investigate the feasibility of obtaining a classically-verifiable argument system for such partitioning circuits. Crucially for our application, the prover in the argument system cannot depend on P since P will contain the description of the FHE secret key.[6] Then, we will need an argument system with (roughly) the following syntax (see Section 5.1 for a formal description).

- $\mathsf{Gen}(1^\lambda, Q) \to \mathsf{pp}$: The parameter generation algorithm outputs public parameters pp. We allow pp to contain the description of a *classical oracle*, and refer to such a protocol as being *in the oracle model*.
- $\mathsf{Prove}(\mathsf{pp}, Q, x) \to \pi$: The prover algorithm outputs a proof π.
- $\mathsf{Ver}(\mathsf{pp}, Q, x, \pi) \to q \cup \{\bot\}$: The verifier checks if the proof is valid, and if so outputs a classical string q.
- $\mathsf{Out}(q, P) \to b$: The output algorithm takes q and the description of a predicate P and outputs a bit b.

For soundness, we require that no computationally bounded prover can produce an (x, π) such that $\mathsf{Ver}(\mathsf{pp}, Q, x, \pi) \to q$ and $\mathsf{Out}(q, P) \neq P(Q(x))$. We refer to such a protocol as a *non-interactive publicly-verifiable classical verification of quantum partitioning circuits.* In Section 6, we follow the intuition given above, and show formally how to use this type of argument system along with QFHE and VBB obfuscation of classical circuits (which is used to obfuscate the classical oracle in pp) to obfuscate pseudo-deterministic quantum circuits.

In the remainder of this overview, we will describe how to construct non-interactive publicly-verifiable classical verification of quantum partitioning circuits in the oracle model.

2.2 Prior work: One-time soundness

Building on [Mah22, CLLW22], the prior work of [Bar21] shows how to construct non-interactive *privately-verifiable* classical verification of quantum partitioning circuits,[7] where soundness breaks down if the prover is given oracle access to the verification functionality. We refer to this security as "one-time soundness". We will eventually build on top of this protocol in two steps.[8]

1. We will first show how to obtain reusable soundness against provers that can access the verification oracle in a limited "single instance" setting. In this setting, there is only one input x that the verification oracle will accept.

[6] And otherwise, this notion would trivially reduce to classical verification of pseudo-deterministic quantum circuits.

[7] In [Bar21], quantum partitioning circuits were referred to as "quantum-classical" circuits.

[8] Breaking this into two steps is only for the purpose of the overview. In Section 5.4, we perform both steps simultaneously.

2. We will upgrade this protocol to the fully reusable setting, thus obtaining a *publicly-verifiable* protocol in the oracle model.

In this section, we describe the protocol of [Bar21] in some detail, as our construction will use these internal details. However, before getting into the protocol, we describe a useful abstraction that is novel to this work: a *Pauli functional commitment*. We will then describe [Bar21]'s protocol using the language of Pauli functional commitments, and, later in the overview, show how a new variation on the notion of Pauli functional commitments will be integral to our final construction.

Pauli functional commitments. Bit commitment schemes traditionally satisfy a notion of binding and a notion of hiding. A *functional* commitment scheme includes an additional notion of functionality, which allows the committer to open its commitment to some *function* of the committed message, up to some limitations imposed by the binding property.

A Pauli functional commitment (PFC) is a traditional (non-interactive) classical bit commitment scheme augmented with a particular quantum functionality property. Note that any classical bit commitment algorithm $\mathsf{Com}(\mathsf{ck}, b) \to (b, u, c)$, where ck is the commitment key, u is opening information, and c is the commitment string, can be used to commit to a qubit $\alpha_0 |0\rangle + \alpha_1 |1\rangle$ in superposition. If the commitment scheme is perfectly hiding, then measuring a commitment string c would leave a remaining state of the form $\alpha_0 |0\rangle |u_0\rangle + \alpha_1 |1\rangle |u_1\rangle$,[9] which preserves the original qubit. A Pauli functional commitment enables the committer to then "open" its state to *either* a standard basis measurement *or* a Hadamard basis measurement of its original qubit $\alpha_0 |0\rangle + \alpha_1 |1\rangle$. More formally, it should satisfy the following syntax.

- $\mathsf{Gen}(1^\lambda) \to (\mathsf{ck}, \mathsf{dk})$: Gen outputs a commitment key ck and a decoding key dk.[10]
- $\mathsf{Com}(\mathsf{ck}, \mathcal{B}) \to (\mathcal{B}, \mathcal{U}, c)$: Com takes as input a single-qubit register $\mathcal{B}$ and produces a classical commitment c along with registers $(\mathcal{B}, \mathcal{U})$, where $\mathcal{U}$ holds opening information.[11]
- $\mathsf{OpenZ}(\mathcal{B}, \mathcal{U}) \to u$: The standard basis opening algorithm performs a measurement on registers $(\mathcal{B}, \mathcal{U})$ to produce a classical string u.
- $\mathsf{OpenX}(\mathcal{B}, \mathcal{U}) \to u$: The Hadamard basis opening algorithm performs a measurement on registers $(\mathcal{B}, \mathcal{U})$ to produce a classical string u.
- $\mathsf{DecZ}(\mathsf{dk}, c, u) \to \{0, 1, \bot\}$: The standard basis decoding algorithm takes the decoding key dk, a commitment c, an opening u, and either decodes a bit 0 or 1, or outputs $\bot$.
- $\mathsf{DecX}(\mathsf{dk}, c, u) \to \{0, 1, \bot\}$: The Hadamard basis decoding algorithm takes the decoding key dk, a commitment c, an opening u, and either decodes a bit 0 or 1, or outputs $\bot$.

A Pauli functional commitment should satisfy *functionality* as described above and some notion (depending on the application) of *binding* to a classical bit. That is, binding is defined with respect to the bit output by the DecZ algorithm. A notion of hiding does not need to be explicitly

[9]Note that depending on the commitment scheme, the second register may contain a superposition over random coins / opening information.

[10]For now, assume ck is classical, though later we will consider commitments with quantum commitment keys.

[11]Whenever we say that an algorithm takes as input or outputs a register, we mean that it operates on a quantum state stored on that register.

considered - the properties of functionality and binding are already enough to make this primitive both non-trivial and useful.

We note that this notion has appeared implicitly in many previous works, e.g. [BCM$^+$21, Mah22, Vid20]. Indeed, [BCM$^+$21, Mah22] essentially showed how to construct a Pauli functional commitment that simultaneously satisfies *two* binding properties from the quantum hardness of learning with errors (QLWE). In our own words, these properties are the following.

- **Dual-mode.** $\mathsf{Gen}(1^\lambda, h)$ now takes as input a bit h indicating the "mode", where $h = 1$ is the regular mode, and $h = 0$ is a *perfectly binding* mode. In perfectly binding mode, for every commitment c there is at most one bit b such that there exists an opening u with $\mathsf{DecZ}(\mathsf{dk}, c, u) = b$. This mode allows for the definition of an algorithm $\mathsf{Invert}(\mathsf{dk}, c) \to b$ that outputs the bit b such that there exists u with $\mathsf{DecZ}(\mathsf{dk}, c, u) = b$ (or outputs $\perp$ if such a b does not exist). Importantly, the ck output on $h = 0$ vs $h = 1$ must be computationally indistinguishable.

- **Uncertainty.** For any polynomial-time adversary that outputs (c, b, u_Z, u_X), it holds that

$$\begin{aligned}&\Pr[\mathsf{DecZ}(\mathsf{dk}, c, u_Z) = b \ \wedge \ \mathsf{DecX}(\mathsf{dk}, c, u_X) = 0] \\ &\approx \Pr[\mathsf{DecZ}(\mathsf{dk}, c, u_Z) = b \wedge \mathsf{DecX}(\mathsf{dk}, c, u_X) = 1].\end{aligned}$$

  That is, if an adversary opens successfully to a standard basis measurement of its committed state, the Hadamard basis measurement is maximally uncertain. Note that this can be considered a binding property for the classical bit b since the ability to measure in the Hadamard basis implies the ability to reflect across the Hadamard basis axis, thus influencing the standard basis measurement.

More precisely, prior work has shown how to construct a PFC satisfying the above binding properties from (what we call) a *dual-mode randomized trapdoor claw-free hash function* with an *adaptive hard-code bit* property. We refer to this primitive as a "Type I" PFC or PFC-I, in order to differentiate it from a "Type II" PFC that we will construct in this work. We also note that in the body of this work, we build our protocols directly from the underlying claw-free hash function, so that we can appeal to theorems from prior work.[12] Thus the primitive of PFC-I does not appear explicitly in the body. However, in the remainder of this overview, we find it more convenient to explain these protocols using the primitive of PFC-I.

Verification of quantum partitioning circuits with one-time soundness. Now, we describe a privately-verifiable scheme for classical verification of quantum partitioning circuits that follows from prior work [Mah22, CLLW22, Bar21].

The starting point is a particular way to prepare a history state $|\psi_{Q,x}\rangle$ of the computation $Q(x)$, due to [CLLW22]. Given $|\psi_{Q,x}\rangle$, the verifier can either measure certain registers in the standard basis to obtain an approximate sample $q \leftarrow Q(x)$, or measure a random local Hamiltonian term (which involves just standard basis and Hadamard basis measurements). In [Bar21], the prover is instructed to prepare multiple copies of the history state, and the verifier chooses some subset

[12]However, we believe it could be interesting to re-prove prior results using the notion of PFC, and we leave an exploration of this possibility to future work. That is, does a PFC that satisfies the dual-mode and uncertainty binding properties generically imply classical verification of quantum computation?

for *sampling* (obtaining an output sample) and the other subset for *verifying* (measuring a local Hamiltonian term). If verification passes, the verifier collects the output samples $\{q_t\}_t$ and outputs the bit $b := \mathsf{Maj}(\{P(q_t)\}_t)$, which should be equal to $P(Q(x))$ with overwhelming probability.

Combining this approach with [Mah22]'s measurement protocol, applying parallel repetition, and finally applying Fiat-Shamir, we obtain the protocol described in Fig. 1.

Classical verification of quantum partitioning circuits with one-time soundness

Parameters: ℓ qubits per round, r total rounds, k Hadamard rounds.

Setup: Random oracle $H : \{0,1\}^* \to \{0,1\}^{\log \binom{r}{k}}$.

<u>$\mathsf{Gen}(1^\lambda, Q)$</u>

- For $i \in [r]$, choose a subset $S_i \subset [\ell]$ of qubits that will be measured in the standard basis to obtain output samples. Then, sample a string $h_i = (h_{i,1}, \ldots, h_{i,\ell}) \in \{0,1\}^\ell$ of basis choices[a] that are 0 on indices in S_i and otherwise correspond to random Hamiltonian terms.
- For $i \in [r], j \in [\ell]$, sample $(\mathsf{ck}_{i,j}, \mathsf{dk}_{i,j}) \leftarrow \mathsf{PFC\text{-}I.Gen}(1^\lambda, h_{i,j})$, and output

$$\mathsf{pp} := \{\mathsf{ck}_{i,j}\}_{i,j}, \ \ \mathsf{sp} := (\{h_i, S_i\}_i, \{\mathsf{dk}_{i,j}\}_{i,j}).$$

<u>$\mathsf{Prove}(1^\lambda, Q, \mathsf{pp}, x)$</u>

- Prepare sufficiently many copies of the history state $|\psi_{Q,x}\rangle$ on register $\mathcal{B} = \{\mathcal{B}_{i,j}\}_{i,j}$.
- For $i \in [r], j \in [\ell]$, apply $\mathsf{PFC\text{-}I.Com}(\mathsf{ck}_{i,j}, \mathcal{B}_{i,j}) \to (\mathcal{B}_{i,j}, \mathcal{U}_{i,j}, c_{i,j})$, and let $c := (c_{1,1}, \ldots, c_{r,\ell})$.
- Compute $T = H(c)$, where $T \in \{0,1\}^r$ has Hamming weight k.
- For $i : T_i = 0$ and $j \in [\ell]$, apply $\mathsf{PFC\text{-}I.OpenZ}(\mathcal{B}_{i,j}, \mathcal{U}_{i,j}) \to u_{i,j}$.
- For $i : T_i = 1$ and $j \in [\ell]$, apply $\mathsf{PFC\text{-}I.OpenX}(\mathcal{B}_{i,j}, \mathcal{U}_{i,j}) \to u_{i,j}$.
- Output $\pi := (c, u)$, where $u := (u_{1,1}, \ldots, u_{r,\ell})$.

<u>$\mathsf{Ver}(1^\lambda, Q, P, \mathsf{sp}, x, \pi)$</u>

- Parse $\pi = (c, u)$ as input and compute $T = H(c)$.
- For $i : T_i = 0$ and $j \in [\ell]$, check that $\mathsf{PFC\text{-}I.DecZ}(\mathsf{dk}_{i,j}, c_{i,j}, u_{i,j}) \neq \bot$.
- For $i : T_i = 1$ and $j \in [\ell]$:
  - If $h_{i,j} = 0$, compute the bit $b_{i,j} := \mathsf{PFC\text{-}I.Invert}(\mathsf{dk}_{i,j}, c_{i,j})$, and abort if $\bot$.
  - If $h_{i,j} = 1$, compute the bit $b_{i,j} := \mathsf{PFC\text{-}I.DecX}(\mathsf{dk}_{i,j}, c_{i,j}, u_{i,j})$, and abort if $\bot$.
- Apply a verification procedure to $\{b_{i,j}\}_{i:T_i=1, j\notin S_i}$ based on the Hamiltonian for $Q(x)$. If this passes, parse the bits $\{b_{i,j}\}_{i:T_i=1, j\in S_i}$ as a set of output samples $\{q_t\}_t$, and output $b := \mathsf{Maj}(\{P(q_t)\}_t)$.[b]

[a]We associate 0 with the standard basis and 1 with the Hadamard basis.

[b]For technical reasons, the final output is actually computed as a "majority of majorities", but we ignore that detail here.

Figure 1: A non-interactive privately-verifiable protocol for classical verification of quantum partitioning circuits, due to [Mah22, CLLW22, Bar21]. The circuit Q and predicate P are such that $P(Q(\cdot))$ is a pseudo-deterministic circuit.

In more detail, Fig. 1 consists of a number r of parallel rounds, where k of them are denoted "Hadamard" rounds, and the rest are denoted "test" rounds. Which rounds are Hadamard rounds are determined by a random oracle H applied to the prover's Pauli functional commitments c.

Each Hadamard round essentially runs a copy of the protocol described above, where the verifier obtains a number of output samples. We let ℓ denote the number of qubits per round, which is the number of history states per round times the number of qubits per history state. The standard basis measurements are obtained by inverting the commitments themselves (since these commitments are generated in mode $h = 0$), and the Hadamard basis measurements are obtained via the OpenX procedure. On the other hand, in the test rounds, the prover opens all of their commitments using the OpenZ procedure, and the verifier simply checks that DecZ does not reject these openings. We also note that the public and secret parameters $(\mathsf{pp}, \mathsf{sp})$ are generated independently of the input x, which was shown to be possible by an observation of [ACGH20].[13]

The one-time soundness of this protocol was proven in [Bar21], and relies on the soundness of the underlying measurement protocol due to [Mah22]. While the proof in [Mah22] actually required an additional property of the claw-free hash function beyond dual-mode and adaptive hard-core bit, the recent work of [BKL+22] showed that these two properties, which correspond to the dual-mode and uncertainty properties of the PFC, suffice for proving soundness.

Challenges with reusability. Now, our goal is to obtain soundness even against provers that have (superposition) oracle access to the verification algorithm. We denote this algorithm $\mathsf{Ver}[\mathsf{sp}](\cdot,\cdot)$, which has the secret parameters sp hard-coded (and implicitly $1^\lambda, Q$, and P), expects (x, π) as input, and outputs either a bit b or $\bot$.

Unfortunately, there is a simple attack on soundness in this setting. The main issue is that the secret parameters sp hard-code the measurement bases $h = (h_1, \dots, h_r)$, and soundness of the underlying information-theoretic protocol would be completely compromised if the prover could figure out h. Note that in the Hadamard rounds, the strings $u_{i,j}$ corresponding to $h_{i,j} = 0$ are completely ignored by the verifier, while the strings $u_{i,j}$ corresponding to $h_{i,j} = 1$ factor into the verifier's response. This discrepancy provides a way for the prover to learn the bits of $h_{i,j}$ by querying the verifier multiple times, ultimately breaking soundness of the protocol (see [BM22] for a more detailed discussion of this issue).

Can signature tokens help? Before coming to our solution, we discuss one promising but flawed attempt at upgrading to reusable soundness via the primitive of *signature tokens* [BS16]. A signature token consists of a quantum signing $|\mathsf{sk}\rangle$ that can be used to sign a *single* arbitrary message x, and then becomes useless.

So suppose we included $|\mathsf{sk}\rangle$ in the public parameters, and ask that the prover sign its proof π before querying $\mathsf{Ver}[\mathsf{sp}]$. That is, $\mathsf{Ver}[\mathsf{sp}]$ will now take as input (x, π, σ), and only respond if σ is a valid signature on π. Intuitively, if the prover tries to start collecting information from multiple malformed proofs in order to learn enough bits of h to break soundness, they should fail to produce the multiple signatures required to learn this information.

Unfortunately, this intuition is false. First, since the prover has *superposition* access to the verifier, they never have to actually output a classical signature σ. Moreover, in known signature token schemes [BS16], the public parameters can be used to implement a projection $|\mathsf{sk}\rangle\langle\mathsf{sk}|$ onto the original signing key. Thus, even though a prover may "damage" its state $|\mathsf{sk}\rangle$ by querying $\mathsf{Ver}[\mathsf{sp}]$ in superposition in order to learn a single bit of information about h, they could then

[13]Technically, Gen just needs to know the size of Q.

project back onto $|\mathsf{sk}\rangle$ via amplitude amplification. Thus, they could launch the same attacks as before, ultimately learning enough about h to break soundness.

2.3 Reusable soundness for a single instance

Classically, the following is a common route for boosting one-time soundness to reusable soundness for, say, an NP argument system. Note that any *fixed* instance x, either x is a yes instance, so we don't have to worry about the prover breaking soundness with respect to x, or x is a no instance, so by the one-time soundness of the protocol, the prover should never be able to make the verification oracle accept, rendering it useless. Thus, we can obtain reusable soundness if each instance x was associated with its own pair of public and secret parameters $(\mathsf{pp}_x, \mathsf{sp}_x)$. One method for achieving this is to fix the actual public parameters as an obfuscation of a program that takes x as input and samples parameters $(\mathsf{pp}_x, \mathsf{sp}_x)$ using randomness derived from a PRF applied to x (see [BGL$^+$15] for an example).

Although we would like to follow this approach, one difficulty is that in our setting the notion of an "instance" is unclear. The inputs x to the circuit cannot be classified into yes and no instances, since they all produce some valid outputs. In particular, note that the attacks on reusability outlined above will work even if the prover always queries the verification oracle on the same input x, eventually producing a π that causes the verifier to output $b \neq P(Q(x))$. A next attempt would be to start with some input x, sample $q \leftarrow Q(x)$, and consider the pair (x, q) to be an instance. However, since Q is a sampling circuit, it may be the case that this particular q is only sampled with small, or even negligible, probability on input x. Our one-time sound scheme is not equipped to prove a statement of the form, "q is in the support of the output of $Q(x)$". Thus, we will need a different approach.

Committing to the history state. Given an input x, we will essentially classify the *history state* of the computation $Q(x)$ into "yes" and "no" instances. That is, an honestly prepared history state $|\psi_{Q,x}\rangle$ should be classified as a yes instance, while any large enough perturbation to $|\psi_{Q,x}\rangle$ should be classified as a no instance. However, looking ahead, it will be crucial that our instances are classical so that we can generate parameters by applying a PRF to the instance. Thus, what we really need is a *classical commitment* to the history state. Moreover, after the state is committed, we still need it to be available for the prover to use in the one-time sound scheme. Fortunately, the prover only needs to perform standard and Hadamard basis measurements on the state (in addition to some operations that are classically controlled on the state). Thus, we have already discussed the exact primitive that we need - a Pauli functional commitment!

In Fig. 2, we outline a protocol where an instance $(x, \widetilde{c})$, consisting of an input x and a commitment $\widetilde{c}$ to a set of history states $|\psi_{Q,x}\rangle$, is generated and fixed before the protocol begins. We use a Pauli functional commitment denoted $\mathsf{PFC\text{-}II}$ to commit to the history states (since we will eventually require $\mathsf{PFC\text{-}II}$ to satisfy different properties than $\mathsf{PFC\text{-}I}$).

We remark that correctness of this protocol relies on a couple of specific properties: (1) $\mathsf{PFC\text{-}I.Com}$ and $\mathsf{PFC\text{-}II.Com}$ are both *classically controlled* on the register $\mathcal{B}$, so they commute with each other, and (2) $\mathsf{PFC\text{-}I.OpenZ}$ (resp. $\mathsf{PFC\text{-}I.OpenX}$) simply measures the register $\mathcal{B}$ in the standard (resp. Hadamard) basis[14] so the first bit of the string u can be computed instead by applying $\mathsf{PFC\text{-}II.Com}$ to $\mathcal{B}$ followed by $\mathsf{PFC\text{-}II.OpenZ}$ and $\mathsf{PFC\text{-}II.DecZ}$ (resp. $\mathsf{PFC\text{-}II.OpenX}$ and $\mathsf{PFC\text{-}II.DecX}$).

[14]Though it could be performing an arbitrary operation to the $\mathcal{U}$ register.

Now, our goal will be to obtain reusable soundness for any fixed instance $(x, \widetilde{c})$. That is, we give the prover oracle access to $\mathsf{Ver}[\mathsf{sp}, \widetilde{\mathsf{dk}}, (x, \widetilde{c})](\cdot)$ where $\widetilde{\mathsf{dk}}$ and $(x, \widetilde{c})$ are now hard-coded and the only input is a proof π, and require that the prover cannot make the verifier output $b \neq P(Q(x))$.

A protocol with reusable soundness for a single "instance"

Parameters: ℓ qubits per round, r total rounds, k Hadamard rounds.

Setup: Random oracle $H : \{0,1\}^* \to \{0,1\}^{\log \binom{r}{k}}$.

Instance generation

- For $i \in [r], j \in [\ell]$, the verifier samples $(\widetilde{\mathsf{ck}}_{i,j}, \widetilde{\mathsf{dk}}_{i,j}) \leftarrow \mathsf{PFC\text{-}II.Gen}(1^\lambda)$, outputs $\widetilde{\mathsf{ck}} := \{\widetilde{\mathsf{ck}}_{i,j}\}_{i,j}$, and keeps $\widetilde{\mathsf{dk}} := \{\widetilde{\mathsf{dk}}_{i,j}\}_{i,j}$ private.
- Given an input x, the prover prepares sufficiently many copies of the history state $|\psi_{Q,x}\rangle$ on register $\mathcal{B} = \{\mathcal{B}_{i,j}\}_{i,j}$.
- For $i \in [r]$, $j \in [\ell]$, the prover applies $\mathsf{PFC\text{-}II.Com}(\widetilde{\mathsf{ck}}_{i,j}, \mathcal{B}_{i,j}) \to (\mathcal{B}_{i,j}, \widetilde{\mathcal{U}}_{i,j}, \widetilde{c}_{i,j})$. Then, it sets $\widetilde{c} := (\widetilde{c}_{1,1}, \ldots, \widetilde{c}_{r,\ell})$ and outputs the instance $(x, \widetilde{c})$.

$\mathsf{Gen}(1^\lambda, Q)$

- The verifier samples $\mathsf{pp} = \{\mathsf{ck}_{i,j}\}_{i,j}, \mathsf{sp} = (\{h_i, S_i\}_i, \{\mathsf{dk}_{i,j}\}_{i,j})$ as in Fig. 1.

$\mathsf{Prove}(1^\lambda, Q, \mathsf{pp}, x)$

- For $i \in [r], j \in [\ell]$, apply $\mathsf{PFC\text{-}I.Com}(\mathsf{ck}_{i,j}, \mathcal{B}_{i,j}) \to (\mathcal{B}_{i,j}, \mathcal{U}_{i,j}, c_{i,j})$, and let $c := (c_{1,1}, \ldots, c_{r,\ell})$.
- Compute $T = H(c)$, where $T \in \{0,1\}^r$ has Hamming weight k.
- For $i : T_i = 0$ and $j \in [\ell]$, apply $\mathsf{PFC\text{-}II.OpenZ}(\mathcal{B}_{i,j}, \widetilde{\mathcal{U}}_{i,j}) \to \widetilde{u}_{i,j}$ followed by $\mathsf{PFC\text{-}I.OpenZ}(\mathcal{B}_{i,j}, \mathcal{U}_{i,j}) \to u_{i,j}$. Let $u'_{i,j}$ be $u_{i,j}$ with the first bit removed.
- For $i : T_i = 1$ and $j \in [\ell]$, apply $\mathsf{PFC\text{-}II.OpenX}(\mathcal{B}_{i,j}, \widetilde{\mathcal{U}}_{i,j}) \to \widetilde{u}_{i,j}$ followed by $\mathsf{PFC\text{-}I.OpenX}(\mathcal{B}_{i,j}, \mathcal{U}_{i,j}) \to u_{i,j}$. Let $u'_{i,j}$ be $u_{i,j}$ with the first bit removed.
- Output $\pi := (c, \widetilde{u}, u)$, where $\widetilde{u} := (\widetilde{u}_{1,1}, \ldots, \widetilde{u}_{r,\ell})$ and $u := (u'_{1,1}, \ldots, u'_{r,\ell})$.

$\mathsf{Ver}(1^\lambda, Q, P, \mathsf{sp}, \widetilde{\mathsf{dk}}, (x, \widetilde{c}), \pi)$

- Parse $\pi = (c, \widetilde{u}, u)$ and compute $T = H(c)$.
- For $i : T_i = 0$ and $j \in [\ell]$, compute $b'_{i,j} := \mathsf{PFC\text{-}II.DecZ}(\widetilde{\mathsf{dk}}_{i,j}, \widetilde{c}_{i,j}, \widetilde{u}_{i,j})$ and check that $\mathsf{PFC\text{-}I.DecZ}(\mathsf{dk}_{i,j}, c_{i,j}, (b'_{i,j}, u'_{i,j})) \neq \bot$.
- For $i : T_i = 1$ and $j \in [\ell]$:
  - If $h_{i,j} = 0$, compute the bit $b_{i,j} := \mathsf{PFC\text{-}I.Invert}(\mathsf{dk}_{i,j}, c_{i,j})$, and abort if $\bot$.
  - If $h_{i,j} = 1$, compute $b'_{i,j} := \mathsf{PFC\text{-}II.DecX}(\widetilde{\mathsf{dk}}_{i,j}, \widetilde{c}_{i,j}, \widetilde{u}_{i,j})$, followed by the bit $b_{i,j} := \mathsf{PFC\text{-}I.DecX}(\mathsf{dk}_{i,j}, c_{i,j}, (b'_{i,j}, u'_{i,j}))$, and abort if $\bot$.
- Apply a verification procedure to $\{b_{i,j}\}_{i:T_i=1, j \notin S_i}$ based on the Hamiltonian for $Q(x)$. If this passes, parse the bits $\{b_{i,j}\}_{i:T_i=1, j \in S_i}$ as a set of output samples $\{q_t\}_t$, and output $b := \mathsf{Maj}(\{P(q_t)\}_t)$.

Figure 2: A protocol for classical verification of quantum partitioning circuits that is reusably sound for each fixed instance $(x, \widetilde{c})$.

Binding. Following the classical intuition, we would like to split $(x, \widetilde{c})$ into yes and no instances:

1. "Yes" instance: $\widetilde{c}$ can only be opened in a way that would cause the verifier to output $b = P(Q(x))$ (or $\bot$). In this case, the prover could potentially learn the secret parameters sp via repeated queries, but would not be able to break soundness.

2. "No" instance: $\widetilde{c}$ can only be opened in a way that would cause the verifier to output $b \neq P(Q(x))$ (or $\bot$). In this case, by one-time soundness of the underlying protocol, the prover should never be able to make the verifier output anything other than $\bot$.

Now, a crucial difference from the classical case is that a prover might launch a *superposition* of both strategies, so we can't exactly classify each $(x, \widetilde{c})$ as either a yes or a no instance. However, in this case we will hope to rely on some notion of binding from the PFC-II commitment scheme in order to guarantee that the prover cannot meaningfully "mix" these two strategies.

As discussed above, Pauli functional commitments satisfy a notion of binding to *classical bits* rather than to quantum states, so we will need to capture these two options using classical openings. For the first option, the parallel repetition theorem of [ACGH20, Bar21] can be used to show that if the verifier accepts, then *many*, say 4/5, of their output samples q_t from indices $\{S_i\}_{i:T_i=1}$ must be such that $P(q_t) = P(Q(x))$. For the second option, it is clear that the verifier will only output $b \neq P(Q(x))$ if at least half of these output samples are such that $P(q_t) \neq P(Q(x))$. Thus, it suffices to show that the prover can't mix the following strategies.

1. Open $\widetilde{c}$ on the positions $\{S_i\}_{i:T_i=1}$ to samples q_t such that *a large fraction* (say 4/5) of them are "honest": $P(q_t) = P(Q(x))$.

2. Open $\widetilde{c}$ on the positions $\{S_i\}_{i:T_i=1}$ to samples q_t such that *a significant fraction* (say 1/2) of them are "dishonest": $P(q_t) \neq P(Q(x))$.

Since the $\{S_i\}_i$ positions are all standard basis positions, and no string can satisfy both requirements, arguing that these strategies can't mix should now reduce to some binding property for the classical strings opened on the $\{S_i\}_{i:T_i=1}$ positions. However, note that in Fig. 2, none of these positions are even opened by PFC-II.OpenZ (that is, opened in the standard basis)! Indeed, *only* the test round positions are opened in the standard basis.

Thus, we need to relate the strings opened on $\{S_i\}_{i:T_i=1}$ to the strings opened on $\{S_i\}_{i:T_i=0}$. Now, we note that T is chosen via a random oracle applied to c, and c already determines the only possible openings for the standard basis positions since the PFC-I parameters are sampled in perfectly binding mode on these positions. Thus, it is possible to argue that the adversary can't significantly change their distribution of opened strings on test round vs. Hadamard round positions. So it suffices to show that the following strategies can't mix:

1. Open $\widetilde{c}$ on the positions $\{S_i\}_{i:T_i=0}$ to samples q_t such that *a large fraction* (say 3/4) of them are "honest": $P(q_t) = P(Q(x))$.

2. Open $\widetilde{c}$ on the positions $\{S_i\}_{i:T_i=0}$ to samples q_t such that *a significant fraction* (say 1/3) of them are "dishonest": $P(q_t) \neq P(Q(x))$.

Thus, we will only need a "vanilla" notion of string binding for PFC-II, which can be reduced (see Section 4.1 for more discussion) to a vanilla notion of single-bit binding for a quantum commitment to a classical bit. That is, given a decoding key $\widetilde{\mathsf{dk}}$, a commitment $\widetilde{c}$, and a bit b, let

$$\Pi_{\widetilde{\mathsf{dk}},\widetilde{c},b} \coloneqq \sum_{\widetilde{u}:\mathsf{DecZ}(\widetilde{\mathsf{dk}},\widetilde{c},\widetilde{u})=b} |\widetilde{u}\rangle\langle\widetilde{u}|$$

be the projection onto strings $\widetilde{u}$ that open to b. Then for any two-part adversary (C, U), where C is the committer, and U is the "opener"[15] (modeled as a unitary), it holds that for any $b \in \{0, 1\}$,

$$\mathop{\mathbb{E}}_{(\widetilde{\mathsf{ck}},\widetilde{\mathsf{dk}})\leftarrow\mathsf{Gen}(1^\lambda)}\left[\left\|\Pi_{\widetilde{\mathsf{dk}},\widetilde{c},1-b}\mathsf{U}\Pi_{\widetilde{\mathsf{dk}},\widetilde{c},b}\left|\psi\right\rangle\right\| : (\left|\psi\right\rangle, \widetilde{c}) \leftarrow \mathsf{C}(\widetilde{\mathsf{ck}})\right] = \mathrm{negl}(\lambda).$$

A couple of remarks:

- Looking at Fig. 2, we see that this binding property should hold *even* if the opener has oracle access to $\mathsf{DecZ}(\widetilde{\mathsf{dk}}, \widetilde{c}, \cdot)$. In fact, in the known construction of PFC-I described above [BCM+21, Mah22], DecZ decoding can be *public*. Moreover, this definition of binding is weaker than both the dual-mode and uncertainty properties, and thus our requirements for PFC-II can *so far* be satisfied by the known construction of PFC-I.

- Note that we only require binding on the standard basis positions, that is, (i, j) such that $h_{i,j} = 0$. Looking at Fig. 2, we see that the prover does *not* have access to $\mathsf{DecX}(\widetilde{\mathsf{dk}}_{i,j}, \widetilde{c}_{i,j}, \cdot)$ on these positions. This is important, because the ability to perform a Hadamard basis measurement on the committed qubit implies the ability to reflect it across the X (Hadamard basis) axis, thus changing its standard basis measurement. Thus, it seems difficult to design a Pauli functional commitment scheme that remains binding when the opener has access to DecX.

Proving soundness for a single instance. Next, we briefly discuss how soundness for a single instance can be proven based on this binding property of PFC-II. We start with an adversary that is assumed to be breaking soundness after a number of queries to the verification oracle. That is, they output a proof π^* that causes the verifier to accept and output $b \neq P(Q(x))$. We know that a significant fraction of the samples q_t from positions $\{S_i\}_{i:T_i=0}$ in π^* must be such that $Q(q_t) \neq P(Q(x))$. Then, we replace each of the adversary's $\mathsf{Ver}[\mathsf{sp}, \widetilde{\mathsf{dk}}, (x, \widetilde{c})]$ queries one by one to being answered with $\perp$. While the adversary may query $\mathsf{Ver}[\mathsf{sp}, \widetilde{\mathsf{dk}}, (x, \widetilde{c})]$ on accepting π, we know that for such π, a large fraction of the samples q_t from positions $\{S_i\}_{i:T_i=0}$ must be such that $Q(q_t) = P(Q(x))$. Thus, by the binding of PFC-II, the fact that we are changing the oracle's response to such π should have a negligible effect on the probability that the adversary continues to output π^*, since π and π^* contain openings to different strings and thus reside in parts of the adversary's state that have negligible overlap. After replacing all of these queries with $\perp$, we see that our adversary is actually breaking soundness of the underlying one-time sound protocol, since they no longer learn anything from their queries to $\mathsf{Ver}[\mathsf{sp}, \widetilde{\mathsf{dk}}, (x, \widetilde{c})]$, which completes the proof. For more details, see the discussion before the "soundness" part of the proof of Theorem 5.12.

2.4 Public verifiability in the oracle model

Next, we show how to obtain full-fledged public-verifiability in the oracle model. As a first attempt, we follow the classical approach, and include in the public parameters the PFC-II parameters $\{\widetilde{\mathsf{ck}}_{i,j}\}_{i,j}$ along with a classical oracle that implements the following program $\mathsf{OGen}[k]$, which

[15] More precisely, U is an algorithm that tries to break binding by rotating a state that is supported on valid openings to b to a state that is supported on valid openings to $1 - b$. We refer to this part of the adversary as the opener.

has a PRF key k hard-coded.

$\underline{\mathsf{OGen}[k]}$:

- Take an x and a commitment $\widetilde{c}$ as input, and compute $s := \mathsf{PRF}_k((x, \widetilde{c}))$.
- Compute $(\mathsf{pp}, \mathsf{sp}) := \mathsf{Gen}(1^\lambda; s)$ from Fig. 1 using random coins s, and output pp.

Unfortunately, this attempt does not result in a sound scheme. To see why, note that the adversary can query the verification oracle on multiple $(x, \widetilde{c})$, thus using it to implement the oracle $\mathsf{PFC\text{-}II.DecX}(\widetilde{\mathsf{dk}}_{i,j}, \cdot, \cdot)$ for *any* index (i, j) of its choice. Indeed, for each index (i, j), the adversary just has to find some $(x, \widetilde{c})$ that generates parameters with $h_{i,j} = 1$. As mentioned above, if the opener has access to $\mathsf{PFC\text{-}II.DecX}(\widetilde{\mathsf{dk}}_{i,j}, \cdot, \cdot)$, it is not clear how to obtain any binding property for the bit on index (i, j). Thus, an adversary could break soundness on a particular instance $(x, \widetilde{c})$ by querying its oracles on *other* instances $(x', \widetilde{c}')$ in order to obtain access to any $\mathsf{PFC\text{-}II.DecX}(\widetilde{\mathsf{dk}}_{i,j}, \cdot, \cdot)$ of its choice.

Using signature tokens. To solve this issue, we use signature tokens to make sure that the adversary's strategy on multiple distinct $(x, \widetilde{c})$ cannot "mix". That is, we include the signing key $|\mathsf{sk}\rangle$ for a signature token scheme in the public parameters, and alter $\mathsf{OGen}[k]$ as follows, where vk is the verification key for the signature token scheme.

$\underline{\mathsf{OGen}[k, \mathsf{vk}]}$:

- Take an x, a commitment $\widetilde{c}$, and a signature σ as input.
- If σ is a valid signature of $(x, \widetilde{c})$ under vk, compute $s := \mathsf{PRF}_k((x, \widetilde{c}, \sigma))$, and otherwise abort.
- Compute $(\mathsf{pp}, \mathsf{sp}) := \mathsf{Gen}(1^\lambda; s)$ from Fig. 1 using random coins s, and output pp.

Moreover, the verification oracle $\mathsf{Ver}[\mathsf{vk}, k]$, which now hard-codes k rather than some fixed secret parameters sp, will also require a valid signature σ on any $(x, \widetilde{c})$ that it takes as input. Intuitively, once the adversary learns the public parameters $\mathsf{pp}_{x,\widetilde{c},\sigma}$ corresponding to some instance $(x, \widetilde{c})$ and signature σ, it can *only* access the oracles $\mathsf{PFC\text{-}II.DecX}(\mathsf{dk}_{i,j}, \cdot, \cdot)$ on the specific indices (i, j) such that $h_{i,j} = 1$ for the h hard-coded in parameters $\mathsf{pp}_{x,\widetilde{c},\sigma}$. Note that this actually requires the signature token scheme to be *strongly unforgeable*. That is, the adversary shouldn't even be able to produce a different signature σ' on the same message $(x, \widetilde{c})$, since then $(x, \widetilde{c}, \sigma')$ could be used to generate a fresh set of parameters with different h. While this notion was not proven explicitly in [BS16], we note that it follows easily from their proof strategy.

To formalize this intuition, we treat the PRF as a random oracle H and make use of the measure and re-program technique of [DFMS19, DFM20]. If the adversary is breaking soundness, it must output a proof π with respect to some $(x, \widetilde{c}, \sigma)$. Thus, we can "pre-measure" one of the adversary's queries to H to obtain $(x, \widetilde{c}, \sigma)$, and then re-program $H((x, \widetilde{c}, \sigma)) \to s$ to fresh randomness s, which defines fresh parameters $(\mathsf{pp}_{x,\widetilde{c},\sigma}, \mathsf{sp}_{x,\widetilde{c},\sigma})$. After this measurement, by the strong unforgeability of the signature token, the adversary won't be able to query the verification oracle on any $(x', \widetilde{c}', \sigma') \neq (x, \widetilde{c}, \sigma)$, so they will only be able to access $\mathsf{DecX}(\widetilde{\mathsf{dk}}_{i,j}, \cdot, \cdot)$ for (i, j) such that $h_{i,j} = 1$ as defined by $\mathsf{pp}_{x,\widetilde{c},\sigma}$. Then, security should reduce to the single instance setting discussed above.

It is useful to note a crucial difference from the more direct but flawed approach to using signature tokens discussed earlier in the overview. There, we could never hope to use the security of the signature token, because we couldn't "force" the adversary to ever measure a signature (and indeed there was an attack on the attempted scheme). Here, since we are using the signature as part of the input to a random oracle, we can make use of measure-and-reprogram to first "force" a measurement of a signature during the security proof, and *then* use signature token security.

The need for public decodability. However, we have so far omitted a crucial detail. Note that *before* the measurement of $(x, \widetilde{c}, \sigma)$, the adversary *can* access any DecX oracle of its choice. Indeed, we can't hope to prevent this, as the adversary has full access to both $\mathsf{OGen}[k, \mathsf{vk}]$ and $\mathsf{Ver}[k, \mathsf{vk}]$, and this measurement anyway only happens during an intermediate hybrid in the proof.

In the reduction to the binding of $\mathsf{PFC}\text{-}\mathsf{II}$, this first part of the adversary corresponds to the *commit* stage. Thus, we will need a Pauli functional commitment scheme where the *committer* has access to both the DecZ and DecX oracles, while the *opener* (necessarily) only has access to DecZ.

We refer to such a commitment scheme as a *Pauli functional commitment with public decodability*. Somewhat more formally, we will require the following binding property, where $\mathsf{DecZ}[\mathsf{dk}]$ (resp. $\mathsf{DecX}[\mathsf{dk}]$) is the oracle implementing the classical functionality $\mathsf{DecZ}(\mathsf{dk}, \cdot, \cdot)$ (resp. $\mathsf{DecX}(\mathsf{dk}, \cdot, \cdot)$). For any polynomial-query adversary (C, U),

$$\Pr_{(\widetilde{\mathsf{ck}}, \widetilde{\mathsf{dk}}) \leftarrow \mathsf{Gen}(1^\lambda)} \left[\left\| \Pi_{\widetilde{\mathsf{dk}}, \widetilde{c}, 1-b} \mathsf{U}^{\mathsf{DecZ}[\widetilde{\mathsf{dk}}]} \Pi_{\widetilde{\mathsf{dk}}, \widetilde{c}, b} \left| \psi \right\rangle \right\| = 1/\mathrm{poly}(\lambda) : (\left| \psi \right\rangle, \widetilde{c}) \leftarrow \mathsf{C}^{\mathsf{DecZ}[\widetilde{\mathsf{dk}}], \mathsf{DecX}[\widetilde{\mathsf{dk}}]}(\widetilde{\mathsf{ck}}) \right] = \mathrm{negl}(\lambda).$$

Unfortunately, the known construction of Pauli functional commitments [BCM+21, Mah22] does not satisfy this property, which we explain in the following section. Thus, in the remainder of this overview, we demonstrate a novel approach to constructing Pauli functional commitments, and describe a construction with public decodability in the oracle model. Once we have this commitment, our construction of non-interactive publicly-verifiable classical verification of quantum partitioning circuits is complete, which also completes our construction of obfuscation for pseudo-deterministic quantum circuits.

2.5 Pauli functional commitments with public decodability

First, we review why the Pauli functional commitment based on claw-free hash functions [BCM+21, Mah22] does not satisfy binding with public decodability. To commit to a state $\left|\psi\right\rangle = \alpha_0 \left|0\right\rangle + \alpha_1 \left|1\right\rangle$, the committer evaluates and measures an (approximately) two-to-one hash function f in superposition to end up with a commitment c and a left-over state $\alpha_0 \left|0\right\rangle \left|x_0\right\rangle + \alpha_1 \left|1\right\rangle \left|x_1\right\rangle$, where x_0, x_1 are n-bit strings such that x_0 starts with 0 and x_1 starts with 1. If they do this honestly, it will hold that $f(x_0) = f(x_1) = c$. Moreover, the receiver has a trapdoor for f and can thus compute both x_0 and x_1 from c.

Now, a standard basis opening to the bit b is the string x_b. To open $\left|\psi\right\rangle$ in the Hadamard basis, the committer measures each qubit of their left-over state in the Hadamard basis, obtaining a bit b' and a string d. It follows that $b \coloneqq b' + d \cdot (x_0 + x_1)$[16] is a decoding of the Hadamard basis measurement of $\left|\psi\right\rangle$. Thus, if we define $S \coloneqq \{0, x_0 + x_1\}$ to be a one-dimensional subspace of $\mathbb{F}_2^n$, access to the DecX oracle provides the committer with a membership oracle for the subspace

[16] Here, and throughout this section, all arithmetic will be over $\mathbb{F}_2$.

$S^\perp$. Since S is just one dimension, it is straightforward to use this oracle to learn a description of S, which is $x_0 + x_1$. But if the committer C computes the string $x_0 + x_1$ and passes it along with $\alpha_0 |0\rangle |x_0\rangle + \alpha_1 |1\rangle |x_1\rangle$ to U, the opener can first measure their state in the standard basis to obtain (b, x_b), and then use $x_0 + x_1$ to compute $(1 - b, x_{1-b})$, obtaining a valid opening for *both* bits in the standard basis. This completely breaks any notion of binding for the commitment scheme.

Using a larger subspace. To solve this issue, we follow this template but increase the dimension of S, thus decreasing the dimension of $S^\perp$. That is, suppose that the left-over state after a commitment to $|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$ was instead

$$\alpha_0 |0\rangle |A_0\rangle + \alpha_1 |1\rangle |A_1\rangle ,$$

where $A = S + v$ is a coset of a random $n/2$-dimensional subspace S,[17] A_0 is the affine subspace of vectors in A that start with 0, and A_1 is the affine subspace of vectors in A that start with 1. Here, we are using the notation

$$|A\rangle := \frac{1}{\sqrt{|A|}} \sum_{s \in A} |s\rangle$$

for any affine subspace A.

It can be shown that if this state is measured in the Hadamard basis to produce b', d, then $b := b' \oplus r_{d,S}$ is a decoding of the Hadamard basis measurement of $|\psi\rangle$, where we define the bit $r_{d,S} = 0$ if $d \in S^\perp$ and $r_{d,S} = 1$ if $d + (1, 0, \ldots, 0) \in S^\perp$. Thus, the DecX oracle can be implemented just given a membership checking oracle for $S^\perp$. Moreover, now that $S^\perp$ has $n/2$ dimensions, and S is random, it is no longer clear that an adversary can use oracle access to $S^\perp$ to learn a description of S.

Completing the construction. Now, two main questions remain: (1) How do we define a commitment key ck that enables the committer to apply the map $|b\rangle \to |b\rangle |A_b\rangle$? (2) What is the actual *commitment string* c? We will first address question (1).

Our commitment key will consist of a quantum state and a classical oracle. The Gen algorithm will sample a random $n/2$-dimensional affine subspace $A = S + v$, set $\mathsf{dk} = A$, and release the quantum state $|A\rangle$, which is a uniform superposition over all vectors in A. Note that $|A\rangle = \frac{1}{\sqrt{2}} |A_0\rangle + \frac{1}{\sqrt{2}} |A_1\rangle$, which can be seen as the "$|+\rangle$" state in the two-dimensional space spanned by $|A_0\rangle$ and $|A_1\rangle$. Thus, for any $b \in \{0, 1\}$, we need to allow the committer to rotate the $|+\rangle$ state to the "$|b\rangle$" state $|A_b\rangle$. It is easy to project onto vectors that start with either 0 or 1, but we will have to implement a reflection across the X-axis of this space if this projection results in $|A_{1-b}\rangle$. While it is clear that this can be done given a quantum oracle implementing the projection $|A\rangle \langle A|$, it was observed by [AGKZ20] that a *classical* oracle for membership in $S^\perp$ suffices! Thus, as a first attempt, we will set the commitment key ck to consist of $|A\rangle$ and an oracle $O[S^\perp]$ for membership in $S^\perp$.

This brings us to our second question. So far, we have shown that a committer, given ck, can perform the map

$$\alpha_0 |0\rangle + \alpha_1 |1\rangle \to \alpha_0 |0\rangle |A_0\rangle + \alpha_1 |1\rangle |A_1\rangle ,$$

[17]Assume that A and S are "balanced", meaning that exactly half of their vectors start with 0.

and give this final state to the opener. However, since the opener also has access to ck and thus to $O[S^\perp]$, there is no sense in which the original state is committed, since the opener could continue to use $O[S^\perp]$ to rotate arbitrarily around the space spanned by $|A_0\rangle$ and $|A_1\rangle$.

To fix this, we use a signature token. We include the signing key $|\mathsf{sk}\rangle$ for a single-bit signature token scheme in ck, and alter the oracle $O[S^\perp]$ so that it only responds given a valid signature on 0. The actual commitment string c will then be a signature on 1. Thus, while the *committer* is free to rotate around $\mathsf{span}\{|A_0\rangle, |A_1\rangle\}$ using access to $S^\perp$, as soon as it outputs a valid classical commitment string c, the membership oracle for $S^\perp$ will become inaccessible and the *opener* will intuitively be unable to make further changes to the state.

The proof of binding. Now, it remains to formalize this intuition, and prove that this scheme satisfies binding with public decodability. After appealing to the security of the signature token scheme, we can reduce this to showing that for any polynomial-query adversary (C, U),

$$\Pr\left[\left\|\Pi_{A_1}\mathsf{U}^{O[A]}\Pi_{A_0}|\psi\rangle\right\| \geq 1/\mathrm{poly}(\lambda) : |\psi\rangle \leftarrow \mathsf{C}^{O[A],O[S^\perp]}(|A\rangle)\right] = \mathrm{negl},$$

where the probability is over a random choice of $n/2$-dimensional affine subspace $A = S + v$, and Π_{A_b} is the projection onto vectors $s \in A_b$. Note that C and U have access to $O[A]$, the membership checking oracle for the affine subspace A since this is needed to implement DecZ, and C has access to $O[S^\perp]$ because it is needed to implement both ck and DecX.

To show this, we will follow [AC12]'s blueprint for proving security in the classical oracle model, and proceed via the following steps.

1. Show that we can instead sample A from a public ambient space of dimension $3n/4$, and remove U's access to the $O[A]$ oracle.

2. Perform a worst-case to average-case reduction over the sampling of A.

3. Have the committer apply amplitude amplification onto Π_{A_0}. At this point, we can reduce the problem to showing that for small enough ϵ, there cannot exist a query-bounded C and a unitary U such that for *all* $n/2$-dimensional affine subspaces A of $\mathbb{F}_2^{3n/4}$,

$$|\psi_A\rangle \in \mathsf{Im}(\Pi_{A_0}) \text{ and } \left\|\Pi_{A_1}\mathsf{U}|\psi_A\rangle\right\| \geq \epsilon,$$

   where $|\psi_A\rangle \leftarrow \mathsf{C}^{O[A],O[S^\perp]}(|A\rangle)$.

4. Apply the "inner-product adversary method" of [AC12]. That is, we (i) define a relation $\mathcal{R}$ on pairs of affine subspaces (A, B) such that $\langle A|B\rangle = 1/2$ for all $(A, B) \in \mathcal{R}$, (ii) argue that for any collection of states $\{|\psi_A\rangle\}_A$ that satisfy the above conditions,

$$\mathop{\mathbb{E}}_{(A,B)\leftarrow\mathcal{R}}[|\langle\psi_A|\psi_B\rangle|] \leq 1/2 - \delta$$

   for some large enough δ, and (iii) conclude that if C can decrease the expected inner product over $\mathcal{R}$ by δ, it must be making "too many" oracle queries, yielding a contradiction.

However, arguing part (ii) of this final step turns out to be significantly more challenging than analogous claims in previous work (e.g. [AC12, BS16, AGKZ20]). Indeed, the condition is neither that $|\psi_A\rangle$ is some *fixed* state (as in [AC12]), or that measuring $|\psi_A\rangle$ in the standard basis yields a classical string in some well-defined set (as in [BS16, AGKZ20]). Rather, the condition involves reasoning about the overlap between two projectors, where one is defined via an *arbitrary* rotation U. Moreover, we only have the guarantee that $|\psi_A\rangle$ is ϵ-close to $\mathsf{Im}(\mathsf{U}^\dagger\Pi_{A_1}\mathsf{U})$, and this value cannot be amplified to 1 (depending on U, the images of Π_{A_0} and $\mathsf{U}^\dagger\Pi_{A_1}\mathsf{U}$ may not intersect at all).

In Appendix B, we show that for our definition of $\mathcal{R}$, $\delta > \epsilon^{13}$, which is enough for us to reach a contradiction and complete the proof. We proceed by contradiction, and eventually reduce to a Welch bound [Wel74], which upper bounds the number of vectors of a given minimum distance that can be packed into a low-dimensional Hilbert space. We defer a further overview and details of this proof to Appendix B. This completes our proof of binding with public decodability.

3 Preliminaries

Let λ denote the security parameter. We write $\mathrm{negl}(\cdot)$ to denote any *negligible* function, which is a function f such that for every constant $c \in \mathbb{N}$ there exists $N \in \mathbb{N}$ such that for all $n > N$, $f(n) < n^{-c}$. We write $\mathsf{non\text{-}negl}(\cdot)$ to denote any function f that is not negligible. That is, there exists a constant c such that for infinitely many n, $f(n) \geq n^{-c}$. Finally, we write $\mathrm{poly}(\cdot)$ to denote any polynomial function f. That is, there exists a constant c such that for all $n \in \mathbb{N}$, $f(n) \leq n^{-c}$. For two probability distributions D_0, D_1 with classical support S, let

$$\mathsf{TV}\left(D_0, D_1\right) \coloneqq \sum_{x \in S} |D_0(x) - D_1(x)|$$

denote the total variation distance. For a set S, we let $x \leftarrow S$ denote sampling a uniformly random element x from S. For a classical randomized algorithm $y \leftarrow C(x)$, we let $y \coloneqq C(x; r)$ denote running C with random coins r.

3.1 Quantum information

An n-qubit register $\mathcal{X}$ is a named Hilbert space $\mathbb{C}^{2^n}$. A pure quantum state on register $\mathcal{X}$ is a unit vector $|\psi\rangle^{\mathcal{X}} \in \mathbb{C}^{2^n}$. A mixed state on register $\mathcal{X}$ is described by a density matrix $\rho^{\mathcal{X}} \in \mathbb{C}^{2^n \times 2^n}$, which is a positive semi-definite Hermitian operator with trace 1.

A *quantum operation* F is a completely-positive trace-preserving (CPTP) map from a register $\mathcal{X}$ to a register $\mathcal{Y}$, which in general may have different dimensions. That is, on input a density matrix $\rho^{\mathcal{X}}$, the operation F produces $F(\rho^{\mathcal{X}}) = \tau^{\mathcal{Y}}$ a mixed state on register $\mathcal{Y}$. A *unitary* $U : \mathcal{X} \to \mathcal{X}$ is a special case of a quantum operation that satisfies $U^\dagger U = UU^\dagger = \mathbb{I}^{\mathcal{X}}$, where $\mathbb{I}^{\mathcal{X}}$ is the identity matrix on register $\mathcal{X}$. A *projector* Π is a Hermitian operator such that $\Pi^2 = \Pi$, and a *projective measurement* is a collection of projectors $\{\Pi_i\}_i$ such that $\sum_i \Pi_i = \mathbb{I}$. Throughout this work, we will often write an expression like $\Pi |\psi\rangle$, where $|\psi\rangle$ has been defined on some multiple registers, say $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$, and Π has only been defined on a subset of these registers, say $\mathcal{Y}$. In this case, we technically mean $(\mathbb{I}^{\mathcal{X}} \otimes \Pi \otimes \mathbb{I}^{\mathcal{Z}}) |\psi\rangle$, but we drop the identity matrices for notational convenience.

A family of quantum circuits is in general a sequence of quantum operations $\{C_\lambda\}_{\lambda \in \mathbb{N}}$, parameterized by the security parameter. We say that the family is *quantum polynomial time* (QPT) if C_λ can be implemented with a $\mathrm{poly}(\lambda)$-size circuit. A family of *oracle-aided* quantum circuits $\{C_\lambda^F\}_{\lambda \in \mathbb{N}}$ have access to an oracle $F : \{0,1\}^* \to \{0,1\}^*$ that implements some classical map. That is, C can apply a unitary that maps $|x\rangle |y\rangle \to |x\rangle |y \oplus F(x)\rangle$. Finally, we will sometimes also consider families of *unitaries* $\{U_\lambda\}_{\lambda\in\mathbb{N}}$ and families of *oracle-aided unitaries* $\{U_\lambda^F\}_{\lambda\in\mathbb{N}}$, where each operation between oracle queries is a unitary.

Let Tr denote the trace operator. For registers $\mathcal{X}, \mathcal{Y}$, the *partial trace* $\mathsf{Tr}^{\mathcal{Y}}$ is the unique operation from $\mathcal{X}, \mathcal{Y}$ to $\mathcal{X}$ such that for all $(\rho,\tau)^{\mathcal{X},\mathcal{Y}}$, $\mathsf{Tr}^{\mathcal{Y}}(\rho,\tau) = \mathsf{Tr}(\tau)\rho$. The *trace distance* between states ρ, τ, denoted $\mathsf{TD}(\rho,\tau)$ is defined as

$$\mathsf{TD}(\rho,\tau) := \frac{1}{2}\mathsf{Tr}\left(\sqrt{(\rho-\tau)^\dagger(\rho-\tau)}\right).$$

The trace distance between two states ρ and τ is an upper bound on the probability that any (unbounded) algorithm can distinguish ρ and τ.

Lemma 3.1 (Gentle measurement [Win99])**.** *Let ρ be a quantum state and let $(\Pi, \mathbb{I}-\Pi)$ be a projective measurement such that $\mathsf{Tr}(\Pi\rho) \geq 1-\delta$. Let*

$$\rho' = \frac{\Pi\rho\Pi}{\mathsf{Tr}(\Pi\rho)}$$

be the state after applying $(\Pi, \mathbb{I}-\Pi)$ to ρ and post-selecting on obtaining the first outcome. Then, $\mathsf{TD}(\rho,\rho') \leq 2\sqrt{\delta}$.

We will also often make use of the following simple claim.

Claim 3.2. *Consider a register $\mathcal{R}$ on n qubits and a distribution $\mathcal{F}$ over classical functions $f : \{0,1\}^n \to \{0,1\}$. For any such f, let Π_f be the projection onto x such that $f(x) = 1$. Then for any $|\psi\rangle$ on register $\mathcal{R}$,*

$$\mathop{\mathbb{E}}_{f\leftarrow\mathcal{F}}\left[\left\|\Pi_f|\psi\rangle\right\|^2\right] \leq \max_x\left\{\Pr_{f\leftarrow\mathcal{F}}[f(x)=1]\right\}.$$

Proof. For any $|\psi\rangle := \sum_x \alpha_x|x\rangle$, write

$$\mathop{\mathbb{E}}_{f\leftarrow\mathcal{F}}\left[\left\|\Pi_f|\psi\rangle\right\|^2\right] = \mathop{\mathbb{E}}_{f\leftarrow\mathcal{F}}\left[\sum_{x:f(x)=1}|\alpha_x|^2\right] = \sum_x \Pr_{f\leftarrow\mathcal{F}}[f(x)=1]\cdot|\alpha_x|^2 \leq \max_x\left\{\Pr_{f\leftarrow\mathcal{F}}[f(x)=1]\right\},$$

where the last inequality holds because $\{|\alpha_x|^2\}_x$ is a probability distribution. □

Finally, we define the notion of a pseudo-deterministic quantum ciruit.

Definition 3.3 (Pseudo-deterministic quantum circuit)**.** *A family of psuedo-deterministic quantum circuits $\{Q_\lambda\}_{\lambda\in\mathbb{N}}$ is defined as follows. The circuit Q_λ takes as input a classical string $x \in \{0,1\}^{n(\lambda)}$ and outputs a bit $b \leftarrow Q_\lambda(x)$. The circuit is pseudo-deterministic if for every sequence of classical inputs $\{x_\lambda\}_{\lambda\in\mathbb{N}}$, there exists a sequence of outputs $\{b_\lambda\}_{\lambda\in\mathbb{N}}$ such that*

$$\Pr[Q_\lambda(x_\lambda) \to b_\lambda] = 1 - \mathrm{negl}(\lambda).$$

We will often leave the dependence on λ implicit, and just refer to pseudo-deterministic circuits Q with input x. In a slight abuse of notation, we will denote by $Q(x)$ the bit b such that $\Pr[Q(x) \to b] = 1 - \mathrm{negl}(\lambda)$.

3.2 Obfuscation

Definition 3.4 (Virtual black-box obfuscation)**.** *A virtual black-box (VBB) obfuscator for a family of pseudo-deterministic quantum (resp. classical) circuits is a pair of QPT algorithms* $(\mathsf{Obf}, \mathsf{Eval})$ *with the following syntax.*

- $\mathsf{Obf}(1^\lambda, Q) \to \widetilde{Q}$*:* Obf *takes as input the security parameter* 1^λ *and the description of a quantum (resp. classical) circuit* Q*, and outputs a (potentially quantum) obfuscated circuit* $\widetilde{Q}$*.*
- $\mathsf{Eval}(\widetilde{Q}, x) \to b$*:* Eval *takes as input an obfuscated circuit* $\widetilde{Q}$ *and an input* x*, and outputs a bit* $b \in \{0,1\}$*.*

A VBB obfuscator should satisfy the following properties for any pseudo-deterministic (resp. classical) family of circuits $Q = \{Q_\lambda\}_{\lambda\in\mathbb{N}}$ *with input length* $n = n(\lambda)$*.*

- ***Correctness****: It holds with probability* $1 - \mathrm{negl}(\lambda)$ *over* $\widetilde{Q} \leftarrow \mathsf{Obf}(1^\lambda, Q)$ *that for all* $x \in \{0,1\}^n$*,* $\Pr[\mathsf{Eval}(\widetilde{Q}, x) \to Q(x)] = 1 - \mathrm{negl}(\lambda)$.
- ***Security****: For any QPT adversary* $\{\mathsf{A}_\lambda\}_{\lambda\in\mathbb{N}}$*, there exists a QPT simulator* $\{\mathsf{S}_\lambda\}_{\lambda\in\mathbb{N}}$ *such that*

$$\left| \Pr\left[1 \leftarrow \mathsf{A}_\lambda\left(\mathsf{Obf}(1^\lambda, Q)\right)\right] - \Pr\left[1 \leftarrow \mathsf{S}_\lambda^{O[Q]}\right] \right| = \mathrm{negl}(\lambda),$$

where $O[Q]$ *is the oracle that computes the map* $x \to Q(x)$*.*

Definition 3.5 (Indistinguishability obfuscation)**.** *An indistinguishability obfuscator (iO) for a family of pseudo-deterministic (resp. classical) circuits is a pair of QPT algorithms* $(\mathsf{Obf}, \mathsf{Eval})$ *that has the same syntax and correctness properties as a VBB obfuscator and satisfies the following security property. For any QPT adversary* $\{\mathsf{A}_\lambda\}_{\lambda\in\mathbb{N}}$ *and pair of functionally equivalent families of pseudo-deterministic (resp. classical) circuits* $Q_0 = \{Q_{0,\lambda}\}_{\lambda\in\mathbb{N}}, Q_1 = \{Q_{1,\lambda}\}_{\lambda\in\mathbb{N}}$*,*

$$\left| \Pr\left[1 \leftarrow \mathsf{A}_\lambda\left(\mathsf{Obf}(1^\lambda, Q_0)\right)\right] - \Pr\left[1 \leftarrow \mathsf{A}_\lambda\left(\mathsf{Obf}(1^\lambda, Q_1)\right)\right] \right| = \mathrm{negl}(\lambda).$$

3.3 Dual-mode randomized trapdoor claw-free hash functions

Definition 3.6. *Let* $\{X_\lambda\}_{\lambda\in\mathbb{N}}$ *and* $\{Y_\lambda\}_{\lambda\in\mathbb{N}}$ *be families of finite sets. Below, we will leave the dependence of these sets on* λ *implicit. A* dual-mode randomized trapdoor claw-free hash function *is described by a tuple of algorithms* $(\mathsf{Gen}, \mathsf{Eval}, \mathsf{Invert}, \mathsf{Check}, \mathsf{IsValid})$ *with the following syntax.*

- $\mathsf{Gen}(1^\lambda, h) \to (\mathsf{pk}, \mathsf{sk})$ *is a randomized classical algorithm that takes as input a security parameter* 1^λ *and a bit* $h \in \{0,1\}$ *(where* $h = 0$ *indicates* injective mode *and* $h = 1$ *indicates* 2-to-1 mode*), and outputs a public key* pk *and a secret key* sk*. The public key* pk *implicitly defines a function* $f_{\mathsf{pk}} : \{0,1\} \times X \to \mathcal{D}_Y$*, where* $\mathcal{D}_Y$ *is the set of probability distributions over* Y*.*
- $\mathsf{Eval}(\mathsf{pk}, b) \to |\psi_{\mathsf{pk},b}\rangle$ *is a QPT algorithm that takes as input a public key* pk *and a bit* b*, and outputs a fixed pure state* $|\psi_{\mathsf{pk},b}\rangle^{\mathcal{X},\mathcal{Y}}$ *on two registers* $\mathcal{X}$ *and* $\mathcal{Y}$*, where* $\mathcal{X}$ *is spanned by the elements of* X *and* $\mathcal{Y}$ *is spanned by the elements of* Y*. We then define*

$$\mathsf{Eval}[\mathsf{pk}] \coloneqq |0\rangle\langle 0|^{\mathcal{B}} \otimes \mathsf{Eval}(\mathsf{pk}, 0) + |1\rangle\langle 1|^{\mathcal{B}} \otimes \mathsf{Eval}(\mathsf{pk}, 1),$$

which is a map from the single qubit register $\mathcal{B}$ *to registers* $(\mathcal{B}, \mathcal{X}, \mathcal{Y})$*.*

- $\mathsf{Invert}(h, \mathsf{sk}, y)$ *is a deterministic classical algorithm that takes as input* $h \in \{0,1\}$, *a secret key* sk, *and an element* $y \in Y$. *If* $h = 0$, *it outputs a pair* $(b, x) \in \{0,1\} \times X$ *or* $\perp$. *If* $h = 1$, *it outputs two pairs* $(0, x_0)$ *and* $(1, x_1)$ *with* $x_0, x_1 \in X$, *or* $\perp$.
- $\mathsf{Check}(\mathsf{pk}, b, x, y) \to \{\top, \perp\}$ *is a deterministic classical algorithm that takes as input a public key* pk, *a bit* $b \in \{0,1\}$, *an element* $x \in X$, *and an element* $y \in Y$, *and outputs either* $\top$ *or* $\perp$.
- $\mathsf{IsValid}(x_0, x_1, d) \to \{\top, \perp\}$ *is a deterministic classical algorithm that takes as input two elements* $x_0, x_1 \in X$ *and a string* d, *and outputs either* $\top, \perp$, *characterizing membership in a set that we call*

$$\mathsf{Valid}_{x_0,x_1} := \{d : \mathsf{IsValid}(x_0, x_1, d) = 1\}.$$

We require that the following properties are satisfied.

1. **Correctness:**

   (a) For all $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{Gen}(1^\lambda, 0)$: *For every* $b \in \{0,1\}$, *every* $x \in X$, *and every* $y \in \mathsf{Supp}(f_{\mathsf{pk}}(b, x))$,
   $$\mathsf{Invert}(0, \mathsf{sk}, y) = (b, x).$$
   (b) For all $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{Gen}(1^\lambda, 1)$: *For every* $b \in \{0,1\}$, *every* $x \in X$, *and every* $y \in \mathsf{Supp}(f_{\mathsf{pk}}(b, x))$,
   $$\mathsf{Invert}(1, \mathsf{sk}, y) = ((0, x_0), (1, x_1))$$
   such that $x_b = x$, $y \in \mathsf{Supp}(f_{\mathsf{pk}}(0, x_0))$, *and* $y \in \mathsf{Supp}(f_{\mathsf{pk}}(1, x_1))$.

   (c) For all $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{Gen}(1^\lambda, 0) \cup \mathsf{Gen}(1^\lambda, 1)$, *every* $b \in \{0,1\}$ *and every* $x \in X$, *it holds that* $\mathsf{Check}(\mathsf{pk}, (b, x), y) = 1$ *if and only if* $y \in \mathsf{Supp}(f_{\mathsf{pk}}(b, x))$.

   (d) For all $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{Gen}(1^\lambda, 0) \cup \mathsf{Gen}(1^\lambda, 1)$ *and every* $b \in \{0,1\}$, *it holds that*
   $$\mathsf{TD}\left(|\psi_{\mathsf{pk},b}\rangle^{\mathcal{X},\mathcal{Y}}, \frac{1}{\sqrt{|X|}} \sum_{x \in X, y \in Y} \sqrt{(f_{\mathsf{pk}}(b, x))(y)}\, |x\rangle^{\mathcal{X}} |y\rangle^{\mathcal{Y}} \right) = \mathrm{negl}(\lambda),$$
   where $|\psi_{\mathsf{pk},b}\rangle \leftarrow \mathsf{Eval}(\mathsf{pk}, b)$.

   (e) For all $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{Gen}(1^\lambda, 1)$ *and every pair of elements* $x_0, x_1 \in X$, *the density of* Valid_{x_0,x_1} *is* $1 - \mathrm{negl}(\lambda)$.

2. **Key indistinguishability:** *For every QPT adversary* $\{\mathsf{A}_\lambda\}_{\lambda \in \mathbb{N}}$,
   $$\left| \Pr\left[1 \leftarrow \mathsf{A}_\lambda(\mathsf{pk}) : (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda, 0)\right] - \Pr\left[1 \leftarrow \mathsf{A}_\lambda(\mathsf{pk}) : (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda, 1)\right] \right| = \mathrm{negl}(\lambda).$$

3. **Adaptive hardcore bit:** *There is an efficiently computable and efficiently invertible injection* $J : X \to \{0,1\}^w$ *such that for every QPT adversary* $\{\mathsf{A}_\lambda\}_{\lambda \in \mathbb{N}}$,
   $$\left| \Pr\left[\begin{array}{lcr} \mathsf{Check}(\mathsf{pk}, b, x, y) = 1 \ \wedge & & (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda, 1) \\ d \in \mathsf{Valid}_{x_0,x_1} \ \wedge & : & (y, b, x, d) \leftarrow \mathsf{A}_\lambda(\mathsf{pk}) \\ d \cdot (J(x_0) \oplus J(x_1)) = 0 & & ((0, x_0), (1, x_1)) := \mathsf{Invert}(1, \mathsf{sk}, y) \end{array} \right] \right.$$
   $$\left. - \Pr\left[\begin{array}{lcr} \mathsf{Check}(\mathsf{pk}, b, x, y) = 1 \ \wedge & & (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda, 1) \\ d \in \mathsf{Valid}_{x_0,x_1} \ \wedge & : & (y, b, x, d) \leftarrow \mathsf{A}_\lambda(\mathsf{pk}) \\ d \cdot (J(x_0) \oplus J(x_1)) = 1 & & ((0, x_0), (1, x_1)) := \mathsf{Invert}(1, \mathsf{sk}, y) \end{array} \right] \right| = \mathrm{negl}(\lambda).$$

The works of [BCM+21, Mah22] showed that, assuming QLWE, there exists a dual-mode randomized trapdoor claw-free hash function.

3.4 Quantum fully-homomorphic encryption

We define quantum fully-homomorphic encryption (QFHE) with classical keys and classical encryption of classical messages. One could also define encryption for quantum states and decryption for quantum ciphertexts, but we will not need that in this work.

Definition 3.7 (Quantum fully-homomorphic encryption). *A quantum fully-homomorphic encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$ *consists of the following efficient algorithms.*

- $\mathsf{Gen}(1^\lambda, D) \to (\mathsf{pk}, \mathsf{sk})$*: On input the security parameter* 1^λ *and a circuit depth* D*, the key generation algorithm returns a public key* pk *and a secret key* sk*.*
- $\mathsf{Enc}(\mathsf{pk}, x) \to \mathsf{ct}$*: On input the public key* pk *and a classical plaintext* x*, the encryption algorithm returns a classical ciphertext* ct*.*
- $\mathsf{Eval}(Q, \mathsf{ct}) \to \widetilde{\mathsf{ct}}$*: On input a quantum circuit* Q *and a ciphertext* ct*, the quantum evaluation algorithm returns an evaluated ciphertext* $\widetilde{\mathsf{ct}}$*.*
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \to x$*: On input the secret key* sk *and a classical ciphertext* ct*, the decryption algorithm returns a message* x*.*

The scheme should satisfy the standard notion of semantic security.

Definition 3.8 (Semantic security). *A QFHE scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$ *is secure if for any QPT adversary* $\{\mathsf{A}_\lambda\}_{\lambda \in \mathbb{N}}$ *and circuit depth* D*,*

$$\left| \Pr\left[\mathsf{A}_\lambda(\mathsf{ct}) = 1 : \begin{array}{r} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda, D) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, 0) \end{array}\right] - \Pr\left[\mathsf{A}_\lambda(\mathsf{ct}) = 1 : \begin{array}{r} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda, D) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, 1) \end{array}\right] \right| = \mathrm{negl}(\lambda).$$

We will also require the following notion of correctness for evaluation of pseudo-deterministic quantum circuits.

Definition 3.9 (Evaluation Correctness). *A QFHE scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$ *is correct if for any polynomial* $D(\lambda)$*, family of pseudo-deterministic quantum circuits* $\{Q_\lambda\}_{\lambda \in \mathbb{N}}$ *of depth* $D(\lambda)$*, inputs* $\{x_\lambda\}_{\lambda \in \mathbb{N}}$*, security parameter* λ*,* $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{Gen}(1^\lambda, D(\lambda))$*, and* $\mathsf{ct} \in \mathsf{Enc}(\mathsf{pk}, x)$*,*

$$\Pr[\mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(Q_\lambda, \mathsf{ct})) = Q_\lambda(x_\lambda)] = 1 - \mathrm{negl}(\lambda).$$

The works of Mahadev [Mah18] and Brakerski [Bra18] show that such a QFHE scheme can be constructed from QLWE.

3.5 Measure and re-program

Imported Theorem 3.10 (Measure and re-program [DFMS19, DFM20]). [18] *Let* A, B *be finite non-empty sets, and let* $q \in \mathbb{N}$*. Let* A *be an oracle-aided quantum circuit that makes* q *queries to a uniformly*

[18]This theorem was stated more generally in [DFMS19, DFM20] to consider the drop in expectation for each specific $a^* \in A$, and also to consider a more general class of quantum predicates.

random function $H : A \to B$ and then outputs classical strings (a, z) where $a \in A$. There exists a two-stage quantum circuit $\mathsf{Sim}[\mathsf{A}]$ *such that for any predicate V, it holds that*

$$\Pr\left[V(a,b,z) = 1 : \begin{array}{r} (a, \mathsf{state}) \leftarrow \mathsf{Sim}[\mathsf{A}] \\ b \leftarrow B \\ z \leftarrow \mathsf{Sim}[\mathsf{A}](b, \mathsf{state}) \end{array}\right] \geq \frac{\Pr\left[V(a, H(a), z) = 1 : (a,z) \leftarrow \mathsf{A}^H\right]}{(2q+1)^2}.$$

Moreover, $\mathsf{Sim}[\mathsf{A}]$ *operates as follows.*

- *Sample $H : A \to B$ as a $2q$-wise independent function and $(i, d) \leftarrow (\{0, \ldots, q-1\} \times \{0,1\}) \cup \{(q, 0)\}$.*
- *Run* A *until it has made i oracle queries, answering each query using H.*
- *When* A *is about to make its $(i+1)$'th oracle query, measure its query registers in the standard basis to obtain a. In the special case that $(i, d) = (q, 0)$, the simulator measures (part of) the final output register of* A *to obtain a.*
- *The simulator receives $b \leftarrow B$.*
- *If $d = 0$, answer* A*'s $(i+1)$'th query using H, and if $d = 1$, answer* A*'s $(i+1)$'th query using $H[a \to b]$, which is the function H except that $H(a)$ is re-programmed to b.*
- *Run* A *until it has made all q oracle queries. For queries $i+2$ through q, answer using $H[a \to b]$.*
- *Measure* A*'s output z.*

Note that the running time of $\mathsf{Sim}[\mathsf{A}]$ *is at most* $\mathrm{poly}(q, \log|A|, \log|B|)$ *times the running time of* A.

3.6 Signature tokens

A signature token scheme consists of algorithms $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ with the following syntax.

- $\mathsf{Gen}(1^\lambda) \to (\mathsf{vk}, |\mathsf{sk}\rangle)$: The Gen algorithm takes as input the security parameter 1^λ and outputs a classical verification key vk and a quantum signing key $|\mathsf{sk}\rangle$.
- $\mathsf{Sign}(b, |\mathsf{sk}\rangle) \to \sigma$: The Sign algorithm takes as input a bit $b \in \{0,1\}$ and the signing key $|\mathsf{sk}\rangle$, and outputs a signature σ.
- $\mathsf{Verify}(\mathsf{vk}, b, \sigma) \to \{\top, \bot\}$: The Verify algorithm takes as input a verification key vk, a bit b, and a signature σ, and outputs $\top$ or $\bot$.

A signature token should satisfy the following definition of correctness.

Definition 3.11. *A signature token scheme* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ *is* correct *if for any $b \in \{0,1\}$,*

$$\Pr\left[\mathsf{Verify}(\mathsf{vk}, b, \sigma) = \top : \begin{array}{r} (\mathsf{vk}, |\mathsf{sk}\rangle) \leftarrow \mathsf{Gen}(1^\lambda) \\ \sigma \leftarrow \mathsf{Sign}(b, |\mathsf{sk}\rangle) \end{array}\right] = 1 - \mathrm{negl}(\lambda).$$

Next, we define notions of unforgeability. In this paper, it suffices to consider security in the *oracle model*, where the adversarial signer has oracle access to the verification function, rather than to the description of the verification key vk itself.

Definition 3.12. *A signature token scheme* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ *satisfies* unforgeability *if for any oracle-aided adversary* $\{\mathsf{A}_\lambda\}_{\lambda \in \mathbb{N}}$ *that makes at most* $\mathrm{poly}(\lambda)$ *oracle queries,*

$$\Pr\left[\begin{array}{l}\mathsf{Verify}(\mathsf{vk}, 0, \sigma_0) = \top \ \wedge \\ \mathsf{Verify}(\mathsf{vk}, 1, \sigma_1) = \top\end{array} : \begin{array}{c}(\mathsf{vk}, |\mathsf{sk}\rangle) \leftarrow \mathsf{Gen}(1^\lambda) \\ (\sigma_0, \sigma_1) \leftarrow \mathsf{A}_\lambda^{\mathsf{Verify}[\mathsf{vk}]}(|\mathsf{sk}\rangle)\end{array}\right] = \mathrm{negl}(\lambda),$$

where $\mathsf{Verify}[\mathsf{vk}]$ *is the functionality* $\mathsf{Verify}(\mathsf{vk}, \cdot, \cdot)$.

Imported Theorem 3.13 ([BS16]). *There exists a signature token scheme in the oracle model that satisfies unforgeability.*

We will also require a signature token with the property of *strong unforgeability*, defined as follows.

Definition 3.14. *A signature token scheme* $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *satisfies* strong unforgeability *if for any oracle-aided adversary* $\{\mathsf{A}_\lambda\}_{\lambda \in \mathbb{N}}$ *that makes at most* $\mathrm{poly}(\lambda)$ *oracle queries,*

$$\Pr\left[\begin{array}{l}(b_0, \sigma_0) \neq (b_1, \sigma_1) \ \wedge \\ \mathsf{Verify}(\mathsf{vk}, b_0, \sigma_0) = \top \ \wedge \\ \mathsf{Verify}(\mathsf{vk}, b_1, \sigma_1) = \top\end{array} : \begin{array}{c}(\mathsf{vk}, |\mathsf{sk}\rangle) \leftarrow \mathsf{Gen}(1^\lambda) \\ (b_0, \sigma_0, b_1, \sigma_1) \leftarrow \mathsf{A}_\lambda^{\mathsf{Verify}[\mathsf{vk}]}(|\mathsf{sk}\rangle)\end{array}\right] = \mathrm{negl}(\lambda),$$

where $\mathsf{Verify}[\mathsf{vk}]$ *is the functionality* $\mathsf{Verify}(\mathsf{vk}, \cdot, \cdot)$.

Claim 3.15. *There exists a signature token scheme in the oracle model that satisfies strong unforgeability.*

Proof. This follows by a slight tweak to arguments in [BS16]. We first note that by a union bound, it suffices to show that each of the following three cases happens with negligible probability: (1) A_λ outputs σ_0, σ_1 such that σ_0 is a valid signature of 0 and σ_1 is a valid signature of 1, (2) A_λ outputs $\sigma_0 \neq \sigma_0'$ that are both valid signatures of 0, and (3) A_λ outputs $\sigma_1 \neq \sigma_1'$ that are both valid signatures of 1. The first case is already proven by [BS16].

The second case can be shown by following the proofs in [BS16] except for one difference: for a subspace $A < \mathbb{F}_2^n$, the "target set" $\Lambda(A)$ (defined on page 25 of [BS16]) is instead defined to consist of pairs of vectors (a, b) such that $a \neq b \in A \setminus \{0^n\}$. The only change in the proof then comes in [BS16, Lemma 19], where we need to show that

$$\max_{A \in S(n), (a,b) \in \Lambda(A)} \Pr_{B \leftarrow \mathcal{R}_A}[(a, b) \in \Lambda(B)] \leq \frac{1}{4},$$

where $S(n)$ is the set of subspaces of $\mathbb{F}_2^n$ of dimension $n/2$, and for any $A \in S(n)$, $\mathcal{R}_A$ is the set of $B \in S(n)$ such that $\dim(A \cap B) = n/2 - 1$. This follows by first noting that any distinct non-zero $a, b \in A$ specify a two-dimensional subspace $\{0, a, b, a + b\}$. Then, following the proof of [BS16, Lemma 19], and defining

$$G(m, k) := \prod_{i=0}^{k-1} \frac{2^{m-i} - 1}{2^{k-i} - 1}$$

to be the number of subspaces of $\mathbb{F}_2^k$ of dimension m, we have that this expression is at most

$$\frac{G(n/2 - 2, n/2 - 3)}{G(n/2, n/2 - 1)} = \frac{2^{n/2-1} - 1}{2^{n/2} - 1} \cdot \frac{2^{n/2-2} - 1}{2^{n/2-1} - 1} \leq \frac{1}{4}.$$

Finally, the third case can be proven in the same way as the second, by defining $\Lambda(A)$ as the set of (a, b) such that $a \neq b \in A^\perp \setminus \{0^n\}$. $\square$

Remark 3.16. *It is straightforward to extend any single-bit signature token scheme (which is described above) to a multi-bit scheme for polynomial-size messages, by signing each bit with a different invocation of the single-bit scheme.*

4 Pauli Functional Commitments

4.1 Definition

A Pauli functional commitment resembles a standard bit commitment scheme with a classical receiver. However, when used to commit to a qubit $|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$ in superposition, it supports the ability to open to either a *standard* or *Hadamard* basis measurement of $|\psi\rangle$. A Pauli functional commitment should also satisfy some notion of binding to a classical bit.

The syntax of a Pauli functional commitment is given below. We present the syntax in the *oracle model*, where the committer obtains access to an efficient classical oracle CK as part of its commitment key. Such a scheme can be heuristically instantiated in the plain model by using a post-quantum indistinguishability obfuscator to obfuscate this oracle. We also specify that the remainder of the commitment key is a quantum state $|\mathsf{ck}\rangle$, but note that this is not inherent to the definition of a Pauli functional commitment.

Definition 4.1 (Pauli functional commitment: Syntax)**.** *A Pauli functional commitment consists of six algorithms* $(\mathsf{Gen}, \mathsf{Com}, \mathsf{OpenZ}, \mathsf{OpenX}, \mathsf{DecZ}, \mathsf{DecX})$ *with the following syntax.*

- $\mathsf{Gen}(1^\lambda) \to (\mathsf{dk}, |\mathsf{ck}\rangle, \mathsf{CK})$ *is a QPT algorithm that takes as input the security parameter* 1^λ *and outputs a classical decoding key* dk *and a quantum commitment key* $(|\mathsf{ck}\rangle, \mathsf{CK})$*, where* $|\mathsf{ck}\rangle$ *is a quantum state on register* $\mathcal{K}$*, and* CK *is the description of a classical deterministic polynomial-time functionality* $\mathsf{CK} : \{0,1\}^* \to \{0,1\}^*$.

- $\mathsf{Com}_b^{\mathsf{CK}}(|\mathsf{ck}\rangle) \to (\mathcal{U}, c)$ *is a QPT algorithm that is parameterized by a bit b and has oracle access to* CK*. It applies a map from register $\mathcal{K}$ (initially holding the commitment key $|\mathsf{ck}\rangle$) to registers $(\mathcal{U}, \mathcal{C})$ and then measures $\mathcal{C}$ in the standard basis to obtain a classical string $c \in \{0,1\}^*$ and a left-over state on register $\mathcal{U}$. We then write*

  $$\mathsf{Com}^{\mathsf{CK}} := |0\rangle\langle 0| \otimes \mathsf{Com}_0^{\mathsf{CK}} + |1\rangle\langle 1| \otimes \mathsf{Com}_1^{\mathsf{CK}}$$

  to refer to the map that applies the $\mathsf{Com}_b^{\mathsf{CK}}$ *map classically controlled on a single-qubit register $\mathcal{B}$ to produce a state on registers $(\mathcal{B}, \mathcal{U}, \mathcal{C})$, and then measures $\mathcal{C}$ in the standard basis to obtain a classical string c along with a left-over quantum state on registers $(\mathcal{B}, \mathcal{U})$.*

- $\mathsf{OpenZ}(\mathcal{B}, \mathcal{U}) \to u$ *is a QPT measurement on registers $(\mathcal{B}, \mathcal{U})$ that outputs a classical string u.*

- $\mathsf{OpenX}(\mathcal{B}, \mathcal{U}) \to u$ *is a QPT measurement on registers $(\mathcal{B}, \mathcal{U})$ that outputs a classical string u.*

- $\mathsf{DecZ}(\mathsf{dk}, c, u) \to \{0, 1, \bot\}$ *is a classical deterministic polynomial-time algorithm that takes as input the decoding key* dk*, a string c, and a string u, and outputs either a bit b or a $\bot$ symbol.*

- $\mathsf{DecX}(\mathsf{dk}, c, u) \to \{0, 1, \bot\}$ *is a classical deterministic polynomial-time algorithm that takes as input a the decoding key* dk*, a string c, and a string u, and outputs either a bit b or a $\bot$ symbol.*

Definition 4.2 (Pauli functional Commitment: Correctness). *A Pauli functional commitment* $(\mathsf{Gen}, \mathsf{Com}, \mathsf{OpenZ}, \mathsf{OpenX}, \mathsf{DecZ}, \mathsf{DecX})$ *is* correct *if for any single-qubit (potentially mixed) state on register $\mathcal{B}$, it holds that*

$$\mathsf{TV}\left(\mathsf{Z}(\mathcal{B}), \mathsf{PFCZ}(1^\lambda, \mathcal{B})\right) = \mathrm{negl}(\lambda), \ \ \textit{and} \ \ \mathsf{TV}\left(\mathsf{X}(\mathcal{B}), \mathsf{PFCX}(1^\lambda, \mathcal{B})\right) = \mathrm{negl}(\lambda),$$

where the distributions are defined as follows.

- $\mathsf{Z}(\mathcal{B})$ *measures $\mathcal{B}$ in the standard basis.*
- $\mathsf{X}(\mathcal{B})$ *measures $\mathcal{B}$ in the Hadamard basis.*
- $\mathsf{PFCZ}(1^\lambda, \mathcal{B})$ *samples* $(\mathsf{dk}, |\mathsf{ck}\rangle, \mathsf{CK}) \leftarrow \mathsf{Gen}(1^\lambda), (\mathcal{B}, \mathcal{U}, c) \leftarrow \mathsf{Com}^{\mathsf{CK}}(\mathcal{B}, |\mathsf{ck}\rangle), u \leftarrow \mathsf{OpenZ}(\mathcal{B}, \mathcal{U})$, *and outputs* $\mathsf{DecZ}(\mathsf{dk}, c, u)$.
- $\mathsf{PFCX}(1^\lambda, \mathcal{B})$ *samples* $(\mathsf{dk}, |\mathsf{ck}\rangle, \mathsf{CK}) \leftarrow \mathsf{Gen}(1^\lambda), (\mathcal{B}, \mathcal{U}, c) \leftarrow \mathsf{Com}^{\mathsf{CK}}(\mathcal{B}, |\mathsf{ck}\rangle), u \leftarrow \mathsf{OpenX}(\mathcal{B}, \mathcal{U})$, *and outputs* $\mathsf{DecX}(\mathsf{dk}, c, u)$.

A Pauli functional commitment that satisfies *binding with public decodability* allows the adversarial Committer to have oracle access to the receiver's decoding functionalities $\mathsf{DecZ}(\mathsf{dk}, \cdot, \cdot)$ and $\mathsf{DecX}(\mathsf{dk}, \cdot, \cdot)$. However, we crucially do not give the adversarial *Opener* access to $\mathsf{DecX}(\mathsf{dk}, \cdot, \cdot)$.

Definition 4.3 (Pauli functional commitment: Single-bit binding with public decodability). *A Pauli functional commitment* $(\mathsf{Gen}, \mathsf{Com}, \mathsf{OpenZ}, \mathsf{OpenX}, \mathsf{DecZ}, \mathsf{DecX})$ *satisfies* single-bit binding with public decodability *if the following holds. Given* dk, c, *and* $b \in \{0,1\}$, *let*

$$\Pi_{\mathsf{dk},c,b} := \sum_{u:\mathsf{DecZ}(\mathsf{dk},c,u)=b} |u\rangle\langle u|.$$

Consider any adversary $\{(\mathsf{C}_\lambda, \mathsf{U}_\lambda)\}_{\lambda\in\mathbb{N}}$, *where each C_λ is an oracle-aided quantum operation, each U_λ is an oracle-aided unitary, and each $(\mathsf{C}_\lambda, \mathsf{U}_\lambda)$ make at most* $\mathrm{poly}(\lambda)$ *oracle queries. Then for any $b \in \{0,1\}$,*

$$\mathbb{E}\left[\left\|\Pi_{\mathsf{dk},c,1-b}\mathsf{U}_\lambda^{\mathsf{CK},\mathsf{DecZ}[\mathsf{dk}]}\Pi_{\mathsf{dk},c,b}|\psi\rangle\right\| : (|\psi\rangle, c) \leftarrow \mathsf{C}_\lambda^{\mathsf{CK},\mathsf{DecZ}[\mathsf{dk}],\mathsf{DecX}[\mathsf{dk}]}(|\mathsf{ck}\rangle)\right] = \mathrm{negl}(\lambda),$$

where the expectation is over $\mathsf{dk}, |\mathsf{ck}\rangle, \mathsf{CK} \leftarrow \mathsf{Gen}(1^\lambda)$. *Here,* $\mathsf{DecZ}[\mathsf{dk}]$ *is the oracle implementing the classical functionality* $\mathsf{DecZ}(\mathsf{dk}, \cdot, \cdot)$ *and* $\mathsf{DecX}[\mathsf{dk}]$ *is the oracle implementing the classical functionality* $\mathsf{DecX}(\mathsf{dk}, \cdot, \cdot)$.

Next, we extend the above single-bit binding property to a notion of *string binding*.

Definition 4.4 (Pauli functional commitment: String binding with public decodability). *A Pauli functional commitment* $(\mathsf{Gen}, \mathsf{Com}, \mathsf{OpenZ}, \mathsf{OpenX}, \mathsf{DecZ}, \mathsf{DecX})$ *satisfies* string binding with public decodability *if the following holds for any polynomial $m = m(\lambda)$ and two disjoint sets $W_0, W_1 \subset \{0,1\}^m$ of m-bit strings. Given a set of m verification keys $\mathbf{dk} = (\mathsf{dk}_1, \ldots, \mathsf{dk}_m)$, m strings $\mathbf{c} = (c_1, \ldots, c_m)$, and $b \in \{0,1\}$, define*

$$\Pi_{\mathbf{dk},\mathbf{c},W_b} := \sum_{w\in W_b}\left(\bigotimes_{i\in[m]}\Pi_{\mathsf{dk}_i,c_i,w_i}\right).$$

Consider any adversary $\{(\mathsf{C}_\lambda, \mathsf{U}_\lambda)\}_{\lambda\in\mathbb{N}}$*, where each* C_λ *is an oracle-aided quantum operation, each* U_λ *is an oracle-aided unitary, and each* $(\mathsf{C}_\lambda, \mathsf{U}_\lambda)$ *make at most* $\mathrm{poly}(\lambda)$ *oracle queries. Then,*

$$\mathbb{E}\left[\left\|\Pi_{\mathbf{dk},\mathbf{c},W_1}\mathsf{U}_\lambda^{\mathbf{CK},\mathsf{DecZ}[\mathbf{dk}]}\Pi_{\mathbf{dk},\mathbf{c},W_0}\left|\psi\right\rangle\right\| : (\left|\psi\right\rangle, \mathbf{c}) \leftarrow \mathsf{C}_\lambda^{\mathbf{CK},\mathsf{DecZ}[\mathbf{dk}],\mathsf{DecX}[\mathbf{dk}]}(\left|\mathbf{ck}\right\rangle)\right] = \mathrm{negl}(\lambda),$$

where the expectation is over $\{\mathsf{dk}_i, \left|\mathsf{ck}_i\right\rangle, \mathsf{CK}_i \leftarrow \mathsf{Gen}(1^\lambda)\}_{i\in[m]}$*. Here,* $\left|\mathbf{ck}\right\rangle = (\left|\mathsf{ck}_1\right\rangle, \ldots, \left|\mathsf{ck}_m\right\rangle)$*,* $\mathbf{CK}$ *is the collection of oracles* $\mathsf{CK}_1, \ldots, \mathsf{CK}_m$*,* $\mathsf{DecZ}[\mathbf{dk}]$ *is the collection of oracles* $\mathsf{DecZ}[\mathsf{dk}_1], \ldots, \mathsf{DecZ}[\mathsf{dk}_m]$*, and* $\mathsf{DecX}[\mathbf{dk}]$ *is the collection of oracles* $\mathsf{DecX}[\mathsf{dk}_1], \ldots, \mathsf{DecX}[\mathsf{dk}_m]$*.*

We prove the following lemma in Appendix A.

Lemma 4.5. *Any Pauli functional commitment that satisfies* single-bit binding with public decodability *also satisfies* string binding with public decodability.

4.2 Construction

Before describing our construction, we introduce some notation.

- A subspace $S < \mathbb{F}_2^n$ is *balanced* if half of its vectors start with 0 and the other half start with 1. Note that S is balanced if and only if at least one of its basis vectors starts with 1. Thus, a random large enough (say $n/2$-dimensional) subspace is balanced with probability $1 - \mathrm{negl}(n)$. By default, we will only consider balanced subspaces in what follows.
- For an affine subspace $A = S + v$ of $\mathbb{F}_2^n$, we write
$$\left|S + v\right\rangle := \frac{1}{\sqrt{|S|}}\sum_{s\in S}\left|s + v\right\rangle.$$
- Given an affine subspace $S + v$, let $(S + v)_0$ be the set of vectors in $S + v$ that start with 0 and let $(S + v)_1$ be the set of vectors in $S + v$ that start with 1.

We describe our construction of a Pauli functional commitment in Fig. 3.

Theorem 4.6. *The Pauli functional commitment described in Fig. 3 satisfies* correctness *(Definition 4.2).*

Proof. We will show correctness assuming that the signature token scheme Tok is perfectly correct. In reality, it may be statistically correct, but in this case we can still conclude that Fig. 3 satisfies correctness, which allows for a negligible statistical distance.

We will first show that the map applied by $\mathsf{Com}_b^{\mathsf{CK}}$ in the case that the measurement of the first qubit of $\mathcal{K}_0$ is $1 - b$ successfully takes $\left|(S + v)_{1-b}\right\rangle \to \left|(S + v)_b\right\rangle$. Since we are assuming perfect correctness from Tok, it suffices to show that for any balanced affine subspace $\left|S + v\right\rangle$,

$$H^{\otimes n}\mathsf{Ph}^{O[S^\perp]}H^{\otimes n}\left|(S + v)_{1-b}\right\rangle \to \left|(S + v)_b\right\rangle,$$

where $\mathsf{Ph}^{O[S^\perp]}$ is the map $\left|s\right\rangle \to (-1)^{O[S^\perp](s)}\left|s\right\rangle$, and $O[S^\perp]$ is the oracle that outputs 0 if $s \in S^\perp$ and 1 if $s \notin S^\perp$. This was actually shown in [AGKZ20], but we repeat it here for completeness.

Pauli Functional Commitment

Parameters: Polynomial $n = n(\lambda) \geq \lambda$.
Ingredients: Signature token scheme $(\mathsf{Tok.Gen}, \mathsf{Tok.Sign}, \mathsf{Tok.Verify})$ (Section 3.6).

- $\mathsf{Gen}(1^\lambda)$: Sample a uniformly random $n/2$-dimensional balanced affine subspace $S + v$ of $\mathbb{F}_2^n$ and sample $(\mathsf{vk}, |\mathsf{sk}\rangle) \leftarrow \mathsf{Tok.Gen}(1^\lambda)$. Set
$$\mathsf{dk} := (S, v, \mathsf{vk}), \quad |\mathsf{ck}\rangle := (|S+v\rangle, |\mathsf{sk}\rangle).$$
Define CK to take as input (σ, s) for $s \in \{0,1\}^n$ and output $\bot$ if $\mathsf{Tok.Verify}(\mathsf{vk}, 0, \sigma) = \bot$, and otherwise output 0 if $s \in S^\perp$ or 1 if $s \notin S^\perp$.
- $\mathsf{Com}_b^{\mathsf{CK}}(|\mathsf{ck}\rangle)$:
  - Parse $|\mathsf{ck}\rangle = (|S+v\rangle^{\mathcal{K}_0}, |\mathsf{sk}\rangle^{\mathcal{K}_1})$.
  - Coherently apply $\mathsf{Tok.Sign}(1^\lambda, 0, \cdot)$ from the $\mathcal{K}_1$ register to a fresh register $\mathcal{G}$, which will now hold a superposition over signatures σ on the bit 0.
  - Measure the first qubit of register $\mathcal{K}_0$ in the standard basis. If the result is b, the state on register $\mathcal{K}_0$ has collapsed to $|(S+v)_b\rangle$, and we continue. Otherwise, perform a rotation from $|(S+v)_{1-b}\rangle$ to $|(S+v)_b\rangle$ by applying the operation $(H^{\otimes n})^{\mathcal{K}_0} \mathsf{Ph}^{\mathsf{CK}(\cdot,\cdot)} (H^{\otimes n})^{\mathcal{K}_0}$ to registers $(\mathcal{K}_0, \mathcal{G})$, where $\mathsf{Ph}^{\mathsf{CK}(\cdot,\cdot)}$ is the map $|s\rangle^{\mathcal{K}_0} |\sigma\rangle^{\mathcal{G}} \rightarrow (-1)^{\mathsf{CK}(\sigma,s)} |s\rangle^{\mathcal{K}_0} |\sigma\rangle^{\mathcal{G}}$.
  - Next, reverse the $\mathsf{Tok.Sign}(1^\lambda, 0, \cdot)$ operation on $(\mathcal{K}_1, \mathcal{G})$ to recover $|\mathsf{sk}\rangle$ on register $\mathcal{K}_1$.
  - Finally, sample and output $c \leftarrow \mathsf{Tok.Sign}(1^\lambda, 1, |\mathsf{sk}\rangle)$, along with the final state on register $\mathcal{U} := \mathcal{K}_0$.
- $\mathsf{OpenZ}(\mathcal{B}, \mathcal{U})$: Measure all registers in the standard basis.
- $\mathsf{OpenX}(\mathcal{B}, \mathcal{U})$: Measure all registers in the Hadamard basis.
- $\mathsf{DecZ}(\mathsf{dk}, c, u)$:
  - Parse $\mathsf{dk} = (S, v, \mathsf{vk})$ and $u = (b, s)$, where $b \in \{0,1\}$ and $s \in \{0,1\}^n$.
  - Check that $\mathsf{Tok.Verify}(\mathsf{vk}, 1, c) = \top$, and if not output $\bot$.
  - If $s \in (S+v)_b$, output b, and otherwise output $\bot$.
- $\mathsf{DecX}(\mathsf{dk}, c, u)$:
  - Parse $\mathsf{dk} = (S, v, \mathsf{vk})$ and $u = (b', s)$, where $b' \in \{0,1\}$ and $s \in \{0,1\}^n$.
  - Check that $\mathsf{Tok.Verify}(\mathsf{vk}, 1, c) = \top$, and if not output $\bot$.
  - If $s \in S^\perp$, then define $r := 0$. If $s \oplus (1, 0, \ldots, 0) \in S^\perp$, then define $r := 1$. Otherwise, abort and output $\bot$. That is, r is set to 0 if $s \in S^\perp$ and to 1 if $s \in (S_0)^\perp \setminus S^\perp$. Then, output $b := b' \oplus r$.

Figure 3: A Pauli functional commitment that satisfies *binding with public decodability*.

We will use the facts that $S_1 = S_0 + w$ for some w, and that $(S+v)_0 = S_0 + v_0$ and $(S+v)_1 = S_0 + v_1$ for some v_0, v_1 such that $v_0 + v_1 = w$. Also note that for any $s \in S^\perp$, $s \cdot w = 0$, and for any $s \in (S_0)^\perp \setminus S^\perp$, $s \cdot w = 1$.

$$
\begin{aligned}
&H^{\otimes n}\mathsf{Ph}^{O[S^\perp]}H^{\otimes n}\left|(S+v)_{1-b}\right\rangle \\
&\quad = H^{\otimes n}\mathsf{Ph}^{O[S^\perp]}H^{\otimes n}\frac{1}{\sqrt{2^{n/2-1}}}\left(\sum_{s\in S_0}\left|s+v_{1-b}\right\rangle\right) \\
&\quad = H^{\otimes n}\mathsf{Ph}^{O[S^\perp]}\frac{1}{\sqrt{2^{n/2+1}}}\left(\sum_{s\in S_0^\perp}(-1)^{s\cdot v_{1-b}}\left|s\right\rangle\right) \\
&\quad = H^{\otimes n}\mathsf{Ph}^{O[S^\perp]}\frac{1}{\sqrt{2^{n/2+1}}}\left(\sum_{s\in S^\perp}(-1)^{s\cdot w+s\cdot v_b}\left|s\right\rangle+\sum_{s\in S_0^\perp\setminus S^\perp}(-1)^{s\cdot w+s\cdot v_b}\left|s\right\rangle\right) \\
&\quad = H^{\otimes n}\mathsf{Ph}^{O[S^\perp]}\frac{1}{\sqrt{2^{n/2+1}}}\left(\sum_{s\in S^\perp}(-1)^{s\cdot v_b}\left|s\right\rangle+\sum_{s\in S_0^\perp\setminus S^\perp}(-1)^{1+s\cdot v_b}\left|s\right\rangle\right) \\
&\quad = H^{\otimes n}\frac{1}{\sqrt{2^{n/2+1}}}\left(\sum_{s\in S^\perp}(-1)^{s\cdot v_b}\left|s\right\rangle+\sum_{s\in S_0^\perp\setminus S^\perp}(-1)^{s\cdot v_b}\left|s\right\rangle\right) \\
&\quad = H^{\otimes n}\frac{1}{\sqrt{2^{n/2+1}}}\left(\sum_{s\in S_0^\perp}(-1)^{s\cdot v_b}\left|s\right\rangle\right) \\
&\quad = \left|(S+v)_b\right\rangle.
\end{aligned}
$$

Thus, applying $\mathsf{Com}^{\mathsf{CK}}$ to a pure state $|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ and commitment key $|\mathsf{ck}\rangle$ produces (up to negligible trace distance) the state

$$
|\psi_{\mathsf{Com}}\rangle = \alpha_0|0\rangle\left|(S+v)_0\right\rangle + \alpha_1|1\rangle\left|(S+v)_1\right\rangle,
$$

and a signature c on the bit 1.

We continue by arguing that measuring and decoding $|\psi_{\mathsf{Com}}\rangle$ in the standard (resp. Hadamard) basis produces the same distribution as directly measuring $|\psi\rangle$ in the standard (resp. Hadamard) basis. As a mixed state is a probability distribution over pure states, this will complete the proof of correctness.

First, it is immediate that measuring $|\psi_{\mathsf{Com}}\rangle$ in the standard basis produces a bit b with probability $|\alpha_b|^2$ along with a vector s such that $s \in (S+v)_b$.

Next, note that applying Hadamard to each qubit of $|\psi_{\mathsf{Com}}\rangle$ except the first results in the state

$$
\alpha_0|0\rangle\left(\sum_{s\in S_0^\perp}(-1)^{s\cdot v_0}|s\rangle\right)+\alpha_1|1\rangle\left(\sum_{s\in S_0^\perp}(-1)^{s\cdot v_1}|s\rangle\right),
$$

and thus, measuring each of these qubits (except the first) in the Hadamard basis produces a vector s and a single-qubit state

$$
(-1)^{s\cdot v_0}\alpha_0|0\rangle + (-1)^{s\cdot v_1}\alpha_1|1\rangle = \alpha_0|0\rangle + (-1)^{s\cdot w}\alpha_1|1\rangle.
$$

So, measuring this qubit in the Hadamard basis is equivalent to measuring $|\psi\rangle$ in the Hadamard basis and masking the result with $s \cdot w$. Recalling that $s \cdot w = 0$ if $s \in S^\perp$ and $s \cdot w = 1$ if $s \in (S_0)^\perp \setminus S^\perp$ completes the proof of correctness. $\square$

4.3 Binding

This section is dedicated to proving the following theorem.

Theorem 4.7. *Assuming that* Tok *satisfies unforgeability (Definition 3.12), the Pauli functional commitment described in Fig. 3 with $n \geq 130\lambda$ satisfies* single-bit binding with public decodability *(Definition 4.3).*

The proof of this theorem will be identical for each choice of $b \in \{0,1\}$ in the statement of Definition 4.3. So, consider any adversary (C,U) attacking the publicly-decodable single-bit binding game for $b = 0$, where we drop the indexing by λ for notational convenience. We first show that it suffices to prove the following claim, in which U no longer has oracle access to CK.

Claim 4.8. *For any (C,U) where C and U each make $\mathrm{poly}(\lambda)$ many oracle queries, it holds that*

$$\Pr_{\mathsf{dk},|\mathsf{ck}\rangle,\mathsf{CK}\leftarrow\mathsf{Gen}(1^\lambda)}\left[\left\|\Pi_{\mathsf{dk},c,1}\mathsf{U}^{\mathsf{DecZ}[\mathsf{dk}]}\Pi_{\mathsf{dk},c,0}\,|\psi\rangle\right\|^2 \geq \frac{1}{2^\lambda} : (|\psi\rangle, c) \leftarrow \mathsf{C}^{\mathsf{CK},\mathsf{DecZ}[\mathsf{dk}],\mathsf{DecX}[\mathsf{dk}]}(|\mathsf{ck}\rangle)\right] = \mathrm{negl}(\lambda).$$

Lemma 4.9. *Claim 4.8 implies Theorem 4.7.*

Proof. First, we note that to prove Theorem 4.7, it suffices to show that for any any (C,U) with $\mathrm{poly}(\lambda)$ many oracle queries and any $\epsilon(\lambda) = 1/\mathrm{poly}(\lambda)$, it holds that

$$\Pr_{\mathsf{dk},|\mathsf{ck}\rangle,\mathsf{CK}\leftarrow\mathsf{Gen}(1^\lambda)}\left[\left\|\Pi_{\mathsf{dk},c,1}\mathsf{U}^{\mathsf{CK},\mathsf{DecZ}[\mathsf{dk}]}\Pi_{\mathsf{dk},c,0}\,|\psi\rangle\right\|^2 \geq \epsilon(\lambda) : (|\psi\rangle, c) \leftarrow \mathsf{C}^{\mathsf{CK},\mathsf{DecZ}[\mathsf{dk}],\mathsf{DecX}[\mathsf{dk}]}(|\mathsf{ck}\rangle)\right] = \mathrm{negl}(\lambda).$$

To show that Claim 4.8 implies the above statement, we define the oracle $O_\perp$ to always map $(\sigma, s) \to \perp$, and then argue that

$$\mathop{\mathbb{E}}_{\substack{\mathsf{dk},|\mathsf{ck}\rangle,\mathsf{CK}\leftarrow\mathsf{Gen}(1^\lambda)\\(|\psi\rangle,c)\leftarrow\mathsf{C}^{\mathsf{CK},\mathsf{DecZ}[\mathsf{dk}],\mathsf{DecX}[\mathsf{dk}]}(|\mathsf{ck}\rangle)}}\left[\left\|\Pi_{\mathsf{dk},c,1}\mathsf{U}^{\mathsf{CK},\mathsf{DecZ}[\mathsf{dk}]}\Pi_{\mathsf{dk},c,0}\,|\psi\rangle\right\|^2 - \left\|\Pi_{\mathsf{dk},c,1}\mathsf{U}^{O_\perp,\mathsf{DecZ}[\mathsf{dk}]}\Pi_{\mathsf{dk},c,0}\,|\psi\rangle\right\|^2\right] = \mathrm{negl}(\lambda).$$

This follows from a standard hybrid argument, by reduction to the unforgeability of the signature token scheme. That is, consider replacing each CK oracle query with a $O_\perp$ oracle query one by one, starting with the last query. That is, we define hybrid $\mathcal{H}_0$ to be

$$\mathop{\mathbb{E}}_{\substack{\mathsf{dk},|\mathsf{ck}\rangle,\mathsf{CK}\leftarrow\mathsf{Gen}(1^\lambda)\\(|\psi\rangle,c)\leftarrow\mathsf{C}^{\mathsf{CK},\mathsf{DecZ}[\mathsf{dk}],\mathsf{DecX}[\mathsf{dk}]}(|\mathsf{ck}\rangle)}}\left[\left\|\Pi_{\mathsf{dk},c,1}\mathsf{U}^{\mathsf{CK},\mathsf{DecZ}[\mathsf{dk}]}\Pi_{\mathsf{dk},c,0}\,|\psi\rangle\right\|^2\right],$$

and in hybrid $\mathcal{H}_i$, we switch the i'th from the last query from being answered by CK to being answered by $O_\perp$. Now, fix any i, and consider measuring the query register of U's i'th from last query to obtain classical strings (σ, s). Then since $\Pi_{\mathsf{dk},c,0}$ is the zero projector when c is not a valid signature on 1, and CK outputs $\bot$ whenever σ is not a valid signature on 0, we have that

$$\mathbb{E}[\mathcal{H}_{i-1} - \mathcal{H}_i] \leq \Pr[\mathsf{Tok}(\mathsf{vk}, 1, c) = 1 \wedge \mathsf{Tok}(\mathsf{vk}, 0, \sigma) = 1] = \text{negl}(\lambda),$$

by the unforgeability of the signature token scheme. Since there are $\text{poly}(\lambda)$ many hybrids, this completes the hybrid argument.

Finally, it follows by Markov that

$$\Pr_{\substack{\mathsf{dk},|\mathsf{ck}\rangle,\mathsf{CK}\leftarrow\mathsf{Gen}(1^\lambda)\\(|\psi\rangle,c)\leftarrow\mathsf{C}^{\mathsf{CK},\mathsf{DecZ}[\mathsf{dk}],\mathsf{DecX}[\mathsf{dk}]}(|\mathsf{ck}\rangle)}}\left[\left\|\Pi_{\mathsf{dk},c,1}\mathsf{U}^{\mathsf{CK},\mathsf{DecZ}[\mathsf{dk}]}\Pi_{\mathsf{dk},c,0}\left|\psi\right\rangle\right\|^2 - \left\|\Pi_{\mathsf{dk},c,1}\mathsf{U}^{O_\bot,\mathsf{DecZ}[\mathsf{dk}]}\Pi_{\mathsf{dk},c,0}\left|\psi\right\rangle\right\|^2 \geq \epsilon(\lambda) - \frac{1}{2^\lambda}\right]$$
$$\leq \frac{\text{negl}(\lambda)}{\epsilon(\lambda) - 1/2^\lambda} = \text{negl}(\lambda),$$

which completes the proof.

□

Now, we introduce some more notation.

- Let $\mathcal{A}_{k,n}$ be the set of balanced k-dimensional affine subspaces of $\mathbb{F}_2^n$.
- For an affine subspace $A = S + v$, let $O[A] : \mathbb{F}_2^n \to \{0,1\}$ be the classical functionality that outputs 1 on input s iff $s \in S + v$, and let $O[A^\perp] : \mathbb{F}_2^n \to \{0,1\}$ be the classical functionality that outputs 1 on input s iff $s \in S^\perp$.
- For an affine subspace $A = S + v$ and a bit $b \in \{0,1\}$, define the projector

$$\Pi[A_b] := \sum_{s\in(S+v)_b} \left|s\right\rangle\left\langle s\right|.$$

We will use this notation to re-define the game in Claim 4.8, and show that it suffices to prove the following claim.

Claim 4.10. *For any two unitaries* $(\mathsf{U}_{\mathsf{Com}}, \mathsf{U}_{\mathsf{Open}})$*, where* $\mathsf{U}_{\mathsf{Com}}$ *and* $\mathsf{U}_{\mathsf{Open}}$ *each make* $\text{poly}(\lambda)$ *many oracle queries, it holds that*

$$\Pr_{A\leftarrow\mathcal{A}_{n/2,n}}\left[\left\|\Pi[A_1]\mathsf{U}_{\mathsf{Open}}^{O[A]}\Pi[A_0]\left|\psi\right\rangle\right\|^2 \geq \frac{1}{2^\lambda} : \left|\psi\right\rangle := \mathsf{U}_{\mathsf{Com}}^{O[A],O[A^\perp]}(\left|A\right\rangle)\right] = \text{negl}(\lambda).$$

Lemma 4.11. *Claim 4.10 implies Claim 4.8.*

Proof. First, we note that re-defining $\Pi_{\mathsf{dk},c,b}$ in the statement of Claim 4.8 to ignore c and only check for membership in the affine subspace $(S+v)_b$ only potentially increases the squared norm of the resulting vector. This means that we can ignore the string c output by C. Then, we can give the committer vk in the clear, and observe that it is now straightforward for the committer to simulate its $\mathsf{DecZ}[\mathsf{dk}]$ oracle with $O[A]$, where A is the affine subspace defined by dk, and also to simulate its $\mathsf{DecX}[\mathsf{dk}]$ oracle with $O[A^\perp]$. Finally, we can purify any operation C to consider a unitary $\mathsf{U}_{\mathsf{Com}}$ that outputs $\left|\psi\right\rangle$. □

Our next step is to remove $\mathsf{U}_{\mathsf{Open}}$'s oracle access to $O[A]$. We will show that it suffices to prove the following.

Claim 4.12. *For any two unitaries* $(\mathsf{U}_{\mathsf{Com}}, \mathsf{U}_{\mathsf{Open}})$*, where* $\mathsf{U}_{\mathsf{Com}}$ *makes* $\mathrm{poly}(\lambda)$ *many oracle queries, it holds that*

$$\Pr_{A \leftarrow \mathcal{A}_{n/2,3n/4}}\left[\left\|\Pi[A_1]\mathsf{U}_{\mathsf{Open}}\Pi[A_0]\left|\psi\right\rangle\right\|^2 \geq \frac{1}{2^{\lambda+1}} : \left|\psi\right\rangle := \mathsf{U}_{\mathsf{Com}}^{O[A],O[A^{\perp}]}(\left|A\right\rangle)\right] = \mathrm{negl}(\lambda).$$

Notice that we are now sampling affine subspaces of a $3n/4$-dimensional space.

Lemma 4.13. *Claim 4.12 implies Claim 4.10.*

Proof. Given an $n/2$-dimensional affine subspace A, let $T \leftarrow \mathsf{Super}(3n/4, A)$ denote sampling a uniformly random $(3n/4)$-dimensional subspace T such that $A \subset T$. Then, define $O[T \setminus \{0^n\}]$ to be the oracle that checks for membership in the set $T \setminus \{0^n\}$.

Now, we will show via a standard hybrid argument that

$$\mathop{\mathbb{E}}_{\substack{A \leftarrow \mathcal{A}_{n/2}, \\ T \leftarrow \mathsf{Super}(3n/4, A) \\ \left|\psi\right\rangle := \mathsf{U}_{\mathsf{Com}}^{O[A],O[A^{\perp}]}(\left|A\right\rangle)}}\left[\left\|\Pi[A_1]\mathsf{U}_{\mathsf{Open}}^{O[A]}\Pi[A_0]\left|\psi\right\rangle\right\|^2 - \left\|\Pi[A_1]\mathsf{U}_{\mathsf{Open}}^{O[T\setminus\{0^n\}]}\Pi[A_0]\left|\psi\right\rangle\right\|^2\right] \leq \frac{\mathrm{poly}(\lambda)}{2^{n/4}}.$$

Consider replacing each $O[A]$ oracle query with a $O[T \setminus \{0^n\}]$ oracle query one by one, starting with the last query. That is, we define hybrid $\mathcal{H}_0$ to be

$$\mathop{\mathbb{E}}_{\substack{A \leftarrow \mathcal{A}_{n/2,n} \\ T \leftarrow \mathsf{Super}(3n/4, A) \\ \left|\psi\right\rangle := \mathsf{U}_{\mathsf{Com}}^{O[A],O[A^{\perp}]}(\left|A\right\rangle)}}\left[\left\|\Pi[A_1]\mathsf{U}_{\mathsf{Open}}^{O[A]}\Pi[A_0]\left|\psi\right\rangle\right\|^2\right],$$

and in hybrid $\mathcal{H}_i$, we switch the i'th from the last query from being answered by $O[A]$ to being answered by $O[T \setminus \{0^n\}]$. By Claim 3.2, we have that

$$\mathbb{E}[\mathcal{H}_{i-1} - \mathcal{H}_i] \leq \max_s \Pr_T[s \in (T \setminus \{0^n\}) \setminus S] \leq \frac{1}{2^{n/4}}.$$

Since there are $\mathrm{poly}(\lambda)$ many hybrids, this completes the hybrid argument. Now, it follows by Markov that

$$\Pr_{\substack{A \leftarrow \mathcal{A}_{n/2,n} \\ T \leftarrow \mathsf{Super}(3n/4, A) \\ \left|\psi\right\rangle := \mathsf{U}_{\mathsf{Com}}^{O[A],O[A^{\perp}]}(\left|A\right\rangle)}}\left[\left\|\Pi[A_1]\mathsf{U}_{\mathsf{Open}}^{O[A]}\Pi[A_0]\left|\psi\right\rangle\right\|^2 - \left\|\Pi[A_1]\mathsf{U}_{\mathsf{Open}}^{O[T\setminus\{0^n\}]}\Pi[A_0]\left|\psi\right\rangle\right\|^2 \geq \frac{1}{2^{\lambda}} - \frac{1}{2^{\lambda+1}}\right]$$

$$\leq \frac{\mathrm{poly}(\lambda)2^{\lambda+1}}{2^{n/4}} = \mathrm{negl}(\lambda),$$

since $n > 5\lambda$. This completes the proof, since we can imagine fixing T as a public ambient space of dimension $3n/4$ and sampling A as a random affine subspace of T.

□

Next, we perform a worst-case to average-case reduction over the sampling of A and thus show that it suffices to prove the following.

Claim 4.14. *There do not exist two unitaries* $(\mathsf{U}_{\mathsf{Com}}, \mathsf{U}_{\mathsf{Open}})$*, where* $\mathsf{U}_{\mathsf{Com}}$ *makes* $\mathrm{poly}(\lambda)$ *many oracle queries, such that for all* $A \in \mathcal{A}_{n/2,3n/4}$ *it holds that*

$$\left\| \Pi[A_1]\mathsf{U}_{\mathsf{Open}}\Pi[A_0] \left|\psi_A\right\rangle \right\|^2 \geq \frac{1}{2^{2\lambda}},$$

where $|\psi_A\rangle \coloneqq \mathsf{U}_{\mathsf{Com}}^{O[A],O[A^\perp]}(|A\rangle)$.

Lemma 4.15. *Claim 4.14 implies Claim 4.12.*

Proof. Suppose that there exists $(\mathsf{U}_{\mathsf{Com}}, \mathsf{U}_{\mathsf{Open}})$ that violates Claim 4.12. We define an adversary $(\widetilde{\mathsf{C}}, \widetilde{\mathsf{U}}_{\mathsf{Open}})$ as follows.

- $\widetilde{\mathsf{C}}$ takes $|A\rangle$ as input and samples a uniformly random change of basis B of $\mathbb{F}_2^{3n/4}$. Define the unitary U_B acting on $3n/4$ qubits to map $|s\rangle \to |B(s)\rangle$.
- Run $\mathsf{U}_{\mathsf{Com}}$ on $|B(A)\rangle$. Answer each of $\mathsf{U}_{\mathsf{Com}}$'s oracle queries with $\mathsf{U}_B O[A]\mathsf{U}_B^\dagger$ or $\mathsf{U}_B O[A^\perp]\mathsf{U}_B^\dagger$, where U_B acts on the query register.
- Let $|\psi\rangle$ be $\mathsf{U}_{\mathsf{Com}}$'s output, and output $|\widetilde{\psi}\rangle \coloneqq (\mathsf{U}_B^\dagger \left|\psi\right\rangle, B)$, where register $\mathcal{B}$ holds B, which is a classical description of the change of basis.
- $\widetilde{\mathsf{U}}_{\mathsf{Open}}$ is defined to be $\mathsf{U}_{\mathsf{CoB}^{-1}}\mathsf{U}_{\mathsf{Open}}\mathsf{U}_{\mathsf{CoB}}$, where

$$\mathsf{U}_{\mathsf{CoB}} \coloneqq \frac{1}{\#B}\sum_{\mathbf{B}} \mathsf{U}_B \otimes |B\rangle\langle B|^{\mathcal{B}}, \quad \text{and} \quad \mathsf{U}_{\mathsf{CoB}^{-1}} \coloneqq \frac{1}{\#B}\sum_{B} \mathsf{U}_B^\dagger \otimes |B\rangle\langle B|^{\mathcal{B}},$$

  where $\#B$ is the total number of change of bases B.

Then it holds that for any $A \in \mathcal{A}_{n/2,3n/4}$,

$$\begin{aligned}
&\Pr\left[\left\| \Pi[A_1]\widetilde{\mathsf{U}}_{\mathsf{Open}}\Pi[A_0] |\widetilde{\psi}\rangle \right\|^2 \geq \frac{1}{2^{\lambda+1}} : |\widetilde{\psi}\rangle \leftarrow \widetilde{\mathsf{C}}^{O[A],O[A^\perp]}(|A\rangle)\right] \\
&= \Pr_{B(A)\leftarrow\mathcal{A}_{n/2,3n/4}}\left[\left\| \Pi[B(A)_1]\mathsf{U}_{\mathsf{Open}}\Pi[B(A)_0] |\psi\rangle \right\|^2 \geq \frac{1}{2^{\lambda+1}} : |\psi\rangle \leftarrow \mathsf{U}_{\mathsf{Com}}^{O[B(A)],O[B(A)^\perp]}(|B(A)\rangle)\right] \\
&= \mathsf{non\text{-}negl}(\lambda),
\end{aligned}$$

where the final equality follows because we are assuming that $(\mathsf{U}_{\mathsf{Com}}, \mathsf{U}_{\mathsf{Open}})$ violates Claim 4.12, and for any fixed balanced A and uniformly random B, it holds that $B(A)$ is a uniformly random balanced affine subspace except with $\mathrm{negl}(n)$ probability. Now, define $|\widetilde{\psi}_B\rangle$ to be the output of $\widetilde{\mathsf{C}}$ conditioned on sampling B. Then define $\widetilde{\mathsf{U}}_{\mathsf{Com}}$ to be a purification of C. It holds that for any fixed $A \in \mathcal{A}_{n/2,3n/4}$ and $|\widetilde{\psi}\rangle \coloneqq \widetilde{\mathsf{U}}_{\mathsf{Com}}^{O[A],O[A^\perp]}(|A\rangle)$,

$$\left\| \Pi[A_1]\widetilde{\mathsf{U}}_{\mathsf{Open}}\Pi[A_0] |\widetilde{\psi}\rangle \right\|^2 = \frac{1}{\#B}\sum_B \left\| \Pi[A_1]\widetilde{\mathsf{U}}_{\mathsf{Open}}\Pi[A_0] |\widetilde{\psi}_B\rangle \right\|^2 \geq \mathsf{non\text{-}negl}(\lambda) \cdot \frac{1}{2^{\lambda+1}} \geq \frac{1}{2^{2\lambda}},$$

which completes the proof. □

Next, we perform amplitude amplification onto $\Pi[A_0]$, showing that it suffices to prove the following claim.

Claim 4.16. *There do not exist two unitaries* $(\mathsf{U}_{\mathsf{Com}}, \mathsf{U}_{\mathsf{Open}})$*, where* $\mathsf{U}_{\mathsf{Com}}$ *makes at most* $2^{2\lambda}$ *oracle queries, such that for all* $A \in \mathcal{A}_{n/2,3n/4}$ *and* $|\psi_A\rangle := \mathsf{U}_{\mathsf{Com}}^{O[A],O[A^\perp]}(|A\rangle)$*, there exists a state* $|\psi'_A\rangle$ *such that*

$$\| |\psi_A\rangle - |\psi'_A\rangle \| \leq \frac{1}{2^{15\lambda}}, \quad |\psi'_A\rangle \in \mathsf{Im}(\Pi[A_0]), \quad \textit{and} \quad \|\Pi[A_1]U_{\mathsf{Open}} |\psi'_A\rangle \| \geq \frac{1}{2^\lambda}.$$

Lemma 4.17. *Claim 4.16 implies Claim 4.14.*

Proof. For any binary projective measurement $(\Pi, \mathbb{I} - \Pi)$, we define U_Π to be a unitary that maps $|\phi\rangle \to -|\phi\rangle$ for any $|\phi\rangle \in \mathsf{Im}(\Pi)$ and acts as the identity on all $|\phi\rangle$ orthogonal to Π. We use the following imported theorem.

Imported Theorem 4.18 (Fixed-point amplitude amplification, [GSLW19] Theorem 27)**.** *There exists an oracle-aided unitary* $\mathsf{Amplify}$ *that is parameterized by* (α, β)*, and has the following properties. Let* $|\psi\rangle$ *and* $|\psi_G\rangle$ *be normalized states and* Π *be a projector such that* $\Pi |\psi\rangle = \gamma |\psi_G\rangle$*, where* $\gamma \geq \alpha$*. Then* $|\widetilde{\psi}_G\rangle := \mathsf{Amplify}_{\alpha,\beta}^{\mathsf{U}_{|\psi\rangle\langle\psi|},\mathsf{U}_\Pi}(|\psi\rangle)$ *is such that* $\| |\psi_G\rangle - |\widetilde{\psi}_G\rangle \| \leq \beta$*, and* $\mathsf{Amplify}_{\alpha,\beta}^{\mathsf{U}_{|\psi\rangle\langle\psi|},\mathsf{U}_\Pi}(|\psi\rangle)$ *makes* $O(\log(1/\beta)/\alpha)$ *oracle queries.*

Now, suppose that $(\mathsf{U}_{\mathsf{Com}}, \mathsf{U}_{\mathsf{Open}})$ violates Claim 4.14. Set $\alpha = 1/2^\lambda$, $\beta = 1/2^{15\lambda}$, and define

$$\widetilde{\mathsf{U}}_{\mathsf{Com}}(|A\rangle) := \mathsf{Amplify}_{\alpha,\beta}^{\mathsf{U}_{|\psi_A\rangle\langle\psi_A|},\mathsf{U}_{\Pi[A_0]}}(|\psi_A\rangle),$$

where $|\psi_A\rangle := \mathsf{U}_{\mathsf{Com}}(|A\rangle)$.

We first argue that $\widetilde{\mathsf{U}}_{\mathsf{Com}}$ can be implemented with just oracle access to $O[A]$ and $O[A^\perp]$. Clearly, the projector $\Pi[A_0]$ can be implemented with $O[A]$, so it remains to show how to implement the projector $|\psi_A\rangle \langle\psi_A|$. Note that

$$|\psi_A\rangle \langle\psi_A| = \mathsf{U}_{\mathsf{Com}} |A\rangle \langle A| \mathsf{U}_{\mathsf{Com}}^\dagger,$$

so it suffices to show how to implement $|A\rangle \langle A|$.

Recalling that $A = S + v$, we claim that

$$|A\rangle \langle A| = H^{\otimes n}\Pi[S^\perp]H^{\otimes n}\Pi[S + v].$$

The proof is essentially shown in [AC12, Lemma 21] (in the case where A is a subspace), and we repeat it here for completeness. It is clear that $H^{\otimes n}\Pi[S^\perp]H^{\otimes n}\Pi[S+v] |A\rangle = |A\rangle$, so it remains to show that for any $|\psi\rangle$ such that $\langle\psi|A\rangle = 0$, $H^{\otimes n}\Pi[S^\perp]H^{\otimes n}\Pi[S + v] |\psi\rangle = 0$. Write $|\psi\rangle = \sum_{s\in\{0,1\}^n} c_s |s\rangle$, where $\sum_{s\in S+v} c_s = 0$. Then

$$\begin{aligned}
H^{\otimes n}\Pi[S^\perp]H^{\otimes n}\Pi[S+v]\,|\psi\rangle &= H^{\otimes n}\Pi[S^\perp]H^{\otimes n}\sum_{s\in S+v} c_s\,|s\rangle \\
&= \frac{1}{2^{n/2}}H^{\otimes n}\Pi[S^\perp]\sum_{t\in\{0,1\}^n}\sum_{s\in S+v}(-1)^{s\cdot t}c_s\,|t\rangle \\
&= \frac{1}{2^{n/2}}H^{\otimes n}\sum_{t\in S^\perp}\sum_{s\in S+v}(-1)^{s\cdot t}c_s\,|t\rangle \\
&= \frac{1}{2^{n/2}}H^{\otimes n}\sum_{t\in S^\perp}\left(\sum_{s\in S+v}c_s\right)|t\rangle = 0.
\end{aligned}$$

Thus, $\widetilde{\mathsf{U}}_{\mathsf{Com}}$ can be implemented with just oracle access to $O[A]$ and $O[A^\perp]$. Moreover, it makes at most $O(\log(1/\beta)\alpha)\cdot\text{poly}(\lambda) \le O(\lambda 2^\lambda)\cdot\text{poly}(\lambda) \le 2^{2\lambda}$ queries to $O[A]$ and $O[A^\perp]$.

Now, define

$$|\psi'_A\rangle := \frac{\Pi[A_0]\,|\psi_A\rangle}{\|\Pi[A_0]\,|\psi_A\rangle\,\|},$$

so $|\psi'_A\rangle \in \mathsf{Im}(\Pi[A_0])$ by definition. By the fact that $(\mathsf{U}_{\mathsf{Com}},\mathsf{U}_{\mathsf{Open}})$ violates Claim 4.14, we know that

$$\left\|\Pi[A_1]\mathsf{U}_{\mathsf{Open}}\,|\psi'_A\rangle\,\right\|^2 \ge \left\|\Pi[A_1]\mathsf{U}_{\mathsf{Open}}\Pi[A_0]\,|\psi_A\rangle\,\right\|^2 \ge \frac{1}{2^{2\lambda}} \implies \left\|\Pi[A_1]\mathsf{U}_{\mathsf{Open}}\,|\psi'_A\rangle\,\right\| \ge \frac{1}{2^\lambda}.$$

Finally, by the definition of $|\psi'_A\rangle$,

$$\|\Pi[A_1]\mathsf{U}_{\mathsf{Open}}\Pi[A_0]\,|\psi_A\rangle\,\|^2 \ge \frac{1}{2^{2\lambda}} \implies \Pi[A_0]\,|\psi_A\rangle = \gamma\,|\psi'_A\rangle \text{ for } \gamma \ge \frac{1}{2^\lambda},$$

so the guarantee of Imported Theorem 4.18 implies that

$$\left\|\widetilde{\mathsf{U}}_{\mathsf{Com}}(|A\rangle) - |\psi'_A\rangle\,\right\| \le \frac{1}{2^{15p}}.$$

Thus, $(\widetilde{\mathsf{U}}_{\mathsf{Com}},\mathsf{U}_{\mathsf{Open}})$ violates Claim 4.16, which completes the proof.

□

Finally, we prove Claim 4.16, which, as we have shown, suffices to prove Theorem 4.7.

Proof. (of Claim 4.16) We will use the following imported theorem.

Imported Theorem 4.19 ([AC12]). *Let $\mathcal{O}$ be a set of classical functionalities $F:\{0,1\}^*\to\{0,1\}$. Let $\mathcal{R}$ be a symmetric binary relation between functionalities where for every $F\in\mathcal{O}$, $(F,F)\notin\mathcal{R}$, and for every $F\in\mathcal{O}$, there exists $G\in\mathcal{O}$ such that $(F,G)\in\mathcal{R}$. Moreover, for any $F\in\mathcal{O}$ and x such that $F(x)=0$, suppose that*

$$\Pr_{G\leftarrow\mathcal{R}_F}[G(x)=1]\le\delta,$$

where $\mathcal{R}_F$ is the set of G such that $(F,G)\in\mathcal{R}$. Now, consider any oracle-aided unitary $\mathsf{U}^F(|\psi_F\rangle)$ that has oracle access to some $F\in\mathcal{O}$, is initialized with some state $|\psi_F\rangle$ that may depend on F, makes T queries, and outputs a state $|\widetilde{\psi}_F\rangle$. Then if $|\langle\psi_F|\psi_G\rangle|\ge c$ for all $(F,G)\in\mathcal{R}$ and $\mathbb{E}_{(F,G)\leftarrow\mathcal{R}}[|\langle\widetilde{\psi}_F|\widetilde{\psi}_G\rangle|]\le d$, then $T=\Omega\left(\frac{c-d}{\sqrt{\delta}}\right)$.

Now, suppose there exists $\mathsf{U}_{\mathsf{Com}}, \mathsf{U}_{\mathsf{Open}}$ that violates Claim 4.16. Recall that $\mathsf{U}_{\mathsf{Com}}$ has access to the oracles $O[A]$ and $O[A^\perp]$, defined by the $n/2$-dimensional balanced affine subspace $A = S + v$ of $\mathbb{F}_2^{3n/4}$. We define a single functionality F_A that takes as input (b, s) and if $b = 0$ outputs whether $s \in S + v$, and if $b = 1$ outputs whether $s \in S^\perp$.

Then, we define a binary symmetric relation on functionalities F_A, F_B as follows. Letting $A = S_A + v_A$ and $B = S_B + v_B$, we define $(F_A, F_B) \in \mathcal{R}$ if and only if $\mathsf{dim}(A_0 \cap B_0) = n/2 - 2$ and $\mathsf{dim}(A_1 \cap B_1) = n/2 - 2$. Note that for any $(F_A, F_B) \in \mathcal{R}$, $\mathsf{dim}(A \cap B) = n/2 - 1$.

Given $\mathcal{R}$ defined this way, we see that for any fixed F_A and (b, s) such that $F_A(b, s) = 0$,

$$\begin{aligned}
&\Pr_{F_B \leftarrow \mathcal{R}_{F_A}}[F_B(b, s) = 1] \\
&\qquad \le \max\left\{\frac{|B \setminus A|}{|\mathbb{F}_2^{3n/4} \setminus A \setminus \{0^{3n/4}\}|}, \frac{|S_B^\perp \setminus S_A^\perp|}{|\mathbb{F}_2^{3n/4} \setminus S_A^\perp|}\right\} \\
&\qquad \le \max\left\{\frac{2^{n/2-1}}{2^{3n/4} - 2^{n/2} - 1}, \frac{2^{n/4-1}}{2^{3n/4} - 2^{n/4}}\right\} \\
&\qquad \le \frac{1}{2^{n/4}}.
\end{aligned}$$

Next, we note that $\mathsf{U}_{\mathsf{Com}}^{F_A}$ is initialized with the state $|A\rangle$, and, for any (A, B) such that $(F_A, F_B) \in \mathcal{R}$, it holds that $|\langle A|B\rangle| = 1/2$. Our goal is then to bound

$$\mathop{\mathbb{E}}_{(F_A, F_B) \leftarrow \mathcal{R}}[|\langle \psi_A|\psi_B\rangle|],$$

where $|\psi_A\rangle = \mathsf{U}_{\mathsf{Com}}^{F_A}(|A\rangle)$. Since $(\mathsf{U}_{\mathsf{Com}}, \mathsf{U}_{\mathsf{Open}})$ violates Claim 4.16, we can write each $|\psi_A\rangle$ as $|\psi'_A\rangle + |\psi_A^{\mathsf{err}}\rangle$, where

$$\| |\psi_A^{\mathsf{err}}\rangle \| \le \frac{1}{2^{15\lambda}}, \quad |\psi'_A\rangle \in \mathsf{Im}(\Pi[A_0]), \quad \text{and} \quad \|\Pi[A_1]\mathsf{U}_{\mathsf{Open}}|\psi'_A\rangle\| \ge \frac{1}{2^\lambda}.$$

Thus, we have that

$$\mathop{\mathbb{E}}_{(F_A, F_B) \leftarrow \mathcal{R}}[|\langle \psi_A|\psi_B\rangle] \le \mathop{\mathbb{E}}_{(F_A, F_B) \leftarrow \mathcal{R}}[|\langle \psi'_A|\psi'_B\rangle|] + \frac{3}{2^{15\lambda}}.$$

Now, we appeal to the following theorem, which is proven in Appendix B.

Theorem 4.20. *Let $n, m, d \in \mathbb{N}, \epsilon \in (0, 1/8)$ be such that $d \ge 2$ and $n - d + 1 > 10\log(1/\epsilon) + 6$. Let $\mathsf{U}^{\mathcal{X},\mathcal{Y}}$ be any (2^{n+m})-dimensional unitary, where register $\mathcal{X}$ is 2^n dimensions and register $\mathcal{Y}$ is 2^m dimensions. Let $\mathcal{A}$ be the set of d-dimensional balanced affine subspaces $A = (A_0, A_1)$ of $\mathbb{F}_2^n$, where A_0 is the affine subspace of vectors in A that start with 0 and A_1 is the affine subspace of vectors in A that start with 1. For any $A = (A_0, A_1)$, let*

$$\Pi_{A_0} := \sum_{v \in A_0} |v\rangle\langle v|^{\mathcal{X}} \otimes \mathbb{I}^{\mathcal{Y}}, \quad \Pi_{A_1} := \mathsf{U}^\dagger \left(\sum_{v \in A_1} |v\rangle\langle v|^{\mathcal{X}} \otimes \mathbb{I}^{\mathcal{Y}}\right) \mathsf{U}.$$

Let $\mathcal{R}$ be the set of pairs (A, B) of d-dimensional affine subspaces of $\mathbb{F}_2^n$ such that $\dim(A_0 \cap B_0) = d - 2$ *and* $\dim(A_1 \cap B_1) = d - 2$. *Then for any set of states $\{|\psi_A\rangle\}_A$ such that for all $A \in \mathcal{A}$, $|\psi_A\rangle \in \mathsf{Im}(\Pi_{A_0})$, and $\|\Pi_{A_1} |\psi_A\rangle\| \geq \epsilon$,*

$$\mathop{\mathbb{E}}_{(A,B)\leftarrow\mathcal{R}}[|\langle\psi_A|\psi_B\rangle|] < \frac{1}{2} - \epsilon^{13}.$$

Setting $\epsilon = 1/2^\lambda$, and noting that $3n/4 - n/2 + 1 > 11\lambda > 10\log(2^\lambda) + 6$, this theorem implies that

$$\mathop{\mathbb{E}}_{(F_A,F_B)\leftarrow\mathcal{R}}[|\langle\psi'_A|\psi'_B\rangle|] \leq \frac{1}{2} - \frac{1}{2^{13\lambda}},$$

and thus we conclude that

$$\mathop{\mathbb{E}}_{(F_A,F_B)\leftarrow\mathcal{R}}[|\langle\psi_A|\psi_B\rangle] \leq \frac{1}{2} - \frac{1}{2^{14\lambda}}.$$

Thus, by Imported Theorem 4.19, $\mathsf{U}_{\mathsf{Com}}$ must be making

$$\Omega\left(\frac{2^{n/8}}{2^{14\lambda}}\right) = \Omega\left(2^{130\lambda/8-14\lambda}\right) > 2^{2\lambda}$$

oracle queries, recalling that $n \geq 130\lambda$. However, $\mathsf{U}_{\mathsf{Com}}$ was assumed to be making at most $2^{2\lambda}$ queries, so this is a contradiction, completing the proof.

□

5 Verification of Quantum Partitioning Circuits

5.1 Definition

A protocol for publicly-verifiable non-interactive classical verification of quantum partitioning circuits consists of the following procedures. We write the syntax in the *oracle model*, where the prover obtains access to a classical oracle as part of its public key. We also specify a quantum proving key $|\mathsf{pk}\rangle$, but note that one could also consider the case where the proving key pk is classical.

- $\mathsf{Gen}(1^\lambda, Q) \rightarrow (\mathsf{vk}, |\mathsf{pk}\rangle, \mathsf{PK})$: The Gen algorithm takes as input the security parameter 1^λ and the description of a quantum circuit $Q : \{0,1\}^{n'} \rightarrow \{0,1\}^n$, and outputs a classical verification key vk and a quantum proving key $(|\mathsf{pk}\rangle, \mathsf{PK})$, which consists of a quantum state $|\mathsf{pk}\rangle$ and the description of a classical deterministic polynomial-time functionality $\mathsf{PK} : \{0,1\}^* \rightarrow \{0,1\}^*$.

- $\mathsf{Prove}^{\mathsf{PK}}(|\mathsf{pk}\rangle, Q, x) \rightarrow \pi$: The Prove algorithm has oracle access to PK, takes as input the quantum proving key $|\mathsf{pk}\rangle$, a circuit Q, and an input $x \in \{0,1\}^{n'}$, and outputs a proof π.

- $\mathsf{Ver}(\mathsf{vk}, x, \pi) \rightarrow \{(q_1, \ldots, q_m)\} \cup \{\bot\}$: The classical Ver algorithm takes as input the verification key vk, an input x, and a proof π, and either outputs a sequence of samples $(q_1, \ldots, q_m)$ or $\bot$.

- $\mathsf{Combine}(b_1, \ldots, b_m) \rightarrow b$: The $\mathsf{Combine}$ algorithm takes as input a sequence of bits $(b_1, \ldots, b_m)$ and outputs a bit b.

The proof should satisfy the following notions of completeness and soundness.

Definition 5.1 (Publicly-verifiable non-interactive classical verification of quantum partitioning circuits: Completeness)**.** *A protocol for publicly-verifiable non-interactive classical verification of quantum partitioning circuits is* complete *if for any family $\{Q_\lambda, P_\lambda\}_{\lambda\in\mathbb{N}}$ such that $\{P_\lambda \circ Q_\lambda\}_{\lambda\in\mathbb{N}}$ is pseudo-deterministic, and any sequence of inputs $\{x_\lambda\}_{\lambda\in\mathbb{N}}$, it holds that (where we leave indexing by λ implicit)*

$$\Pr\left[\begin{array}{l}\mathsf{Ver}(\mathsf{vk}, x, \pi) = (q_1, \dots, q_m) \wedge \\ \mathsf{Combine}(P(q_1), \dots, P(q_m)) = P(Q(x))\end{array} : \begin{array}{r}(\mathsf{vk}, |\mathsf{pk}\rangle, \mathsf{PK}) \leftarrow \mathsf{Gen}(1^\lambda, Q) \\ \pi \leftarrow \mathsf{Prove}^{\mathsf{PK}}(|\mathsf{pk}\rangle, Q, x)\end{array}\right] = 1 - \mathrm{negl}(\lambda).$$

We define soundness in the oracle model, where the adversarial prover gets access to an oracle for the functionality $\mathsf{Ver}(\mathsf{vk}, \cdot, \cdot)$.

Definition 5.2 (Publicly-verifiable non-interactive classical verification of quantum partitioning circuits: Soundness)**.** *A protocol for publicly-verifiable non-interactive classical verification of quantum partitioning circuits is* sound *if for any family $\{Q_\lambda, P_\lambda\}_{\lambda\in\mathbb{N}}$ such that $\{P_\lambda \circ Q_\lambda\}_{\lambda\in\mathbb{N}}$ is pseudo-deterministic, and any QPT adversarial prover $\{\mathsf{A}_\lambda\}_{\lambda\in\mathbb{N}}$, it holds that (where we leave indexing by λ implicit)*

$$\Pr\left[\begin{array}{l}\mathsf{Ver}(\mathsf{vk}, x, \pi) = (q_1, \dots, q_m) \wedge \\ \mathsf{Combine}(P(q_1), \dots, P(q_m)) = 1 - P(Q(x))\end{array} : \begin{array}{r}(\mathsf{vk}, |\mathsf{pk}\rangle, \mathsf{PK}) \leftarrow \mathsf{Gen}(1^\lambda, Q) \\ (x, \pi) \leftarrow \mathsf{A}^{\mathsf{PK}, \mathsf{Ver}[\mathsf{vk}]}(|\mathsf{pk}\rangle)\end{array}\right] = \mathrm{negl}(\lambda),$$

where $\mathsf{Ver}[\mathsf{vk}]$ *is the classical functionality* $\mathsf{Ver}(\mathsf{vk}, \cdot, \cdot) : (x, \pi) \to \{(q_1, \dots, q_m)\} \cup \{\bot\}$.

5.2 QPIP$_1$ verification

First, we recall an information-theoretic protocol for verifying quantum partitioning circuits using only single-qubit standard and Hadamard basis measurements.[19] This protocol is a λ-wise parallel repetition of the quantum sampling verification protocol from [CLLW22], and was described in [Bar21]. Most of the underlying details of the protocol will not be important to us, but we provide a high-level description.

The prover prepares multiple copies of a history state of the computation $Q(x)$, which is in general a sampling circuit. Each history state is prepared in a special way [CLLW22] to satisfy the following properties: (i) a sample approximately from the output distribution may be obtained by measuring certain registers of the state in the *standard basis*, which can be achieved by adding enough dummy identity gates to ensure that the output state is a large fraction of the history state, and (ii) the history state is the *unique* ground state of the Hamiltonian, and all orthogonal states have much higher energy, ensuring that the verifier can test the validity of the entire computation by testing the energy of the history state.

Then, the verifier samples certain copies for *verifying* and other copies for *sampling*. In the verify copies, it samples a random Hamiltonian term, and measures in the corresponding standard and Hadamard bases, while in the sample copies, the verifier measures the output register in the standard basis. If the verifier accepts the results from measuring the verify copies, it outputs the collection of samples obtained from the sample copies. It was shown by [Bar21] that if Q is a partitioning circuit with predicate P, then one can set parameters so that conditioned on

[19]Quantum interactive protocols where the verifier only requires the ability to measure single qubits have been referred to as QPIP$_1$ protocols.

verification passing, it holds with *overwhelming probability* that *at least half* of the output samples q_t are such that $P(q_t) = P(Q(x))$. We describe the formal syntax of this protocol in Fig. 4, where the prover state $|\psi\rangle$ consists of sufficiently many copies of the history state, and the verifier's string h of measurement bases consists of (mostly) indices used for verification as well as some indices used for sampling outputs, which we denote by S. By an observation of [ACGH20], the sampling of h can be performed independently of the input x, which is reflected in the syntax of Fig. 4 (technically, it only needs the size $|Q|$ rather than Q itself).

Next, we introduce some notation, and then state the correctness and soundness guarantees of this protocol that follow from prior work.

Definition 5.3. *Define* Maj *to be the predicate that takes as input a set of bits* $\{b_i\}_i$ *and outputs the most frequently occurring bit b. In the event of a tie, we arbitrarily set the output to 0.*

Definition 5.4. *For a string* $x \in \{0,1\}^n$ *and a subset* $S \subseteq [n]$, *define* $x[S]$ *to be the string consisting of bits* $\{x_i\}_{i\in S}$.

Definition 5.5. *Given an* $h \in \{0,1\}^n$ *and an* n*-qubit state* $|\psi\rangle$, *let* $M(h, |\psi\rangle)$ *denote the distribution over* n*-bit strings that results from measuring each qubit* i *of* $|\psi\rangle$ *in basis* h_i, *where the bit* $h_i = 0$ *indicates standard basis and* $h_i = 1$ *indicates Hadamard basis.*

Imported Theorem 5.6 ([CLLW22, Bar21]). *The protocol* Π^{QV} *(Fig. 4) that satisfies the following properties.*

- ***Completeness.*** *For any family* $\{Q_\lambda, P_\lambda\}_{\lambda\in\mathbb{N}}$ *such that* $\{P_\lambda \circ Q_\lambda,\}_{\lambda\in\mathbb{N}}$ *is pseudo-deterministic, and any sequence of inputs* $\{x_\lambda\}_{\lambda\in\mathbb{N}}$,

$$\Pr\left[\mathsf{V}^{\mathsf{QV}}_{\mathsf{Ver}}(Q,x,h,m) = \top \wedge \mathsf{Maj}(\{P(q_t)\}_t) = P(Q(x)) : \begin{array}{r} |\psi\rangle \leftarrow \mathsf{P}^{\mathsf{QV}}(1^\lambda, Q, x) \\ (h,S) \leftarrow \mathsf{V}^{\mathsf{QV}}_{\mathsf{Gen}}(1^\lambda, Q) \\ m \leftarrow M(h, |\psi\rangle) \\ \{q_t\}_{t\in[\lambda]} \coloneqq m[S] \end{array}\right] = 1-\mathrm{negl}(\lambda).$$

- ***Soundness.*** *For any family* $\{Q_\lambda, P_\lambda\}_{\lambda\in\mathbb{N}}$ *such that* $\{P_\lambda \circ Q_\lambda\}_{\lambda\in\mathbb{N}}$ *is pseudo-deterministic, any sequence of inputs* $\{x_\lambda\}_{\lambda\in\mathbb{N}}$, *and any sequence of states* $\{|\psi^*_\lambda\rangle\}_{\lambda\in\mathbb{N}}$,

$$\Pr\left[\mathsf{V}^{\mathsf{QV}}_{\mathsf{Ver}}(Q,x,h,m) = \top \wedge \mathsf{Maj}(\{P(q_t)\}_t) = 1 - P(Q(x)) : \begin{array}{r} (h,S) \leftarrow \mathsf{V}^{\mathsf{QV}}_{\mathsf{Gen}}(1^\lambda, Q) \\ m \leftarrow M(h, |\psi^*\rangle) \\ \{q_t\}_{t\in[\lambda]} \coloneqq m[S] \end{array}\right] = \mathrm{negl}(\lambda).$$

5.3 Classical verification

Next, we compile the above information-theoretic protocol into a classically-verifiable but computationally-sound protocol, using Mahadev's measurement protocol [Mah22]. The measurement protocol itself is a four-message protocol with a single bit challenge from the verifier. Then, we apply parallel repetition and Fiat-Shamir, following [ACGH20, CCY20, Bar21], which results in a two-message negligibly-sound protocol in the quantum random oracle model.

The resulting protocol $\Pi^{\mathsf{CV}} = (\mathsf{P}^{\mathsf{CV}}_{\mathsf{Prep}}, \mathsf{V}^{\mathsf{CV}}_{\mathsf{Gen}}, \mathsf{P}^{\mathsf{CV}}_{\mathsf{Prove}}, \mathsf{P}^{\mathsf{CV}}_{\mathsf{Meas}}, \mathsf{V}^{\mathsf{CV}}_{\mathsf{Ver}})$ makes use of a dual-mode randomized trapdoor claw-free hash function ($\mathsf{TCF.Gen}, \mathsf{TCF.Eval}, \mathsf{TCF.Invert}, \mathsf{TCF.Check}, \mathsf{TCF.IsValid}$)

QPIP$_1$ protocol $\Pi^{\mathsf{QV}} = \left(\mathsf{P}^{\mathsf{QV}}, \mathsf{V}^{\mathsf{QV}}_{\mathsf{Gen}}, \mathsf{V}^{\mathsf{QV}}_{\mathsf{Ver}}\right)$

Parameters: Number of bits n output by Q, and number of qubits $\ell = \ell(\lambda)$ in the prover's state.

Prover's computation

- $\mathsf{P}^{\mathsf{QV}}(1^\lambda, Q, x) \to |\psi\rangle$: on input the security parameter 1^λ, the description of a quantum circuit Q, and an input x, the prover prepares a state $|\psi\rangle$ on ℓ qubits, and sends it to the verifier.

Verifier's computation

- $\mathsf{V}^{\mathsf{QV}}_{\mathsf{Gen}}(1^\lambda, Q) \to (h, S)$: on input the security parameter 1^λ and the description of a quantum circuit Q, the verifier's Gen algorithm samples a string $h \in \{0,1\}^\ell$ and a subset $S \subset [\ell]$ of size $n \cdot \lambda$ with the property that for all $i \in S$, $h_i = 0$.
- Next, the verifier measures $m \leftarrow M(h, |\psi\rangle)$ to obtain a string of measurement results $m \in \{0,1\}^\ell$.
- $\mathsf{V}^{\mathsf{QV}}_{\mathsf{Ver}}(Q, x, h, m) \to \{\top, \bot\}$: on input a circuit Q, input x, string of bases h, and measurement results m, the verifier's Ver algorithm outputs $\top$ or $\bot$.
- If $\top$, the verifier outputs the string $m[S]$ which is parsed as $\{q_t\}_{t \in [\lambda]}$ where each $q_t \in \{0,1\}^n$ and otherwise the verifier outputs $\bot$.

Figure 4: Syntax for a QPIP$_1$ protocol that verifies the output of a quantum partitioning circuit Q.

(Definition 3.6), and is described in Fig. 5. We choose to explicitly split the second prover's algorithm into two parts $\mathsf{P}^{\mathsf{CV}}_{\mathsf{Prove}}$ and $\mathsf{P}^{\mathsf{CV}}_{\mathsf{Meas}}$ for ease of notation when we build on top of this protocol in the next section.

We introduce some notation needed for describing the security properties of this protocol.

- Fix a security parameter λ, circuit Q, input x, and parameters $(\mathsf{pp}, \mathsf{sp}) \in \mathsf{V}^{\mathsf{CV}}_{\mathsf{Gen}}(1^\lambda, Q)$.
- Based on $\mathsf{sp} = \{h_i, S_i, \{\mathsf{sk}_{i,j}\}_{j \in [\ell]}\}_{i \in [r]}$, we define the set $S := \{S_i\}_{i \in [r]}$. For any proof $\pi = \{b_{i,j}, y_{i,j}, z_{i,j}\}_{i,j}$ generated by P^{CV}, we let $w := \mathsf{TestRoundOutputs}[\mathsf{sp}](\pi)$ be a string $w \in \{0,1\}^{|S|}$ defined as follows. Let $T := H(y_{1,1}, \ldots, y_{r,\ell})$. The string w consists of r substrings $w_1, \ldots, w_r$, where for each $i : T_i = 0$, w_i consists of the bits $\{b_{i,j}\}_{j \in S_i}$, and for each $i : T_i = 1$, $w_i = 0^{|S_i|}$.
- For any predicate P and bit $b \in \{0,1\}$, we define the set $D_{\mathsf{in}}[P, b] \subset \{0,1\}^{|S|}$ to consist of $w := (w_1, \ldots, w_r)$ with the following property. There are at least $3/4$ fraction of w_i such that, parsing w_i as $(w_{i,1}, \ldots, w_{i,\lambda})$, it holds that $\mathsf{Maj}\left(\{P(w_{i,t})\}_{t \in [\lambda]}\right) = b$.
- For any predicate P and bit $b \in \{0,1\}$, we define the set $D_{\mathsf{out}}[P, b] \subset \{0,1\}^{|S|}$ to consist of $w := (w_1, \ldots, w_r)$ with the following property. There are at least $1/3$ fraction of w_i such that, parsing w_i as $(w_{i,1}, \ldots, w_{i,\lambda})$, it holds that $\mathsf{Maj}\left(\{P(w_{i,t})\}_{t \in [\lambda]}\right) = 1 - b$.

Note that for any predicate P and $b \in \{0,1\}$, $D_{\mathsf{in}}[P, b]$ and $D_{\mathsf{out}}[P, b]$ are disjoint sets of strings.

Now, we state four properties that Π^{CV} satisfies. The proof of Lemma 5.8 follows immediately from the completeness of Π^{QV} (Imported Theorem 5.6) and the correctness of the dual-mode randomized trapdoor claw-free hash function (Definition 3.6). The proofs of the remaining three lemmas mostly follow from the prior work of [Bar21], and we show this formally in Appendix C.

Classically-verifiable protocol $\Pi^{\mathsf{CV}} = \left(\mathsf{P}^{\mathsf{CV}}_{\mathsf{Prep}}, \mathsf{V}^{\mathsf{CV}}_{\mathsf{Gen}}, \mathsf{P}^{\mathsf{CV}}_{\mathsf{Prove}}, \mathsf{P}^{\mathsf{CV}}_{\mathsf{Meas}}, \mathsf{V}^{\mathsf{CV}}_{\mathsf{Ver}}\right)$

Parameters: Number of qubits per round $\ell := \ell(\lambda)$, number of parallel rounds $r := r(\lambda)$, number of Hadamard rounds $k := k(\lambda)$, and random oracle $H : \{0,1\}^* \to \{0,1\}^{\log\binom{r}{k}}$.

- $\mathsf{P}^{\mathsf{CV}}_{\mathsf{Prep}}(1^\lambda, Q, x) \to (\mathcal{B}_1, \ldots, \mathcal{B}_r)$: For each $i \in [r]$, prepare the state $|\psi_i\rangle := \mathsf{P}^{\mathsf{QV}}(1^\lambda, Q, x)$ on register $\mathcal{B}_i = (\mathcal{B}_{i,1}, \ldots, \mathcal{B}_{i,\ell})$, which we write as
$$|\psi_i\rangle := \sum_{v \in \{0,1\}^\ell} \alpha_v |v\rangle^{\mathcal{B}_i}.$$
- $\mathsf{V}^{\mathsf{CV}}_{\mathsf{Gen}}(1^\lambda, Q) \to (\mathsf{pp}, \mathsf{sp})$: For each $i \in [r]$, sample $(h_i, S_i) \leftarrow \mathsf{V}^{\mathsf{QV}}_{\mathsf{Gen}}(1^\lambda, Q)$ where $h_i = (h_{i,1}, \ldots, h_{i,\ell})$, and sample $\{(\mathsf{pk}_{i,j}, \mathsf{sk}_{i,j}) \leftarrow \mathsf{TCF.Gen}(1^\lambda, h_{i,j})\}_{j \in [\ell]}$. Then, set
$$\mathsf{pp} := \{\{\mathsf{pk}_{i,j}\}_{j\in[\ell]}\}_{i\in[r]}, \mathsf{sp} := \{h_i, S_i, \{\mathsf{sk}_{i,j}\}_{j\in[\ell]}\}_{i\in[r]}.$$
- $\mathsf{P}^{\mathsf{CV}}_{\mathsf{Prove}}(\mathcal{B}_1, \ldots, \mathcal{B}_r, \mathsf{pp}) \to (\mathcal{B}_1, \ldots, \mathcal{B}_r, \{y_{i,j}, z_{i,j}\}_{i\in[r], j\in[\ell]})$:
  - Do the following for each $i \in [r]$: For each $j \in [\ell]$, apply $\mathsf{TCF.Eval}[\mathsf{pk}_{i,j}](\mathcal{B}_{i,j}) \to (\mathcal{B}_{i,j}, \mathcal{Z}_{i,j}, \mathcal{Y}_{i,j})$, resulting in the state
  $$\sum_{v\in\{0,1\}^\ell} \alpha_v |v\rangle^{\mathcal{B}_i} |\psi_{\mathsf{pk}_{i,1}, v_1}\rangle^{\mathcal{Z}_{i,1}, \mathcal{Y}_{i,1}}, \ldots, |\psi_{\mathsf{pk}_{i,\ell}, v_\ell}\rangle^{\mathcal{Z}_{i,\ell}, \mathcal{Y}_{i,\ell}},$$
  and measure registers $\mathcal{Y}_{i,1}, \ldots, \mathcal{Y}_{i,\ell}$ in the standard basis to obtain strings $y_{i,1}, \ldots, y_{i,\ell}$.
  - Compute $T := H(y_{1,1}, \ldots, y_{r,\ell})$, where $T \in \{0,1\}^r$ with Hamming weight k.
  - For each $i : T_i = 0$, measure $\mathcal{Z}_{i,1}, \ldots, \mathcal{Z}_{i,\ell}$ in the standard basis to obtain strings $z_{i,1}, \ldots, z_{i,\ell}$.
  - For each $i : T_i = 1$, apply $J(\cdot)$ coherently to each register $\mathcal{Z}_{i,1}, \ldots, \mathcal{Z}_{i,\ell}$ and then measure in the Hadamard basis to obtain strings $z_{i,1}, \ldots, z_{i,\ell}$.
- $\mathsf{P}^{\mathsf{CV}}_{\mathsf{Meas}}(\mathcal{B}_1, \ldots, \mathcal{B}_r) \to \{b_{i,j}\}_{i\in[r], j\in[\ell]}$: Measure registers $\{\mathcal{B}_{i,j}\}_{i:T_i=0, j\in[\ell]}$ in the standard basis to obtain bits $\{b_{i,j}\}_{i:T_i=0, j\in[\ell]}$ and measure registers $\{\mathcal{B}_{i,j}\}_{i:T_i=1, j\in[\ell]}$ in the Hadamard basis to obtain bits $\{b_{i,j}\}_{i:T_i=1, j\in[\ell]}$.
- $\mathsf{V}^{\mathsf{CV}}_{\mathsf{Ver}}(Q, x, \mathsf{sp}, \pi) \to \{\{\{q_{i,t}\}_{t\in[\lambda]}\}_{i:T_i=1}\} \cup \{\bot\}$:
  - Parse $\pi := \{b_{i,j}, y_{i,j}, z_{i,j}\}_{i\in[r], j\in[\ell]}$ and compute $T := H(y_{1,1}, \ldots, y_{r,\ell})$.
  - For each $i : T_i = 0$ and $j \in [\ell]$, compute $\mathsf{TCF.Check}(\mathsf{pk}_{i,j}, b_{i,j}, z_{i,j}, y_{i,j})$. If any are $\bot$, then output $\bot$.
  - For each $i : T_i = 1$, do the following.
    * For each $j \in [\ell]$: If $h_{i,j} = 0$, compute $\mathsf{TCF.Invert}(0, \mathsf{sk}_{i,j}, y_{i,j})$, output $\bot$ if the output is $\bot$, and otherwise parse the output as $(m_{i,j}, x_{i,j})$. If $h_{i,j} = 1$, compute $\mathsf{TCF.Invert}(1, \mathsf{sk}_{i,j}, y_{i,j})$, output $\bot$ if the output is $\bot$, and otherwise parse the output as $(0, x_{i,j,0}), (1, x_{i,j,1})$. Then, check $\mathsf{TCF.IsValid}(x_{i,j,0}, x_{i,j,1}, z_{i,j})$ and output $\bot$ if the result is $\bot$. Finally, set $m_{i,j} := b_{i,j} \oplus z_{i,j} \cdot (J(x_{i,j,0}) \oplus J(x_{i,j,1}))$.
    * Let $m_i = (m_{i,1}, \ldots, m_{i,\ell})$, compute $\mathsf{V}^{\mathsf{QV}}_{\mathsf{Ver}}(Q, x, h_i, m_i)$, output $\bot$ if the result is $\bot$, and otherwise set $\{q_{i,t}\}_{t\in[\lambda]} := m_i[S_i]$.
  - Output $\{\{q_{i,t}\}_{t\in[\lambda]}\}_{i:T_i=1}$.

Figure 5: Two-message protocol for verifying quantum partitioning circuits with a classical verifier.

Definition 5.7. *Let* MM_λ *be the predicate that takes as input a set of bits* $\{\{b_{i,t}\}_{t\in[\lambda]}\}_i$*, and outputs the bit*

$$\mathsf{MM}_\lambda(\{\{b_{i,t}\}_{t\in[\lambda]}\}_i) \coloneqq \mathsf{Maj}\left(\left\{\mathsf{Maj}\left(\{b_{i,t}\}_{t\in[\lambda]}\right)\right\}_i\right).$$

Lemma 5.8 (Completeness). *The protocol* Π^{CV} *(Fig. 5) with* $r(\lambda) = \lambda^2$ *and* $k(\lambda) = \lambda$ *satisfies* completeness, *which stipulates that for any family* $\{Q_\lambda, P_\lambda\}_{\lambda\in\mathbb{N}}$ *such that* $\{P_\lambda \circ Q_\lambda\}_{\lambda\in\mathbb{N}}$ *is pseudo-deterministic and sequence of inputs* $\{x_\lambda\}_{\lambda\in\mathbb{N}}$*,*

$$\Pr\left[\begin{array}{c} \mathsf{V}^{\mathsf{CV}}_{\mathsf{Ver}}(Q,x,\mathsf{sp},\pi) = \{\{q_{i,t}\}_{t\in[\lambda]}\}_{i:T_i=1} \ \wedge \\ \mathsf{MM}_\lambda(\{\{P(q_{i,t})\}_{t\in[\lambda]}\}_{i:T_i=1}) = P(Q(x)) \end{array} : \begin{array}{r} (\mathcal{B}_1,\dots,\mathcal{B}_r) \leftarrow \mathsf{P}^{\mathsf{CV}}_{\mathsf{Prep}}(1^\lambda, Q, x) \\ (\mathsf{pp},\mathsf{sp}) \leftarrow \mathsf{V}^{\mathsf{CV}}_{\mathsf{Gen}}(1^\lambda, Q) \\ \{y_{i,j}, z_{i,j}\}_{i\in[r],j\in[\ell]} \leftarrow \mathsf{P}^{\mathsf{CV}}_{\mathsf{Prove}}(\mathcal{B}_1,\dots,\mathcal{B}_r,\mathsf{pp}) \\ \{b_{i,j}\}_{i\in[r],j\in[\ell]} \leftarrow \mathsf{P}^{\mathsf{CV}}_{\mathsf{Meas}}(\mathcal{B}_1,\dots,\mathcal{B}_r) \\ \pi \coloneqq \{b_{i,j}, y_{i,j}, z_{i,j}\}_{i\in[r],j\in[\ell]} \end{array}\right] = 1 - \mathrm{negl}(\lambda).$$

Lemma 5.9 (Soundness). *The protocol* Π^{CV} *(Fig. 5) with* $r(\lambda) = \lambda^2$ *and* $k(\lambda) = \lambda$ *satisfies* soundness, *which stipulates that for any family* $\{Q_\lambda, P_\lambda\}_{\lambda\in\mathbb{N}}$ *such that* $\{P_\lambda \circ Q_\lambda\}_{\lambda\in\mathbb{N}}$ *is pseudo-deterministic, sequence of inputs* $\{x_\lambda\}_{\lambda\in\mathbb{N}}$*, and QPT adversary* $\{\mathsf{A}_\lambda\}_{\lambda\in\mathbb{N}}$*, it holds that*

$$\Pr\left[\begin{array}{c} \mathsf{V}^{\mathsf{CV}}_{\mathsf{Ver}}(Q,x,\mathsf{sp},\pi) = \{\{q_{i,t}\}_{t\in[\lambda]}\}_{i:T_i=1} \ \wedge \\ \mathsf{MM}_\lambda(\{\{P(q_{i,t})\}_{t\in[\lambda]}\}_{i:T_i=1}) = 1 - P(Q(x)) \end{array} : \begin{array}{r} (\mathsf{pp},\mathsf{sp}) \leftarrow \mathsf{V}^{\mathsf{CV}}_{\mathsf{Gen}}(1^\lambda, Q) \\ \pi \leftarrow \mathsf{A}(\mathsf{pp}) \end{array}\right] = \mathrm{negl}(\lambda).$$

Lemma 5.10 (D_{in} if accept). *The protocol* Π^{CV} *(Fig. 5) with* $r(\lambda) = \lambda^2$ *and* $k(\lambda) = \lambda$ *satisfies the following property. For any family* $\{Q_\lambda, P_\lambda\}_{\lambda\in\mathbb{N}}$ *such that* $\{P_\lambda \circ Q_\lambda\}_{\lambda\in\mathbb{N}}$ *is pseudo-deterministic, sequence of inputs* $\{x_\lambda\}_{\lambda\in\mathbb{N}}$*, and QPT adversary* $\{\mathsf{A}_\lambda\}_{\lambda\in\mathbb{N}}$*, it holds that*

$$\Pr\left[\begin{array}{l} \mathsf{V}^{\mathsf{CV}}_{\mathsf{Ver}}(Q,x,\mathsf{sp},\pi) \neq \bot \ \wedge \\ w \notin D_{\mathsf{in}}[P, P(Q(x))] \end{array} : \begin{array}{r} (\mathsf{pp},\mathsf{sp}) \leftarrow \mathsf{V}^{\mathsf{CV}}_{\mathsf{Gen}}(1^\lambda, Q) \\ \pi \leftarrow \mathsf{A}(\mathsf{pp}) \\ w \coloneqq \mathsf{TestRoundOutputs}[\mathsf{sp}](\pi) \end{array}\right] = \mathrm{negl}(\lambda).$$

Lemma 5.11 (D_{out} if accept wrong output). *The protocol* Π^{CV} *(Fig. 5) with* $r(\lambda) = \lambda^2$ *and* $k(\lambda) = \lambda$ *satisfies the following property. For any family* $\{Q_\lambda, P_\lambda\}_{\lambda\in\mathbb{N}}$ *such that* $\{P_\lambda \circ Q_\lambda\}_{\lambda\in\mathbb{N}}$ *is pseudo-deterministic, sequence of inputs* $\{x_\lambda\}_{\lambda\in\mathbb{N}}$*, and QPT adversary* $\{\mathsf{A}_\lambda\}_{\lambda\in\mathbb{N}}$*, it holds that*

$$\Pr\left[\begin{array}{l} \mathsf{V}^{\mathsf{CV}}_{\mathsf{Ver}}(Q,x,\mathsf{sp},\pi) = \{\{q_{i,t}\}_{t\in[\lambda]}\}_{i:T_i=1} \ \wedge \\ \mathsf{MM}_\lambda(\{\{P(q_{i,t})\}_{t\in[\lambda]}\}_{i:T_i=1}) = 1 - P(Q(x)) \ \wedge \\ w \notin D_{\mathsf{out}}[P, P(Q(x))] \end{array} : \begin{array}{r} (\mathsf{pp},\mathsf{sp}) \leftarrow \mathsf{V}^{\mathsf{CV}}_{\mathsf{Gen}}(1^\lambda, Q) \\ \pi \leftarrow \mathsf{A}(\mathsf{pp},\mathsf{sp}) \\ w \coloneqq \mathsf{TestRoundOutputs}[\mathsf{sp}](\pi) \end{array}\right] = \mathrm{negl}(\lambda).$$

Note that in this final lemma, A_λ is given access to sp, so this does not trivially follow from soundness.

5.4 Public verification

Next, we compile the above protocol into a *publicly-verifiable* protocol for quantum partitioning circuits in the oracle model. We will use the following ingredients in addition to Π^{CV} (Protocol 5).

- A Pauli functional commitment $\mathsf{PFC} = (\mathsf{PFC.Gen}, \mathsf{PFC.Com}, \mathsf{PFC.OpenZ}, \mathsf{PFC.OpenX}, \mathsf{PFC.DecZ}, \mathsf{PFC.DecX})$ that satisfies *string binding with public decodability* (Definition 4.4).

- A strongly unforgeable signature token scheme $\mathsf{Tok} = (\mathsf{Tok.Gen}, \mathsf{Tok.Sign}, \mathsf{Tok.Verify})$ (Definition 3.14).
- A pseudorandom function F_k secure against superposition-query attacks [Zha12].

Publicly-verifiable protocol $\Pi^{\mathsf{PV}} = (\mathsf{PV.Gen}, \mathsf{PV.Prove}, \mathsf{PV.Ver}, \mathsf{PV.Out})$

Parameters: Let λ be the security parameter and define parameters (ℓ, r, k) as in Π^{CV} (Fig. 5).

- $\mathsf{PV.Gen}(1^\lambda, Q) \to (\mathsf{vk}, |\mathsf{pk}\rangle, \mathsf{PK})$:
  - Sample $\{(\mathsf{dk}_{i,j}, |\mathsf{ck}_{i,j}\rangle, \mathsf{CK}_{i,j}) \leftarrow \mathsf{PFC.Gen}(1^\lambda)\}_{i\in[r],j\in[\ell]}$.
  - Sample $(\mathsf{vk}_{\mathsf{Tok}}, |\mathsf{sk}_{\mathsf{Tok}}\rangle) \leftarrow \mathsf{Tok.Gen}(1^\lambda)$.
  - Sample PRF keys $k_1, k_2 \leftarrow \{0,1\}^\lambda$.
  - Define the functionality $\mathsf{H}(\cdot) \coloneqq F_{k_1}(\cdot)$, which will be used as the random oracle H in Π^{CV}.
  - Define the functionality $\mathsf{CVGen}(\cdot)$ as follows, where its input is parsed as (x, c, σ).
    - If $\mathsf{Tok.Verify}(\mathsf{vk}_{\mathsf{Tok}}, (x,c), \sigma) = \top$ then continue, and otherwise return $\bot$.
    - Compute $(\mathsf{pp}, \mathsf{sp}) \coloneqq \mathsf{V}^{\mathsf{CV}}_{\mathsf{Gen}}(1^\lambda, Q; F_{k_2}(x, c, \sigma))$ and output pp.
  - Set $\mathsf{vk} \coloneqq (Q, k_1, k_2, \mathsf{vk}_{\mathsf{Tok}}, \{\mathsf{dk}_{i,j}\}_{i\in[r],j\in[\ell]})$, $|\mathsf{pk}\rangle \coloneqq (|\mathsf{sk}_{\mathsf{Tok}}\rangle, \{|\mathsf{ck}_{i,j}\rangle\}_{i\in[r],j\in[\ell]})$, and $\mathsf{PK} \coloneqq (\mathsf{H}, \mathsf{CVGen}, \{\mathsf{CK}_{i,j}\}_{i\in[r],j\in[\ell]})$.
- $\mathsf{PV.Prove}^{\mathsf{PK}}(|\mathsf{pk}\rangle, Q, x) \to \pi$:
  - Prepare $|\psi_1\rangle^{\mathcal{B}_1}, \ldots, |\psi_r\rangle^{\mathcal{B}_r} \leftarrow \mathsf{P}^{\mathsf{CV}}_{\mathsf{Prep}}(1^\lambda, Q, x)$.
  - For each $i \in [r], j \in [\ell]$ apply $\mathsf{PFC.Com}^{\mathsf{CK}_{i,j}}(\mathcal{B}_{i,j}, |\mathsf{ck}_{i,j}\rangle) \to (\mathcal{B}_{i,j}, \mathcal{U}_{i,j}, c_{i,j})$ (see Definition 4.1).
  - Set $c \coloneqq (c_{1,1}, \ldots, c_{r,\ell})$, compute $\sigma \leftarrow \mathsf{Tok.Sign}((x,c), |\mathsf{sk}_{\mathsf{Tok}}\rangle)$, and compute $\mathsf{pp} \coloneqq \mathsf{CVGen}(x, c, \sigma)$.
  - Apply $\mathsf{P}^{\mathsf{CV}}_{\mathsf{Prove}}(\mathcal{B}_1, \ldots, \mathcal{B}_r, \mathsf{pp}) \to (\mathcal{B}_1, \ldots, \mathcal{B}_r, \{y_{i,j}, z_{i,j}\}_{i\in[r],j\in[\ell]})$, and define $T \coloneqq \mathsf{H}(y_{1,1}, \ldots, y_{r,\ell})$
  - For each $i : T_i = 0, j \in [\ell]$, apply $\mathsf{PFC.OpenZ}(\mathcal{B}_{i,j}, \mathcal{U}_{i,j}) \to u_{i,j}$.
  - For each $i : T_i = 1, j \in [\ell]$, apply $\mathsf{PFC.OpenX}(\mathcal{B}_{i,j}, \mathcal{U}_{i,j}) \to u_{i,j}$.
  - Set $\pi \coloneqq (c, \sigma, \{u_{i,j}, y_{i,j}, z_{i,j}\}_{i\in[r],j\in[\ell]})$.
- $\mathsf{PV.Ver}(\mathsf{vk}, x, \pi) \to \{\{\{q_{i,t}\}_{t\in[\lambda]}\}_{i:T_i=1}\} \cup \{\bot\}$:
  - Parse $\mathsf{vk} \coloneqq (Q, k_1, k_2, \mathsf{vk}_{\mathsf{Tok}}, \{\mathsf{dk}_{i,j}\}_{i\in[r],j\in[\ell]})$ and $\pi \coloneqq (c, \sigma, \mu)$.
  - If $\mathsf{Tok.Verify}(\mathsf{vk}_{\mathsf{Tok}}, (x,c), \sigma) = \top$, then set $(\mathsf{pp}, \mathsf{sp}) \coloneqq \mathsf{V}^{\mathsf{CV}}_{\mathsf{Gen}}(1^\lambda, Q; F_{k_2}(x, c, \sigma))$, and let $\{h_i\}_{i\in[r]}$ be the string of basis choices defined by sp. Otherwise, return $\bot$.
  - Parse μ as $\{u_{i,j}, y_{i,j}, z_{i,j}\}_{i\in[r],j\in[\ell]}$, and define $T \coloneqq F_{k_1}(y_{1,1}, \ldots, y_{r,\ell})$.
  - For all $i : T_i = 0, j \in [\ell]$, compute $b_{i,j} \coloneqq \mathsf{PFC.DecZ}(\mathsf{dk}_{i,j}, c_{i,j}, u_{i,j})$, and return $\bot$ if $b_{i,j} = \bot$.
  - For all $i : T_i = 1, j \in [\ell]$ such that $h_{i,j} = 1$, compute $b_{i,j} \coloneqq \mathsf{PFC.DecX}(\mathsf{dk}_{i,j}, c_{i,j}, u_{i,j})$, and return $\bot$ if $b_{i,j} = \bot$.
  - For all $i : T_i = 1, j \in [\ell]$ such that $h_{i,j} = 0$, set $b_{i,j} = 0$.
  - Let $\widetilde{\pi} \coloneqq \{b_{i,j}, y_{i,j}, z_{i,j}\}_{i\in[r],j\in[\ell]}$ and return $\{\{q_{i,t}\}_{t\in[\lambda]}\}_{i:T_i=1} \coloneqq \mathsf{V}^{\mathsf{CV}}_{\mathsf{Ver}}(Q, x, \mathsf{sp}, \widetilde{\pi})$.
- $\mathsf{PV.Combine} \equiv \mathsf{MM}_\lambda$ (see Definition 5.7).

Figure 6: Publicly-verifiable non-interactive classical verification of quantum partitioning circuits.

Theorem 5.12. *The protocol Π^{PV} (Fig. 6) satisfies Definition 5.1 and Definition 5.2.*

Proof. We argue completeness (Definition 5.1) and soundness (Definition 5.2).

Completeness. Consider some circuit Q, input x, and sample $(\mathsf{vk}, |\mathsf{pk}\rangle, \mathsf{PK}) \leftarrow \mathsf{PV.Gen}(1^\lambda, Q)$. By the correctness of Tok (Definition 3.11), we know that the call to CVGen during $\mathsf{PV.Prove}^{\mathsf{PK}}(|\mathsf{pk}\rangle, Q, x)$ only outputs $\bot$ with $\mathrm{negl}(\lambda)$ probability. Also, by the security of the PRF, we can answer this query using uniformly sampled random coins s in place of $F_{k_2}(x, c, \sigma)$.

Now, imagine sampling s and fixing $(\mathsf{pp}, \mathsf{sp}) := \mathsf{V}^{\mathsf{CV}}_{\mathsf{Gen}}(1^\lambda, Q; s)$ before computing $\mathsf{PV.Prove}^{\mathsf{PK}}(|\mathsf{pk}\rangle, Q, x)$. Then, since pp no longer depends on c, we can move the application of each $\mathsf{PFC.Com}^{\mathsf{CK}_{i,j}}(\mathcal{B}_{i,j}, |\mathsf{ck}_{i,j}\rangle)$ past the computation of pp, and thus right before $\mathsf{P}^{\mathsf{CV}}_{\mathsf{Prove}}(\mathcal{B}_1, \ldots, \mathcal{B}_r, \mathsf{pp})$. Moreover, since both PFC.Com and $\mathsf{P}^{\mathsf{CV}}_{\mathsf{Prove}}$ are *classically controlled* on registers $\mathcal{B}_1, \ldots, \mathcal{B}_r$, and otherwise operate on disjoint registers, we can further commute each PFC.Com past $\mathsf{P}^{\mathsf{CV}}_{\mathsf{Prove}}$.

Then, the bits $\{b_{i,j}\}_{i,j}$ for $i : T_i = 0$ computed during $\mathsf{PV.Ver}(\mathsf{vk}, x, \pi)$ are now computed by applying PFC.Com, PFC.OpenZ, and PFC.DecZ in succession to $\mathcal{B}_{i,j}$, and the bits $\{b_{i,j}\}_{i,j}$ for $i : T_i = 1, h_{i,j} = 1$ computed during $\mathsf{PV.Ver}(\mathsf{vk}, x, \pi)$ are now computed by applying PFC.Com, PFC.OpenX, and PFC.DecX in succession to $\mathcal{B}_{i,j}$. Thus, by the correctness of PFC (Definition 4.2), we can replace these operations by directly measuring $\mathcal{B}_{i,j}$ in the standard (resp. Hadamard) basis. Now, completeness follows directly from the completeness of Π^{CV} (Lemma 5.8), since the remaining bits $\{b_{i,j}\}_{i,j}$ for $i : T_i = 1, h_{i,j} = 0$ (which are arbitrarily set to 0 in PV.Ver) are ignored by $\mathsf{V}^{\mathsf{CV}}_{\mathsf{Ver}}$, and the rest of $\widetilde{\pi}$ is now computed by applying $\mathsf{P}^{\mathsf{CV}}_{\mathsf{Prove}}$ followed by $\mathsf{P}^{\mathsf{CV}}_{\mathsf{Meas}}$ to $\mathcal{B}_1, \ldots, \mathcal{B}_r$.

Soundness. Before getting into the formal proof, we provide a high-level overview. We will go via the following steps.

- A_1: Begin with an adversary A_1 that is assumed to violate soundness of the protocol. Thus, with non-negl(λ) probability, it's final (classical) output consists of an input x^* and a proof π^* such that $\mathsf{PV.Ver}(\mathsf{vk}, x^*, \pi^*) \neq \bot$ and $\mathsf{PV.Out}(\mathsf{PV.Ver}(\mathsf{vk}, x^*, \pi^*), P) \neq P(Q(x^*))$.
- A_2: Replace F_{k_2} with a random oracle, and call the resulting oracle algorithm A_2.
- A_3: Apply Measure-and-Reprogram (Imported Theorem 3.10) to obtain a two-stage adversary A_3, where the first stage outputs x^*, a PFC commitment c^*, and a token signature σ^*, and the second stage outputs the remainder μ^* of the proof $\pi^* := (c^*, \sigma^*, \mu^*)$. The parameters $(\mathsf{pp}_{x^*,c^*,\sigma^*}, \mathsf{sp}_{x^*,c^*,\sigma^*})$ for Π^{CV} are re-sampled at the beginning of the second stage.
- A_4: Use the strong unforgeability of the signature token scheme (Definition 3.14) to argue that during the second stage of A_3, all queries to PV.Ver except for (x^*, c^*, σ^*) can be ignored. Call the resulting adversary A_4.
- $D_{\mathsf{out}}[P, P(Q(x^*))]$: Appeal to Lemma 5.11 to show that whenever A_4 breaks soundness, its output yields a proof $\widetilde{\pi}$ for Π^{CV} such that

$$\mathsf{TestRoundOutputs}[\mathsf{sp}_{x^*,c^*,\sigma^*}](\widetilde{\pi}) \in D_{\mathsf{out}}[P, P(Q(x^*))].$$

- $\mathcal{H}_0, \ldots, \mathcal{H}_p$: Define a hybrid for each of the $p = \mathrm{poly}(\lambda)$ queries that the second stage of A_4 makes to PV.Ver. In each hybrid ι, begin answering query ι with $\bot$, and let $\Pr[\mathcal{H}_\iota = 1]$ be the probability that A_4 still breaks soundness.

- $\Pr[\mathcal{H}_0 = 1] = \mathsf{non\text{-}negl}(\lambda)$: This has already been proven, by assumption that A_1 breaks soundness with $\mathsf{non\text{-}negl}(\lambda)$ probability, and the hybrids above.

- $\Pr[\mathcal{H}_p = 1] = \mathrm{negl}(\lambda)$: This is implied by the soundness of Π^{CV} (Lemma 5.9) because in this experiment, A_4 does not have access to $\mathsf{sp}_{x^*,c^*,\sigma^*}$ before producing its final proof.

- $\Pr[\mathcal{H}_\iota = 1] \geq \Pr[\mathcal{H}_{\iota-1} = 1] - \mathrm{negl}(\lambda)$: This is proven in two parts.

  1. By Lemma 5.10, we can say that since A_4 does not have access to $\mathsf{sp}_{x^*,c^*,\sigma^*}$ before preparing its ι'th query, each classical basis state in the query superposition that is not answered with $\bot$ yields a proof $\widetilde{\pi}$ for Π^{CV} such that

  $$\mathsf{TestRoundOutputs}[\mathsf{sp}_{x^*,c^*,\sigma^*}](\widetilde{\pi}) \in D_{\mathsf{in}}[P, P(Q(x^*))].$$

  2. We appeal to the string binding with public decodability of PFC (Definition 4.4) to show that replacing these answers with $\bot$ only affects the probability that A_4 breaks soundness by a negligible amount.

     This follows because any part of the query that contains PFC openings for a string in $D_{\mathsf{in}}[P, P(Q(x^*))]$ cannot have noticeable overlap with the part of the state (after running the rest of A_4) that contains PFC openings for a string in $D_{\mathsf{out}}[P, P(Q(x^*))]$. Otherwise, we can prepare an adversarial committer, where the part of A_4 up to query ι is the "Commit" stage, and the remainder of A_4 is the "Open" stage. Crucially, since all queries to $\mathsf{PV.Ver}$ except (x^*, c^*, σ^*) are ignored during the Open stage, we do not have to give the Open stage access to the receiver's Hadamard basis decoding functionalities on the indices that are checked by $D_{\mathsf{in}}[P, P(Q(x^*))]$ and $D_{\mathsf{out}}[P, P(Q(x^*))]$, which are all standard basis positions with respect to the parameters $(\mathsf{pp}_{x^*,c^*,\sigma^*}, \mathsf{sp}_{x^*,c^*,\sigma^*})$.

- This completes the proof, as the previous three bullet points produce a contradiction.

Now we provide the formal proof. Suppose there exists Q, P and $\mathsf{A}_1^{\mathsf{PK},\mathsf{PV.Ver}[\mathsf{vk}]}$ that violates Definition 5.2, where we have dropped the indexing by λ for convenience. Our first step will be to replace the PRF $F_{k_2}(\cdot)$ with a random oracle G. Note that A_1 only has polynomially-bounded oracle access to this functionality, so this has a negligible affect on the output of A_1 [Zha12]. This defines an oracle algorithm A_2^G based on $\mathsf{A}_1^{\mathsf{PK},\mathsf{PV.Ver}[\mathsf{vk}]}$ that operates as follows.

- Sample $(\mathsf{vk}, |\mathsf{pk}\rangle, \mathsf{PK})$ as in $\mathsf{PV.Gen}(1^\lambda, Q)$, except $F_{k_2}(\cdot)$ is replaced with $G(\cdot)$.

- Run $\mathsf{A}_1^{\mathsf{PK},\mathsf{PV.Ver}[\mathsf{vk}]}(|\mathsf{pk}\rangle)$, forwarding calls to G (which occur as part of calls to CVGen and $\mathsf{PV.Ver}[\mathsf{vk}]$) to the external random oracle G.

- Measure A_1's output (x^*, π^*), parse π^* as (c^*, σ^*, μ^*) and output $a \coloneqq (x^*, c^*, \sigma^*)$ and $\mathsf{aux} \coloneqq (\mu^*, \mathsf{vk})$.

Note that A_2 makes $p = \mathrm{poly}(\lambda)$ total queries to G, since A_1 makes $\mathrm{poly}(\lambda)$ queries. Now, define V as in Fig. 7. Then since A_1 breaks soundness,

$$\Pr\left[V(a, G(a), \mathsf{aux}) = 1 : (a, \mathsf{aux}) \leftarrow \mathsf{A}_2^G\right] = \mathsf{non\text{-}negl}(\lambda).$$

Functionalities used in the proof of Theorem 5.12

Fixed parameters: Security parameter λ, circuit Q, and predicate P.

- $\mathsf{PV.Ver}[\mathsf{vk}](x, \pi)$: Same as $\mathsf{PV.Ver}(\mathsf{vk}, x, \pi)$.
- $\mathsf{PV.Ver}[\mathsf{vk}, s](x, \pi)$: Same as $\mathsf{PV.Ver}[\mathsf{vk}](x, \pi)$ except that s is used instead of $F_{k_2}(x, c, \sigma)$ when generating $(\mathsf{pp}, \mathsf{sp}) := \mathsf{V}^{\mathsf{CV}}_{\mathsf{Gen}}(1^\lambda, Q; s)$.
- $\mathsf{PV.Ver}[\mathsf{vk}, s, (x^*, c^*, \sigma^*)](x, \pi)$: Same as $\mathsf{PV.Ver}[\mathsf{vk}, s](x, \pi)$, except that after the input is parsed as x and $\pi := (c, \sigma, \mu)$, output $\bot$ if
$$(x, c, \sigma) \neq (x^*, c^*, \sigma^*).$$
- $\mathsf{PV.Ver}[\mathsf{vk}, s, (x^*, c^*, \sigma^*), \mathsf{in}](x, \pi)$: Same as $\mathsf{PV.Ver}[\mathsf{vk}, s, (x^*, c^*, \sigma^*)](x, \pi)$ except that after $\widetilde{\pi} := \{b_{i,j}, y_{i,j}, z_{i,j}\}_{i \in [r], j \in [\ell]}$ has been computed, output $\bot$ if
$$\mathsf{TestRoundOutputs}[\mathsf{sp}]\,(\widetilde{\pi}) \notin D_{\mathsf{in}}[P, P(Q(x))].$$
- $\mathsf{PV.Ver}[\mathsf{vk}, s, (x^*, c^*, \sigma^*), \mathsf{out}](x, \pi)$: Same as $\mathsf{PV.Ver}[\mathsf{vk}, s, (x^*, c^*, \sigma^*)](x, \pi)$ except that after $\widetilde{\pi} := \{b_{i,j}, y_{i,j}, z_{i,j}\}_{i \in [r], j \in [\ell]}$ has been computed, output $\bot$ if
$$\mathsf{TestRoundOutputs}[\mathsf{sp}]\,(\widetilde{\pi}) \notin D_{\mathsf{out}}[P, P(Q(x))].$$
- $V(a, s, \mathsf{aux})$:
  - Parse $a := (x^*, c^*, \sigma^*)$ and $\mathsf{aux} := (\mu^*, \mathsf{vk})$.
  - Compute $q := \mathsf{PV.Ver}[\mathsf{vk}, s](x^*, (c^*, \sigma^*, \mu^*))$.
  - Output 1 iff $q \neq \bot$ and $\mathsf{PV.Out}(q, P) = 1 - P(Q(x))$.
- $V[\mathsf{out}](a, s, \mathsf{aux})$:
  - Parse $a := (x^*, c^*, \sigma^*)$ and $\mathsf{aux} := (\mu^*, \mathsf{vk})$.
  - Compute $q := \mathsf{PV.Ver}[\mathsf{vk}, s, (x^*, c^*, \sigma^*), \mathsf{out}](x^*, (c^*, \sigma^*, \mu^*))$.
  - Output 1 iff $q \neq \bot$ and $\mathsf{PV.Out}(q, P) = 1 - P(Q(x))$.

Figure 7: Description of functionalities used in the proof of Theorem 5.12.

Next, since $p = \mathrm{poly}(\lambda)$, by Imported Theorem 3.10 there exists an algorithm $\mathsf{A}_3 := \mathsf{Sim}[\mathsf{A}_2]$ such that

$$\Pr\left[V((x^*, c^*, \sigma^*), s, (\mu^*, \mathsf{vk})) = 1 : \begin{array}{r} ((x^*, c^*, \sigma^*), \mathsf{state}) \leftarrow \mathsf{A}_3 \\ s \leftarrow \{0,1\}^\lambda \\ (\mu^*, \mathsf{vk}) \leftarrow \mathsf{A}_3(s, \mathsf{state}) \end{array}\right] = \mathsf{non\text{-}negl}(\lambda).$$

Moreover, A_3 operates as follows.

- Sample G as a $2p$-wise independent function and $(i, d) \leftarrow (\{0, \dots, p-1\} \times \{0, 1\}) \cup \{(p, 0)\}$.
- Run A_2 for i oracle queries, answering each query using the function G.
- When A_2 is about to make its $(i+1)$'th oracle query, measure its query register in the standard basis to obtain $a := (x^*, c^*, \sigma^*)$. In the special case that $(i, d) = (p, 0)$, just measure (part of) the final output register of A_2 to obtain a.
- Receive s externally.

- If $d = 0$, answer A_2's $(i+1)$'th query with G. If $d = 1$, answer A_2's $(i+1)$'th query instead with $G[(x^*, c^*, \sigma^*) \to s]$.
- Run A_2 until it has made all p queries to G. For queries $i+2$ through p, answer with $G[(x^*, c^*, \sigma^*) \to s]$.
- Measure A_2's output $\mathsf{aux} := (\mu^*, \mathsf{vk})$.

Recall that A_3 is internally running A_1, who expects oracle access to H, CVGen, $\{\mathsf{CK}_{i,j}\}_{i,j}$ and $\mathsf{PV.Ver}[\mathsf{vk}]$. These oracle queries will be answered by A_3. Next, we define A_4 to be the same as A_3, except that after (x^*, c^*, σ^*) is measured by A_3, A_1's queries to $\mathsf{PV.Ver}[\mathsf{vk}]$ are answered instead with $\mathsf{PV.Ver}[\mathsf{vk}, s, (x^*, c^*, \sigma^*)]$ from Fig. 7.

Claim 5.13.

$$\Pr\left[V((x^*, c^*, \sigma^*), s, (\mu^*, \mathsf{vk})) = 1 : \begin{array}{r} ((x^*, c^*, \sigma^*), \mathsf{state}) \leftarrow \mathsf{A}_4 \\ s \leftarrow \{0,1\}^\lambda \\ (\mu^*, \mathsf{vk}) \leftarrow \mathsf{A}_4(s, \mathsf{state}) \end{array}\right] = \mathsf{non\text{-}negl}(\lambda).$$

Proof. We can condition on $\mathsf{Tok.Ver}(\mathsf{vk}_{\mathsf{Tok}}, (x^*, c^*), \sigma^*) = \top$, since otherwise V would output 0. Then, by the strong unforgeability of Tok (Definition 3.14), once (x^*, c^*, σ^*) is measured, A_1 cannot produce any query that has noticeable amplitude on any (x, c, σ) such that

$$(x, c, \sigma) \neq (x^*, c^*, \sigma^*) \text{ and } \mathsf{Tok.Ver}(\mathsf{vk}_{\mathsf{Tok}}, (x, c), \sigma) = \top.$$

But after (x^*, c^*, σ^*) is measured and s is sampled, $\mathsf{PV.Ver}[\mathsf{vk}]$ and $\mathsf{PV.Ver}[\mathsf{vk}, s, (x^*, c^*, \sigma^*)]$ can only differ on (x, c, σ) such that

$$(x, c, \sigma) \neq (x^*, c^*, \sigma^*) \text{ and } \mathsf{Tok.Ver}(\mathsf{vk}_{\mathsf{Tok}}, (x, c), \sigma) = \top.$$

Thus, since A_1 only has polynomially-many queries, changing the oracle in this way can only have a negligible affect on the final probability, which completes the proof. $\square$

Next, we claim the following, where $V[\mathsf{out}]$ is defined in Protocol 7.

Claim 5.14.

$$\Pr\left[V[\mathsf{out}]((x^*, c^*, \sigma^*), s, (\mu^*, \mathsf{vk})) = 1 : \begin{array}{r} ((x^*, c^*, \sigma^*), \mathsf{state}) \leftarrow \mathsf{A}_4 \\ s \leftarrow \{0,1\}^\lambda \\ (\mu^*, \mathsf{vk}) \leftarrow \mathsf{A}_4(s, \mathsf{state}) \end{array}\right] = \mathsf{non\text{-}negl}(\lambda).$$

Proof. First, if we replace the PRF $F_{k_1}(\cdot)$ with an external random oracle H, then the probabilities in Claim 5.13 and Claim 5.14 remain the same up to a negligible difference [Zha12]. Next, note that the only event that differentiates Claim 5.13 and Claim 5.14 is when A_4 outputs $(x^*, c^*, \sigma^*, \mu^*)$ such that

$$q \neq \bot \;\wedge\; \mathsf{Out}_\lambda[P](q) = 1 - P(Q(x^*)) \;\wedge\; \mathsf{TestRoundOutputs}[\mathsf{sp}](\widetilde{\pi}) \notin D_{\mathsf{out}}[P, P(Q(x^*))],$$

where $(\mathsf{pp}, \mathsf{sp}) := \mathsf{V}^{\mathsf{CV}}_{\mathsf{Gen}}(1^\lambda, Q; s)$, $\widetilde{\pi} := \{b_{i,j}, y_{i,j}, z_{i,j}\}_{i \in [r], j \in [\ell]}$ is computed during

$$\mathsf{PV.Ver}[\mathsf{vk}, s, (x^*, c^*, \sigma^*)](x^*, (c^*, \sigma^*, \mu^*)),$$

and $q := \mathsf{V}^{\mathsf{CV}}_{\mathsf{Ver}}(Q, x^*, \mathsf{sp}, \widetilde{\pi})$. If this event occurs with noticeable probability, there must be some fixed x^* such that it occurs with noticeable probability conditioned on x^*. However, this would contradict Lemma 5.11. Thus, the difference in probability must be negligible, completing the proof. $\square$

Finally, we will define a sequence of hybrids $\{\mathcal{H}_\iota\}_{\iota\in[0,p]}$ based on A_4. Hybrid $\mathcal{H}_\iota$ is defined as follows.

- Run $((x^*, c^*, \sigma^*), \mathsf{state}) \leftarrow \mathsf{A}_4$.
- Sample $s \leftarrow \{0,1\}^\lambda$.
- Run $(\mu^*, \mathsf{vk}) \leftarrow \mathsf{A}_4(s, \mathsf{state})$ with the following difference. Recall that at some point, A_4 begins using the oracle $G[(x^*, c^*, \sigma^*) \to s]$ while answering A_1's queries. For the first ι times that A_1 queries $\mathsf{PV.Ver}[\mathsf{vk}, s, (x^*, c^*, \sigma^*)]$ after this point, respond using the oracle $O_\perp$ that outputs $\perp$ on every input.
- Output $V[\mathsf{out}]((x^*, c^*, \sigma^*), s, (\mu^*, \mathsf{vk}))$.

Note that Claim 5.14 is stating exactly that $\Pr[\mathcal{H}_0 = 1] = \mathsf{non\text{-}negl}(\lambda)$. Next, we have the following claim.

Claim 5.15. $\Pr[\mathcal{H}_p = 1] = \mathrm{negl}(\lambda)$.

Proof. First, if we replace the PRF $F_{k_1}(\cdot)$ with an external random oracle H, then the probability remains the same up to a negligible difference [Zha12]. Now, the claim follows by a reduction to the soundness of Π^{CV} (Lemma 5.9). Note that A_4 never needs to know the sp such that $(\mathsf{pp}, \mathsf{sp}) := \mathsf{V}^{\mathsf{CV}}_{\mathsf{Gen}}(1^\lambda, Q; s)$, since all of the (at most p) calls that A_1 makes to $\mathsf{PV.Ver}[\mathsf{vk}, s, (x^*, c^*, \sigma^*)]$ once G is programmed so that $G[(x^*, c^*, \sigma^*) \to s]$ are answered with $O_\perp$. Thus, we can view A_4^H as an adversarial prover for Π^{CV}, where the first stage of A_4^H outputs x^*, and the second stage receives pp and outputs $\widetilde{\pi} := \{b_{i,j}, y_{i,j}, z_{i,j}\}_{i,j}$ (which can be computed from μ^*). By the definition of the predicate $V[\mathsf{out}]$, the probability that $\mathcal{H}_p = 1$ is at most the probability that $\mathsf{MM}_\lambda[P](q) = 1 - P(Q(x))$, where $q := \mathsf{V}^{\mathsf{CV}}_{\mathsf{Ver}}(Q, x^*, \mathsf{sp}, \widetilde{\pi})$, which by Lemma 5.9 must be $\mathrm{negl}(\lambda)$. $\square$

Finally, we prove the following Claim 5.16. Since $p = \mathrm{poly}(\lambda)$, this contradicts Claim 5.14 and Claim 5.15, which completes the proof.

$\square$

Claim 5.16. *For any* $\iota \in [p]$, $\Pr[\mathcal{H}_\iota = 1] \geq \Pr[\mathcal{H}_{\iota-1} = 1] - \mathrm{negl}(\lambda)$.

Proof. Throughout this proof, when we refer to "query ι" in some hybrid, we mean the ι'th query that A_1 makes to $\mathsf{PV.Ver}[\mathsf{vk}, x, (x^*, c^*, \sigma^*)]$ after A_4 has begun using the oracle $G[(x^*, c^*, \sigma^*) \to s]$ (if such a query exists).

Now, we introduce an intermediate hybrid $\mathcal{H}'_{\iota-1}$ which is the same as $\mathcal{H}_{\iota-1}$ except that query ι is answered with the functionality $\mathsf{PV.Ver}[\mathsf{vk}, s, (x^*, c^*, \sigma^*), \mathsf{in}]$ defined in Protocol 7.

So, it suffices to show that

- $\Pr[\mathcal{H}'_{\iota-1} = 1] \geq \Pr[\mathcal{H}_{\iota-1} = 1] - \mathrm{negl}(\lambda)$, and
- $\Pr[\mathcal{H}_\iota = 1] \geq \Pr[\mathcal{H}'_{\iota-1} = 1] - \mathrm{negl}(\lambda)$.

We note that the only difference between the three hybrids is how query ι is answered:

- In $\mathcal{H}_{\iota-1}$, query ι is answered with $\mathsf{PV.Ver}[\mathsf{vk}, s, (x^*, c^*, \sigma^*)]$.
- In $\mathcal{H}'_{\iota-1}$, query ι is answered with $\mathsf{PV.Ver}[\mathsf{vk}, s, (x^*, c^*, \sigma^*), \mathsf{in}]$.
- In $\mathcal{H}_{\iota}$, query ι is answered with $O_{\perp}$.

Now, the proof is completed by appealing to the following two claims. □

Claim 5.17. $\Pr[\mathcal{H}'_{\iota-1} = 1] \geq \Pr[\mathcal{H}_{\iota-1} = 1] - \mathrm{negl}(\lambda)$.

Proof. First, if we replace the PRF $F_{k_1}(\cdot)$ with an external random oracle H, then $\Pr[\mathcal{H}_{\iota-1} = 1]$ and $\Pr[\mathcal{H}'_{\iota-1} = 1]$ remain the same up to negligible difference [Zha12]. Now, this follows from a reduction to Lemma 5.10. Indeed, note that if $|\Pr[\mathcal{H}'_{\iota-1} = 1] - \Pr[\mathcal{H}_{\iota-1} = 1]| = \mathsf{non\text{-}negl}(\lambda)$, then in $\mathcal{H}_{\iota-1}$, A_1's ι'th query must have noticeable amplitude on $(x^*, \pi^* = (c^*, \sigma^*, \mu^*))$ such that

$$q \neq \perp \ \wedge \ \mathsf{TestRoundOutputs}[\mathsf{sp}](\widetilde{\pi}) \notin D_{\mathsf{in}}[P, P(Q(x^*))],$$

where $(\mathsf{pp}, \mathsf{sp}) := \mathsf{V}^{\mathsf{CV}}_{\mathsf{Gen}}(1^\lambda, Q; s)$, $\widetilde{\pi} := \{b_{i,j}, y_{i,j}, z_{i,j}\}_{i \in [r], j \in [\ell]}$ is computed during

$$\mathsf{PV.Ver}[\mathsf{vk}, s, (x^*, c^*, \sigma^*)](x^*, (c^*, \sigma^*, \mu^*)),$$

and $q := \mathsf{V}^{\mathsf{CV}}_{\mathsf{Ver}}(Q, x^*, \mathsf{sp}, \widetilde{\pi})$. However, A_4 never needs to know sp prior to this query, since all of the calls that A_1 makes to $\mathsf{PV.Ver}[\mathsf{vk}, s, (x^*, c^*, \sigma^*)]$ once G is programmed so that $G[(x^*, c^*, \sigma^*) \to s]$ are answered with $O_{\perp}$. Thus, we can view A_4^H has an adversarial prover for Π^{CV}, where the first part of A_4^H outputs x^*, and the second part receives pp and outputs $\widetilde{\pi} := \{b_{i,j}, y_{i,j}, z_{i,j}\}_{i,j}$ (which can be computed from μ^*). Then, by Lemma 5.10, the above event occurs with negligible probability.

□

Claim 5.18. $\Pr[\mathcal{H}_{\iota} = 1] \geq \Pr[\mathcal{H}'_{\iota-1} = 1] - \mathrm{negl}(\lambda)$

Proof. We will show this by reduction to the string binding with public decodability property of PFC. Recall from Section 5.3 that based on any $(\mathsf{pp}, \mathsf{sp}) \in \mathsf{V}^{\mathsf{CV}}_{\mathsf{Gen}}(1^\lambda, Q)$, we define a subset of indices $S := \{S_i\}_{i \in [r]} \subset [r] \times [\ell]$ by the subsets $\{S_i\}_{i \in [r]}$ defined by sp. This subset S is used in turn to define the predicates $D_{\mathsf{in}}[P, b]$ and $D_{\mathsf{out}}[P, b]$. Throughout this proof, we will always let S be defined based on $(\mathsf{pp}, \mathsf{sp}) := \mathsf{V}^{\mathsf{CV}}_{\mathsf{Gen}}(1^\lambda, Q; s)$, where the coins s will always be clear from context. We also define $m := |S|$, which we assume is the same for all coins s.

Now we define an oracle-aided operation C as follows.

- C takes as input $\{|\mathsf{ck}_\tau\rangle\}_{\tau \in [m]}$, where $\{\mathsf{dk}_\tau, |\mathsf{ck}_\tau\rangle, \mathsf{CK}_\tau \leftarrow \mathsf{PFC.Gen}(1^\lambda)\}_{\tau \in [m]}$.
- C samples $s \leftarrow \{0,1\}^\lambda$ and sets $(\mathsf{pp}, \mathsf{sp}) := \mathsf{V}^{\mathsf{CV}}_{\mathsf{Gen}}(1^\lambda, Q; s)$. For $(i,j) \notin S$, sample $\mathsf{dk}_{i,j}, |\mathsf{ck}_{i,j}\rangle, \mathsf{CK}_{i,j} \leftarrow \mathsf{PFC.Gen}(1^\lambda)$. Let $f : [m] \to S$ be an arbitrary bijection, and re-define $\{\mathsf{dk}_\tau, |\mathsf{ck}_\tau\rangle, \mathsf{CK}_\tau\}_{\tau \in [m]}$ as $\{\mathsf{dk}_{f(\tau)}, |\mathsf{ck}_{f(\tau)}\rangle, \mathsf{CK}_{f(\tau)}\}_{\tau \in [m]}$.
- C runs A_4 as defined by $\mathcal{H}'_{\iota-1}$ until right before query ι is answered. All queries to $\mathsf{CK}_{i,j}$, $\mathsf{PFC.DecZ}[\mathsf{dk}_{i,j}]$, or $\mathsf{PFC.DecX}[\mathsf{dk}_{i,j}]$ for $(i,j) \in S$ are forwarded to external oracles.

That is, we can write the operation of C as

$$|\psi\rangle \leftarrow \mathsf{C}^{\mathbf{CK},\mathsf{PFC.DecZ}[\mathbf{dk}],\mathsf{PFC.DecX}[\mathbf{dk}]}(|\mathbf{ck}\rangle),$$

where $\mathbf{CK}$ is the collection oracles $\mathsf{CK}_1,\ldots,\mathsf{CK}_m$, $|\mathbf{ck}\rangle = (|\mathsf{ck}_1\rangle,\ldots,|\mathsf{ck}_m\rangle)$, $\mathsf{PFC.DecZ}[\mathbf{dk}]$ is the collection of oracles $\mathsf{PFC.DecZ}[\mathsf{dk}_1],\ldots,\mathsf{PFC.DecZ}[\mathsf{dk}_m]$, and $\mathsf{PFC.DecX}[\mathbf{dk}]$ is the collection of oracles $\mathsf{PFC.DecX}[\mathsf{dk}_1],\ldots,\mathsf{PFC.DecX}[\mathsf{dk}_m]$.

Next, we define an oracle-aided unitary U as follows.

- U takes as input the state $|\psi\rangle$ output by C.
- It coherently runs the remainder of A_4 as defined by $\mathcal{H}'_{\iota-1}$. Any queries to $\mathsf{CK}_{i,j}$ or $\mathsf{PFC.DecZ}[\mathsf{dk}_{i,j}]$ for $(i,j)\in S$ are forwarded to external oracles. Note that this portion of A_4 does not require access to the Hadamard basis decoding oracles $\mathsf{PFC.DecX}[\mathsf{dk}_{i,j}]$ for $(i,j)\in S$. This follows because for each such (i,j), $h_{i,j}=0$, which means that $\mathsf{PV.Ver}[\mathsf{vk},s,(x^*,c^*,\sigma^*),\mathsf{in}]$ only requires access to the standard basis decoding oracles at these positions.

That is, we can write the operation of U as

$$|\psi'\rangle := \mathsf{U}^{\mathbf{CK},\mathsf{PFC.DecZ}[\mathbf{dk}]}(|\psi\rangle).$$

Now, we give a name to three registers of the space operated on by U, as follows.

- $\mathcal{Q}$ is the query register for A_1's ι'th query. That is, the state $|\psi\rangle$ contains a superposition over strings (x,π) on register $\mathcal{Q}$.
- $\mathcal{A}$ holds classical information $(\mathsf{vk},s,x^*,c^*,\sigma^*)$ that has been sampled previously by C. Thus, the state $|\psi\rangle$ contains a standard basis state on register $\mathcal{A}$, and U is classically controlled on this register.
- $\mathcal{V}$ is the register that is measured to produce the string μ^* output at the end of A_4's operation. Thus, the state $|\psi'\rangle$ contains a superposition over μ^* on register $\mathcal{V}$.

We also define $\widetilde{\mathsf{U}}$ to be identical to U except that it runs the remainder of A_4 as defined by $\mathcal{H}_\iota$. Note that the only difference between U and $\widetilde{\mathsf{U}}$ is how query ι is answered at the very beginning.

Next, we define the following two projectors.

$$\Pi_{\mathsf{in}}^{\mathcal{Q},\mathcal{A}} := \sum_{\substack{(x,\pi),(\mathsf{vk},s,x^*,c^*,\sigma^*)\ \text{s.t.}\\ \mathsf{PV.Ver}[\mathsf{vk},s,(x^*,c^*,\sigma^*),\mathsf{in}](x,\pi)\neq\bot}} |(x,\pi),(\mathsf{vk},s,x^*,c^*,\sigma^*)\rangle\langle(x,\pi),(\mathsf{vk},s,x^*,c^*,\sigma^*)|$$

$$\Pi_{\mathsf{out}}^{\mathcal{A},\mathcal{V}} := \sum_{\substack{(\mathsf{vk},s,x^*,c^*,\sigma^*),\mu^*\ \text{s.t.}\\ V[\mathsf{out}]((x^*,c^*,\sigma^*),s,(\mu^*,\mathsf{vk}))=1}} |(\mathsf{vk},s,x^*,c^*,\sigma^*),\mu^*\rangle\langle(\mathsf{vk},s,x^*,c^*,\sigma^*),\mu^*|$$

Now, observe that

$$\Pr[\mathcal{H}'_{\iota-1}=1] = \mathop{\mathbb{E}}_{\mathbf{CK},\mathbf{dk},|\mathbf{ck}\rangle}\left[\left\|\Pi_{\mathsf{out}}^{\mathcal{A},\mathcal{V}}\mathsf{U}^{\mathbf{CK},\mathsf{PFC.DecZ}[\mathbf{dk}]}|\psi\rangle\right\|^2 : |\psi\rangle\leftarrow\mathsf{C}^{\mathbf{CK},\mathsf{PFC.DecZ}[\mathbf{dk}],\mathsf{PFC.DecX}[\mathbf{dk}]}(|\mathbf{ck}\rangle)\right],$$

and

$$\Pr[\mathcal{H}_\iota = 1] = \mathop{\mathbb{E}}_{\mathbf{CK},\mathbf{dk},|\mathbf{ck}\rangle}\left[\left\|\Pi_{\mathsf{out}}^{\mathcal{A},\mathcal{V}}\widetilde{\mathsf{U}}^{\mathbf{CK},\mathsf{PFC.DecZ}[\mathbf{dk}]}\,|\psi\rangle\right\|^2 : |\psi\rangle \leftarrow \mathsf{C}^{\mathbf{CK},\mathsf{PFC.DecZ}[\mathbf{dk}],\mathsf{PFC.DecX}[\mathbf{dk}]}(|\mathbf{ck}\rangle)\right].$$

Furthermore, for any state $|\psi\rangle$ output by C, we can write $|\psi\rangle \coloneqq |\psi_{\mathsf{in}}\rangle + |\psi_{\mathsf{in}}^{\perp}\rangle$, where $|\psi_{\mathsf{in}}\rangle \coloneqq \Pi_{\mathsf{in}}^{\mathcal{Q},\mathcal{A}}\,|\psi\rangle$. Notice that for any such $|\psi_{\mathsf{in}}^{\perp}\rangle$, it holds that $\mathsf{U}\,|\psi_{\mathsf{in}}^{\perp}\rangle = \widetilde{\mathsf{U}}\,|\psi_{\mathsf{in}}^{\perp}\rangle$, since query ι is answered with $\perp$ on both states and U and $\widetilde{\mathsf{U}}$ are otherwise identical. Thus, defining

$$\begin{aligned}\Pi_{\mathsf{out},\mathsf{U}} &\coloneqq \left(\mathsf{U}^{\mathbf{CK},\mathsf{PFC.DecZ}[\mathbf{dk}]}\right)^{\dagger}\Pi_{\mathsf{out}}\left(\mathsf{U}^{\mathbf{CK},\mathsf{PFC.DecZ}[\mathbf{dk}]}\right),\\ \Pi_{\mathsf{out},\widetilde{\mathsf{U}}} &\coloneqq \left(\widetilde{\mathsf{U}}^{\mathbf{CK},\mathsf{PFC.DecZ}[\mathbf{dk}]}\right)^{\dagger}\Pi_{\mathsf{out}}\left(\widetilde{\mathsf{U}}^{\mathbf{CK},\mathsf{PFC.DecZ}[\mathbf{dk}]}\right),\end{aligned}$$

we have that for any $|\psi\rangle \coloneqq |\psi_{\mathsf{in}}\rangle + |\psi_{\mathsf{in}}^{\perp}\rangle$,

$$\begin{aligned}&\left\|\Pi_{\mathsf{out},\mathsf{U}}(|\psi_{\mathsf{in}}\rangle + |\psi_{\mathsf{in}}^{\perp}\rangle)\right\|^2 - \left\|\Pi_{\mathsf{out},\widetilde{\mathsf{U}}}(|\psi_{\mathsf{in}}\rangle + |\psi_{\mathsf{in}}^{\perp}\rangle)\right\|^2\\ &= \langle\psi_{\mathsf{in}}|\,\Pi_{\mathsf{out},\mathsf{U}}\,|\psi_{\mathsf{in}}\rangle + \langle\psi_{\mathsf{in}}|\,\Pi_{\mathsf{out},\mathsf{U}}\,|\psi_{\mathsf{in}}^{\perp}\rangle + \langle\psi_{\mathsf{in}}^{\perp}|\,\Pi_{\mathsf{out},\mathsf{U}}\,|\psi_{\mathsf{in}}\rangle\\ &\quad - \langle\psi_{\mathsf{in}}|\,\Pi_{\mathsf{out},\widetilde{\mathsf{U}}}\,|\psi_{\mathsf{in}}\rangle - \langle\psi_{\mathsf{in}}|\,\Pi_{\mathsf{out},\widetilde{\mathsf{U}}}\,|\psi_{\mathsf{in}}^{\perp}\rangle - \langle\psi_{\mathsf{in}}^{\perp}|\,\Pi_{\mathsf{out},\widetilde{\mathsf{U}}}\,|\psi_{\mathsf{in}}\rangle\\ &\leq 3\left\|\Pi_{\mathsf{out},\mathsf{U}}\,|\psi_{\mathsf{in}}\rangle\right\| + 3\left\|\Pi_{\mathsf{out},\widetilde{\mathsf{U}}}\,|\psi_{\mathsf{in}}\rangle\right\|.\end{aligned}$$

So, we can bound $\Pr[\mathcal{H}'_{\iota-1} = 1] - \Pr[\mathcal{H}_\iota = 1]$ by

$$\mathop{\mathbb{E}}_{\mathbf{CK},\mathbf{dk},|\mathbf{ck}\rangle}\left[3\left\|\Pi_{\mathsf{out},\mathsf{U}}\,|\psi_{\mathsf{in}}\rangle\right\| + 3\left\|\Pi_{\mathsf{out},\widetilde{\mathsf{U}}}\,|\psi_{\mathsf{in}}\rangle\right\| : \begin{array}{r}|\psi\rangle \leftarrow \mathsf{C}^{\mathbf{CK},\mathsf{PFC.DecZ}[\mathbf{dk}],\mathsf{PFC.DecX}[\mathbf{dk}]}(|\mathbf{ck}\rangle)\\ |\psi\rangle \coloneqq |\psi_{\mathsf{in}}\rangle + |\psi_{\mathsf{in}}^{\perp}\rangle\end{array}\right],$$

and thus it suffices to show that

$$\mathop{\mathbb{E}}_{\mathbf{CK},\mathbf{dk},|\mathbf{ck}\rangle}\left[\left\|\Pi_{\mathsf{out}}\mathsf{U}^{\mathbf{CK},\mathsf{PFC.DecZ}[\mathbf{dk}]}\Pi_{\mathsf{in}}\,|\psi\rangle\right\|^2 : |\psi\rangle \leftarrow \mathsf{C}^{\mathbf{CK},\mathsf{PFC.DecZ}[\mathbf{dk}],\mathsf{PFC.DecX}[\mathbf{dk}]}(|\mathbf{ck}\rangle)\right] = \mathrm{negl}(\lambda),$$

and

$$\mathop{\mathbb{E}}_{\mathbf{CK},\mathbf{dk},|\mathbf{ck}\rangle}\left[\left\|\Pi_{\mathsf{out}}\widetilde{\mathsf{U}}^{\mathbf{CK},\mathsf{PFC.DecZ}[\mathbf{dk}]}\Pi_{\mathsf{in}}\,|\psi\rangle\right\|^2 : |\psi\rangle \leftarrow \mathsf{C}^{\mathbf{CK},\mathsf{PFC.DecZ}[\mathbf{dk}],\mathsf{PFC.DecX}[\mathbf{dk}]}(|\mathbf{ck}\rangle)\right] = \mathrm{negl}(\lambda).$$

The rest of this proof will be identical in either case, so we consider U. Towards proving this, we first recall that s is sampled uniformly at random at the very beginning of C, and the rest of C and U are classically controlled on s. So, let C_s be the same as C except that it is initialized with the string s. Then it suffices to show that for any fixed s,

$$\mathop{\mathbb{E}}_{\mathbf{CK},\mathbf{dk},|\mathbf{ck}\rangle}\left[\left\|\Pi_{\mathsf{out}}\mathsf{U}^{\mathbf{CK},\mathsf{PFC.DecZ}[\mathbf{dk}]}\Pi_{\mathsf{in}}\,|\psi\rangle\right\|^2 : |\psi\rangle \leftarrow \mathsf{C}_s^{\mathbf{CK},\mathsf{PFC.DecZ}[\mathbf{dk}],\mathsf{PFC.DecX}[\mathbf{dk}]}(|\mathbf{ck}\rangle)\right] = \mathrm{negl}(\lambda).$$

Now, we observe that the register $\mathcal{A}$ output by C contains a standard basis state holding $(\mathsf{vk}, s, (x^*, c^*, \sigma^*))$, where $c^* \coloneqq \{c^*_{i,j}\}_{i\in[r],j\in[\ell]}$. Define commitments $\mathbf{c} \coloneqq \{c^*_{i,j}\}_{(i,j)\in S}$ and write the output of C_s as $(|\psi\rangle, \mathbf{c})$ to make these commitments explicit. Then, define the following predicates, where f is the bijection from $[m] \to S$ defined earlier.

$\widetilde{D}_{\mathsf{in}}[\mathbf{dk}, \mathbf{c}]$:

- Take as input (b, π), where π is parsed as $(\cdot, \cdot, \{u_{i,j}, y_{i,j}, z_{i,j}\}_{i\in[r],j\in[\ell]})$.
- Output 1 if for some $w \in D_{\mathsf{in}}[P, b]$ and all $(i,j) \in S$, $w_{f^{-1}(i,j)} = \mathsf{PFC.DecZ}(\mathsf{dk}_{i,j}, c^*_{i,j}, u_{i,j})$.

$\widetilde{D}_{\mathsf{out}}[\mathbf{dk}, \mathbf{c}]$:

- Take as input (b, μ^*), where μ^* is parsed as $\{u_{i,j}, y_{i,j}, z_{i,j}\}_{i\in[r],j\in[\ell]}$.
- Output 1 if for some $w \in D_{\mathsf{out}}[P, b]$ and all $(i,j) \in S$, $w_{f^{-1}(i,j)} = \mathsf{PFC.DecZ}(\mathsf{dk}_{i,j}, c^*_{i,j}, u_{i,j})$.

Next, we define the following two projectors.

$$\Pi^{\mathcal{Q},\mathcal{A}}_{\mathbf{dk},\mathbf{c},\mathsf{in}} \coloneqq \sum_{\substack{(\cdot,\pi),(\cdot,\cdot,x^*,\cdot,\cdot) \text{ s.t.} \\ \widetilde{D}_{\mathsf{in}}[\mathbf{dk},\mathbf{c}](P(Q(x^*)),\pi)=1}} |(\cdot,\pi),(\cdot,\cdot,x^*,\cdot,\cdot)\rangle\langle(\cdot,\pi),(\cdot,\cdot,x^*,\cdot,\cdot)|$$

$$\Pi^{\mathcal{A},\mathcal{V}}_{\mathbf{dk},\mathbf{c},\mathsf{out}} \coloneqq \sum_{\substack{(\cdot,\cdot,x^*,\cdot,\cdot),\mu^* \text{ s.t.} \\ \widetilde{D}_{\mathsf{out}}[\mathbf{dk},\mathbf{c}](P(Q(x^*)),\mu^*)=1}} |(\cdot,\cdot,x^*,\cdot,\cdot),\mu^*\rangle\langle(\cdot,\cdot,x^*,\cdot,\cdot),\mu^*|$$

Note that $\Pi^{\mathcal{Q},\mathcal{A}}_{\mathsf{in}} \leq \Pi^{\mathcal{Q},\mathcal{A}}_{\mathbf{dk},\mathbf{c},\mathsf{in}}$ and $\Pi^{\mathcal{A},\mathcal{V}}_{\mathsf{out}} \leq \Pi^{\mathcal{A},\mathcal{V}}_{\mathbf{dk},\mathbf{c},\mathsf{out}}$, and thus it suffices to show that

$$\mathop{\mathbb{E}}_{\mathbf{CK},\mathbf{dk},|\mathbf{ck}\rangle}\left[\left\|\Pi^{\mathcal{A},\mathcal{V}}_{\mathbf{dk},\mathbf{c},\mathsf{out}}\mathsf{U}^{\mathbf{CK},\mathsf{PFC.DecZ}[\mathbf{dk}]}\Pi^{\mathcal{Q},\mathcal{A}}_{\mathbf{dk},\mathbf{c},\mathsf{in}}|\psi\rangle\right\|^2 : (|\psi\rangle,\mathbf{c}) \leftarrow \mathsf{C}_s^{\mathbf{CK},\mathsf{PFC.DecZ}[\mathbf{dk}],\mathsf{PFC.DecX}[\mathbf{dk}]}(|\mathbf{ck}\rangle)\right] = \mathrm{negl}(\lambda).$$

Finally, for each $b \in \{0,1\}$, we define

$$\Pi^{\mathcal{Q}}_{\mathbf{dk},\mathbf{c},\mathsf{in},b} \coloneqq \sum_{(\cdot,\pi):\widetilde{D}_{\mathsf{in}}[\mathbf{dk},\mathbf{c}](b,\pi)=1} |(\cdot,\pi)\rangle\langle(\cdot,\pi)|\,,\ \ \Pi^{\mathcal{V}}_{\mathbf{dk},\mathbf{c},\mathsf{out},b} \coloneqq \sum_{\rho^*:\widetilde{D}_{\mathsf{out}}[\mathbf{dk},\mathbf{c}](b,\mu^*)=1} |\mu^*\rangle\langle\mu^*|\,.$$

In fact, these projectors now only operate on the sub-registers of $\mathcal{Q}$ and $\mathcal{V}$ that hold the strings

$$\{u_{i,j}\}_{(i,j)\in S} = \{u_{f(\tau)}\}_{\tau\in[m]}.$$

Naming these sub-registers $\mathcal{Q}' = (\mathcal{Q}_1, \dots, \mathcal{Q}_m)$ and $\mathcal{V}' = (\mathcal{V}_1, \dots, \mathcal{V}_m)$, we can write

$$\Pi^{\mathcal{Q}'}_{\mathbf{dk},\mathbf{c},\mathsf{in},b} \coloneqq \sum_{w\in D_{\mathsf{in}}[P,b]}\left(\bigotimes_{\tau\in[m]}\Pi^{\mathcal{Q}_\tau}_{\mathsf{dk}_\tau,c^*_\tau,w_\tau}\right),\ \ \Pi^{\mathcal{V}'}_{\mathbf{dk},\mathbf{c},\mathsf{out},b} \coloneqq \sum_{w\in D_{\mathsf{out}}[P,b]}\left(\bigotimes_{\tau\in[m]}\Pi^{\mathcal{V}_\tau}_{\mathsf{dk}_\tau,c^*_\tau,w_\tau}\right),$$

where

$$\Pi_{\mathsf{dk}_\tau, c^*_\tau, w_\tau} := \sum_{u : \mathsf{PFC.DecZ}(\mathsf{dk}_\tau, c^*_\tau, u) = w_\tau} |u\rangle \langle u| .$$

Now, to complete the proof, we note that

$$\underset{\mathbf{CK}, \mathbf{dk}, |\mathbf{ck}\rangle}{\mathbb{E}} \left[\left\| \Pi_{\mathbf{dk}, \mathbf{c}, \mathsf{out}} \mathsf{U}^{\mathbf{CK}, \mathsf{PFC.DecZ}[\mathbf{dk}]} \Pi_{\mathbf{dk}, \mathbf{c}, \mathsf{in}} |\psi\rangle \right\|^2 : (|\psi\rangle, \mathbf{c}) \leftarrow \mathsf{C}_s^{\mathbf{CK}, \mathsf{PFC.DecZ}[\mathbf{dk}], \mathsf{PFC.DecX}[\mathbf{dk}]}(|\mathbf{ck}\rangle) \right]$$
$$\leq \underset{\mathbf{CK}, \mathbf{dk}, |\mathbf{ck}\rangle}{\mathbb{E}} \left[\left\| \Pi_{\mathbf{dk}, \mathbf{c}, \mathsf{out}, 0} \mathsf{U}^{\mathbf{CK}, \mathsf{PFC.DecZ}[\mathbf{dk}]} \Pi_{\mathbf{dk}, \mathbf{c}, \mathsf{in}, 0} |\psi\rangle \right\|^2 : (|\psi\rangle, \mathbf{c}) \leftarrow \mathsf{C}_s^{\mathbf{CK}, \mathsf{PFC.DecZ}[\mathbf{dk}], \mathsf{PFC.DecX}[\mathbf{dk}]}(|\mathbf{ck}\rangle) \right]$$
$$+ \underset{\mathbf{CK}, \mathbf{dk}, |\mathbf{ck}\rangle}{\mathbb{E}} \left[\left\| \Pi_{\mathbf{dk}, \mathbf{c}, \mathsf{out}, 1} \mathsf{U}^{\mathbf{CK}, \mathsf{PFC.DecZ}[\mathbf{dk}]} \Pi_{\mathbf{dk}, \mathbf{c}, \mathsf{in}, 1} |\psi\rangle \right\|^2 : (|\psi\rangle, \mathbf{c}) \leftarrow \mathsf{C}_s^{\mathbf{CK}, \mathsf{PFC.DecZ}[\mathbf{dk}], \mathsf{PFC.DecX}[\mathbf{dk}]}(|\mathbf{ck}\rangle) \right],$$

and by the string binding with public decodability of PFC (Definition 4.4), and the fact that $D_{\mathsf{in}}[P, b]$ and $D_{\mathsf{out}}[P, b]$ are disjoint sets of strings, we have that for any $b \in \{0, 1\}$,

$$\underset{\mathbf{CK}, \mathbf{dk}, |\mathbf{ck}\rangle}{\mathbb{E}} \left[\left\| \Pi_{\mathbf{dk}, \mathbf{c}, \mathsf{out}, b} \mathsf{U}^{\mathbf{CK}, \mathsf{PFC.DecZ}[\mathbf{dk}]} \Pi_{\mathbf{dk}, \mathbf{c}, \mathsf{in}, b} |\psi\rangle \right\| : (|\psi\rangle, \mathbf{c}) \leftarrow \mathsf{C}_s^{\mathbf{CK}, \mathsf{PFC.DecZ}[\mathbf{dk}], \mathsf{PFC.DecX}[\mathbf{dk}]}(|\mathbf{ck}\rangle) \right] = \operatorname{negl}(\lambda).$$

□

5.5 Application: Publicly-Verifiable QFHE

Now, we apply our general framework for verification of quantum partitioning circuits to the specific case of quantum fully-homomorphic encryption (QFHE). First, we define the notion of publicly-verifiable QFHE for pseudo-deterministic circuits. We write the syntax in the *oracle model*, where the parameters used for proving and verifying include an efficient classical oracle PP. Such a scheme can be heuristically instantiated in the plain model by using post-quantum indistinguishability obfuscation to obfuscate this oracle.

Definition 5.19 (Publicly-verifiable QFHE for pseudo-deterministic circuits)**.** *A publicly-verifiable quantum fully-homomorphic encryption scheme for pseudo-deterministic circuits consists of the following algorithms* (Gen, Enc, VerGen, Eval, Ver, Dec).

- $\mathsf{Gen}(1^\lambda, D) \to (\mathsf{pk}, \mathsf{sk})$*: On input the security parameter 1^λ and a circuit depth D, the key generation algorithm returns a public key* pk *and a secret key* sk.
- $\mathsf{Enc}(\mathsf{pk}, x) \to \mathsf{ct}$*: On input the public key* pk *and a classical plaintext x, the encryption algorithm outputs a ciphertext* ct.
- $\mathsf{VerGen}(\mathsf{ct}, Q) \to (|\mathsf{pp}\rangle, \mathsf{PP})$*: On input a ciphertext* ct *and the description of a quantum circuit Q, the verification parameter generation algorithm returns public parameters* $(|\mathsf{pp}\rangle, \mathsf{PP})$*, where* PP *is the description of a classical deterministic polynomial-time functionality.*
- $\mathsf{Eval}^{\mathsf{PP}}(\mathsf{ct}, |\mathsf{pp}\rangle, y) \to (\widetilde{\mathsf{ct}}, \pi)$*: The evaluation algorithm has oracle access to* PP*, takes as input a ciphertext* ct*, a quantum state* $|\mathsf{pp}\rangle$*, and a classical string y, and outputs a ciphertext* $\widetilde{\mathsf{ct}}$ *and proof π.*

- $\mathsf{Ver}^{\mathsf{PP}}(y, \widetilde{\mathsf{ct}}, \pi) \to \{\top, \bot\}$*: The classical verification algorithm has oracle access to* PP*, takes as input a string* y*, a ciphertext* $\widetilde{\mathsf{ct}}$*, and a proof* π*, and outputs either* $\top$ *or* $\bot$*.*
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \to x$*: On input the secret key* sk *and a classical ciphertext* ct*, the decryption algorithm returns a message* x*.*

These algorithms should satisfy the following properties.

- ***Correctness.*** *For any family* $\{Q_\lambda, x_\lambda, y_\lambda\}_{\lambda\in\mathbb{N}}$ *where* Q_λ *takes two inputs,* $\{Q_\lambda(x_\lambda, \cdot)\}_{\lambda\in\mathbb{N}}$ *is pseudo-deterministic, and* Q_λ *has depth* $D = D(\lambda)$*, it holds that*

$$\Pr\left[\begin{array}{l}\mathsf{Ver}^{\mathsf{PP}}(y, \widetilde{\mathsf{ct}}, \pi) = \top \ \wedge \\ \mathsf{Dec}(\mathsf{sk}, \widetilde{\mathsf{ct}}) = Q(x, y)\end{array} : \begin{array}{r}(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda, D) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, x) \\ (|\mathsf{pp}\rangle, \mathsf{PP}) \leftarrow \mathsf{VerGen}(\mathsf{ct}, Q) \\ (\widetilde{\mathsf{ct}}, \pi) \leftarrow \mathsf{Eval}^{\mathsf{PP}}(\mathsf{ct}, |\mathsf{pp}\rangle, y)\end{array}\right] = 1 - \mathrm{negl}(\lambda).$$

- ***Security.*** *For any QPT adversary* $\{\mathsf{A}_\lambda\}_{\lambda\in\mathbb{N}}$*, depth* $D = D(\lambda)$*, and messages* $\{x_{\lambda,0}, x_{\lambda,1}\}_{\lambda\in\mathbb{N}}$*,*

$$\begin{aligned}\Bigg| &\Pr\left[\mathsf{A}(\mathsf{pk}, \mathsf{ct}) = 1 : \begin{array}{r}(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda, D) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, x_0)\end{array}\right] \\ &- \Pr\left[\mathsf{A}(\mathsf{pk}, \mathsf{ct}) = 1 : \begin{array}{r}(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda, D) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, x_1)\end{array}\right]\Bigg| = \mathrm{negl}(\lambda)\end{aligned}$$

- ***Soundness.*** *For any QPT adversary* $\{\mathsf{A}_\lambda\}_{\lambda\in\mathbb{N}}$*, depth* $D = D(\lambda)$*, and family* $\{Q_\lambda, x_\lambda\}_{\lambda\in\mathbb{N}}$*, where* Q_λ *takes two inputs and* $\{Q_\lambda(x_\lambda, \cdot)\}_{\lambda\in\mathbb{N}}$ *is pseudo-deterministic,*

$$\Pr\left[\begin{array}{l}\mathsf{Ver}^{\mathsf{PP}}(y, \widetilde{\mathsf{ct}}, \pi) = \top \ \wedge \\ \mathsf{Dec}(\mathsf{sk}, \widetilde{\mathsf{ct}}) \neq Q(x, y)\end{array} : \begin{array}{r}(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda, D) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, x) \\ (|\mathsf{pp}\rangle, \mathsf{PP}) \leftarrow \mathsf{VerGen}(\mathsf{ct}, Q) \\ (y, \widetilde{\mathsf{ct}}, \pi) \leftarrow \mathsf{A}^{\mathsf{PP}}(\mathsf{ct}, |\mathsf{pp}\rangle)\end{array}\right] = \mathrm{negl}(\lambda).$$

We will now construct publicly-verifiable QFHE for pseudo-deterministic circuits from the following ingredients.

- A quantum fully-homomorphic encryption scheme ($\mathsf{QFHE.Gen}$, $\mathsf{QFHE.Enc}$, $\mathsf{QFHE.Eval}$, $\mathsf{QFHE.Dec}$) (Section 3.4).
- A protocol for publicly-verifiable non-interactive classical verification of quantum partitioning circuits in the oracle model ($\mathsf{PV.Gen}$, $\mathsf{PV.Prove}$, $\mathsf{PV.Verify}$, $\mathsf{PV.Combine}$) (Section 5.4).

Our construction goes as follows.

- $\mathsf{PVQFHE.Gen}(1^\lambda, D)$: Same as $\mathsf{QFHE.Gen}(1^\lambda, D)$.
- $\mathsf{PVQFHE.Enc}(\mathsf{pk}, x)$: Same as $\mathsf{QFHE.Enc}(\mathsf{pk}, x)$.
- $\mathsf{PVQFHE.VerGen}(\mathsf{ct}, Q)$:
  - Define the quantum circuit $E[\mathsf{ct}] : y \to \mathsf{QFHE.Eval}(Q(\cdot, y), \mathsf{ct})$.

- Sample $(\mathsf{PV.vk}, |\mathsf{PV.pk}\rangle, \mathsf{PV.PK}) \leftarrow \mathsf{PV.Gen}(1^\lambda, E[\mathsf{ct}])$.
- Let $\mathsf{VK}(y, \pi)$ be the following classical functionality. First, run $\mathsf{PV.Ver}(\mathsf{PV.vk}, y, \pi)$. Output $\bot$ if the output was $\bot$. Otherwise, parse the output as $(\mathsf{ct}_1, \ldots, \mathsf{ct}_m)$, compute $\widetilde{\mathsf{ct}} := \mathsf{QFHE.Eval}(\mathsf{PV.Combine}, (\mathsf{ct}_1, \ldots, \mathsf{ct}_m))$, and output $\widetilde{\mathsf{ct}}$.[20]
- Output $|\mathsf{pp}\rangle := |\mathsf{PV.pk}\rangle$, $\mathsf{PP} := (\mathsf{PV.PK}, \mathsf{VK})$.

- $\mathsf{PVQFHE.Eval}^{\mathsf{PP}}(\mathsf{ct}, |\mathsf{pp}\rangle, y)$:
  - Run $\pi \leftarrow \mathsf{PV.Prove}^{\mathsf{PV.PK}}(|\mathsf{pp}\rangle, E[\mathsf{ct}], y)$.
  - Compute $\widetilde{\mathsf{ct}} = \mathsf{VK}(y, \pi)$, and output $(\widetilde{\mathsf{ct}}, \pi)$.

- $\mathsf{PVQFHE.Ver}^{\mathsf{PP}}(y, \widetilde{\mathsf{ct}}, \pi)$: Output $\top$ iff $\mathsf{VK}(y, \pi) = \widetilde{\mathsf{ct}}$.

- $\mathsf{PVQFHE.Dec}(\mathsf{sk}, \mathsf{ct})$: Same as $\mathsf{QFHE.Dec}(\mathsf{sk}, \mathsf{ct})$.

Theorem 5.20. *The scheme described above satisfies Definition 5.19.*

Proof. Correctness follows immediately from the evaluation correctness of QFHE (Definition 3.9) and the completeness of PV (Definition 5.1). Security follows immediately from the semantic security of QFHE (Definition 3.8). Soundness follows immediately from the correctness of QFHE (Definition 3.9) and soundness of PV (Definition 5.2), since $\mathsf{QFHE.Dec}(\mathsf{sk}, \cdot) \circ E[\mathsf{ct}]$ is pseudo-deterministic and the VK oracle is nothing but the $\mathsf{PV.Ver}[\mathsf{vk}]$ oracle plus post-processing. □

6 Quantum Obfuscation

6.1 Construction

In this section, we construct virtual black-box (VBB) obfuscation for pseudo-deterministic quantum circuits from the following ingredients.

- A VBB obfuscator $(\mathsf{CObf}, \mathsf{CEval})$ for classical circuits (Definition 3.4).

- A publicly-verifiable QFHE for pseudo-deterministic circuits in the oracle model $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{VerGen}, \mathsf{Eval}, \mathsf{Ver}, \mathsf{Dec})$ (Definition 5.19).

The construction is given in Fig. 8.

Theorem 6.1. $(\mathsf{QObf}, \mathsf{QEval})$ *described in Fig. 8 is a virtual black-box obfuscator for pseudo-deterministic quantum circuits, satisfying Definition 3.4.*

Proof. First, correctness follows immediately from the correctness of the VBB obfuscator (Definition 3.4) and the correctness of the publicly-verifiable QFHE scheme (Definition 5.19). Note that even though the evaluation procedure may include measurements, an evaluator could run coherently, measure just the output bit b, and reverse. By Gentle Measurement (Lemma 3.1), this implies the ability to run the obfuscated program on any $\mathrm{poly}(\lambda)$ number of inputs.

Next, we show security. For any QPT adversary $\{\mathsf{A}_\lambda\}_{\lambda\in\mathbb{N}}$, we define a simulator $\{\mathsf{S}_\lambda\}_{\lambda\in\mathbb{N}}$ as follows, where $\{\widetilde{\mathsf{A}}_\lambda\}_{\lambda\in\mathbb{N}}$ is the simulator for the classical obfuscation scheme $(\mathsf{CObf}, \mathsf{CEval})$, defined based on $\{\mathsf{A}_\lambda\}_{\lambda\in\mathbb{N}}$.

[20]Here, we are using the fact that QFHE.Eval is a deterministic classical functionality when evaluating a deterministic classical functionality.

Obfuscation scheme $(\mathsf{QObf}, \mathsf{QEval})$ **for pseudo-deterministic quantum circuits**

- $\mathsf{QObf}(1^\lambda, Q)$:
  - Let U be the universal quantum circuit that takes as input the description of a circuit of size $|Q|$ and an input of size n, where n is the length of an input to Q. Let D be the depth of U.
  - Sample $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda, D)$, $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, Q)$, and $(|\mathsf{pp}\rangle, \mathsf{PP}) \leftarrow \mathsf{VerGen}(\mathsf{ct}, U)$.
  - Let $\mathsf{DK}(x, \widetilde{\mathsf{ct}}, \pi)$ be the following functionality. First, run $\mathsf{Ver}^{\mathsf{PP}}(x, \widetilde{\mathsf{ct}}, \pi)$. If the output was $\bot$, then output $\bot$, and otherwise output $\mathsf{Dec}(\mathsf{sk}, \widetilde{\mathsf{ct}})$.
  - Sample $\widetilde{\mathsf{PP}} \leftarrow \mathsf{CObf}(1^\lambda, \mathsf{PP})$ and $\widetilde{\mathsf{DK}} \leftarrow \mathsf{CObf}(1^\lambda, \mathsf{DK})$.
  - Output $\widetilde{Q} := \left(\mathsf{ct}, |\mathsf{pp}\rangle, \widetilde{\mathsf{PP}}, \widetilde{\mathsf{DK}}\right)$.
- $\mathsf{QEval}(\widetilde{Q}, x)$:
  - Parse $\widetilde{Q}$ as $\left(\mathsf{ct}, |\mathsf{pp}\rangle, \widetilde{\mathsf{PP}}, \widetilde{\mathsf{DK}}\right)$.
  - Compute $(\widetilde{\mathsf{ct}}, \pi) \leftarrow \mathsf{Eval}^{\widetilde{\mathsf{PP}}}(\mathsf{ct}, |\mathsf{pp}\rangle, x)$.
  - Output $b := \widetilde{\mathsf{DK}}(x, \widetilde{\mathsf{ct}}, \pi)$.

Figure 8: Obfuscation for pseudo-deterministic quantum circuits.

- Sample $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda, D)$, $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, 0^{|Q|})$, and $(|\mathsf{pp}\rangle, \mathsf{PP}) \leftarrow \mathsf{VerGen}(\mathsf{ct}, U)$.
- Run $\widetilde{\mathsf{A}}_\lambda^{\mathsf{PP},\mathsf{DK}}(\mathsf{ct}, |\mathsf{pp}\rangle)$, answering PP calls honestly, and DK calls as follows.
  - Take $(x, \widetilde{\mathsf{ct}}, \pi)$ as input.
  - Run $\mathsf{Ver}^{\mathsf{PP}}(x, \widetilde{\mathsf{ct}}, \pi)$. If the output was $\bot$ then output $\bot$.
  - Otherwise, forward x to the external oracle $O[Q]$, and return the result $b = O[Q](x)$.
- Output $\widetilde{\mathsf{A}}_\lambda$'s output.

Now, for any circuit Q, we define a sequence of hybrids.

- $\mathcal{H}_0$: Sample $\widetilde{Q} \leftarrow \mathsf{QObf}(1^\lambda, Q)$ and run $\mathsf{A}_\lambda(1^\lambda, \widetilde{Q})$.
- $\mathcal{H}_1$: Sample $(\mathsf{ct}, |\mathsf{pp}\rangle, \mathsf{PP}, \mathsf{DK})$ as in $\mathsf{QObf}(1^\lambda, Q)$, and run $\widetilde{\mathsf{A}}_\lambda^{\mathsf{PP},\mathsf{DK}}(\mathsf{ct}, |\mathsf{pp}\rangle)$.
- $\mathcal{H}_2$: Same as $\mathcal{H}_1$, except that calls to DK are answered as in the description of S_λ.
- $\mathcal{H}_3$: Same as $\mathcal{H}_2$, except that we sample $(\mathsf{ct}, |\mathsf{pp}\rangle, \mathsf{PP}) \leftarrow \mathsf{Enc}(\mathsf{pk}, 0^{|Q|}, U)$. This is S_λ.

We complete the proof by showing the following.

- $|\Pr[\mathcal{H}_0 = 1] - \Pr[\mathcal{H}_1 = 1]| = \mathrm{negl}(\lambda)$. This follows from the security of the classical obfuscation scheme $(\mathsf{CObf}, \mathsf{CEval})$.

- $|\Pr[\mathcal{H}_1 = 1] - \Pr[\mathcal{H}_2 = 1]| = \text{negl}(\lambda)$. Suppose otherwise. Then there must exist some query made by $\widetilde{\mathsf{A}}_\lambda$ to DK with noticeable amplitude on $(x, \widetilde{\mathsf{ct}}, \pi)$ such that DK does not return $\bot$ but $\mathsf{Dec}(\mathsf{sk}, \widetilde{\mathsf{ct}}) \neq Q(x)$. Thus, we can measure a random one of the $\text{poly}(\lambda)$ many queries made by $\widetilde{\mathsf{A}}_\lambda$ to obtain such an $(x, \widetilde{\mathsf{ct}}, \pi)$, which violates the soundness of the publicly-verifiable QFHE scheme (Definition 5.19).
- $|\Pr[\mathcal{H}_2 = 1] - \Pr[\mathcal{H}_3 = 1]| = \text{negl}(\lambda)$. Since sk is no longer used in $\mathcal{H}_2$ to respond to DK queries, this follows directly from the security of the publicly-verifiable QFHE scheme (Definition 5.19).

□

6.2 Application: Copy-protection

We sketch an application of our obfuscation scheme to copy-protection of *quantum* programs. Let $(\mathsf{QObf}, \mathsf{QEval})$ be a VBB obfuscation scheme for pseudo-deterministic quantum circuits, and let F_k be a pseudo-random function secure against superposition-query attacks. In Fig. 9, we describe [ALL+21]'s construction of a software copy-protection scheme, generalized to copy-protect pseudo-deterministic quantum circuits.

We refer the reader to [ALL+21] for definitions of (generalized) quantum unlearnable function families and anti-piracy of quantum copy-protection schemes. Here, we observe that if Q is a pseudo-deterministic circuit, then both O_1 and O_2 are as well, and thus they can be obfuscated by our scheme. Finally, it is straightforward to see that any classical functionality f sampled from a distribution $\mathcal{F}$ can be replaced with a pseudo-deterministic quantum functionality Q sampled from a distribution $\mathcal{Q}$ in the definitions and proofs from [ALL+21]. Thus, we can generalize their main theorem as follows.

Theorem 6.2. *(Corollary of [ALL+21, Theorem 4] and Theorem 6.1) Let $\mathcal{Q}$ be a family of pseudo-deterministic quantum ciruits that is γ-quantum-unlearnable with respect to distribution $\mathcal{D}$ (where γ is a non-negligible function of λ). Then Protocol 9 is a copy protection scheme for $\mathcal{Q}, \mathcal{D}$ that has $(\gamma(\lambda) - 1/\text{poly}(\lambda))$-anti-piracy security, for any polynomial $\text{poly}(\lambda)$.*

6.3 Application: Functional encryption

We sketch an application of our obfuscation scheme to functional encryption for pseudo-deterministic quantum functionalities. Let $(\mathsf{QObf}, \mathsf{QEval})$ be a VBB obfuscation scheme for pseudo-deterministic quantum circuits,[21] let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a (post-quantum) public-key encryption scheme, and let $(\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$ be a (post-quantum) statistically simulation sound non-interactive zero-knowledge proof system (SSS-NIZK). We refer the reader to [GGH+16] for preliminaries on SSS-NIZK, and for definitions of functional encryption.

Consider the following construction of functional encryption for pseudo-deterministic quantum functionalities.

- $\mathsf{FE.Setup}(1^\lambda)$: Sample $(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{Gen}(1^\lambda)$, $(\mathsf{pk}_2, \mathsf{sk}_2) \leftarrow \mathsf{Gen}(1^\lambda)$, $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$, and output $\mathsf{pp} := (\mathsf{pk}_1, \mathsf{pk}_2, \mathsf{crs})$ and $\mathsf{msk} := \mathsf{sk}_1$.

[21]For this application, we technically only require the weaker notion of indistinguishability obfuscation (Definition 3.5).

Quantum copy-protection scheme [ALL+21]

- $\mathsf{Setup}(1^\lambda) \to \mathsf{sk}$:
  - Take as input the security parameter 1^λ.
  - Sample a uniformly random subspace $S < \mathbb{F}_2^\lambda$ of dimension $\lambda/2$.
  - Sample a PRF key $k \leftarrow \{0,1\}^\lambda$.
  - Out $\mathsf{sk} := (S, k)$.
- $\mathsf{Generate}(\mathsf{sk}, Q) \to \widehat{Q}$:
  - Take as input $\mathsf{sk} = (S, k)$ and the description of a pseudo-deterministic quantum circuit Q.
  - Let O_1 be the functionality that takes (x, v) as input and outputs $Q(x) \oplus F_k(x)$ if $v \in S \setminus \{0\}$, and $\perp$ otherwise.
  - Let O_2 be the functionality that takes (x, v) as input and outputs $F_k(x)$ if $v \in S^\perp \setminus \{0\}$, and $\perp$ otherwise.
  - Sample $\widetilde{O}_1 \leftarrow \mathsf{QObf}(1^\lambda, O_1)$ and $\widetilde{O}_2 \leftarrow \mathsf{QObf}(1^\lambda, O_2)$
  - Output $\widehat{Q} := \left(|S\rangle, \widetilde{O}_1, \widetilde{O}_2\right)$.
- $\mathsf{Compute}(\widehat{Q}, x) \to y$:
  - Parse $\widehat{Q}$ as $|S\rangle, \widetilde{O}_1, \widetilde{O}_2$, where $|S\rangle$ is on register $\mathcal{S}$.
  - Apply $\mathsf{QEval}(\widetilde{O}_1, \cdot)$ coherently to register $\mathcal{S}$, measure the output to obtain y_1, and reverse the computation of $\mathsf{QEval}(\widetilde{O}_1, \cdot)$.
  - Apply $H^{\otimes\lambda}$ to register $\mathcal{S}$, apply $\mathsf{QEval}(\widetilde{O}_2, \cdot)$ coherently to register $\mathcal{S}$, measure the output to obtain y_2, reverse the computation of $\mathsf{QEval}(\widetilde{O}_2, \cdot)$, and finally apply $H^{\otimes\lambda}$ to register $\mathcal{S}$.
  - Output $y := y_1 \oplus y_2$.

Figure 9: A description of the quantum copy protection scheme from [ALL+21], where the Generate algorithm may now take as input the description of a pseudo-deterministic quantum functionality.

- $\mathsf{FE.KeyGen}(\mathsf{msk}, Q)$: On input the master secret key msk and the description of a pseudo-deterministic quantum circuit Q, define the following pseudo-deterministic quantum circuit $C[Q, \mathsf{crs}, \mathsf{sk}_1]$.
  - Take $(\mathsf{ct}_1, \mathsf{ct}_2, \pi)$ as input.
  - Check that π is a valid SSS-NIZK proof under crs that there exists (m, r_1, r_2) such that $\mathsf{ct}_1 = \mathsf{Enc}(\mathsf{pk}_1, m; r_1)$ and $\mathsf{ct}_2 = \mathsf{Enc}(\mathsf{pk}_2, m; r_2)$.
  - If so, output $Q(\mathsf{Dec}(\mathsf{sk}_1, \mathsf{ct}_1))$, and otherwise output $\perp$.

  Finally, sample and output $\mathsf{sk}_Q \leftarrow \mathsf{QObf}(1^\lambda, C[Q, \mathsf{crs}, \mathsf{sk}_1])$.
- $\mathsf{FE.Enc}(\mathsf{pp}, m)$: Sample $r_1, r_2 \leftarrow \{0,1\}^\lambda$, compute $\mathsf{ct}_1 := \mathsf{Enc}(\mathsf{pk}_1, m; r_1)$, $\mathsf{ct}_2 := \mathsf{Enc}(\mathsf{pk}_2, m; r_2)$, compute a SSS-NIZK proof π that there exists (m, r_1, r_2) such that $\mathsf{ct}_1 = \mathsf{Enc}(\mathsf{pk}_1, m; r_1)$ and $\mathsf{ct}_2 = \mathsf{Enc}(\mathsf{pk}_2, m; r_2)$, and output $\mathsf{ct} := (\mathsf{ct}_1, \mathsf{ct}_2, \pi)$.

- $\mathsf{FE.Dec}(\mathsf{sk}_Q, \mathsf{ct})$: Run the obfuscated program sk_Q on input ct to obtain the output.

It is straightforward to extend the definitions and proofs in Section 6 of [GGH$^+$16] to consider functional encryption and obfuscation of pseudo-deterministic quantum circuits. As a result, we obtain the following theorem.

Theorem 6.3 (Corollary of [GGH$^+$16] Section 6 and Theorem 6.1)**.** *The above construction is a functional encryption scheme satisfying indistinguishability security for the class of polynomial-size pseudo-deterministic quantum functionalities.*

References

[Aar09] Scott Aaronson. Quantum copy-protection and quantum money. In *2009 24th Annual IEEE Conference on Computational Complexity*, pages 229–242, 2009.

[ABDS21] Gorjan Alagic, Zvika Brakerski, Yfke Dulek, and Christian Schaffner. Impossibility of quantum virtual black-box obfuscation of classical circuits. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 497–525, Virtual Event, August 2021. Springer, Heidelberg.

[ABOEM18] Dorit Aharonov, Michael Ben-Or, Elad Eban, and Urmila Mahadev. Interactive proofs for quantum computations. *arXiv (CoRR)*, abs/1804.00640, 2018.

[AC12] Scott Aaronson and Paul Christiano. Quantum money from hidden subspaces. In *Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing*, STOC '12, page 41–60, New York, NY, USA, 2012. Association for Computing Machinery.

[ACGH20] Gorjan Alagic, Andrew M. Childs, Alex B. Grilo, and Shih-Han Hung. Non-interactive classical verification of quantum computation. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part III*, volume 12552 of *LNCS*, pages 153–180. Springer, Heidelberg, November 2020.

[ADSS17] Gorjan Alagic, Yfke Dulek, Christian Schaffner, and Florian Speelman. Quantum fully homomorphic encryption with verification. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part I*, volume 10624 of *LNCS*, pages 438–467. Springer, Heidelberg, December 2017.

[AF16] Gorjan Alagic and Bill Fefferman. On quantum obfuscation. *arXiv (CoRR)*, abs/1602.01771, 2016.

[AGKZ20] Ryan Amos, Marios Georgiou, Aggelos Kiayias, and Mark Zhandry. One-shot signatures and applications to hybrid quantum/classical authentication. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *52nd ACM STOC*, pages 255–268. ACM Press, June 2020.

[AJJ14] Gorjan Alagic, Stacey Jeffery, and Stephen Jordan. Circuit Obfuscation Using Braids. In Steven T. Flammia and Aram W. Harrow, editors, *9th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2014)*, volume 27 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 141–160, Dagstuhl, Germany, 2014. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[AK21] Prabhanjan Ananth and Fatih Kaleoglu. Unclonable encryption, revisited. In Kobbi Nissim and Brent Waters, editors, *TCC 2021, Part I*, volume 13042 of *LNCS*, pages 299–329. Springer, Heidelberg, November 2021.

[AKL$^+$22] Prabhanjan Ananth, Fatih Kaleoglu, Xingjian Li, Qipeng Liu, and Mark Zhandry. On the feasibility of unclonable encryption, and more. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022*, Lecture Notes in Computer Science. Springer, 2022.

[AL21] Prabhanjan Ananth and Rolando L. La Placa. Secure software leasing. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part II*, volume 12697 of *LNCS*, pages 501–530. Springer, Heidelberg, October 2021.

[ALL$^+$21] Scott Aaronson, Jiahui Liu, Qipeng Liu, Mark Zhandry, and Ruizhe Zhang. New approaches for quantum copy-protection. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 526–555, Virtual Event, August 2021. Springer, Heidelberg.

[AMTDW00] A. Ambainis, M. Mosca, A. Tapp, and R. De Wolf. Private quantum channels. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 547–553, 2000.

[ARU14] Andris Ambainis, Ansis Rosmanis, and Dominique Unruh. Quantum attacks on classical proof systems: The hardness of quantum rewinding. In *55th FOCS*, pages 474–483. IEEE Computer Society Press, October 2014.

[Bar21] James Bartusek. Secure quantum computation with classical communication. In Kobbi Nissim and Brent Waters, editors, *TCC 2021, Part I*, volume 13042 of *LNCS*, pages 1–30. Springer, Heidelberg, November 2021.

[BB84] Charles H Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of the IEEE International Conference on Computers, Systems, and Signal Processing*, pages 175–179, 1984.

[BCG$^+$02] H. Barnum, C. Crepeau, D. Gottesman, A. Smith, and A. Tapp. Authentication of quantum messages. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pages 449–458, 2002.

[BCM$^+$21] Zvika Brakerski, Paul F. Christiano, Urmila Mahadev, Umesh V. Vazirani, and Thomas Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. *J. ACM*, 68(5):31:1–31:47, 2021.

[BDGM22] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Factoring and Pairings Are Not Necessary for IO: Circular-Secure LWE Suffices. In Mikołaj Bojańczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*, volume 229 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 28:1–28:20, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[BFK09] Anne Broadbent, Joseph Fitzsimons, and Elham Kashefi. Universal blind quantum computation. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*. IEEE, oct 2009.

[BGI$^+$12] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. *J. ACM*, 59(2):6:1–6:48, 2012.

[BGL$^+$15] Nir Bitansky, Sanjam Garg, Huijia Lin, Rafael Pass, and Sidharth Telang. Succinct randomized encodings and their applications. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 439–448. ACM, 2015.

[BGMZ18] James Bartusek, Jiaxin Guan, Fermi Ma, and Mark Zhandry. Return of GGH15: Provable security against zeroizing attacks. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part II*, volume 11240 of *LNCS*, pages 544–574. Springer, Heidelberg, November 2018.

[BJ15] Anne Broadbent and Stacey Jeffery. Quantum homomorphic encryption for circuits of low T-gate complexity. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 609–629. Springer, Heidelberg, August 2015.

[BJSW20] Anne Broadbent, Zhengfeng Ji, Fang Song, and John Watrous. Zero-knowledge proof systems for QMA. *SIAM Journal on Computing*, 49(2):245–283, 2020.

[BK21] Anne Broadbent and Raza Ali Kazmi. Constructions for quantum indistinguishability obfuscation. In Patrick Longa and Carla Ràfols, editors, *Progress in Cryptology – LATINCRYPT 2021*, pages 24–43, Cham, 2021. Springer International Publishing.

[BKL$^+$22] James Bartusek, Yael Tauman Kalai, Alex Lombardi, Fermi Ma, Giulio Malavolta, Vinod Vaikuntanathan, Thomas Vidick, and Lisa Yang. Succinct classical verification of quantum computation. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part II*, volume 13508 of *Lecture Notes in Computer Science*, pages 195–211. Springer, 2022.

[BM22] James Bartusek and Giulio Malavolta. Indistinguishability obfuscation of null quantum circuits and applications. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 - February 3, 2022, Berkeley, CA, USA*, volume 215 of *LIPIcs*, pages 15:1–15:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[BR95] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. pages 62–73. ACM Press, 1995.

[Bra18] Zvika Brakerski. Quantum FHE (almost) as secure as classical. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 67–95. Springer, Heidelberg, August 2018.

[BS16] Shalev Ben-David and Or Sattath. Quantum tokens for digital signatures. *arXiv (CoRR)*, abs/1609.09047, 2016.

[CCY20] Nai-Hui Chia, Kai-Min Chung, and Takashi Yamakawa. Classical verification of quantum computations with efficient verifier. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part III*, volume 12552 of *LNCS*, pages 181–206. Springer, Heidelberg, November 2020.

[CGS02] Claude Crépeau, Daniel Gottesman, and Adam Smith. Secure multi-party quantum computation. In *Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing*, STOC '02, page 643–652, New York, NY, USA, 2002. Association for Computing Machinery.

[Chi05] Andrew M. Childs. Secure assisted quantum computation. *Quantum Info. Comput.*, 5(6):456–466, sep 2005.

[CHN$^+$18] Aloni Cohen, Justin Holmgren, Ryo Nishimaki, Vinod Vaikuntanathan, and Daniel Wichs. Watermarking cryptographic capabilities. *SIAM J. Comput.*, 47(6):2157–2202, 2018.

[CLLW22] Kai-Min Chung, Yi Lee, Han-Hsuan Lin, and Xiaodi Wu. Constant-round blind classical verification of quantum sampling. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part III*, volume 13277 of *LNCS*, pages 707–736. Springer, Heidelberg, May / June 2022.

[CLLZ21] Andrea Coladangelo, Jiahui Liu, Qipeng Liu, and Mark Zhandry. Hidden cosets and applications to unclonable cryptography. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 556–584, Virtual Event, August 2021. Springer, Heidelberg.

[CMP20] Andrea Coladangelo, Christian Majenz, and Alexander Poremba. Quantum copy-protection of compute-and-compare programs in the quantum random oracle model. *arXiv (CoRR)*, abs/2009.13865, 2020.

[CVW18] Yilei Chen, Vinod Vaikuntanathan, and Hoeteck Wee. GGH15 beyond permutation branching programs: Proofs, attacks, and candidates. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 577–607. Springer, Heidelberg, August 2018.

[DFM20] Jelle Don, Serge Fehr, and Christian Majenz. The measure-and-reprogram technique 2.0: Multi-round fiat-shamir and more. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 602–631. Springer, Heidelberg, August 2020.

[DFMS19] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Security of the Fiat-Shamir transformation in the quantum random-oracle model. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 356–383. Springer, Heidelberg, August 2019.

[DGH$^+$20] Nico Döttling, Sanjam Garg, Mohammad Hajiabadi, Daniel Masny, and Daniel Wichs. Two-round oblivious transfer from CDH or LPN. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 768–797. Springer, Heidelberg, May 2020.

[DH76] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[DNS10] Frédéric Dupuis, Jesper Buus Nielsen, and Louis Salvail. Secure two-party quantum evaluation of unitaries against specious adversaries. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 685–706. Springer, Heidelberg, August 2010.

[DQV$^+$21] Lalita Devadas, Willy Quach, Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. Succinct LWE sampling, random polynomials, and obfuscation. In Kobbi Nissim and Brent Waters, editors, *TCC 2021, Part II*, volume 13043 of *LNCS*, pages 256–287. Springer, Heidelberg, November 2021.

[DS22] Marcel Dall'Agnol and Nicholas Spooner. On the necessity of collapsing. Cryptology ePrint Archive, Paper 2022/786, 2022. https://eprint.iacr.org/2022/786.

[DSS18] Yfke Dulek, Christian Schaffner, and Florian Speelman. *Theory of Computing*, 14(1):1–45, 2018.

[GGH$^+$16] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. *SIAM J. Comput.*, 45(3):882–929, 2016.

[GP21] Romain Gay and Rafael Pass. Indistinguishability obfuscation from circular security. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, page 736–749, New York, NY, USA, 2021. Association for Computing Machinery.

[GSLW19] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In Moses Charikar and Edith Cohen, editors, *51st ACM STOC*, pages 193–204. ACM Press, June 2019.

[JLS21] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, page 60–73, New York, NY, USA, 2021. Association for Computing Machinery.

[KN22] Fuyuki Kitagawa and Ryo Nishimaki. Watermarking PRFs against quantum adversaries. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part III*, volume 13277 of *LNCS*, pages 488–518. Springer, Heidelberg, May / June 2022.

[KN23] Fuyuki Kitagawa and Ryo Nishimaki. Functional encryption with secure key leasing. In *Advances in Cryptology – ASIACRYPT 2022: 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5–9, 2022, Proceedings, Part IV*, page 569–598, Berlin, Heidelberg, 2023. Springer-Verlag.

[LMS22] Alex Lombardi, Fermi Ma, and Nicholas Spooner. Post-quantum zero knowledge, revisited or: How to do quantum rewinding undetectably. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 851–859, 2022.

[Mah18] Urmila Mahadev. Classical homomorphic encryption for quantum circuits. In Mikkel Thorup, editor, *59th FOCS*, pages 332–338. IEEE Computer Society Press, October 2018.

[Mah22] Urmila Mahadev. Classical verification of quantum computations. *SIAM J. Comput.*, 51(4):1172–1229, 2022.

[RUV13] Ben W. Reichardt, Falk Unger, and Umesh V. Vazirani. Classical command of quantum systems. *Nat.*, 496(7446):456–460, 2013.

[Sho97a] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, oct 1997.

[Sho97b] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *Advances in Cryptology — EUROCRYPT '97*, pages 256–266, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.

[SW21] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: Deniable encryption, and more. *SIAM J. Comput.*, 50(3):857–908, 2021.

[Vid20] Thomas Vidick. Interactions with quantum devices (course), 2020. `http://users.cms.caltech.edu/~vidick/teaching/fsmp/fsmp.pdf`.

[Wel74] L. Welch. Lower bounds on the maximum cross correlation of signals (corresp.). *IEEE Transactions on Information Theory*, 20(3):397–399, 1974.

[Win99] Andreas J. Winter. Coding theorem and strong converse for quantum channels. *IEEE Trans. Inf. Theory*, 45(7):2481–2485, 1999.

[WW21] Hoeteck Wee and Daniel Wichs. Candidate obfuscation via oblivious LWE sampling. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part III*, volume 12698 of *LNCS*, pages 127–156. Springer, Heidelberg, October 2021.

[Zha12] Mark Zhandry. How to construct quantum random functions. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 679–687, 2012.

A Remaining Proofs from Section 4.1

Lemma A.1. *Any Pauli functional commitment that satisfies* single-bit binding with public decodability *also satisfies* string binding with public decodability.

Proof. For this proof, we will need a couple of different binding definitions, as well as a couple of imported theorems.

Definition A.2 (Collapse binding)**.** *A Pauli functional commitment* $(\mathsf{Gen}, \mathsf{Com}, \mathsf{OpenZ}, \mathsf{OpenX}, \mathsf{DecZ}, \mathsf{DecX})$ *satisfies* collapse binding *if the following holds. For any adversary* $\mathsf{A} := \{(\mathsf{C}_\lambda, \mathsf{U}_\lambda)\}_{\lambda\in\mathbb{N}}$*, where each of* C_λ *and* U_λ *are oracle-aided quantum operations that make at most* $\mathrm{poly}(\lambda)$ *oracle queries, define the experiment* $\mathsf{EXP}^{\mathsf{A}}_{\mathsf{CB}}(\lambda)$ *as follows.*

- *Sample* $\mathsf{dk}, |\mathsf{ck}\rangle, \mathsf{CK} \leftarrow \mathsf{Gen}(1^\lambda)$.
- *Run* $\mathsf{C}_\lambda^{\mathsf{CK},\mathsf{DecZ}[\mathsf{dk}],\mathsf{DecX}[\mathsf{dk}]}(|\mathsf{ck}\rangle)$ *until it outputs a commitment c and a state on registers* $(\mathcal{B},\mathcal{U},\mathcal{A})$.
- *Sample $b \leftarrow \{0,1\}$. If $b = 0$, do nothing, and otherwise measure* $(\mathcal{B},\mathcal{U})$ *with* $\{\Pi_{\mathsf{dk},c,0}, \Pi_{\mathsf{dk},c,1}\}$.[22]
- *Run* $\mathsf{U}_\lambda^{\mathsf{CK},\mathsf{DecZ}[\mathsf{dk}]}(\mathcal{B},\mathcal{U},\mathcal{A})$ *until it outputs a bit b'. The experiment outputs 1 if $b = b'$.*

We say that A *is* valid *if the state on* $(\mathcal{B},\mathcal{U})$ *output by* C_λ *is in the image of* $\Pi_{\mathsf{dk},c,0} + \Pi_{\mathsf{dk},c,1}$*. Then, it must hold that for all valid adversaries* A,

$$\left|\Pr\left[\mathsf{EXP}^{\mathsf{A}}_{\mathsf{CB}}(\lambda) = 1\right] - \frac{1}{2}\right| = \mathrm{negl}(\lambda).$$

Definition A.3 (Unique message binding)**.** *A Pauli functional commitment* $(\mathsf{Gen}, \mathsf{Com}, \mathsf{OpenZ}, \mathsf{OpenX}, \mathsf{DecZ}, \mathsf{DecX})$ *satisfies* unique message binding *if for any polynomial $m(\lambda)$ and any adversary* $\{(\mathsf{C}_\lambda, \mathsf{U}_\lambda)\}_{\lambda\in\mathbb{N}}$*, where each of* C_λ *and* U_λ *are oracle-aided quantum operations that make at most* $\mathrm{poly}(\lambda)$ *oracle queries, the following experiment outputs 1 with probability* $\mathrm{negl}(\lambda)$.

- *Sample* $\{\mathsf{dk}_i, |\mathsf{ck}_i\rangle, \mathsf{CK}_i \leftarrow \mathsf{Gen}(1^\lambda)\}_{i\in[m]}$.
- *Run* $\mathsf{C}_\lambda^{\mathbf{CK},\mathsf{DecZ}[\mathbf{dk}],\mathsf{DecX}[\mathbf{dk}]}(|\mathbf{ck}\rangle)$ *until it outputs a commitment* $\mathbf{c} := (c_1, \ldots, c_m)$*, a message* $x_1 \in \{0,1\}^m$*, and a state on registers* $(\mathcal{B}_1,\mathcal{U}_1, \ldots, \mathcal{B}_m,\mathcal{U}_m, \mathcal{A})$.
- *For each $i \in [m]$, apply* $\Pi_{\mathsf{dk}_i,c_i,x_{1,i}}$ *to* $(\mathcal{B}_i,\mathcal{U}_i)$ *and abort and output 0 if this projection rejects.*
- *Run* $\mathsf{U}_\lambda^{\mathbf{CK},\mathsf{DecZ}[\mathbf{dk}]}(\mathcal{B}_1,\mathcal{U}_1, \ldots, \mathcal{B}_m,\mathcal{U}_m, \mathcal{A})$ *until it outputs a message* $x_2 \in \{0,1\}^m$*, and a state on registers* $(\mathcal{B}_1,\mathcal{U}_1, \ldots, \mathcal{B}_m,\mathcal{U}_m)$*. If $x_1 = x_2$, abort and output 0.*
- *For each $i \in [m]$, apply* $\Pi_{\mathsf{dk}_i,c_i,x_{2,i}}$ *to* $(\mathcal{B}_i,\mathcal{U}_i)$ *and abort and output 0 if this projection rejects. Otherwise, output 1.*

Imported Theorem A.4 ([LMS22])**.** *Any commitment that satisfies collapse binding also satisfies unique message binding.*

Imported Theorem A.5 ([DS22])**.** *Let* D *be a projector,* Π_0, Π_1 *be orthogonal projectors, and* $|\psi\rangle \in \mathsf{Im}(\Pi_0 + \Pi_1)$*. Then,*

$$\|\Pi_1 \mathsf{D} \Pi_0 |\psi\rangle\|^2 + \|\Pi_0 \mathsf{D} \Pi_1 |\psi\rangle\|^2 \geq \frac{1}{2}\left(\|\mathsf{D}|\psi\rangle\|^2 - \left(\|\mathsf{D}\Pi_0|\psi\rangle\|^2 + \|\mathsf{D}\Pi_1|\psi\rangle\|^2\right)\right)^2.$$

Given these imported theorems, the proof of our lemma is quite straightforward.

[22]These projectors are defined in Definition 4.3.

- First, we establish using Imported Theorem A.5 that any Pauli functional commitment that satisfies single-bit binding also satisfies collapse binding. To see this, suppose there exists an adversary (C, U) that breaks collapse binding, let $\Pi_0 = \Pi_{\mathsf{dk},c,0}$, $\Pi_1 = \Pi_{\mathsf{dk},c,1}$, let D be a projective implementation of $\mathsf{U}^{\mathsf{CK},\mathsf{DecZ}[\mathsf{dk}]}$, and let $|\psi\rangle$ be the state of the collapse binding experiment that is output by $\mathsf{C}^{\mathsf{CK},\mathsf{DecZ}[\mathsf{dk}],\mathsf{DecX}[\mathsf{dk}]}$. Then the RHS of Imported Theorem A.5 is half the squared advantage of the adversary in the collapse binding game. This implies that at least one of the terms on the LHS is non-negligible, which immediately implies that this adversary can be used to break the single-bit binding game.

- Next, appealing to Imported Theorem A.4, we see that any Pauli functional commitment that satisfies single-bit binding also satisfies unique message binding.

- Finally, suppose there is a Pauli functional commitment that is single-bit binding, but there exists an adversary that breaks the string binding of this commitment for some pair of disjoint sets W_0, W_1. We define an experiment where we insert a measurement of $\mathsf{DecZ}(\mathbf{dk}, \mathbf{c}, \cdot)$ applied to the state $\Pi_{\mathbf{dk},\mathbf{c},W_0} |\psi\rangle$, which by definition will return some string $x_0 \in W_0$. By the collapse binding of the commitment, inserting this measurement will only have a negligible affect on the experiment. But now, since W_0 and W_1 are disjoint sets, this adversary breaks the unique message binding of the commitment. This completes the proof.

□

B Remaining Proofs from Section 4.3

In this appendix, we prove the following theorem.

Theorem B.1. *Let $n, m, d \in \mathbb{N}, \epsilon \in (0, 1/8)$ be such that $d \geq 2$ and $n - d + 1 > 10\log(1/\epsilon) + 6$. Let $\mathsf{U}^{\mathcal{X},\mathcal{Y}}$ be any (2^{n+m})-dimensional unitary, where register $\mathcal{X}$ is 2^n dimensions and register $\mathcal{Y}$ is 2^m dimensions. Let $\mathcal{A}$ be the set of d-dimensional balanced affine subspaces $A = A_0 \cup A_1$ of $\mathbb{F}_2^n$, where A_0 is the affine subspace of vectors in A that start with 0 and A_1 is the affine subspace of vectors in A that start with 1. For any $A = A_0 \cup A_1$, let*

$$\Pi_{A_0} := \sum_{v \in A_0} |v\rangle\langle v|^{\mathcal{X}} \otimes \mathbb{I}^{\mathcal{Y}}, \quad \Pi_{A_1} := \mathsf{U}^\dagger \left(\sum_{v \in A_1} |v\rangle\langle v|^{\mathcal{X}} \otimes \mathbb{I}^{\mathcal{Y}} \right) \mathsf{U}.$$

Let $\mathcal{R}$ be the set of pairs (A, B) of d-dimensional affine subspaces of $\mathbb{F}_2^n$ such that $\dim(A_0 \cap B_0) = d-2$ and $\dim(A_1 \cap B_1) = d - 2$. Then for any set of states $\{|\psi_A\rangle\}_A$ such that for all $A \in \mathcal{A}$, $|\psi_A\rangle \in \mathsf{Im}(\Pi_{A_0})$, and $\|\Pi_{A_1} |\psi_A\rangle\| \geq \epsilon$,

$$\mathop{\mathbb{E}}_{(A,B)\leftarrow\mathcal{R}}[|\langle\psi_A|\psi_B\rangle|] < \frac{1}{2} - \epsilon^{13}.$$

We will first simplify the problem by reducing to the case where each A is two-dimensional, consisting of just four vectors. This case is proven later in Appendix B.1. In the reduction, which follows below, we begin with the observation that each $(A, B) \in \mathcal{R}$ consists of six cosets of a particular $(d-2)$-dimensional subspace S. Then, we partition $\mathcal{R}$ based on this underlying subspace, and prove the claim separately for each S. Finally, the process of sampling (A, B) from $\mathcal{R}$ conditioned on an underlying subspace S can be seen as sampling A and B as two-dimensional spaces in the subspace of cosets of S.

Proof. (of Theorem B.1) First, note that for any $(A, B) \in \mathcal{R}$, $A_0 \cap B_0$ is an intersection of affine subspaces, so is an affine subspace itself. So, we write $A_0 \cap B_0 = S + v_0$ for some $(d-2)$-dimensional subspace S. Since all vectors in $S + v_0$ start with 0, it must be the case that all vectors in S start with 0 and v_0 starts with 0. Moreover, $A = A_0 \cup A_1$ and $B = B_0 \cup B_1$ are both cosets of superspaces of S, and thus we can write

$$A = (S+v_0) \cup (S+w_0) \cup (S+v_1) \cup (S+w_1), \quad B = (S+v_0) \cup (S+u_0) \cup (S+v_1) \cup (S+u_1)$$

for v_0, w_0, u_0 that start with 0, v_1, w_1, u_1 that start with 1, and where $v_0 + w_0 = v_1 + w_1$ and $v_0 + u_0 = v_1 + u_1$.

Now, for any $(d-2)$-dimensional subspace $S := \mathsf{span}(z_1, \dots, z_{d-2})$ such all vectors in S start with 0, let $z_{d-1}, \dots, z_n$ be such that $(z_1, \dots, z_n)$ is an orthonormal basis of $\mathbb{F}_2^n$ and z_{d-1} is the only basis vector that starts with 1. Define the subspace $\mathsf{co}(S) := \mathsf{span}(z_{d-1}, \dots, z_n)$. Furthermore, let $\mathsf{co}(S)_0$ be the subspace of vectors in $\mathsf{co}(S)$ that start with 0, and let $\mathsf{co}(S)_1$ be the affine subspace of vectors in $\mathsf{co}(S)$ that starts with 1.

Then we can sample from $\mathcal{R}$ by first sampling a random $(d-2)$-dimensional subspace S such that all vectors in S start with 0, then sampling distinct $v_0, w_0, u_0 \leftarrow \mathsf{co}(S)_0$ and distinct $v_1, w_1, u_1 \leftarrow \mathsf{co}(S)_1$ such that $v_0 + w_0 = v_1 + w_1$ and $v_0 + u_0 = v_1 + u_1$, and finally setting

$$A = (S+v_0) \cup (S+w_0) \cup (S+v_1) \cup (S+w_1), \quad B = (S+v_0) \cup (S+u_0) \cup (S+v_1) \cup (S+u_1)$$

For any subspace S, let $\mathcal{R}[S]$ be the set of $(A, B) \in \mathcal{R}$ such that $A_0 \cap B_0$ is a coset of S. Thus, it suffices to prove that for *each fixed* S,

$$\mathop{\mathbb{E}}_{(A,B) \leftarrow \mathcal{R}[S]}[|\langle \psi_A | \psi_B \rangle|] < \frac{1}{2} - \epsilon^{13}.$$

Now consider any fixed S. For each A that could be sampled by $\mathcal{R}[S]$, we write

$$A = (S+v_0) \cup (S+w_0) \cup (S+v_1) \cup (S+w_1)$$

for $v_0, w_0 \in \mathsf{co}(S)_0$ and $v_1, w_1 \in \mathsf{co}(S)_1$ such that $v_0 + w_0 = v_1 + w_1$. Moreover, we can express v_0, w_0 as $(0, v_0'), (0, w_0') \in \mathbb{F}_2^{n-d+2}$ and v_1, w_1 as $(1, v_1'), (1, w_1') \in \mathbb{F}_2^{n-d+2}$ in the $(z_{d-1}, \dots, z_n)$-basis. Thus we can associate each A with vectors $v_0', w_0', v_1', w_1' \in \mathbb{F}_2^{n-d+1}$ such that $v_0' + w_0' = v_1' + w_1'$.

Let $\mathsf{U}_{S,\mathsf{co}(S)}$ be the unitary that implements the change of basis $(e_1, \dots, e_n) \to (z_1, \dots, z_n)$, where the e_i are the standard basis vectors, and let

$$\widetilde{\mathsf{U}} := \left(\mathsf{U}_{S,\mathsf{co}(S)} \otimes \mathbb{I}^{\mathcal{Y}}\right) \mathsf{U}^{\mathcal{X},\mathcal{Y}} \left(\mathsf{U}^\dagger_{S,\mathsf{co}(S)} \otimes \mathbb{I}^{\mathcal{Y}}\right).$$

Then, re-defining

$$\begin{aligned}
|\widetilde{\psi}_A\rangle &:= \mathsf{U}_{S,\mathsf{co}(S)} |\psi_A\rangle, \\
\widetilde{\Pi}_{A_0} &:= \mathbb{I}^{\otimes d-2} \otimes |0\rangle\langle 0| \otimes \left(|v_0'\rangle\langle v_0'| + |w_0'\rangle\langle w_0'|\right) \otimes \mathbb{I}^{\mathcal{Y}}, \\
\widetilde{\Pi}_{A_1} &:= \widetilde{\mathsf{U}}^\dagger \left(\mathbb{I}^{\otimes d-2} \otimes |1\rangle\langle 1| \otimes \left(|v_1'\rangle\langle v_1'| + |w_1'\rangle\langle w_1'|\right) \otimes \mathbb{I}^{\mathcal{Y}}\right) \widetilde{\mathsf{U}},
\end{aligned}$$

we have that $|\widetilde{\psi}_A\rangle \in \mathsf{Im}(\widetilde{\Pi}_{A_0})$ and $\|\widetilde{\Pi}_{A_1}|\widetilde{\psi}_A\rangle\| \geq \epsilon$ for all A that could be sampled by $\mathcal{R}[S]$. Moreover, we can replace the projections on the $d-1$'st qubit with identities, defining

$$\widetilde{\Pi}'_{A_0} := \mathbb{I}^{\otimes d-1} \otimes \left(|v'_0\rangle\langle v'_0| + |w'_0\rangle\langle w'_0|\right) \otimes \mathbb{I}^{\mathcal{Y}},$$
$$\widetilde{\Pi}'_{A_1} := \widetilde{\mathsf{U}}^\dagger \left(\mathbb{I}^{\otimes d-1} \otimes \left(|v'_1\rangle\langle v'_1| + |w'_1\rangle\langle w'_1|\right) \otimes \mathbb{I}^{\mathcal{Y}}\right) \widetilde{\mathsf{U}},$$

and still have that $|\widetilde{\psi}_A\rangle \in \mathsf{Im}(\widetilde{\Pi}'_{A_0})$ and $\|\widetilde{\Pi}'_{A_1}|\widetilde{\psi}_A\rangle\| \geq \epsilon$ for all A that could be sampled by $\mathcal{R}[S]$. Thus, we have reduced this problem to the "two-dimensional" case, which is covered in the next section. Since $n-d+1 > 10\log(1/\epsilon)+6$, Theorem B.2 implies that

$$\mathop{\mathbb{E}}_{(A,B)\leftarrow\mathcal{R}[S]}[|\langle\widetilde{\psi}_A|\widetilde{\psi}_B\rangle|] < \frac{1}{2} - \epsilon^{13},$$

which implies that

$$\mathop{\mathbb{E}}_{(A,B)\leftarrow\mathcal{R}[S]}[|\langle\psi_A|\psi_B\rangle|] < \frac{1}{2} - \epsilon^{13},$$

completing the proof.

□

B.1 Two-dimensional case

Theorem B.2. *Let $n, m \in \mathbb{N}, \epsilon \in (0, 1/8)$ be such that $n > 10\log(1/\epsilon)+6$. Let $\mathsf{U}^{\mathcal{X},\mathcal{Y}}$ be a (2^{n+m})-dimensional unitary, where register $\mathcal{X}$ is 2^n dimensions and register $\mathcal{Y}$ is 2^m dimensions. Let $\mathcal{A}$ be the set of pairs of sets $(\{v_0, w_0\}, \{v_1, w_1\})$ such that $v_0, w_0, v_1, w_1 \in \mathbb{F}_2^n$ and $v_0 + w_0 = v_1 + w_1$.*[23] *We will write any $A \in \mathcal{A}$ as $A := (A_0, A_1)$, where $A_0 := \{v_0, w_0\}$ and $A_1 = \{v_1, w_1\}$. For any such A, let*

$$\Pi_{A_0} := (|v_0\rangle\langle v_0| + |w_0\rangle\langle w_0|)^{\mathcal{X}} \otimes \mathbb{I}^{\mathcal{Y}}, \quad \Pi_{A_1} := \mathsf{U}^\dagger\left((|v_1\rangle\langle v_1| + |w_1\rangle\langle w_1|)^{\mathcal{X}} \otimes \mathbb{I}^{\mathcal{Y}}\right)\mathsf{U}.$$

Let $\mathcal{R}$ be the set of pairs (A, B) such that $|A_0 \cap B_0| = 1$ and $|A_1 \cap B_1| = 1$. Then for any set of states $\{|\psi_A\rangle\}_A$ such that for all $A \in \mathcal{A}$, $|\psi_A\rangle \in \mathsf{Im}(\Pi_{A_0})$ and $\|\Pi_{A_1}|\psi_A\rangle\| \geq \epsilon$,

$$\mathop{\mathbb{E}}_{(A,B)\leftarrow\mathcal{R}}[|\langle\psi_A|\psi_B\rangle|] < \frac{1}{2} - \epsilon^{13}.$$

First, we provide a high-level overview the proof. We note that it is easy to show that

$$\mathop{\mathbb{E}}_{(A,B)\leftarrow\mathcal{R}}[|\langle\psi_A|\psi_B\rangle|] \leq \frac{1}{2},$$

which only requires the condition that for all $A \in \mathcal{A}$, $|\psi_A\rangle \in \mathsf{Im}(\Pi_{A_0})$. Adding the condition that $\|\Pi_{A_1}|\psi_A\rangle\| \geq \epsilon$ should intuitively only decrease this expected inner product, since many of the

[23]Note that this theorem is not strictly the two-dimensional version of Theorem B.1, since $\mathcal{A}$ is not exactly defined to be the set of two-dimensional affine subspaces. Rather it consists of pairs of two sets $\{v_0, w_0\}, \{v_1, w_1\}$ where the vectors are arbitrary but satisfy $v_0 + w_0 = v_1 + w_1$. That is, v_0, w_0, v_1, w_1 here play the role of v'_0, w'_0, v'_1, w'_1 in the proof of Theorem B.1, and in particular v_0, w_0 do not necessarily start with 0 and v_1, w_1 do not necessarily start with 1.

Π_{A_1} are orthogonal. In particular, for any A_0, all the Π_{A_1} such that $(A_0, A_1) \in \mathcal{A}$ are orthogonal. To formalize this intuition, we proceed by contradiction, and assume that

$$\mathop{\mathbb{E}}_{(A,B)\leftarrow\mathcal{R}}[|\langle\psi_A|\psi_B\rangle|] \geq \frac{1}{2} - \epsilon^{13}.$$

For each $A = (\{v_0, w_0\}, \{v_1, w_1\})$, we will write $|\psi_A\rangle$ as

$$|\psi_A\rangle := \alpha_A^{v_0}|v_0\rangle^{\mathcal{X}}|\phi_A^{v_0}\rangle^{\mathcal{Y}} + \alpha_A^{w_0}|w_0\rangle^{\mathcal{X}}|\phi_A^{w_0}\rangle^{\mathcal{Y}},$$

and note that

$$\mathop{\mathbb{E}}_{(A,B)\leftarrow\mathcal{R}}[|\langle\psi_A|\psi_B\rangle|] \leq \mathop{\mathbb{E}}_{(A,B)\leftarrow\mathcal{R}}[|\alpha_A^{v_{A,B}}|\cdot|\alpha_B^{v_{A,B}}|\cdot|\langle\phi_A^{v_{A,B}}|\phi_B^{v_{A,B}}\rangle|],$$

where $\{v_{A,B}\} := A_0 \cap B_0$.

Then, we proceed via the following steps.

1. If we only require that $|\psi_A\rangle \in \mathsf{Im}(\Pi_{A_0})$, then one way to obtain the maximum expected inner product of $1/2$ is to set each $|\alpha_A^{v_0}| = 1/\sqrt{2}$ and for each v_0, let all $|\phi_A^{v_0}\rangle$ be the same vector. Then, each $|\alpha_A^{v_{A,B}}|\cdot|\alpha_B^{v_{A,B}}| = 1/2$ and each $|\langle\phi_A^{v_{A,B}}|\phi_B^{v_{A,B}}\rangle| = 1$. We show that this way of defining the $\alpha_A^{v_0}$ is "robust" in the sense that if the expected inner product is *close* to $1/2$, then for *many* of the (A, B), $|\alpha_A^{v_{A,B}}|\cdot|\alpha_B^{v_{A,B}}|$ is *close* to $1/2$ (Claim B.3).

2. We show that Step 1 implies that this way of defining $|\phi_A^{v_0}\rangle$ is also "robust", in the sense that for *many* of the (A, B), $|\langle\phi_A^{v_{A,B}}|\phi_B^{v_{A,B}}\rangle|$ is *close* to 1 (Claim B.4). Thus, this property must be satisfied if our expected inner product is at least $1/2 - \epsilon^{13}$.

3. By analyzing the graph of "connections" induced by $\mathcal{R}$ between the elements of $\mathcal{A}$, we show that Step 2 implies that there must exist *some* $A_0^* = \{v_0^*, w_0^*\}$ with the following property. There any *many* (exponential in n) states

$$\left\{|\psi_{(A_0^*,A_1)}\rangle := \alpha_{(A_0^*,A_1)}^{v_0^*}|v_0^*\rangle|\phi_{(A_0^*,A_1)}^{v_0^*}\rangle + \alpha_{(A_0^*,A_1)}^{w_0^*}|w_0^*\rangle|\phi_{(A_0^*,A_1)}^{w_0^*}\rangle\right\}_{A_1:(A_0^*,A_1)\in\mathcal{A}}$$

   such that the $\{|\phi_{(A_0^*,A_1)}^{v_0^*}\rangle\}$ are all close to each other, *and* the $\{|\phi_{(A_0^*,A_1)}^{w_0^*}\rangle\}$ are all close to each other (Claim B.5).

4. Step 3 implies that there exists a *large* (exponential in n) collection of states $|\psi_{(A_0^*,A_1)}\rangle$ such that (i) all $|\psi_{(A_0^*,A_1)}\rangle$ are *close* to the *same two-dimensional subspace*, and (ii) each $|\psi_{(A_0^*,A_1)}\rangle$ has ϵ overlap with a *different orthogonal subspace* Π_{A_1}. We complete the proof by showing that this is impossible when n is large enough compared to $1/\epsilon$. This relies on a Welch bound, which bounds the number of distinct vectors of some minimum distance from each other that can be packed into a low-dimensional subspace.

Proof. (of Theorem B.2) Assume that

$$\mathop{\mathbb{E}}_{(A,B)\leftarrow\mathcal{R}}[|\langle\psi_A|\psi_B\rangle|] \geq \frac{1}{2} - \epsilon^{13}.$$

Using the fact that each $|\psi_A\rangle \in \mathsf{Im}(\Pi_{A_0})$, write each

$$|\psi_A\rangle \coloneqq \alpha_A^{v_0} |v_0\rangle^{\mathcal{X}} |\phi_A^{v_0}\rangle^{\mathcal{Y}} + \alpha_A^{w_0} |w_0\rangle^{\mathcal{X}} |\phi_A^{w_0}\rangle^{\mathcal{Y}},$$

where $A_0 = \{v_0, w_0\}$. For any $(A,B) \in \mathcal{R}$, define $\{v_{A,B}\} = A_0 \cap B_0$. Then, we have the following series of inequalities.

$$\begin{aligned}
\frac{1}{2} - \epsilon^{13} &\leq \mathop{\mathbb{E}}_{(A,B)\leftarrow\mathcal{R}}[|\langle\psi_A|\psi_B\rangle|] \\
&= \mathop{\mathbb{E}}_{(A,B)\leftarrow\mathcal{R}}[|\alpha_A^{v_{A,B}*}\alpha_B^{v_{A,B}}\langle\phi_A^{v_{A,B}}|\phi_B^{v_{A,B}}\rangle|] \\
&\leq \mathop{\mathbb{E}}_{(A,B)\leftarrow\mathcal{R}}[|\alpha_A^{v_{A,B}}|\cdot|\alpha_B^{v_{A,B}}|\cdot|\langle\phi_A^{v_{A,B}}|\phi_B^{v_{A,B}}\rangle|] \\
&\leq \mathop{\mathbb{E}}_{(A,B)\leftarrow\mathcal{R}}[|\alpha_A^{v_{A,B}}|\cdot|\alpha_B^{v_{A,B}}|].
\end{aligned}$$

Next, we show the following.

Claim B.3.

$$\Pr_{(A,B)\leftarrow\mathcal{R}}\left[|\alpha_A^{v_{A,B}}|\cdot|\alpha_B^{v_{A,B}}| \geq \frac{1}{2} - 2\epsilon^2\right] \geq 1 - \epsilon^6.$$

Proof. First, note that for any $(A,B) \in \mathcal{R}$ where $A = (\{v_0,w_0\},\{v_1,w_1\})$ and $B = (\{v_0,u_0\},\{v_1,u_1\})$, the set $C = (\{w_0,u_0\},\{w_1,u_1\}) \in \mathcal{A}$. This follows because

$$\begin{aligned}
&A \in \mathcal{A} \implies v_0 + w_0 = v_1 + w_1 \implies w_0 = v_0 + v_1 + w_1 \\
&B \in \mathcal{A} \implies v_0 + u_0 = v_1 + w_1 \implies u_0 = v_0 + v_1 + u_1, \\
&\text{so } w_0 + u_0 = w_1 + u_1 \implies C \in \mathcal{A}.
\end{aligned}$$

This means that each $(A,B) \in \mathcal{R}$ uniquely define a $C \in \mathcal{A}$ such that all

$$(A,B),(B,C),(C,A) \in \mathcal{R}.$$

Thus, we will imagine sampling $(A,B) \leftarrow \mathcal{R}$ as follows. First, sample distinct $v_0, w_0, u_0 \leftarrow \mathbb{F}_2^n$. Then, sample v_1, w_1, u_1 such that

$$C_1 \coloneqq (\{v_0,w_0\},\{v_1,w_1\}),\ \ C_2 \coloneqq (\{v_0,u_0\},\{v_1,u_1\}),\ \ C_3 \coloneqq (\{w_0,u_0\},\{w_1,u_1\}) \in \mathcal{A}.$$

Let $(C_1,C_2,C_3) \leftarrow \mathcal{S}$ denote this sampling procedure. Finally, choose

$$(A,B) \leftarrow \mathcal{R}[C_1,C_2,C_3] \coloneqq \{(C_1,C_2),(C_2,C_3),(C_3,C_1)\}.$$

Let

$$E[C_1,C_2,C_3] \coloneqq \mathop{\mathbb{E}}_{(A,B)\leftarrow\mathcal{R}[C_1,C_2,C_3]}\left[|\alpha_A^{v_{A,B}}|\cdot|\alpha_B^{v_{A,B}}|\right].$$

Then,

$$\mathop{\mathbb{E}}_{(A,B)\leftarrow\mathcal{R}}\left[|\alpha_A^{v_{A,B}}|\cdot|\alpha_B^{v_{A,B}}|\right] = \mathop{\mathbb{E}}_{(C_1,C_2,C_3)\leftarrow\mathcal{S}}[E[C_1,C_2,C_3]] \geq \frac{1}{2} - \epsilon^{13} > \frac{1}{2} - \epsilon^{12}.$$

Now, given any (C_1, C_2, C_3) and corresponding

$$\begin{aligned}
|\psi_{C_1}\rangle &\coloneqq \alpha^{v_0}_{C_1} |v_0\rangle |\phi^{v_0}_{C_1}\rangle + \alpha^{w_0}_{C_1} |w_0\rangle |\phi^{w_0}_{C_1}\rangle , \\
|\psi_{C_2}\rangle &\coloneqq \alpha^{v_0}_{C_2} |v_0\rangle |\phi^{v_0}_{C_2}\rangle + \alpha^{u_0}_{C_2} |u_0\rangle |\phi^{u_0}_{C_2}\rangle , \\
|\psi_{C_3}\rangle &\coloneqq \alpha^{w_0}_{C_3} |w_0\rangle |\phi^{w_0}_{C_3}\rangle + \alpha^{u_0}_{C_3} |u_0\rangle |\phi^{u_0}_{C_3}\rangle ,
\end{aligned}$$

we have that

$$E[C_1, C_2, C_3] \le \frac{1}{3}\left(|\alpha^{v_0}_{C_1}| \cdot |\alpha^{v_0}_{C_2}| + |\alpha^{w_0}_{C_1}| \cdot |\alpha^{w_0}_{C_3}| + |\alpha^{u_0}_{C_2}| \cdot |\alpha^{u_0}_{C_3}|\right).$$

By Fact B.8, $E[C_1, C_2, C_3] \le 1/2$, so by Markov,

$$\Pr_{(C_1,C_2,C_3)\leftarrow\mathcal{S}}\left[\frac{1}{2} - E[C_1, C_2, C_3] \ge \epsilon^6\right] \le \epsilon^6 \implies \Pr_{(C_1,C_2,C_3)\leftarrow\mathcal{S}}\left[E[C_1, C_2, C_3] \ge \frac{1}{2} - \epsilon^6\right] \ge 1 - \epsilon^6.$$

Moreover, whenever $E[C_1, C_2, C_3] \ge 1/2 - \epsilon^6$, we have that

$$|\alpha^{v_0}_{C_1}| \cdot |\alpha^{v_0}_{C_2}| + |\alpha^{w_0}_{C_1}| \cdot |\alpha^{w_0}_{C_3}| + |\alpha^{u_0}_{C_2}| \cdot |\alpha^{u_0}_{C_3}| \ge \frac{3}{2} - \frac{6\epsilon^6}{2},$$

so by Fact B.8,

$$|\alpha^{v_0}_{C_1}| \cdot |\alpha^{v_0}_{C_2}|, \;\; |\alpha^{w_0}_{C_1}| \cdot |\alpha^{w_0}_{C_3}|, \;\; |\alpha^{u_0}_{C_2}| \cdot |\alpha^{u_0}_{C_3}| \ge \frac{1}{2} - 2\epsilon^2,$$

which completes the proof of the claim.

□

Claim B.4.

$$\Pr_{(A,B)\leftarrow\mathcal{R}}\left[|\langle\phi^{v_{A,B}}_A|\phi^{v_{A,B}}_B\rangle| \ge 1 - \epsilon^6\right] \ge 1 - 2\epsilon^6.$$

Proof. First, note that the proof of Claim B.3 also shows that

$$\mathop{\mathbb{E}}_{(A,B)\leftarrow\mathcal{R}}\left[|\alpha^{v_{A,B}}_A| \cdot |\alpha^{v_{A,B}}_B|\right] \le \frac{1}{2},$$

since each $E[C_1, C_2, C_3] \le 1/2$.

By our assumption that

$$\mathop{\mathbb{E}}_{(A,B)\leftarrow\mathcal{R}}\left[|\alpha^{v_{A,B}}_A| \cdot |\alpha^{v_{A,B}}_B| \cdot |\langle\phi^{v_{A,B}}_A|\phi^{v_{A,B}}_B\rangle|\right] \ge \frac{1}{2} - \epsilon^{13}$$

and linearity of expectation,

$$\mathop{\mathbb{E}}_{(A,B)\leftarrow\mathcal{R}}\left[|\alpha^{v_{A,B}}_A| \cdot |\alpha^{v_{A,B}}_B| \cdot \left(1 - |\langle\phi^{v_{A,B}}_A|\phi^{v_{A,B}}_B\rangle|\right)\right] \le \epsilon^{13}.$$

Now, assume for contradiction that

$$\Pr_{(A,B)\leftarrow\mathcal{R}}\left[|\langle\phi^{v_{A,B}}_A|\phi^{v_{A,B}}_B\rangle| < 1 - \epsilon^6\right] > 2\epsilon^6 \implies \Pr_{(A,B)\leftarrow\mathcal{R}}\left[1 - |\langle\phi^{v_{A,B}}_A|\phi^{v_{A,B}}_B\rangle| > \epsilon^6\right] > 2\epsilon^6.$$

By Claim B.3, this implies that

$$\Pr_{(A,B)\leftarrow\mathcal{R}}\left[\left(1-|\langle\phi_A^{v_{A,B}}|\phi_B^{v_{A,B}}\rangle|>\epsilon^6\right)\wedge\left(|\alpha_A^{v_{A,B}}|\cdot|\alpha_B^{v_{A,B}}|\geq\frac{1}{2}-2\epsilon^2\right)\right]\geq\epsilon^6.$$

But then,

$$\mathop{\mathbb{E}}_{(A,B)\leftarrow\mathcal{R}}\left[|\alpha_A^{v_{A,B}}|\cdot|\alpha_B^{v_{A,B}}|\cdot\left(1-|\langle\phi_A^{v_{A,B}}|\phi_B^{v_{A,B}}\rangle|\right)\right]>\epsilon^6\cdot\epsilon^6\cdot\left(\frac{1}{2}-2\epsilon^2\right)\geq\frac{\epsilon^{12}}{4}>\epsilon^{13},$$

whenever $\epsilon<1/4$.

□

Claim B.5. *There exists an $A_0^*=\{v_0^*,w_0^*\}$ and two unit vectors $|\tau^{v_0^*}\rangle,|\tau^{w_0^*}\rangle$ such that the following holds. Let*

$$\left\{|\psi_{(A_0^*,A_1)}\rangle:=\alpha_{(A_0^*,A_1)}^{v_0^*}|v_0^*\rangle|\phi_{(A_0^*,A_1)}^{v_0^*}\rangle+\alpha_{(A_0^*,A_1)}^{w_0^*}|w_0^*\rangle|\phi_{(A_0^*,A_1)}^{w_0^*}\rangle\right\}_{A_1:(A_0^*,A_1)\in\mathcal{A}}$$

be the set of 2^{n-1} states indexed by A_1 such that $(A_0^*,A_1)\in\mathcal{A}$.[24] *Then there exists a set $\mathcal{A}_1^*$ of size at least 2^{n-2} such that for all $A_1\in\mathcal{A}_1^*$,*

$$|\langle\phi_{(A_0^*,A_1)}^{v_0^*}|\tau^{v_0^*}\rangle|\geq 1-2\epsilon^3\ \text{ and }\ |\langle\phi_{(A_0^*,A_1)}^{w_0^*}|\tau^{w_0^*}\rangle|\geq 1-2\epsilon^3.$$

Proof. For each ordered pair (v_0,w_0) where $v_0\neq w_0\in\mathbb{F}_2^n$, define

$$\mathcal{R}[(v_0,w_0)]:=\{(A,B)\in\mathcal{R}:A_0=\{v_0,w_0\}\wedge v_{A,B}=v_0\}.$$

Then Claim B.4 implies that there exists some set $\{v_0^*,w_0^*\}$ such that

$$\Pr_{(A,B)\leftarrow\mathcal{R}[(v_0^*,w_0^*)]}\left[|\langle\phi_A^{v_{A,B}}|\phi_B^{v_{A,B}}\rangle|=|\langle\phi_A^{v_0^*}|\phi_B^{v_0^*}\rangle|\geq 1-\epsilon^6\right]\geq 1-4\epsilon^6,\ \text{ and}$$
$$\Pr_{(A,B)\leftarrow\mathcal{R}[(w_0^*,v_0^*)]}\left[|\langle\phi_A^{v_{A,B}}|\phi_B^{v_{A,B}}\rangle|=|\langle\phi_A^{w_0^*}|\phi_B^{w_0^*}\rangle|\geq 1-\epsilon^6\right]\geq 1-4\epsilon^6.$$

Let $A_0^*=\{v_0^*,w_0^*\}$, let $\mathcal{A}_1:=\{\{v_1,w_1\}\}_{v_1+w_1=v_0^*+w_0^*}$ be the set of A_1 such that $(A_0^*,A_1)\in\mathcal{A}$, let

$$\left\{|\psi_{(A_0^*,A_1)}\rangle:=\alpha_{(A_0^*,A_1)}^{v_0^*}|v_0^*\rangle|\phi_{(A_0^*,A_1)}^{v_0^*}\rangle+\alpha_{(A_0^*,A_1)}^{w_0^*}|w_0^*\rangle|\phi_{(A_0^*,A_1)}^{w_0^*}\rangle\right\}_{A_1\in\mathcal{A}_1},$$

and let

$$\mathcal{A}_1^{\times 2}=\{\{A_1,A_1'\}\}_{A_1\neq A_1'\in\mathcal{A}_1}.$$

Note that by the definition of $\mathcal{A}_1$, for any $\{A_1,A_1'\}\in\mathcal{A}_1^{\times 2}$, it holds that $A_1\cap A_1'=\emptyset$. Now, we will argue that there exists a vector $|\tau^{v_0^*}\rangle$ and a set $\mathcal{A}_1^{v_0^*}$ of size at least $\frac{3}{4}2^{n-1}$ such that for all $A_1\in\mathcal{A}_1^*$,

$$|\langle\phi_{(A_0^*,A_1)}^{v_0^*}|\tau^{v_0^*}\rangle|\geq 1-2\epsilon^3.$$

Consider any $\{A_1,A_1'\}\in\mathcal{A}_1^{\times 2}$, where $A_1=\{v_1,w_1\}$ and $A_1'=\{v_1',w_1'\}$. There are exactly four B such that

$$((A_0^*,A_1),B)\in\mathcal{R}[(v_0^*,w_0^*)]\ \text{ and }\ ((A_0^*,A_1'),B)\in\mathcal{R}[(v_0^*,w_0^*)],$$

[24]Note that there are 2^{n-1} possible states because the A_1 partition of the set $\mathbb{F}_2^n$ into disjoint unordered pairs of vectors, where each pair $\{v_1,w_1\}$ is such that $v_1+w_1=v_0^*+w_0^*$.

which are[25]

$$B \in \left\{ \begin{array}{l} (\{v_0^*, v_0^* + v_1 + v_1'\}, \{v_1, v_1'\}), \\ (\{v_0^*, v_0^* + w_1 + w_1'\}, \{w_1, w_1'\}), \\ (\{v_0^*, v_0^* + v_1 + w_1'\}, \{v_1, w_1'\}), \\ (\{v_0^*, v_0^* + w_1 + v_1'\}, \{w_1, v_1'\}) \end{array} \right\}.$$

Define

$$\mathcal{R}[(v_0^*, w_0^*), \{A_1, A_1'\}] \coloneqq \{((A_0^*, A_1), B)\}_B \cup \{((A_0^*, A_1'), B)\}_B$$

where the indexing is over the four B such that

$$((A_0^*, A_1), B) \in \mathcal{R}[(v_0^*, w_0^*)] \text{ and } ((A_0^*, A_1'), B) \in \mathcal{R}[(v_0^*, w_0^*)].$$

Note that for any two $\{A_1, A_1'\} \neq \{\widetilde{A}_1, \widetilde{A}_1'\} \in \mathcal{A}_1^{\times 2}$, the sets $\mathcal{R}[(v_0^*, w_0^*), \{A_1, A_1'\}]$ and $\mathcal{R}[(v_0^*, w_0^*), \{\widetilde{A}_1, \widetilde{A}_1'\}]$ are disjoint, which can be seen by noting that B_1 always includes one vector from A_1 and one from A_1'.

Next, we claim that

$$\mathcal{R}[(v_0^*, w_0^*)] = \bigcup_{\{A_1, A_1'\} \in \mathcal{A}_1^{\times 2}} \mathcal{R}[(v_0^*, w_0^*), \{A_1, A_1'\}],$$

which follows from a counting argument. First,

$$\left| \bigcup_{\{A_1, A_1'\} \in \mathcal{A}_1^{\times 2}} \mathcal{R}[(v_0^*, w_0^*), \{A_1, A_1'\}] \right| = 8 \cdot \binom{2^{n-1}}{2} = 2^{2n} - 2^{n+1}.$$

Then, counting $|\mathcal{R}[(v_0^*, w_0^*)]|$ directly, we can choose from any of the 2^{n-1} possible A_1, any $2^n - 2$ of the possible B_0, and then, given B_0, the two possible B_1 that intersect A_1. Thus,

$$\left|\mathcal{R}[(v_0^*, w_0^*)]\right| = 2^{n-1} \cdot (2^n - 2) \cdot 2 = 2^{2n} - 2^{n+1}.$$

This establishes that the sets

$$\left\{\mathcal{R}[(v_0^*, w_0^*), \{A_1, A_1'\}]\right\}_{\{A_1, A_1'\} \in \mathcal{A}_1^{\times 2}}$$

partition $\mathcal{R}[(v_0^*, w_0^*)]$ equally into sets of size 8. Thus,[26]

$$\Pr_{\{A_1, A_1'\} \leftarrow \mathcal{A}_1^{\times 2}} \left[\forall (A, B) \in \mathcal{R}[(v_0^*, w_0^*), \{A_1, A_1'\}], |\langle \phi_A^{v_0^*} | \phi_B^{v_0^*} \rangle| \geq 1 - \epsilon^6\right] \geq 1 - 32\epsilon^6,$$

which means that there exists some $A_1^* = \{v_1^*, w_1^*\}$ such that

[25]Note that $v_0^* + v_1 + v_1' \neq w_0^*$ since otherwise $w_1 = v_1 + (v_0^* + w_0^*) = v_1'$ and $w_1' = v_1 + (v_0^* + w_0^*) = v_1$ which would mean that $A_1 = A_1'$. Thus, for the first B listed, $((A_0^*, A_1), B) \in \mathcal{R}[(v_0^*, w_0^*)]$, and a similar argument holds for the rest of the B.

[26]Here, we show that there exists a large fraction of $\{A_1, A_1'\}$ such that *all* $(A, B) \in \mathcal{R}[(v_0^*, w_0^*), \{A_1, A_1'\}]$ are "good", meaning that $|\langle \phi_A^{v_{A,B}} | \phi_B^{v_{A,B}} \rangle| \geq 1 - \epsilon^6$. As we will see later, it would have sufficed to prove the slightly weaker claim that there exists a large fraction of $\{A_1, A_1'\}$ such that *at least* $5/8$ of the $(A, B) \in \mathcal{R}[(v_0^*, w_0^*), \{A_1, A_1'\}]$ are good. This is because for each such $\{A_1, A_1'\}$, we will just need a single B (rather that all four) such that $((A_0^*, A_1), B)$ *and* $((A_0^*, A_1'), B)$ are good.

$$\Pr_{A_1 \leftarrow \mathcal{A}_1 \setminus \{A_1^*\}} \left[\forall (A,B) \in \mathcal{R}[(v_0^*, w_0^*), \{A_1^*, A_1\}], |\langle \phi_A^{v_0^*} | \phi_B^{v_0^*} \rangle| \geq 1 - \epsilon^6 \right] \geq 1 - 32\epsilon^6 \geq \frac{7}{8},$$

which holds for all $\epsilon \leq 1/8$.

Let $\mathcal{A}_1^{v_0^*}$ be the set of A_1 such that

$$\forall (A,B) \in \mathcal{R}[(v_0^*, w_0^*), \{A_1^*, A_1\}], |\langle \phi_A^{v_0^*} | \phi_B^{v_0^*} \rangle| \geq 1 - \epsilon^6,$$

and note that $|\mathcal{A}_1^{v_0^*}| \geq \frac{7}{8}(2^{n-1} - 1) > \frac{3}{4}2^{n-1}$.

Now consider any $A_1 = \{v_1, w_1\} \in \mathcal{A}_1^{v_0^*}$, and note that for $B = (\{v_0^*, v_0^* + v_1^* + v_1\}, \{v_1^*, v_1\})$, we have that

$$((A_0^*, A_1^*), B), ((A_0^*, A_1), B) \in \mathcal{R}[(v_0^*, w_0^*), \{A_1^*, A_1\}].$$

Thus, we know that

$$|\langle \phi_{(A_0^*, A_1^*)}^{v_0^*} | \phi_B^{v_0^*} \rangle| \geq 1 - \epsilon^6, \text{ and } |\langle \phi_{(A_0^*, A_1)}^{v_0^*} | \phi_B^{v_0^*} \rangle| \geq 1 - \epsilon^6,$$

so by Fact B.7,

$$|\langle \phi_{(A_0^*, A_1)}^{v_0^*} | \phi_{(A_0^*, A_1^*)}^{v_0^*} \rangle| \geq (1 - \epsilon^6)^2 - \sqrt{2\epsilon^6} \geq 1 - 2\epsilon^3.$$

Then if we set $|\tau^{v_0^*}\rangle := |\phi_{(A_0^*, A_1^*)}^{v_0^*}\rangle$, we have that for all $A_1 \in \mathcal{A}_1^{v_0^*}$,

$$|\langle \phi_{(A_0^*, A_1)}^{v_0^*} | \tau^{v_0^*} \rangle| \geq 1 - 2\epsilon^3.$$

Finally, repeating the analysis for $\mathcal{R}[(w_0^*, v_0^*)]$, there exists a $|\tau^{w_0^*}\rangle$ and a set $\mathcal{A}_1^{w_0^*}$ of size at least $\frac{3}{4}2^{n-1}$ such that for all $A_1 \in \mathcal{A}_1^{w_0^*}$,

$$|\langle \phi_{(A_0^*, A_1)}^{w_0^*} | \tau^{w_0^*} \rangle| \geq 1 - 2\epsilon^3.$$

Thus, setting $\mathcal{A}_1^* := \mathcal{A}_1^{v_0^*} \cap \mathcal{A}_1^{w_0^*}$ (which has size $\geq 2^{n-2}$) completes the proof.

□

Finally, we can reach a contradiction by using the fact that for any fixed A_0^*, all of the Π_{A_1} such that $(A_0^*, A_1) \in \mathcal{A}$ are orthogonal, which follows from the definition of the Π_{A_1}.

Now, define the rank-two projector

$$\Pi^* := |v_0^*\rangle |\tau^{v_0^*}\rangle \langle \tau^{v_0^*}| \langle v_0^*| + |w_0^*\rangle |\tau^{w_0^*}\rangle \langle \tau^{w_0^*}| \langle w_0^*|.$$

By Claim B.5 and the assumption of the theorem, for each $A_1 \in \mathcal{A}_1^*$ we know that

$$\|\Pi^* |\psi_{(A_0^*, A_1)}\rangle\| \geq 1 - 2\epsilon^3 \text{ and } \|\Pi_{A_1} |\psi_{(A_0^*, A_1)}\rangle\| \geq \epsilon.$$

For each $A_1 \in \mathcal{A}_1^*$, define

$$|\psi_{A_1}^*\rangle := \frac{\Pi^* |\psi_{(A_0^*, A_1)}\rangle}{\|\Pi^* |\psi_{(A_0^*, A_1)}\rangle\|}.$$

Thus, since $|\langle\psi^*_{A_1}|\psi_{(A_0^*,A_1)}\rangle| \geq 1-2\epsilon^3$ and $\|\Pi_{A_1}|\phi_{(A_0^*,A_1)}\rangle\| \geq \epsilon$, by Fact B.7 (second part) it holds that

$$\|\Pi_{A_1}|\psi^*_{A_1}\rangle\| \geq \epsilon(1-2\epsilon^3) - 2\epsilon^{3/2} \geq \frac{\epsilon}{2},$$

which holds for all $\epsilon \leq 1/8$.

Consider the following algorithm, which will eventually select all $\{|\psi^*_{A_1}\rangle\}_{A_1\in\mathcal{A}_1^*}$.

1. Set $i = 1$.
2. Select an arbitrary (not yet selected) $|\psi^*_{A_1}\rangle$, and define $|\psi_i\rangle := |\psi^*_{A_1}\rangle$.
3. Select all (not yet selected) $|\psi^*_{A_1}\rangle$ such that $|\langle\psi^*_{A_1}|\psi_i\rangle| \geq 1-\epsilon^4$.
4. Set $i = i+1$ and go back to Step 2.

First, we claim that in each invocation of Step 3, we select at most $16/\epsilon^2$ vectors. To see this, note that for each $|\psi^*_{A_1}\rangle$ selected in Step 3 during the i'th loop of the procedure, $|\langle\psi^*_{A_1}|\psi_i\rangle| \geq 1-\epsilon^4$ and $\|\Pi_{A_1}|\psi^*_{A_1}\rangle\| \geq \epsilon/2$. Thus, by Fact B.7 (second part),

$$\|\Pi_{A_1}|\psi_i\rangle\| \geq \frac{\epsilon}{2}(1-\epsilon^4) - \sqrt{2}\epsilon^2 \geq \frac{\epsilon}{4},$$

which holds for all $\epsilon \leq 1/8$. Since the Π_{A_1} are all orthogonal, and $|\psi_i\rangle$ has a component of at least $\epsilon^2/16$ squared norm on each, we conclude that there can be at most $16/\epsilon^2$ such A_1.

Second, let I be the value of i when the procedure terminates. Note that the $\{|\psi_i\rangle\}_{i\in[I]}$ are all in the image of a two-dimensional subspace $\mathsf{Im}(\Pi^*)$, and for all $i \neq j$, $|\langle\psi_i|\psi_j\rangle| < 1-\epsilon^4$.

Now, we use a Welch bound.

Imported Theorem B.6 ([Wel74]). *Let $\{x_1,\ldots,x_I\}$ be unit vectors in $\mathbb{C}^d$, and define $c = \max_{i\neq j}|\langle x_i|x_j\rangle|$. Then for every $k \in \mathbb{N}$,*

$$c^{2k} \geq \frac{1}{I-1}\left(\frac{I}{\binom{k+d-1}{k}} - 1\right).$$

Setting $d = 2$ and $k = I/2 - 1$, we have that

$$\frac{1}{I-1} \leq (1-\epsilon^4)^{I-2} \leq e^{-\epsilon^4(I-2)} \implies \frac{1}{\epsilon^4} \geq \frac{I-2}{\ln(I-1)} \geq \sqrt{I} \implies I \leq \frac{1}{\epsilon^8}.$$

Putting these two facts together, we have that the size of $\mathcal{A}_1^*$ is at most $16/\epsilon^{10}$, meaning that

$$2^{n-2} \leq \frac{16}{\epsilon^{10}} \implies 2^n \leq \frac{64}{\epsilon^{10}},$$

and contradicting the fact that $n > 10\log(1/\epsilon) + 6$.

□

B.2 Useful facts

Fact B.7. *Let $|\phi_a\rangle, |\phi_b\rangle$ be complex unit vectors such that $|\langle\phi_a|\phi_b\rangle| \geq 1-\alpha$. Then the following hold.*

1. *If $|\phi_c\rangle$ is a complex unit vector such that $|\langle\phi_b|\phi_c\rangle| \geq \beta$, then $|\langle\phi_a|\phi_c\rangle| \geq \beta(1-\alpha) - \sqrt{2\alpha}$.*
2. *If Π is a projector such that $\|\Pi|\phi_b\rangle\| \geq \beta$, then $\|\Pi|\phi_a\rangle\| \geq \beta(1-\alpha) - \sqrt{2\alpha}$.*

Proof. To show the first part, write $|\phi_a\rangle = e^{i\theta}(1-\alpha)|\phi_b\rangle + \sqrt{2\alpha-\alpha^2}|\phi_b^\perp\rangle$ for some θ and $|\phi_b^\perp\rangle$ orthogonal to $|\phi_b\rangle$. Then

$$\begin{aligned} |\langle\phi_a|\phi_c\rangle| &= |e^{i\theta}(1-\alpha)\langle\phi_b|\phi_c\rangle + \sqrt{2\alpha-\alpha^2}\langle\phi_b^\perp|\phi_c\rangle| \\ &\geq |e^{i\theta}(1-\alpha)\langle\phi_b|\phi_c\rangle| - \sqrt{2\alpha-\alpha^2} \\ &\geq \beta(1-\alpha) - \sqrt{2\alpha}. \end{aligned}$$

To show the second part, define

$$|\phi_c\rangle := \frac{\Pi|\phi_b\rangle}{\|\Pi|\phi_b\rangle\|},$$

and note that

$$|\langle\phi_b|\phi_c\rangle| = \frac{\langle\phi_b|\Pi|\phi_b\rangle}{\|\Pi|\phi_b\rangle\|} = \frac{\|\Pi|\phi_b\rangle\|^2}{\|\Pi|\phi_b\rangle\|} = \|\Pi|\phi_b\rangle\| \geq \beta.$$

Thus,

$$\|\Pi|\phi_a\rangle\| \geq \||\phi_c\rangle\langle\phi_c||\phi_a\rangle\| = |\langle\phi_a|\phi_c\rangle| \geq \beta(1-\alpha) - \sqrt{2\alpha},$$

where the first inequality follows because $|\phi_c\rangle \in \mathsf{Im}(\Pi)$ and the second inequality follows from the first part.

□

Fact B.8. *Let*

$$u_1 := \begin{pmatrix} a_1 \\ a_2 \\ 0 \end{pmatrix}, u_2 := \begin{pmatrix} b_1 \\ 0 \\ b_2 \end{pmatrix}, u_3 := \begin{pmatrix} 0 \\ c_1 \\ c_2 \end{pmatrix}$$

be three unit vectors in $\mathbb{R}^3_{\geq 0}$. Then,

$$u_1 \cdot u_2 + u_1 \cdot u_3 + u_2 \cdot u_3 \leq \frac{3}{2}.$$

Moreover, for any $\delta \in [0, 1/2]$, if

$$u_1 \cdot u_2 + u_1 \cdot u_3 + u_2 \cdot u_3 \geq \frac{3}{2} - \frac{\delta^3}{2},$$

then

$$u_1 \cdot u_2 \geq \frac{1}{2} - \delta, \;\; u_1 \cdot u_3 \geq \frac{1}{2} - \delta, \;\; \textit{and} \;\; u_2 \cdot u_3 \geq \frac{1}{2} - \delta.$$

Proof. We begin with the first part of the claim. Let $v_1 := (a_1\, a_2\, b_1\, b_2\, c_1\, c_2)$ and $v_2 := (b_1\, c_1\, a_1\, c_2\, a_2\, b_2)$. Then,

$$u_1 \cdot u_2 + u_1 \cdot u_3 + u_2 \cdot u_3 = \frac{1}{2} v_1 \cdot v_2^\top \leq \frac{1}{2}(a_1^2 + a_2^2 + b_1^2 + b_2^2 + c_1^2 + c_2^2) = \frac{3}{2},$$

where the inequality is Cauchy-Schwartz.

Now, we prove the "moreover" part. This is trivial when $\delta = 1/2$, so suppose that $u_1 \cdot u_2 = 1/2 - \delta$ for some $\delta \in [0, 1/2)$. We will show that this implies that

$$u_1 \cdot u_2 + u_1 \cdot u_3 + u_2 \cdot u_3 \leq \frac{3}{2} - \frac{\delta^3}{2},$$

which, by symmetry, would complete the proof.

Define the value

$$m := \max_{\substack{u_1, u_2, u_3, \\ u_1 \cdot u_2 = 1/2 - \delta}} \left\{ u_1 \cdot u_3 + u_2 \cdot u_3 \right\},$$

and let $a_1 = \sqrt{1-x}, a_2 = \sqrt{x}, b_1 = \sqrt{1-y}$ and $b_2 = \sqrt{y}$ for some $x, y \in [0, 1)$. Then,

$$\begin{aligned} m &= \max_{x,y \in [0,1), \sqrt{1-x}\sqrt{1-y} = 1/2 - \delta} \left\{ \sqrt{x} c_1 + \sqrt{y} c_2 \right\} \\ &\leq \max_{x,y \in [0,1), \sqrt{1-x}\sqrt{1-y} = 1/2 - \delta} \left\{ \sqrt{x+y} \sqrt{c_1 + c_2} \right\} \\ &= \max_{x,y \in [0,1), \sqrt{1-x}\sqrt{1-y} = 1/2 - \delta} \left\{ \sqrt{x+y} \right\}, \end{aligned}$$

where the inequality is Cauchy-Schwartz.

Next, we solve for

$$y = 1 - \frac{(\frac{1}{2} - \delta)^2}{1-x},$$

and see that

$$\begin{aligned} m^2 &= \max_{x \in [0,1)} \left\{ x + 1 - \frac{(\frac{1}{2} - \delta)^2}{1-x} \right\} \\ &= \max_{x \in [0,1)} \left\{ 2 - \frac{2}{1-x} \left(\frac{(1-x)^2 + \left(\frac{1}{2} - \delta\right)^2}{2} \right) \right\} \\ &\leq 2 - 2\left(\frac{1}{2} - \delta\right) \\ &= 1 + 2\delta, \end{aligned}$$

where the inequality is AM-GM.

Thus, to complete the proof it suffices to show that

$$\frac{1}{2} - \delta + \sqrt{1 + 2\delta} \leq \frac{3}{2} - \frac{\delta^3}{2}.$$

If $\delta = 0$, then both sides are 1, so now assume that $\delta > 0$. Then

$$\begin{aligned}\frac{1}{2} - \delta + \sqrt{1+2\delta} \leq \frac{3}{2} - \frac{\delta^3}{2} &\iff \sqrt{1+2\delta} \leq 1 + \delta - \frac{\delta^3}{2} \\ &\iff 1 + 2\delta \leq 1 + 2\delta + \delta^2 - (1+\delta)\delta^3 + \frac{\delta^6}{4} \\ &\iff \delta + \delta^2 - \frac{\delta^4}{4} \leq 1,\end{aligned}$$

which is true for all $\delta \in (0, 1/2)$. □

C Remaining Proofs from Section 5.3

In this appendix, we prove Lemma 5.9, Lemma 5.10, and Lemma 5.11. We proceed via three steps.

1. Compile the information-theoretic protocol Π^{QV} from Section 5.2 into a 4-message quantum "commit-challenge-response" protocol Π^{CCR} with a classical verifier. This compilation is achieved via the use of Mahadev's measurement protocol [Mah22]. As argued in [Bar21], the resulting protocol satisfies a "computationally orthogonal projectors" property, which was first described by [ACGH20].

2. Apply parallel repetition to Π^{CCR} to obtain Π^{parl}, and observe that the parallel repetition theorem of [Bar21] implies that the analogues of Lemma 5.9, Lemma 5.10, and Lemma 5.11 hold in Π^{parl}.

3. Apply Fiat-Shamir to Π^{parl} to obtain the protocol Π^{CV} from Protocol 5, and observe that Measure and Re-program (Imported Theorem 3.10) implies that Lemma 5.9, Lemma 5.10, and Lemma 5.11 must also hold with respect to Π^{CV}.

Proof. (of Lemma 5.9, Lemma 5.10, and Lemma 5.11)

Step 1. We first describe the syntax of a generic commit-challenge-response protocol between a quantum prover P and a classical verifier V.

- Commit: $\mathsf{P}(1^\lambda)$ and $\mathsf{V}(1^\lambda; r)$ engage in a two-message commitment protocol, where r are the random coins used by V to generate the first message of the protocol, and the prover responds with a classical commitment string.
- Challenge: V samples a random bit $d \leftarrow \{0,1\}$ and sends it to P.
- Response: P computes a (classical) response z and sends it to V.
- Output: V receives z and decides to either accept and output $\top$ or reject and output $\bot$.

Consider any QPT adversarial prover P^*, and let $|\psi_{\lambda,r}^{\mathsf{P}^*}\rangle^{\mathcal{A},\mathcal{C}}$ be the (purified) state of the prover after interacting with $\mathsf{V}(1^\lambda; r)$ in the commit phase, where $\mathcal{C}$ holds the (classical) prover message output during this phase, and $\mathcal{A}$ holds its remaining state.

Commit-challenge-response protocol $\Pi^{\mathsf{CCR}} = (\mathsf{V}^{\mathsf{CCR}}_{\mathsf{Gen}}, \mathsf{P}^{\mathsf{CCR}}_{\mathsf{Com}}, \mathsf{P}^{\mathsf{CCR}}_{\mathsf{Prove}}, \mathsf{V}^{\mathsf{CCR}}_{\mathsf{Ver}})$

Parameters: Number of qubits $\ell = \ell(\lambda)$ in the prover's state.

- $\mathsf{V}^{\mathsf{CCR}}_{\mathsf{Gen}}(1^\lambda, Q) \to (\mathsf{pp}, \mathsf{sp})$: Sample $(h, S) \leftarrow \mathsf{V}^{\mathsf{QV}}_{\mathsf{Gen}}(1^\lambda, Q)$ and $\{(\mathsf{pk}_j, \mathsf{sk}_j) \leftarrow \mathsf{TCF.Gen}(1^\lambda, h_j)\}_{j\in[\ell]}$, and set
$$\mathsf{pp} \coloneqq \{\mathsf{pk}_j\}_{j\in[\ell]}, \quad \mathsf{sp} \coloneqq (h, S, \{\mathsf{sk}_j\}_{j\in[\ell]}).$$
- $\mathsf{P}^{\mathsf{CCR}}_{\mathsf{Com}}(1^\lambda, Q, x, \mathsf{pp}) \to (\mathcal{B}, \mathcal{Z}, y)$: Prepare the state $|\psi\rangle \leftarrow \mathsf{P}^{\mathsf{QV}}(1^\lambda, Q, x)$ on register $\mathcal{B} = (\mathcal{B}_1, \dots, \mathcal{B}_\ell)$, which we write as
$$|\psi\rangle \coloneqq \sum_{v\in\{0,1\}^\ell} \alpha_v |v\rangle^{\mathcal{B}},$$
and then for each $j \in [\ell]$, apply $\mathsf{TCF.Eval}[\mathsf{pk}_j](\mathcal{B}_j) \to (\mathcal{B}_j, \mathcal{Z}_j, \mathcal{Y}_j)$, resulting in the state
$$\sum_{v\in\{0,1\}^\ell} \alpha_v |v\rangle^{\mathcal{B}} |\psi_{\mathsf{pk}_1, v_1}\rangle^{\mathcal{Z}_1, \mathcal{Y}_1}, \dots, |\psi_{\mathsf{pk}_\ell, v_\ell}\rangle^{\mathcal{Z}_\ell, \mathcal{Y}_\ell}.$$
Finally, measure registers $\mathcal{Y}_1, \dots, \mathcal{Y}_\ell$ in the standard basis to obtain string $y \coloneqq \{y_j\}_{j\in[\ell]}$.
- The verifier samples a random bit $d \leftarrow \{0,1\}$, and sends d to the prover.
- $\mathsf{P}^{\mathsf{CCR}}_{\mathsf{Prove}}(\mathcal{B}, \mathcal{Z}, d) \to z$: If $d = 0$, the prover measures registers $\mathcal{B}, \mathcal{Z}$ in the standard basis to obtain $z \coloneqq \{b_j, z_j\}_{j\in[\ell]}$. If $d = 1$, the prover applies $J(\cdot)$ coherently to each register $\mathcal{Z}_j$ and then measures registers $\mathcal{B}, \mathcal{Z}$ in the Hadamard basis to obtain $z \coloneqq \{b_j, z_j\}_{j\in[\ell]}$.
- $\mathsf{V}^{\mathsf{CCR}}_{\mathsf{Ver}}(Q, x, \mathsf{sp}, y, d, z) \to \{\{q_t\}_{t\in[\lambda]}\} \cup \{\top, \bot\}$:
  - Parse $y \coloneqq \{y_j\}_{j\in[\ell]}$ and $z \coloneqq \{b_j, z_j\}_{j\in[\ell]}$.
  - If $d = 0$, for each $j \in [\ell]$ compute $\mathsf{TCF.Check}(\mathsf{pk}_j, b_j, z_j, y_j)$. If any are $\bot$, then output $\bot$, and otherwise output $\top$.
  - If $d = 1$, do the following for each $j \in [\ell]$.
    * If $h_j = 0$, compute $\mathsf{TCF.Invert}(0, \mathsf{sk}_j, y_j)$, abort and output $\bot$ if the output is $\bot$, and otherwise parse the output as (m_j, x_j).
    * If $h_j = 1$, compute $\mathsf{TCF.Invert}(1, \mathsf{sk}_j, y_j)$, abort and output $\bot$ if the output is $\bot$, and otherwise parse the output as $(0, x_{j,0}), (1, x_{j,1})$. Then, check $\mathsf{TCF.IsValid}(x_{j,0}, x_{j,1}, z_j)$ and abort and output $\bot$ if the result is $\bot$. Next, set $m_j \coloneqq b_j \oplus z_j \cdot (J(x_{j,0}) \oplus J(x_{j,1}))$.

    Then, let $m \coloneqq (m_1, \dots, m_\ell)$ and compute $\mathsf{V}^{\mathsf{QV}}_{\mathsf{Ver}}(Q, x, h, m)$. Output $\bot$ if the result is $\bot$, and otherwise output $\{q_t\}_{t\in[\lambda]} \coloneqq m[S]$.

Figure 10: A quantum "commit-challenge-response" protocol for verifying quantum partitioning circuits.

The remaining strategy of the prover can be described by family of unitaries $\left\{\mathsf{U}^{\mathsf{P}^*}_{\lambda,0}, \mathsf{U}^{\mathsf{P}^*}_{\lambda,1}\right\}_{\lambda\in\mathbb{N}}$, where $\mathsf{U}^{\mathsf{P}^*}_{\lambda,0}$ is applied to $|\psi^{\mathsf{P}^*}_{\lambda,r}\rangle$ on challenge 0 (followed by a measurement of z), and $\mathsf{U}^{\mathsf{P}^*}_{\lambda,1}$ is applied to $|\psi^{\mathsf{P}^*}_{\lambda,r}\rangle$ on challenge 1 (followed by a measurement of z).

Let $\mathsf{V}_{\lambda,r,0}$ denote the accept projector applied by the verifier to the prover messages when $d = 0$, and define $\mathsf{V}_{\lambda,r,1}$ analogously. Then define the following projectors on registers $(\mathcal{A}, \mathcal{C})$.
$$\Pi^{\mathsf{P}^*}_{\lambda,r,0} \coloneqq \mathsf{U}^{\mathsf{P}^*\,\dagger}_{\lambda,0} \mathsf{V}_{\lambda,r,0} \mathsf{U}^{\mathsf{P}^*}_{\lambda,0}, \quad \Pi^{\mathsf{P}^*}_{\lambda,r,1} \coloneqq \mathsf{U}^{\mathsf{P}^*\,\dagger}_{\lambda,1} \mathsf{V}_{\lambda,r,1} \mathsf{U}^{\mathsf{P}^*}_{\lambda,1}.$$

Definition C.1. *A* commit-challenge-response *protocol has* computationally orthogonal projectors *if for any QPT prover* $\{\mathsf{P}^*_\lambda\}_{\lambda\in\mathbb{N}}$,

$$\mathop{\mathbb{E}}_r\left[\langle\psi^{\mathsf{P}^*}_{\lambda,r}|\,\Pi^{\mathsf{P}^*}_{\lambda,r,0}\Pi^{\mathsf{P}^*}_{\lambda,r,1}\Pi^{\mathsf{P}^*}_{\lambda,r,0}\,|\psi^{\mathsf{P}^*}_{\lambda,r}\rangle\right] = \mathrm{negl}(\lambda).$$

Now, consider running protocol Π^{CCR} with some fixed circuit Q and input x, and suppose that P is a predicate such that $P(Q(\cdot))$ is pseudo-deterministic. We define the verifier acceptance predicates as follows.

- $\mathsf{V}_{\lambda,r,0}$ runs $\mathsf{V}^{\mathsf{CCR}}_{\mathsf{Ver}}$ on $d=0$.
- $\mathsf{V}_{\lambda,r,1}$ runs $\mathsf{V}^{\mathsf{CCR}}_{\mathsf{Ver}}$ on $d=1$ to obtain either $\bot$ or $\{q_t\}_{t\in[\lambda]}$. In the latter case, it outputs $\top$ if $\mathsf{Maj}\left(\{P(q_t)\}_{t\in[\lambda]}\right) = 1-P(Q(x))$ and $\bot$ otherwise.

Then, by [Bar21, Lemma 4.4], which uses the soundness of Π^{QV} (Imported Theorem 5.6) and the soundness of the measurement protocol ([Mah22]), we have the following claim.

Claim C.2. *For any* $\{\mathsf{P}^*_\lambda\}_{\lambda\in\mathbb{N}}$ *attacking* Π^{CCR} *(Protocol in Fig. 10), it holds that*

$$\mathop{\mathbb{E}}_r\left[\langle\psi^{\mathsf{P}^*}_{\lambda,r}|\,\Pi^{\mathsf{P}^*}_{\lambda,r,0}\Pi^{\mathsf{P}^*}_{\lambda,r,1}\Pi^{\mathsf{P}^*}_{\lambda,r,0}\,|\psi^{\mathsf{P}^*}_{\lambda,r}\rangle\right] = \mathrm{negl}(\lambda),$$

where the verifier acceptance predicates $\mathsf{V}_{\lambda,r,0},\mathsf{V}_{\lambda,r,1}$ *used to define* $\Pi^{\mathsf{P}^*}_{\lambda,r,0}$ *and* $\Pi^{\mathsf{P}^*}_{\lambda,r,1}$ *are as described above.*

Step 2. In this step, we will use the following imported theorem.

Imported Theorem C.3 ([Bar21], Theorem 3.1)**.** *Let* $\epsilon > 0$ *and* $0<\delta<1$ *be constants. Let* Π *be a commit-challenge-response protocol with computationally orthogonal projectors, and where the verifier's* $d=0$ *acceptance predicate is publicly computable given the verifier's first message. Let* Π^{parl} *be the* $\lambda^{1+\epsilon}$ *parallel repetition of* Π*, where the verifier's challenge string* T *is sampled as a uniformly random* $\lambda^{1+\epsilon}$ *bit string with Hamming weight* λ*. Then for any QPT adversarial prover* P^* *attacking* Π^{parl}*, the probability that the verifier accepts all rounds* i *such that* $T_i = 0$ *and* $\geq \delta\cdot\lambda$ *rounds* i *such that* $T_i=1$ *is* $\mathrm{negl}(\lambda)$*.*

Now, we define the protocol $\Pi^{\mathsf{parl}} = (\mathsf{V}^{\mathsf{parl}}_{\mathsf{Gen}},\mathsf{P}^{\mathsf{parl}}_{\mathsf{Com}},\mathsf{P}^{\mathsf{parl}}_{\mathsf{Prove}},\mathsf{V}^{\mathsf{parl}}_{\mathsf{Ver}})$ to be the λ^2 parallel repetition of Π^{CCR}, where the verifier's challenge string T is sampled as a uniformly random λ^2 bit string with Hamming weight λ. Then, we can prove the following lemmas about Π^{parl}.

Lemma C.4 (Π^{parl} analogue of Lemma 5.9)**.** *For any family* $\{Q_\lambda,P_\lambda\}_{\lambda\in\mathbb{N}}$ *such that* $\{P_\lambda\circ Q_\lambda\}_{\lambda\in\mathbb{N}}$ *is pseudo-deterministic, sequence of inputs* $\{x_\lambda\}_{\lambda\in\mathbb{N}}$*, and QPT adversary* $\{\mathsf{A}_\lambda\}_{\lambda\in\mathbb{N}}$*, it holds that*

$$\Pr\left[\begin{array}{c}\mathsf{V}^{\mathsf{parl}}_{\mathsf{Ver}}(Q,x,\mathsf{sp},y,T,z) = \{\{q_{i,t}\}_{t\in[\lambda]}\}_{i:T_i=1}\ \wedge\\ \mathsf{MM}_\lambda\left(\{\{P(q_{i,t})\}_{t\in[\lambda]}\}_{i:T_i=1}\right) = 1-P(Q(x))\end{array} : \begin{array}{r}(\mathsf{pp},\mathsf{sp})\leftarrow\mathsf{V}^{\mathsf{parl}}_{\mathsf{Gen}}(1^\lambda,Q)\\ y\leftarrow\mathsf{A}(\mathsf{pp})\\ T\leftarrow\{0,1\}^{\binom{\lambda^2}{\lambda}}\\ z\leftarrow\mathsf{A}(T)\end{array}\right] = \mathrm{negl}(\lambda),$$

where A *maintains an internal state, which we leave implicit above.*

Proof. We have to rule out a prover that makes the verifier of Π^{CCR} accept each of the $\lambda^2-\lambda$ rounds where $T_i=0$, and, for a majority of the rounds i where $T_i=1$, accepts and outputs $\{q_{i,t}\}_{t\in[\lambda]}$ such that $\mathsf{Maj}\left(\{P(q_{i,t})\}_{t\in[\lambda]}\right) = 1-P(Q(x))$. This is directly ruled out by Claim C.2 and Imported Theorem C.3 with $\epsilon=1$ and $\delta=1/2$. □

Lemma C.5 (Π^{parl} analogue of Lemma 5.10)**.** *For any family $\{Q_\lambda, P_\lambda\}_{\lambda\in\mathbb{N}}$ such that $\{P_\lambda \circ Q_\lambda\}_{\lambda\in\mathbb{N}}$ is pseudo-deterministic, sequence of inputs $\{x_\lambda\}_{\lambda\in\mathbb{N}}$, and QPT adversary $\{\mathsf{A}_\lambda\}_{\lambda\in\mathbb{N}}$, it holds that*

$$\Pr\left[\begin{array}{l} \mathsf{V}^{\mathsf{parl}}_{\mathsf{Ver}}(Q,x,\mathsf{sp},y,T,z) \neq \bot \ \wedge \\ w \notin D_{\mathsf{in}}[P,P(Q(x))] \end{array} : \begin{array}{r} (\mathsf{pp},\mathsf{sp}) \leftarrow \mathsf{V}^{\mathsf{parl}}_{\mathsf{Gen}}(1^\lambda,Q) \\ y \leftarrow \mathsf{A}(\mathsf{pp}) \\ T \leftarrow \{0,1\}^{\binom{\lambda^2}{\lambda}} \\ z \leftarrow \mathsf{A}(T) \\ w := \mathsf{TestRoundOutputs}[\mathsf{sp}](y,T,z) \end{array}\right] = \mathrm{negl}(\lambda),$$

where A *maintains an internal state, which we leave implicit above, and where* $\mathsf{TestRoundOutputs}$ *is defined as in Section 5.3, except that string T is explicitly given rather than being computed by a random oracle H.*

Proof. First, we make the following observation. For every $i \in [\lambda^2]$, the strings $\{q_{i,t}\}_{t\in[\lambda]}$ that the verifier would output conditioned on accepting and on $T_i = 1$ are already determined by the prover's first message $y_i := (y_{i,1}, \cdots, y_{i,\ell})$ and the secret parameters sp. Indeed, recall from the description of Π^{QV} that the bits in $\{q_{i,t}\}_{t\in[\lambda]}$ are computed from indices $j \in [\ell]$ where the basis $h_{i,j} = 0$ (that is, they are the result of standard basis measurements). Moreover, when $h_{i,j} = 0$, $\mathsf{pk}_{i,j}$ defines an injective function, which follows from Definition 3.6, correctness properties (a) and (c). Thus, each string $y_{i,j}$ either has one or zero pre-images. If it has zero, the verifier would never accept when $T_i = 1$, and if it has one, the verifier would only accept the first bit $b_{i,j}$ of the pre-image.

So, we can define $\{\{q_{i,t}\}_{t\in[\lambda]}\}_{i\in[\lambda^2]}$ based on the prover's first message $\{y_i\}_{i\in[\lambda^2]}$. Then,

- Let a be the fraction of $\{q_{i,t}\}_{t\in[\lambda]}$ such that $\mathsf{Maj}\left(\{P(q_{i,t})\}_{t\in[\lambda]}\right) = P(Q(x))$ over $i \in [\lambda^2]$.
- Let b be the fraction of $\{q_{i,t}\}_{t\in[\lambda]}$ such that $\mathsf{Maj}\left(\{P(q_{i,t})\}_{t\in[\lambda]}\right) = P(Q(x))$ over $i : T_i = 1$.

By the definition of $D_{\mathsf{in}}[P, P(Q(x))]$,

$$w \notin D_{\mathsf{in}}[P,P(Q(x))] \implies a \leq \frac{3}{4} + \frac{1}{\lambda}.$$

Moreover, by Claim C.2 and Imported Theorem C.3 with $\epsilon = 1$ and $\delta = 1/5$,

$$\Pr\left[\mathsf{V}^{\mathsf{parl}}_{\mathsf{Ver}}(Q,x,\mathsf{sp},y,T,z) \neq \bot \wedge b < \frac{4}{5}\right] = \mathrm{negl}(\lambda).$$

Thus, the proof is completed by showing that

$$\Pr\left[b - a \geq \frac{4}{5} - \left(\frac{3}{4} + \frac{1}{\lambda}\right) > \frac{1}{30}\right] \leq e^{-2(\lambda/30)^2} = \mathrm{negl}(\lambda),$$

where the expression inside the probability holds for large enough λ, and the inequality is Hoeffding's inequality (using the case where the random variables are sampled without replacement).

□

Lemma C.6 (Π^{parl} analogue of Lemma 5.11)**.** *For any family $\{Q_\lambda, P_\lambda\}_{\lambda\in\mathbb{N}}$ such that $\{P_\lambda \circ Q_\lambda\}_{\lambda\in\mathbb{N}}$ is pseudo-deterministic, sequence of inputs $\{x_\lambda\}_{\lambda\in\mathbb{N}}$, and QPT adversary $\{\mathsf{A}_\lambda\}_{\lambda\in\mathbb{N}}$, it holds that*

$$\Pr\left[\begin{array}{l} \mathsf{V}^{\mathsf{parl}}_{\mathsf{Ver}}(Q,x,\mathsf{sp},y,T,z) = \{\{q_{i,t}\}_{t\in[\lambda]}\}_{i:T_i=1} \ \wedge \\ \mathsf{MM}_\lambda\left(\{\{P(q_{i,t})\}_{t\in[\lambda]}\}_{i:T_i=1}\right) = 1 - P(Q(x)) \ \wedge \\ w \notin D_{\mathsf{out}}[P, P(Q(x))] \end{array} : \begin{array}{r} (\mathsf{pp},\mathsf{sp}) \leftarrow \mathsf{V}^{\mathsf{parl}}_{\mathsf{Gen}}(1^\lambda, Q) \\ y \leftarrow \mathsf{A}(\mathsf{pp},\mathsf{sp}) \\ T \leftarrow \{0,1\}^{\binom{\lambda^2}{\lambda}} \\ z \leftarrow \mathsf{A}(T) \\ w := \mathsf{TestRoundOutputs}[\mathsf{sp}](y,T,z) \end{array}\right] = \mathrm{negl}(\lambda),$$

where A *maintains an internal state, which we leave implicit above, and where* $\mathsf{TestRoundOutputs}$ *is defined as in Section 5.3, except that string T is explicitly given rather than being computed by a random oracle H.*

Proof. We again define $\{\{q_{i,t}\}_{t\in[\lambda]}\}_{i\in[\lambda^2]}$ based on the prover's first message $\{y_i\}_{i\in[\lambda^2]}$, and

- Let a be the fraction of $\{q_{i,t}\}_{t\in[\lambda]}$ such that $\mathsf{Maj}\left(\{P(q_{i,t})\}_{t\in[\lambda]}\right) = 1 - P(Q(x))$ over $i \in [\lambda^2]$.
- Let b be the fraction of $\{q_{i,t}\}_{t\in[\lambda]}$ such that $\mathsf{Maj}\left(\{P(q_{i,t})\}_{t\in[\lambda]}\right) = 1 - P(Q(x))$ over $i : T_i = 1$.

By the definition of $D_{\mathsf{out}}[P, P(Q(x))]$,

$$w \notin D_{\mathsf{out}}[P, P(Q(x))] \implies a \le \frac{1}{3} + \frac{1}{\lambda}.$$

Thus, the proof is completed by showing that

$$\Pr\left[b - a \ge \frac{1}{2} - \left(\frac{1}{3} + \frac{1}{\lambda}\right) > \frac{1}{10}\right] \le e^{-2(\lambda/10)^2} = \mathrm{negl}(\lambda),$$

which again follows from Hoeffding's inequality. Note that this argument is entirely statistical, and holds even if A_λ has sp. □

Step 3. Note that the protocol Π^{CV} is exactly Fiat-Shamir applied to Π^{parl}. That is, take Π^{parl} and let the verifier's challenge T be computed by applying a random oracle H to the prover's first message y. This results in exactly the protocol Π^{CV}, where we have re-defined the prover operations $(\mathsf{P}^{\mathsf{parl}}_{\mathsf{Com}}, \mathsf{P}^{\mathsf{parl}}_{\mathsf{Prove}})$ as $(\mathsf{P}^{\mathsf{CV}}_{\mathsf{Prep}}, \mathsf{P}^{\mathsf{CV}}_{\mathsf{Prove}}, \mathsf{P}^{\mathsf{CV}}_{\mathsf{Meas}})$. Then, straightforward applications of Measure-and-Reprogram (Imported Theorem 3.10) show that Lemma C.4, Lemma C.5, and Lemma C.6 imply Lemma 5.9, Lemma 5.10, and Lemma 5.11 respectively.

In more detail, suppose that Lemma 5.9 is false, and fix P, Q, x, and an adversary A that breaks that claim. Define a predicate V that takes as input y, $H(y)$, the rest of the transcript of the protocol, and the verifier's secret parameters sp, and outputs whether

$$\mathsf{V}^{\mathsf{CV}}_{\mathsf{Ver}}(Q,x,\mathsf{sp},\pi) = \{\{q_{i,t}\}_{t\in[\lambda]}\}_{i:T_i=1} \ \wedge \ \mathsf{MM}_\lambda(\{\{P(q_{i,t})\}_{t\in[\lambda]}\}_{i:T_i=1}) = 1 - P(Q(x)).$$

Define adversary B^H to run an interaction between A and the verifier V^{CV}, forwarding random oracles calls to an external oracle H, and output y along with auxiliary information aux that includes the rest of the transcript and sp. Then we have that

$$\Pr\left[V(y, H(y), \mathsf{aux}) = 1 : (y, \mathsf{aux}) \leftarrow \mathsf{B}^H\right] = \mathsf{non\text{-}negl}(\lambda).$$

Since B makes $\mathrm{poly}(\lambda)$ queries to H, Imported Theorem 3.10 implies that there exists a simulator Sim such that

$$\Pr\left[V(y, T, \mathsf{aux}) = 1 : \begin{array}{r} (y, \mathsf{state}) \leftarrow \mathsf{Sim}[\mathsf{B}] \\ T \leftarrow \{0,1\}^{\binom{\lambda^2}{\lambda}} \\ \mathsf{aux} \leftarrow \mathsf{Sim}[\mathsf{B}](T, \mathsf{state}) \end{array}\right] = \mathsf{non\text{-}negl}(\lambda).$$

Moreover, by definition (Imported Theorem 3.10), $\mathsf{Sim}[\mathsf{B}]$ runs B honestly except that it simulates H and measures one of B's queries to H. Thus, $\mathsf{Sim}[\mathsf{B}]$ can be used as an adversarial prover interacting in Π^{parl}, where y is sent to the verifier as the prover's first message, and T is sampled and given in response. Thus, $\mathsf{Sim}[\mathsf{B}]$ can be used to violate Lemma C.4.

Finally, the fact that Lemma C.5 implies Lemma 5.10 and Lemma C.6 implies Lemma 5.11 can be shown in exactly the same way, by defining the appropriate predicate V. This completes the proof.

□