Obfuscation of Quantum Computation

James Bartusek

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2024-169
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2024/EECS-2024-169.html

August 9, 2024

Obfuscation of Quantum Computation

by

James Bartusek

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Sanjam Garg, Chair
Professor Dakshita Khurana
Professor Umesh Vazirani
Professor John Wright

Summer 2024

Obfuscation of Quantum Computation

Abstract

Obfuscation of Quantum Computation

by

James Bartusek

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Sanjam Garg, Chair

A program obfuscator is a compiler that renders code unintelligible without harming its functionality. The ability to obfuscate computation is an immensely powerful cryptographic tool and, as such, developing techniques for program obfuscation is one of the central goals of modern cryptography.

This thesis presents the first known methods for obfuscating useful classes of *quantum* computations. We propose several obfuscation schemes and prove their security in the classical oracle model, obtaining the following results.

- **Obfuscation for null quantum circuits.** This yields several novel applications, including witness encryption for QMA and succinct non-interactive zero-knowledge arguments for QMA.
- **Obfuscation for pseudo-deterministic quantum circuits.** We obfuscate any quantum circuit that takes a classical input and computes a (nearly) deterministic classical output. Thus, we obtain the first scheme that is powerful enough to obfuscate Shor's algorithm.
- **Quantum state obfuscation.** A quantum state obfuscator supports obfuscation of (pseudo-deterministic) quantum circuits with *auxiliary quantum input*. This result gives the first candidate "best-possible" copy-protection scheme for general-purpose software.

Along the way, we present the first constructions of several quantum cryptographic primitives, including publicly-decodable X-measurable commitments, publicly-verifiable quantum fully-homomorphic encryption, and publicly-verifiable linearly-homomorphic authentication of quantum data.

Contents

Acknowledgments

First and foremost, I would like to thank my advisor Sanjam Garg for cultivating a wonderful research environment at Berkeley, and for making all of this work possible. His energy and enthusiasm are unmatched, and his example has been absolutely vital to my graduate school experience. Even if we weren't always working in the same area, I would often catch myself wondering - how would Sanjam approach this problem?

I am also grateful to Dakshita Khurana for her indispensable support, advice, and collaboration throughout my time as a graduate student. She has been a constant source of inspiration, and I feel that I come away from every meeting with either progress or excitement for the next idea.

Thank you to Mark Zhandry and Fermi Ma for introducing me to the world of research during my time as a master's student at Princeton, and stimulating a lifelong fascination with the subject of cryptography. I cannot imagine a better environment in which to take the first steps as a researcher.

I am lucky to have spent five years at Berkeley, surrounded by the most inspiring peers. Thank you to Akshayaram Srinivasan and Giulio Malavolta for showing me the ropes during my first year, and for continued collaboration and friendship. Thank you to everyone in the theory group who made my Berkeley experience exceptional, including but not limited to Orr, Seri, Chinmay, Nick, Prashant, Sam, Thiago, Bhaskar, and Vamsi.

I am grateful to have been hosted during several occasions over the course of my PhD. Thank you to Zvika Brakerski, Vinod Vaikuntanathan, Amit Sahai, Ryo Nishimaki, Dakshita Khurana, Justin Holmgren, and Pratyay Mukherjee for these visits, which were an invaluable part of my graduate school experience. I have also had the pleasure of collaborating with many other outstanding researchers during the last five years: Amit Agarwal, Andrea Coladangelo, Vipul Goyal, Abhishek Jain, Yael Kalai, Fuyuki Kitagawa, Nishant Kumar, Alex Lombardi, Daniel Masny, Saachi Mutreja, Alexander Poremba, Justin Raizes, Thomas Vidick, Michael Walter, Takashi Yamakawa, Lisa Yang, and Yinuo Zhang.

Finally, I would like to thank my parents and my brothers for their love, support, and grounding, and Sachi for brightening every day.

1 Introduction

Modern cryptography has had tremendous success developing methods to safeguard information, achieving tasks that are foundational to internet security such as public-key encryption and digital signatures [DH76, RSA78], as well as more versatile tasks such as secure computation [Yao82] and zero-knowledge proofs [GMR85]. But since the birth of the field, researchers have wondered whether it is possible to take these results a step further [DH76]: Is it feasible to encrypt not just bits of information, but *computer programs themselves*? Of course, one could simply encrypt the code of the computer program. However, this renders the program useless to anyone without the secret key to the encryption scheme. Instead, the goal is to compile the program P into a new program $\widetilde{P}$ such that $\widetilde{P}$ has the same functionality as P, but reveals as little as possible about the implementation details of P. This, intuitively, is the notion of *program obfuscation*.

Finding provably secure methods for program obfuscation has been one of the central questions driving research in cryptography over the past couple of decades. On the one hand, it is now understood that obfuscation is an extraordinarily powerful tool, not only for protecting software against piracy and intellectual property theft, but also for seemingly unrelated cryptographic tasks such as succinct arguments [SW14], non-interactive multi-party key exchange [BZ14], and fully-homomorphic encryption [CLTV15], just to name a few. In fact, in 2014 [SW14], program obfuscation was dubbed a "central hub" of cryptography due to its myriad applications, and it has certainly lived up to that billing. On the other hand, secure program obfuscation has been notoriously difficult to achieve. Despite being introduced informally in 1976 [DH76] and given a formal treatment in 2001 [BGI+01], it was not until 2013 [GGH+13] that the first plausibly secure candidate for obfuscating classical computation was proposed, and not until 2021 [JLS21] that researchers developed the first classical obfuscation scheme proven secure under well-understood mathematical conjectures.

While there has certainly been significant progress over the past decade, the feasibility and scope of program obfuscation remain poorly understood in the big picture, and several open questions are yet unanswered. The goal of this thesis is to take the first steps towards addressing one of these open problems, and expand the boundary encompassing the class of computations that are known to be obfuscatable.

Quantum computation. Quantum information processing is a powerful paradigm in computer science that leverages quantum mechanical properties to help simulate physical systems, secure information, and speed up computations, among other applications. The past several years have witnessed increased investment in quantum computing technology, and continued progress towards building general-purpose quantum computers and quantum communication networks.

In anticipation of full-scale quantum computing and networking, researchers have begun to investigate basic questions pertaining to the privacy and integrity of quantum information, resulting in a remarkable series of feasibility results. For example, we

now know how to encrypt [AMTDW00] and authenticate [BCG+02] quantum information, prove quantum statements in zero-knowledge [BJSW16], perform secure multi-party quantum computations [CGS02, DNS10, DGH+20], and delegate quantum computations privately and verifiably [Chi05, ABOEM18, BFK09, RUV13, Mah18a, Mah18b].

Motivating questions. However, despite the efforts and results listed above, the feasibility of *obfuscating* quantum computation has remained elusive. Since obfuscation is such a central primitive in the study of classical cryptography, this begs the following question, which has remained largely open.

Is it possible to obfuscate quantum computation?

This thesis will take the first significant steps towards answering this question in the affirmative. In light of recent breakthrough results establishing the feasibility of "classical control" over quantum systems via the use of cryptography, we seek to make progress by "reducing" obfuscation of quantum computations to obfuscation of classical computation, and then appeal to results from the long line of literature on classical obfuscation. Given the apparent difficulty of and lack of prior progress on this question, we operate in an idealized setting called the "classical oracle model", where black-box obfuscation of efficient classical computation comes for free.[1] Establishing the feasibility of quantum obfuscation in this model provides the first concrete evidence that quantum obfuscation is achievable, and hence, we ask the following question.

What class of quantum computations can be obfuscated in the classical oracle model?

Progress on this question begs a deeper understanding of the *usefulness* of quantum obfuscation, perhaps even beyond straightforward applications like protecting the intellectual property in novel quantum algorithms. Thus, a secondary focus of this thesis is to make progress on the following questions.

What are the applications quantum obfuscation? Should it be considered a "central hub" of quantum cryptography?

1.1 Results

We consider three flavors of obfuscation for (various classes of) quantum computation, and show that each is achievable in the classical oracle model (under additional computational assumptions). We introduce these notions and give informal theorem statements below, directing the reader to Section 2.2.1 for formal definitions, and Section 3.1, Section 4.4.1, and Section 5.4 for formal theorem statements.

[1] See Section 1.1 for more discussion on the classical oracle model. Informally, in the classical oracle model, the obfuscated program as well as the (possibly adversarial) evaluator are granted oracle access to an efficiently computable classical function sampled by the obfuscator.

First, we construct *obfuscation for null quantum circuits*, extending the powerful notion of "null-iO" [WZ17] from the classical to the quantum setting. Informally, an obfuscator for null quantum circuits takes as input the (classical) description of a quantum circuit Q where Q produces a single classical bit of output, and outputs the (classical) description of an obfuscated circuit $\widetilde{Q}$. For correctness, we require that on any (potentially quantum) input $|\psi\rangle$ such that $Q(|\psi\rangle) = b$ for some bit b,[2] it holds that $\widetilde{Q}(|\psi\rangle) = b$.[3] For security, we require that for any two *null* quantum circuits Q_0, Q_1, meaning that for all inputs $|\psi\rangle$, $Q_0(|\psi\rangle) = Q_1(|\psi\rangle) = 0$, it holds that the obfuscations of Q_0 and Q_1 are indistinguishable (to any quantum polynomial-time adversary). We show that this notion is achievable under a standard post-quantum hardness assumption in the classical oracle model.

Theorem 1.1 (Informal, [BM22])**.** *Assuming the quantum hardness of learning with errors (LWE), there exists obfuscation for null quantum circuits in the classical oracle model.*

Much like the classical setting, obfuscation for null quantum circuits has several powerful cryptographic applications, including quantum analogues of witness encryption and non-interactive zero-knowledge (see Section 1.2 for more details). However, the fact that security only holds for quantum circuits that always output 0 severely limits its ability to protect more general-purpose quantum software.

To address this, we dramatically expand the class of quantum computations that can be obfuscated by constructing an obfuscation scheme for all "pseudo-deterministic" quantum circuits. A pseudo-deterministic circuit Q takes classical inputs and produces classical outputs, and satisfies the property that for each input x, there exists an output y such that $\Pr[Q(x) = y]$ is negligibly close to 1. That is, Q implements a (nearly) deterministic map from classical inputs to classical outputs. One prominent example of such a pseudo-deterministic circuit is the circuit implementing Shor's factoring algorithm.

Our next result establishes that it is possible to obfuscate these circuits in the classical oracle model, but where the description of the obfuscated circuit includes a *quantum* state (as we discuss later, we leave it open to improve the result to an obfuscated circuit with classical description). We note that while our obfuscation scheme for null quantum circuits follows without too much difficulty given prior techniques, expanding the class to all pseudo-deterministic circuits is significantly more technically involved, as we begin to discuss in Section 1.3.

Theorem 1.2 (Informal, [BKNY23])**.** *Assuming the quantum hardness of LWE, there exists obfuscation for pseudo-deterministic quantum circuits in the classical oracle model, where the obfuscated program is a quantum state.*

[2] In general, the output of Q might be some distribution over $\{0, 1\}$, but here, we restrict our attention to inputs that produce a *deterministic* outcome. In fact, we can slightly relax this requirement to inputs that produce a "pseudo-deterministic" outcome, meaning that there is some bit b such that $\Pr[Q(|\psi\rangle) = b]$ is negligibly close to 1.

[3] Technically, we weaken the correctness requirement to allow $\widetilde{Q}$ access to *multiple copies* of the quantum input $|\psi\rangle$.

Finally, we consider a notion of obfuscation introduced in [CG24] called *quantum state obfuscation*. Here, the program to be obfuscated consists of a (classically-described) quantum circuit Q along with an *auxiliary quantum input* $|\psi\rangle$. To evaluate the program on some classical input x, the circuit is run on both $|\psi\rangle$ and x in order to produce a classical output y. Again, we restrict our attention to *pseudo-deterministic* quantum programs, where for each input x, there is a string y such $\Pr[Q(x, |\psi\rangle) = y]$ is negligibly close to 1. As observed in [CG24] and discussed further in Section 1.2, obfuscation for this class of programs enjoys a deep connection to the powerful notion of *quantum copy-protection* [Aar09], one of the holy grails of quantum cryptography.

Theorem 1.3 (Informal, [BBV24])**.** *Assuming quantum-hard one-way functions, there exists obfuscation for pseudo-deterministic quantum programs with auxiliary quantum input in the classical oracle model.*

Note that this result, as stated, strictly improves upon Theorem 1.2. We weaken the assumption from LWE to one-way functions, and generalize the class of obfuscatable programs to include those with auxiliary quantum input. However, as we discuss further in Section 1.3, the techniques involved in each construction are quite different and independently interesting. For example, one direction for future work is to instantiate the building blocks used in the [BKNY23] construction with classical rather than quantum communication, which would then yield a classical obfuscated program. We believe this to be plausible, and consider it more difficult (though perhaps not impossible) to solve the analogous question of "de-quantizing" the [BBV24] approach.

On the classical oracle model. Before proceeding further, we discuss the classical oracle model in more detail. In this model, the obfuscation algorithm may output an efficient deterministic classical functionality F, and we include *oracle access* to F as part of the obfuscated program. That is, both the evaluator of the obfuscated program and the adversary are allowed to make queries to F (even in quantum superposition), but are not allowed to inspect the internal details of the implementation of F.

As mentioned earlier, a result in this model can be interpreted as reducing the problem of obfuscating quantum functionalities to classical functionalities. Indeed, instantiating the oracle using a candidate quantum-secure indistinguishability obfuscation (iO) scheme for classical computation (e.g. [BGMZ18, CVW18, BDGM22, GP21, WW21]) gives a heuristically secure construction in the plain model.[4] Furthermore, by the "best-possible" security guarantee of iO [GR07], if there exists a secure implementation of the oracle in the plain model, iO is one such.

[4] An analogy can be drawn here to other common idealized models utilized in cryptography. Indeed, 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 [Sho97] idealizes a cryptographic group. Often a scheme is proven secure in the random oracle or generic group model, but when used in practice, the random oracle is heuristically instantiated with a cryptographic hash function or the generic group is heuristically instantiated with a cryptographic group. Here, we consider heuristically instantiating the classical oracle with a classical obfuscation scheme.

Work	Obfuscator input	Obfuscator output	Program input	Program output	Program class	Assumption/ model	Result
[BK21]	Classical	Quantum°	Quantum	Quantum	Unitaries w/ logarithmically many non-Clifford gates	iO for classical circuits	iO
[BM22]*	Classical	Classical	Quantum°	Classical	Null circuits	Classical oracle model + LWE	iO
[BKNY23]*	Classical	Quantum	Classical	Classical	(Pseudo)-Deterministic circuits	Classical oracle model + LWE	Ideal
[CG24]	Quantum	Quantum	Classical	Classical	Deterministic circuits	Quantum oracle model	iO
[BBV24]*	Quantum	Quantum	Classical	Classical	(Pseudo)-Deterministic circuits	Classical oracle model + OWF	Ideal

Table 1: * Indicates work included in this thesis. A couple of additional notes about the schemes: In [BK21], the obfuscator outputs a quantum state that can only be used to evaluate the program *on one input*, and then is potentially destroyed. In [BM22], the obfuscated program can be run on quantum inputs, but requires *multiple copies* of the quantum input. The last column refers to the definition of obfuscation that is achieved in each work, where iO stands for indistinguishability obfuscation. Finally, we note that while achieving the notion of ideal obfuscation is only possible in the oracle model, the results in the *classical* oracle model yield heuristic candidates for iO in the plain model.

Moreover, results in the classical oracle model have historically inspired research that showed analogous results without the aid of an oracle. For example, quantum money [AC12] and signature tokens [BS16] were first achieved in the classical oracle model before being de-oracle-ized [Zha21, CLLZ21], and copy-protection for unlearnable programs was first achieved in a *quantum* oracle model [Aar09] before it was achieved in the classical oracle model [ALL+21] and later without oracles, for certain classes of functionalities (e.g. [CLLZ21, CMP22, LLQZ22, CG24]).

Related work. Alagic and Fefferman [AF16] presented definitions for obfuscating quantum circuits (and obfuscating classical circuits using quantum states), though without any positive constructive results. Alagic, Brakerski, Dulek and Schaffner [ABDS21] presented a negative result, establishing that virtual black-box (VBB) obfuscation of *classical circuits*, even with the aid of quantum information, is impossible.

Broadbent and Kazmi [BK21] showed how to obfuscate quantum circuits that have only a few non-Clifford gates. In their construction, the size of the obfuscated circuit blows up exponentially with the number of non-Clifford gates, thus achieving a result that, in some sense, goes barely beyond obfuscation for classical circuits. Finally, Coladangelo and Gunn [CG24], define the notion of quantum state (indistinguishability) obfuscation, show applications of this notion to software copy-protection, and construct a quantum state indistinguishability obfuscator in the *quantum oracle model*. We summarize these results, along with the contributions in this thesis, in Table 1.

1.2 Applications

Now, we discuss in more detail several applications that follow from our quantum obfuscation schemes.

Advanced cryptographic primitives. First, our obfuscators yield the first candidate[5] constructions of a number of quantum cryptographic primitives of interest. We list a few here. Further applications are discussed in [BM22].

- **Witness encryption for QMA.** Informally, a witness encryption scheme [GGSW13] supports the ability to encrypt a message with respect to some puzzle such that the message can be decrypted only given a solution to the puzzle. Our obfuscation scheme for null quantum circuits implies the ability to encrypt messages with respect to puzzles with *quantum* solutions. A bit more formally, we obtain the first candidate witness encryption for QMA (quantum Merlin-Arthur, the quantum analogue of NP), where one can encrypt a message with respect to an instance x of any language in QMA.

- **Non-interactive zero-knowledge for QMA.** In turn, witness encryption for QMA (plus iO for classical circuits) implies a one-message protocol for proving the validity of any statement in QMA, without revealing any additional information about the witness. This gives the first candidate construction of non-interactive zero-knowledge for QMA (in the common reference string model).

- **Succinct non-interactive arguments for QMA.** In fact, the zero-knowledge protocol we obtain has a *succinct* proof, meaning that its size does not grow with the length of the witness. Thus, we obtain the first candidate construction of a SNARG (succinct non-interactive argument) for QMA, the quantum analogue of an extremely useful and versatile primitive from classical cryptography.

- **Attribute-based encryption for quantum predicates.** Attribute-based encryption is a "fine-grained" notion of encryption, where messages are encrypted with respect to some (traditionally, classical) predicate P, and keys are sampled with respect to some "attribute" x. A ciphertext for predicate P should only be able to be decrypted by a key for x if $P(x) = 1$. We extend this notion to the quantum setting, allowing the predicate to be computable by any pseudo-deterministic quantum circuit, and show that such an encryption scheme follows from witness encryption for QMA (plus iO for classical circuits).

- **Functional encryption for quantum predicates.** Finally, we consider functional encryption, which is the most general form of fine-grained encryption. In a functional

[5] We specify that these schemes are "candidates" because they all rely on the security of our quantum obfuscators, which are only proven secure in the classical oracle model, and thus only yield plain model schemes with heuristic security.

encryption scheme, keys are sampled with respect to some (traditionally, classical) functionality f, and, given a ciphertext encrypting message m and key for function f, it should only be possible to recover the value $f(m)$. We present the first candidate construction of functional encryption where f may be any function computable by a pseudo-deterministic quantum circuit. To achieve this, we actually require full-fledged obfuscation of pseudo-deterministic quantum circuits (as opposed to null quantum circuits, which sufficed for the previous applications).

Quantum software protection. Next, we discuss another natural class of applications for quantum obfuscation, namely, quantum software protection. Suppose a company develops a novel quantum algorithm, but wants to keep their algorithmic insights private. A natural idea is to *obfuscate* the algorithm before releasing or selling it to the general public. Our result establishing the feasibility of obfuscating *any pseudo-deterministic quantum circuit* constitutes the first evidence that such quantum software protection exists. Indeed, perhaps the most famous quantum algorithm to date, Shor's algorithm, fits into the class of pseudo-deterministic quantum computations.

In the classical setting, obfuscation has also been identified as a useful tool for digital watermarking [BGI+01, CHN+18], which allows for embedding an unremovable "mark" into a program, and acts as a deterrent against software piracy. In fact, 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]. Since its introduction, this notion of "copy-protection" has been a prominent subject of research within quantum cryptography. However, positive progress on copy-protecting general-purpose software has been difficult to obtain.

A mentioned earlier, the notion of *quantum state obfuscation* was recently conceptualized and defined by [CG24] precisely for the purpose of obtaining better software copy-protection schemes. Indeed, they demonstrate how a quantum state *indistinguishability obfuscator* can be used to construct a "best-possible" copy-protection scheme, i.e. they show a scheme to copy-protect any classical function that can be copy-protected at all! Their observation is that if there *exists* some (as yet unknown) copy-protection scheme for a classical (deterministic) function $x \to F(x)$ that produces an unclonable quantum program $(|\psi\rangle, Q)$, then a quantum state obfuscation of F is just as good a copy-protection scheme. Indeed, this follows from the fact that the obfuscation of F is indistinguishable from an obfuscation of $(|\psi\rangle, Q)$. However, they left the *existence* of quantum state indistinguishability obfuscation as an open question, only providing a construction with respect to a quantum oracle that has no known (even heuristic) real-world instantiation. Our construction of quantum state obfuscation in the classical oracle model gives the first concrete candidate construction of quantum state obfuscation, and thus the first concrete "best-possible" copy-protection scheme for general-purpose software.

1.3 Approach

Our approach for program obfuscation begins with the notion of *fully-homomorphic encryption* (FHE). An FHE scheme encodes data x into a ciphertext $\mathsf{Enc}(x)$ so that anyone holding $\mathsf{Enc}(x)$ and a function f can produce a ciphertext $\mathsf{Enc}(f(x))$. Suppose that we have the description of a quantum circuit Q to be obfuscated. Given a *quantum fully-homomorphic encryption* (QFHE) scheme [Mah18a], which supports the evaluation of *quantum functionalities*, consider releasing an encryption $\mathsf{Enc}(Q)$. Then, any evaluator with an input x can obtain $\mathsf{Enc}(Q(x))$ by running an appropriate (quantum) 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 complete the construction, it may be tempting to simply release the QFHE secret key sk, allowing the evaluator to decrypt $\mathsf{Enc}(Q(x))$ and learn $Q(x)$. However, this would *also* allow the evaluator to decrypt $\mathsf{Enc}(Q)$ and learn Q, which is exactly what we are trying to hide. Instead, the idea is to prepare a "constrained" secret key that *only* allows decryption of ciphertexts $\mathsf{Enc}(Q(x))$ that hold an honestly evaluated output $Q(x)$.

This is the high-level idea behind each of our obfuscation constructions, though we instantiate the approach in multiple ways. First, we will discuss the approach taken by [BM22] and [BKNY23], which uses the QFHE scheme as "black-box", and later we'll see how to make "non-black-box" use of QFHE techniques to obtain an obfuscation scheme that can be applied to quantum programs with a quantum description [BBV24].

For the black-box approach, we need an additional building block: Classical verification of quantum computation (CVQC). In a CVQC scheme, a classical verifier requests the help of a quantum prover for computing a quantum circuit $Q(x)$. The result y obtained by the verifier is trusted in the sense that no malicious (quantum polynomial-time) prover should be able to cause the verifier to output $y \neq Q(x)$. In particular, we will build on a "commitment"-based approach to CVQC pioneered by [BCM+18, Mah18b].

1.3.1 Null quantum circuits

In order to prepare a constrained secret key, we consider the following idea. Given $\mathsf{Enc}(Q)$ and an input x, rather than having the evaluator simply compute the (encrypted) output $\mathsf{Enc}(Q(x))$, instruct them to utilize CVQC in order to include a *proof* π that the output was honestly evaluated. That is, they now compute a ciphertext $\mathsf{Enc}(Q(x), \pi)$ that holds the output $Q(x)$ along with a proof π that $Q(x)$ was honestly computed. Then, obfuscate the classical functionality $F[\mathsf{sk}]$ that has the QFHE secret key sk hard-coded, takes as input a ciphertext $\mathsf{Enc}(y, \pi)$, decrypts it to obtain y and π, and then only outputs the result y if π is a valid proof that $y = Q(x)$ (and otherwise just outputs a special reject symbol $\perp$).

This idea actually has a major flaw. Given oracle access to $F[\mathsf{sk}]$, an adversary can launch the following attack. First, prepare an honest output ciphertext $\mathsf{Enc}(Q(x), \pi)$. Then, depending on the first bit of the description of Q, either do nothing, or replace π with a bogus proof. Note that this operation is something that the adversary can do "under the hood" of the QFHE scheme. Then, when the resulting ciphertext is queried to $F[\mathsf{sk}]$, the

adversary can determine the first bit of Q by observing whether $F[\mathsf{sk}]$ returns a decrypted plaintext or rejects. Thus, for general Q, this approach is not secure.

However, in [BM22] we observed that this approach *is* secure in the special case that Q is a *null* quantum circuit. Indeed, if we replace the reject symbol $\perp$ with the bit 0, and Q always outputs 0 anyway, then no adversary should be able to find an input on which the oracle doesn't output 0, which thwarts the above attack. In Section 3.1, we show how to formalize this approach, obtaining an obfuscation scheme for null quantum circuits, which, as discussed earlier, yields several powerful quantum cryptographic applications.

1.3.2 Pseudo-deterministic quantum circuits

Now, in order to generalize the class of quantum circuits supported, we will have to take a different approach. Rather than verifying the quantum computation *under the QFHE*, what if we instead verified the QFHE evaluation procedure itself? That is, we hope to construct an argument system with a classical verifier V that satisfies the following properties.

- Given $\mathsf{Enc}(Q)$, some public parameters pp, and any input x, it is possible to compute a ciphertext $\mathsf{ct} = \mathsf{Enc}(Q(x))$ and a proof π such that $V(\mathsf{pp}, \mathsf{Enc}(Q), x, \mathsf{ct}, \pi) = 1$.
- Given $\mathsf{Enc}(Q)$ and the public parameters pp, no quantum polynomial-time adversary can find (x, ct, π) such that $V(\mathsf{pp}, \mathsf{Enc}(Q), x, \mathsf{ct}, \pi) = 1$ and $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \neq Q(x)$.

This argument system would, in particular, imply a notion of *publicly-verifiable* QFHE. If we could show that such a scheme exists, then an obfuscation of Q could consist of $\mathsf{pp}, \mathsf{Enc}(Q)$ along with a classical oracle that implements the function $F[\mathsf{pp}, \mathsf{Enc}(Q), \mathsf{sk}]$ defined as follows. Take as input (x, ct, π), check if $V(\mathsf{pp}, \mathsf{Enc}(Q), x, \mathsf{ct}, \pi) = 1$, and if so output $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$, and otherwise output $\perp$.

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. This thwarts the attack described above, since the secret key of the QFHE scheme is only ever used once the proof is deemed to be valid.

CVQC and its limitations. Thus, it suffices to construct a (non-interactive) classically-verifiable argument system that is powerful enough to handle computations $\mathsf{Eval}[\mathsf{Enc}(Q)]$ that map

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

where Enc is a QFHE scheme and Q is a (pseudo-)deterministic quantum circuit.

While the breakthrough work of [Mah18b] established the feasibility of CVQC, there are two major problems with using this scheme out of the box for this application.

- **Sampling circuits.** [Mah18b]'s scheme only supports the verification of (pseudo)-deterministic quantum circuits. However, QFHE evaluation procedures [Mah18a,

Bra18] are 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.[6]

- **Public verifiability.** The adversary attacking our obfuscation scheme will have (obfuscated) access to the CVQC 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, [Mah18b]'s scheme is privately-verifiable, and can be broken given this repeated access to the verifier.

In what follows, we will briefly cover the high-level ideas that we use to address each of these issues, and defer a much more detailed overview to Section 4.1.

Quantum partitioning circuits. 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.

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 $P := \mathsf{Dec}(\mathsf{sk}, \cdot)$ such that $P(\mathsf{Eval}[\mathsf{Enc}(Q)](\cdot))$ is (pseudo)-deterministic.

We say that C is a *quantum partitioning circuit* if there exists a predicate P such that $P(C(\cdot))$ is pseudo-deterministic. We think of C as abstracting the circuit $\mathsf{Eval}[\mathsf{Enc}(Q)]$ that we will ultimately want to verify, and we construct 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.[7] That is, we will need an argument system with (roughly) the following syntax (see Section 4.3.1 for a formal description).

- $\mathsf{Gen}(1^\lambda, C) \to \mathsf{pp}$: 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*.
- $\mathsf{Prove}(\mathsf{pp}, C, x) \to \pi$: The prover algorithm outputs a proof π.

[6] 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.

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

- $\mathsf{Ver}(\mathsf{pp}, C, x, \pi) \to q \cup \{\bot\}$: 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 quantum polynomial-time prover can produce an (x, π) such that $\mathsf{Ver}(\mathsf{pp}, C, x, \pi) \to q$ and $\mathsf{Out}(q, C) \neq P(C(x))$. We refer to such a protocol as a *non-interactive publicly-verifiable classical verification of quantum partitioning circuits.*

Constructing this object turns out to be quite technically involved, and the bulk of the work in [BKNY23]. Again, we provide a full overview of the techniques involved in Section 4.1, and here discuss our approach for obtaining *public verifiability*.

Publicly-decodable X-measurable commitments. As mentioned earlier, our approach builds on the "commitment"-based CVQC of [BCM+18, Mah18b]. We first abstract out a primitive at the heart of this approach, which we call an X-measurable commitment.[8]

An X-measurable commitment (XMC) is a traditional (non-interactive) classical bit commitment scheme augmented with a 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 $|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$ 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, in some sense, preserves the original qubit. One can always measure this state in the standard basis to obtain a bit b and opening information u_b, which can be considered an opening to a standard basis measurement of $|\psi\rangle$. An X-measurable commitment allows the committer the option to instead open to a *Hadamard basis* measurement of $|\psi\rangle$. More formally, it should satisfy the following syntax.

- $\mathsf{Gen}(1^\lambda) \to (\mathsf{ck}, \mathsf{rk})$: Gen outputs a public commitment key ck and a private receiver's key rk.[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]
- $\mathsf{Open}(\mathcal{B}, \mathcal{U}) \to u$: The standard basis opening algorithm performs a measurement[12] on registers $(\mathcal{B}, \mathcal{U})$ to produce a classical string u.

[8] In [BKNY23], this was called a *Pauli functional commitment*, but on second thought, we find the name X-measurable to be more precise.

[9] Note that depending on the commitment scheme, the second register may contain a superposition over random coins / opening information. For this overview, we make the simplifying assumption that there is a single string consistent with each bit and commitment.

[10] For now, assume ck is classical, though later we will consider 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.

[12] Without loss of generality, this can simply be a standard basis measurement of all registers.

- $\mathsf{Dec}(\mathsf{rk}, c, u) \to \{0, 1, \bot\}$: The standard basis decoding algorithm takes the receiver's key rk, a commitment c, an opening u, and either outputs a bit 0 or 1, or outputs $\bot$.
- $\mathsf{OpenX}(\mathcal{B}, \mathcal{U}) \to u$: The Hadamard basis opening algorithm performs a measurement on registers $(\mathcal{B}, \mathcal{U})$ to produce a classical string u.
- $\mathsf{DecX}(\mathsf{rk}, c, u) \to \{0, 1, \bot\}$: The Hadamard basis decoding algorithm takes the receiver's key rk, a commitment c, an opening u, and either outputs a bit 0 or 1, or outputs $\bot$.

The new correctness property now guarantees that committing to a qubit, opening in the Hadamard basis, and then decoding the result gives the same result as directly measuring the original qubit in the Hadamard basis.

This notion has appeared implicitly in prior work, e.g. [BCM+18, Mah18b, Vid20]. Indeed, [BCM+18, Mah18b] showed how to construct an X-measurable commitment that satisfies certain binding-like properties,[13] under the quantum hardness of LWE. Unfortunately, this X-measurable commitment is not "publicly-decodable", in the following sense. If the committer is given oracle access to the functionality $\mathsf{DecX}(\mathsf{rk}, \cdot, \cdot)$, they can repeatedly issues queries (c, u) and eventually learn the receiver's secret key rk. Then, they can use knowledge of rk to break the binding property of the commitment scheme. Indeed, this is the essential reason that previous approaches to CVQC are not publicly-verifiable.

To address this, we define a publicly-decodable notion of X-measurable commitments, in which the committer is allowed oracle access to both $\mathsf{Dec}(\mathsf{rk}, \cdot, \cdot)$ and $\mathsf{DecX}(\mathsf{rk}, \cdot, \cdot)$. We show how to construct such a commitment in the oracle model that satisfies the traditional notion of binding to standard basis measurements. To do so, we combine some of the ideas from [BCM+18, Mah18b] with techniques originally developed in the context of publicly-verifiable unclonable cryptography [AC12, BS16, AGKZ20]. Once this is done, we combine our publicly-decodable commitments with the privately-verifiable commitments of [BCM+18, Mah18b] (which satisfy extra binding properties, as alluded to above) to obtain our publicly-verifiable classical verification of quantum partitioning circuits protocol. For the details, see Section 4.1.4 and Section 4.2.

1.3.3 Quantum state obfuscation

Finally, we provide a brief overview of our approach for *quantum state obfuscation*, where the input program may have a quantum description.

Note that the previous idea of using classical verification of quantum computation to check for honest QFHE evaluation no longer works, because the "statement" to prove now includes a quantum state. This is of course not handled by techniques developed for *classical* verification. Thus, we take a different approach. In particular, we open up

[13] For example, they require that once the committer produces a valid standard basis opening, they have negligible advantage in predicting the result of a Hadamard basis opening.

the black box of QFHE, and develop an approach based on the underlying techniques proposed in [Mah18a].

QFHE review. First, we review the high-level approach underlying [Mah18a]'s construction of QFHE. An encrypted quantum state takes the form $X^x Z^z |\psi\rangle$, $\mathsf{Enc}(x, z)$, where (x, z) are quantum one-time pad (QOTP) [AMTDW00] keys, and Enc is a classical fully-homomorphic encryption of the QOTP keys. It turns out that universal quantum computation can be performed on this encrypted data by combining the following two types of operations.

- **Clifford operation:** Given an operation C that is comprised solely of Clifford gates, the evaluator can apply $CX^x Z^z |\psi\rangle = X^{x'} Z^{z'} C |\psi\rangle$, and update the QOTP keys accordingly under the classical FHE: $\mathsf{Enc}(x, z) \to \mathsf{Enc}(x', z')$.
- **Oblivious measurement:** Given an encrypted bit $\mathsf{Enc}(b)$ and a two-qubit state $|\phi\rangle$, perform one of the following two measurements.
  - If $b = 0$, measure the second qubit in the standard basis.
  - If $b = 1$, measure the XOR of the two qubits.

  Crucially, the evaluator should be able to perform this measurement without ever learning the bit b, meaning without learning *which measurement they actually performed*.[14]

[Mah18a] showed that there exists a classical FHE scheme with the property that a ciphertext $\mathsf{Enc}(b)$ can be used to perform the required oblivious measurement.

QFHE on authenticated data. As we have already seen, in order to bridge the gap between QFHE and quantum obfuscation, we will have to incorporate some notion of *verifiability*. Here, rather than trying to verify the evaluation procedure all at once, we take a more step-by-step approach. In particular, we will apply QFHE on *authenticated quantum data*, so that any dishonest operation will result in an intermediate state that no longer passes verification.

Quantum authentication has a long history of study, dating back to the original protocol of [BCG+02]. Briefly, a quantum authentication scheme encodes a state $|\psi\rangle \to_{\mathsf{vk}} |\widetilde{\psi}\rangle$ so that anyone with the verification key vk can check if the state $|\widetilde{\psi}\rangle$ has been tampered with. However, our setting imposes several additional requirements on the quantum authentication scheme, which we describe now.

[14] Technically, this is only possible if the resulting state is only correct up to a random phase flip, but this is not a problem as long as whether or not the phase flip was applied is known under the classical FHE.

- **Publicly-verifiable:** The evaluator of the obfuscated program will be able to check the authenticity of their intermediate states as they perform the obfuscated computation. Thus, the authentication scheme must remain secure even if the tampering adversary is given oracle access to the verification functionality. This situation is typically not considered in the literature on quantum authentication (with some exceptions [DS18, GYZ17]).

- **Linearly-homomorphic:** Recall that in the QFHE template described above, Clifford operations are essentially "for free". Thus, we should also be able to easily perform Clifford operations on the authenticated data. One of our observations is that, in fact, in suffices to be able to perform only CNOT operations on the authenticated data, while the remainder of the Clifford operations can be "absorbed" into the oblivious measurement step. Still, this requires the authenticated scheme to support homomorphic CNOT operations, which we view as a natural quantum analogue of linear-homomorphism.

- **ZX-measurable:** In order to perform the oblivious measurement step, it must be possible to perform both standard and Hadamard basis measurements[15] on authenticated data, and *classically* decode the results of these measurements. We refer to this property as ZX-measurable.

Construction from random CSS codes. We construct an authentication scheme that satisfies all of these properties simultaneously via a particular instantiation of the "encode-encrypt" paradigm [BGS13]. Such schemes are parameterized by a family of CSS codes $\mathscr{C}$, and operate as follows. To encode a qubit $|\psi\rangle$, sample a random code $C \leftarrow \mathscr{C}$ from the family, sample a quantum one-time pad key (x, z), and output the "encoded-and-encrypted" state $X^x Z^x C |\psi\rangle$.

We let $\mathscr{C}$ be the family of *all* (single-qubit) CSS codes, which means that sampling a random code from this family corresponds to sampling a uniformly random subspace S (say, of dimension λ in an ambient space of dimension $2\lambda + 1$), along with a random shift Δ. Then, an authentication of qubit $\alpha_0 |0\rangle + \alpha_1 |1\rangle$ takes the form

$$X^x Z^z \left(\alpha_0 |S\rangle + \alpha_1 |S + \Delta\rangle\right),$$

where for any (affine) subspace S, the state $|S\rangle$ is the uniform superposition over all vectors in S.

By encoding all qubits using the same subspace S and shift Δ (but potentially with different one-time pad keys), we can support homomorphic CNOT operations. Moreover, it is not hard to see that the scheme supports classically-decodable standard and Hadamard basis measurements (as do all encode-encrypt schemes). Finally, we show that this scheme is indeed secure in the public-verification setting, drawing a connection

[15] More generally, it should be possible to perform even entangled measurements, as long as they are diagonal in a tensor product of Z and X bases.

to proof techniques used in the setting of publicly-verifiable unclonable cryptography, which also relies on such uniformly random subspace (or coset) states.

While this is one of the central building blocks underlying our construction of quantum state obfuscation, several details remain in the implementation of our obfuscation scheme. We refer the reader to Section 5.1 to a more thorough overview of the techniques involved.

1.4 Future directions

The results contained in this thesis constitute the first evidence that interesting and useful classes of quantum computations can be obfuscated. However, this progress perhaps raises more questions than it answers. In this section, we'll highlight three directions for future research.

Classical obfuscated programs. Ideally, we would like to show that given the classical description of a quantum program (i.e., a description of the sequence of gates to apply), it is possible to output a classically-described obfuscated program. Unfortunately, we currently only know how to achieve this for null quantum circuits. Indeed, our schemes that support obfuscation of general (pseudo)-deterministic quantum circuits output a quantum state as part of the obfuscated program, even if the original program only had a classical description.

We leave open the question of obfuscation for pseudo-deterministic quantum circuits where the obfuscated program has a classical description. However, we observe that the [BKNY23] approach based on classical verification of QFHE evaluation appears amenable to "de-quantization". The main problem to solve here is to construct publicly-verifiable X-measurable commitments using a classical commitment key rather than a quantum commitment key.[16] Conceptually, this seems to require classical parameters that allow for the sampling of publicly-verifiable unclonable states (with some additional structure). A candidate for this task is an (obfuscated) *affine partition function*, as proposed by [AGKZ20]. However, establishing the security of this approach, even in the classical oracle model, remains an open question.

Security in the plain model. We prove security of our obfuscation schemes in the idealized classical oracle model. An important question going forward is whether we can establish security in the plain model, i.e. assuming only indistinguishability obfuscation for classical circuits, or perhaps some other concrete assumption on the classical obfuscation scheme. For example, does "best-possible" obfuscation for classical computation imply best-possible copy-protection? Does non-interactive zero-knowledge for QMA exist under any concrete assumption(s)?

[16] We would also have to de-quantize signature tokens, but this would likely follow from the same techniques needed to de-quantize the X-measurable commitment.

Broadening the class of obfuscatable computation. Finally, perhaps the most exciting question left is the same question that inspired this research in the first place. Simply put, what class of (quantum) computations can be (plausibly) obfuscated?

Ultimately, we want to show that arbitrary polynomial-time quantum operations can be obfuscated: Given the classical description of any efficiently-computable quantum map M (that is, a completely positive trace-preserving map), produce a classically-described *obfuscated* program $\widetilde{M}$ that implements the same (or very close to the same) map as M. We leave it as an open problem to come up with any plausible candidate for this task.

There are also interesting intermediate goals. For example, can we go beyond pseudo-deterministic circuits and obfuscate *any* quantum circuit with classical inputs and classical outputs? That is, can we obfuscate circuits that output arbitrary distributions over classical outputs? We note that the [BBV24] scheme is actually a candidate obfuscator for all such circuits, but it remains an open problem to prove its security when applied to non-pseudo-deterministic circuits.

2 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 exist constants c and N such that for all $n > N$, $f(n) < n^c$.

For two probability distributions D_0, D_1 with support S, let

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

denote the total variation distance. For a set S, we let $x \leftarrow S$ denote sampling a uniformly random element x from S. If D is a distribution, we let $x \leftarrow D$ denote sampling from D, and let

$$\{x : x \leftarrow D_0\} \approx_\epsilon \{x : x \leftarrow D_1\}$$

denote that $\mathsf{TV}(D_0, D_1) \leq \epsilon$. Finally, we denote a linear combination of distributions by

$$(1-\delta)\{x : x \leftarrow D_0\} + \delta\{x : x \leftarrow D_1\},$$

meaning with probability $1-\delta$, sample from D_0 and with probability δ, sample from D_1.

2.1 Quantum information

2.1.1 Background

An n-qubit register $\mathcal{X}$ refers to a Hilbert space $\mathbb{C}^{2^n}$. A *pure 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 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 $\mathcal{Y}$. A *unitary* $U : \mathcal{X} \to \mathcal{X}$ is a special case of a quantum operation that satisfies $U^\dagger U = UU^\dagger = \mathcal{I}^{\mathcal{X}}$, where $\mathcal{I}^{\mathcal{X}}$ 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 $\sum_i \Pi_i = \mathcal{I}$. Throughout, we will often write an expression like $\Pi|\psi\rangle$, where $|\psi\rangle$ is supported on multiple registers, say $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$, while Π has only been defined on a subset of these registers, say $\mathcal{Y}$. In this case, we technically mean $(\mathcal{I}^{\mathcal{X}} \otimes \Pi^{\mathcal{Y}} \otimes \mathcal{I}^{\mathcal{Z}})|\psi\rangle$, but we drop the identity matrices to reduce notational clutter.

A family of quantum circuits is a sequence of quantum operations $\{C_\lambda\}_{\lambda \in \mathbb{N}}$, parameterized by the security parameter λ. We say that the family is *quantum polynomial time* (QPT) if C_λ can be implemented with a $\mathrm{poly}(\lambda)$-size quantum circuit. A family of *oracle-aided* quantum circuits $\{C_\lambda^{\mathsf{F}}\}_{\lambda \in \mathbb{N}}$ has access to an oracle $\mathsf{F} : \{0,1\}^* \to \{0,1\}^*$ that implements some deterministic classical map. That is, C_λ can query a unitary that applies the map

$|x\rangle|y\rangle \to |x\rangle|y \oplus \mathsf{F}(x)\rangle$. We say that the family is *quantum polynomial query* (QPQ) if C_λ only makes $\mathrm{poly}(\lambda)$-many queries to F, but is otherwise computationally unbounded.

Let Tr denote the trace operator. The *trace distance* between states ρ, τ, denoted $\mathsf{TD}(\rho, \tau)$ is defined as

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

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.

For any set S, we define $O[S]$ to be the boolean function that checks for membership in S and define the projector

$$\Pi[S] = \sum_{s\in S} |s\rangle\langle s|.$$

Definition 2.1 (Pseudo-deterministic quantum circuit)**.** *A family of pseudo-deterministic quantum circuits $\{Q_\lambda\}_{\lambda\in\mathbb{N}}$ is defined as follows. The circuit Q_λ takes as input a classical string $x \in \{0,1\}^{m(\lambda)}$ 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*

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

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 $\Pr[Q(x) = b] = 1 - \mathrm{negl}(\lambda)$.

Next, we define a notion of quantum computation where the program itself has a potentially quantum description.

Definition 2.2 (Quantum program)**.** *A quantum implementation of a functionality with classical inputs and outputs, or, a* quantum program*, is a pair $(|\psi\rangle, C)$, where $|\psi\rangle$ is a state and C is the classical description of a quantum circuit. For any classical input $x \in \{0,1\}^m$, we write $y \leftarrow C(|x\rangle|\psi\rangle)$ to denote the result of running C and then measuring a dedicated m'-qubit output register in the standard basis to obtain y.*

- *We say that the program is* deterministic *if for all x, there exists $y \in \{0,1\}^{m'}$ such that*

$$\Pr[C(|x\rangle|\psi\rangle) = y] = 1.$$

- *We say that a family of quantum programs $\{(|\psi_\lambda\rangle, C_\lambda)\}_{\lambda\in\mathbb{N}}$ is* ϵ-pseudo-deterministic *for some $\epsilon = \epsilon(\lambda)$ if for all sequences of inputs $\{x_\lambda\}_{\lambda\in\mathbb{N}}$, there exists a sequence of outputs $\{y_\lambda\}_{\lambda\in\mathbb{N}}$ such that*

$$\Pr[C_\lambda(|x_\lambda\rangle|\psi_\lambda\rangle) = y_\lambda] \geq 1 - \epsilon(\lambda).$$

We will often leave the dependence on λ implicit, and just refer to (pseudo)-deterministic programs $(|\psi\rangle, C)$. We will denote by $Q(x)$ the string y such that $\Pr[C(|x\rangle|\psi\rangle) = y] \geq 1 - \epsilon(\lambda)$, and refer to Q as the map induced by *$(|\psi\rangle, C)$.*

2.1.2 Lemmas

Lemma 2.3 (Gentle measurement [Win99])**.** *Let ρ be a quantum state and let $(\Pi, \mathcal{I} - \Pi)$ be a projective measurement such that $\mathrm{Tr}(\Pi\rho) \geq 1 - \delta$. Let*

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

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

Lemma 2.4. *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. □

Lemma 2.5 (Pauli Twirl over Affine Subspaces)**.** *Let $R, \widehat{R} \subseteq \mathbb{F}_2^n$ be subspaces of $\mathbb{F}_2^n$, and let (x_0, z_0, x_1, z_1) be such that either $x_0 \oplus x_1 \notin \widehat{R}^\perp$ or $z_0 \oplus z_1 \notin R^\perp$. Then for any $\Delta \notin R, \widehat{\Delta} \notin \widehat{R}$ and density matrix ρ on $m \geq n$ qubits,*

$$\sum_{x \in R + \Delta, z \in \widehat{R} + \widehat{\Delta}} (Z^z X^x)(X^{x_0} Z^{z_0})(X^x Z^z) \rho (Z^z X^x)(Z^{z_1} X^{x_1})(X^x Z^z) = 0.$$

Proof. Using the fact that $X^x Z^z = (-1)^{x \cdot z} Z^z X^x$, we write

$$\begin{aligned}
&\sum_{x \in R + \Delta, z \in \widehat{R} + \widehat{\Delta}} (Z^z X^x)(X^{x_0} Z^{z_0})(X^x Z^z) \rho (Z^z X^x)(Z^{z_1} X^{x_1})(X^x Z^z) \\
&= \sum_{x \in R + \Delta, z \in \widehat{R} + \widehat{\Delta}} (-1)^{x \cdot z_0 + z \cdot x_0 + x \cdot z_1 + z \cdot x_1} (X^{x_0} Z^{z_0}) \rho (Z^{z_1} X^{x_1}) \\
&= \left(\sum_{z \in \widehat{R} + \widehat{\Delta}} (-1)^{z \cdot (x_0 \oplus x_1)} \right) \left(\sum_{x \in R + \Delta} (-1)^{x \cdot (z_0 \oplus z_1)} \right) (x^{x_0} Z^{z_0}) \rho (Z^{z_1} X^{x_1}) \\
&= \left(\sum_{z \in \widehat{R}} (-1)^{z \cdot (x_0 \oplus x_1)} \right) \left(\sum_{x \in R} (-1)^{x \cdot (z_0 \oplus z_1)} \right) (-1)^{\widehat{\Delta} \cdot (x_0 \oplus x_1)} (-1)^{\Delta \cdot (z_0 \oplus z_1)} (x^{x_0} Z^{z_0}) \rho (Z^{z_1} X^{x_1}) \\
&= 0,
\end{aligned}$$

where the last equality follows because either $x_0 \oplus x_1 \notin \widehat{R}^\perp$ or $z_0 \oplus z_1 \notin R^\perp$. □

The following two lemmas are applications of Cauchy-Schwarz. The first is adapted from [DS23].

Lemma 2.6. *Let $\mathcal{K}$ be a set of keys, N an integer, and $\{|\psi_k\rangle, \{\Pi_{k,i}\}_{i\in[N]}, O_k\}_{k\in\mathcal{K}}$ be a set of states $|\psi_k\rangle$, projective submeasurements $\{\Pi_{k,i}\}_{i\in[N]}$, and classical functions O_k such that $|\psi_k\rangle \in \mathsf{Im}(\sum_i \Pi_{k,i})$ for each k. Then for any distinguisher D, which we take to be an oracle-aided binary outcome projector, it holds that*

$$\mathop{\mathbb{E}}_{k\leftarrow\mathcal{K}}\left[\|D^{O_k}|\psi_k\rangle\|^2\right] - \sum_i \mathop{\mathbb{E}}_{k\leftarrow\mathcal{K}}\|D^{O_k}\Pi_{k,i}|\psi_k\rangle\|^2 \leq N\cdot\left[\sum_{i\neq j}\mathop{\mathbb{E}}_{k\leftarrow\mathcal{K}}\|\Pi_{k,j}D^{O_k}\Pi_{k,i}|\psi_k\rangle\|^2\right]^{1/2}.$$

Proof.

$$\begin{aligned}
&\mathop{\mathbb{E}}_{k\leftarrow\mathcal{K}}\left[\left\|D^{O_k}|\psi_k\rangle\right\|^2\right]\\
&= \mathop{\mathbb{E}}_{k\leftarrow\mathcal{K}}\left[\left|\langle\psi_k|\left(\sum_i\Pi_{k,i}\right)D^{O_k}\left(\sum_i\Pi_{k,i}\right)|\psi_k\rangle\right|\right]\\
&= \mathop{\mathbb{E}}_{k\leftarrow\mathcal{K}}\left[\left|\sum_i\langle\psi_k|\Pi_{k,i}D^{O_k}\Pi_{k,i}|\psi_k\rangle + \sum_{i\neq j}\langle\psi_k|\Pi_{k,j}D^{O_k}\Pi_{k,i}|\psi_k\rangle\right|\right]\\
&\leq \sum_i\mathop{\mathbb{E}}_{k\leftarrow\mathcal{K}}\left\|D^{O_k}\Pi_{k,i}|\psi_k\rangle\right\|^2 + \mathop{\mathbb{E}}_{k\leftarrow\mathcal{K}}\left[\sum_{i\neq j}\left|\langle\psi_k|\Pi_{k,j}D^{O_k}\Pi_{k,i}|\psi_k\rangle\right|\right]\\
&\leq \sum_i\mathop{\mathbb{E}}_{k\leftarrow\mathcal{K}}\left\|D^{O_k}\Pi_{k,i}|\psi_k\rangle\right\|^2 + \mathop{\mathbb{E}}_{k\leftarrow\mathcal{K}}\left[\sum_{i\neq j}\sqrt{\langle\psi_k|\Pi_{k,i}D^{O_k}\Pi_{k,j}D^{O_k}\Pi_{k,i}|\psi_k\rangle}\right]\\
&\leq \sum_i\mathop{\mathbb{E}}_{k\leftarrow\mathcal{K}}\left\|D^{O_k}\Pi_{k,i}|\psi_k\rangle\right\|^2 + N\cdot\mathop{\mathbb{E}}_{k\leftarrow\mathcal{K}}\left[\left(\sum_{i\neq j}\|\Pi_{k,j}D^{O_k}\Pi_{k,i}|\psi_k\rangle\|^2\right)^{1/2}\right]\\
&\leq \sum_i\mathop{\mathbb{E}}_{k\leftarrow\mathcal{K}}\left\|D^{O_k}\Pi_{k,i}|\psi_k\rangle\right\|^2 + N\cdot\left[\sum_{i\neq j}\mathop{\mathbb{E}}_{k\leftarrow\mathcal{K}}\|\Pi_{k,j}D^{O_k}\Pi_{k,i}|\psi_k\rangle\|^2\right]^{1/2},
\end{aligned}$$

where the first inequality is the triangle inequality, the second follows from Cauchy-Schwarz applied to vectors $|\psi_k\rangle$ and $\Pi_{k,j}D^{O_k}\Pi_{k,i}|\psi\rangle$, the third follows from Cauchy-Shwarz applied to the length N^2 vector $(1,\ldots,1)$ and the vector with (i,j)'th entry equal to

$$\sqrt{\langle\psi_k|\Pi_{k,i}D^{O_k}\Pi_{k,j}D^{O_k}\Pi_{k,i}|\psi_k\rangle},$$

and the fourth is Jensen's inequality.

□

Lemma 2.7. *Let $\mathcal{K}$ be a set of keys, N an integer, and $\{|\psi_k\rangle, \{\Pi_{k,i}\}_{i\in[N]}, O_k, \Gamma_k\}_{k\in\mathcal{K}}$ be a set of states $|\psi_k\rangle$, projective submeasurements $\{\Pi_{k,i}\}_{i\in[N]}$, classical function O_k, and projective measurements Γ_k such that $|\psi_k\rangle \in \mathsf{Im}(\sum_i \Pi_{k,i})$ for each k. Then for any oracle-aided unitary U, it holds that*

$$\mathop{\mathbb{E}}_{k\leftarrow\mathcal{K}}[\|\Gamma_k U^{O_k}|\psi_k\rangle\|^2] \leq N \cdot \sum_i \mathop{\mathbb{E}}_{k\leftarrow\mathcal{K}} \|\Gamma_k U^{O_k}\Pi_{k,i}|\psi_k\rangle\|^2.$$

Proof.

$$\begin{aligned}
&\mathop{\mathbb{E}}_{k\leftarrow\mathcal{K}}[\|\Gamma_k U^{O_k}|\psi_k\rangle\|^2] \\
&\leq \mathop{\mathbb{E}}_{k\leftarrow\mathcal{K}}\left[\left(\sum_i \|\Gamma_k U^{O_k}\Pi_{k,i}|\psi_k\rangle\|\right)^2\right] \\
&\leq \mathop{\mathbb{E}}_{k\leftarrow\mathcal{K}}\left[\left(\sqrt{N}\sqrt{\sum_i \|\Gamma_k U^{O_k}\Pi_{k,i}|\psi_k\rangle\|^2}\right)^2\right] \\
&= N \cdot \sum_i \mathop{\mathbb{E}}_{k\leftarrow\mathcal{K}} \|\Gamma_k U^{O_k}\Pi_{k,i}|\psi_k\rangle\|^2,
\end{aligned}$$

where the first inequality is the triangle inequality, and the second follows from Cauchy-Shwarz applied to the length N vector $(1,\ldots,1)$ and the vector with i'th entry equal to $\|\Gamma_k U^{O_k}\Pi_{k,i}|\psi_k\rangle\|$.

□

We will frequently invoke the following lemma in order to switch between two oracles that differ on hard-to-find inputs. The proof is a standard oracle hybrid argument.

Lemma 2.8. *For each $\lambda\in\mathbb{N}$, let $\mathcal{K}_\lambda$ be a set of keys, and $\{|\psi_k\rangle, O_k^0, O_k^1, S_k\}_{k\in\mathcal{K}_\lambda}$ be a set of states $|\psi_k\rangle$, classical functions O_k^0, O_k^1, and sets of inputs S_k. Suppose that the following properties holds.*

1. *The oracles O_k^0 and O_k^1 are identical on inputs outside of S_k.*
2. *For any oracle-aided unitary U with $q = q(\lambda)$ queries, there is some $\epsilon = \epsilon(\lambda)$ such that*

$$\mathop{\mathbb{E}}_{k\leftarrow\mathcal{K}}\left[\left\|\Pi[S_k]U^{O_k^0}|\psi_k\rangle\right\|^2\right] \leq \epsilon.$$

Then, for any oracle-aided unitary U with $q(\lambda)$ queries and distinguisher D,

$$\left|\Pr_{k\leftarrow\mathcal{K}}\left[D\left(k, U^{O_k^0}|\psi_k\rangle\right)=0\right] - \Pr_{k\leftarrow\mathcal{K}}\left[D\left(k, U^{O_k^1}|\psi_k\rangle\right)=0\right]\right| \leq 4q\sqrt{\epsilon}.$$

Proof. For each $i\in[0,\ldots,q]$, define hybrid $\mathcal{H}_i$ to sample $k\leftarrow\mathcal{K}$ and output $(k, U^{(\cdot)}|\psi_k\rangle)$, where U's first $q-i$ oracle queries are answered with O_k^0 and U's final i oracle queries are answered with O_k^1. For each $i\in[0,\ldots,q-1]$, define hybrid $\mathcal{H}_i'$ to be identical to $\mathcal{H}_i$ except that we apply the measurement $\{\Pi[S_k], \mathcal{I}-\Pi[S_k]\}$ to U's state right before the $q-i$'th oracle query, and post-select on obtaining the second outcome. Then for any $i\in[0,\ldots,q-1]$,

- By condition 2 of the lemma statement and Lemma 2.3, it holds that $\mathsf{TD}(\mathcal{H}_i, \mathcal{H}'_i) \leq 2\sqrt{\epsilon}$.
- By conditions 1 and 2 of the lemma statement and Lemma 2.3, it holds that $\mathsf{TD}(\mathcal{H}'_i, \mathcal{H}_{i+1}) \leq 2\sqrt{\epsilon}$.

The lemma follows by summing over differences in trace distance induced by the $2q$ hybrid switches.

□

Lemma 2.9 (Measure and re-program [DFMS19, DFM20]). [17] *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 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*

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

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.

[17] 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.

2.2 Cryptography

2.2.1 Obfuscation

Definition 2.10 (Obfuscation)**.** *An obfuscator for pseudo-deterministic quantum (resp. classical) circuits is a pair of algorithms* $(\mathsf{Obf}, \mathsf{Eval})$ *with the following syntax.*

- $\mathsf{Obf}(1^\lambda, Q) \to \widetilde{Q}$*: The obfuscator 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* $\widetilde{Q}$*.*
- $\mathsf{Eval}\left(\widetilde{Q}, x\right) \to b$*: The evaluator takes as input an obfuscated circuit* $\widetilde{Q}$ *and an input* x*, and outputs a bit* $b \in \{0,1\}$*.*

Correctness *is defined as follows for any pseudo-deterministic (resp. classical) family of circuits* $\{Q_\lambda\}_{\lambda\in\mathbb{N}}$ *with input length* $m(\lambda)$*.*

$$\forall x \in \{0,1\}^{m(\lambda)}, \Pr\left[\mathsf{Eval}\left(\widetilde{Q}, x\right) = Q_\lambda(x) : \widetilde{Q} \leftarrow \mathsf{Obf}(1^\lambda, Q_\lambda)\right] = 1 - \mathrm{negl}(\lambda).$$

We define two notions of security.

- ***Ideal obfuscation****: For any QPT adversary* $\{A_\lambda\}_{\lambda\in\mathbb{N}}$*, there exists a QPT simulator* $\{\mathsf{Sim}_\lambda\}_{\lambda\in\mathbb{N}}$ *such that for any polynomial* $n(\lambda)$*, pseudo-deterministic family of circuits* $\{Q_\lambda\}_{\lambda\in\mathbb{N}}$ *with input length* $m(\lambda)$ *and size at most* $n(\lambda)$*, and QPT distinguisher* $\{D_\lambda\}_{\lambda\in\mathbb{N}}$*,*

$$\left|\Pr\left[1 \leftarrow D_\lambda\left(A_\lambda\left(\mathsf{Obf}(1^\lambda, Q_\lambda)\right)\right)\right] - \Pr\left[1 \leftarrow D_\lambda\left(\mathsf{Sim}_\lambda^{Q_\lambda}\left(1^\lambda, n(\lambda), m(\lambda)\right)\right)\right]\right| = \mathrm{negl}(\lambda).$$

- ***Indistinguishability obfuscation****: For any QPT adversary* $\{A_\lambda\}_{\lambda\in\mathbb{N}}$ *and pair of functionally equivalent families of pseudo-deterministic (resp. classical) circuits* $\{Q_{0,\lambda}\}_{\lambda\in\mathbb{N}}, \{Q_{1,\lambda}\}_{\lambda\in\mathbb{N}}$*,*

$$\left|\Pr\left[1 \leftarrow A_\lambda\left(\mathsf{Obf}(1^\lambda, Q_{0,\lambda})\right)\right] - \Pr\left[1 \leftarrow A_\lambda\left(\mathsf{Obf}(1^\lambda, Q_{1,\lambda})\right)\right]\right| = \mathrm{negl}(\lambda).$$

Next, we define a notion of obfuscation for null quantum circuits.

Definition 2.11 (Obfuscation of null quantum circuits)**.** *An obfuscator for null quantum circuits is a pair of QPT algorithms* $(\mathsf{NObf}, \mathsf{NEval})$ *with the following syntax.*

- $\mathsf{NObf}(1^\lambda, Q) \to \widetilde{Q}$*: The obfuscator takes as input the security parameter* 1^λ *and the description of a quantum circuit with quantum input and one bit of classical output, and outputs an obfuscated circuit* $\widetilde{Q}$*.*
- $\mathsf{NEval}\left(\widetilde{Q}, |\psi\rangle\right) \to b$*: The evaluator takes as input an obfuscated circuit* $\widetilde{Q}$ *and an input* $|\psi\rangle$ *and outputs a bit* $b \in \{0,1\}$*.*

An obfuscator $(\mathsf{NObf}, \mathsf{NEval})$ *for null quantum circuits is* ***correct*** *if there exists a polynomial* $k(\lambda)$ *such that for all polynomial-size families of quantum circuits* $\{Q_\lambda\}_{\lambda\in\mathbb{N}}$ *and inputs* $\{|\psi_\lambda\rangle\}_{\lambda\in\mathbb{N}}$ *such that there exists* $\{b_\lambda\}_{\lambda\in\mathbb{N}}$ *such that* $\Pr[Q_\lambda(|\psi_\lambda\rangle) = b_\lambda] = 1 - \mathrm{negl}(\lambda)$*, it holds that*

$$\Pr\left[\mathsf{NEval}\left(\widetilde{Q}, |\psi_\lambda\rangle^{\otimes k(\lambda)}\right) = b_\lambda : \widetilde{Q} \leftarrow \mathsf{NObf}(1^\lambda, Q_\lambda)\right] = 1 - \mathrm{negl}(\lambda).$$

An obfuscator $(\mathsf{NObf}, \mathsf{NEval})$ *for null quantum circuits is* ***secure*** *if for any QPT adversary* $\{A_\lambda\}_{\lambda\in\mathbb{N}}$ *and polynomial-size sequences of quantum circuits* $\{Q_{0,\lambda}\}_{\lambda\in\mathbb{N}}, \{Q_{1,\lambda}\}_{\lambda\in\mathbb{N}}$ *such that for all inputs* $\{|\psi_\lambda\rangle\}_{\lambda\in\mathbb{N}}$*,*

$$\Pr[Q_{0,\lambda}(|\psi_\lambda\rangle) = 0] = 1 - \mathrm{negl}(\lambda) \text{ and } \Pr[Q_{1,\lambda}(|\psi_\lambda\rangle) = 0] = 1 - \mathrm{negl}(\lambda),$$

it holds that

$$\left|\Pr\left[1 \leftarrow A_\lambda\left(\mathsf{NObf}(1^\lambda, Q_{0,\lambda})\right)\right] - \Pr\left[1 \leftarrow A_\lambda\left(\mathsf{NObf}(1^\lambda, Q_{1,\lambda})\right)\right]\right| = \mathrm{negl}(\lambda).$$

Remark 2.12. *Note that our definition of obfuscation for null quantum circuits only guarantees correctness for inputs that either accept or reject with high probability, and, moreover, requires the evaluator to possess multiple copies of any quantum input. Thus, this definition can be considered obfuscation for null* pseudo-deterministic *quantum circuits (strictly generalizing the standard notion of obfuscation for null classical circuits) with an additional correctness guarantee that must hold for quantum inputs that yield a (nearly) deterministic outcome. There is no correctness guarantee for quantum inputs that do not yield a (nearly) deterministic outcome.*

Next, we define a notion of obfuscation for quantum computation where the program itself has a potentially quantum description.

Definition 2.13 (Quantum state obfuscation)**.** *For any* $\epsilon = \epsilon(\lambda)$*, let* $\mathcal{C}_\epsilon$ *be the set of families of* ϵ*-pseudo-deterministic quantum programs (Definition 2.2), where each family* $\{|\psi_\lambda\rangle, C_\lambda\}_{\lambda\in\mathbb{N}} \in \mathcal{C}_\epsilon$ *is associated with an induced family of maps* $\{Q_\lambda : \{0,1\}^{m(\lambda)} \to \{0,1\}^{m'(\lambda)}\}_{\lambda\in\mathbb{N}}$*. A quantum state obfuscator is a pair of QPT algorithms* $(\mathsf{QSObf}, \mathsf{QSEval})$ *with the following syntax.*

- $\mathsf{QSObf}\left(1^\lambda, |\psi\rangle, C\right) \to |\widetilde{\psi}\rangle$*: The obfuscator takes as input the security parameter* 1^λ *and a quantum program* $(|\psi\rangle, C)$*, and outputs an obfuscated state* $|\widetilde{\psi}\rangle$*.*
- $\mathsf{QSEval}\left(|\widetilde{\psi}\rangle, x\right) \to y$*: The evaluator takes as input an obfuscated state* $|\widetilde{\psi}\rangle$ *and an input* $x \in \{0,1\}^{m(\lambda)}$*, and outputs* $y \in \{0,1\}^{m'(\lambda)}$*.*

Correctness *is defined as follows for any quantum program* $\{|\psi_\lambda\rangle, C_\lambda\}_{\lambda\in\mathbb{N}}$*.*

$$\forall x \in \{0,1\}^{m(\lambda)}, \Pr\left[\mathsf{QSEval}\left(|\widetilde{\psi}\rangle, x\right) = Q_\lambda(x) : |\widetilde{\psi}\rangle \leftarrow \mathsf{QSObf}\left(1^\lambda, |\psi_\lambda\rangle, C_\lambda\right)\right] = 1 - \mathrm{negl}(\lambda).$$

We define two notions of security.

- ***Ideal obfuscation**: For any QPT adversary $\{A_\lambda\}_{\lambda\in\mathbb{N}}$, there exists a QPT simulator $\{\mathsf{Sim}_\lambda\}_{\lambda\in\mathbb{N}}$ such that for any polynomial $n(\lambda)$, program $\{|\psi_\lambda\rangle, C_\lambda\}_{\lambda\in\mathbb{N}} \in \mathcal{C}_\epsilon$ with induced family of maps $\{Q_\lambda : \{0,1\}^{m(\lambda)} \to \{0,1\}^{m'(\lambda)}\}_{\lambda\in\mathbb{N}}$ such that $|\psi_\lambda\rangle$ has at most $n(\lambda)$ qubits and C_λ has at most $n(\lambda)$ gates, and QPT distinguisher $\{D_\lambda\}_{\lambda\in\mathbb{N}}$,*

$$\Big|\Pr\left[1 \leftarrow D_\lambda\left(A_\lambda\left(\mathsf{QSObf}\left(1^\lambda, |\psi_\lambda\rangle, C_\lambda\right)\right)\right)\right] - \Pr\left[1 \leftarrow D_\lambda\left(\mathsf{Sim}_\lambda^{Q_\lambda}\left(1^\lambda, n(\lambda), m(\lambda), m'(\lambda)\right)\right)\right]\Big| = \mathrm{negl}(\lambda).$$

- ***Indistinguishability obfuscation**: For any polynomial $n(\lambda)$, pair of families $\{|\psi_{\lambda,0}\rangle, C_{\lambda,0}\}_{\lambda\in\mathbb{N}}$, $\{|\psi_{\lambda,1}\rangle, C_{\lambda,1}\}_{\lambda\in\mathbb{N}} \in \mathcal{C}_\epsilon$ with the same induced map $\{Q_\lambda : \{0,1\}^{m(\lambda)} \to \{0,1\}^{m'(\lambda)}\}_{\lambda\in\mathbb{N}}$ such that $|\psi_{\lambda,0}\rangle$ and $|\psi_{\lambda,1}\rangle$ both have at most $n(\lambda)$ qubits and $C_{\lambda,0}$ and $C_{\lambda,1}$ both have at most $n(\lambda)$ gates, and QPT adversary $\{A_\lambda\}_{\lambda\in\mathbb{N}}$,*

$$\Big|\Pr\left[1 \leftarrow A_\lambda\left(\mathsf{QSObf}\left(1^\lambda, |\psi_{\lambda,0}\rangle, C_{\lambda,0}\right)\right)\right] - \Pr\left[1 \leftarrow A_\lambda\left(\mathsf{QSObf}\left(1^\lambda, |\psi_{\lambda,1}\rangle, C_{\lambda,1}\right)\right)\right]\Big| = \mathrm{negl}(\lambda).$$

Remark 2.14 (Classical oracle model)**.** *In this work, we construct obfuscation schemes in the* classical oracle model*. In this model, we allow the obfuscation algorithm to additionally output the description of a classical deterministic functionality O, and both the evaluation algorithm and the adversary are granted (quantum-accessible) oracle access to O. Any scheme in the classical oracle model may be heuristically instantiated in the plain model by using a post-quantum indistinguishability obfuscator to obfuscate O and include its obfuscation in the description of the obfuscated circuit.*

2.2.2 Trapdoor claw-free functions

In this section, we define a variant of trapdoor claw-free functions, which we refer to as a dual-mode noisy trapdoor claw-free function family (dTCF) with an adaptive hardcore bit.

Definition 2.15 (dTCF with adaptive hardcore bit)**.** *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 noisy trapdoor claw-free function *family with an* adaptive hardcore bit *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 further define*

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

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* $\bot$*. If* $h = 1$*, it outputs two pairs* $(0, x_0)$ *and* $(1, x_1)$ *with* $x_0, x_1 \in X$*, or* $\bot$*.*

- $\mathsf{Check}(\mathsf{pk}, b, x, y) \to \{\top, \bot\}$ *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* $\bot$*.*

- $\mathsf{IsValid}(x_0, x_1, d) \to \{\top, \bot\}$ *is a deterministic classical algorithm that takes as input two elements* $x_0, x_1 \in X$ *and a string* d*, and outputs either* $\top, \bot$*, 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. ***Mode indistinguishability:*** *For every QPT adversary* $\{A_\lambda\}_{\lambda\in\mathbb{N}}$*,*

$$\left|\Pr\left[1 \leftarrow A_\lambda(\mathsf{pk}) : (\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda, 0)\right] - \Pr\left[1 \leftarrow A_\lambda(\mathsf{pk}) : (\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda, 1)\right]\right| = \mathrm{negl}(\lambda).$$

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* $\{A_\lambda\}_{\lambda\in\mathbb{N}}$*,*

$$\left|\Pr\left[\begin{array}{ll} \mathsf{Check}(\mathsf{pk},b,x,y) = 1 \ \wedge & (\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda,1) \\ d \in \mathsf{Valid}_{x_0,x_1} \ \wedge & : \quad (y,b,x,d) \leftarrow A_\lambda(\mathsf{pk}) \\ d\cdot(J(x_0)\oplus J(x_1)) = 0 & ((0,x_0),(1,x_1)) \coloneqq \mathsf{Invert}(1,\mathsf{sk},y) \end{array}\right] - \Pr\left[\begin{array}{ll} \mathsf{Check}(\mathsf{pk},b,x,y) = 1 \ \wedge & (\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda,1) \\ d \in \mathsf{Valid}_{x_0,x_1} \ \wedge & : \quad (y,b,x,d) \leftarrow A_\lambda(\mathsf{pk}) \\ d\cdot(J(x_0)\oplus J(x_1)) = 1 & ((0,x_0),(1,x_1)) \coloneqq \mathsf{Invert}(1,\mathsf{sk},y) \end{array}\right]\right| = \mathrm{negl}(\lambda).$$

The works of [BCM+18, Mah18b] showed that, assuming the quantum hardness of learning with errors (LWE), there exists a dual-mode noisy trapdoor claw-free function family with an adaptive hardcore bit.

2.2.3 Quantum fully-homomorphic encryption

We define quantum fully-homomorphic encryption (QFHE) with classical keys and classical encryption of classical messages.

Definition 2.16 (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 2.17 (Semantic security)**.** *A QFHE scheme* $(\mathsf{Gen},\mathsf{Enc},\mathsf{Eval},\mathsf{Dec})$ *is secure if for any QPT adversary* $\{A_\lambda\}_{\lambda\in\mathbb{N}}$ *and circuit depth* D*,*

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

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

Definition 2.18 (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 [Mah18a] and Brakerski [Bra18] show that such a QFHE scheme can be constructed from the quantum hardness of learning with errors (LWE).

2.2.4 Signature tokens

Definition 2.19 (Signature token)**.** *A signature token scheme consists of algorithms* $(\mathsf{TokGen}, \mathsf{TokSign}, \mathsf{TokVer})$ *with the following syntax.*

- $\mathsf{TokGen}(1^\lambda) \to (\mathsf{vk}, |\mathsf{sk}\rangle)$*: The* TokGen *algorithm takes as input the security parameter* 1^λ *and outputs a classical verification key* vk *and a quantum signing key* $|\mathsf{sk}\rangle$*.*
- $\mathsf{TokSign}(b, |\mathsf{sk}\rangle) \to \sigma$*: The* TokSign *algorithm takes as input a bit* $b \in \{0, 1\}$ *and the signing key* $|\mathsf{sk}\rangle$*, and outputs a classical signature* σ*.*
- $\mathsf{TokVer}(\mathsf{vk}, b, \sigma) \to \{\top, \bot\}$*: The* TokVer *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 2.20. *A signature token scheme* $(\mathsf{TokGen}, \mathsf{TokSign}, \mathsf{TokVer})$ *is* correct *if for any* $b \in \{0, 1\}$*,*

$$\Pr\left[\mathsf{TokVer}(\mathsf{vk}, b, \sigma) = \top : \begin{array}{c}(\mathsf{vk}, |\mathsf{sk}\rangle) \leftarrow \mathsf{TokGen}(1^\lambda) \\ \sigma \leftarrow \mathsf{TokSign}(b, |\mathsf{sk}\rangle)\end{array}\right] = 1.$$

A signature token should satisfy the following definition of security. Note that we give the adversary oracle access to the verification functionality, and ask for exponential security.

Definition 2.21. *A signature token scheme* $(\mathsf{TokGen}, \mathsf{TokSign}, \mathsf{TokVer})$ *satisfies* unforgeability *if for any QPQ adversary* $\{\mathsf{A}_\lambda\}_{\lambda\in\mathbb{N}}$*,*

$$\Pr\left[\begin{array}{l}\mathsf{TokVer}(\mathsf{vk}, 0, \sigma_0) = \top \ \wedge \\ \mathsf{TokVer}(\mathsf{vk}, 1, \sigma_1) = \top\end{array} : \begin{array}{c}(\mathsf{vk}, |\mathsf{sk}\rangle) \leftarrow \mathsf{TokGen}(1^\lambda) \\ (\sigma_0, \sigma_1) \leftarrow \mathsf{A}_\lambda^{\mathsf{TokVer}[\mathsf{vk}]}(|\mathsf{sk}\rangle)\end{array}\right] = 2^{-\Omega(\lambda)},$$

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

Theorem 2.22 ([BS16])**.** *There exists a signature token scheme that satisfies Definition 2.20 and Definition 2.21.*

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

Definition 2.23. *A signature token scheme* $(\mathsf{TokGen}, \mathsf{TokSign}, \mathsf{TokVer})$ *satisfies* strong unforgeability *if for any QPQ adversary* $\{\mathsf{A}_\lambda\}_{\lambda\in\mathbb{N}}$,

$$\Pr\left[\begin{array}{l} (b_0,\sigma_0)\neq(b_1,\sigma_1)\ \wedge \\ \mathsf{TokVer}(\mathsf{vk},b_0,\sigma_0)=\top\ \wedge \\ \mathsf{TokVer}(\mathsf{vk},b_1,\sigma_1)=\top \end{array} : \begin{array}{r} (\mathsf{vk},|\mathsf{sk}\rangle)\leftarrow\mathsf{TokGen}(1^\lambda) \\ (b_0,\sigma_0,b_1,\sigma_1)\leftarrow\mathsf{A}_\lambda^{\mathsf{TokVer}[\mathsf{vk}]}(|\mathsf{sk}\rangle) \end{array}\right] = 2^{-\Omega(\lambda)},$$

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

Claim 2.24. *There exists a signature token scheme that satisfies Definition 2.20 and Definition 2.23.*

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 $a\neq b\in A^\perp\setminus\{0^n\}$. □

2.2.5 Puncturable pseudorandom functions

Definition 2.25 (Puncturable pseudorandom function)**.** *A puncturable pseudorandom function* $(\mathsf{PRF.Gen}, \mathsf{PRF.Puncture}, \mathsf{PRF.Eval})$ *consists of the following PPT algorithms.*

- $\mathsf{PRF.Gen}(1^\lambda) \to k$*: On input the security parameter, the key generation algorithm returns a key* k*.*
- $\mathsf{PRF.Puncture}(k, z) \to k_z$*: On input a key* k *and a point* z*, the puncturing algorithm returns the punctured key* k_z*.*
- $\mathsf{PRF.Eval}(k, x) \to y$*: On input a key* k *and a string* $x \in \{0,1\}^\lambda$*, the evaluation algorithm returns a string* $y \in \{0,1\}^\lambda$*.*

Correctness requires that evaluation using the punctured key agrees with evaluation using the non-punctured key on all but the punctured point.

Definition 2.26 (Correctness)**.** *A puncturable PRF* $(\mathsf{PRF.Gen}, \mathsf{PRF.Puncture}, \mathsf{PRF.Eval})$ *is correct if for all* $\lambda \in \mathbb{N}$*, all* $z \in \{0,1\}^\lambda$*, and all strings* $x \neq z$ *it holds that*

$$\Pr\left[\mathsf{PRF.Eval}(k, x) = \mathsf{PRF.Eval}(k_z, x)\right] = 1,$$

where $k \leftarrow \mathsf{PRF.Gen}(1^\lambda)$ *and* $k_z \leftarrow \mathsf{PRF.Puncture}(k, z)$*.*

Pseudorandomness requires that the evaluation of the PRF at any point z is computationally indistinguishable from random, even given the punctured key k_z.

Definition 2.27 (Pseudorandomness)**.** *A puncturable PRF* $(\mathsf{PRF.Gen}, \mathsf{PRF.Puncture}, \mathsf{PRF.Eval})$ *is pseudorandom if for all* $z \in \{0,1\}^\lambda$ *and QPT* $\{A_\lambda\}_{\lambda\in\mathbb{N}}$*, it holds that*

$$\left|\Pr\left[A_\lambda(k_z, y) = 1 : \begin{array}{r} k \leftarrow \mathsf{PRF.Gen}(1^\lambda) \\ k_z \leftarrow \mathsf{PRF.Puncture}(k, z) \\ y \leftarrow \mathsf{PRF.Eval}(k, z) \end{array}\right] - \Pr\left[A_\lambda(k_z, u) = 1 : \begin{array}{r} k \leftarrow \mathsf{PRF.Gen}(1^\lambda) \\ k_z \leftarrow \mathsf{PRF.Puncture}(k, z) \\ u \leftarrow \{0,1\}^\lambda \end{array}\right]\right| = \mathrm{negl}(\lambda).$$

3 Obfuscation of Null Quantum Circuits

In Section 3.1, we formalize the approach outlined Section 1.3 for obfuscating null quantum circuits. Then, in Section 3.2, we present several applications, including witness encryption for QMA and non-interactive zero-knowledge for QMA. For further applications, see [BM22].

3.1 Construction

We begin by defining the notion of a classical-verifier generalized sigma protocol for QMA.

Definition 3.1 (Classical-verifier generalized sigma protocol for QMA)**.** *A classical-verifier generalized sigma protocol for QMA consists of algorithms* $(\mathsf{Gen}, \mathsf{Prove}_1, \mathsf{Prove}_2, \mathsf{Ver})$ *with the following syntax.*

- $\mathsf{Gen}(1^\lambda, Q) \to (\mathsf{pp}, \mathsf{sp})$*: The PPT parameter generation algorithm takes as input the security parameter* 1^λ *and a quantum circuit* Q *with quantum input and one bit of classical output, and outputs public parameters* pp *and secret parameters* sp*.*
- $\mathsf{Prove}_1(\mathsf{pp}, |\psi\rangle) \to (\pi_1, |\mathsf{st}\rangle)$*: The QPT first prover algorithm takes as input the public parameters* pp *and a quantum state* $|\psi\rangle$*, and outputs a classical string* π_1 *and a quantum state* $|\mathsf{st}\rangle$*.*
- $\mathsf{Prove}_2(|\mathsf{st}\rangle, r) \to \pi_2$*: The QPT second prover algorithm takes as input a quantum state* $|\mathsf{st}\rangle$ *and a classical string* r*, and outputs a classical string* π_2*.*
- $\mathsf{Ver}(\mathsf{sp}, \pi_1, r, \pi_2) \to b$*: The PPT verification algorithm takes as input the public parameters* pp*, and classical strings* π_1, r, π_2*, and outputs a bit* b *indicating acceptance or rejection.*

The protocol satisfies ***completeness*** *if there exists a polynomial* $k(\lambda)$ *such that for any polynomial-size family of quantum circuits* $\{Q_\lambda\}_{\lambda\in\mathbb{N}}$ *and inputs* $\{|\psi_\lambda\rangle\}_{\lambda\in\mathbb{N}}$ *such that* $\Pr[Q_\lambda(|\psi_\lambda\rangle) = 1] = 1 - \mathrm{negl}(\lambda)$*, it holds that*

$$\Pr\left[\mathsf{Ver}(\mathsf{sp}, \pi_1, r, \pi_2) = 1 : \begin{array}{r}(\mathsf{pp}, \mathsf{sp}) \leftarrow \mathsf{Gen}(1^\lambda, Q_\lambda)\\ (\pi_1, |\mathsf{st}\rangle) \leftarrow \mathsf{Prove}_1\left(\mathsf{pp}, |\psi_\lambda\rangle^{\otimes k(\lambda)}\right)\\ r \leftarrow \{0,1\}^\lambda\\ \pi_2 \leftarrow \mathsf{Prove}_2(|\mathsf{st}\rangle, r)\end{array}\right] = 1 - \mathrm{negl}(\lambda).$$

The protocol satisfies ***soundness*** *if for any polynomial-size family of quantum circuits* $\{Q_\lambda\}_{\lambda\in\mathbb{N}}$ *such that for all inputs* $\{|\psi_\lambda\rangle\}_{\lambda\in\mathbb{N}}$*,* $\Pr[Q_\lambda(|\psi_\lambda\rangle) = 1] = \mathrm{negl}(\lambda)$*, and any QPT adversary* $\{A_{1,\lambda}, A_{2,\lambda}\}_{\lambda\in\mathbb{N}}$*, it holds that*

$$\Pr\left[\mathsf{Ver}(\mathsf{sp},\pi_1,r,\pi_2)=1 : \begin{array}{r}(\mathsf{pp},\mathsf{sp})\leftarrow \mathsf{Gen}(1^\lambda,Q_\lambda)\\ (\pi_1,|\mathsf{st}\rangle)\leftarrow A_1(\mathsf{pp})\\ r\leftarrow\{0,1\}^\lambda\\ \pi_2\leftarrow A_2(|\mathsf{st}\rangle,r)\end{array}\right]=\mathrm{negl}(\lambda).$$

*The protocol satisfies **blindness** if for any two polynomial-size families of quantum ciruits $\{Q_{0,\lambda}\}_{\lambda\in\mathbb{N}}$ and $\{Q_{1,\lambda}\}_{\lambda\in\mathbb{N}}$, and any QPT adversary $\{A_\lambda\}_{\lambda\in\mathbb{N}}$, it holds that*

$$\left|\Pr\left[1\leftarrow A_\lambda(\mathsf{pp}):(\mathsf{pp},\mathsf{sp})\leftarrow\mathsf{Gen}(1^\lambda,Q_{0,\lambda})\right]-\Pr\left[1\leftarrow A_\lambda(\mathsf{pp}):(\mathsf{pp},\mathsf{sp})\leftarrow\mathsf{Gen}(1^\lambda,Q_{1,\lambda})\right]\right|=\mathrm{negl}(\lambda).$$

Imported Theorem 3.2 ([Mah18b, ACGH20]). *Assuming a dTCF with an adaptive hardcore bit (Definition 2.15), there exists a classical-verifier generalized sigma protocol for QMA that satisfies completeness and soundness.*

Next, we show a simple transformation based on quantum fully-homomorphic encryption that generically adds blindness to any classical-verifier generalized sigma protocol for QMA.

Claim 3.3. *Assuming QFHE (Definition 2.16) and a classical-verifier generalized sigma protocol for QMA that satisfies completeness and soundness, there exists a classical-verifier generalized sigma protocol for QMA that satisfies completeness, soundness, and blindness.*

Proof. The complete and sound protocol $(\mathsf{Gen},\mathsf{Prove}_1,\mathsf{Prove}_2,\mathsf{Verify})$ can be turned into a blind protocol $(\mathsf{Gen}',\mathsf{Prove}'_1,\mathsf{Prove}'_2,\mathsf{Verify}')$ as follows. Gen' will run Gen, sample a random QFHE key pair $(\mathsf{pk},\mathsf{sk})$, and output $\mathsf{pp}'\leftarrow\mathsf{Enc}(\mathsf{pk},\mathsf{pp})$ and $\mathsf{sp}':=(\mathsf{sp},\mathsf{sk})$. Prove'_1 will run Prove_1 under the QFHE to produce $(\pi'_1,|\mathsf{st}'\rangle):=\mathsf{Enc}(\mathsf{pk},\pi_1),\mathsf{Enc}(\mathsf{pk},|\mathsf{st}\rangle)$. Prove'_2, given r, will run Prove_2 under the QFHE to produce $\pi'_2:=\mathsf{Enc}(\mathsf{pk},\pi_2)$. Finally, the verification procedure $\mathsf{Ver}'(\mathsf{sp}',\pi'_1,r,\pi'_2)$ will compute $\pi_1:=\mathsf{Dec}(\mathsf{sk},\pi'_1)$, $\pi_2:=\mathsf{Dec}(\mathsf{sk},\pi'_2)$, and output $\mathsf{Verify}(\mathsf{sp},\pi_1,r,\pi_2)$.

Correctness of $(\mathsf{Gen}',\mathsf{Prove}'_1,\mathsf{Prove}'_2,\mathsf{Ver}')$ follows immediately from correctness of QFHE. Soundness follows by a reduction to the soundness of $(\mathsf{Gen},\mathsf{Prove}_1,\mathsf{Prove}_2,\mathsf{Ver})$, in which the reduction samples the $(\mathsf{pk},\mathsf{sk})$ key pair, encrypts pp received from its challenger, and decrypts each of $\mathsf{Enc}(\mathsf{pk},\pi_1)$ and $\mathsf{Enc}(\mathsf{pk},\pi_2)$ before forwarding them to its challenger. Blindness follows immediately from the semantic security of QFHE. □

Imported Theorem 3.4 ([DFMS19, ACGH20]). *Let $(\mathsf{Gen},\mathsf{Prove}_1,\mathsf{Prove}_2,\mathsf{Ver})$ be a classical-verifier generalized sigma protocol for QMA that satisfies soundness, and let $H:\{0,1\}^*\to\{0,1\}^\lambda$ be a (quantum-accessible) random oracle. Then for any polynomial-size family of quantum circuits $\{Q_\lambda\}_{\lambda\in\mathbb{N}}$ such that for all inputs $\{|\psi_\lambda\rangle\}_{\lambda\in\mathbb{N}}$, $\Pr[Q_\lambda(|\psi_\lambda\rangle)=1]=\mathrm{negl}(\lambda)$, and any oracle-aided QPT adversary $\{A_\lambda\}_{\lambda\in\mathbb{N}}$, it holds that*

$$\Pr\left[\mathsf{Ver}(\mathsf{sp},\pi_1,H(\pi_1),\pi_2)=1 : \begin{array}{r}(\mathsf{pp},\mathsf{sp})\leftarrow\mathsf{Gen}(1^\lambda,Q_\lambda)\\ (\pi_1,\pi_2)\leftarrow A^H_\lambda(\mathsf{pp})\end{array}\right]=\mathrm{negl}(\lambda).$$

Next, we present our construction of obfuscation for null quantum circuits (Definition 2.11). We make use of the following ingredients, and our construction is in the classical oracle model (see Remark 2.14)

- A pseudorandom function F_k secure against superposition-query attacks [Zha12].
- A classical-verifier generalized sigma protocol for QMA $(\mathsf{Gen}, \mathsf{Prove}_1, \mathsf{Prove}_2, \mathsf{Ver})$ that satisfies completeness, soundness, and blindness (Definition 3.1).

Our construction is given in Fig. 1.

Obfuscation for null quantum circuits

- $\mathsf{NObf}(1^\lambda, Q)$:
  - Sample $(\mathsf{pp}, \mathsf{sp}) \leftarrow \mathsf{Gen}(1^\lambda, Q)$.
  - Sample a PRF key $k \leftarrow \{0,1\}^\lambda$ and define the function $H(\cdot) = F(k, \cdot)$.
  - Let $V(\pi_1, \pi_2)$ be the function that computes and outputs $b = \mathsf{Ver}(\mathsf{sp}, \pi_1, H(\pi_1), \pi_2)$.
  - Output $\widetilde{Q} = \mathsf{pp}, O = (H, V)$.
- $\mathsf{NEval}^O\left(\widetilde{Q}, |\psi\rangle\right)$:
  - Compute $(\pi_1, |\mathsf{st}\rangle) \leftarrow \mathsf{Prove}_1(\mathsf{pp}, |\psi\rangle)$.
  - Compute $\pi_2 \leftarrow \mathsf{Prove}_2(|\mathsf{st}\rangle, H(\pi_1))$.
  - Output $V(\pi_1, \pi_2)$.

Figure 1: An obfuscator for null quantum circuits in the classical oracle model.

Theorem 3.5. *The scheme described in Fig. 1 is an obfuscator for null quantum circuits satisfying Definition 2.11.*

Proof. To argue correctness, we first switch H to a uniformly random function, which only introduces a negligible difference in the output of the evaluation procedure, by the security of the PRF. Then correctness for inputs that map to 1 follows from the completeness of the sigma protocol, and correctness for inputs that map to 0 follows from the soundness of the sigma protocol.

Next, we argue security. Fix any two null circuits Q_0, Q_1 (these are technically families of circuits, but we drop the indexing by λ to avoid clutter), and a QPT adversary A. Let q be an upper bound on the number of queries that A makes to its oracles. Consider the following sequence of hybrids.

- $\mathcal{H}_0$: Run $A^{H,V}(\mathsf{pp})$, where pp, H, V are sampled by the honest obfuscator $\mathsf{NObf}(1^\lambda, Q_0)$.

- $\mathcal{H}_1$: Switch H to a uniformly random function.
- $\mathcal{H}_{2,i}$ for i from $1, \ldots, q$: Switch the adversary's first i queries to V to be answered by a function that outputs 0 on all inputs.
- $\mathcal{H}_3$: Switch pp to be sampled as $(\mathsf{pp}, \mathsf{sp}) \leftarrow \mathsf{Gen}(1^\lambda, Q_1)$.
- $\mathcal{H}_{4,i}$ for i from $q, \ldots, 1$: Reverse the change from $\mathcal{H}_{2,i}$.
- $\mathcal{H}_5$: Reverse the change from $\mathcal{H}_1$. This results in $A^{H,V}(\mathsf{pp})$, where pp, H, V are sampled by the honest obfuscator $\mathsf{NObf}(1^\lambda, Q_1)$.

To complete the proof, we have the following sequence of claims.

- $|\Pr[\mathcal{H}_1 = 1] - \Pr[\mathcal{H}_0 = 1]| = \mathrm{negl}(\lambda)$. This follows directly from the superposition-query security of the PRF.
- For all $i \in [q]$, $|\Pr[\mathcal{H}_{2,i} = 1] - \Pr[\mathcal{H}_{2,i-1} = 1]| = \mathrm{negl}(\lambda)$, where $\mathcal{H}_{2,0} \coloneqq \mathcal{H}_1$. This follows from the soundness of the sigma protocol and Imported Theorem 3.4. Indeed, suppose there was a non-negligible difference between $\mathcal{H}_{2,i}$ and $\mathcal{H}_{2,i-1}$ for some $i \in [q]$. Then it must be the case that in $\mathcal{H}_{i-1}$,the adversary's i'th query to V has non-negligible amplitude on (π_1, π_2) such that $\mathsf{Ver}(\mathsf{sp}, \pi_1, H(\pi_1), \pi_2) = 1$. However, this would violate Imported Theorem 3.4, since the adversary's $i-1$ previous queries to V can be answered without the use of sp (in particular, by answering with the function that outputs 0 on all inputs).
- $|\Pr[\mathcal{H}_3 = 1] - \Pr[\mathcal{H}_{2,q} = 1]| = \mathrm{negl}(\lambda)$. This follows directly from the blindness of the sigma protocol, since sp is no longer needed to simulate the adversary's calls to V.
- For all $i \in [q]$, $|\Pr[\mathcal{H}_{4,i+1} = 1] - \Pr[\mathcal{H}_{4,i} = 1]| = \mathrm{negl}(\lambda)$, where $\mathcal{H}_{4,i+1} \coloneqq \mathcal{H}_3$. This follows from the soundness of the sigma protocol and Imported Theorem 3.4.
- $|\Pr[\mathcal{H}_{4,1} = 1] - \Pr[\mathcal{H}_5 = 1]| = \mathrm{negl}(\lambda)$. This follows directly from the superposition-query security of the PRF.

□

3.2 Applications

3.2.1 Witness encryption for QMA

A language $\mathcal{L} = (\mathcal{L}_{\mathsf{yes}}, \mathcal{L}_{\mathsf{no}})$ in QMA is defined by a tuple $(\mathcal{V}, p, \alpha, \beta)$, where p is a polynomial, $\mathcal{V} = \{V_\lambda\}_{\lambda \in \mathbb{N}}$ is a uniformly generated family of circuits such that for every λ, V_λ takes as input a string $x \in \{0,1\}^\lambda$ and a quantum state $|\psi\rangle$ on $p(\lambda)$ qubits and returns a single bit, and $\alpha, \beta : \mathbb{N} \to [0,1]$ are such that $\alpha(\lambda) - \beta(\lambda) \geq 1/p(\lambda)$. The language is then defined as follows.

- For all $x \in \mathcal{L}_{\mathsf{yes}}$ of length λ, there exists a quantum state $|\psi\rangle$ of size at most $p(\lambda)$ such that the probability that V_λ accepts $(x, |\psi\rangle)$ is at least $\alpha(\lambda)$. We denote the (possibly infinite) set of quantum witnesses that make V_λ accept x by $\mathcal{R}_{\mathcal{L}}(x)$.
- For all $x \in \mathcal{L}_{\mathsf{no}}$ of length λ, and all quantum states $|\psi\rangle$ of size at most $p(\lambda)$, it holds that V_λ accepts on input $(x, |\psi\rangle)$ with probability at most $\beta(\lambda)$.

We now recall the definition of witness encryption [GGSW13], and adapt it to the quantum setting. Note that we define encryption only with respect to classical messages. This is without loss of generality, since one can encrypt a quantum state with the quantum one-time pad [AMTDW00] and then use the witness encryption to encrypt the corresponding (classical) one-time pad keys.

Definition 3.6 (Witness encryption for QMA)**.** *Witness encryption* $(\mathsf{WE.Enc}, \mathsf{WE.Dec})$ *for a language* $\mathcal{L} \in \mathsf{QMA}$ *with relation* $\mathcal{R}_{\mathcal{L}}$ *consists of the following algorithms.*

- $\mathsf{WE.Enc}(1^\lambda, x, m) \to \mathsf{ct}$*: On input the security parameter* 1^λ*, a statement* x*, and a message* $m \in \{0,1\}$*, the QPT encryption algorithm returns a ciphertext* ct*.*
- $\mathsf{WE.Dec}(x, \mathsf{ct}, |\psi\rangle) \to m$*: On input a statement* x*, a ciphertext* ct*, and a quantum state* $|\psi\rangle$*, the QPT decryption algorithm returns a message* m*.*

We define correctness below.

Definition 3.7 (Correctness)**.** *A witness encryption* $(\mathsf{WE.Enc}, \mathsf{WE.Dec})$ *for a language* $\mathcal{L} \in \mathsf{QMA}$ *is correct if there exists a polynomial* $k(\lambda)$ *such that for any* $m \in \{0,1\}$*, all polynomial-length sequences of instances* $\{x_\lambda\}_{\lambda\in\mathbb{N}}$ *and witnesses* $\{|\psi_\lambda\rangle\}_{\lambda\in\mathbb{N}}$ *where each* $x_\lambda \in \mathcal{L}_{\mathsf{yes}}$ *and* $|\psi_\lambda\rangle \in \mathcal{R}_{\mathcal{L}}(x_\lambda)$*, it holds that*

$$\Pr\left[\mathsf{WE.Dec}\left(x_\lambda, \mathsf{WE.Enc}\left(1^\lambda, x_\lambda, m\right), |\psi_\lambda\rangle^{\otimes k(\lambda)}\right) = m\right] = 1 - \mathrm{negl}(\lambda).$$

Next, we define security.

Definition 3.8 (Security)**.** *A witness encryption* $(\mathsf{WE.Enc}, \mathsf{WE.Dec})$ *for a language* $\mathcal{L} \in \mathsf{QMA}$ *is secure if for all polynomial-length sequences of instances* $\{x_\lambda\}_{\lambda\in\mathbb{N}}$ *where each* $x_\lambda \in \mathcal{L}_{\mathsf{no}}$*, and any QPT* $\{A_\lambda\}_{\lambda\in\mathbb{N}}$*, it holds that*

$$\left|\Pr\left[A_\lambda(\mathsf{WE.Enc}(1^\lambda, x_\lambda, 0)) = 1\right] - \Pr\left[A_\lambda(\mathsf{WE.Enc}(1^\lambda, x_\lambda, 1)) = 1\right]\right| = \mathrm{negl}(\lambda).$$

Lemma 3.9. *Assuming an obfuscator for null quantum circuits (Definition 2.11) where the obfuscator algorithm is classical, there exists witness encryption for all languages in QMA where the encryption algorithm is classical.*

Proof. $\mathsf{WE.Enc}(1^\lambda, x, m)$ will simply output $\mathsf{ct} \leftarrow \mathsf{NObf}(1^\lambda, Q[x,m])$, where $Q[x,m]$ is the quantum circuit that takes as input $|\psi\rangle$, runs the QMA verification procedure for instance x and witness $|\psi\rangle$, and then outputs m if verification accepts, and otherwise outputs 0. $\mathsf{WE.Dec}(x, \mathsf{ct}, |\psi\rangle^{\otimes k})$ will take k copies of the witness $|\psi\rangle$ and run $\mathsf{NEval}(c, |\psi\rangle^{\otimes k})$ to produce either m or $\perp$. Assuming that the QMA language $\mathcal{L}$ is such that $\alpha(\lambda) = 1 - \mathrm{negl}(\lambda)$ and $\beta(\lambda) = \mathrm{negl}(\lambda)$ (which is without loss of generality by applying standard QMA amplification), correctness and security of the witness encryption scheme follow immediately from correctness and security of the obfuscator for null quantum circuits. In particular, for $x \in \mathcal{L}_{\mathsf{no}}$, $Q[x,m]$ is a null circuit, which implies that for any QPT $\{A_\lambda\}_{\lambda\in\mathbb{N}}$,

$$\left| \Pr\left[A_\lambda(\mathsf{NObf}(1^\lambda, Q[x,m])) = 1\right] - \Pr\left[A_\lambda(\mathsf{NObf}(1^\lambda, Q[x,0])) = 1\right] \right| = \mathrm{negl}(\lambda).$$

□

3.2.2 ZK-SNARG for QMA

In this section, we show how to construct a non-interactive zero-knowledge (NIZK) argument for QMA by utilizing witness encryption for QMA with a classical encryption algorithm. In fact, the resulting NIZK for QMA will also satisfy *succinctness*, yielding a ZK-SNARG (zero-knowledge succinct non-interactive argument) for QMA.

We first recall the definition of NIZK for QMA.

Definition 3.10 (NIZK argument)**.** *A NIZK argument* $(\mathsf{NIZK.Setup}, \mathsf{NIZK.Prove}, \mathsf{NIZK.Verify})$ *for a language* $\mathcal{L} \in \mathsf{QMA}$ *with relation* $\mathcal{R}_\mathcal{L}$ *consists of the following efficient algorithms.*

- $\mathsf{NIZK.Setup}(1^\lambda) \to \mathsf{crs}$*: On input the security parameter* 1^λ*, the setup algorithm returns a common reference string* crs*.*
- $\mathsf{NIZK.Prove}\left(\mathsf{crs}, x, |\psi\rangle^{\otimes k(\lambda)}\right) \to \pi$*: On input a common reference string* crs*, a statement* x*, and* $k(\lambda)$ *copies of the witness* $|\psi\rangle$*, the proving algorithm returns a proof* π*.*
- $\mathsf{NIZK.Verify}(\mathsf{crs}, x, \pi) \to b$*: On input a common reference string* crs*, a statement* x*, and a proof* π*, the verification algorithm returns a bit* b *indicating acceptance or rejection.*

We define completeness below.

Definition 3.11 (Completeness)**.** *A NIZK argument* $(\mathsf{NIZK.Setup}, \mathsf{NIZK.Prove}, \mathsf{NIZK.Verify})$ *is complete if for all* $x \in \mathcal{L}_{\mathsf{yes}}$*, and all* $|\psi\rangle \in \mathcal{R}_\mathcal{L}(x)$ *it holds that*

$$\Pr\left[\mathsf{NIZK.Verify}\left(\mathsf{crs}, x, \mathsf{NIZK.Prove}\left(\mathsf{crs}, x, |\psi\rangle^{\otimes k(\lambda)}\right)\right) = 1\right] = 1 - \mathrm{negl}(\lambda),$$

where $\mathsf{crs} \leftarrow \mathsf{NIZK.Setup}(1^\lambda)$.

Next, we define (non-adaptive) computational soundness.

Definition 3.12 (Computational soundness)**.** *A NIZK argument* $(\mathsf{NIZK.Setup}, \mathsf{NIZK.Prove}, \mathsf{NIZK.Verify})$ *is computationally sound if for all QPT* $\{A_\lambda\}_{\lambda\in\mathbb{N}}$ *and all* $x^* \in \mathcal{L}_{\mathsf{no}}$*, it holds that*

$$\Pr\left[\mathsf{NIZK.Verify}(\mathsf{crs}, x^*, A_\lambda(\mathsf{crs}, x^*)) = 1\right] = \mathrm{negl}(\lambda)$$

where $\mathsf{crs} \leftarrow \mathsf{NIZK.Setup}(1^\lambda)$.

Next, we define statistical zero-knowledge.

Definition 3.13 (Statistical zero-knowledge)**.** *A NIZK argument* $(\mathsf{NIZK.Setup}, \mathsf{NIZK.Prove}, \mathsf{NIZK.Verify})$ *is statistically zero-knowledge if there exists a QPT simulator* Sim *such that for any statement* $x \in \mathcal{L}_{\mathsf{yes}}$*, any witness* $|\psi\rangle \in \mathcal{R}_{\mathcal{L}}(x)$*, and any random coins* $r \in \{0,1\}^\lambda$*, it holds that*

$$\mathsf{Sim}(1^\lambda, x, r) \approx_{\mathrm{negl}(\lambda)} \mathsf{NIZK.Prove}\left(\mathsf{crs}, x, |\psi\rangle^{\otimes k(\lambda)}\right),$$

where $\mathsf{crs} \coloneqq \mathsf{NIZK.Setup}(1^\lambda; r)$.

Finally, we define succinctness.

Definition 3.14 (Succintness)**.** *A NIZK argument* $(\mathsf{NIZK.Setup}, \mathsf{NIZK.Prove}, \mathsf{NIZK.Verify})$ *is succinct if there is a fixed polynomial* $p(\lambda)$ *such that for any language* $\mathcal{L} \in \mathsf{QMA}$*, the size of the proof* π *is at most* $p(\lambda)$*.*

Next, we describe a succinct NIZK argument system (ZK-SNARG) for any language $\mathcal{L} \in \mathsf{QMA}$ with relation $\mathcal{R}_{\mathcal{L}}$. We use the following ingredients.

- A witness encryption scheme $(\mathsf{WE.Enc}, \mathsf{WE.Dec})$ for the language $\mathcal{L}$ with a classical encryption algorithm (Definition 3.6).
- A puncturable PRF $(\mathsf{PRF.Gen}, \mathsf{PRF.Puncture}, \mathsf{PRF.Eval})$ (Definition 2.25).
- A one-way function OWF.
- An indistinguishability obfuscator $\mathsf{i}\mathcal{O}$ for classical polynomial-size circuits (Definition 2.10)

Our ZK-SNARG $(\mathsf{NIZK.Setup}, \mathsf{NIZK.Prove}, \mathsf{NIZK.Verify})$ is presented in Fig. 2.

Theorem 3.15. *The scheme in Fig. 2 is a NIZK argument for QMA that satisfies completeness, computational soundness, statistical zero-knowledge, and succinctness.*

Proof. Completeness follows directly from the correctness of the building blocks. To show statistical zero-knowledge, we define the simulator to compute crs as in the $\mathsf{NIZK.Setup}$ algorithm and then set $\pi = \mathsf{PRF.Eval}(k_0, x)$. This distribution is identical to the one induced by the honest algorithms, except for when $\mathsf{WE.Dec}$ fails, which happens only with negligible probability. Succinctness is immediate by inspection. Thus, it remains to argue soundness.

The proof proceeds by defining a series of hybrid distributions for the computation of the crs that we argue to be computationally indistinguishable from each other. In the last hybrid, the probability that any prover can cause the verifier to accept some $x^* \in \mathcal{L}_{\mathsf{no}}$ will be negligible.

ZK-SNARG for QMA

- $\mathsf{NIZK.Setup}(1^\lambda)$:
  - Sample two keys $k_0 \leftarrow \mathsf{PRF.Gen}(1^\lambda)$ and $k_1 \leftarrow \mathsf{PRF.Gen}(1^\lambda)$.
  - Compute the obfuscation $\widetilde{\mathbf{P}} \leftarrow \mathsf{i}\mathcal{O}(1^\lambda, \mathbf{P})$ where $\mathbf{P}$ is the circuit that, on input some statement x, returns $\mathsf{WE.Enc}(1^\lambda, x, \mathsf{PRF.Eval}(k_0, x); \mathsf{PRF.Eval}(k_1, x))$. The circuit $\mathbf{P}$ is padded to the maximum size of $\mathbf{P}^*$ (defined in the proof of Theorem 3.15).
  - Compute the obfuscation $\widetilde{\mathbf{V}} \leftarrow \mathsf{i}\mathcal{O}(1^\lambda, \mathbf{V})$ where $\mathbf{V}$ is the circuit that, on input some statement x and a string y, returns 1 if and only if $\mathsf{OWF}(\mathsf{PRF.Eval}(k_0, x)) = \mathsf{OWF}(y)$. The circuit $\mathbf{V}$ is padded to the maximum size of $\mathbf{V}^*$ (defined in the proof of Theorem 3.15).
  - Return $\mathsf{crs} = (\widetilde{\mathbf{P}}, \widetilde{\mathbf{V}})$.
- $\mathsf{NIZK.Prove}\left(\mathsf{crs}, |\psi\rangle^{\otimes k(\lambda)}, x\right)$:
  - Compute $c = \widetilde{\mathbf{P}}(x)$.
  - Return $\pi = \mathsf{WE.Dec}\left(x, c, |\psi\rangle^{\otimes k(\lambda)}\right)$.
- $\mathsf{NIZK.Verify}(\mathsf{crs}, \pi, x)$:
  - Return $\widetilde{\mathbf{V}}(x, \pi)$.

Figure 2: A succinct NIZK argument for QMA.

- Hybrid $\mathcal{H}_0$: This is the original distribution where the crs is sampled by $\mathsf{crs} \leftarrow \mathsf{NIZK.Setup}(1^\lambda)$.
- Hybrid $\mathcal{H}_1$: In this hybrid we compute $\widetilde{\mathbf{P}} \leftarrow \mathsf{i}\mathcal{O}(1^\lambda, \mathbf{P}_1)$ where $\mathbf{P}_1$ is the circuit that on input some statement x, checks whether $x = x^*$. If that is the case, then it returns the ciphertext

  $$\mathsf{ct} = \mathsf{WE.Enc}(1^\lambda, x^*, \mathsf{PRF.Eval}(k_0, x^*); \mathsf{PRF.Eval}(k_1, x^*)).$$

  Otherwise compute $\mathsf{ct} = \mathsf{WE.Enc}(1^\lambda, x, \mathsf{PRF.Eval}(k_0, x); \mathsf{PRF.Eval}(k_{1,x^*}, x))$, where $k_{1,x^*} \leftarrow \mathsf{PRF.Puncture}(k_1, x^*)$.

  Note that the circuits $\mathbf{P}$ and $\mathbf{P}_1$ have different representations but are functionally equivalent. Thus, $\mathcal{H}_0$ and $\mathcal{H}_1$ are computationally indistinguishable by the security of the obfuscator $\mathsf{i}\mathcal{O}$.
- Hybrid $\mathcal{H}_2$: In this hybrid we compute $\widetilde{\mathbf{P}} \leftarrow \mathsf{i}\mathcal{O}(1^\lambda, \mathbf{P}_2)$ where $\mathbf{P}_2$ is defined as $\mathbf{P}_1$ except that if $x = x^*$, then it returns the ciphertext

  $$\mathsf{ct} = \mathsf{WE.Enc}(1^\lambda, x^*, \mathsf{PRF.Eval}(k_0, x^*); u)$$

  where $u \leftarrow \{0,1\}^\lambda$.

Indistinguishability from $\mathcal{H}_1$ follows from the pseudorandomness of the puncturable PRF.

- Hybrid $\mathcal{H}_3$: Here we compute $\widetilde{\mathbf{P}} \leftarrow \mathsf{i}\mathcal{O}(1^\lambda, \mathbf{P}_3)$ where $\mathbf{P}_3$ is defined as $\mathbf{P}_2$ except that if $x \neq x^*$, then it returns the ciphertext

$$\mathsf{ct} = \mathsf{WE.Enc}(1^\lambda, x, \mathsf{PRF.Eval}(k_{0,x^*}, x); \mathsf{PRF.Eval}(k_{1,x^*}, x))$$

  where $k_{0,x^*} \leftarrow \mathsf{PRF.Puncture}(k_0, x^*)$.

  By the correctness of the puncturable PRF, the two circuits are functionally identical and therefore indistinguishability from $\mathcal{H}_2$ follows from the security of $\mathsf{i}\mathcal{O}$.

- Hybrid $\mathcal{H}_4$: In this hybrid we compute $\widetilde{\mathbf{P}} \leftarrow \mathsf{i}\mathcal{O}(1^\lambda, \mathbf{P}^*)$ where $\mathbf{P}^*$ is defined as $\mathbf{P}_3$ except that if $x = x^*$, then it returns the ciphertext

$$\mathsf{ct} = \mathsf{WE.Enc}(1^\lambda, x^*, 0^\lambda; u)$$

  where $u \leftarrow \{0,1\}^\lambda$.

  Recall that $x^* \in \mathcal{L}_{\mathsf{no}}$ and thus indistinguishability between $\mathcal{H}_3$ and $\mathcal{H}_4$ follows from the security of the witness encryption scheme.

- Hybrid $\mathcal{H}_5$: We now compute $\widetilde{\mathbf{V}} \leftarrow \mathsf{i}\mathcal{O}(1^\lambda, \mathbf{V}_1)$ where $\mathbf{V}_1$ is the circuit that, on input a pair of strings (x, y) checks whether $x = x^*$. If this is the case, then it returns 1 if $\mathsf{OWF}(\mathsf{PRF.Eval}(k_0, x^*)) = \mathsf{OWF}(y)$ and 0 otherwise. If $x \neq x^*$ it returns 1 if and only if $\mathsf{OWF}(\mathsf{PRF.Eval}(k_{0,x^*}, x)) = \mathsf{OWF}(y)$ where k_{0,x^*} is the punctured key.

  Observe that the circuits $\mathbf{V}$ and $\mathbf{V}_1$ are functionally equivalent and thus we can invoke the security of $\mathsf{i}\mathcal{O}$ to show indistinguishability from $\mathcal{H}_4$.

- Hybrid $\mathcal{H}_6$: In this hybrid we compute $\widetilde{\mathbf{V}} \leftarrow \mathsf{i}\mathcal{O}(1^\lambda, \mathbf{V}_2)$ where $\mathbf{V}_2$ is defined as $\mathbf{V}_1$ except for the case where $x = x^*$. In this case the circuit returns 1 if and only if $\mathsf{OWF}(r) = \mathsf{OWF}(y)$, where $r \leftarrow \{0,1\}^\lambda$.

  Indistinguishability from $\mathcal{H}_5$ follows from a reduction to the pseudorandomness of the puncturable PRF.

- Hybrid $\mathcal{H}_7$: In the final hybrid we compute $\widetilde{\mathbf{V}} \leftarrow \mathsf{i}\mathcal{O}(1^\lambda, \mathbf{V}^*)$ where $\mathbf{V}^*$ is defined as $\mathbf{V}_2$ except for the case where $x = x^*$. In this case the circuit returns 1 if and only if $R = \mathsf{OWF}(y)$, where $R = \mathsf{OWF}(r)$, i.e. the image of the one-way function is hardwired in the circuit.

  Since the two circuits are functionally equivalent, indistinguishability from $\mathcal{H}_6$ follows from the security of $\mathsf{i}\mathcal{O}$.

Observe that in $\mathcal{H}_7$, causing the verifier to accept a proof π for $x^* \in \mathcal{L}_{\mathsf{no}}$ requires one to output a valid preimage of $R = \mathsf{OWF}(r)$, where r is uniformly sampled. This is a contradiction to the one-wayness of OWF, and concludes the proof. □

3.2.3 Attribute-based encryption for BQP

We first recall the definition of attribute-based encryption (ABE). For convenience we consider the notion of *ciphertext-policy* ABE where messages are encrypted with respect to circuits and keys are issued for attribute strings. If the class of circuits supported by the scheme is large enough, then one can switch to the complementary notion (i.e. *key-policy* ABE) by encoding universal (quantum) circuits. We also consider without loss of generality an ABE that encrypts a single (classical) bit of information.

Definition 3.16 (Attribute-based encryption for BQP)**.** *An ABE scheme for BQP* $(\mathsf{ABE.Gen}, \mathsf{ABE.Enc}, \mathsf{ABE.KeyGen}, \mathsf{ABE.Dec})$ *consists of the following efficient algorithms.*

- $\mathsf{ABE.Gen}(1^\lambda, 1^\ell)$*: On input the security parameter* 1^λ *and the length* ℓ *of attributes, the parameter generation algorithm outputs a master public key* mpk *and a master secret key* msk*.*
- $\mathsf{ABE.Enc}(\mathsf{mpk}, Q, m)$*: On input the master public key* mpk*, a pseudo-deterministic quantum circuit* Q *with one bit of output, and a message* m*, the encryption algorithm outputs a ciphertext* ct_Q*.*
- $\mathsf{ABE.KeyGen}(\mathsf{msk}, x)$*: On input the master secret key* msk *and an attribute* x*, the key generation algorithm outputs a secret key* sk_x*.*
- $\mathsf{ABE.Dec}(\mathsf{sk}_x, \mathsf{ct}_Q)$*: On input a secret key* sk_x *and a ciphertext* ct_Q*, the decryption algorithms either outputs a message* m *or* $\perp$*.*

We always assume that the ciphertexts contain a description of the corresponding circuit Q and the keys contain a description of the corresponding attribute x. We now define correctness.

Definition 3.17 (Correctness)**.** *An ABE scheme* $(\mathsf{ABE.Gen}, \mathsf{ABE.Enc}, \mathsf{ABE.KeyGen}, \mathsf{ABE.Dec})$ *is correct if for any* $\lambda \in \mathbb{N}, \ell \in \mathbb{N}, m \in \{0,1\}, x \in \{0,1\}^\ell$*, and any quantum circuit* Q *on* ℓ *input bits such that* $\Pr[Q(x) = 1] = 1 - \mathrm{negl}(\lambda)$*, it holds that*

$$\Pr\left[\mathsf{ABE.Dec}\left(\mathsf{ABE.KeyGen}(\mathsf{msk}, x), \mathsf{ABE.Enc}(\mathsf{mpk}, Q, m)\right) = m\right] = 1 - \mathrm{negl}(\lambda),$$

where $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{ABE.Gen}(1^\lambda, 1^\ell)$*.*

Finally we define the notion of security for ABE. We consider the selective notion of security, where the quantum circuit associated with the challenge ciphertext is known ahead of time. It is well known that this can be generically upgraded to the stronger notion of adaptive security via complexity leveraging, i.e. at the cost of an exponential decrease in the quality of the reduction.

Definition 3.18 (Security)**.** *An ABE scheme* $(\mathsf{ABE.Gen}, \mathsf{ABE.Enc}, \mathsf{ABE.KeyGen}, \mathsf{ABE.Dec})$ *is secure if for all quantum circuits* Q^**, and all admissible QPT distinguishers* $\{A_\lambda\}_{\lambda\in\mathbb{N}}$*, it holds that*

$$\left|\Pr\left[b = A_\lambda(\mathsf{ct}_{Q^*}, \mathsf{mpk}; \rho_\lambda)^{\mathsf{ABE.KeyGen}(\mathsf{msk},\cdot)} : \begin{array}{r}(\mathsf{mpk},\mathsf{msk}) \leftarrow \mathsf{ABE.Gen}(1^\lambda, 1^\ell)\\ b \leftarrow \{0,1\}\\ \mathsf{ct}_{Q^*} \leftarrow \mathsf{ABE.Enc}(\mathsf{mpk}, Q^*, b)\end{array}\right] - 1/2\right| = \mathrm{negl}(\lambda)$$

where $\{A_\lambda\}_{\lambda\in\mathbb{N}}$ *is admissible if each query* x *to* $\mathsf{ABE.KeyGen}(\mathsf{msk},\cdot)$ *is such that* $\Pr[Q^*(x) = 1] \leq \mathrm{negl}(\lambda)$.

We are now ready the present our construction of ABE for BQP. We use the following ingredients, all with sub-exponential security.

- A witness encryption scheme $(\mathsf{WE.Enc}, \mathsf{WE.Dec})$ for any language $\mathcal{L}$ in BQP with a classical encryption algorithm (Definition 3.6).
- A puncturable PRF $(\mathsf{PRF.Gen}, \mathsf{PRF.Puncture}, \mathsf{PRF.Eval})$ (Definition 2.25).
- A pseudorandom generator $\mathsf{PRG} : \{0,1\}^\lambda \to \{0,1\}^{\ell\cdot\lambda}$.
- An indistinguishability obfuscator $\mathsf{i}\mathcal{O}$ for classical polynomial-size circuits (Definition 2.10)

Our scheme $(\mathsf{ABE.Gen}, \mathsf{ABE.Enc}, \mathsf{ABE.KeyGen}, \mathsf{ABE.Dec})$ is described in Fig. 3.

Theorem 3.19. *The scheme in Fig. 3 is an attribute-based encryption scheme for BQP (Definition 3.16) that satisfies correctness and (selective) security.*

Proof. Correctness of the scheme follows immediately from the correctness of the building blocks.

To show security, we proceed by defining an exponentially long series of hybrids, iterating over all possible attributes $x \in \{0,1\}^\ell$. More specifically, starting from hybrid $\mathcal{H}_0$ (the original experiment with the bit b fixed to $b = 0$) we define, for each $i \in \{0,1\}^\ell$, a different sequence of hybrids and we argue about the indistinguishability of neighbouring distributions. As mentioned earlier, we assume all primitives we use are sub-exponentially secure, that is, there exists an $\epsilon > 0$ such that no efficient adversary can break the primitive with probability better than $2^{-\lambda^\epsilon}$. Thus, we can set the security parameter for each to be at least ℓ^c for some $c > 1/\epsilon$, ensuring that efficient adversaries have advantage $\mathrm{negl}(\lambda)/2^\ell$.

- Hybrid $\mathcal{H}_{i,0}$: Defined like $\mathcal{H}_0$, except that we change the way we compute the challenge ciphertext. We begin by computing a punctured key $r_i \leftarrow \mathsf{PRF.Puncture}(r, i)$. Then we compute $\widetilde{\mathbf{E}} \leftarrow \mathsf{i}\mathcal{O}(1^\lambda, \mathbf{E}_1)$, where $\mathbf{E}_1$ takes as input a pair (x, s) and does the following.

ABE for BQP

- $\mathsf{ABE.Gen}(1^\lambda, 1^\ell)$:
  - Sample a key $k \leftarrow \mathsf{PRF.Gen}(1^\lambda)$.
  - Compute the obfuscation $\widetilde{\mathbf{P}} \leftarrow \mathsf{i}\mathcal{O}(1^\lambda, \mathbf{P})$ where $\mathbf{P}$ is the circuit that, on input some attribute $x \in \{0,1\}^\ell$ and a string $s \in \{0,1\}^\lambda$, returns 1 if and only if $\mathsf{PRG}(s) = \mathsf{PRG}(\mathsf{PRF.Eval}(k, x))$. The circuit $\mathbf{P}[k]$ is padded to the maximum size of $\mathbf{P}^*$ (defined in the proof of Theorem 3.19).
  - Return $\mathsf{msk} = k$ and $\mathsf{mpk} = \widetilde{\mathbf{P}}$.
- $\mathsf{ABE.Enc}(\mathsf{mpk}, Q, m)$:
  - Sample a key $r \leftarrow \mathsf{PRF.Gen}(1^\lambda)$.
  - Compute the obfuscation $\widetilde{\mathbf{E}} \leftarrow \mathsf{i}\mathcal{O}(1^\lambda, \mathbf{E})$ where $\mathbf{E}$ is the circuit that, on input some attribute $x \in \{0,1\}^\ell$ and a string $s \in \{0,1\}^\lambda$, checks whether $\widetilde{\mathbf{P}}(x, s) = 1$ and returns $\mathsf{WE.Enc}(1^\lambda, (Q, x), m; \mathsf{PRF.Eval}(r, x))$ if this is the case. The circuit $\mathbf{E}[m, r]$ is padded to the maximum size of $\mathbf{E}^*$ (defined in the proof of Theorem 3.19).
  - Return $\widetilde{\mathbf{E}}$.
- $\mathsf{ABE.KeyGen}(\mathsf{msk}, x)$:
  - Return $\mathsf{PRF.Eval}(k, x)$.
- $\mathsf{ABE.Dec}(\mathsf{sk}_x, \mathsf{ct}_Q)$:
  - Parse ct_Q as $\widetilde{\mathbf{E}}$ and compute $c = \widetilde{\mathbf{E}}(x, \mathsf{sk}_x)$.
  - Return $\mathsf{WE.Dec}((Q, x), c)$.[a]

[a] Note that WE.Dec does not need to take a third input (the witness) since the statement is in BQP.

Figure 3: An attribute-based encryption scheme for BQP

- If $x < i$: Check whether $\widetilde{\mathbf{P}}(x, s) = 1$ and return
$$\mathsf{WE.Enc}(1^\lambda, (Q, x), 1; \mathsf{PRF.Eval}(r_i, x))$$
if this is the case.
- If $x = i$: Check whether $\widetilde{\mathbf{P}}(x, s) = 1$ and return
$$\mathsf{WE.Enc}(1^\lambda, (Q, x), 0; \mathsf{PRF.Eval}(r, x))$$
if this is the case.
- If $x > i$: Check whether $\widetilde{\mathbf{P}}(x, s) = 1$ and return
$$\mathsf{WE.Enc}(1^\lambda, (Q, x), 0; \mathsf{PRF.Eval}(r_i, x))$$
if this is the case.

Note that, by the correctness of the puncturable PRF, the circuits $\mathbf{E}$ and $\mathbf{E}_1$ are functionally equivalent and therefore indistinguishability follows from the security of the classical obfuscator $\mathsf{i}\mathcal{O}$.

- Hybrid $\mathcal{H}_{i,1}$: Defined like the previous hybrid, except that we compute $\widetilde{\mathbf{E}} \leftarrow \mathsf{i}\mathcal{O}(1^\lambda, \mathbf{E}_2)$, where $\mathbf{E}_2$ takes as input a pair (x, s) and does the following.
  - If $x < i$: Same as $\mathbf{E}_1$.
  - If $x = i$: Check whether $\widetilde{\mathbf{P}}(x, s) = 1$ and return $\mathsf{WE.Enc}(1^\lambda, (Q, x), 0; \tilde{r})$ if this is the case, where $\tilde{r} \leftarrow \{0,1\}^\lambda$.
  - If $x > i$: Same as $\mathbf{E}_1$.

  Note that the two hybrids differ only in the definition of $\tilde{r}$, which is uniformly sampled in $\mathcal{H}_{i,1}$ and computed according to the puncturable PRF in $\mathcal{H}_{i,0}$. By the indistinguishability of the puncturable PRF, we have that the two distributions are computationally close.

- Hybrid $\mathcal{H}_{i,2}$: In this hybrid we check whether $Q^*(i) = 0$. If this is not the case, then we proceed as before. Otherwise, we compute $\widetilde{\mathbf{E}} \leftarrow \mathsf{i}\mathcal{O}(1^\lambda, \mathbf{E}_3)$, where $\mathbf{E}_3$ takes as input a pair (x, s) and does the following.
  - If $x < i$: Same as $\mathbf{E}_2$.
  - If $x = i$: Check whether $\widetilde{\mathbf{P}}(x, s) = 1$ and return $\mathsf{WE.Enc}(1^\lambda, (Q, x), 1; \tilde{r})$ if this is the case, where $\tilde{r} \leftarrow \{0,1\}^\lambda$.
  - If $x > i$: Same as $\mathbf{E}_2$.

  Note that we change the view of the adversary only if $Q^*(i) = 0$, which implies that the statement (Q, i) is false. Thus indistinguishability follows from the security of the witness encryption scheme.

- Hybrid $\mathcal{H}_{i,3}$: This is defined as the previous one, except that we compute a punctured key $k_i \leftarrow \mathsf{PRF.Puncture}(k, i)$ and we modify the public parameters as follows. We obfuscate $\widetilde{\mathbf{P}} \leftarrow \mathsf{i}\mathcal{O}(1^\lambda, \mathbf{P}_1)$ where $\mathbf{P}_1$ is the circuit that, on input some attribute $x \in \{0,1\}^\ell$ and a string $s \in \{0,1\}^\lambda$, does the following.
  - If $x \neq i$: Return 1 if and only if $\mathsf{PRG}(s) = \mathsf{PRG}(\mathsf{PRF.Eval}(k_i, x))$.
  - If $x = i$: Return 1 if and only if $\mathsf{PRG}(s) = \mathsf{PRG}(\mathsf{PRF.Eval}(k, i))$.

  By the perfect correctness of the puncturable PRF, the two circuits are functionally equivalent and therefore the indistinguishability follows from the security of the obfuscator $\mathsf{i}\mathcal{O}$.

- Hybrid $\mathcal{H}_{i,4}$: In this hybrid we compute $\widetilde{\mathbf{P}} \leftarrow \mathsf{i}\mathcal{O}(1^\lambda, \mathbf{P}_2)$ where $\mathbf{P}_2$ is the circuit that, on input some attribute $x \in \{0,1\}^\ell$ and a string $s \in \{0,1\}^\lambda$, does the following.

- If $x \neq i$: Same as $\mathbf{P}_1$.
- If $x = i$: Return 1 if and only if $\mathsf{PRG}(s) = \mathsf{PRG}(\tilde{k})$, where $\tilde{k} \leftarrow \{0,1\}^\lambda$.

Additionally, we answer the query of the adversary to the key generation oracle with $\tilde{k}$, if queried on attribute i.

Note that this hybrid is identical to the previous one, except that $\tilde{k}$ is sampled uniformly. By the security of the puncturable PRF, the two hybrids are computationally indistinguishable.

- Hybrid $\mathcal{H}_{i,5}$: Before sampling the public parameters, we check whether $Q^*(i) = 1$. If this is not the case, then we proceed as before. Otherwise we obfuscate $\widetilde{\mathbf{P}} \leftarrow \mathsf{i}\mathcal{O}(1^\lambda, \mathbf{P}_3)$ where $\mathbf{P}_3$ is the circuit that, on input some attribute $x \in \{0,1\}^\ell$ and a string $s \in \{0,1\}^\lambda$, does the following.
  - If $x \neq i$: Same as $\mathbf{P}_2$.
  - If $x = i$: Return 1 if and only if $\mathsf{PRG}(s) = K$, where $K \leftarrow \{0,1\}^{\lambda \cdot \ell}$.

  Note that if $Q^*(i) \neq 1$, then the distribution induced by this hybrid is identical to the previous one, so we only consider the case where $Q^*(i) = 1$. Observe that an admissible adversary never queries the key generation oracle on i. Thus, the key $\tilde{k}$ is not present in the view of the distinguisher. Indistinguishability follows from the pseudorandomness of PRG.

- Hybrid $\mathcal{H}_{i,6}$: Here we again check whether $Q^*(i) = 1$. If this is not the case, then we proceed as before. Otherwise we compute $\widetilde{\mathbf{P}} \leftarrow \mathsf{i}\mathcal{O}(1^\lambda, \mathbf{P}^*)$ where $\mathbf{P}^*$ is the circuit that, on input some attribute $x \in \{0,1\}^\ell$ and a string $s \in \{0,1\}^\lambda$, does the following.
  - If $x \neq i$: Same as $\mathbf{P}_3$.
  - If $x = i$: Return 0.

  Note that the programs $\mathbf{P}_3$ and $\mathbf{P}^*$ are identical except if K falls within the range of PRG. Since this happens only with negligible probability, then the two hybrids are computationally indistinguishable by the security of the obfuscator $\mathsf{i}\mathcal{O}$.

- Hybrid $\mathcal{H}_{i,7}$: In this hybrid we check whether $Q^*(i) = 1$. If this is not the case, then we proceed as before. Otherwise, we compute the challenge ciphertext as $\widetilde{\mathbf{E}} \leftarrow \mathsf{i}\mathcal{O}(1^\lambda, \mathbf{E}^*)$, where $\mathbf{E}^*$ takes as input a pair (x, s) and does the following.
  - If $x < i$: Same as $\mathbf{E}_3$.
  - If $x = i$: Check whether $\widetilde{\mathbf{P}}(x, s) = 1$ and return $\mathsf{WE.Enc}(1^\lambda, (Q, x), 1; \tilde{r})$ if this is the case, where $\tilde{r} \leftarrow \{0,1\}^\lambda$.
  - If $x > i$: Same as $\mathbf{E}_3$.

Observe that at this point $\widetilde{\mathbf{P}}$ always returns 0 whenever queried on i, and thus the programs $\mathbf{E}_3$ and $\mathbf{E}^*$ are functionally equivalent. Indistinguishability follows from the security of $i\mathcal{O}$.

- Hybrid $\mathcal{H}_{i,8}$: We revert the change done in $\mathcal{H}_{i,6}$.
- Hybrid $\mathcal{H}_{i,9}$: We revert the change done in $\mathcal{H}_{i,5}$.
- Hybrid $\mathcal{H}_{i,10}$: We revert the change done in $\mathcal{H}_{i,4}$.
- Hybrid $\mathcal{H}_{i,11}$: We revert the change done in $\mathcal{H}_{i,3}$.
- Hybrid $\mathcal{H}_{i,12}$: We revert the change done in $\mathcal{H}_{i,1}$.
- Hybrid $\mathcal{H}_{i,13}$: We revert the change done in $\mathcal{H}_{i,0}$.

We denote by $\mathcal{H}_1$ the last hybrid of the sequence $\mathcal{H}_{2^\ell,13}$. Observe that such an hybrid is identical to the original experiment with the bit b fixed to $b = 1$. This concludes our proof. □

4 Obfuscation of Pseudo-Deterministic Quantum Circuits

In this section, we cover our construction of obfuscation for all pseudo-deterministic quantum circuits. We begin with a technical overview, filling in many of the details that were missing from the high-level discussion in Section 1.3. Next, in Section 4.2, we construct and prove the security of one of our central building blocks: a publicly-decodable X-measurable commitment. Then, in Section 4.3, we leverage these commitments in our construction of classical verification of partitioning circuits, which we use to obtain a publicly-verifiable quantum fully-homomorphic encryption scheme. Finally, we complete the construction of obfuscation in Section 4.4.

4.1 Technical overview

As discussed in Section 1.3, we reduce the task of obfuscating pseudo-deterministic quantum computation in the classical oracle model to the task of constructing a non-interactive publicly-verifiable classical verification protocol for quantum partitioning circuits.

To recap, we say that Q is a *quantum partitioning circuit* if there exists a predicate P such that $P(Q(\cdot))$ is pseudo-deterministic. The argument system we need has roughly the following syntax (see Section 4.3.1 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 quantum polynomial-time prover can produce an (x, π) such that $\mathsf{Ver}(\mathsf{pp}, Q, x, \pi) \to q$ and $\mathsf{Out}(q, P) \neq P(Q(x))$.

4.1.1 Private verification of quantum partitioning circuits

First, we describe a *privately-verifiable* scheme for classical verification of quantum partitioning circuits that follows readily from prior work [Mah18b, 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]. The state $|\psi_{Q,x}\rangle$ is prepared in such a way that, 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 the protocol from [Bar21],

the prover is instructed to prepare multiple copies of this history state, and the verifier chooses some subset 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 $b := \mathsf{Maj}\left(\{P(q_t)\}_t\right)$, which should be equal to $P(Q(x))$ with overwhelming probability.

Instantiating this protocol using only classical communication relies on [Mah18b]'s "measurement protocol" based on X-measurable commitments, which were introduced informally in Section 1.3. Recall that X-measurable commitments are non-interactive bit commitments that allow the committer to measure a committed state in the *Hadamard* basis, and have the following syntax.

- $\mathsf{Gen}(1^\lambda) \to (\mathsf{ck}, \mathsf{rk})$: Gen outputs a public commitment key ck and a private receiver's key rk.
- $\mathsf{Com}(\mathsf{ck}, \mathcal{B}) \to (\mathcal{B}, \mathcal{U}, c)$: 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.
- $\mathsf{Open}(\mathcal{B}, \mathcal{U}) \to u$: The standard basis opening algorithm performs a measurement on registers $(\mathcal{B}, \mathcal{U})$ to produce a classical string u.
- $\mathsf{Dec}(\mathsf{rk}, c, u) \to \{0, 1, \bot\}$: The standard basis decoding algorithm takes the receiver's key rk, a commitment c, an opening u, and either output a bit 0 or 1, or outputs $\bot$.
- $\mathsf{OpenX}(\mathcal{B}, \mathcal{U}) \to u$: The Hadamard basis opening algorithm performs a measurement on registers $(\mathcal{B}, \mathcal{U})$ to produce a classical string u.
- $\mathsf{DecX}(\mathsf{rk}, c, u) \to \{0, 1, \bot\}$: The Hadamard basis decoding algorithm takes the receiver's key rk, a commitment c, an opening u, and either outputs a bit 0 or 1, or outputs $\bot$.

In fact, [Mah18b]'s measurement protocol requires an extra *dual-mode* property of the X-measurable commitment. That is, $\mathsf{Gen}(1^\lambda, h)$ 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 $\mathsf{Dec}(\mathsf{rk}, c, u) = b$. This mode allows for the definition of an algorithm $\mathsf{Invert}(\mathsf{rk}, c) \to b$ that outputs the bit b such that there exists u with $\mathsf{Dec}(\mathsf{rk}, c, u) = b$ (or outputs $\bot$ if such a b does not exist). Importantly, the ck output on $h = 0$ vs $h = 1$ must be computationally indistinguishable.

Now, given these building blocks, we obtain the protocol described in Fig. 4.

In more detail, Fig. 4 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 which is determined by a random oracle H applied to the prover's X-measurable commitments c.

Classical verification of quantum partitioning circuits with one-time soundness

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

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

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

- 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 $(\mathsf{ck}_{i,j}, \mathsf{rk}_{i,j}) \leftarrow \mathsf{XMC.Gen}(1^\lambda, h_{i,j})$, and output

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

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

- 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 $\mathsf{XMC.Com}(\mathsf{ck}_{i,j}, \mathcal{B}_{i,j}) \to (\mathcal{B}_{i,j}, \mathcal{U}_{i,j}, c_{i,j})$, 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 $\mathsf{XMC.Open}(\mathcal{B}_{i,j}, \mathcal{U}_{i,j}) \to u_{i,j}$.
- For $i : T_i = 1$ and $j \in [\ell]$, apply $\mathsf{XMC.OpenX}(\mathcal{B}_{i,j}, \mathcal{U}_{i,j}) \to u_{i,j}$.
- Output $\pi := (c, u)$, where $u := (u_{1,1}, \ldots, u_{r,\ell})$.

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

- Parse $\pi = (c, u)$ and compute $T = H(c)$.
- For $i : T_i = 0$ and $j \in [\ell]$, check that $\mathsf{XMC.Dec}(\mathsf{rk}_{i,j}, c_{i,j}, u_{i,j}) \neq \bot$.
- For $i : T_i = 1$ and $j \in [\ell]$:
  - If $h_{i,j} = 0$, compute the bit $b_{i,j} := \mathsf{XMC.Invert}(\mathsf{rk}_{i,j}, c_{i,j})$, and abort if $\bot$.
  - If $h_{i,j} = 1$, compute the bit $b_{i,j} := \mathsf{XMC.DecX}(\mathsf{rk}_{i,j}, c_{i,j}, u_{i,j})$, 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 4: A non-interactive privately-verifiable protocol for classical verification of quantum partitioning circuits, based on prior work [Mah18b, CLLW22, Bar21]. The circuit Q and predicate P are such that $P(Q(\cdot))$ is a pseudo-deterministic circuit.

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 Open procedure, and the verifier simply checks that Dec 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].[18]

The one-time soundness of this protocol was proven in [Bar21] based on the soundness of the measurement protocol due to [Mah18b], which in turn relies on particular properties on the X-measurable commitment scheme.

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 $h = (h_1, \ldots, h_r)$, 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

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

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 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.

4.1.2 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 for 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 - an X-measurable commitment!

In Fig. 5, 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 pro-

tocol begins. We use an X-measurable commitment denoted PD-XMC to commit to the history states, where the PD stands for "publicly-decodable", a crucial property that we will discuss later in this overview.

We remark that correctness of this protocol relies on a couple of specific properties: (1) XMC.Com and PD-XMC.Com are both *classically controlled* on the register $\mathcal{B}$, so they commute with each other, and (2) XMC.Open (resp. XMC.OpenX) simply measures the register $\mathcal{B}$ in the standard (resp. Hadamard) basis[19] so the first bit of the string u can be computed instead by applying PD-XMC.Com to $\mathcal{B}$ followed by PD-XMC.Open and PD-XMC.Dec (resp. PD-XMC.OpenX and PD-XMC.DecX).

Now, our next goal will be to obtain reusable soundness for any fixed instance $(x, \widetilde{c})$. That is, we give the prover oracle access to $\mathsf{Ver}[\mathsf{sp}, \widetilde{\mathsf{rk}}, (x, \widetilde{c})](\cdot)$ where $\widetilde{\mathsf{rk}}$ 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 $b \neq P(Q(x))$.

Binding. Following the classical intuition, for the purpose of analyzing the scheme 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 $\perp$). 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 $\perp$). 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 $\perp$.

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 PD-XMC commitment scheme in order to guarantee that the prover cannot meaningfully "mix" these two strategies.

As discussed above, X-measurable commitments only 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": $P(q_t) = P(Q(x))$.

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

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 $(\widetilde{\mathsf{ck}}_{i,j}, \widetilde{\mathsf{rk}}_{i,j}) \leftarrow \mathsf{PD\text{-}XMC.Gen}(1^\lambda)$, outputs $\widetilde{\mathsf{ck}} := \{\widetilde{\mathsf{ck}}_{i,j}\}_{i,j}$, and keeps $\widetilde{\mathsf{rk}} := \{\widetilde{\mathsf{rk}}_{i,j}\}_{i,j}$ 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 $\mathsf{PD\text{-}XMC.Com}(\widetilde{\mathsf{ck}}_{i,j}, \mathcal{B}_{i,j}) \to (\mathcal{B}_{i,j}, \widetilde{\mathcal{U}}_{i,j}, \widetilde{c}_{i,j})$. 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 $\mathsf{pp} = \{\mathsf{ck}_{i,j}\}_{i,j}, \mathsf{sp} = (\{h_i, S_i\}_i, \{\mathsf{rk}_{i,j}\}_{i,j})$ as in Fig. 4.

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

- For $i \in [r], j \in [\ell]$, apply $\mathsf{XMC.Com}(\mathsf{ck}_{i,j}, \mathcal{B}_{i,j}) \to (\mathcal{B}_{i,j}, \mathcal{U}_{i,j}, c_{i,j})$, 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 $\mathsf{PD\text{-}XMC.Open}(\mathcal{B}_{i,j}, \widetilde{\mathcal{U}}_{i,j}) \to \widetilde{u}_{i,j}$ followed by $\mathsf{XMC.Open}(\mathcal{B}_{i,j}, \mathcal{U}_{i,j}) \to u_{i,j}$. Let $u'_{i,j}$ be $u_{i,j}$ with the first bit removed.
- For $i : T_i = 1$ and $j \in [\ell]$, apply $\mathsf{PD\text{-}XMC.OpenX}(\mathcal{B}_{i,j}, \widetilde{\mathcal{U}}_{i,j}) \to \widetilde{u}_{i,j}$ followed by $\mathsf{XMC.OpenX}(\mathcal{B}_{i,j}, \mathcal{U}_{i,j}) \to u_{i,j}$. 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 $u := (u'_{1,1}, \ldots, u'_{r,\ell})$.

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

- Parse $\pi = (c, \widetilde{u}, u)$ and compute $T = H(c)$.
- For $i : T_i = 0$ and $j \in [\ell]$, compute $b'_{i,j} := \mathsf{PD\text{-}XMC.Dec}(\widetilde{\mathsf{rk}}_{i,j}, \widetilde{c}_{i,j}, \widetilde{u}_{i,j})$ and check that $\mathsf{XMC.Dec}(\mathsf{rk}_{i,j}, c_{i,j}, (b'_{i,j}, u'_{i,j})) \neq \bot$.
- For $i : T_i = 1$ and $j \in [\ell]$:
  - If $h_{i,j} = 0$, compute the bit $b_{i,j} := \mathsf{XMC.Invert}(\mathsf{rk}_{i,j}, c_{i,j})$, and abort if $\bot$.
  - If $h_{i,j} = 1$, compute $b'_{i,j} := \mathsf{PD\text{-}XMC.DecX}(\widetilde{\mathsf{rk}}_{i,j}, \widetilde{c}_{i,j}, \widetilde{u}_{i,j})$, followed by the bit $b_{i,j} := \mathsf{XMC.DecX}(\mathsf{rk}_{i,j}, c_{i,j}, (b'_{i,j}, u'_{i,j}))$, 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 5: A protocol for classical verification of quantum partitioning circuits that is reusably sound for each fixed instance $(x, \widetilde{c})$.

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. 5, none of these positions are even opened by PD-XMC.Open (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 XMC 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 PD-XMC, which can be reduced (see Section 4.2.1 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 $\widetilde{\mathsf{rk}}$, a commitment $\widetilde{c}$, and a bit b, let

$$\Pi_{\widetilde{\mathsf{rk}},\widetilde{c},b} := \sum_{\widetilde{u}:\mathsf{Dec}(\widetilde{\mathsf{rk}},\widetilde{c},\widetilde{u})=b} |\widetilde{u}\rangle\langle\widetilde{u}|$$

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"[20] (modeled as a unitary), it holds that for any $b \in \{0,1\}$,

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

A couple of remarks:

- Looking at Fig. 5, we see that this binding property should hold *even* if the opener has oracle access to $\mathsf{Dec}(\widetilde{\mathsf{rk}},\widetilde{c},\cdot)$. In fact, in the known construction of XMC described above [BCM+18, Mah18b], this standard basis decoding is indeed public. Moreover, this definition of binding is implied by the dual-mode property of XMC, and thus our requirements for PD-XMC can *so far* be satisfied by the known construction of XMC.

[20] 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.

- Note that we only require binding on the standard basis positions, that is, (i, j) such that $h_{i,j} = 0$. Looking at Fig. 5, we see that the prover does *not* have access to $\mathsf{DecX}(\widetilde{\mathsf{rk}}_{i,j}, \widetilde{c}_{i,j}, \cdot)$ 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 X-measurable 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 PD-XMC. 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 $\mathsf{Ver}[\mathsf{sp}, \widetilde{\mathsf{rk}}, (x, \widetilde{c})]$ queries one by one to being answered with $\perp$. While the adversary may query $\mathsf{Ver}[\mathsf{sp}, \widetilde{\mathsf{rk}}, (x, \widetilde{c})]$ 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 PD-XMC, 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 $\mathsf{Ver}[\mathsf{sp}, \widetilde{\mathsf{rk}}, (x, \widetilde{c})]$, which completes the proof. For more details, see the discussion before the "soundness" part of the proof of Theorem 4.32.

4.1.3 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 PD-XMC parameters $\{\widetilde{\mathsf{ck}}_{i,j}\}_{i,j}$ along with a classical oracle that implements the following program $\mathsf{OGen}[k]$, which has a PRF key k hard-coded.

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

- Take an x and a commitment $\widetilde{c}$ as input, and compute $s \coloneqq \mathsf{PRF}_k((x, \widetilde{c}))$.
- Compute $(\mathsf{pp}, \mathsf{sp}) \coloneqq \mathsf{Gen}(1^\lambda; s)$ from Fig. 4 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 $\mathsf{PD\text{-}XMC.DecX}(\widetilde{\mathsf{rk}}_{i,j}, \cdot, \cdot)$ 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 $\mathsf{PD\text{-}XMC.DecX}(\widetilde{\mathsf{rk}}_{i,j}, \cdot, \cdot)$, 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 $\mathsf{PD\text{-}XMC.DecX}(\widetilde{\mathsf{rk}}_{i,j}, \cdot, \cdot)$ 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 $s \coloneqq \mathsf{PRF}_k((x, \widetilde{c}, \sigma))$, and otherwise abort.
- Compute $(\mathsf{pp}, \mathsf{sp}) \coloneqq \mathsf{Gen}(1^\lambda; s)$ from Fig. 4 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 $\mathsf{PD\text{-}XMC.DecX}(\mathsf{rk}_{i,j}, \cdot, \cdot)$ 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 $\mathsf{DecX}(\widetilde{\mathsf{rk}}_{i,j}, \cdot, \cdot)$ 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 PD-XMC, this first part of the adversary corresponds to the *commit* stage. Thus, we will need an X-measurable commitment scheme where the *committer* has access to both the Dec and DecX oracles, while the *opener* (necessarily) only has access to Dec.

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

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

Unfortunately, the known construction of X-measurable commitments [BCM+18, Mah18b] 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 X-measurable 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.

4.1.4 Publicly-decodable X-measurable commitments

First, we review why the X-measurable commitment based on claw-free hash functions [BCM+18, Mah18b] 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 $b := b' + d \cdot (x_0 + x_1)$[21] is a decoding of the Hadamard basis measurement of $\left|\psi\right\rangle$. Thus, if we define $S := \{0, x_0 + x_1\}$ to be a

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

one-dimensional subspace of $\mathbb{F}_2^n$, access to the DecX oracle provides the committer with a membership oracle for the subspace $S^\perp$. 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,[22] 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 $\mathsf{rk} = A$, 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

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

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

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

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),

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

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 Dec, 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]). 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]). 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 Section 4.5.2, 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 Section 4.5.2. This completes our proof of binding with public decodability.

4.2 Publicly-decodable X-measurable commitments

4.2.1 Definition

Recall that an X-measurable commitment augments the standard notion of a (non-interactive) bit commitment with an additional functionality property. It includes traditional key generation, commit, and decommit procedures, and must satisfy a notion of binding to a classical bit. However, when used to commit to a qubit $|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ in superposition, it supports the additional ability to open to a *Hadamard* basis measurement of $|\psi\rangle$.

The syntax of an X-measurable 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 allow the remainder of the commitment key to be a quantum state $|\mathsf{ck}\rangle$, but note that quantum commitment keys are not inherent to the definition of an X-measurable commitment.

Definition 4.1 (X-measurable commitment: Syntax)**.** *An X-measurable commitment consists of six algorithms* $(\mathsf{Gen}, \mathsf{Com}, \mathsf{Open}, \mathsf{Dec}, \mathsf{OpenX}, \mathsf{DecX})$ *with the following syntax.*

- $\mathsf{Gen}(1^\lambda) \to (\mathsf{rk}, |\mathsf{ck}\rangle, \mathsf{CK})$ *is a QPT algorithm that takes as input the security parameter* 1^λ *and outputs a classical receiver key* rk *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* $(\mathcal{B}, \mathcal{U})$*.*

- $\mathsf{Open}(\mathcal{B}, \mathcal{U}) \to u$ *is a QPT measurement on* $(\mathcal{B}, \mathcal{U})$ *that outputs a classical string* u*.*[23]

- $\mathsf{Dec}(\mathsf{rk}, c, u) \to \{0, 1, \bot\}$ *is a classical deterministic polynomial-time algorithm that takes as input the receiver key* rk*, a commitment* c*, and an opening* u*, and outputs either a bit* b *or a* $\bot$ *symbol.*

- $\mathsf{OpenX}(\mathcal{B}, \mathcal{U}) \to u$ *is a QPT measurement on* $(\mathcal{B}, \mathcal{U})$ *that outputs a classical string* u*.*[24]

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

Definition 4.2 (X-measurable commitment: Correctness)**.** *An X-measurable commitment* $(\mathsf{Gen}, \mathsf{Com}, \mathsf{Open}, \mathsf{Dec}, \mathsf{OpenX}, \mathsf{DecX})$ *is* correct *if for any single-qubit (potentially mixed) state on register* $\mathcal{B}$*, it holds that*

$$\mathsf{TV}\left(\mathsf{IdealZ}(\mathcal{B}), \mathsf{RealZ}(1^\lambda, \mathcal{B})\right) = \mathrm{negl}(\lambda), \quad \textit{and} \quad \mathsf{TV}\left(\mathsf{IdealX}(\mathcal{B}), \mathsf{RealX}(1^\lambda, \mathcal{B})\right) = \mathrm{negl}(\lambda),$$

where the distributions are defined as follows.

[23] Without loss of generality, we can assume that this procedure simply measures all registers in the standard basis (by including any potential pre-processing as part of the Com procedure). Thus, strictly speaking, we don't have to include an explicit Open algorithm in the syntax of an X-measurable commitment, though we do here for notational convenience later.

[24] One could define the "canonical" OpenX procedure to simply measure all registers in the Hadamard basis. Indeed, this is the case for all known approaches to building X-measurable commitments. However, we leave open the possibility that there exist interesting X-measurable commitments with more general OpenX procedures, and thus include an arbitrary OpenX procedure as part of the syntax.

- $\mathsf{IdealZ}(\mathcal{B})$ *measures* $\mathcal{B}$ *in the standard basis.*
- $\mathsf{IdealX}(\mathcal{B})$ *measures* $\mathcal{B}$ *in the Hadamard basis.*
- $\mathsf{RealZ}(1^\lambda, \mathcal{B})$ *samples* $(\mathsf{rk}, |\mathsf{ck}\rangle, \mathsf{CK}) \leftarrow \mathsf{Gen}(1^\lambda), (\mathcal{B}, \mathcal{U}, c) \leftarrow \mathsf{Com}^{\mathsf{CK}}(\mathcal{B}, |\mathsf{ck}\rangle), u \leftarrow \mathsf{Open}(\mathcal{B}, \mathcal{U})$*, and outputs* $\mathsf{Dec}(\mathsf{rk}, c, u)$*.*
- $\mathsf{RealX}(1^\lambda, \mathcal{B})$ *samples* $(\mathsf{rk}, |\mathsf{ck}\rangle, \mathsf{CK}) \leftarrow \mathsf{Gen}(1^\lambda), (\mathcal{B}, \mathcal{U}, c) \leftarrow \mathsf{Com}^{\mathsf{CK}}(\mathcal{B}, |\mathsf{ck}\rangle), u \leftarrow \mathsf{OpenX}(\mathcal{B}, \mathcal{U})$*, and outputs* $\mathsf{DecX}(\mathsf{rk}, c, u)$*.*

An X-measurable commitment that satisfies *binding with public decodability* allows the adversarial Committer to have oracle access to the receiver's functionalities $\mathsf{Dec}(\mathsf{rk}, \cdot, \cdot)$ and $\mathsf{DecX}(\mathsf{rk}, \cdot, \cdot)$. However, we crucially do not give the adversarial *Opener* access to $\mathsf{DecX}(\mathsf{rk}, \cdot, \cdot)$.

Definition 4.3 (X-measurable commitment: Single-bit binding with public decodability)**.** *An X-measurable commitment* $(\mathsf{Gen}, \mathsf{Com}, \mathsf{Open}, \mathsf{Dec}, \mathsf{OpenX}, \mathsf{DecX})$ *satisfies* single-bit binding with public decodability *if the following holds. Given* $\mathsf{rk}, c,$ *and* $b \in \{0,1\}$*, let*

$$\Pi_{\mathsf{rk},c,b} \coloneqq \sum_{u:\mathsf{Dec}(\mathsf{rk},c,u)=b} |u\rangle\langle u| \,.$$

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* $b \in \{0,1\}$*,*

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

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

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

Definition 4.4 (X-measurable commitment: String binding with public decodability)**.** *An X-measurable commitment* $(\mathsf{Gen}, \mathsf{Com}, \mathsf{Open}, \mathsf{Dec}, \mathsf{OpenX}, \mathsf{DecX})$ *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 *receiver keys* $\mathbf{rk} = (\mathsf{rk}_1, \ldots, \mathsf{rk}_m)$*,* m *strings* $\mathbf{c} = (c_1, \ldots, c_m)$*, and* $b \in \{0,1\}$*, define*

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

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,*

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

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

We prove the following lemma in Section 4.5.1.

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

4.2.2 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 an X-measurable commitment in Fig. 6.

Theorem 4.6. *The X-measurable commitment described in Fig. 6 satisfies* correctness *(Definition 4.2).*

Proof. 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 Tok is perfectly correct, 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.

Publicly-decodable X-measurable commitment

Parameters: Polynomial $n = n(\lambda) \geq \lambda$.
Ingredients: Signature token scheme $(\mathsf{Tok.Gen}, \mathsf{Tok.Sign}, \mathsf{Tok.Verify})$ (Section 2.2.4).

- $\mathsf{Gen}(1^\lambda)$: 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

$$\mathsf{rk} := (S, v, \mathsf{vk}), \quad |\mathsf{ck}\rangle := (|S + v\rangle, |\mathsf{sk}\rangle).$$

  Define CK to take as input (σ, s) for $s \in \{0,1\}^n$ and output $\perp$ if $\mathsf{Tok.Verify}(\mathsf{vk}, 0, \sigma) = \perp$, 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 $|s\rangle^{\mathcal{K}_0} |\sigma\rangle^{\mathcal{G}} \to (-1)^{\mathsf{CK}(\sigma,s)} |s\rangle^{\mathcal{K}_0} |\sigma\rangle^{\mathcal{G}}$.
  - 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 $\mathcal{U} := \mathcal{K}_0$.
- $\mathsf{Open}(\mathcal{B}, \mathcal{U})$: Measure all registers in the standard basis.
- $\mathsf{Dec}(\mathsf{rk}, c, u)$:
  - Parse $\mathsf{rk} = (S, v, \mathsf{vk})$ 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 $\perp$.
  - If $s \in (S + v)_b$, output b, and otherwise output $\perp$.
- $\mathsf{OpenX}(\mathcal{B}, \mathcal{U})$: Measure all registers in the Hadamard basis.
- $\mathsf{DecX}(\mathsf{rk}, c, u)$:
  - Parse $\mathsf{rk} = (S, v, \mathsf{vk})$ 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 $\perp$.
  - 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 $\perp$. 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 6: An X-measurable 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 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

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

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

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

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. □

4.2.3 Proof of binding

This section is dedicated to proving the following theorem.

Theorem 4.7. *Assuming that* Tok *satisfies unforgeability (Definition 2.21), the X-measurable commitment described in Fig. 6 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*

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

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

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

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

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

This follows from a standard hybrid argument, by reduction to the unforgeability of the signature token scheme. 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

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

and in hybrid $\mathcal{H}_i$, we switch the i'th-from-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 $\Pi_{\mathsf{rk},c,0}$ is the zero projector when c is not a valid signature on 1, and CK outputs $\perp$ whenever σ is not a valid signature on 0, we have that

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

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

Finally, it follows by Markov that

$$\begin{aligned}\Pr_{\substack{\mathsf{rk},|\mathsf{ck}\rangle,\mathsf{CK}\leftarrow\mathsf{Gen}(1^\lambda)\\(|\psi\rangle,c)\leftarrow\mathsf{C}^{\mathsf{CK},\mathsf{Dec}[\mathsf{rk}],\mathsf{DecX}[\mathsf{rk}]}(|\mathsf{ck}\rangle)}}&\left[\left\|\Pi_{\mathsf{rk},c,1}\mathsf{U}^{\mathsf{CK},\mathsf{Dec}[\mathsf{rk}]}\Pi_{\mathsf{rk},c,0}\left|\psi\right\rangle\right\|^2 - \left\|\Pi_{\mathsf{rk},c,1}\mathsf{U}^{O_\perp,\mathsf{Dec}[\mathsf{rk}]}\Pi_{\mathsf{rk},c,0}\left|\psi\right\rangle\right\|^2 \geq \epsilon(\lambda) - \frac{1}{2^\lambda}\right]\\ &\leq \frac{\mathrm{negl}(\lambda)}{\epsilon(\lambda) - 1/2^\lambda} = \mathrm{negl}(\lambda),\end{aligned}$$

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

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

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* $\mathrm{poly}(\lambda)$ *many oracle queries, it holds that*

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

Lemma 4.11. *Claim 4.10 implies Claim 4.8.*

Proof. First, we note that re-defining $\Pi_{\mathsf{rk},c,b}$ 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 $\mathsf{Dec}[\mathsf{rk}]$ oracle with $O[A]$, where A is the affine subspace defined by rk, and also to simulate its $\mathsf{DecX}[\mathsf{rk}]$ 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*

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

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

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

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

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

and in hybrid $\mathcal{H}_i$, we switch the i'th-from-last query from being answered by $O[A]$ to being answered by $O[T \setminus \{0^n\}]$. By Lemma 2.4, 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

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

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*

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

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

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 $|\widetilde{\psi}\rangle := (\mathsf{U}_B^\dagger |\psi\rangle, B)$, 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 $|\psi_A\rangle \coloneqq \mathsf{U}_{\mathsf{Com}}^{O[A],O[A^\perp]}(|A\rangle)$, there exists a state $|\psi'_A\rangle$ such that*

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

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* $|\widetilde{\psi}_G\rangle \coloneqq \mathsf{Amplify}_{\alpha,\beta}^{\mathsf{U}_{|\psi\rangle\langle\psi|},\mathsf{U}_\Pi}(|\psi\rangle)$ *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

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

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

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 $O(\log(1/\beta)\alpha)\cdot\mathrm{poly}(\lambda) \leq O(\lambda 2^\lambda)\cdot\mathrm{poly}(\lambda) \leq 2^{2\lambda}$ queries to $O[A]$ and $O[A^\perp]$.

Now, define

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

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

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

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

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

so the guarantee of Imported Theorem 4.18 implies that

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

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*

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

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 $|*\rangle\,\widetilde{\psi}_F$. Then if $|\langle\psi_F|\psi_G\rangle| \geq c$ for all $(F,G) \in \mathcal{R}$ and $\mathbb{E}_{(F,G)\leftarrow\mathcal{R}}[|\langle\widetilde{\psi}_F|\widetilde{\psi}_G\rangle|] \leq d$, 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 $\dim(A_0 \cap B_0) = n/2-2$ and $\dim(A_1 \cap B_1) = n/2-2$. Note that for any $(F_A, F_B) \in \mathcal{R}$, $\dim(A \cap B) = n/2-1$.

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

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

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

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

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

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

Thus, we have that

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

Now, we appeal to the following theorem, which is proven in Section 4.5.2.

Theorem 4.20. *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 $A=(A_0,A_1)$ 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.

□

4.3 Verification of quantum partitioning circuits

4.3.1 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.

- $\mathsf{Gen}(1^\lambda, Q) \to (\mathsf{vk}, |\mathsf{pk}\rangle, \mathsf{PK})$: The Gen algorithm takes as input the security parameter 1^λ and the description of a quantum circuit $Q : \{0,1\}^{n'} \to \{0,1\}^n$, 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 $\mathsf{PK} : \{0,1\}^* \to \{0,1\}^*$.

- $\mathsf{Prove}^{\mathsf{PK}}(|\mathsf{pk}\rangle, Q, x) \to \pi$: 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.

- $\mathsf{Ver}(\mathsf{vk}, x, \pi) \to \{(q_1, \ldots, q_m)\} \cup \{\bot\}$: 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 $\bot$.

- $\mathsf{Combine}(b_1, \ldots, b_m) \to b$: The 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 4.21 (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)*

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

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 4.22 (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)*

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

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

4.3.2 QPIP$_1$ verification

First, we recall an information-theoretic protocol for verifying quantum partitioning circuits using only single-qubit standard and Hadamard basis measurements.[25] 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.

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

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 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. 7, 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. 7 (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 4.23. *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 4.24. *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 4.25. *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 4.26 ([CLLW22, Bar21])**.** *The protocol* Π^{QV} *(Fig. 7) 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* $\{|\psi^*_\lambda\rangle\}_{\lambda\in\mathbb{N}}$*,*

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

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

4.3.3 Classical verification

Next, we compile the above information-theoretic protocol into a classically-verifiable but computationally-sound protocol, using Mahadev's measurement protocol [Mah18b]. 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 $(\mathsf{TCF.Gen}, \mathsf{TCF.Eval}, \mathsf{TCF.Invert}, \mathsf{TCF.Check}, \mathsf{TCF.IsValid})$ (Definition 2.15), and is described in Fig. 8. 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

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

$w \in \{0,1\}^{|S|}$ defined as follows. Let $T := H(y_{1,1}, \ldots, y_{r,\ell})$. The string w consists of r sub-strings $w_1, \ldots, w_r$, 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 4.28 follows immediately from the completeness of Π^{QV} (Imported Theorem 4.26) and the correctness of the dual-mode randomized trapdoor claw-free hash function (Definition 2.15). The proofs of the remaining three lemmas mostly follow from the prior work of [Bar21], and we show this formally in Section 4.5.3.

Definition 4.27. *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*

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

Lemma 4.28 (Completeness)**.** *The protocol Π^{CV} (Fig. 8) 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 $\{x_\lambda\}_{\lambda \in \mathbb{N}}$,*

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

Lemma 4.29 (Soundness)**.** *The protocol Π^{CV} (Fig. 8) 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*

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

Lemma 4.30 (D_{in} if accept)**.** *The protocol Π^{CV} (Fig. 8) 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*

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

Lemma 4.31 (D_{out} if accept wrong output)**.** *The protocol Π^{CV} (Fig. 8) 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*

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

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

4.3.4 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 8).

- An X-measurable commitment $\mathsf{XMC} = (\mathsf{XMC.Gen}, \mathsf{XMC.Com}, \mathsf{XMC.Open}, \mathsf{XMC.Dec}, \mathsf{XMC.OpenX}, \mathsf{XMC.DecX})$ 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 2.23).
- A pseudorandom function F_k secure against superposition-query attacks [Zha12].

Theorem 4.32. *The protocol Π^{PV} (Fig. 9) satisfies Definition 4.21 and Definition 4.22.*

Proof. We argue completeness (Definition 4.21) and soundness (Definition 4.22).

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 2.20), 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 $F_{k_2}(x, c, \sigma)$.

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. 8).

- $\mathsf{PV.Gen}(1^\lambda, Q) \to (\mathsf{vk}, |\mathsf{pk}\rangle, \mathsf{PK})$:
  - Sample $\{(\mathsf{rk}_{i,j}, |\mathsf{ck}_{i,j}\rangle, \mathsf{CK}_{i,j}) \leftarrow \mathsf{XMC.Gen}(1^\lambda)\}_{i\in[r],j\in[\ell]}$.
  - 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 $\mathsf{H}(\cdot) := F_{k_1}(\cdot)$, 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 $(\mathsf{pp}, \mathsf{sp}) := \mathsf{V}^{\mathsf{CV}}_{\mathsf{Gen}}(1^\lambda, Q; F_{k_2}(x, c, \sigma))$ and output pp.
  - Set $\mathsf{vk} := (Q, k_1, k_2, \mathsf{vk}_{\mathsf{Tok}}, \{\mathsf{rk}_{i,j}\}_{i\in[r],j\in[\ell]})$, $|\mathsf{pk}\rangle := (|\mathsf{sk}_{\mathsf{Tok}}\rangle, \{|\mathsf{ck}_{i,j}\rangle\}_{i\in[r],j\in[\ell]})$, and $\mathsf{PK} := (\mathsf{H}, \mathsf{CVGen}, \{\mathsf{CK}_{i,j}\}_{i\in[r],j\in[\ell]})$.
- $\mathsf{PV.Prove}^{\mathsf{PK}}(|\mathsf{pk}\rangle, Q, x) \to \pi$:
  - 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 $\mathsf{XMC.Com}^{\mathsf{CK}_{i,j}}(\mathcal{B}_{i,j}, |\mathsf{ck}_{i,j}\rangle) \to (\mathcal{B}_{i,j}, \mathcal{U}_{i,j}, c_{i,j})$ (see Definition 4.1).
  - Set $c := (c_{1,1}, \ldots, c_{r,\ell})$, compute $\sigma \leftarrow \mathsf{Tok.Sign}((x, c), |\mathsf{sk}_{\mathsf{Tok}}\rangle)$, and compute $\mathsf{pp} := \mathsf{CVGen}(x, c, \sigma)$.
  - 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 $T := \mathsf{H}(y_{1,1}, \ldots, y_{r,\ell})$
  - For each $i : T_i = 0, j \in [\ell]$, apply $\mathsf{XMC.Open}(\mathcal{B}_{i,j}, \mathcal{U}_{i,j}) \to u_{i,j}$.
  - For each $i : T_i = 1, j \in [\ell]$, apply $\mathsf{XMC.OpenX}(\mathcal{B}_{i,j}, \mathcal{U}_{i,j}) \to u_{i,j}$.
  - Set $\pi := (c, \sigma, \{u_{i,j}, y_{i,j}, z_{i,j}\}_{i\in[r],j\in[\ell]})$.
- $\mathsf{PV.Ver}(\mathsf{vk}, x, \pi) \to \{\{\{q_{i,t}\}_{t\in[\lambda]}\}_{i:T_i=1}\} \cup \{\bot\}$:
  - Parse $\mathsf{vk} := (Q, k_1, k_2, \mathsf{vk}_{\mathsf{Tok}}, \{\mathsf{rk}_{i,j}\}_{i\in[r],j\in[\ell]})$ and $\pi := (c, \sigma, \mu)$.
  - If $\mathsf{Tok.Verify}(\mathsf{vk}_{\mathsf{Tok}}, (x, c), \sigma) = \top$, then set $(\mathsf{pp}, \mathsf{sp}) := \mathsf{V}^{\mathsf{CV}}_{\mathsf{Gen}}(1^\lambda, Q; F_{k_2}(x, c, \sigma))$, 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 $T := F_{k_1}(y_{1,1}, \ldots, y_{r,\ell})$.
  - For all $i : T_i = 0, j \in [\ell]$, compute $b_{i,j} := \mathsf{XMC.Dec}(\mathsf{rk}_{i,j}, c_{i,j}, u_{i,j})$, and return $\bot$ if $b_{i,j} = \bot$.
  - For all $i : T_i = 1, j \in [\ell]$ such that $h_{i,j} = 1$, compute $b_{i,j} := \mathsf{XMC.DecX}(\mathsf{rk}_{i,j}, c_{i,j}, u_{i,j})$, 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 $\widetilde{\pi} := \{b_{i,j}, y_{i,j}, z_{i,j}\}_{i\in[r],j\in[\ell]}$ and return $\{\{q_{i,t}\}_{t\in[\lambda]}\}_{i:T_i=1} := \mathsf{V}^{\mathsf{CV}}_{\mathsf{Ver}}(Q, x, \mathsf{sp}, \widetilde{\pi})$.
- $\mathsf{PV.Combine} \equiv \mathsf{MM}_\lambda$ (see Definition 4.27).

Figure 9: Publicly-verifiable non-interactive classical verification of quantum partitioning circuits.

Now, imagine sampling s and fixing $(\mathsf{pp}, \mathsf{sp}) \coloneqq \mathsf{V}^{\mathsf{CV}}_{\mathsf{Gen}}(1^\lambda, Q; s)$ 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

$$\mathsf{XMC.Com}^{\mathsf{CK}_{i,j}}(\mathcal{B}_{i,j}, |\mathsf{ck}_{i,j}\rangle)$$

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 $\mathsf{XMC.Com}$ 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 $\mathsf{XMC.Com}$ 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 $\mathsf{XMC.Com}, \mathsf{XMC.Open}$, and $\mathsf{XMC.Dec}$ 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 $\mathsf{XMC.Com}, \mathsf{XMC.OpenX}$, and $\mathsf{XMC.DecX}$ in succession to $\mathcal{B}_{i,j}$. Thus, by the correctness of XMC (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 4.28), 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 $\mathsf{non\text{-}negl}(\lambda)$ 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 (Lemma 2.9) to obtain a two-stage adversary A_3, where the first stage outputs x^*, a XMC commitment c^*, and a token signature σ^*, and the second stage outputs the remainder μ^* of the proof $\pi^* \coloneqq (c^*, \sigma^*, \mu^*)$. 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 2.23) 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 4.31 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 $\mathsf{PV.Ver}$. 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 4.29) 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 4.30, 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 XMC (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 XMC 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 XMC 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 4.22, 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 $a := (x^*, c^*, \sigma^*)$ and $\mathsf{aux} := (\mu^*, \mathsf{vk})$.

Functionalities used in the proof of Theorem 4.32

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 $(\mathsf{pp}, \mathsf{sp}) := \mathsf{V}_{\mathsf{Gen}}^{\mathsf{CV}}(1^\lambda, Q; s)$.
- $\mathsf{PV.Ver}[\mathsf{vk}, s, (x^*, c^*, \sigma^*)](x, \pi)$: 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 10: Description of functionalities used in the proof of Theorem 4.32.

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. 10. Then since A_1 breaks soundness,

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

Next, since $p = \mathrm{poly}(\lambda)$, by Lemma 2.9 there exists an algorithm $\mathsf{A}_3 := \mathsf{Sim}[\mathsf{A}_2]$ such that

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

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. 10.

Claim 4.33.

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

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 2.23), 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. □

Next, we claim the following, where $V[\mathsf{out}]$ is defined in Protocol 10.

Claim 4.34.

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

Proof. First, if we replace the PRF $F_{k_1}(\cdot)$ with an external random oracle H, then the probabilities in Claim 4.33 and Claim 4.34 remain the same up to a negligible difference [Zha12]. Next, note that the only event that differentiates Claim 4.33 and Claim 4.34 is when A_4 outputs (x^*,c^*,σ^*,μ^*) 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 4.31. Thus, the difference in probability must be negligible, completing the proof. □

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 $((x^*,c^*,\sigma^*),\mathsf{st})\leftarrow\mathsf{A}_4$.
- Sample $s\leftarrow\{0,1\}^\lambda$.
- Run $(\mu^*,\mathsf{vk})\leftarrow\mathsf{A}_4(s,\mathsf{st})$ with the following difference. Recall that at some point, A_4 begins using the oracle $G[(x^*,c^*,\sigma^*)\rightarrow s]$ 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 $O_\bot$ that outputs $\bot$ on every input.
- Output $V[\mathsf{out}]((x^*,c^*,\sigma^*),s,(\mu^*,\mathsf{vk}))$.

Note that Claim 4.34 is stating exactly that $\Pr[\mathcal{H}_0=1]=\mathsf{non\text{-}negl}(\lambda)$. Next, we have the following claim.

Claim 4.35. $\Pr[\mathcal{H}_p=1]=\mathrm{negl}(\lambda)$.

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 4.29). Note that A_4 never needs to know the sp such that $(\mathsf{pp}, \mathsf{sp}) \coloneqq \mathsf{V}^{\mathsf{CV}}_{\mathsf{Gen}}(1^\lambda, Q; s)$, 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 $\widetilde{\pi} \coloneqq \{b_{i,j}, y_{i,j}, z_{i,j}\}_{i,j}$ (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 $q \coloneqq \mathsf{V}^{\mathsf{CV}}_{\mathsf{Ver}}(Q, x^*, \mathsf{sp}, \widetilde{\pi})$, which by Lemma 4.29 must be $\mathrm{negl}(\lambda)$. □

Finally, we prove the following Claim 4.36. Since $p = \mathrm{poly}(\lambda)$, this contradicts Claim 4.34 and Claim 4.35, which completes the proof.

□

Claim 4.36. *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 10.

So, it suffices to show that

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

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

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 $\Pr\left[\mathcal{H}'_{\iota-1} = 1\right]$ remain the same up to negligible difference [Zha12]. Now, this follows from a reduction to Lemma 4.30. Indeed, note that if $|\Pr\left[\mathcal{H}'_{\iota-1} = 1\right] - \Pr[\mathcal{H}_{\iota-1} = 1]| = \mathsf{non\text{-}negl}(\lambda)$, 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 4.30, the above event occurs with negligible probability.

□

Claim 4.38. $\Pr[\mathcal{H}_\iota = 1] \geq \Pr\left[\mathcal{H}'_{\iota-1} = 1\right] - \mathrm{negl}(\lambda)$

Proof. We will show this by reduction to the string binding with public decodability property of XMC. Recall from Section 4.3.3 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 $\{\mathsf{rk}_\tau, |\mathsf{ck}_\tau\rangle, \mathsf{CK}_\tau \leftarrow \mathsf{XMC.Gen}(1^\lambda)\}_{\tau\in[m]}$.
- 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 $\mathsf{rk}_{i,j}, |\mathsf{ck}_{i,j}\rangle, \mathsf{CK}_{i,j} \leftarrow \mathsf{XMC.Gen}(1^\lambda)$. Let $f : [m] \to S$ be an arbitrary bijection, and re-define $\{\mathsf{rk}_\tau, |\mathsf{ck}_\tau\rangle, \mathsf{CK}_\tau\}_{\tau\in[m]}$ as $\{\mathsf{rk}_{f(\tau)}, |\mathsf{ck}_{f(\tau)}\rangle, \mathsf{CK}_{f(\tau)}\}_{\tau\in[m]}$.
- C runs A_4 as defined by $\mathcal{H}'_{\iota-1}$ until right before query ι is answered. All queries to $\mathsf{CK}_{i,j}$, $\mathsf{XMC.Dec}[\mathsf{rk}_{i,j}]$, or $\mathsf{XMC.DecX}[\mathsf{rk}_{i,j}]$ for $(i,j) \in S$ are forwarded to external oracles.

That is, we can write the operation of C as

$$|\psi\rangle \leftarrow \mathsf{C}^{\mathbf{CK}, \mathsf{XMC.Dec}[\mathbf{rk}], \mathsf{XMC.DecX}[\mathbf{rk}]}(|\mathbf{ck}\rangle),$$

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

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 $\mathsf{XMC.Dec}[\mathsf{rk}_{i,j}]$ 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 $\mathsf{XMC.DecX}[\mathsf{rk}_{i,j}]$ 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

$$|\psi'\rangle \coloneqq \mathsf{U}^{\mathbf{CK},\mathsf{XMC.Dec}[\mathbf{rk}]}(|\psi\rangle).$$

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.

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

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

Now, observe that

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

and

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

Furthermore, for any state $|\psi\rangle$ output by C, we can write $|\psi\rangle := |\psi_{\mathsf{in}}\rangle + |\psi_{\mathsf{in}}^{\perp}\rangle$, where $|\psi_{\mathsf{in}}\rangle := \Pi_{\mathsf{in}}^{\mathcal{Q},\mathcal{A}} |\psi\rangle$. 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

$$\Pi_{\mathsf{out},\mathsf{U}} := \left(\mathsf{U}^{\mathbf{CK},\mathsf{XMC.Dec}[\mathbf{rk}]}\right)^{\dagger} \Pi_{\mathsf{out}} \left(\mathsf{U}^{\mathbf{CK},\mathsf{XMC.Dec}[\mathbf{rk}]}\right),$$
$$\Pi_{\mathsf{out},\widetilde{\mathsf{U}}} := \left(\widetilde{\mathsf{U}}^{\mathbf{CK},\mathsf{XMC.Dec}[\mathbf{rk}]}\right)^{\dagger} \Pi_{\mathsf{out}} \left(\widetilde{\mathsf{U}}^{\mathbf{CK},\mathsf{XMC.Dec}[\mathbf{rk}]}\right),$$

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

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

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

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

and thus it suffices to show that

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

and

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

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,

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

Now, we observe that the register $\mathcal{A}$ output by C contains a standard basis state holding $(\mathsf{vk}, s, (x^*, c^*, \sigma^*))$, where $c^* := \{c^*_{i,j}\}_{i\in[r],j\in[\ell]}$. Define commitments $\mathbf{c} := \{c^*_{i,j}\}_{(i,j)\in S}$ 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.

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

- 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 $(i, j) \in S$, $w_{f^{-1}(i,j)} = \mathsf{XMC.Dec}(\mathsf{rk}_{i,j}, c^*_{i,j}, u_{i,j})$.

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

- 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 $(i, j) \in S$, $w_{f^{-1}(i,j)} = \mathsf{XMC.Dec}(\mathsf{rk}_{i,j}, c^*_{i,j}, u_{i,j})$.

Next, we define the following two projectors.

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

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

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

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

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

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

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 $\mathcal{Q}' = (\mathcal{Q}_1, \ldots, \mathcal{Q}_m)$ and $\mathcal{V}' = (\mathcal{V}_1, \ldots, \mathcal{V}_m)$, we can write

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

where

$$\Pi_{\mathsf{rk}_\tau,c^*_\tau,w_\tau} \coloneqq \sum_{u:\mathsf{XMC.Dec}(\mathsf{rk}_\tau,c^*_\tau,u)=w_\tau} |u\rangle\langle u| .$$

Now, to complete the proof, we note that

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathbf{CK},\mathbf{rk},|\mathbf{ck}\rangle}\left[\left\|\Pi_{\mathbf{rk},\mathbf{c},\mathsf{out}}\mathsf{U}^{\mathbf{CK},\mathsf{XMC.Dec}[\mathbf{rk}]}\Pi_{\mathbf{rk},\mathbf{c},\mathsf{in}}|\psi\rangle\right\|^2 : (|\psi\rangle,\mathbf{c}) \leftarrow \mathsf{C}_s^{\mathbf{CK},\mathsf{XMC.Dec}[\mathbf{rk}],\mathsf{XMC.DecX}[\mathbf{rk}]}(|\mathbf{ck}\rangle)\right]\\
\leq &\mathop{\mathbb{E}}_{\mathbf{CK},\mathbf{rk},|\mathbf{ck}\rangle}\left[\left\|\Pi_{\mathbf{rk},\mathbf{c},\mathsf{out},0}\mathsf{U}^{\mathbf{CK},\mathsf{XMC.Dec}[\mathbf{rk}]}\Pi_{\mathbf{rk},\mathbf{c},\mathsf{in},0}|\psi\rangle\right\|^2 : (|\psi\rangle,\mathbf{c}) \leftarrow \mathsf{C}_s^{\mathbf{CK},\mathsf{XMC.Dec}[\mathbf{rk}],\mathsf{XMC.DecX}[\mathbf{rk}]}(|\mathbf{ck}\rangle)\right]\\
&+\mathop{\mathbb{E}}_{\mathbf{CK},\mathbf{rk},|\mathbf{ck}\rangle}\left[\left\|\Pi_{\mathbf{rk},\mathbf{c},\mathsf{out},1}\mathsf{U}^{\mathbf{CK},\mathsf{XMC.Dec}[\mathbf{rk}]}\Pi_{\mathbf{rk},\mathbf{c},\mathsf{in},1}|\psi\rangle\right\|^2 : (|\psi\rangle,\mathbf{c}) \leftarrow \mathsf{C}_s^{\mathbf{CK},\mathsf{XMC.Dec}[\mathbf{rk}],\mathsf{XMC.DecX}[\mathbf{rk}]}(|\mathbf{ck}\rangle)\right],
\end{aligned}$$

and by the string binding with public decodability of XMC (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 $b\in\{0,1\}$,

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathbf{CK},\mathbf{rk},|\mathbf{ck}\rangle}\left[\left\|\Pi_{\mathbf{rk},\mathbf{c},\mathsf{out},b}\mathsf{U}^{\mathbf{CK},\mathsf{XMC.Dec}[\mathbf{rk}]}\Pi_{\mathbf{rk},\mathbf{c},\mathsf{in},b}|\psi\rangle\right\| : (|\psi\rangle,\mathbf{c}) \leftarrow \mathsf{C}_s^{\mathbf{CK},\mathsf{XMC.Dec}[\mathbf{rk}],\mathsf{XMC.DecX}[\mathbf{rk}]}(|\mathbf{ck}\rangle)\right]\\
&\qquad = \mathrm{negl}(\lambda).
\end{aligned}$$

□

4.3.5 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 4.39 (Publicly-verifiable QFHE for pseudo-deterministic circuits)**.** *A publicly-verifiable quantum fully-homomorphic encryption scheme for pseudo-deterministic circuits consists of the following algorithms* $(\mathsf{Gen},\mathsf{Enc},\mathsf{VerGen},\mathsf{Eval},\mathsf{Ver},\mathsf{Dec})$.

- $\mathsf{Gen}(1^\lambda, D)\to(\mathsf{pk},\mathsf{sk})$*: 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* $\{x_{\lambda,0}, x_{\lambda,1}\}_{\lambda\in\mathbb{N}}$*,*

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

- ***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 2.2.3).
- 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 4.3.4).

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 $\widetilde{\mathsf{ct}}$.[26]
  - 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 4.40. *The scheme described above satisfies Definition 4.39.*

Proof. Correctness follows immediately from the evaluation correctness of QFHE (Definition 2.18) and the completeness of PV (Definition 4.21). Security follows immediately from the semantic security of QFHE (Definition 2.17). Soundness follows immediately from the correctness of QFHE (Definition 2.18) and soundness of PV (Definition 4.22), 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. □

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

4.4 Obfuscation

4.4.1 Construction

In this section, we construct ideal obfuscation for pseudo-deterministic quantum circuits (Definition 2.10) from any publicly-verifiable QFHE for pseudo-deterministic circuits in the oracle model $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{VerGen}, \mathsf{Eval}, \mathsf{Ver}, \mathsf{Dec})$ (Definition 4.39).

The construction is given in Fig. 11.

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 $\perp$, then output $\perp$, and otherwise output $\mathsf{Dec}(\mathsf{sk}, \widetilde{\mathsf{ct}})$.
  - Output $\widetilde{Q} := (\mathsf{ct}, |\mathsf{pp}\rangle), O := (\mathsf{PP}, \mathsf{DK})$.
- $\mathsf{QEval}^O(\widetilde{Q}, x)$:
  - Parse $\widetilde{Q}$ as $\mathsf{ct}, |\mathsf{pp}\rangle$ and O as PP, DK.
  - Compute $(\widetilde{\mathsf{ct}}, \pi) \leftarrow \mathsf{Eval}^{\mathsf{PP}}(\mathsf{ct}, |\mathsf{pp}\rangle, x)$.
  - Output $b := \mathsf{DK}(x, \widetilde{\mathsf{ct}}, \pi)$.

Figure 11: Obfuscation for pseudo-deterministic quantum circuits.

Theorem 4.41. $(\mathsf{QObf}, \mathsf{QEval})$ *described in Fig. 11 is an ideal obfuscator for pseudo-deterministic quantum circuits, satisfying Definition 2.10.*

Proof. First, correctness follows immediately from the correctness of the publicly-verifiable QFHE scheme (Definition 4.39). 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 2.3), 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.

- 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 $\mathsf{A}_\lambda^{\mathsf{PP},\mathsf{DK}}(\mathsf{ct}, |\mathsf{pp}\rangle)$, 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 Q, and return the result $b = Q(x)$.

- Output A_λ's output.

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

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

We complete the proof by showing the following for any QPT distinguisher $\{\mathsf{D}_\lambda\}_{\lambda\in\mathbb{N}}$.

- $|\Pr[\mathsf{D}_\lambda(\mathcal{H}_0) = 1] - \Pr[\mathsf{D}_\lambda(\mathcal{H}_1) = 1]| = \mathrm{negl}(\lambda)$. Suppose otherwise. Then there must exist some query made by A_λ 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 $\mathrm{poly}(\lambda)$ many queries made by A_λ to obtain such an $(x, \widetilde{\mathsf{ct}}, \pi)$, which violates the soundness of the publicly-verifiable QFHE scheme (Definition 4.39).
- $|\Pr[\mathsf{D}_\lambda(\mathcal{H}_1) = 1] - \Pr[\mathsf{D}_\lambda(\mathcal{H}_2) = 1]| = \mathrm{negl}(\lambda)$. Since sk is no longer used in $\mathcal{H}_1$ to respond to DK queries, this follows directly from the security of the publicly-verifiable QFHE scheme (Definition 4.39).

□

4.4.2 Application: Functional encryption for BQP

We sketch an application of our obfuscation scheme to functional encryption for pseudo-deterministic quantum functionalities. Let $(\mathsf{QObf}, \mathsf{QEval})$ be an ideal obfuscation scheme for pseudo-deterministic quantum circuits,[27] 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+13] 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 $\mathsf{msk} := \mathsf{sk}_1$.

[27] For this application, we technically only require the weaker notion of indistinguishability obfuscation (see Definition 2.10).

- $\mathsf{FE.KeyGen}(\mathsf{msk}, Q)$: 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 $\bot$.

  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+13] to consider functional encryption and obfuscation of pseudo-deterministic quantum circuits. As a result, we obtain the following theorem.

Theorem 4.42 (Corollary of [GGH+13] Section 6 and Theorem 4.41)**.** *The above construction is a functional encryption scheme satisfying indistinguishability security for the class of polynomial-size pseudo-deterministic quantum functionalities.*

4.5 Deferred proofs

4.5.1 Proofs from Section 4.2.1

Lemma 4.43. *Any X-measurable 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 4.44 (Collapse binding)**.** *An X-measurable commitment* $(\mathsf{Gen}, \mathsf{Com}, \mathsf{Open}, \mathsf{Dec}, \mathsf{OpenX}, \mathsf{DecX})$ *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* $\mathsf{rk}, |\mathsf{ck}\rangle, \mathsf{CK} \leftarrow \mathsf{Gen}(1^\lambda)$.
- *Run* $\mathsf{C}_\lambda^{\mathsf{CK},\mathsf{Dec}[\mathsf{rk}],\mathsf{DecX}[\mathsf{rk}]}(|\mathsf{ck}\rangle)$ *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* $\{\Pi_{\mathsf{rk},c,0}, \Pi_{\mathsf{rk},c,1}\}$.[28]

[28] These projectors are defined in Definition 4.3.

- *Run* $\mathsf{U}_\lambda^{\mathsf{CK},\mathsf{Dec}[\mathsf{rk}]}(\mathcal{B},\mathcal{U},\mathcal{A})$ *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* $\Pi_{\mathsf{rk},c,0} + \Pi_{\mathsf{rk},c,1}$*. Then, it must hold that for all valid adversaries* A*,*

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

Definition 4.45 (Unique message binding)**.** *An X-measurable commitment* $(\mathsf{Gen},\mathsf{Com},\mathsf{Open},\mathsf{Dec},\mathsf{OpenX},\mathsf{DecX})$ *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* $\{\mathsf{rk}_i, |\mathsf{ck}_i\rangle, \mathsf{CK}_i \leftarrow \mathsf{Gen}(1^\lambda)\}_{i\in[m]}$*.*
- *Run* $\mathsf{C}_\lambda^{\mathbf{CK},\mathsf{Dec}[\mathbf{rk}],\mathsf{DecX}[\mathbf{rk}]}(|\mathbf{ck}\rangle)$ *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* $\Pi_{\mathsf{rk}_i,c_i,x_{1,i}}$ *to* $(\mathcal{B}_i,\mathcal{U}_i)$ *and abort and output 0 if this projection rejects.*
- *Run* $\mathsf{U}_\lambda^{\mathbf{CK},\mathsf{Dec}[\mathbf{rk}]}(\mathcal{B}_1,\mathcal{U}_1,\ldots,\mathcal{B}_m,\mathcal{U}_m,\mathcal{A})$ *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* $\Pi_{\mathsf{rk}_i,c_i,x_{2,i}}$ *to* $(\mathcal{B}_i,\mathcal{U}_i)$ *and abort and output 0 if this projection rejects. Otherwise, output 1.*

Imported Theorem 4.46 ([LMS22])**.** *Any commitment that satisfies collapse binding also satisfies unique message binding.*

Imported Theorem 4.47 ([DS23])**.** *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.

- First, we establish using Imported Theorem 4.47 that any X-measurable 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 $\Pi_0 = \Pi_{\mathsf{rk},c,0}$, $\Pi_1 = \Pi_{\mathsf{rk},c,1}$, let D be a projective implementation of $\mathsf{U}^{\mathsf{CK},\mathsf{Dec}[\mathsf{rk}]}$, and let $|\psi\rangle$ be the state of the collapse binding experiment that is output by $\mathsf{C}^{\mathsf{CK},\mathsf{Dec}[\mathsf{rk}],\mathsf{DecX}[\mathsf{rk}]}$. Then the RHS of Imported Theorem 4.47 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 4.46, we see that any X-measurable commitment that satisfies single-bit binding also satisfies unique message binding.
- Finally, suppose there is An X-measurable 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 $\mathsf{Dec}(\mathbf{rk}, \mathbf{c}, \cdot)$ applied to the state $\Pi_{\mathbf{rk},\mathbf{c},W_0} |\psi\rangle$, 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.

□

4.5.2 Proofs from Section 4.2.3

In this section, we prove the following theorem.

Theorem 4.48. *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 (Theorem 4.49). 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 4.48) 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 $S \coloneqq \mathsf{span}(z_1, \ldots, z_{d-2})$ such all vectors in S start with 0, let $z_{d-1}, \ldots, z_n$ be such that $(z_1, \ldots, z_n)$ 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 $\mathsf{co}(S) \coloneqq \mathsf{span}(z_{d-1}, \ldots, z_n)$. 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 $(z_{d-1}, \ldots, z_n)$-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 $(e_1, \ldots, e_n) \to (z_1, \ldots, z_n)$, where the e_i are the standard basis vectors, and let

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

Then, re-defining

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

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

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

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 4.49 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.

□

Theorem 4.49. *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 $v_0 + w_0 = v_1 + w_1$.[29] We will write any $A \in \mathcal{A}$ as $A \coloneqq (A_0, A_1)$, where $A_0 \coloneqq \{v_0, w_0\}$ and $A_1 = \{v_1, w_1\}$. For any such A, let*

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

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

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

[29] Note that this theorem is not strictly the two-dimensional version of Theorem 4.48, 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 4.48, and in particular v_0, w_0 do not necessarily start with 0 and v_1, w_1 do not necessarily start with 1.

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 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 4.50).

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 4.51). 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 4.52).

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 4.49) 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 4.50.

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

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

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

we have that

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

By Fact 4.55, $E[C_1, C_2, C_3] \leq 1/2$, so by Markov,

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

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

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

so by Fact 4.55,

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

which completes the proof of the claim. □

Claim 4.51.

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

Proof. First, note that the proof of Claim 4.50 also shows that

$$\mathop{\mathbb{E}}_{(A,B)\leftarrow\mathcal{R}} \left[|\alpha_A^{v_{A,B}}| \cdot |\alpha_B^{v_{A,B}}| \right] \leq \frac{1}{2},$$

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

By our assumption that

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

and linearity of expectation,

$$\mathop{\mathbb{E}}_{(A,B)\leftarrow\mathcal{R}}\left[|\alpha_A^{v_{A,B}}|\cdot|\alpha_B^{v_{A,B}}|\cdot\left(1-|\left\langle\phi_A^{v_{A,B}}|\phi_B^{v_{A,B}}\right\rangle|\right)\right]\leq\epsilon^{13}.$$

Now, assume for contradiction that

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

By Claim 4.50, this implies that

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

But then,

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

whenever $\epsilon<1/4$.

□

Claim 4.52. *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*

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

be the set of 2^{n-1} states indexed by A_1 such that $(A_0^*,A_1)\in\mathcal{A}$.[30] Then there exists a set $\mathcal{A}_1^*$ of size at least 2^{n-2} such that for all $A_1\in\mathcal{A}_1^*$,

$$|\left\langle\phi_{(A_0^*,A_1)}^{v_0^*}|\tau^{v_0^*}\right\rangle|\geq1-2\epsilon^3\ \text{ and }\ |\left\langle\phi_{(A_0^*,A_1)}^{w_0^*}|\tau^{w_0^*}\right\rangle|\geq1-2\epsilon^3.$$

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

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

Then Claim 4.51 implies that there exists some set $\{v_0^*,w_0^*\}$ such that

$$\Pr_{(A,B)\leftarrow\mathcal{R}[(v_0^*,w_0^*)]}\left[|\left\langle\phi_A^{v_{A,B}}|\phi_B^{v_{A,B}}\right\rangle|=|\left\langle\phi_A^{v_0^*}|\phi_B^{v_0^*}\right\rangle|\geq1-\epsilon^6\right]\geq1-4\epsilon^6,\ \text{ and}$$
$$\Pr_{(A,B)\leftarrow\mathcal{R}[(w_0^*,v_0^*)]}\left[|\left\langle\phi_A^{v_{A,B}}|\phi_B^{v_{A,B}}\right\rangle|=|\left\langle\phi_A^{w_0^*}|\phi_B^{w_0^*}\right\rangle|\geq1-\epsilon^6\right]\geq1-4\epsilon^6.$$

[30]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 $v_1+w_1=v_0^*+w_0^*$.

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

$$\left\{|\psi_{(A_0^*,A_1)}\rangle \coloneqq \alpha^{v_0^*}_{(A_0^*,A_1)} |v_0^*\rangle |\phi^{v_0^*}_{(A_0^*,A_1)}\rangle + \alpha^{w_0^*}_{(A_0^*,A_1)} |w_0^*\rangle |\phi^{w_0^*}_{(A_0^*,A_1)}\rangle\right\}_{A_1\in\mathcal{A}_1},$$

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 $A_1 \in \mathcal{A}_1^*$,

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

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

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

which are[31]

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

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,

$$\left| \bigcup_{\{A_1, A_1'\}\in\mathcal{A}_1^{\times 2}} \mathcal{R}[(v_0^*, w_0^*), \{A_1, A_1'\}] \right| = 8 \cdot \binom{2^{n-1}}{2} = 2^{2n} - 2^{n+1}.$$

[31] 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.

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

$$\{\mathcal{R}[(v_0^*, w_0^*), \{A_1, A_1'\}]\}_{\{A_1, A_1'\} \in \mathcal{A}_1^{\times 2}}$$

partition $\mathcal{R}[(v_0^*, w_0^*)]$ equally into sets of size 8. Thus,[32]

$$\Pr_{\{A_1, A_1'\} \leftarrow \mathcal{A}_1^{\times 2}} \left[\forall (A, B) \in \mathcal{R}[(v_0^*, w_0^*), \{A_1, A_1'\}], |\langle \phi_A^{v_0^*} | \phi_B^{v_0^*} \rangle| \geq 1 - \epsilon^6\right] \geq 1 - 32\epsilon^6,$$

which means that there exists some $A_1^* = \{v_1^*, w_1^*\}$ such that

$$\Pr_{A_1 \leftarrow \mathcal{A}_1 \setminus \{A_1^*\}} \left[\forall (A, B) \in \mathcal{R}[(v_0^*, w_0^*), \{A_1^*, A_1\}], |\langle \phi_A^{v_0^*} | \phi_B^{v_0^*} \rangle| \geq 1 - \epsilon^6\right] \geq 1 - 32\epsilon^6 \geq \frac{7}{8},$$

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 4.54,

$$|\langle \phi_{(A_0^*, A_1)}^{v_0^*} | \phi_{(A_0^*, A_1^*)}^{v_0^*} \rangle| \geq (1 - \epsilon^6)^2 - \sqrt{2\epsilon^6} \geq 1 - 2\epsilon^3.$$

Then if we set $|\tau^{v_0^*}\rangle \coloneqq |\phi_{(A_0^*, A_1^*)}^{v_0^*}\rangle$, we have that for all $A_1 \in \mathcal{A}_1^{v_0^*}$,

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

[32] 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.

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 $A_1 \in \mathcal{A}_1^{w_0^*}$,

$$|\langle \phi^{w_0^*}_{(A_0^*,A_1)} | \tau^{w_0^*} \rangle| \geq 1 - 2\epsilon^3.$$

Thus, setting $\mathcal{A}_1^* \coloneqq \mathcal{A}_1^{v_0^*} \cap \mathcal{A}_1^{w_0^*}$ (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

$$\Pi^* \coloneqq |v_0^*\rangle \left|\tau^{v_0^*}\right\rangle\!\left\langle\tau^{v_0^*}\right| \langle v_0^*| + |w_0^*\rangle \left|\tau^{w_0^*}\right\rangle\!\left\langle\tau^{w_0^*}\right| \langle w_0^*| .$$

By Claim 4.52 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

$$|\psi^*_{A_1}\rangle \coloneqq \frac{\Pi^* |\psi_{(A_0^*,A_1)}\rangle}{\|\Pi^* |\psi_{(A_0^*,A_1)}\rangle\|}.$$

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 4.54 (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 $|\psi_i\rangle \coloneqq |\psi^*_{A_1}\rangle$.
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 4.54 (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 4.53 ([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 $n > 10\log(1/\epsilon)+6$.

□

Finally, we record some facts that were used in the preceding proof.

Fact 4.54. *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

$$|\phi_c\rangle \coloneqq \frac{\Pi|\phi_b\rangle}{\|\Pi|\phi_b\rangle\|},$$

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 4.55. *Let*

$$u_1 \coloneqq \begin{pmatrix} a_1 \\ a_2 \\ 0 \end{pmatrix}, u_2 \coloneqq \begin{pmatrix} b_1 \\ 0 \\ b_2 \end{pmatrix}, u_3 \coloneqq \begin{pmatrix} 0 \\ c_1 \\ c_2 \end{pmatrix}$$

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 $v_1 \coloneqq (a_1 \; a_2 \; b_1 \; b_2 \; c_1 \; c_2)$ and $v_2 \coloneqq (b_1 \; c_1 \; a_1 \; c_2 \; a_2 \; b_2)$. 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

$$m \coloneqq \max_{\substack{u_1, u_2, u_3, \\ u_1 \cdot u_2 = 1/2 - \delta}} \{u_1 \cdot u_3 + u_2 \cdot u_3\},$$

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 $\delta \in (0, 1/2)$. □

4.5.3 Proofs from Section 4.3.3

In this section, we prove Lemma 4.29, Lemma 4.30, and Lemma 4.31. We proceed via three steps.

1. Compile the information-theoretic protocol Π^{QV} from Section 4.3.2 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 [Mah18b]. 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 4.29, Lemma 4.30, and Lemma 4.31 hold in Π^{parl}.
3. Apply Fiat-Shamir to Π^{parl} to obtain the protocol Π^{CV} from Protocol 8, and observe that Measure and Re-program (Lemma 2.9) implies that Lemma 4.29, Lemma 4.30, and Lemma 4.31 must also hold with respect to Π^{CV}.

Proof. (of Lemma 4.29, Lemma 4.30, and Lemma 4.31)

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 $|\psi^{\mathsf{P}^*}_{\lambda,r}\rangle^{\mathcal{A},\mathcal{C}}$ 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.

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 $(\mathcal{A},\mathcal{C})$.

$$\Pi^{\mathsf{P}^*}_{\lambda,r,0} := \mathsf{U}^{\mathsf{P}^*\dagger}_{\lambda,0}\mathsf{V}_{\lambda,r,0}\mathsf{U}^{\mathsf{P}^*}_{\lambda,0}, \quad \Pi^{\mathsf{P}^*}_{\lambda,r,1} := \mathsf{U}^{\mathsf{P}^*\dagger}_{\lambda,1}\mathsf{V}_{\lambda,r,1}\mathsf{U}^{\mathsf{P}^*}_{\lambda,1}.$$

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 $\mathcal{B} = (\mathcal{B}_1, \ldots, \mathcal{B}_\ell)$, 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
$$\sum_{v\in\{0,1\}^\ell} \alpha_v |v\rangle^{\mathcal{B}} |\psi_{\mathsf{pk}_1, v_1}\rangle^{\mathcal{Z}_1,\mathcal{Y}_1}, \ldots, |\psi_{\mathsf{pk}_\ell, v_\ell}\rangle^{\mathcal{Z}_\ell,\mathcal{Y}_\ell}.$$
Finally, measure registers $\mathcal{Y}_1, \ldots, \mathcal{Y}_\ell$ 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 $m \coloneqq (m_1, \ldots, m_\ell)$ 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 12: A quantum "commit-challenge-response" protocol for verifying quantum partitioning circuits.

Definition 4.56. *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 4.26) and the soundness of the measurement protocol ([Mah18b]), we have the following claim.

Claim 4.57. *For any $\{\mathsf{P}^*_\lambda\}_{\lambda\in\mathbb{N}}$ attacking Π^{CCR} (Protocol in Fig. 12), 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 4.58 ([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 4.59 (Π^{parl} analogue of Lemma 4.29)**.** *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*

$$\Pr\left[\begin{array}{l}\mathsf{V}^{\mathsf{parl}}_{\mathsf{Ver}}(Q,x,\mathsf{sp},y,T,z)=\{\{q_{i,t}\}_{t\in[\lambda]}\}_{i:T_i=1}\ \wedge\\ \mathsf{MM}_\lambda\left(\{\{P(q_{i,t})\}_{t\in[\lambda]}\}_{i:T_i=1}\right)=1-P(Q(x))\end{array}\ :\ \begin{array}{r}(\mathsf{pp},\mathsf{sp})\leftarrow\mathsf{V}^{\mathsf{parl}}_{\mathsf{Gen}}(1^\lambda,Q)\\ y\leftarrow\mathsf{A}(\mathsf{pp})\\ T\leftarrow\{0,1\}^{\binom{\lambda^2}{\lambda}}\\ z\leftarrow\mathsf{A}(T)\end{array}\right]=\mathrm{negl}(\lambda),$$

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 4.57 and Imported Theorem 4.58 with $\epsilon = 1$ and $\delta = 1/2$. □

Lemma 4.60 (Π^{parl} analogue of Lemma 4.30)**.** *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 4.3.3, 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 2.15, 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 4.57 and Imported Theorem 4.58 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 4.61 (Π^{parl} analogue of Lemma 4.31)**.** *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 4.3.3, 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 $D_{\mathsf{out}}[P, P(Q(x))]$,

$$w \notin D_{\mathsf{out}}[P, P(Q(x))] \implies a \leq \frac{1}{3} + \frac{1}{\lambda}.$$

Thus, the proof is completed by showing that

$$\Pr\left[b - a \geq \frac{1}{2} - \left(\frac{1}{3} + \frac{1}{\lambda}\right) > \frac{1}{10}\right] \leq e^{-2(\lambda/10)^2} = \mathrm{negl}(\lambda),$$

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 redefined 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 (Lemma 2.9) show that Lemma 4.59, Lemma 4.60, and Lemma 4.61 imply Lemma 4.29, Lemma 4.30, and Lemma 4.31 respectively.

In more detail, suppose that Lemma 4.29 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

$$\mathsf{V}^{\mathsf{CV}}_{\mathsf{Ver}}(Q, x, \mathsf{sp}, \pi) = \{\{q_{i,t}\}_{t\in[\lambda]}\}_{i:T_i=1} \quad \wedge \quad \mathsf{MM}_\lambda(\{\{P(q_{i,t})\}_{t\in[\lambda]}\}_{i:T_i=1}) = 1 - P(Q(x)).$$

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, Lemma 2.9 implies that there exists a simulator Sim such that

$$\Pr\left[V(y, T, \mathsf{aux}) = 1 : \begin{array}{c}(y, \mathsf{st}) \leftarrow \mathsf{Sim}[\mathsf{B}] \\ T \leftarrow \{0,1\}^{\binom{\lambda^2}{\lambda}} \\ \mathsf{aux} \leftarrow \mathsf{Sim}[\mathsf{B}](T, \mathsf{st})\end{array}\right] = \mathsf{non\text{-}negl}(\lambda).$$

Moreover, by definition (Lemma 2.9), $\mathsf{Sim}[\mathsf{B}]$ 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 4.59.

Finally, the fact that Lemma 4.60 implies Lemma 4.30 and Lemma 4.61 implies Lemma 4.31 can be shown in exactly the same way, by defining the appropriate predicate V. This completes the proof.

□

5 Quantum State Obfuscation

In this section, we present our construction of quantum state obfuscation in the classical oracle model. We begin with a technical overview, extending the discussion from Section 1.3. In Section 5.2 we construct publicly-verifiable linearly-homomorphic authentication of quantum data, a key tool in our obfuscation scheme. Then, in Section 5.3, we discuss a compiler for quantum computation that outputs what we call a "linear+measurement" quantum program, which has a convenient form that we take advantage of in our obfuscation construction. Finally, in Section 5.4 we construct and prove the security of our quantum state obfuscation scheme.

5.1 Technical overview

During this overview, we'll slowly build up to our construction of quantum state obfuscation, highlighting the main ideas along the way. But first, it may be useful to convey a high level feel for the construction. To obfuscate a quantum program $(|\psi\rangle, C)$ that implements the computation $x \to Q(x)$, we first encode the state

$$|\widetilde{\psi}\rangle \leftarrow \mathsf{Enc}_k\left(|\psi\rangle\right)$$

using a novel quantum authentication scheme (QAS) that we design with particular properties in mind. Next, we compile C into what we call a "linear + measurement", or LM, quantum program. Such programs consist solely of operations that can be performed on data authenticated with our QAS. Finally, we prepare a sequence of classical oracles $\mathsf{F}_1, \ldots, \mathsf{F}_t, \mathsf{G}$, where t is the number of "measurement layers" in the LM quantum program. The oracles $\mathsf{F}_1, \ldots, \mathsf{F}_t$ are designed to help the evaluator implement an encrypted sequence of adaptive measurements. They output random labels encoding the measurement results, which are then fed into downstream oracles. The oracle G is designed to return the output $Q(x)$ if the evaluation was performed honestly. The final obfuscation then consists of the state $|\widetilde{\psi}\rangle$ and the oracles $\mathsf{F}_1, \ldots, \mathsf{F}_t, \mathsf{G}$. We will describe each of these pieces and how they fit together in more detail.

We begin this technical overview by presenting our quantum authentication scheme (Section 5.1.1). Next, we discuss the notion of LM quantum programs, and describe a compiler that writes any quantum program as an LM quantum program (Section 5.1.2). Next, we show how to use these building blocks to construct a garbling scheme and then a full-fledged obfuscation scheme for quantum computation (Section 5.1.3), and mention a couple of key ideas behind proving security. We defer a more detailed proof overview to Section 5.4.2.

5.1.1 Quantum authentication from random subspaces

Encode-encrypt authentication. Our starting point is the notion of an "encode-encrypt" authentication scheme, as defined by Broadbent, Gutoski and Stebila [BGS13]. Such

schemes are parameterized by a family of CSS codes $\mathscr{C}$, and operate as follows. To encode a qubit $|\psi\rangle$, sample a random code $C \leftarrow \mathscr{C}$ from the family, sample a quantum one-time pad key (x, z), and output the "encoded-and-encrypted" state $X^x Z^x C |\psi\rangle$. As discussed by [BGS13], various choices of the code family give rise to popular quantum authentication schemes (QAS), e.g., the polynomial scheme used for multi-party quantum computation [BOCG+06] and verifiable delegation [ABOEM18], and the trap code used for quantum one-time programs [BGS13] and zero-knowledge proofs for QMA [BJSW16].

Our instantiation. A crucial aspect of obfuscation that does not arise in these other settings is the need to preserve security when we allow the adversary to access the verifier of the authentication scheme an *a-priori unbounded* number of times. Indeed, the oracles released as part of our obfuscation scheme include subroutines that perform checks on authenticated data, and hence implicitly give the adversary reusable access to the verifier. This requirement of "public-verifiability" is not always satisfied by encode-encrypt schemes: for example, the trap code is completely insecure in this setting, as it is possible to learn the location of the traps via repeated queries to the verifier.

While certain flavors of public-verifiability have been considered previously in the quantum authentication literature (e.g. [DS18, GYZ17]), we find that a particularly simple instantiation of the encode-encrypt framework suffices for us: sample a random subspace S, a random shift Δ, and use the CSS code defined by the isometry $E_{S,\Delta}$ that maps $|0\rangle \to |S\rangle, |1\rangle \to |S + \Delta\rangle$.[33] That is, to encode an n-qubit state $|\psi\rangle$, sample a key $k = (S, \Delta, x, z)$ where S is a λ-dimensional subspace of $\mathbb{F}_2^{2\lambda+1}$, $\Delta \in \mathbb{F}_2^{2\lambda+1} \setminus S$, and $x, z \in \{0,1\}^{n\cdot(2\lambda+1)}$, and output

$$|\widetilde{\psi}\rangle = X^x Z^z E_{S,\Delta}^{\otimes n} |\psi\rangle \coloneqq \mathsf{Enc}_k(|\psi\rangle).$$

Beyond satisfying a natural notion of public-verifiability (which will be discussed below), the resulting QAS satisfies the following desirable properties: (i) linear-homomorphism, and (ii) classically-decodable standard and Hadamard basis measurements (that is, a classical machine can decode the results of standard and Hadamard basis measurements performed on authenticated data). We note that these latter properties are in fact endemic to encode-encrypt schemes (see discussion in [BGS13]), but we confirm them here for completeness.

Useful properties. First, since S is a subspace, one can confirm that CNOTs are transversal for this scheme as long as the same (S, Δ) is used to encode each qubit. That is, applying $2\lambda + 1$ CNOT gates qubit-wise to an encoding of b_1 and b_2 yields

$$X^{x_1,x_2} Z^{z_1,z_2} |S + b_1 \cdot \Delta\rangle |S + b_2 \cdot \Delta\rangle \to X^{x_1, x_1 \oplus x_2} Z^{z_1 \oplus z_2, z_2} |S + b_1 \cdot \Delta\rangle |S + (b_1 \oplus b_2) \cdot \Delta\rangle \ ,$$

which is indeed an encoding of the output of the CNOT operation using quantum one-time pad keys $(x_1, z_1 \oplus z_2)$ and $(x_1 \oplus x_2, z_2)$. Thus, an evaluator can apply any sequence of

[33] Here, we use the standard subspace state notation: for an (affine) subspace S, $|S\rangle \propto \sum_{s \in S} |s\rangle$.

CNOT gates, which we refer to as a "linear"[34] function, to authenticated data, as long as the decoder performs the analogous updates to their one-time pad keys.

Next, we note that standard basis measurements of an encoded qubit $X^x Z^z(\alpha |S\rangle + \beta |S+\Delta\rangle)$ can be decoded *classically*. Indeed, any vector in $S + x$ can be interpreted as a 0, while any vector in $S + \Delta + x$ can be interpreted as a 1.

Finally, we check that the results of a Hadamard basis measurement can also be decoded *classically*. To do so, we'll define the "primal" codespace $S_\Delta := S \cup (S + \Delta)$, and define the "dual" codespace to consist of $\widehat{S} := S_\Delta^\perp$ and $\widehat{S} + \widehat{\Delta}$, where $\widehat{\Delta}$ is such that

$$\widehat{S}_{\widehat{\Delta}} := \widehat{S} \cup (\widehat{S} + \widehat{\Delta}) = S^\perp .$$

Then, it is not hard to check and confirm that

$$H^{\otimes(2\lambda+1)} X^x Z^z (\alpha |S\rangle + \beta |S+\Delta\rangle) = X^z Z^x \left(\frac{\alpha+\beta}{\sqrt{2}} |\widehat{S}\rangle + \frac{\alpha-\beta}{\sqrt{2}} |\widehat{S}+\widehat{\Delta}\rangle \right).$$

Thus, any vector in $\widehat{S} + z$ can be interpreted as a 0 measurement result in the Hadamard basis, and any vector in $\widehat{S} + \widehat{\Delta} + z$ can be interpreted as a 1 measurement result in the Hadamard basis.

Reusable security. Now we turn to the security of our scheme. Intuitively, we want to capture the fact that no adversary can successfully tamper with authenticated data, even given the ability to verify authenticated data. In more detail, given an authentication key $k = (S, \Delta, x, z)$, where $x = (x_1, \ldots, x_n), z = (z_1, \ldots, z_n)$, we define the following classical functionalities, which are parameterized by the key k and a choice of bases $\theta \in \{0,1\}^n$.

- $\mathsf{Dec}_{k,\theta}(\widetilde{v})$: On input a tuple of vectors $\widetilde{v}$ parsed as $(\widetilde{v}_1, \ldots, \widetilde{v}_n)$, the decoding algorithm defines $v \in \{0,1\}^n$ as follows. For each $i \in [n]$:

$$\text{if } \theta_i = 0 : v_i = \begin{cases} 0 \text{ if } \widetilde{v}_i \in S + x_i \\ 1 \text{ if } \widetilde{v}_i \in S + \Delta + x_i \\ \perp \text{ otherwise} \end{cases} \qquad \text{if } \theta_i = 1 : v_i = \begin{cases} 0 \text{ if } \widetilde{v}_i \in \widehat{S} + z_i \\ 1 \text{ if } \widetilde{v}_i \in \widehat{S} + \widehat{\Delta} + z_i \\ \perp \text{ otherwise} \end{cases} .$$

  If $v_i = \perp$ for some i, then output $\perp$, and otherwise output v.

- $\mathsf{Ver}_{k,\theta}(\widetilde{v})$: On input a tuple of vectors $\widetilde{v}$ parsed as $(\widetilde{v}_1, \ldots, \widetilde{v}_n)$, the verification algorithm defines $v \in \{\top, \perp\}^n$ as follows.

$$\text{if } \theta_i = 0 : v_i = \begin{cases} \top \text{ if } \widetilde{v}_i \in S_\Delta + x_i \\ \perp \text{ otherwise} \end{cases} \qquad \text{if } \theta_i = 1 : v_i = \begin{cases} \top \text{ if } \widetilde{v}_i \in \widehat{S}_{\widehat{\Delta}} + z_i \\ \perp \text{ otherwise} \end{cases} .$$

  If $v_i = \perp$ for some i, then output $\perp$, and otherwise output $\top$.

[34] Of course, all quantum gates are linear with respect to the ambient Hilbert space of exponential dimension. Here, linearity specifically refers to the fact that any sequence of CNOT gates applies a linear function over $\mathbb{F}_2$ to each standard basis vector.

That is, the verification algorithm just checks whether its inputs lie in the primal (resp. dual) codespace, while the decoding algorithm additionally computes the logical bits encoded by its inputs. We show that for any state $|\psi\rangle$, sequence of measurement bases $\theta \in \{0,1\}^n$, and adversarial measurement Adv that samples

$$\widetilde{v} \leftarrow \mathsf{Adv}^{\mathsf{Ver}_{k,\cdot,\cdot}(\cdot)}\left(X^x Z^z E_{S,\Delta}^{\otimes n} |\psi\rangle\right),$$

the decoded value $v \leftarrow \mathsf{Dec}_{k,\theta}(\widetilde{v})$ is either $\perp$, or its distribution is very close in total variation distance to the distribution that results from directly measuring $|\psi\rangle$ in the bases θ.

In fact, we also consider the possibility that the adversary is supposed to homomorphically apply some sequence of CNOT gates (that is, a linear function L) to the authenticated data before measuring. Thus, in full generality we also parameterize the decoding $\mathsf{Dec}_{k,\theta,L}$ and verification $\mathsf{Ver}_{k,\theta,L}$ algorithms by a linear function L, which determines an updated sequence of one-time pad keys $(x_{L,1},\ldots,x_{L,n}),(z_{L,1},\ldots,z_{L,n})$ to be used in the decoding and verification.

A word on the proof of security. Our proof combines two useful tricks from the literature: superspace sampling ([Zha19, CLLZ21]) and the Pauli twirl [ABOEM18]. Briefly, our first step is to sample random (say, $(3\lambda/2+1)$-dimensional) superspaces $R \supset S_\Delta$, $\widehat{R} \supset \widehat{S}_{\widehat{\Delta}}$ and use $(R,\widehat{R})$ in lieu of $(S_\Delta, \widehat{S}_{\widehat{\Delta}})$ in the definition of the oracle $\mathsf{Ver}_{k,\cdot,\cdot}(\cdot)$. Since R and $\widehat{R}$ are random and small enough compared to the ambient space, the adversary cannot notice this change except with negligible probability. Next, we imagine sampling each one-time pad vector in two parts: for x_i we sample an $x_{i,R} \leftarrow R$ and an $x_{i,\mathsf{co}(R)} \leftarrow \mathsf{co}(R)$, where $\mathsf{co}(R)$ is a set of coset representatives of R, and define $x_i = x_{i,R} + x_{i,\mathsf{co}(R)}$, and for z_i we sample a $z_{i,\widehat{R}} \leftarrow \widehat{R}$ and a $z_{i,\mathsf{co}(\widehat{R})} \leftarrow \mathsf{co}(\widehat{R})$, and define $z_i = z_{i,\widehat{R}} + z_{i,\mathsf{co}(\widehat{R})}$. Finally, we consider the following experiment:

- Sample $R, \widehat{R}, \{x_{i,\mathsf{co}(R)}, z_{i,\mathsf{co}(\widehat{R})}\}_{i\in[n]}$ and give this information to the adversary in the clear. Note that this is now sufficient to implement the oracle $\mathsf{Ver}_{k,\cdot,\cdot}(\cdot)$.
- Sample random S, Δ such that $\widehat{R}^\perp \subset S \subset S_\Delta \subset R$ and $\{x_{i,R}, z_{i,\widehat{R}}\}_{i\in[n]}$ to complete the description of the authentication key (S,Δ,x,z). Send $X^x Z^z E_{S,\Delta}^{\otimes n} |\psi\rangle$ to the adversary, who mounts its attack.

At this point, we use the Pauli twirl over the space in between $\widehat{R}^\perp$ and R to show that any adversarial operation can be decomposed into a fixed linear combination of Pauli attacks. To conclude, we use the randomness of S, Δ to show that any fixed Pauli attack will either be rejected with overwhelming probability or act as the identity on the encoded qubit, which completes the proof. See Section 5.2 for the full details of our definitions, construction, and security proofs.

5.1.2 Linear + measurement quantum programs

Next, we discuss our quantum program compiler. We start with any quantum circuit written using the $\{\mathsf{CNOT}, H, T\}$ universal gate set, where H is the Hadamard gate, and T applies a phase of $e^{i\pi/4}$. With the help of magic states, we compile the circuit into an alternating sequence of layers of CNOT gates (i.e. linear functions) and partial standard and Hadamard basis measurements, which we refer to as "ZX measurements".[35] We refer to the resulting program as a linear + measurement (LM) quantum program. We note that the measurements are in fact partial in two aspects: (i) they may only operate on a subset of the qubits, and (ii) the measurement operators are projectors with rank potentially greater than 1. Furthermore, we allow the measurements to be adaptive, that is, their description may depend on previous measurement results (and the classical input to the computation).

More specifically, our goal will be to write each of the H and T gates as a sequence of CNOT gates, ZX measurements, and Pauli corrections derived from these measurement results. Then, the Pauli corrections can be commuted past future CNOT gates using the update rule $(x_1, z_1), (x_2, z_2) \to (x_1, z_1 \oplus z_2), (x_1 \oplus x_2, z_2)$, and incorporated into the description of future ZX measurements.

Handling the H gate. Following [BGS13], we prepare the two-qubit magic state

$$|\phi_H\rangle \propto |00\rangle + |01\rangle + |10\rangle - |11\rangle\,,$$

and perform the Hadamard gate as shown in Fig. 15 (Page 144), using one CNOT gate and Pauli corrections derived from a standard basis and a Hadamard basis measurement. As remarked in [BGS13], it might seem strange at first that we are replacing a Hadamard gate with a circuit that nonetheless performs a Hadamard basis measurement. However, in our setting this does represent real progress: our authentication scheme does not support applying Hadamard gates directly to authenticated data,[36] but does support the decoding of Hadamard basis measurements.

Handling the T gate. As we show on the bottom left of Fig. 16 (Page 146), the T gate can be implemented using the two magic states

$$|\phi_T\rangle \propto |0\rangle + e^{i\pi/4}\,|1\rangle \quad \text{and} \quad |\phi_{PX}\rangle \propto i\,|0\rangle + |1\rangle\,,$$

a CNOT gate, a *classically controlled CNOT gate*, and Pauli corrections.

Unfortunately, controlled CNOT is a "multi-linear" operation that we don't know how to directly implement on data authenticated with our authentication scheme. Therefore,

[35] Here, ZX is not meant to denote the composition of the Z and X operators, rather, it is meant as a shorthand for "standard + Hadamard basis".

[36] At least, while preserving its linear-homomorphism. Applying a Hadamard gate transversally to authenticated data would result in an encoding with respect to the dual subspace, which would no longer support transversal CNOTs with data encoded using the primal subspace.

taking inspiration from the "encrypted CNOT" operation introduced in [Mah18a],[37] we replace the controlled CNOT operation with a *projective measurement* Γ_c controlled on the classical control bit c, where

- $\Gamma_0 = \{|00\rangle\langle 00| + |10\rangle\langle 10|, |01\rangle\langle 01| + |11\rangle\langle 11|\}$. That is, it measures its second input in the standard basis and has no effect on its first input.
- $\Gamma_1 = \{|00\rangle\langle 00| + |11\rangle\langle 11|, |01\rangle\langle 01| + |10\rangle\langle 10|\}$. That is, it measures the XOR of its two inputs, partially collapsing both.

Note that Γ_1 can roughly be seen as applying CNOT "out of place", writing the result to a third register, and then measuring it. We also remark that both of these measurements are diagonal in the standard basis, and thus can be performed on our authenticated data.

In the implementation of the T gate shown on the bottom left of Fig. 16, the c-CNOT operation is applied to the two magic states, followed by a measurement of the second magic state wire in the standard basis, and finally a Z correction to the first magic state wire conditioned on both c and the measurement outcome. One can show that the result of these operations is identical to what is shown on the bottom right of Fig. 16: measure Γ_c on the two magic state wires, then measure the second magic state wire in the *Hadamard* basis, and finally apply a Z correction to the first magic state wire conditioned on both c and the *XOR* of the two measurement results. We make this precise in the proof of Claim 5.17, showing that our Γ_c-based implementation of the T gate works as expected.

Formalizing LM quantum programs. By combining these observations, we are able to specify any quantum program with t many T gates using an n-qubit state $|\psi\rangle$ (which in particular includes all of the necessary magic state) along with a sequence

$$L_1, M_{\theta_1, f_1^{(\cdot)}}, \ldots, L_t, M_{\theta_t, f_t^{(\cdot)}}, L_{t+1}, M_{\theta_{t+1}, g^{(\cdot)}}$$

where

- Each L_i is a sequence of CNOT gates.
- Each $M_{\theta_i, f_i^{(\cdot)}}$ (and $M_{\theta_i, g^{(\cdot)}}$) describes a partial ZX measurement in the following way:
  - $\theta_i \in \{0, 1, \perp\}^n$ defines a partial set of measurement bases. We define $\Phi_{i,0} := \{j : \theta_{i,j} = 0\}$ to be the set of registers measured in the standard basis and $\Phi_{i,1} := \{j : \theta_{i,j} = 1\}$ to be the set of registers measured in the Hadamard basis, and define $\Phi_i := (\Phi_{i,0}, \Phi_{i,1})$ to be the total set of registers on which the i'th measurement is performed.

[37] Those familiar with [Mah18a]'s encrypted CNOT may notice the parallels: in [Mah18a]'s setting, these two measurements correspond to the two types of "claws" generated by the lattice-based encryption of c.

– Each $f_i^{(\cdot)}$ is a function that assigns measurement outcomes to basis states. The superscript indicates that its description may depend on previously generated information, i.e. the classical input x to the computation and previous measurement results.

– To be precise, $M_{\theta_i, f_i^{(\cdot)}}$ can be described by the following measurement operators:

$$\left\{ H^{\Phi_{i,1}} \left(\sum_{m : f_i^{(\cdot)}(m_{\Phi_i}) = y} |m\rangle\langle m| \right) H^{\Phi_{i,1}} \right\}_y ,$$

where $H^{\Phi_{i,1}}$ applies a Hadamard gate to each qubit in the set $\Phi_{i,1}$, and m_{Φ_i} is the substring of m consisting of the indices in Φ_i.

Thus, we have written our quantum program as an alternating sequence of linear operations and partial ZX measurements. We formalize this notion of an "LM quantum program" in Definition 5.13, and provide an example diagram of an LM quantum program in Fig. 14.

However, looking ahead, it will be convenient to apply our obfuscator not to a completely arbitrary LM quantum program, but rather to an LM quantum program that satisfies a particular structural property. This property is described in Definition 5.14, and satisfied by LM quantum programs output by our compiler described above. In order to formalize such programs (and our obfuscator), it will be necessary to introduce some further notation. For the purpose of this technical overview, we will introduce the notation and show how it is applied to the concrete compiler described above, but defer further details and a formalization of the property given by Definition 5.14 to the body.

It may be helpful to refer to Fig. 16 (our implementation of the T gate) and Fig. 14 (the example LM quantum program) while reading what follows. We begin by specifying (disjoint) sets $V_1, \ldots, V_{t+1}$ and (disjoint) sets $W_1, \ldots, W_t$ with the following properties.

- $\Phi_1 = (V_1, W_1), \Phi_2 = (V_1, V_2, W_2), \ldots, \Phi_t = (V_1, \ldots, V_t, W_t), \Phi_{t+1} = (V_1, \ldots, V_{t+1}) = [n]$.
- V_i are the set of registers that are *fully* collapsed in the standard or Hadamard basis by the i'th measurement. Concretely, V_i consists of the 3rd wire of the $(i-1)$'th T-gate circuit (Fig. 16), 1st and 2nd wires of any H-gate circuit (Fig. 15) in the i'th layer, and 1st wire of the i'th T-gate circuit (Fig. 16).
- W_i are the set of registers that are *partially* collapsed by the i'th measurement. Concretely, W_i consists of the two magic state wires used for the i'th T gate circuit. Indeed, the i'th measurement applies the controlled measurement Γ_{c_i} to these wires, which only partially collapses them.

Then, we are able to specify further details about the $f_i^{(\cdot)}$ and $g^{(\cdot)}$ measurements.

- Each $f_i^{(\cdot)}$ takes as input some sequence $(v_1, \ldots, v_i, w_i)$ and outputs v_i (fully collapsing the V_i registers) along with a bit r_i (partially collapsing the W_i registers).

- In order to compute the bit r_i, the function $f_i^{(\cdot)}$ first needs to compute the i'th control bit c_i, which may depend on the input x and all previous measurement results $(v_1, \ldots, v_i, r_1, \ldots, r_{i-1})$. While we are able to provide $f_i^{(\cdot)}$ with the values $v_1, \ldots, v_{i-1}$ on registers $V_1, \ldots, V_{i-1}$ (which have been collapsed by previous measurements), this is not the case for the bits $r_1, \ldots, r_{i-1}$, since the $W_1, \ldots, W_{i-1}$ registers may have been computed on since previous measurements. To handle this, we remember the previous results $r_1, \ldots, r_{i-1}$, and paramaterize $f_i^{x,r_1,\ldots,r_{i-1}}$ by the input x and previously generated bits $r_1, \ldots, r_{i-1}$. Thus, the actual measurements are specified *adaptively* using the previously generated bits $r_1, \ldots, r_{i-1}$.

- In a similar manner, the function $g^{x,r_1,\ldots,r_t}$ is parameterized by the input x and previously generated bits $r_1, \ldots, r_t$. It takes as input some sequence $(v_1, \ldots, v_{t+1})$ and instead of performing an intermediate measurement, it computes the final output $y = Q(x)$.

Finally, we have set up enough notation to start discussing our actual obfuscation construction, which follows.

5.1.3 Obfuscation construction

So far, we have discussed a method for authenticating quantum states $|\widetilde{\psi}\rangle = \mathsf{Enc}_k(|\psi\rangle)$ using key $k = (S, \Delta, x, z)$, and a method for writing any quantum program as

$$|\psi\rangle, L_1, M_{\theta_1, f_1^{(\cdot)}}, \ldots, L_t, M_{\theta_t, f_t^{(\cdot)}}, L_{t+1}, M_{\theta_{t+1}, g^{(\cdot)}}$$

where $|\psi\rangle$ consists of a quantum state that was part of the description of the original program, as well as some magic states.

Garbling via encrypted measurements. We will build up to our full obfuscation construction by first describing how to *garble* quantum circuits using our approach. That is, we'll suppose that the evaluator is only interested in computing the output on a particular input x, and show how to design oracles $\mathsf{F}_1[x], \ldots, \mathsf{F}_t[x], \mathsf{G}[x]$ that, along with the authenticated state $|\widetilde{\psi}\rangle = \mathsf{Enc}_k(|\psi\rangle)$, allow the evaluator to perform the entire computation on top of authenticated data, and eventually obtain $Q(x)$ without learning anything else about the program's implementation.

The basic idea is to implement the measurement $M_{\theta_i, f_i^{(x,\cdot)}}$ on authenticated data using an oracle $\mathsf{F}_i[x]$, that, instead of outputting the results (v_i, r_i) in the clear, outputs the encoded version $\widetilde{v}_i$ of v_i (that is, $\widetilde{v}_i$ is in the support of the authenticated state that encodes the logical string v_i) along with a random *label* ℓ_i representing the bit r_i. We will always

denote vectors that result from measuring authenticated states (but not decoding) with a tilde (e.g. $\widetilde{v}_i$).

Roughly $\mathsf{F}_i[x]$ will be implemented as follows. It takes as input vectors $\widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i$ from the support of authenticated states (obtained from authenticated wires $V_1, \ldots, V_i, W_i$), as well as labels $\ell_1, \ldots, \ell_{i-1}$ that encode the results $r_1, \ldots, r_{i-1}$ of previous measurements. It first decodes its inputs, and then uses the decoded values to compute the next measurement results (v_i, r_i). Finally, it outputs the encodings $(\widetilde{v}_i, \ell_i)$ where ℓ_i is a label for r_i computed via a random oracle H.

We will implement $\mathsf{G}[x]$ in exactly the same way, except that it directly outputs the result y. Sketches of these oracles follow.

$\underline{\mathsf{F}_i[x](\widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1})}$

1. $(v_1, \ldots, v_i, w_i) \leftarrow \mathsf{Dec}_{k,\theta_i,L_i \ldots L_1}(\widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i)$.[38] Abort if the output is $\bot$.
2. For each $\iota \in [i-1]$, let

$$\ell_{\iota,0} = H(\widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 0), \quad \ell_{\iota,1} = H(\widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 1),$$

   and let r_ι be the bit such that $\ell_\iota = \ell_{\iota,r_\iota}$, or abort if there is no such bit.
3. Compute $(v_i, r_i) = f_i^{x,r_1,\ldots,r_{i-1}}(v_1, \ldots, v_i, w_i)$.
4. Set $\ell_i \coloneqq H(\widetilde{v}_1, \ldots, \widetilde{v}_i, \ell_1, \ldots, \ell_{i-1}, r_i)$, and output $(\widetilde{v}_i, \ell_i)$.

$\underline{\mathsf{G}[x](\widetilde{v}_1, \ldots, \widetilde{v}_{t+1}, \ell_1, \ldots, \ell_t)}$

1. $(v_1, \ldots, v_{t+1}) \leftarrow \mathsf{Dec}_{k,\theta_i,L_{t+1} \ldots L_1}(\widetilde{v}_1, \ldots, \widetilde{v}_{t+1})$. Abort if the output is $\bot$.
2. For each $\iota \in [t]$, let

$$\ell_{\iota,0} = H(\widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 0), \quad \ell_{\iota,1} = H(\widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 1),$$

   and let r_ι be the bit such that $\ell_\iota = \ell_{\iota,r_\iota}$, or abort if there is no such bit.
3. Compute and output $y = g^{x,r_1,\ldots,r_t}(v_1, \ldots, v_{t+1})$.

Proving the security of this garbled program consists of two main steps: (1) a "soundness" argument establishing that no adversary, given $|\widetilde{\psi}\rangle$ and oracle access to $\mathsf{F}_1[x], \ldots, \mathsf{F}_t[x]$ should be able to output classical strings $(\widetilde{v}_1, \ldots, \widetilde{v}_{t+1}, \ell_1, \ldots, \ell_t)$ such that $\mathsf{G}[x](\widetilde{v}_1, \ldots, \widetilde{v}_{t+1}, \ell_1, \ldots, \ell_t) \notin \{Q(x), \bot\}$, and (2) a "simulation" argument establishing that the $\mathsf{F}_1[x], \ldots, \mathsf{F}_t[x]$ oracles can be simulated using a verification oracle $\mathsf{Ver}_{k,\cdot,\cdot}(\cdot)$ for the authentication scheme *instead of* the decoding functionality $\mathsf{Dec}_{k,\cdot,\cdot}(\cdot)$. Indeed, a common theme throughout our

[38] Recall the description of the decoding oracle Dec from Section 5.1.1. We additionally parameterize the oracle with a concatenation of the linear functions $L_i \ldots L_1$, which determines the sequence of Pauli one-time-pad keys to be used during the decoding.

proof strategy is understanding how we can replace $\mathsf{Dec}_{k,\cdot,\cdot}(\cdot)$ with $\mathsf{Ver}_{k,\cdot,\cdot}(\cdot)$ so that we can then appeal to the security of the authentication scheme. Further discussion on these two steps can be found in our proof intuition section, Section 5.4.2. Here, we just mention that the main idea for the soundness argument is an inductive strategy, where we perform the first measurement and appeal to soundness of a garbled program with one fewer measurement layer.

From garbling to obfuscation via signature tokens. To complete our construction of full-fledged obfuscation, it remains to show how to grant the evaluator the ability to execute the circuit on *any* input x of its choice, without risking any other leakage on the description of the program. A natural idea is to re-define $\mathsf{F}_1, \ldots, \mathsf{F}_t, \mathsf{G}$ so that they additionally take x as input, and include x in the hashes $H(x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \ell_1, \ldots, \ell_{i-1}, r_i)$ that define the output labels. We intuitively want to sample different output labels for each x so that the resulting obfuscation scheme can roughly be seen as a "concatenation" of *independently sampled* garbling schemes for each x (that share the same initial authenticated state). However, it turns out that this is not yet enough to ensure security.

Indeed, nothing is preventing the adversary from applying a type of "mixed input" attack, where they evaluate honestly on an input x, but at some point insert a measurement implemented by the oracle $\mathsf{F}_i(x', \cdot)$ on some input $x' \neq x$. That is, at layer i, the adversary would first implement $\mathsf{F}_i(x', \cdot)$ (and ignore the output labels) to collapse the state in some way before continuing with their honest evaluation procedure using $\mathsf{F}_i(x, \cdot)$. Unfortunately, this rogue call to $\mathsf{F}_i(x', \cdot)$ wouldn't destroy the current state enough to cause the remaining oracle calls to $\mathsf{F}_i(x, \cdot), \ldots, \mathsf{F}_t(x, \cdot), \mathsf{G}(x, \cdot)$ to abort, but *might* collapse the state in a manner inconsistent with an honest evaluation on input x, eventually allowing the adversary to break the "soundness" of the scheme by finding an input to $\mathsf{G}(x, \cdot)$ that results in an output $y \neq Q(x)$.

Taking inspiration from [BKNY23] who faced a similar issue, we solve our problem via the use of *signature tokens* [BS16]. This quantum cryptographic primitive consists of a quantum signing key $|\mathsf{sk}\rangle$ that may be used to produce a classical signature σ_x on any *single* message x but never *two* signatures $\sigma_x, \sigma_{x'}$ on two different messages x, x' simultaneously. We include a quantum signing key $|\mathsf{sk}\rangle$ as part of our obfuscation construction, and re-define the oracles $\mathsf{F}_1, \ldots, \mathsf{F}_t, \mathsf{G}$ to take x *and* a signature σ_x as input, abort if the signature is invalid, and otherwise include both in the hashes $H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \ell_1, \ldots, \ell_{i-1}, r_i)$ that define the output labels.

Intuitively, this prevents the above attack. Once the adversary has begun an honest evaluation on some input x, it must "know" some valid signature σ_x, preventing it from querying the oracles on any input that starts with $(x', \sigma_{x'})$. That is, if it actually want to evaluate on x', it must uncompute everything it has computed so far to return to $|\mathsf{sk}\rangle$ before it can produce a signature $\sigma_{x'}$ and begin evaluating on x'.

We provide more details about this approach in the proof intuition section, Section 5.4.2. Of note is the fact that we crucially use a *purified* random oracle [Zha19] in order to extract a signature token on x from any adversary who has begun evaluating the obfuscated

program on x. Formalizing this approach is one of trickier aspects of the proof, and we describe a toy problem in Section 5.4.2 that may provide some intuition for the actual proof.

5.2 Publicly-verifiable quantum authentication

In this section, we introduce the notion of a "publicly-verifiable linearly-homomorphic QAS (Quantum Authentication Scheme) with classically-decodable ZX measurements." We then provide a construction and security proof.

5.2.1 Definitions

The following notation will be heavily referenced both throughout this section, and throughout the remainder of the paper.

Partial ZX measurements. Given a string $\theta \in \{0,1,\perp\}^n$, define sets

$$\Phi_\theta \coloneqq \{i : \theta_i \neq \perp\},\ \ \Phi_{\theta,0} = \{i : \theta_i = 0\},\ \ \Phi_{\theta,1} = \{i : \theta_i = 1\},\ \ \Phi_{\theta,\perp} \coloneqq \{i : \theta_i = \perp\}.$$

We will often write $\Phi, \Phi_0, \Phi_1, \Phi_\perp$ instead of $\Phi_\theta, \Phi_{\theta,0}, \Phi_{\theta,1}, \Phi_{\theta,\perp}$ when the choice of θ is clear from context. The string θ will be used to denote the basis of a partial measurement on n qubits, where the 0 indices are measured in the standard basis and the 1 indices are measured in the Hadamard basis. We will also need the following notation.

- Given a string $m \in \{0,1\}^n$ and a set $\Phi \subseteq [n]$, let m_Φ denote the substring of m consisting of bits $\{m_i\}_{i\in\Phi}$.
- Given $\theta \in \{0,1,\perp\}^n$ and a set $\Phi \subseteq [n]$, define $\theta[\Phi]$ to be equal to θ on indices in V and $\perp$ everywhere else.
- Given a register $\mathcal{M}$, an operation U on $\mathcal{M}$, and a subset $\Phi \subseteq [n]$, let U^Φ be the operation on $\mathcal{M}^{\otimes n}$ that applies U to the i'th copy of $\mathcal{M}$ for each $i \in \Phi$.

For any $\theta \in \{0,1,\perp\}^n$ and classical function $f : \{0,1\}^{|\Phi|} \to \{0,1\}^*$, let $M_{\theta,f}$ be the projective measurement on n qubits defined by the operators

$$\left\{ H^{\Phi_1} \left(\sum_{m : f(m_\Phi) = y} |m\rangle\langle m| \right) H^{\Phi_1} \right\}_y .$$

For any n-qubit register $\mathcal{M}$, we write

$$M_{\theta,f}(\mathcal{M}) \to \mathcal{M}, y$$

to refer to the operation that measures $\mathcal{M}$ according to $M_{\theta,f}$ and then writes the classical result y to a new register. Sometime we will write $M_{\theta,f}(\mathcal{M}) \to y$ to denote just the classical measurement result y.

Linear operations. We will use L to denote a sequence of CNOT gates on n qubits, which we refer to as a *linear operation*. While all quantum gates are linear with respect to the ambient Hilbert space of exponential dimension, here linearity specifically refers to the fact that any sequence of CNOT gates applies a linear function over $\mathbb{F}_2$ to each standard basis vector. In an abuse of notation, L will either refer to the classical description of a series of CNOT gates or to the actual unitary operation that applies these gates. Which case should be clear from context.

Syntax. A publicly-verifiable, linearly-homomorphic quantum authentiation scheme (QAS) with classically-decodable ZX measurements has the following syntax. Let $p = p(\lambda)$ be a polynomial.

- $\mathsf{Gen}(1^\lambda, n) \to k$: The key generation algorithm takes as input a security parameter 1^λ and number of qubits $n = \mathrm{poly}(\lambda)$, and outputs an authentication key k.
- $\mathsf{Enc}_k(\mathcal{M}) \to \mathcal{C}$: The encoding algorithm is an isometry parameterized by an authentication key k that maps a state on an n-qubit register $\mathcal{M} \coloneqq \mathcal{M}_1 \otimes \cdots \otimes \mathcal{M}_n$ to a state on an np-qubit register $\mathcal{C} \coloneqq \mathcal{C}_1 \otimes \cdots \otimes \mathcal{C}_n$, where each $\mathcal{C}_i$ is an p-qubit register.
- $\mathsf{LinEval}_L(\mathcal{C}) \to \mathcal{C}$: The linearly-homomorphic evaluation procedure is a unitary parameterized by a linear operation L that operates on register $\mathcal{C}$.
- $\mathsf{Dec}_{k,L,\theta}(c) \to m \cup \{\bot\}$: The classical decoding algorithm is parameterized by an authentication key k, a linear operation L, and a choice of bases $\theta \in \{0,1,\bot\}^n$. It takes as input a string $c \in \{0,1\}^{|\Phi| \cdot p}$ and outputs either a classical string $m \in \{0,1\}^{|\Phi|}$ or $\bot$.
- $\mathsf{Ver}_{k,L,\theta}(c) \to \{\top, \bot\}$: The classical verification algorithm is identical to Dec except that whenever Dec outputs an $m \neq \bot$, Ver outputs $\top$.

Partial ZX measurements on authenticated states. First, given the parameter p, define

$$\widetilde{\Phi} \coloneqq \bigcup_{i \in \Phi} \{(i-1)p + 1, \ldots, ip\} \subseteq [np].$$

That is, $\widetilde{\Phi}$ contains the i'th chunk of p indices for each $i \in \Phi$. Define $\widetilde{\Phi}_0, \widetilde{\Phi}_1, \widetilde{\Phi}_\bot$ analogously. For any $\theta \in \{0,1,\bot\}^n$, classical function $f : \{0,1\}^{|\Phi|} \to \{0,1\}^*$, authentication key k, and linear operation L, let $\widetilde{M}_{\theta,f,k,L}$ be the projective measurement on np qubits defined by the operators

$$\left\{ H^{\widetilde{\Phi}_1} \left(\sum_{c : f\left(\mathsf{Dec}_{k,L,\theta}\left(c_{\widetilde{\Phi}}\right)\right) = y} |c\rangle\langle c| \right) H^{\widetilde{\Phi}_1} \right\}_y \cup \left\{ H^{\widetilde{\Phi}_1} \left(\sum_{c : \mathsf{Dec}_{k,L,\theta}\left(c_{\widetilde{\Phi}}\right) = \bot} |c\rangle\langle c| \right) H^{\widetilde{\Phi}_1} \right\}.$$

For any np-qubit register $\mathcal{C}$, we write

$$\widetilde{M}_{\theta,f,k,L}(\mathcal{C}) \to \mathcal{C}, y$$

to refer to the operation that measures $\mathcal{C}$ according to $M_{\theta,f,k,L}$ and then writes the classical result y to a new register. Sometimes we will write $\widetilde{M}_{\theta,f,k,L}(\mathcal{C}) \to y$ to denote just the classical measurement result y.

Correctness. Our definition of correctness roughly states that encoding, applying a linear homomorphism, and then applying a partial measurement to the encoded state is equivalent to first applying the linear operation, applying the partial measurement, and then encoding. This definition supports composition of multiple partial measurements on encoded data, which will be necessary for our application to obfuscation.

Definition 5.1 (Correctness)**.** *A publicly-verifiable linearly-homomorphic QAS with classically-decodable ZX measurements is* correct *if the following holds. For any linear operation L, bases $\theta \in \{0,1,\perp\}^n$, $f : \{0,1\}^{|\Phi|} \to \{0,1\}^*$, and $k \in \mathsf{Gen}(1^\lambda, n)$,*

$$\mathsf{LinEval}_L^\dagger \circ \widetilde{M}_{\theta,f,k,L} \circ \mathsf{LinEval}_L \circ \mathsf{Enc}_k = \mathsf{Enc}_k \circ L^\dagger \circ M_{\theta,f} \circ L.$$

Note that both sequences of operations above map $\mathcal{M} \to (\mathcal{C}, y)$, where $\mathcal{M}$ is an n-qubit register, $\mathcal{C}$ is an np-qubit register, and y is a classical measurement outcome.

Security Next, we formalize two security properties. The first roughly states that no adversary with access to the verification oracle can change the distribution resulting from a partial measurement on the encoded state.

Definition 5.2 (Security)**.** *A publicly-verifiable linearly-homomorphic QAS with classically-decodable ZX measurements is* secure *if the following holds. For any linear operation L, bases $\theta \in \{0,1,\perp\}^n$, $f : \{0,1\}^{|\Phi|} \to \{0,1\}^*$, and oracle-aided adversary $A : \mathcal{C} \to \mathcal{C}$, there exists an $\epsilon(\lambda) \in [0,1]$ such that for any n-qubit state $|\psi\rangle$,*

$$\left\{ y : \begin{array}{r} k \leftarrow \mathsf{Gen}(1^\lambda, n) \\ y \leftarrow \widetilde{M}_{\theta,f,k,L} \circ A^{\mathsf{Ver}_{k,\cdot,\cdot}(\cdot)} \circ \mathsf{Enc}_k(\mathcal{M}) \end{array} \right\} \approx_{2^{-\Omega(\lambda)}} (1-\epsilon(\lambda))\{y : y \leftarrow M_{\theta,f} \circ L(\mathcal{M})\} + \epsilon(\lambda)\{\perp\}.$$

Remark 5.3. *Although the adversary A is defined as a (oracle-aided) general quantum map from $\mathcal{C} \to \mathcal{C}$, we can without loss of generality take it to be a (oracle-aided) unitary that additionally operates on some workspace register $\mathcal{A}$ initialized to $|0\rangle$. We leave the workspace register $\mathcal{A}$ implicit when writing $y \leftarrow \widetilde{M}_{\theta,f,k,L} \circ A^{\mathsf{Ver}_{k,\cdot,\cdot}(\cdot)} \circ \mathsf{Enc}_k(\mathcal{M})$, and note that $\widetilde{M}_{\theta,f,k,L}$ only operates on $\mathcal{C}$.*

Next, we describe a weaker security property that is immediately implied by Definition 5.2, but will be convenient to use in our application to obfuscation.

Definition 5.4 (Mapping Security)**.** *For any linear operation L, bases $\theta \in \{0,1,\perp\}^n$, $f : \{0,1\}^{|\Phi|} \to \{0,1\}^*$, n-qubit state $|\psi\rangle$, and set $B \subset \{0,1\}^*$ such that*

$$\Pr[y \in B : y \leftarrow M_{\theta,f} \circ L(|\psi\rangle)] = 0,$$

it holds that for any oracle-aided adversary $A : \mathcal{C} \to \mathcal{C}$,

$$\Pr\left[y \in B : \begin{array}{r} k \leftarrow \mathsf{Gen}(1^\lambda, 1^n) \\ y \leftarrow \widetilde{M}_{\theta,f,k,L} \circ A^{\mathsf{Ver}_{k,\cdot,\cdot}(\cdot)} \circ \mathsf{Enc}_k(|\psi\rangle) \end{array}\right] = 2^{-\Omega(\lambda)}.$$

Finally, we define a notion of privacy, which states that any two encoded states are indistinguishable, even given the verification oracle.

Definition 5.5 (Privacy)**.** *For any n-qubit states $|\psi_0\rangle, |\psi_1\rangle$ and oracle-aided binary outcome projector D,*

$$\left| \Pr_{k \leftarrow \mathsf{Gen}(1^\lambda,n)} \left[1 \leftarrow D^{\mathsf{Ver}_{k,\cdot,\cdot}(\cdot)} \circ \mathsf{Enc}_k(|\psi_0\rangle)\right] - \Pr_{k \leftarrow \mathsf{Gen}(1^\lambda,n)} \left[1 \leftarrow D^{\mathsf{Ver}_{k,\cdot,\cdot}(\cdot)} \circ \mathsf{Enc}_k(|\psi_1\rangle)\right] \right| = 2^{-\Omega(\lambda)}.$$

5.2.2 Construction

Paulis and updates. We specify several notational conventions regarding sets $\{x_i\}_{i\in[n]}, \{z_i\}_{i\in[n]}$ that describe Pauli corrections on n registers.

- As n will be clear from context, let $x \coloneqq (x_1, \ldots, x_n)$ and $z \coloneqq (z_1, \ldots, z_n)$.
- Given a linear operation L on n qubits, let $L(x,z) \coloneqq (x_L, z_L)$ be the result of starting with (x,z), and, for each CNOT gate in L, sequentially applying the CNOT update rule $(x_i, z_i), (x_j, z_j) \to (x_i, z_i \oplus z_j), (x_i \oplus x_j, z_j)$. Note that this is yet another interpretation for L, which in another context could refer to the unitary that applies the sequence of CNOT gates.
- Let L^{-1} be the inverse of L, and note that $L^{-1}(x_L, z_L) = (x, z)$.
- Given $x = (x_1, \ldots, x_n)$ or $x_L = (x_{L,1}, \ldots, x_{L,n})$ and a subset $\Phi \subseteq [n]$, let $x_\Phi \coloneqq \{x_i\}_{i\in\Phi}$ and $x_{L,\Phi} \coloneqq \{x_{L,i}\}_{i\in\Phi}$. Given disjoint sets $\Phi_0, \Phi_1 \subset [n]$, we let x_{Φ_0}, x_{Φ_1} refer to the union $\{x_i\}_{i\in\Phi_0} \cup \{x_i\}_{i\in\Phi_1}$.

Subspaces. Given a λ-dimensional subspace $S \subset \mathbb{F}_2^{2\lambda+1}$ and a vector $\Delta \in \mathbb{F}_2^{2\lambda+1} \setminus S$, define the $(\lambda+1)$-dimensional subspace

$$S_\Delta \coloneqq S \cup (S + \Delta).$$

Let the dual subspace of S_Δ be $\widehat{S} \coloneqq S_\Delta^\perp$; note that

- $\widehat{S}$ is λ-dimensional; and

- since $S_\Delta \supset S$, its dual subspace $\widehat{S} := S_\Delta^\perp \subset S^\perp$.

Let $\widehat{\Delta}$ be an arbitrary choice of a vector such that $S^\perp = \widehat{S} \cup (\widehat{S} + \widehat{\Delta})$, and define

$$\widehat{S}_{\widehat{\Delta}} := S^\perp = \widehat{S} \cup (\widehat{S} + \widehat{\Delta}) \,.$$

Given a subspace S, define the state

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

and note that

$$H^{\otimes 2\lambda+1} |S\rangle = |S^\perp\rangle \,.$$

Next, given any λ-dimensional subspace S and $\Delta \notin S$, define the isometry $E_{S,\Delta}$ from 1 qubit to $2\lambda+1$ qubits that maps $|0\rangle \to |S\rangle$ and $|1\rangle \to |S+\Delta\rangle$.

Publicly-verifiable linearly-homomorphic QAS with classically-decodable ZX measurements

- $\mathsf{Gen}(1^\lambda, n)$: Sample a uniformly random λ-dimensional subspace $S \subset \mathbb{F}_2^{2\lambda+1}$, vector $\Delta \leftarrow \mathbb{F}_2^{2\lambda+1} \setminus S$, and $x_i, z_i \leftarrow \mathbb{F}_2^{2\lambda+1}$ for each $i \in [n]$. Output $k := (S, \Delta, x, z)$.
- $\mathsf{Enc}_k = X^x Z^z E_{S,\Delta}^{\otimes n}$.
- $\mathsf{LinEval}_L(\mathcal{C})$: Parse register $\mathcal{C} = \mathcal{C}_1 \otimes \cdots \otimes \mathcal{C}_n$, where each $\mathcal{C}_i$ is a $(2\lambda+1)$-qubit register. For each CNOT in L from qubit i to j, apply $\mathsf{CNOT}^{\otimes 2\lambda+1}$ from register $\mathcal{C}_i$ to $\mathcal{C}_j$.
- $\mathsf{Dec}_{k,L,\theta}(c)$: Parse $c = \{c_i\}_{i\in\Phi}$. Define $\{m_i\}_{i\in\Phi}$ as follows.

$$\forall i \in \Phi_0 : m_i = \begin{cases} 0 \text{ if } c_i \in S + x_{L,i} \\ 1 \text{ if } c_i \in S + \Delta + x_{L,i} \\ \perp \text{ otherwise} \end{cases} \qquad \forall i \in \Phi_1 : m_i = \begin{cases} 0 \text{ if } c_i \in \widehat{S} + z_{L,i} \\ 1 \text{ if } c_i \in \widehat{S} + \widehat{\Delta} + z_{L,i} \\ \perp \text{ otherwise} \end{cases}$$

  If any $m_i = \perp$, then output $\perp$, and otherwise output $m = \{m_i\}_{i\in\Phi}$.
- $\mathsf{Ver}_{k,L,\theta}(c)$:[a] Parse $c = \{c_i\}_{i\in\Phi}$. For each $i \in \Phi_0$, output $\perp$ if $c_i \notin S_\Delta + x_{L,i}$. For each $i \in \Phi_1$, output $\perp$ if $c_i \notin \widehat{S}_{\widehat{\Delta}} + z_{L,i}$. Otherwise, output $\top$.

[a] This procedure is already determined by $\mathsf{Dec}_{k,L,\theta}(c)$, but we write it explicitly for clarity in the proof.

Figure 13: Our construction of a publicly-verifiable linearly-homomorphic QAS with classically-decodable ZX measurements.

Theorem 5.6. *The QAS described in Protocol 13 satisfies correctness (Definition 5.1).*

Proof. First, we show two key claims.

Claim 5.7. *For any S and Δ, it holds that $H^{\otimes 2\lambda+1} E_{\widehat{S},\widehat{\Delta}} = E_{S,\Delta} H$.*

Proof. We show that the maps are equivalent by checking their behavior on the basis $\{|+\rangle, |-\rangle\}$. First,

$$H^{\otimes 2\lambda+1} E_{\widehat{S},\widehat{\Delta}} |+\rangle = H^{\otimes 2\lambda+1} |\widehat{S}_{\widehat{\Delta}}\rangle = |S\rangle = E_{S,\Delta} |0\rangle = E_{S,\Delta} H |+\rangle .$$

Next,

$$H^{\otimes 2\lambda+1} E_{\widehat{S},\widehat{\Delta}} |-\rangle = H^{\otimes 2\lambda+1} \left(|\widehat{S}\rangle - |\widehat{S} + \widehat{\Delta}\rangle \right) = H^{\otimes 2\lambda+1} Z^{\Delta} |\widehat{S}_{\widehat{\Delta}}\rangle = |S + \Delta\rangle = E_{S,\Delta} |1\rangle = E_{S,\Delta} H |-\rangle .$$

□

Claim 5.8. *For any S, Δ, and L, $\mathsf{LinEval}_L E_{S,\Delta}^{\otimes n} = E_{S,\Delta}^{\otimes n} L$*

Proof. We show this for the case where L contains a single CNOT gate, and the full proof follows by applying the argument sequentially. We show that the maps are equivalent by checking their behavior on the basis $\{|b_1, b_2\rangle\}_{b_1,b_2\in\{0,1\}}$.

$$\begin{aligned}
&\mathsf{CNOT}^{\otimes 2\lambda+1} E_{S,\Delta}^{\otimes 2} |b_1, b_2\rangle \\
&= \mathsf{CNOT}^{\otimes 2\lambda+1} |S + b_1 \cdot \Delta\rangle |S + b_2 \cdot \Delta\rangle \\
&= \frac{1}{2^\lambda} \mathsf{CNOT}^{\otimes 2\lambda+1} \sum_{s_1 \in S} |s_1 + b_1 \cdot \Delta\rangle \sum_{s_2 \in S} |s_2 + b_2 \cdot \Delta\rangle \\
&= \frac{1}{2^\lambda} \sum_{s_1 \in S} |s_1 + b_1 \cdot \Delta\rangle \sum_{s_2 \in S} |(s_1 + s_2) + (b_1 + b_2) \cdot \Delta\rangle \\
&= |S + b_1 \cdot \Delta\rangle |S + (b_1 + b_2) \cdot \Delta\rangle \\
&= E_{S,\Delta}^{\otimes 2} \mathsf{CNOT} |b_1, b_2\rangle .
\end{aligned}$$

□

Now, define measurements $M'_{\theta,f}, \widetilde{M}'_{\theta,f,k,L}$ so that

$$M_{\theta,f} = H^{\Phi_1} M'_{\theta,f} H^{\Phi_1}, \quad \text{and} \quad \widetilde{M}_{\theta,f,k,L} = H^{\tilde{\Phi}_1} X^{x_{L,\Phi_0}, z_{L,\Phi_1}} \widetilde{M}'_{\theta,f,k,L} X^{x_{L,\Phi_0}, z_{L,\Phi_1}} H^{\tilde{\Phi}_1}.$$

To be concrete,

$$M'_{\theta,f} := \left\{ \sum_{m : f(m_\Phi) = y} |m\rangle\langle m| \right\}_y ,$$

and

$$\widetilde{M}'_{\theta,f,k,L} \coloneqq \left\{ \sum_{m:f(m_\Phi)=y} \left(\sum_{\substack{c:\{c_i \in S+m_i\cdot\Delta\}_{i\in\Phi_0},\\ \{c_i\in\widehat{S}+m_i\cdot\widehat{\Delta}\}_{i\in\Phi_1}}} |c\rangle\langle c| \right) \right\}_y \cup \left\{ \sum_{\substack{c:\exists i\in\Phi_0 \text{ s.t. } c_i\notin S_\Delta\\ \vee \exists i\in\Phi_1 \text{ s.t. } c_i\notin \widehat{S}_{\widehat{\Delta}}}} |c\rangle\langle c| \right\}.$$

Observe that

$$\widetilde{M}'_{\theta,f,k,L}\left(E_{S,\Delta}^{\otimes|\Phi_\perp\cup\Phi_0|}\otimes E_{\widehat{S},\widehat{\Delta}}^{\otimes|\Phi_1|}\right) = \left(E_{S,\Delta}^{\otimes|\Phi_\perp\cup\Phi_0|}\otimes E_{\widehat{S},\widehat{\Delta}}^{\otimes|\Phi_1|}\right) M'_{\theta,f}.$$

Then,

$$\begin{aligned}
&\mathsf{LinEval}_L^\dagger \widetilde{M}_{\theta,f,k,L}\mathsf{LinEval}_L\mathsf{Enc}_k \\
&= \mathsf{LinEval}_L^\dagger \widetilde{M}_{\theta,f,k,L}\mathsf{LinEval}_L X^x Z^z E_{S,\Delta}^{\otimes n} \\
&= \mathsf{LinEval}_L^\dagger \widetilde{M}_{\theta,f,k,L} X^{x_L} Z^{z_L}\mathsf{LinEval}_L E_{S,\Delta}^{\otimes n} \\
&= \mathsf{LinEval}_L^\dagger \widetilde{M}_{\theta,f,k,L} X^{x_L} Z^{z_L} E_{S,\Delta}^{\otimes n} L && \text{(Claim 5.8)} \\
&= \mathsf{LinEval}_L^\dagger H^{\widetilde{\Phi}_1} X^{x_{L,\Phi_0} z_{L,\Phi_1}} \widetilde{M}'_{\theta,f,k,L} X^{x_{L,\Phi_0},z_{L,\Phi_1}} H^{\widetilde{\Phi}_1} X^{x_L} Z^{z_L} E_{S,\Delta}^{\otimes n} L \\
&= \mathsf{LinEval}_L^\dagger H^{\widetilde{\Phi}_1} X^{x_{L,\Phi_0} z_{L,\Phi_1}} \widetilde{M}'_{\theta,f,k,L} X^{x_{L,\Phi_\perp}} Z^{z_{L,\Phi_\perp},z_{L,\Phi_0},x_{L,\Phi_1}} H^{\widetilde{\Phi}_1} E_{S,\Delta}^{\otimes n} L \\
&= \mathsf{LinEval}_L^\dagger H^{\widetilde{\Phi}_1} X^{x_{L,\Phi_\perp},x_{L,\Phi_0},z_{L,\Phi_1}} Z^{z_{L,\Phi_\perp},z_{L,\Phi_0},x_{L,\Phi_1}} \widetilde{M}'_{\theta,f,k,L} H^{\widetilde{\Phi}_1} E_{S,\Delta}^{\otimes n} L \\
&= \mathsf{LinEval}_L^\dagger X^{x_L} Z^{z_L} H^{\widetilde{\Phi}_1} \widetilde{M}'_{\theta,f,k,L} H^{\widetilde{\Phi}_1} E_{S,\Delta}^{\otimes n} L \\
&= \mathsf{LinEval}_L^\dagger X^{x_L} Z^{z_L} H^{\widetilde{\Phi}_1} \widetilde{M}'_{\theta,f,k,L}\left(E_{S,\Delta}^{\otimes|\Phi_\perp\cup\Phi_0|}\otimes E_{\widehat{S},\widehat{\Delta}}^{\otimes|\Phi_1|}\right) H^{\Phi_1} L && \text{(Claim 5.7)} \\
&= \mathsf{LinEval}_L^\dagger X^{x_L} Z^{z_L} H^{\widetilde{\Phi}_1}\left(E_{S,\Delta}^{\otimes|\Phi_\perp\cup\Phi_0|}\otimes E_{\widehat{S},\widehat{\Delta}}^{\otimes|\Phi_1|}\right) M'_{\theta,f} H^{\Phi_1} L \\
&= \mathsf{LinEval}_L^\dagger X^{x_L} Z^{z_L} E_{S,\Delta}^{\otimes n} H^{\Phi_1} M'_{\theta,f} H^{\Phi_1} L \\
&= X^x Z^z \mathsf{LinEval}_L^\dagger E_{S,\Delta}^{\otimes n} M_{\theta,f} L \\
&= X^x Z^z E_{S,\Delta}^{\otimes n} L^\dagger M_{\theta,f} L \\
&= \mathsf{Enc}_k L^\dagger M_{\theta,f} L.
\end{aligned}$$

□

5.2.3 Proof of security

Theorem 5.9. *The QAS described in Protocol 13 satisfies security (Definition 5.2).*

Proof. We begin by modifying the Gen procedure and $\mathsf{Ver}_{k,\cdot,\cdot}(\cdot)$ oracle, and arguing that the output of the experiment remains (almost) unaffected. In particular, we will "expand" the verification oracle with random superspaces $R \supset S_\Delta$ and $\widehat{R} \supset \widehat{S}_{\widehat{\Delta}}$. Consider the following procedures.

- $\mathsf{Gen}'(1^\lambda, n)$: Sample a uniformly random λ-dimensional subspace $S \subset \mathbb{F}_2^{2\lambda+1}$, vector $\Delta \leftarrow \mathbb{F}_2^{2\lambda+1} \setminus S$, and $x_i, z_i \leftarrow \mathbb{F}_2^{2\lambda+1}$ for each $i \in [n]$. Sample uniformly random $(3\lambda/2+1)$-dimensional subspaces $R, \widehat{R} \subset \mathbb{F}_2^{2\lambda+1}$ conditioned on $S_\Delta \subset R$ and $\widehat{S}_{\widehat{\Delta}} \subset \widehat{R}$. Output $k := (S, \Delta, x, z)$ along with $(R, \widehat{R})$.
- $\mathsf{Ver}'_{k,R,\widehat{R},L,\theta}(c)$: Parse $c = \{c_i\}_{i\in\Phi}$. For each $i \in \Phi_0$, output $\perp$ if $c_i \notin R + x_{L,i}$. For each $i \in \Phi_1$, output $\perp$ if $c_i \notin \widehat{R} + z_{L,i}$. Otherwise, output $\top$.

Claim 5.10. *For any L, θ, f, A, and $|\psi\rangle$, it holds that*

$$\left\{ y : \begin{array}{r} k \leftarrow \mathsf{Gen}(1^\lambda, n) \\ y \leftarrow \widetilde{M}_{\theta,f,k,L} \circ A^{\mathsf{Ver}_{k,\cdot,\cdot}(\cdot)} \circ \mathsf{Enc}_k(|\psi\rangle) \end{array} \right\} \approx_{2^{-\Omega(\lambda)}} \left\{ y : \begin{array}{r} (k, R, \widehat{R}) \leftarrow \mathsf{Gen}'(1^\lambda, n) \\ y \leftarrow \widetilde{M}_{\theta,f,k,L} \circ A^{\mathsf{Ver}'_{k,R,\widehat{R},\cdot,\cdot}(\cdot)} \circ \mathsf{Enc}_k(|\psi\rangle) \end{array} \right\}.$$

Proof. Note that these distributions can be sampled by a reduction given oracle access to either $(O[S_\Delta], O[\widehat{S}_{\widehat{\Delta}}])$ or $(O[R], O[\widehat{R}])$. Now, for any fixed $(\lambda+1)$-dimensional subspaces $S_\Delta, \widehat{S}_\Delta$ and any vector v,

$$\Pr_{R,\widehat{R}}[v \in R \setminus S_\Delta \cup \widehat{R} \setminus \widehat{S}_{\widehat{\Delta}}] \le \frac{|R \setminus S_\Delta|}{|\mathbb{F}_2^{2\lambda+1} \setminus S_\Delta|} + \frac{|\widehat{R} \setminus \widehat{S}_{\widehat{\Delta}}|}{|\mathbb{F}_2^{2\lambda+1} \setminus \widehat{S}_{\widehat{\Delta}}|} = 2 \cdot \frac{2^{3\lambda/2+1} - 2^{\lambda+1}}{2^{2\lambda+1} - 2^{\lambda+1}} = 2^{-\Omega(\lambda)},$$

where the probability is over sampling random $(3\lambda/2+1)$-dimensional subspaces $R, \widehat{R}$ conditioned on $S_\Delta \subset R$ and $\widehat{S}_{\widehat{\Delta}} \subset \widehat{R}$. Then the claim follows by noting that $(O[S_\Delta], O[\widehat{S}_{\widehat{\Delta}}])$ and $(O[R], O[\widehat{R}])$ are identical outside of $R \setminus S_\Delta$ and $\widehat{R} \setminus \widehat{S}_\Delta$, and applying Lemma 2.8 (a standard oracle hybrid argument). □

Now, fix any $(3\lambda/2+1)$-dimensional subspaces $R, \widehat{R}$ such that $\widehat{R}^\perp \subset R$, and consider the following procedure.

- $\mathsf{Gen}'_{R,\widehat{R}}(1^\lambda, n)$: Sample a uniformly random subspace $S \subset \mathbb{F}_2^{2\lambda+1}$ conditioned on $\widehat{R}^\perp \subset S \subset R$, sample a uniformly random vector $\Delta \leftarrow R \setminus S$, and sample uniformly random $x_i^R, z_i^{\widehat{R}} \leftarrow (R, \widehat{R})$ for each $i \in [n]$. Set $x^R = (x_1^R, \ldots, x_n^R), z^{\widehat{R}} = (z_1^{\widehat{R}}, \ldots, z_n^{\widehat{R}})$ and output $(S, \Delta, x^R, z^{\widehat{R}})$.

Next, let $\mathsf{co}(R)$ be an arbitrary set of coset representatives of R, let $\mathsf{co}(\widehat{R})$ be an arbitrary set of coset representatives of $\widehat{R}$, and fix any

$$x^{\mathsf{co}(R)} = \left(x_1^{\mathsf{co}(R)}, \ldots, x_n^{\mathsf{co}(R)}\right), \quad z^{\mathsf{co}(\widehat{R})} = \left(z_1^{\mathsf{co}(\widehat{R})}, \ldots, z_n^{\mathsf{co}(\widehat{R})}\right),$$

where each $x_i^{\mathsf{co}(R)} \in \mathsf{co}(R)$ and $z_i^{\mathsf{co}(\widehat{R})} \in \mathsf{co}(\widehat{R})$. Then the proof of the theorem follows by combining Claim 5.10 with the following claim. Notice that the adversary A in the following claim no longer requires access to the "expanded" oracle $\mathsf{Ver}'_{k,R,\widehat{R},\cdot,\cdot}(\cdot)$, since A is allowed to depend on $\left(R, \widehat{R}, x^{\mathsf{co}(R)}, z^{\mathsf{co}(\widehat{R})}\right)$, which suffice to implement $\mathsf{Ver}'_{k,R,\widehat{R},\cdot,\cdot}(\cdot)$.

Claim 5.11. *Fix any $R, \widehat{R}, x^{\mathsf{co}(R)}, z^{\mathsf{co}(\widehat{R})}$. Then for any L, θ, f, and unitary A,[39] there exists an $\epsilon = \epsilon(\lambda) \in [0,1]$ such that for any $|\psi\rangle$,*

$$\left\{ y : \begin{array}{r} \left(S, \Delta, x^R, z^{\widehat{R}}\right) \leftarrow \mathsf{Gen}'_{R,\widehat{R}}(1^\lambda, n) \\ x := x^R + x^{\mathsf{co}(R)}, z := z^{\widehat{R}} + z^{\mathsf{co}(\widehat{R})} \\ k := (S, \Delta, x, z) \\ y \leftarrow \widetilde{M}_{\theta,f,k,L} \circ A \circ \mathsf{Enc}_k(|\psi\rangle) \end{array} \right\} \approx_{2^{-\Omega(\lambda)}} (1-\epsilon)\{y : y \leftarrow M_{\theta,f} \circ L(|\psi\rangle)\} + \epsilon\{\bot\}.$$

Proof. Let $\mathcal{D}$ be the distribution described by the LHS of the statement in the claim. Next, define $x_L^{\mathsf{co}(R)}, z_L^{\mathsf{co}(\widehat{R})} := L(x^{\mathsf{co}(R)}, z^{\mathsf{co}(\widehat{R})})$, and define the distribution $\mathcal{K}_L$ as follows.

$$\mathcal{K}_L := \left\{ (S, \Delta, x_L, z_L) : \begin{array}{r} \left(S, \Delta, x^R, z^{\widehat{R}}\right) \leftarrow \mathsf{Gen}'_{R,\widehat{R}}(1^\lambda, n) \\ x_L^R, z_L^{\widehat{R}} := L(x^R, z^{\widehat{R}}) \\ x_L := x_L^R + x_L^{\mathsf{co}(R)}, z_L := z_L^{\widehat{R}} + z_L^{\mathsf{co}(\widehat{R})} \end{array} \right\}.$$

Observe that the distribution over $k = (S, \Delta, x, z)$ as sampled by $\mathcal{D}$ is equivalent to the distribution that results from sampling $(S, \Delta, x_L, z_L) \leftarrow \mathcal{K}_L$ and setting $(x, z) = L^{-1}(x_L, z_L)$. Thus, we can write $\mathcal{D}$ equivalently as

$$\mathcal{D} = \left\{ y : \begin{array}{r} (S, \Delta, x_L, z_L) \leftarrow \mathcal{K}_L \\ (x, z) := L^{-1}(x_L, z_L) \\ k := (S, \Delta, x, z) \\ y \leftarrow \widetilde{M}_{\theta,f,k,L} \circ A \circ \mathsf{Enc}_k(|\psi\rangle) \end{array} \right\}.$$

Moreover, the vectors (x_L, z_L) obtained by sampling $(S, \Delta, x_L, z_L) \leftarrow \mathcal{K}_L$ are such that x_L and z_L are uniformly random over an affine subspaces, namely,

$$x_L \leftarrow R^{\oplus n} + x_L^{\mathsf{co}(R)}, \quad \text{and} \quad z_L \leftarrow \widehat{R}^{\oplus n} + z_L^{\mathsf{co}(\widehat{R})}.$$

This follows because L is full rank, and the vectors $x^R, z^{\widehat{R}} = (x_1^R, \ldots, x_n^R), (z_1^{\widehat{R}}, \ldots, z_n^{\widehat{R}})$ obtained by sampling

$$\left(S, \Delta, x^R, z^{\widehat{R}}\right) \leftarrow \mathsf{Gen}'_{R,\widehat{R}}(1^\lambda, n)$$

are such that each x_i^R is uniformly random over R and each $z_i^{\widehat{R}}$ is uniformly random over $\widehat{R}$. The fact that x_L and z_L are uniform over affine subspaces will be used later in the proof when we apply the Pauli twirl over affine subspaces (Lemma 2.5).

Next, we introduce some more notation.

[39] As noted in Remark 5.3, by introducing a sufficiently large workspace register $\mathcal{A}$ initialized to $|0\rangle$, we can assume without loss of generality that the adversary A is unitary. This additional workspace register $\mathcal{A}$ is left implicit in the description of the claim and proof.

- For each $y \in \mathsf{range}(f)$, define

$$V_y \coloneqq \bigcup_{m:f(m_\Phi)=y} \left(\bigotimes_{i\in\Phi_0} (S + m_i \cdot \Delta) \bigotimes_{i\in\Phi_1} \left(\widehat{S} + m_i \cdot \widehat{\Delta}\right) \right),$$

  and define

$$V_\perp \coloneqq \{0,1\}^{(2\lambda+1)n} \setminus \bigcup_{y\in\mathsf{range}(f)} V_y.$$

  For $y \in \mathsf{range}(f) \cup \{\perp\}$, define $|V_y\rangle \coloneqq \sum_{v\in V_y} |v\rangle$.

- Define the unitary $B \coloneqq A \circ \mathsf{LinEval}_L^\dagger$. Note that the "honest" A operation just applies $\mathsf{LinEval}_L$, so in this case B is the identity.

- For any pure state $|\phi\rangle$, define $\mathsf{Mx}[|\phi\rangle] \coloneqq |\phi\rangle\langle\phi|$.

Now, given any $(S, \Delta, x_L, z_L) \in \mathcal{K}_L$, which defines $(x, z) = L^{-1}(x_L, z_L)$, and any $y \in \mathsf{range}(f) \cup \{\perp\}$, we can write the probability that $\mathcal{D}$ outputs y as

$$\begin{aligned}
&\left\| \Pi[V_y] X^{x_{L,\Phi_0}, z_{L,\Phi_1}} H^{\widetilde{\Phi}_1} A\mathsf{Enc}_{(S,\Delta,x,z)} |\psi\rangle \right\|^2 \\
&= \left\| \Pi[V_y] X^{x_{L,\Phi_0}, z_{L,\Phi_1}} H^{\widetilde{\Phi}_1} B\mathsf{LinEval}_L X^x Z^z E_{S,\Delta}^{\otimes n} |\psi\rangle \right\|^2 \\
&= \left\| \Pi[V_y] H^{\widetilde{\Phi}_1} X^{x_{L,\Phi_0}} Z^{z_{L,\Phi_1}} B X^{x_L} Z^{z_L} E_{S,\Delta}^{\otimes n} L |\psi\rangle \right\|^2 \\
&= \left\| \Pi[V_y] H^{\widetilde{\Phi}_1} X^{x_L} Z^{z_L} B X^{x_L} Z^{z_L} E_{S,\Delta}^{\otimes n} L |\psi\rangle \right\|^2,
\end{aligned}$$

where in the last line, we have inserted Pauli X operations on registers that are either measured in the Hadamard basis or not measured at all and Pauli Z operations on registers that are either measured in the standard basis or not measured at all. Doing this has no effect on the outcome. Thus, we can write the distribution $\mathcal{D}$ concisely as

$$\mathcal{D} = \sum_{y\in\mathsf{range}(f)\cup\{\perp\}} |y\rangle\langle y| \frac{1}{|\mathcal{K}_L|} \sum_{(S,\Delta,x_L,z_L)\in\mathcal{K}_L} \langle V_y| \,\mathsf{Mx}\left[H^{\widetilde{\Phi}_1} Z^{z_L} X^{x_L} B X^{x_L} Z^{z_L} E_{S,\Delta}^{\otimes n} L |\psi\rangle\right] |V_y\rangle .$$

To complete the proof, we will decompose B as a sum of Paulis, and factor out terms that will cause $\mathcal{D}$ to output $\perp$ (with high probability). Eventually, we'll be left with terms that do not affect the outcome of directly measuring $L|\psi\rangle$. To begin with, let

$$\mathcal{P} \coloneqq \left\{ X^x Z^z : x = (x_1, \ldots, x_n), z = (z_1, \ldots, z_n) \in \{0,1\}^{(2\lambda+1)n} \right\},$$

and define the subsets

$$\mathcal{P}_\perp \coloneqq \left\{ X^x Z^z : \exists i \in \Phi_0 \text{ s.t. } x_i \notin R \text{ or } \exists i \in \Phi_1 \text{ s.t. } z_i \notin \widehat{R} \right\}, \qquad \mathcal{P}_\top = \mathcal{P} \setminus \mathcal{P}_\perp.$$

Then we can write B as

$$B = \sum_{P \in \mathcal{P}_\top} \alpha_P P + \sum_{P \in \mathcal{P}_\perp} \alpha_P P \coloneqq B_\top + B_\perp,$$

for some coefficients α_P.

Note that for any $y \in \mathsf{range}(f)$, $(S, \Delta, x_L, z_L) \in \mathcal{K}_L$, and $P \in P_\perp$,

$$\langle V_y | H^{\widetilde{\Phi}_1} Z^{z_L} X^{x_L} P X^{x_L} Z^{z_L} E_{S,\Delta}^{\otimes n} L |\psi\rangle = 0,$$

which follows by definition of V_y, since $S_\Delta \subset R$ and $\widehat{S}_{\widehat{\Delta}} \subset \widehat{R}$. Thus there exists an $\epsilon_\perp$ such that

$$\mathcal{D} = \sum_{y \in \mathsf{range}(f) \cup \{\perp\}} |y\rangle\langle y| \frac{1}{|\mathcal{K}_L|} \sum_{(S,\Delta,x_L,z_L)} \langle V_y | \mathsf{Mx}\left[H^{\widetilde{\Phi}_1} Z^{z_L} X^{x_L} B_\top X^{x_L} Z^{z_L} E_{S,\Delta}^{\otimes n} L |\psi\rangle \right] |V_y\rangle + \epsilon_\perp |\perp\rangle\langle\perp| .$$

Next, we define the following.

- Let $C \simeq R/\widehat{R}^\perp$ be a subspace of coset representatives of $\widehat{R}^\perp$ in R.
- Let $\widehat{C} \simeq \widehat{R}/R^\perp$ be a subspace of coset representatives of $R^\perp$ in $\widehat{R}$.
- Define the set of Paulis

$$\mathcal{P}_{C,\widehat{C}} \coloneqq \left\{ X^x Z^z : \begin{array}{c} \left\{x_i \in C, z_i = 0^{2\lambda+1}\right\}_{i\in\Phi_0}, \left\{x_i = 0^{2\lambda+1}, z_i \in \widehat{C}\right\}_{i\in\Phi_1}, \\ \left\{x_i = 0^{2\lambda+1}, z_i = 0^{2\lambda+1}\right\}_{i\in\Phi_\perp} \end{array} \right\}.$$

Now, for any $(x, z) = (x_1, \ldots, x_n, z_1, \ldots, z_n)$ such that $P = X^x Z^z \in \mathcal{P}_\top$, define $(x', z') = (x'_1, \ldots, x'_n, z'_1, \ldots, z'_n)$ such that $X^{x'} Z^{z'} \in \mathcal{P}_{C,\widehat{C}}$ as follows.

- For $i \in \Phi_0$, let $x'_i \in C$ be the representative of x_i's coset (recall that $x_i \in R$ by definition of $\mathcal{P}_\top$), and let $z'_i = 0^{2\lambda+1}$.
- For $i \in \Phi_1$, let $z'_i \in \widehat{C}$ be the representative of z_i's coset (recall that $z_i \in \widehat{R}$ by definition of $\mathcal{P}_\top$), and let $x'_i = 0^{2\lambda+1}$.
- For $i \in \Phi_\perp$, let $x'_i = 0^{2\lambda+1}$, $z'_i = 0^{2\lambda+1}$.

Then note that for all $y \in \mathsf{range}(f) \cup \{\bot\}$ and $(S, \Delta, x_L, z_L) \in \mathcal{K}_L$, it holds that

$$\langle V_y | H^{\widetilde{\Phi}_1} Z^{z_L} X^{x_L} X^x Z^z X^{x_L} Z^{z_L} E_{S,\Delta}^{\otimes n} L | \psi \rangle = \langle V_y | H^{\widetilde{\Phi}_1} Z^{z_L} X^{x_L} X^{x'} Z^{z'} X^{x_L} Z^{z_L} E_{S,\Delta}^{\otimes n} L | \psi \rangle ,$$

which follows by definition of V_y and the fact that $S, \widehat{S}$ are always sampled so that $\widehat{R}^\perp \subset S$ and $R^\perp \subset \widehat{S}$.

That is, we have identified for any $P \in \mathcal{P}_\top$ a canonical $P' \in \mathcal{P}_{C,\widehat{C}}$ for which P' will have the same behavior as P over all $(S, \Delta, x_L, z_L) \in \mathcal{K}_L$. Thus, we can replace $B_\top$ in the expression for $\mathcal{D}$ with

$$\sum_{P \in \mathcal{P}_{C,\widehat{C}}} \alpha_P P$$

for some coefficients α_P, and write $\mathcal{D}$ as

$$\begin{aligned}
&\sum_{y \in \mathsf{range}(f) \cup \{\bot\}} |y\rangle\langle y| \frac{1}{|\mathcal{K}_L|} \sum_{(S,\Delta,x_L,z_L)} \langle V_y | \mathsf{Mx} \left[H^{\widetilde{\Phi}_1} Z^{z_L} X^{x_L} \left(\sum_{P \in \mathcal{P}_{C,\widehat{C}}} \alpha_P P \right) X^{x_L} Z^{z_L} E_{S,\Delta}^{\otimes n} L | \psi \rangle \right] | V_y \rangle \\
&\quad + \epsilon_\bot |\bot\rangle\langle\bot| \\
= &\sum_{y \in \mathsf{range}(f) \cup \{\bot\}} |y\rangle\langle y| \sum_{P_0, P_1 \in \mathcal{P}_{C,\widehat{C}}} \alpha_{P_0} \alpha_{P_1}^* \frac{1}{|\mathcal{K}_L|} \\
&\qquad \left(\sum_{(S,\Delta,x_L,z_L)} \langle V_y | H^{\widetilde{\Phi}_1} Z^{z_L} X^{x_L} P_0 X^{x_L} Z^{z_L} \mathsf{Mx} \left[E_{S,\Delta}^{\otimes n} L | \psi \rangle \right] Z^{z_L} X^{x_L} P_1^\dagger X^{x_L} Z^{z_L} H^{\widetilde{\Phi}_1} | V_y \rangle \right) \\
&\quad + \epsilon_\bot |\bot\rangle\langle\bot|
\end{aligned}$$

Now, we are finally ready to apply the the Pauli twirl over affine subspaces (Lemma 2.5). To do so, we make the following observations.

- As noted above, $x_L \leftarrow R^{\oplus n} + x_L^{\mathsf{co}(R)}$, and $z_L \leftarrow \widehat{R}^{\oplus n} + z_L^{\mathsf{co}(\widehat{R})}$ are uniformly random over affine subspaces of $R^{\oplus n}$ and $\widehat{R}^{\oplus n}$ respectively.
- Consider any $X^{x_0} Z^{z_0} \neq X^{x_1} Z^{z_1} \in \mathcal{P}_{C,\widehat{C}}$. If $x_0 \neq x_1$, then there exists some index $i \in [n]$ such that $x_{0,i} \oplus x_{1,i} \notin \widehat{R}^\perp$ and thus, $x_0 \oplus x_1 \notin (\widehat{R}^{\oplus n})^\perp$. Otherwise, $z_0 \neq z_1$, and there exists some index $i \in [n]$ such that $z_{0,i} \oplus z_{1,i} \notin R^\perp$ and thus, $z_0 \oplus z_1 \notin (R^{\oplus n})^\perp$.

Then by Lemma 2.5, all the cross-terms $P_0 \neq P_1$ are killed in the above expression for $\mathcal{D}$, which we can now write as

$$\begin{aligned}
&\sum_{y \in \mathsf{range}(f) \cup \{\bot\}} |y\rangle\langle y| \sum_{P \in \mathcal{P}_{C,\widehat{C}}} \alpha_P \alpha_P^* \frac{1}{|\mathcal{K}_L|} \left(\sum_{(S,\Delta,x_L,z_L)} \langle V_y | \mathsf{Mx} \left[H^{\widetilde{\Phi}_1} Z^{z_L} X^{x_L} P X^{x_L} Z^{z_L} E_{S,\Delta}^{\otimes n} L | \psi \rangle \right] | V_y \rangle \right) \\
&\quad + \epsilon_\bot |\bot\rangle\langle\bot| ,
\end{aligned}$$

Finally, since S, Δ are chosen uniformly at random conditioned on $\widehat{R}^{\perp} \subset S_{\Delta} \subset R$, we have that for any fixed $P \in \mathcal{P}_{C,\widehat{C}} \setminus \mathcal{I}$,

$$\sum_{y \in \mathsf{range}(f) \cup \{\perp\}} \frac{1}{|\mathcal{K}_L|} \sum_{(S,\Delta,x_L,z_L)} \langle V_y | \, \mathsf{Mx}\left[H^{\widetilde{\Phi}_1} Z^{z_L} X^{x_L} P X^{x_L} Z^{z_L} E_{S,\Delta}^{\otimes n} L \, |\psi\rangle\right] |V_y\rangle$$
$$\leq \frac{|S_{\Delta} \setminus \widehat{R}_{\perp}|}{|R \setminus \widehat{R}_{\perp}|} + \frac{|\widehat{S}_{\widehat{\Delta}} \setminus R_{\perp}|}{|\widehat{R} \setminus R_{\perp}|} = 2 \cdot \frac{2^{\lambda+1} - 2^{\lambda}}{2^{3\lambda/2+1} - 2^{\lambda}} = 2^{-\Omega(\lambda)}.$$

Thus, $\mathcal{D}$ is within $2^{-\Omega(\lambda)}$ total variation distance of

$$(1-\epsilon_{\perp}) \sum_y |y\rangle\langle y| \frac{1}{|\mathcal{K}_L|} \sum_{(S,\Delta,x_L,z_L)} \langle V_y | \, \mathsf{Mx}\left[H^{\widetilde{\Phi}_1} Z^{z_L} X^{x_L} \mathcal{I} X^{x_L} Z^{z_L} E_{S,\Delta}^{\otimes n} L \, |\psi\rangle\right] |V_y\rangle + \epsilon_{\perp} |\perp\rangle\langle\perp|$$
$$= (1-\epsilon_{\perp}) \sum_y |y\rangle\langle y| \frac{1}{|\mathcal{K}_L|} \sum_{(S,\Delta,x_L,z_L)} \langle V_y | \, \mathsf{Mx}\left[H^{\widetilde{\Phi}_1} E_{S,\Delta}^{\otimes n} L \, |\psi\rangle\right] |V_y\rangle + \epsilon_{\perp} |\perp\rangle\langle\perp|$$
$$= (1-\epsilon_{\perp}) \sum_y |y\rangle\langle y| \left(\sum_{m:f(m)=y} \langle m|\right) \mathsf{Mx}\left[H^{\Phi_1} L \, |\psi\rangle\right] \left(\sum_{m:f(m)=y} |m\rangle\right) + \epsilon_{\perp} |\perp\rangle\langle\perp|$$
$$= (1-\epsilon_{\perp})\{y : y \leftarrow M_{\theta,f} \circ L(|\psi\rangle)\} + \epsilon_{\perp}\{\perp\},$$

which completes the proof.

□

□

Theorem 5.12. *The QAS described in Protocol 13 satisfies privacy (Definition 5.5).*

Proof. First, recalling the definitions of $\mathsf{Gen}', \mathsf{Ver}'$ in the proof of Theorem 5.9, and applying the oracle indistinguishability argued during the proof of Claim 5.10, it suffices to show that

$$\left| \Pr_{k,R,\widehat{R} \leftarrow \mathsf{Gen}'(1^{\lambda},n)} \left[1 \leftarrow A^{\mathsf{Ver}'_{k,R,\widehat{R},\cdot,\cdot}(\cdot)} \circ \mathsf{Enc}_k(|\psi_0\rangle)\right] - \Pr_{k,R,\widehat{R} \leftarrow \mathsf{Gen}'(1^{\lambda},n)} \left[1 \leftarrow A^{\mathsf{Ver}'_{k,R,\widehat{R},\cdot,\cdot}(\cdot)} \circ \mathsf{Enc}_k(|\psi_1\rangle)\right] \right| = 0.$$

To see this, we'll show that we can give enough information to A for it to implement $\mathsf{Ver}'_{k,R,\widehat{R},\cdot,\cdot}(\cdot)$ while preserving sufficient randomness to one-time pad the input state.

Consider the following equivalent description of $\mathsf{Gen}'(1^{\lambda}, n)$.

$\mathsf{Gen}'(1^{\lambda}, n)$:

- Sample a uniformly random λ-dimensional subspace $S \subset \mathbb{F}_2^{2\lambda+1}$, vector $\Delta \leftarrow \mathbb{F}_2^{2\lambda+1} \setminus S$, and uniformly random $(3\lambda/2+1)$-dimensional subspaces $R, \widehat{R} \subset \mathbb{F}_2^{2\lambda+1}$ conditioned on $S_{\Delta} \subset R$ and $\widehat{S}_{\widehat{\Delta}} \subset \widehat{R}$.

- Let H_Δ be the 2λ-dimensional subspace perpendicular to Δ and $H_{\widehat{\Delta}}$ be the 2λ-dimensional subspace perpendicular to $\widehat{\Delta}$. For each $i \in [n]$, sample $x_{i,\Delta} \leftarrow H_\Delta, b_i \leftarrow \{0,1\}, z_{i,\widehat{\Delta}} \leftarrow H_{\widehat{\Delta}}, c_i \leftarrow \{0,1\}$, and define $x_i = x_{i,\Delta} + b_i \cdot \Delta$ and $z_i = z_{i,\widehat{\Delta}} + c_i \cdot \widehat{\Delta}$.
- Output $(S, \Delta, x, z), R, \widehat{R}$.

Now, fix any choice of $S, \Delta, R, \widehat{R}, x_\Delta, z_{\widehat{\Delta}}$ sampled during the procedure $\mathsf{Gen}'(1^\lambda, n)$, where $x_\Delta := (x_{1,\Delta}, \cdots, x_{n,\Delta})$ and $z_{\widehat{\Delta}} := \left(z_{1,\widehat{\Delta}}, \cdots, z_{n,\widehat{\Delta}}\right)$, and consider the following procedure that completes the sampling of the key.

$\mathsf{Gen}'_{S,\Delta,R,\widehat{R},x_\Delta,z_{\widehat{\Delta}}}(1^\lambda, n)$:

- For each $i \in [n]$, sample $b_i, c_i \leftarrow \{0,1\}$, and define $x_i = x_{i,\Delta} + b_i \cdot \Delta$ and $z_i = z_{i,\widehat{\Delta}} + c_i \cdot \widehat{\Delta}$.
- Output (S, Δ, x, z).

Since the oracle $\mathsf{Ver}'_{k,R,\widehat{R},\cdot,\cdot}(\cdot)$ can be implemented given just the fixed information $S, \Delta, R, \widehat{R}, x_\Delta, z_{\widehat{\Delta}}$, it suffices to show that for any $S, \Delta, R, \widehat{R}, x_\Delta, z_{\widehat{\Delta}}$ and any adversary A (whose description may depend on this information), it holds that

$$\left| \Pr_k \left[1 \leftarrow A(\mathsf{Enc}_k(|\psi_0\rangle))\right] - \Pr_k \left[1 \leftarrow A(\mathsf{Enc}_k(|\psi_1\rangle))\right] \right| = 0,$$

where the probability is over $k \leftarrow \mathsf{Gen}'_{S,\Delta,R,\widehat{R},x_\Delta,z_{\widehat{\Delta}}}(1^\lambda, n)$. Since

$$\mathsf{Enc}_k = X^x Z^z E_{S,\Delta}^{\otimes n} = X^{x_\Delta} Z^{z_{\widehat{\Delta}}} X^{b_1 \cdot \Delta \ldots b_n \cdot \Delta} Z^{c_1 \cdots \widehat{\Delta} \ldots c_n \cdot \widehat{\Delta}} E_{S,\Delta}^{\otimes n} = X^{x_\Delta} Z^{z_{\widehat{\Delta}}} E_{S,\Delta}^{\otimes n} X^{b_1 \ldots b_n} Z^{c_1 \ldots c_n},$$

this follows from the quantum one-time pad [AMTDW00]. That is, we use the fact that

$$\sum_{b_1,\ldots,b_n,c_1,\ldots,c_n} \mathsf{Mx}\left[X^{b_1,\ldots,b_n} Z^{c_1,\ldots,c_n} |\psi_0\rangle\right] = \sum_{b_1,\ldots,b_n,c_1,\ldots,c_n} \mathsf{Mx}\left[X^{b_1,\ldots,b_n} Z^{c_1,\ldots,c_n} |\psi_1\rangle\right].$$

□

5.3 Linear + measurement quantum programs

In this section, we show that any quantum program with classical input and output (Definition 2.2) can be implemented using a "linear + measurement" (LM) quantum program.

In slightly more detail, we make use of magic states in order to write any quantum circuit as an alternating sequence of linear operations L_i (by which we mean a sequence of CNOT gates) and partial ZX measurements M_{θ_i,f_i}, where the description of each f_i may depend on the classical input x as well as previous measurement results. We encourage the reader to review our notation for partial ZX measurements M_{θ_i,f_i} described at the

beginning of Section 5.2.1. We remark here that these measurements are "partial" in two aspects: (i) they may only operate on a subset of the qubits, and (ii) each measurement outcome may be associated with multiple basis vectors, meaning that the input qubits are not necessarily fully collapsed. We also remark that for the purpose of this paper, we restrict attention to circuits with classical inputs and outputs, but note that one could consider circuits with quantum inputs and outputs as well.

5.3.1 Definition

We first formally define LM quantum programs, and accompany this with a diagram in Fig. 14.

Definition 5.13 (LM quantum program)**.** *An* LM *quantum program with classical input and output is described by:*

- *A quantum state $|\psi\rangle$ on n qubits.*
- *Linear operations $L_1, \ldots, L_{t+1}$, where each L_i is a sequence of CNOT gates.*
- *Partial ZX measurements $M_{\theta_1, f_1^{(\cdot)}}, M_{\theta_2, f_2^{(\cdot)}}, \ldots, M_{\theta_t, f_t^{(\cdot)}}, M_{\theta_{t+1}, g^{(\cdot)}}$ defined by sets of bases $\{\theta_i\}_{i\in[t+1]}$ and classical functions $\{f_i^{(\cdot)}\}_{i\in[t]}, g^{(\cdot)}$, which will be parameterized by the input x as well as previous measurement results. In line with the notation introduced in Section 5.2.1, for each $i \in [t+1]$, we define $\Phi_i \subseteq [n]$ be the set of wires such that $\theta_i \neq \perp$. That is, Φ_i is the set of wires on which the i'th partial measurement operates.*

Now, we will find it useful to introduce further notation drawing attention to which wires are simply measured in either the standard or Hamadard basis by the i'th partial ZX measurement, and which are not fully collapsed. In particular, we define disjoint sets $V_1, \ldots, V_{t+1}$ and sets $W_1, \ldots, W_t$ with the following properties.

- $\Phi_1 = (V_1, W_1), \Phi_2 = (V_1, V_2, W_2), \ldots, \Phi_t = (V_1, \ldots, V_t, W_t), \Phi_{t+1} = (V_1, \ldots, V_{t+1}) = [n]$.
- *The i'th measurement takes previously collapsed wires $V_1, \ldots, V_{i-1}$ as input, "fully" collapses wires V_i, and "partially" collapses wires W_i. This will be made precise by the evaluation procedure defined below, where the v_i are inputs from the V_i wires and w_i are inputs from the W_i wires.*
- *Each L_i does not operate on $\{V_j\}_{j<i}$. That is, fully collapsed registers are no longer computed on.*

Finally, given an input $x \in \{0,1\}^m$, let LMEval $\left(x, |\psi\rangle, \{L_i\}_{i\in[t+1]}, \{\theta_i\}_{i\in[t+1]}, \{f_i\}_{i\in[t]}, g\right) \to y$ *be the formal evaluation procedure, defined as follows:*

- *Initialize an n-qubit register $\mathcal{M}$ with $|\psi\rangle$.*
- *Compute $((v_1, r_1), \mathcal{M}) \leftarrow M_{\theta_1, f_1^x} \circ L_1(\mathcal{M})$, where $f_1^x(v_1, w_1) = (v_1, r_1)$.*

- *Compute* $((v_2, r_2), \mathcal{M}) \leftarrow M_{\theta_2, f_2^{x,r_1}} \circ L_2(\mathcal{M})$*, where* $f_2^{x,r_1}(v_1, v_2, w_2) = (v_2, r_2)$*.*
- $\ldots$
- *Compute* $((v_t, r_t), \mathcal{M}) \leftarrow M_{\theta_t, f_t^{x,r_1,\ldots,r_{t-1}}} \circ L_t(\mathcal{M})$*, where* $f_t^{x,r_1,\ldots,r_{t-1}}(v_1, \ldots, v_t, w_t) = (v_t, r_t)$*.*
- *Compute output* $y \leftarrow M_{\theta_{t+1}, g^{x,r_1,\ldots,r_t}} \circ L_{t+1}(\mathcal{M})$*, where* $g^{x,r_1,\ldots,r_t}(v_1, \ldots, v_{t+1}) = y$*.*

Observe that any alternating sequence of linear operations and partial ZX measurements may be written in the form introduced in the above definition, by simply defining each of the sets V_i to be empty, and writing each function as $f_i^{x,r_1,\ldots,r_{i-1}}(w_i) \to r_i$ (that is, we can always choose not to treat any of the wires as fully collapsed in the above formalism). So, why did we bother explicitly defining the V_i and W_i sets? The reason is that we will actually be interested in a "subclass" of LM quantum programs whose partially collapsed wires (the W_i wires) have a particularly simple structure. The diagram shown in Fig. 14 is indeed an example of such an LM quantum program.

Definition 5.14 (LM quantum program with standard-basis-collapsible W wires)**.** *An* LM *quantum program has* standard-basis-collapsible W wires *if:*

- $W_1, \ldots, W_n$ *consist of only standard basis indices, that is,* $\theta_{i,j} = 0$ *for* $i \in [t]$ *and* $j \in W_i$*.*
- *For* $i \in [t]$*,* W_i *is disjoint from* $\Phi_1 \cup \cdots \cup \Phi_{i-1}$*, and the operations* $L_1, \ldots, L_{i-1}$ *are either classically controlled on or do not operate on* W_i*. In particular, for each* $i \in [t]$*, the entire operation of the* LM *program up to and including the* i*'th measurement is diagonal in the standard basis on the wires* W_i*.*

This *standard-basis-collapsible W wires* property ensures that if one were to measure ("collapse") the $W_1, \ldots, W_n$ wires in the standard basis before executing the program, the W_i wires would remain completely unaffected throughout the execution of the program up to and *including* the i'th measurement (though they could be affected after the i'th measurement). Note that this is not a correctness property, indeed, collapsing the $W_1, \ldots, W_n$ wires at the beginning of the computation would likely completely change the desired functionality. However, it turns out that this property will be crucial for arguing the *security* of our obfuscation scheme in the following sections (in particular, refer to the "Collapsing the F oracles" discussion in the proof intuition section, Section 5.4.2).

5.3.2 Compiler

Theorem 5.15. *Any quantum program* $(|\psi\rangle, C)$ *(Definition 2.2) can be compiled into an equivalent* LM *quantum program* $(|\psi'\rangle, \{L_i\}_{i\in[t+1]}, \{\theta_i\}_{i\in[t+1]}, \{f_i\}_{i\in[t]}, g)$ *with standard-basis-collapsible W wires, where* $\{L_i\}_{i\in[t+1]}, \{\theta_i\}_{i\in[t+1]}, \{f_i\}_{i\in[t]}, g$ *only depend on the description of* C *(and not* $|\psi\rangle$*). Moreover, the compiler runs in polynomial time in the size of its input* $(|\psi\rangle, C)$*.*

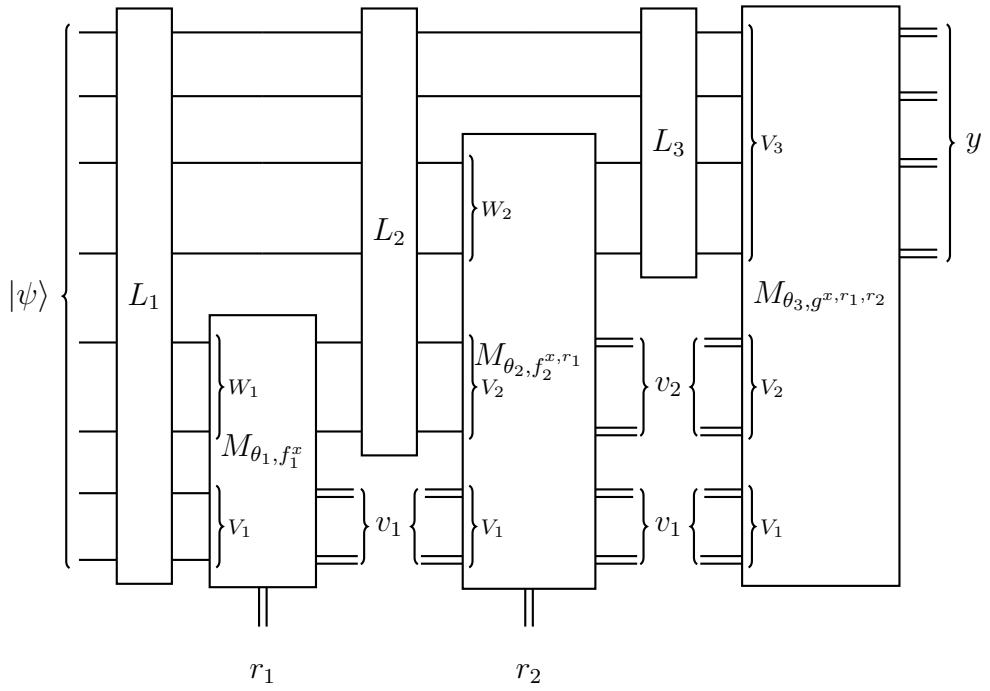

Figure 14: Diagram of an LM quantum program. We use some non-standard quantum circuit notation, so we provide some explanation. Each partial ZX measurement $M_{\theta_1,f_1^{(\cdot)}}, M_{\theta_2,f_2^{(\cdot)}}, M_{\theta_3,g^{(\cdot)}}$ is applied to the wires coming from the left of the corresponding box, some of which may be classical. Some wires (namely, V_1, V_2, and V_3) are fully collapsed by the measurement, producing classical output wires coming from the right. Other wires (namely, W_1 and W_2) are only partially collapsed, so their corresponding output wires are still quantum. Additional classical outputs (namely, r_1 and r_2) are produced by these measurements, which are denoted by classical wires coming out of the bottom. Note that the description of later measurements depend on r_1, r_2. Finally, we remark that one could instead introduce explicit ancillary wires for the input x and intermediate measurement results r_1, r_2, but writing the circuit in the manner above is visually suggestive of the structure of our eventual obfuscation scheme.

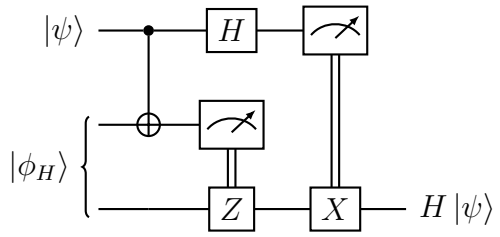

Figure 15: Implementation of the H gate with the H-magic state $|\phi_H\rangle \propto |00\rangle + |01\rangle + |10\rangle - |11\rangle$.

Proof. We will use a circuit representation very similar to that described in [BGS13], except for a key difference in how we implement the T gate, inspired by the encrypted CNOT operation introduced in [Mah18a]. We write the quantum circuit C using the $\{\mathsf{CNOT}, H, T\}$ universal gate set, where T is the gate that applies a phase of $e^{i\pi/4}$. Given magic states, we'll show how to implement H and T gates using only CNOT gates and Pauli (X and Z) gates controlled on the results of partial ZX measurements. Then, we will observe that the Pauli gates can be subsumed into the description of the measurements, leaving only layers of CNOT gates and partial ZX measurements.

First, we'll describe our implementations of the H and T gates and prove that they are correct. Then, we'll complete the proof with an inductive argument, showing how to build an LM quantum program one gate at a time.

Implementing the H gate. Following [BGS13], we use a two-qubit magic state

$$|\phi_H\rangle \propto |00\rangle + |01\rangle + |10\rangle - |11\rangle$$

to implement the Hadamard gate, via the circuit in Figure 15. For completeness, we show that the circuit indeed implements the Hadamard gate.

Claim 5.16. *The circuit in Figure 15 implements the Hadamard gate.*

Proof. Write $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. After the CNOT gate, the joint state of all three qubits can be written as

$$\begin{aligned}
&\alpha|000\rangle + \alpha|001\rangle + \alpha|010\rangle - \alpha|011\rangle + \beta|100\rangle - \beta|101\rangle + \beta|110\rangle + \beta|111\rangle \\
&\quad = \Big((\alpha+\beta)|+\rangle|00\rangle + (\alpha-\beta)|+\rangle|01\rangle\Big) + \Big((\alpha+\beta)|+\rangle|10\rangle - (\alpha-\beta)|+\rangle|11\rangle\Big) \\
&\quad + \Big((\alpha-\beta)|-\rangle|00\rangle + (\alpha+\beta)|-\rangle|01\rangle\Big) + \Big(\alpha-\beta)|-\rangle|10\rangle - (\alpha+\beta)|-\rangle|11\rangle\Big)
\end{aligned}$$

After the Hadamard basis measurement on the first wire resulting in a bit x and the standard basis measurement on the second wire resulting in a bit z, the resulting state on the

third wire is

$$\begin{aligned}
(\alpha+\beta)\,|0\rangle + (\alpha-\beta)\,|1\rangle &= H\,|\psi\rangle && \text{if } x=0 \text{ and } z=0\\
(\alpha+\beta)\,|0\rangle - (\alpha-\beta)\,|1\rangle &= ZH\,|\psi\rangle && \text{if } x=0 \text{ and } z=1\\
(\alpha-\beta)\,|0\rangle + (\alpha+\beta)\,|1\rangle &= XH\,|\psi\rangle && \text{if } x=1 \text{ and } z=0\\
(\alpha-\beta)\,|0\rangle - (\alpha+\beta)\,|1\rangle &= ZXH\,|\psi\rangle && \text{if } x=1 \text{ and } z=1
\end{aligned}$$

Applying the Z and X corrections now gives the state $H\,|\psi\rangle$. □

Implementing the T gate. We will use two magic states

$$|\phi_T\rangle \propto |0\rangle + e^{i\pi/4}\,|1\rangle \quad \text{and} \quad |\phi_{PX}\rangle \propto i\,|0\rangle + |1\rangle$$

and the circuit on the bottom right of Figure 16. First, we clarify notation in the figure. Γ_c is a projective measurement controlled on the bit c from the first wire, and is defined as follows.

- $\Gamma_0 = \{|00\rangle\langle 00| + |10\rangle\langle 10|\,, |01\rangle\langle 01| + |11\rangle\langle 11|\}$. That is, it measures its second input in the standard basis.
- $\Gamma_1 = \{|00\rangle\langle 00| + |11\rangle\langle 11|\,, |01\rangle\langle 01| + |10\rangle\langle 10|\}$. That is, it measures the XOR of its two inputs.

The measurement Γ_c is applied to the second and third wires, which remain quantum wires, and produces a classical bit r indicating which of the two measurement results was observed. In this figure, this bit r is carried on the classical wire coming from the right of Γ_c. In an abuse of notation, we will also use Γ_c as a function to define measurement outcomes:

$$\begin{aligned}
\Gamma_0(00) = \Gamma_0(10) = 0, \quad \Gamma_0(01) = \Gamma_0(11) = 1,\\
\Gamma_1(00) = \Gamma_1(11) = 0, \quad \Gamma_1(01) = \Gamma_1(10) = 1
\end{aligned}$$

The control logic for the Z gate is $c \cdot (r \oplus h)$, where c is the result of measuring the first wire, r is the result of measuring Γ_c, and h is the result of measuring the third wire in the Hadamard basis. We will now confirm this representation of the T gate works as expected.

Claim 5.17. *The bottom right circuit in Figure 16 implements the T gate.*

Proof. Write $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$. Applying the first CNOT yields

$$\begin{aligned}
&\alpha\,|00\rangle + e^{i\pi/4}\beta\,|01\rangle + \beta\,|10\rangle + e^{i\pi/4}\alpha\,|11\rangle\\
&= |0\rangle\,(\alpha\,|0\rangle + e^{i\pi/4}\beta\,|1\rangle) + |1\rangle\,(\beta\,|0\rangle + e^{i\pi/4}\alpha\,|1\rangle).
\end{aligned}$$

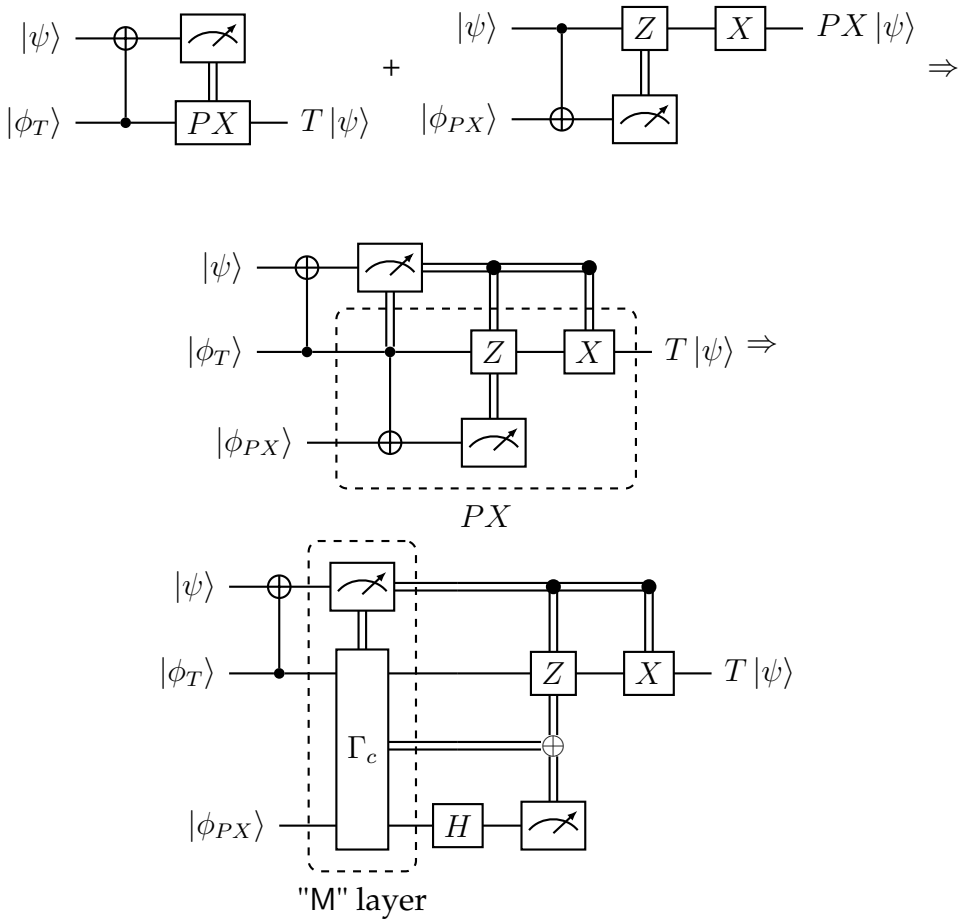

Figure 16: Implementation of the T gate. First, we combine an implementation of the T gate using the T-magic state $|\phi_T\rangle \propto |0\rangle + e^{i\pi/4}|1\rangle$ (upper left) with an implementation of the PX gate using the PX-magic state $|\phi_{PX}\rangle \propto i|0\rangle + |1\rangle$ (upper right) to obtain the circuit on the bottom left. This circuit includes a classically controlled CNOT gate, which is not supported by LM quantum programs. We replace the classically controlled CNOT with a classically controlled *projective measurement* to arrive at the circuit on the bottom right. Here, Γ_c represents a measurement controlled on the bit c from the first wire to be applied to the second and third wires. These wires are only partially collapsed by this measurement, so they remain quantum wires. However, Γ_c also produces a measurement result r, which is carried on the classical wire coming from the right. The final Z gate is controlled on the bit $c \cdot (r \oplus h)$, where h is the result of measuring the third wire in the Hadamard basis. The dashed box will eventually become a measurement layer in our implementation of an LM circuit (though we remark that the input for this measurement will also include wires from previous H-gate and T-gate circuits).

If the result of measuring the first wire is $c = 0$, the state on the second wire is already

$$\alpha |0\rangle + e^{i\pi/4}\beta |1\rangle = T |\psi\rangle .$$

In this case, we measure Γ_0, which only collapses the third wire, and neither the Z nor X corrections is applied to the second wire, which remains in the state $T |\psi\rangle$.

If the result of measuring the first wire is $c = 1$, then the second wire is in the state

$$\beta |0\rangle + e^{i\pi/4}\alpha |1\rangle .$$

In this case, we measure Γ_1 on

$$(\beta |0\rangle + e^{i\pi/4}\alpha |1\rangle)(i |0\rangle + |1\rangle).$$

If the result is $r = 0$, the state has collapsed to

$$\begin{aligned}
&i\beta |00\rangle + e^{i\pi/4}\alpha |11\rangle \\
&= e^{i\pi/4}\beta |00\rangle + \alpha |11\rangle \\
&= \left(e^{i\pi/4}\beta |0\rangle + \alpha |1\rangle\right) |+\rangle + \left(e^{i\pi/4}\beta |0\rangle - \alpha |1\rangle\right) |-\rangle \\
&= XT |\psi\rangle |+\rangle + ZXT |\psi\rangle |-\rangle ,
\end{aligned}$$

so applying the Z correction controlled on $c \cdot (r \oplus h) = h$ followed by the X correction results in $T |\psi\rangle$. If the result is $r = 1$, the state has collapsed to

$$\begin{aligned}
&\beta |01\rangle + ie^{i\pi/4}\alpha |10\rangle \\
&= e^{i\pi/4}\beta |01\rangle - \alpha |10\rangle \\
&= \left(e^{i\pi/4} |0\rangle - \alpha |1\rangle\right) |+\rangle - \left(e^{i\pi/4}\beta |0\rangle + \alpha |1\rangle\right) |-\rangle \\
&= ZXT |\psi\rangle |+\rangle - ZT |\psi\rangle |-\rangle ,
\end{aligned}$$

so applying the Z correction controlled on $c \cdot (r \oplus h) = 1 \oplus h$ followed by the X correction results in $T |\psi\rangle$. □

Inductive argument. In order to carry out an inductive argument, we will first generalize the notion of an LM quantum program to support quantum output. We will actually allow the output to be correct up to some Pauli errors that can be computed by measuring some ancillary registers in the standard or Hadamard basis and applying a classical function to the measurement results.

First, we fix some notation. Throughout the proof, we'll keep track of disjoint sets of wires $V_1, \ldots, V_t, V_{t+1}^*, O$, where $V_1 \cup \cdots \cup V_t \cup V_{t+1}^* \cup O = [n]$. We will also keep track of strings $v, \widehat{x}, \widehat{z} \in \{0, 1, \perp\}^n$, where v denotes a subset of measurement results (from wires $V_1 \cup \cdots \cup V_t \cup V_{t+1}^*$), and $\widehat{x}, \widehat{z}$ denote Pauli corrections to be applied to the wires in O. For any set V, we let $v^{(V)}$ (resp. $\widehat{x}^{(V)}, \widehat{z}^{(V)}$) be the string restricted to indices in the set V. In particular, for an index $i \in [n]$, $v^{(i)}$ is just the i'th entry of v. Finally, for each $i \in [t]$, we define $v_i \coloneqq v^{(V_i)}$, and we define $v_{t+1}^* \coloneqq v^{(V_{t+1}^*)}$.

Definition 5.18 (LM quantum program with Pauli-encoded quantum output)**.** *An* LM *quantum program with Pauli-encoded quantum output is defined like a standard* LM *quantum program (Definition 5.13), except for the following differences.*

- *There is no final measurement $M_{\theta_{t+1},g^{(\cdot)}}$, and thus θ_{t+1} and $g^{(\cdot)}$ are undefined.*
- *The set $[n] \setminus (V_1 \cup \cdots \cup V_t)$ consists of disjoint sets V^*_{t+1} and O, where O contains the Pauli-encoded output. We will refer to O as the "active" set of wires.*
- *There is a string $\theta^*_{t+1} \in \{0,1,\perp\}^n$ that is 0 or 1 on V^*_{t+1} (determining whether these registers are measured in the standard or Hadamard basis) and $\perp$ everywhere else.*
- *There is a classical function $h(x, v_1, \ldots, v_t, v^*_{t+1}, r_1, \ldots, r_t) \to \{0,1\}^{2|O|}$ that operates on the program input x and previous measurement results, and outputs Pauli corrections $\widehat{x}^{(O)}, \widehat{z}^{(O)} \in \{0,1\}^{|O|}$ to be applied to the O registers.*

Note that the program is defined by a state $|\psi\rangle$ along with $\{L_i\}_{i\in[t+1]}, \{\theta_i\}_{i\in[t]}, \{f_i^{(\cdot)}\}_{i\in[t]}, \theta^*_{t+1}, h$.

First, the following claim will confirm that it suffices to compile the quantum program $(|\psi\rangle, C)$ into an LM quantum program with Pauli-encoded quantum output.

Claim 5.19. *Consider any* LM *quantum program with Pauli-encoded quantum output that computes a classical output functionality. That is, the (Pauli encoding of the) output we are interested in is determined by measuring the O registers in the standard basis. Then, this program can be written as a standard* LM *quantum program (Definition 5.13).*

Proof. It suffices to define the final measurement $M_{\theta_{t+1},g^{(\cdot)}}$. Set $\theta_{t+1} \in \{0,1\}^n$ to be equal to θ_t on the sets $V_1, \ldots, V_t$, equal to θ^*_{t+1} on the set V^*_{t+1} and equal to 0 on the set O. Let $g^{x,r_1,\ldots,r_t}(v_1, \ldots, v_{t+1})$ be defined as follows. Parse v_{t+1} as (v^*_{t+1}, y'), compute $h(x, v_1, \ldots, v_t, v^*_{t+1}, r_1, \ldots, r_t) = (\widehat{x}^{(O)}, \widehat{z}^{(O)})$, and output $y = y' \oplus \widehat{x}^{(O)}$. □

Now, we show how to compile any quantum program $(|\psi\rangle, C)$ into an LM quantum program with Pauli-encoded quantum output. We proceed by induction over the number of gates ℓ in C.

Base case. Suppose that C contains 0 gates. That is, there is no state $|\psi\rangle$ and the functionality is just the identity applied to input x. In this case, $n = |x|$, $t = 0$, L_1 is empty, $\theta^*_1 = \perp^n$, and the LM quantum program is defined by $(|0^n\rangle, h)$, where $h(x) = (x, 0^n)$.

Inductive step. Consider a quantum program $(|\psi\rangle, C)$ with $\ell + 1$ gates, and begin by writing C as (C_ℓ, G), where C_ℓ contains the first ℓ gates, and $G \in \{\mathsf{CNOT}, H, T\}$ is the final gate. By the inductive hypothesis, we know that $(|\psi\rangle, C_\ell)$ can be written as an LM quantum program with Pauli-encoded quantum output: $|\psi'\rangle, \{L_i\}_{i\in[t+1]}, \{\theta_i\}_{i\in[t]}, \{f_i^{(\cdot)}\}_{i\in[t]}, \theta^*_{t+1}, h$, where t is the number of T gates in C_ℓ. Now, we consider three cases corresponding to the gate G, which by definition will be applied to one (or two) wire(s) in the "active" set O.

In each case, we describe how to update the description of the LM quantum program for $(|\psi\rangle, C_\ell)$ so that it now has the same functionality as the full quantum program $(|\psi\rangle, C)$. The fact that these updates implement the desired functionality follow from Claim 5.16 and Claim 5.17 above.

- CNOT from wire i to j: Append the description of this gate to the end of L_{t+1} and append the operation $(\widehat{x}^{(i)}, \widehat{z}^{(i)}), (\widehat{x}^{(j)}, \widehat{z}^{(j)}) \to (\widehat{x}^{(i)}, \widehat{z}^{(i)} \oplus \widehat{z}^{(j)}), (\widehat{x}^{(i)} \oplus \widehat{x}^{(j)}, \widehat{z}^{(j)})$ to the end of h.
- H on wire i: Refer to Fig. 15.
  - Introduce two new wires $(n+1, n+2)$ and append $|\phi_H\rangle$ to $|\psi'\rangle$.
  - Append the description of a CNOT gate from wire i to $n+1$ to the end of L_{t+1}.
  - Remove wire i from and add wire $n+2$ to O.
  - Add wires i and $n+1$ to V^*_{t+1}.
  - For $\tau \in [t]$, define $\theta_{\tau,n+1} = \theta_{\tau,n+2} = \perp$. Define $\theta^*_{t+1,i} = 1$, $\theta^*_{t+1,n+1} = 0$, and $\theta^*_{t+1,n+2} = \perp$.
  - Update h to h' as follows. The function h' will now take two additional input bits $v^{(i)}, v^{(n+1)}$ as part of v^*_{t+1} and its output will now include $(\widehat{x}^{(n+2)}, \widehat{z}^{(n+2)})$ rather than $(\widehat{x}^{(i)}, \widehat{z}^{(i)})$, computed as follows. Let $\widehat{x}^{(O)}, \widehat{z}^{(O)} = h(x, v_1, \ldots, v_t, v^*_{t+1}, r_1, \ldots, r_t)$ be the output of the original h, which includes $(\widehat{x}^{(i)}, \widehat{z}^{(i)})$. Then the output of h' includes $\widehat{x}^{(n+2)} := v^{(i)} \oplus \widehat{z}^{(i)}$ and $\widehat{z}^{(n+2)} := v^{(n+1)} \oplus \widehat{x}^{(i)}$.
- T on wire i: Refer to Fig. 16.
  - Introduce two new wires $(n+1, n+2)$ and append $|\phi_T\rangle |\psi_{PX}\rangle$ to $|\psi'\rangle$.
  - Append the description of a CNOT gate from wire $n+1$ to i to the end of L_{t+1}.
  - Remove wire i from and add wire $n+1$ to O.
  - Define $V_{t+1} := V^*_{t+1} \cup \{i\}$ and define $W_{t+1} := \{n+1, n+2\}$.
  - For $\tau \in [t]$, define $\theta_{\tau,n+1} = \theta_{\tau,n+2} = \perp$. Define θ_{t+1} to be equal to θ_t on the sets $V_1, \ldots, V_t$, equal to θ^*_{t+1} on the set V^*_{t+1}, equal to 0 on index i, and equal to $\perp$ everywhere else.
  - Define a function $f^{x,r_1,\ldots,r_t}_{t+1}(v_1, \ldots, v_{t+1}, w_{t+1})$ as follows, where $v_{t+1} = (v^*_{t+1}, v^{(i)})$. First, compute $(\widehat{x}^{(O)}, \widehat{z}^{(O)}) = h(x, v_1, \ldots, v_t, v^*_{t+1}, r_1, \ldots, r_t)$, which includes $(\widehat{x}^{(i)}, \widehat{z}^{(i)})$. Then, set $c = v^{(i)} \oplus \widehat{x}^{(i)}$, and output $(v_{t+1}, \Gamma_c(w_{t+1}))$. This defines measurement $M_{\theta_{t+1}, f^{(\cdot)}_{t+1}}$.
  - Initialize $V^*_{t+2} := \{n+2\}$, θ^*_{t+2} to be equal to 1 at index $n+2$ and $\perp$ everywhere else, and L_{t+2} to be empty.

– Update h to h' as follows. The function h' will now take an additional input bit $v^{(i)}$ as part of v_{t+1}, an additional input bit $v^{(n+2)}$ as part of v^*_{t+2}, and an additional input bit r_{t+1}. Its output will now include $(\widehat{x}^{(n+1)}, \widehat{z}^{(n+1)})$ rather than $(\widehat{x}^{(i)}, \widehat{z}^{(i)})$, computed as follows. Let $\widehat{x}^{(O)}, \widehat{z}^{(O)} = h(x, v_1, \ldots, v_t, v^*_{t+1}, r_1, \ldots, r_t)$ be the output of the original h, which includes $(\widehat{x}^{(i)}, \widehat{z}^{(i)})$. Then the output of h' includes $\widehat{x}^{(n+1)} := v^{(i)} \oplus \widehat{x}^{(i)}$ and $\widehat{z}^{(n+1)} := (v^{(i)} \oplus \widehat{x}^{(i)}) \cdot (v^{(n+2)} \oplus r_{t+1})$.

This completes the description of the compiler. Observe that the W_i wires consist of the two magic state wires used to implement the i'th T gate (one initialized with $|\phi_T\rangle$ and the other initialized with $|\phi_{PX}\rangle$). The $|\phi_T\rangle$ wire is used as the control for a single CNOT gate just prior to the i'th measurement, and the $|\phi_{PX}\rangle$ wire is not touched until the i'th measurement. Thus, since Γ_c is a standard basis projector, all the requirements of the *standard-basis-collapsible W wires* property (Definition 5.14) are fulfilled.

□

5.4 Obfuscation

5.4.1 Construction

Our construction of quantum state obfuscation in the classical oracle model makes use of the following ingredients.

- Publicly-verifiable, linearly-homomorphic QAS with classically-decodable ZX measurements $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{LinEval}, \mathsf{Dec}, \mathsf{Ver})$, defined in Section 5.2.
- Signature token $(\mathsf{TokGen}, \mathsf{TokSign}, \mathsf{TokVer})$, defined in Section 2.2.4.
- A pseudorandom function F_k secure against superposition-query attacks [Zha12].

For any polynomials $n = n(\lambda)$, $m = m(\lambda)$, and $m' = m'(\lambda)$, let

$$\left\{|\psi^{\mathsf{unv}}_{\lambda,n,m,m'}\rangle, C^{\mathsf{unv}}_{\lambda,n,m,m'}\right\}_{\lambda\in\mathbb{N}}$$

be the family of *universal* (n, m, m') quantum programs. That is, for any family of quantum programs $\{|\phi_\lambda\rangle, C_\lambda\}_{\lambda\in\mathbb{N}}$ where $|\phi_\lambda\rangle$ has at most n qubits, C_λ has as most n gates, and $Q : \{0,1\}^m \to \{0,1\}^{m'}$, it holds that for all λ and inputs x,

$$C^{\mathsf{unv}}_{\lambda,n,m,m'}\left(|x\rangle |C_\lambda\rangle |\phi_\lambda\rangle |\psi^{\mathsf{unv}}_{\lambda,n,m,m'}\rangle\right) = C_\lambda\left(|x\rangle |\phi_\lambda\rangle\right).$$

By Theorem 5.15, for any (n, m, m') family $\{|\phi_\lambda\rangle, C_\lambda\}_{\lambda\in\mathbb{N}}$, we can write each quantum program

$$|C_\lambda\rangle |\phi_\lambda\rangle |\psi^{\mathsf{unv}}_{\lambda,n,m,m'}\rangle, C^{\mathsf{unv}}_{\lambda,n,m,m'}$$

as an LM quantum program

$$\left(|\psi\rangle, \{L_i\}_{i\in[t+1]}, \{\theta_i\}_{i\in[t+1]}, \{f_i\}_{i\in[t]}, g\right)$$

that satisfies the *standard-basis-collapsible W wires* property (Definition 5.14), where we have dropped the indexing by λ to reduce notational clutter. Note that Theorem 5.15 guarantees that $|\psi\rangle$ contains the complete description of $(|\phi_\lambda\rangle, C_\lambda)$, and that the classical part of the program $(\{L_i\}_{i\in[t+1]}, \{\theta_i\}_{i\in[t+1]}, \{f_i\}_{i\in[t]}, g)$ only depends on $C^{\mathsf{unv}}_{\lambda,n,m,m'}$. Thus, we consider everything but $|\psi\rangle$ to be public, and our obfuscator will take as input a quantum state $|\psi\rangle$, and its goal is to hide $|\psi\rangle$. Finally, we assume without loss of generality that each r_i (being part of the output of f_i) is a single bit, which is convenient (though not strictly necessary) for describing the construction and proof, and is satisfied by the output of the compiler given in Theorem 5.15.

The construction is given in Protocol 17, and incorporates the following public parameters:

- Security parameter λ.
- Classical part of the LM quantum program $\{L_i\}_{i\in[t+1]}, \{\theta_i\}_{i\in[t+1]}, \{f_i\}_{i\in[t]}, g$. This information determines the number of qubits $n = \mathrm{poly}(\lambda)$ in the input state $|\psi\rangle$, classical input size $m = \mathrm{poly}(\lambda)$, and classical output size $m'(\lambda)$. Recall from Section 5.2.1 that each θ_i defines subsets

  $$\Phi_{\theta_i}, \Phi_{\theta_i,0}, \Phi_{\theta_i,1}, \Phi_{\theta_i,\perp}, \widetilde{\Phi}_{\theta_i}, \widetilde{\Phi}_{\theta_i,0}, \widetilde{\Phi}_{\theta_i,1}, \widetilde{\Phi}_{\theta_i,\perp},$$

  and in what follows we drop the θ and write these subsets as

  $$\Phi_i, \Phi_{i,0}, \Phi_{i,1}, \Phi_{i,\perp}, \widetilde{\Phi}_i, \widetilde{\Phi}_{i,0}, \widetilde{\Phi}_{i,1}, \widetilde{\Phi}_{i,\perp}.$$

- Derived security parameter $\kappa := \max\{\lambda, n^4\} = \mathrm{poly}(\lambda)$.

We will also make use of the following notation. Given a register $\mathcal{X}$ and classical functionality F, we let

$$(y, \mathcal{X}) \leftarrow \mathsf{F}(\mathcal{X})$$

denote the result of initializing a new register $\mathcal{Y}$, coherently applying the map

$$|x\rangle^{\mathcal{X}} |0\rangle^{\mathcal{Y}} \to |x\rangle^{\mathcal{X}} |\mathsf{F}(x)\rangle^{\mathcal{Y}},$$

and then measuring register $\mathcal{Y}$ to obtain output y.

Theorem 5.20. *The scheme described in Fig. 17 is a quantum state obfuscator that satisfies correctness (Definition 2.13).*

Proof. Fix the classical part of an LM quantum program $\{L_i\}_{i\in[t+1]}, \{\theta_i\}_{i\in[t+1]}, \{f_i\}_{i\in[t]}, g$, and let $x, |\psi\rangle$ be such that there exists y such that

$$\Pr\left[\mathsf{LMEval}\left(x, |\psi\rangle, \{L_i\}_{i\in[t+1]}, \{\theta_i\}_{i\in[t+1]}, \{f_i\}_{i\in[t]}, g\right) \to y\right] = 1 - \mathrm{negl}(\lambda).$$

Technically, we mean an infinite family of programs, inputs, states, and outputs, parameterized by the security parameter λ, but we keep this implicit. We will show via a sequence of hybrids that

Quantum State Obfuscation

$\mathsf{QSObf}\left(1^\lambda, |\psi\rangle\right)$:

- Sample $k \leftarrow \mathsf{Gen}(1^\kappa, n)$, and compute $|\psi_k\rangle = \mathsf{Enc}_k(|\psi\rangle)$.
- Sample a signature token $(\mathsf{vk}, |\mathsf{sk}\rangle) \leftarrow \mathsf{TokGen}(1^\kappa)$.
- Sample $k' \leftarrow \{0,1\}^\lambda$ for PRF $F_{k'} : \{0,1\}^* \to \{0,1\}^\kappa$ and let $H(\cdot) \coloneqq F_{k'}(\cdot)$.
- For each $i \in [t]$, define the function $\mathsf{F}_i\left(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1}\right)$:
  - Output $\perp$ if $\mathsf{TokVer}(\mathsf{vk}, x, \sigma_x) = \perp$. Otherwise, for each $\iota \in [i-1]$, let
    $$\ell_{\iota,0} = H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 0), \;\; \ell_{\iota,1} = H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 1),$$
    and output $\perp$ if $\ell_{\iota,0} = \ell_{\iota,1}$ or $\ell_\iota \notin \{\ell_{\iota,0}, \ell_{\iota,1}\}$. Otherwise, let r_ι be such that $\ell_\iota = h_{\iota,r_\iota}$.
  - Compute $(v_1, \ldots, v_i, w_i) = \mathsf{Dec}_{k, L_i \ldots L_1, \theta_i}(\widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i)$ and output $\perp$ if the result is $\perp$.
  - Compute $(\cdot, r_i) = f_i^{x, r_1, \ldots, r_{i-1}}(v_1, \ldots, v_i, w_i)$.
  - Set $\ell_i \coloneqq H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \ell_1, \ldots, \ell_{i-1}, r_i)$, and output $(\widetilde{v}_i, \ell_i)$.
- Define the function $\mathsf{G}(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_{t+1}, \ell_1, \ldots, \ell_t)$:
  - Output $\perp$ if $\mathsf{TokVer}(\mathsf{vk}, x, \sigma_x) = \perp$. Otherwise, for each $\iota \in [t]$, let
    $$\ell_{\iota,0} = H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 0), \;\; \ell_{\iota,1} = H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 1),$$
    and output $\perp$ if $\ell_{\iota,0} = \ell_{\iota,1}$ or $\ell_\iota \notin \{\ell_{\iota,0}, \ell_{\iota,1}\}$. Otherwise, let r_ι be such that $\ell_\iota = \ell_{\iota,r_\iota}$.
  - Compute $(v_1, \ldots, v_{t+1}) = \mathsf{Dec}_{k, L_{t+1} \ldots L_1, \theta_{t+1}}(\widetilde{v}_1, \ldots, \widetilde{v}_{t+1})$ and output $\perp$ if the result is $\perp$.
  - Output $y = g^{x, r_1, \ldots, r_t}(v_1, \ldots, v_{t+1})$.
- Output $|\widetilde{\psi}\rangle = |\psi_k\rangle\,|\mathsf{sk}\rangle\,, O = (\mathsf{F}_1, \ldots, \mathsf{F}_t, \mathsf{G})$.

$\mathsf{QSEval}^O\left(x, |\widetilde{\psi}\rangle\right)$:

- Sample $\sigma_x \leftarrow \mathsf{TokSign}(x, |\mathsf{sk}\rangle)$.
- Initialize a register $\mathcal{C}$ with $|\psi_k\rangle$, and do the following for $i \in [t]$:
  - $\mathcal{C} \leftarrow H^{\widetilde{\Phi}_{i,1}} \mathsf{LinEval}_{L_i}(\mathcal{C})$.
  - Measure $\left(\widetilde{v}_i, \ell_i, \mathcal{C}_{\widetilde{\Phi}_i}\right) \leftarrow \mathsf{F}_i\left(x, \sigma_x, \mathcal{C}_{\widetilde{\Phi}_i}, \ell_1, \ldots, \ell_{i-1}\right)$.
  - $\mathcal{C} \leftarrow H^{\widetilde{\Phi}_{i,1}}(\mathcal{C})$.
- $\mathcal{C} \leftarrow H^{\widetilde{\Phi}_{t+1,1}} \mathsf{LinEval}_{L_{t+1}}(\mathcal{C})$
- Measure $y \leftarrow \mathsf{G}\left(x, \sigma_x, \mathcal{C}_{\widetilde{\Phi}_{t+1}}, \ell_1, \ldots, \ell_t\right)$ and output y.

Figure 17: Construction of quantum state obfuscation.

$$\Pr\left[y^* = y : \begin{array}{c} |\widetilde{\psi}\rangle, O \leftarrow \mathsf{QSObf}(1^\lambda, |\psi\rangle) \\ y^* \leftarrow \mathsf{QSEval}^O\left(x, |\widetilde{\psi}\rangle\right) \end{array}\right] = 1 - \mathrm{negl}(\lambda).$$

Each hybrid will describe a distribution over y^*, beginning with the distribution above, which we denote $\mathcal{H}_0$.

- $\mathcal{H}_1$: This is the same as $\mathcal{H}_0$ except that $H(\cdot)$ is defined to be a uniformly random function with range $\{0,1\}^\kappa$ rather than the PRF $F_{k'}$.
- $\mathcal{H}_2$: This is the same as $\mathcal{H}_1$ except that the functions $\mathsf{F}_1, \ldots, \mathsf{F}_t, \mathsf{G}$ ignore their input σ_x and don't apply TokVer.
- $\mathcal{H}_3$: This is the same as $\mathcal{H}_2$ except that instead of inputting and outputting the labels ℓ_ι, the functions $F_1, \ldots, F_t, G$ directly input and output the bits r_ι. That is, these functions are defined as follows.

  $\mathsf{F}_i(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, r_1, \ldots, r_{i-1})$:

  - Compute $(v_1, \ldots, v_i, w_i) = \mathsf{Dec}_{k, L_i \ldots L_1, \theta_i}(\widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i)$ and output $\perp$ if the result is $\perp$.
  - Compute $(\cdot, r_i) = f_i^{x, r_1, \ldots, r_{i-1}}(v_1, \ldots, v_i, w_i)$.
  - Output $(\widetilde{v}_i, r_i)$.

  $\mathsf{G}(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_{t+1}, r_1, \ldots, r_t)$:

  - Compute $(v_1, \ldots, v_{t+1}) = \mathsf{Dec}_{k, L_{t+1} \ldots L_1, \theta_{t+1}}(\widetilde{v}_1, \ldots, \widetilde{v}_{t+1})$ and output $\perp$ if the result is $\perp$.
  - Output $y = g^{x, r_1, \ldots, r_t}(v_1, \ldots, v_{t+1})$.
- $\mathcal{H}_4$: This is the same as $\mathcal{H}_3$ except that the functions $\mathsf{F}_1, \ldots, \mathsf{F}_t$ output v_ι rather than $\widetilde{v}_\iota$. That is, these functions are defined as follows.

  $\mathsf{F}_i(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, r_1, \ldots, r_{i-1})$:

  - Compute $(v_1, \ldots, v_i, w_i) = \mathsf{Dec}_{k, L_i \ldots L_1, \theta_i}(\widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i)$ and output $\perp$ if the result is $\perp$.
  - Compute $(\cdot, r_i) = f_i^{x, r_1, \ldots, r_{i-1}}(v_1, \ldots, v_i, w_i)$.
  - Output (v_i, r_i).

  $\mathsf{G}(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_t, \widetilde{v}_{t+1}, r_1, \ldots, r_t)$:

  - Compute $v_1, \ldots, v_{t+1} = \mathsf{Dec}_{k, L_{t+1} \ldots L_1, \theta_{t+1}}(\widetilde{v}_1, \ldots, \widetilde{v}_{t+1})$ and output $\perp$ if the result is $\perp$.

– Output $y = g^{x,r_1,\ldots,r_t}(v_1,\ldots,v_{t+1})$.

To complete the proof, we combine the following observations.

- $\mathcal{H}_0 \approx_{\mathrm{negl}(\lambda)} \mathcal{H}_1$: This follows from the (superposition-query) security of the PRF.
- $\mathcal{H}_1 \equiv \mathcal{H}_2$: This follows from the correctness of the signature token (Definition 2.20).
- $\mathcal{H}_2 \approx_{\mathrm{negl}(\lambda)} \mathcal{H}_3$: The only difference between these hybrids occurs if in $\mathcal{H}_2$, a query to F_i or G outputs $\bot$ due to the fact that $\ell_{\iota,0} = \ell_{\iota,1}$. Since H is a uniformly random function, each $\ell_{\iota,0} = \ell_{\iota,1}$ with probability $1/2^\kappa = \mathrm{negl}(\lambda)$, and the observation follows because $t = \mathrm{poly}(\lambda)$.
- $\mathcal{H}_3 \equiv \mathcal{H}_4$: Starting with $\mathcal{H}_4$, in which the logical measurements of $v_1,\ldots,v_{t+1}$ are performed, we can imagine, after the i'th measurement, further collapsing the V_i register to obtain the outcome $\widetilde{v}_i$ as in $\mathcal{H}_3$. This has no effect on the rest of the computation, since these registers are no longer computed on after the i'th measurement, and will continue to decode to v_i.
- $\mathcal{H}_4 \equiv \mathsf{LMEval}\left(x, |\psi\rangle, \{L_i\}_{i\in[t+1]}, \{\theta_i\}_{i\in[t+1]}, \{f_i\}_{i\in[t]}, g\right)$: This follows from the correctness of the authentication scheme (Definition 5.1). Indeed, for a given key $k \in \mathsf{Gen}(1^\kappa, n)$, we can write the distribution sampled by $\mathcal{H}_4$ as follows:

  – Initialize register $\mathcal{C}$ to $\mathsf{Enc}_k(|\psi\rangle)$.
  – Compute $((v_1,r_1),\mathcal{C}) \leftarrow \widetilde{M}_{\theta_1,f_1^x,k,L_1} \circ \mathsf{LinEval}_{L_1}(\mathcal{C})$.
  – $\ldots$
  – Compute $((v_t,r_t),\mathcal{C}) \leftarrow \widetilde{M}_{\theta_t,f_t^{x,r_1,\ldots,r_{t-1}},k,L_t\ldots L_1} \circ \mathsf{LinEval}_{L_t}(\mathcal{C})$.
  – Compute output $y \leftarrow \widetilde{M}_{\theta_{t+1},g^{x,r_1,\ldots,r_t},k,L_{t+1}\ldots L_1} \circ \mathsf{LinEval}_{L_{t+1}}(\mathcal{C})$.

  Now, we apply the expression in the definition of correctness (Definition 5.1) to the first measurement to obtain an equivalent sampling procedure:

  – Initialize register $\mathcal{M}$ to $|\psi\rangle$.
  – Compute $((v_1,r_1),\mathcal{M}) \leftarrow L_1^\dagger \circ M_{\theta_1,f_1^x} \circ L_1(\mathcal{M})$.
  – Compute $\mathcal{C} \leftarrow \mathsf{Enc}_k(\mathcal{M})$
  – Compute $((v_2,r_2),\mathcal{C}) \leftarrow \widetilde{M}_{\theta_2,f_2^{x,r_1},k,L_2L_1} \circ \mathsf{LinEval}_{L_2L_1}(\mathcal{C})$.
  – $\ldots$
  – Compute $((v_t,r_t),\mathcal{C}) \leftarrow \widetilde{M}_{\theta_t,f_t^{x,r_1,\ldots,r_{t-1}},k,L_t\ldots L_1} \circ \mathsf{LinEval}_{L_t}(\mathcal{C})$.
  – Compute output $y \leftarrow \widetilde{M}_{\theta_{t+1},g^{x,r_1,\ldots,r_t},k,L_{t+1}\ldots L_1} \circ \mathsf{LinEval}_{L_{t+1}}(\mathcal{C})$.

  By applying the expression iteratively for each measurement, we obtain:

- Initialize register $\mathcal{M}$ to $|\psi\rangle$.
- Compute $((v_1, r_1), \mathcal{M}) \leftarrow L_1^\dagger \circ M_{\theta_1, f_1^x} \circ L_1(\mathcal{M})$.
- Compute $((v_2, r_2), \mathcal{M}) \leftarrow L_1^\dagger L_2^\dagger \circ M_{\theta_2, f_2^{x,r_1}} \circ L_2 L_1(\mathcal{M})$.
- ...
- Compute $((v_t, r_t), \mathcal{M}) \leftarrow L_1^\dagger \dots L_t^\dagger \circ M_{\theta_t, f_t^{x,r_1,\dots,r_{t-1}}} \circ L_t \dots L_1(\mathcal{M})$.
- Compute output $y \leftarrow M_{\theta_{t+1}, g^{x,r_1,\dots,r_t}} \circ L_{t+1} \dots L_1(\mathcal{M})$.

By canceling $L_i^\dagger L_i = \mathcal{I}$, we obtain $\mathsf{LMEval}\left(x, |\psi\rangle, \{L_i\}_{i\in[t+1]}, \{\theta_i\}_{i\in[t+1]}, \{f_i\}_{i\in[t]}, g\right)$:

- Initialize register $\mathcal{M}$ to $|\psi\rangle$.
- Compute $((v_1, r_1), \mathcal{M}) \leftarrow M_{\theta_1, f_1^x} \circ L_1(\mathcal{M})$.
- Compute $((v_2, r_2), \mathcal{M}) \leftarrow M_{\theta_2, f_2^{x,r_1}} \circ L_2(\mathcal{M})$.
- ...
- Compute $((v_t, r_t), \mathcal{M}) \leftarrow M_{\theta_t, f_t^{x,r_1,\dots,r_{t-1}}} \circ L_t(\mathcal{M})$.
- Compute output $y \leftarrow M_{\theta_{t+1}, g^{x,r_1,\dots,r_t}} \circ L_{t+1}(\mathcal{M})$.

□

5.4.2 Proof intuition

We begin by discussing three main ideas used in our proof. This will not be a step-by-step outline of the proof, rather, it will try to convey the main intuitive ideas. Broadly speaking, we will want to simulate the oracles $\mathsf{F}_1, \dots, \mathsf{F}_t, \mathsf{G}$ so that they no longer require access to the decoding functionality of the authentication scheme, and G can get by with just oracle access to the induced functionality Q. Once this is done, we can appeal to privacy of the authentication scheme (Definition 5.5) in order to switch $|\psi\rangle$ to $|0^n\rangle$, thus removing all information about the input state.

The first idea below will help us simulate the G oracle, the second will us help simulate the F oracles, and the third is a way to extract signature tokens from the adversary using a purified random oracle, which we will use when proving the indistinguishability of the simulated oracles.

Proving soundness by induction. As mentioned in the technical overview (Section 5.1.3), one of the main steps in our proof of security is to show the following soundness guarantee. Fix any input x^*. Then we would like to show that, given $|\psi\rangle$ and oracle access to $\mathsf{F}_1, \dots, \mathsf{F}_t, \mathsf{G}$, the adversary cannot prepare a "bad" query $(x^*, \sigma_{x^*}, \widetilde{v}_1, \dots, \widetilde{v}_{t+1}, \ell_1, \dots, \ell_t)$ with the property that

$$\mathsf{G}(x^*, \sigma_{x^*}, \widetilde{v}_1, \dots, \widetilde{v}_{t+1}, \ell_1, \dots, \ell_t) \notin \{Q(x^*), \perp\}.$$

That is, if G does not abort on an input that starts with x^*, then it better be the case that it returns the correct output $Q(x^*)$.

To simplify the discussion for now, let's consider the simpler case of a garbled program, which only allows the adversary to evaluate on a single input x^*. That is, suppose we hard-code x^* into the oracles $\mathsf{F}_1[x^*], \ldots, \mathsf{F}_t[x^*], \mathsf{G}[x^*]$, which now only accept inputs that begin with x^*.

Our goal will be to show that the adversary is "forced" to follow an honest evaluation path on input x^*. Since the honest evaluation path actually branches at each measurement, we will essentially analyze each of these possible branching executions. To do so, let's suppose by induction that the soundness condition holds for any program with $t-1$ measurement layers. Then, for each possible outcome (v_1, r_1) of the first measurement (using input x^*) that occurs with non-zero probability, define $\Pi[x^*, v_1, r_1]$ to be the projector onto the space of initial states $|\psi\rangle$ that produce that outcome. Thus, we can write

$$|\psi\rangle = \sum_{v_1, r_1} \Pi[x^*, v_1, r_1] \, |\psi\rangle \, ,$$

and analyze each component $|\widetilde{\psi}[x^*, v_1, r_1]\rangle \coloneqq \mathsf{Enc}_k(\Pi[x^*, v_1, r_1] \, |\psi\rangle)$ separately.

The key step is to show that, if the adversary is initialized with $|\widetilde{\psi}[x^*, v_1^*, r_1^*]\rangle$ for some (v_1^*, r_1^*), then we can *hard-code* the measurement results (v_1^*, r_1^*) into the oracles $\mathsf{F}_1[x^*], \ldots, \mathsf{F}_t[x^*]$ without the adversary noticing. That is, we define oracles $\mathsf{F}_1[x^*, v_1^*, r_1^*], \ldots, \mathsf{F}_t[x^*, v_1^*, r_1^*]$ that operate like $\mathsf{F}_1[x^*], \ldots, \mathsf{F}_t[x^*]$, except that $\mathsf{F}_1[x^*, v_1^*, r_1^*]$ always outputs the label representing r_1^*, and $\mathsf{F}_2[x^*, v_1^*, r_1^*], \ldots, \mathsf{F}_t[x^*, v_1^*, r_1^*]$ use (v_1^*, r_1^*) instead of decoding their inputs $\widetilde{v}_1$ and ℓ_1.

We will prove the indistinguishability of $\mathsf{F}_1[x^*], \ldots, \mathsf{F}_t[x^*]$ and $\mathsf{F}_1[x^*, v_1^*, r_1^*], \ldots, \mathsf{F}_t[x^*, v_1^*, r_1^*]$ by reducing to security of the authentication scheme. Note that distinguishing these oracles requires the adversary to find a *differing input* to one of the oracles. Now, assuming that the oracles can be simulated using $\mathsf{Ver}_{k,\cdot,\cdot}(\cdot)$ instead of $\mathsf{Dec}_{k,\cdot,\cdot}(\cdot)$ (which we have yet to argue, but will address in the following section), this means that it suffices to show that the adversary cannot map

$$|\widetilde{\psi}[x^*, v_1^*, r_1^*]\rangle \to |\widetilde{\psi}[x^*, v_1, r_1]\rangle$$

for some $(v_1, r_1) \neq (v_1^*, r_1^*)$ just given access to the verification oracle $\mathsf{Ver}_{k,\cdot,\cdot}(\cdot)$. However, doing so would certainly imply that the adversary can change the measurement results of an authenticated state (given the verification oracle), which violates the security of the authentication scheme.

Finally, we view the state $|\widetilde{\psi}[x^*, v_1^*, r_1^*]\rangle$ and oracles $\mathsf{F}_1[x^*, v_1^*, r_1^*], \ldots, \mathsf{F}_t[x^*, v_1^*, r_1^*]$ as an example of a garbled $(t-1)$-layer program, and finish the proof of soundness by appealing to the induction hypothesis.

The formal inductive argument is given in Section 5.4.5.

Collapsing the F oracles. Now, we address the claim made above that the F oracles can be simulated given $\mathsf{Ver}_{k,\cdot,\cdot}(\cdot)$ instead of $\mathsf{Dec}_{k,\cdot,\cdot}(\cdot)$. Note that we can only hope that this

simulation is indistinguishable to an adversary with no access to the G oracle, since the real F oracles can be used to actually implement the computation $x \to Q(x)$, while the simulated oracles cannot since they don't actually decode their inputs. However, it turns out that this suffices for us, since we can simulate the G oracle when we need to apply this indistinguishability.

The main idea is to "collapse" the oracles $\mathsf{F}_1, \ldots, \mathsf{F}_t$, showing that the adversary cannot distinguish them from oracles $\mathsf{FSim}_1, \ldots, \mathsf{FSim}_t$ that *always* output either the "zero" label

$$H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \ell_1, \ldots, \ell_{i-1}, 0)$$

or $\perp$. These oracles now do not have to actually run $f_i^{x, r_1, \ldots, r_{i-1}}(v_1, \ldots, v_i, w_i)$ to compute the bit r_i, meaning that the decoding operation in F_i can be replaced with a verification operation.

But how do we show that the oracles can be collapsed? Again, we will use the idea of splitting $|\psi\rangle$ up into orthogonal components and analyzing each component separately. Here is where we make use of the *standard-basis-collapsible W wires* property of the LM quantum program (Definition 5.14). In particular, we will define the orthogonal components by measuring the wires $W_1, \ldots, W_t$ in the standard basis. That is, we will write

$$|\psi\rangle = \sum_w \Pi[w] |\psi\rangle ,$$

where $\Pi[w]$ is the projection of wires $W_1, \ldots, W_t$ onto standard basis measurement results $w = (w_1, \ldots, w_t)$.

The point is that if the oracle F_i only receives inputs that include encodings $\widetilde{w} = (\widetilde{w}_1, \ldots, \widetilde{w}_t)$ of some *fixed* $w = (w_1, \ldots, w_t)$, then, for each "prefix" $(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \ell_1, \ldots, \ell_{i-1})$, it will only ever query the random oracle H on

$$\text{either} \quad (x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \ell_1, \ldots, \ell_{i-1}, 0) \quad \text{or} \quad (x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \ell_1, \ldots, \ell_{i-1}, 1),$$

where the last bit is a deterministic function of w and the prefix. But since each of these values is distributed as a uniformly random string, this behavior is identical (from the adversary's perspective) to, for each prefix, always querying the zero label

$$H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \ell_1, \ldots, \ell_{i-1}, 0).$$

Thus, it suffices to show, roughly, that given $|\widetilde{\psi}[w]\rangle := \mathsf{Enc}_k(\Pi[w] |\psi\rangle)$, the adversary cannot map

$$|\widetilde{\psi}[w]\rangle \to |\widetilde{\psi}[w']\rangle$$

for $w' \neq w$, given access to the verification oracle $\mathsf{Ver}_{k,\cdot,\cdot}(\cdot)$. This again follows directly from security of our authentication scheme.

The ideas sketched here are used to simulate the F oracles in our main sequence of hybrids given in Section 5.4.4, and also in lower-level hybrids in Section 5.4.6.

Extracting signature tokens. Recall that the inductive argument sketched above for proving the soundness condition assumed that the oracles only respond on a single fixed input x^*. Unfortunately, as discussed in Section 5.1.3, the situation gets more complicated when we grant the adversary access to the oracles on any input x of their choice. The reason is that it is no longer clear that the adversary cannot perform the map

$$|\widetilde{\psi}[x^*, v_1^*, r_1^*]\rangle \to |\widetilde{\psi}[x^*, v_1, r_1]\rangle$$

by using an oracle query on an input $x \neq x^*$. Indeed, while the (V_1, W_1) registers of $|\widetilde{\psi}[x^*, v_1^*, r_1^*]\rangle$ are collapsed to a state that yields a fixed $(v_1^*, r_1^*) \leftarrow f_1^{x^*}(V_1, W_1)$, applying $f_1^x(V_1, W_1)$ for some $x \neq x^*$ might *disturb* the registers V_1, W_1 (since $f_1^{x^*}$ and f_1^x might be different functions!), thus changing the outcome of $f_1^{x^*}(V_1, W_1)$.

Now, as discussed in Section 5.1.3, we use signature tokens to prevent this potential attack. Recall that the oracle F_1 only responds on input x if additionally given a valid signature token σ_x. Thus, we will want to formalize the following claim: if the adversary uses $\mathsf{F}_1(x, \dots)$ to perform some "non-trivial" operation on the authenticated state $|\widetilde{\psi}[x^*, v_1^*, r_1^*]\rangle$, it is possible to *extract* a valid signature σ_x from the adversary. Once this σ_x is extracted, the security of the signature token scheme implies that the adversary won't be able to continue evaluating on x^*, and, in particular, we won't have to worry about the adversary breaking the soundness condition for input x^*.

We will show this claim by purifying the random oracle [Zha19], introducing a "database" register that is initialized with a different state in uniform superposition for each input to H. Then, we'll argue that if the adversary has used $\mathsf{F}_i(x, \cdot)$ to execute a measurement on the authenticated state, the database register must be disturbed at some inputs that begin with (x, σ_x). Thus, an extractor can simply measure the database register in the Hadamard basis, and obtain a signature on x by observing which registers were no longer in uniform superposition.

For the purpose of this overview, we consider a simplified version of this problem that still conveys the fundamental ideas in our proof. We'll take the random oracle to have a single bit of output, only consider the authentication of a single qubit state, and analyze the concrete authentication scheme based on coset states (though we stress that our eventual proof just makes use of generic properties of the authentication scheme). Here is the setup:

- An authentication key $k = (S, \Delta, x, z)$ is sampled.
- The adversary A is given an authenticated 0 state $X^x Z^z |S\rangle$ along with access to the following oracle O that can be used to implement a logical Hadamard basis measurement:

$$O(\widetilde{v}) = \begin{cases} H(0) \text{ if } \widetilde{v} \in \widehat{S} + z \\ H(1) \text{ if } \widetilde{v} \in \widehat{S} + \widehat{\Delta} + z \\ \perp \text{ otherwise} \end{cases},$$

  where H is a random oracle $\{0,1\} \to \{0,1\}$. H will be purified and implemented using a database register initialized to $|{+}{+}\rangle$.

- We claim that the adversary cannot produce any vector in $S+\Delta+x$ (that is, a vector in the support of an authenticated 1 state) at the same time that the database register is in the state $|{+}{+}\rangle$:

$$\mathbb{E}\left[\left\| \left(\Pi[S+\Delta+x]\otimes|{+}{+}\rangle\langle{+}{+}|\right) A^{O}\left(X^{x}Z^{z}|S\rangle\right)|{+}{+}\rangle \right\|^{2}\right] = \mathrm{negl}(\lambda).$$

  To be clear, A operates on input state $X^{x}Z^{z}|S\rangle$ (along with some potential extra workspace), and the database registers are operated on by O when answering A's queries.

Notice that it is easy for the adversary to produce just a vector in $S+\Delta+x$ (with constant probability) by using the oracle O to honestly to implement a Hadamard basis measurement. The trick is to show that once they do this, it is impossible for them to make queries to O that return the state of the database to $|{+}{+}\rangle$, while still remembering their vector in $S+\Delta+x$.

Our first step is to decompose $X^{x}Z^{z}|S\rangle$ into orthogonal components corresponding to the authenticated plus and minus state. That is,

$$X^{x}Z^{z}|S\rangle = \frac{1}{\sqrt{2}}H^{\otimes 2\lambda+1}X^{z}Z^{x}|\widehat{S}\rangle + \frac{1}{\sqrt{2}}H^{\otimes 2\lambda+1}X^{z}Z^{x}|\widehat{S}+\widehat{\Delta}\rangle := |\widetilde{+}\rangle + |\widetilde{-}\rangle\,.$$

Then, for $b\in\{0,1\}$, we define oracles

$$O[b](\widetilde{v}) = \begin{cases} H(b) \text{ if } \widetilde{v}\in\widehat{S}_{\widehat{\Delta}}+z \\ \perp \text{ otherwise} \end{cases},$$

that are identical to O, except that they always query the random oracle on bit b. Then we observe that

$$A^{O}|\widetilde{+}\rangle \approx_{\mathrm{negl}(\lambda)} A^{O[0]}|\widetilde{+}\rangle\,, \quad \text{and} \quad A^{O}|\widetilde{-}\rangle \approx_{\mathrm{negl}(\lambda)} A^{O[1]}|\widetilde{-}\rangle\,.$$

This follows from the security of the authentication scheme, which implies that A cannot map between $|\widetilde{+}\rangle$ and $|\widetilde{-}\rangle$. That is, on input $|\widetilde{+}\rangle$, A won't be able to find any input on which O and $O[0]$ differ, and on input $|\widetilde{-}\rangle$, A won't be able to find any input on which O and $O[1]$ differ.

Next, we observe that the state of the system that results from using oracle $O[1]$ is actually equivalent to the state that results from first swapping the database registers of H, using $O[0]$, and then swapping back. Thus, it holds that

$$
\begin{aligned}
&\mathbb{E}\left[\left\|\left(\Pi[S+\Delta+x]\otimes|{+}{+}\rangle\langle{+}{+}|\right)A^{O}\left(X^{x}Z^{z}|S\rangle\right)|{+}{+}\rangle\right\|^{2}\right]\\
&=\mathbb{E}\left[\left\|\left(\Pi[S+\Delta+x]\otimes|{+}{+}\rangle\langle{+}{+}|\right)\left(A^{O}|\widetilde{+}\rangle|{+}{+}\rangle+A^{O}|\widetilde{-}\rangle|{+}{+}\rangle\right)\right\|^{2}\right]\\
&\approx_{\mathrm{negl}(\lambda)}\mathbb{E}\left[\left\|\left(\Pi[S+\Delta+x]\otimes|{+}{+}\rangle\langle{+}{+}|\right)\left(A^{O[0]}|\widetilde{+}\rangle|{+}{+}\rangle+A^{O[1]}|\widetilde{-}\rangle|{+}{+}\rangle\right)\right\|^{2}\right]\\
&=\mathbb{E}\left[\left\|\left(\Pi[S+\Delta+x]\otimes|{+}{+}\rangle\langle{+}{+}|\right)\left(A^{O[0]}|\widetilde{+}\rangle|{+}{+}\rangle+\mathsf{SWAP}A^{O[0]}|\widetilde{-}\rangle\,\mathsf{SWAP}|{+}{+}\rangle\right)\right\|^{2}\right]\\
&=\mathbb{E}\left[\left\|\Pi[S+\Delta+x]\left(|{+}{+}\rangle\langle{+}{+}|A^{O[0]}|\widetilde{+}\rangle|{+}{+}\rangle+|{+}{+}\rangle\langle{+}{+}|\mathsf{SWAP}A^{O[0]}|\widetilde{-}\rangle\,\mathsf{SWAP}|{+}{+}\rangle\right)\right\|^{2}\right]\\
&=\mathbb{E}\left[\left\|\Pi[S+\Delta+x]\left(|{+}{+}\rangle\langle{+}{+}|A^{O[0]}|\widetilde{+}\rangle|{+}{+}\rangle+|{+}{+}\rangle\langle{+}{+}|A^{O[0]}|\widetilde{-}\rangle|{+}{+}\rangle\right)\right\|^{2}\right]\\
&=\mathbb{E}\left[\left\|\left(\Pi[S+\Delta+x]\otimes|{+}{+}\rangle\langle{+}{+}|\right)\left(A^{O[0]}|\widetilde{+}\rangle|{+}{+}\rangle+A^{O[0]}|\widetilde{-}\rangle|{+}{+}\rangle\right)\right\|^{2}\right]\\
&=\mathbb{E}\left[\left\|\left(\Pi[S+\Delta+x]\otimes|{+}{+}\rangle\langle{+}{+}|\right)A^{O[0]}X^{x}Z^{z}|S\rangle|{+}{+}\rangle\right\|^{2}\right]\\
&=\mathrm{negl}(\lambda),
\end{aligned}
$$

where the last step follows from security of the authentication scheme, since $O[0]$ can be implemented with just the verification oracle of the authentication scheme. Note that we crucially used the fact that we are projecting back onto $|{+}{+}\rangle\langle{+}{+}|$ in the step where we remove the left-most SWAP operation, which follows because $\mathsf{SWAP}|{+}{+}\rangle = |{+}{+}\rangle$. Indeed, as discussed above, the claim would not be true without the projection onto $|{+}{+}\rangle\langle{+}{+}|$, since the adversary *can* obtain a vector in $\Pi[S+\Delta+x]$ while disturbing the database register.

To conclude, we note that this same logic can be extended to more general measurements on more general authenticated states, which ultimately will be used to extract a valid signature on x from any adversary that is actively using the oracles $\mathsf{F}_1,\ldots,\mathsf{F}_t,\mathsf{G}$ to evaluate the computation on input x. This implies that the adversary cannot launch mixed input attacks, which is one of the main hurdles to overcome in proving the security of our quantum state obfuscator.

The ideas sketched here are used in Section 5.4.6, and in particular in the proof of Lemma 5.45.

5.4.3 Notation

In this section, we review some important notation and define new notation that will be used throughout the proof.

- $|\psi\rangle$ is the n-qubit input state.
- $\{L_i\}_{i\in[t+1]},\{\theta_i\}_{i\in[t+1]},\{f_i\}_{i\in[t]},g$ is the classical part of the description of an LM quantum program.

- Parameters: m is the size of the classical input, and κ is a derived security parameter that is sufficiently larger than λ, n. We will often use the facts that $n \geq t, m$ and $\kappa = \omega(n)$.

- See Definition 5.13 for the definition of sets $(V_1, \ldots, V_{t+1})$ and $(W_1, \ldots, W_t)$. The i'th measurement for $i \in [t]$ operates on $(V_1, \ldots, V_i, W_i)$ and the $t+1$'st measurement operates on $(V_1, \ldots, V_{t+1}) = [n]$. We will assume that the LM quantum program has *standard-basis-collapsible W wires* (Definition 5.14) so in particular, $\theta_{i,j} = 0$ for all $j \in W_i$ (that is, W_i are standard basis registers for the i'th measurement).

- $k = (S, \Delta, x, z)$ is a key for the authentication scheme.

- See Section 5.2.1 for the description of our partial ZX measurement notation $M_{\theta,f}$ and $\widetilde{M}_{\theta,f,k,L}$. In particular,

$$\left\{M_{\theta_i, f_i^{x, r_1, \ldots, r_{i-1}}}\right\}_{i \in [t]}, M_{\theta_{t+1}, g^{x, r_1, \ldots, r_t}}$$

  are the sequence of measurements applied by the LM quantum program, and

$$\left\{\widetilde{M}_{\theta_i, f_i^{x, r_1, \ldots, r_{i-1}}, k, L_i \ldots L_1}\right\}_{i \in [t]}, \widetilde{M}_{\theta_{t+1}, g_i^{x, r_1, \ldots, r_t}, k, L_{t+1} \ldots L_1}$$

  are the corresponding sequence of measurements applied to qubits authenticated using the key k.

- We will often refer to a partial set of measurement results $\{v_\iota^*, r_\iota^*\}_{\iota \in [\tau]}$ for some $\tau \in [t+1]$. Note that in the case $\tau = t+1$, the value r_{t+1}^* will always be empty, since the final measurement of an LM quantum program only outputs v_{t+1}. We only include this empty value so that the notation is consistent across each measurement layer.

- Fix any input x^* and partial set of measurement results $\{v_\iota^*, r_\iota^*\}_{\iota \in [\tau]}$ for some $\tau \in [t+1]$. First, we explicitly define the projectors that constitute the M measurements:

$$M_{\theta_\tau, f_\tau^{x^*, r_1^*, \ldots, r_{\tau-1}^*}} := \left\{\Pi^{x^*, r_1^*, \ldots, r_{\tau-1}^*}[v_\tau, r_\tau]\right\}_{v_\tau, r_\tau},$$

$$M_{\theta_{t+1}, g^{x^*, r_1^*, \ldots, r_t^*}} := \left\{\Pi^{x^*, r_1^*, \ldots, r_t^*}[v_{t+1}]\right\}_{v_{t+1}}.$$

  Next, we define a (sub-normalized) initial state for each set of partial measurement results $\{v_\iota^*, r_\iota^*\}_{\iota \in [\tau]}$:

$$|\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota \in [\tau]}]\rangle := L_1^\dagger \ldots L_\tau^\dagger \Pi^{x^*, r_1^*, \ldots, r_{\tau-1}^*}[v_\tau^*, r_\tau^*] L_\tau \ldots \Pi^{x^*}[v_1^*, r_1^*] L_1 |\psi\rangle.$$

- During the proof, we will make use of the collapsible W wires property, and consider measuring (some subset of) the W wires in the standard basis at the beginning of the computation. For any string of measurement results $w_i^* \in \{0,1\}^{|W_i|}$, define corresponding sets

$$P[w_i^*] := \left\{\widetilde{w}_i : \mathsf{Dec}_{k,\emptyset,\theta_i[W_i]}(\widetilde{w}_i) = w_i^*\right\}, \quad P[\neg w_i^*] := \left\{\widetilde{w}_i : \mathsf{Dec}_{k,\emptyset,\theta_i[W_i]}(\widetilde{w}_i) \notin \{w_i^*, \bot\}\right\},$$

where $\emptyset$ indicates an empty sequence of CNOT gates, in other words, the identity. Also recall the notation $\theta[V]$ defined in Section 5.2.1 indicating a string that is equal to θ on the subset of indices V and $\bot$ everywhere else. Next, define the projectors

$$\Pi[w^*] := \Pi[P[w_i^*]], \quad \Pi[\neg w^*] := \Pi[P[\neg w_i^*]].$$

Finally, for any set $S \subseteq [t]$ and $\{w_i^*\}_{i\in S}$ where each $w_i^* \in \{0,1\}^{|W_i|}$, define

$$\Pi[\{w_i^*\}_{i\in S}] := \bigotimes_{i\in S} \Pi[w_i^*], \quad \Pi[\neg\{w_i^*\}_{i\in S}] := \sum_{i\in S} \Pi[\neg w_i^*].$$

5.4.4 Main theorem

Theorem 5.21. *For any $\epsilon = \epsilon(\lambda) = \mathrm{negl}(\lambda) \cdot 2^{-2m(\lambda)}$, the scheme described in Protocol 17 is a quantum state obfuscator that satisfies ideal obfuscation (Definition 2.13) for ϵ-pseudo-deterministic families of quantum programs.*

Remark 5.22. *One might hope that the above theorem could be shown for any $\epsilon = \mathrm{negl}(\lambda)$, and we leave this open. However, we remark that in the case where the input program has a completely classical description (e.g. the case handled by [BKNY23]), one can first repeat the circuit $\mathrm{poly}(\lambda)$ times to generically go from $\mathrm{negl}(\lambda)$-pseudo-determinism to $\mathrm{negl}(\lambda)\cdot 2^{-2m(\lambda)}$-pseudo-determinism. Thus, this result captures a strictly more general class of programs than [BKNY23]. Moreover, the application to best-possible copy-protection [CG24] only requires obfuscating* fully *deterministic computation.*

Proof. Throughout this proof, we will often drop the dependence of functions and circuit families on the parameter λ in order to reduce notational clutter. Let n, m, m' be any polynomials (in λ), and suppose that $|\psi\rangle, \{L_i\}_{i\in[t+1]}, \{\theta_i\}_{i\in[t+1]}, \{f_i\}_{i\in[t]}, g$ is an LM quantum program such that $|\psi\rangle$ has at most n qubits, there are at most n gates in the circuit, and the classical input has m bits. Suppose that this LM quantum program is ϵ-pseudo-deterministic for a small enough ϵ as specified by the theorem statement, and let Q be the induced map of this LM quantum program. Consider any QPT[40] adversary A and distinguisher D, and define $D[A]$ to be the procedure that runs A, feeds its output to D, and then

[40]The only reason that we restrict our adversary to be quantum polynomial-*time* as opposed to quantum polynomial-*query* is the very first step in our proof, where we replace the PRF with a random oracle. If we allow the obfuscation scheme to use a true random oracle (thus sacrificing the efficiency of the oracles), then we obtain security against any QPQ (quantum polynomial-query, see Section 2.1) adversary.

runs D to produce a binary-valued outcome. Thus, we can write the "real" obfuscation experiment as

$$\Pr\left[1 \leftarrow D\left(A\left(\mathsf{QSObf}\left(1^\lambda, |\psi\rangle\right)\right)\right)\right] = \mathop{\mathbb{E}}_{\left(|\widetilde{\psi}\rangle, O\right) \leftarrow \mathsf{QSObf}(1^\lambda, |\psi\rangle)} \left[\left\| D[A]^O \, |\widetilde{\psi}\rangle \right\|^2\right].$$

Now, we will consider a sequence of hybrid distributions over (state, oracle) pairs $(|\widetilde{\psi}\rangle, O)$, beginning with the real distribution $\mathsf{QSObf}_0 := \mathsf{QSObf}$, and ending with a fully simulated distribution QSObf_6 that no longer needs to take $|\psi\rangle$ as input (and instead uses oracle access to Q). Our first step will be to switch the oracle G to a simulated oracle GSim that *verifies* rather than *decodes* the intermediate labels and final authenticated measurement, and uses oracle access to Q to respond in the case that verification passes. Next, we'll "collapse" the oracles $\mathsf{F}_1, \ldots, \mathsf{F}_t$ as described in Section 5.4.2, using a strategy derived from the collapsible W wires property of the LM quantum program. Finally, we'll replace the input $|\psi\rangle$ with the all zeros input $|0^n\rangle$.

The description of these distributions follow (but no claims about indistinguishability yet). The difference between adjacent distributions are highlighted in red, and whenever we write w^*, we parse it at $w^* = \{w_i^*\}_{i \in [t]}$, where each $w_i^* \in \{0,1\}^{|W_i|}$.

$\mathsf{QSObf}_1(1^\lambda, |\psi\rangle)$:

- Sample $k \leftarrow \mathsf{Gen}(1^\kappa, n)$, and compute $|\psi_k\rangle = \mathsf{Enc}_k(|\psi\rangle)$.
- Sample a signature token $(\mathsf{vk}, |\mathsf{sk}\rangle) \leftarrow \mathsf{TokGen}(1^\kappa)$.
- Let $H : \{0,1\}^* \to \{0,1\}^\kappa$ be a random oracle.
- Define $\mathsf{F}_1, \ldots, \mathsf{F}_t$ as in QSObf_0.
- Define G as in QSObf_0.
- Output $|\widetilde{\psi}\rangle = |\psi_k\rangle\,|\mathsf{sk}\rangle$, $O = (\mathsf{F}_1, \ldots, \mathsf{F}_t, \mathsf{G})$.

$\mathsf{QSObf}_2(1^\lambda, |\psi\rangle)$:

- Sample $k \leftarrow \mathsf{Gen}(1^\kappa, n)$, and compute $|\psi_k\rangle = \mathsf{Enc}_k(|\psi\rangle)$.
- Sample a signature token $(\mathsf{vk}, |\mathsf{sk}\rangle) \leftarrow \mathsf{TokGen}(1^\kappa)$.
- Let $H : \{0,1\}^* \to \{0,1\}^\kappa$ be a random oracle.
- Define $\mathsf{F}_1, \ldots, \mathsf{F}_t$ as in QSObf_0.
- Define the function GSim $(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_{t+1}, \ell_1, \ldots, \ell_t)$:
  - Output $\bot$ if $\mathsf{TokVer}(\mathsf{vk}, x, \sigma_x) = \bot$.
  - For each $\iota \in [t]$, let

$$\ell_{\iota,0} = H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 0), \;\; \ell_{\iota,1} = H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 1),$$

    and output $\bot$ if $\ell_{\iota,0} = \ell_{\iota,1}$ or $\ell_\iota \notin \{\ell_{\iota,0}, \ell_{\iota,1}\}$.

- Output $\bot$ if $\mathsf{Ver}_{k,L_{t+1}\ldots L_1,\theta_{t+1}}(\widetilde{v}_1,\ldots,\widetilde{v}_{t+1}) = \bot$.
- Output $Q(x)$.

- Output $|\widetilde{\psi}\rangle = |\psi_k\rangle\,|\mathsf{sk}\rangle\,, O = (\mathsf{F}_1,\ldots,\mathsf{F}_t,\mathsf{GSim})$.

$\mathsf{QSObf}_3(1^\lambda, |\psi\rangle)$:

- Sample $k \leftarrow \mathsf{Gen}(1^\kappa, n)$, and compute $|\psi_k\rangle = \mathsf{Enc}_k(|\psi\rangle)$.
- Sample a signature token $(\mathsf{vk}, |\mathsf{sk}\rangle) \leftarrow \mathsf{TokGen}(1^\kappa)$.
- Let $H : \{0,1\}^* \to \{0,1\}^\kappa$ be a random oracle.
- For each $i \in [t]$, define the function $\mathsf{F}_i[w^*]\,(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1})$:
  - Output $\bot$ if $\mathsf{TokVer}(\mathsf{vk}, x, \sigma_x) = \bot$.
  - For each $\iota \in [i-1]$, let

    $$\ell_{\iota,0} = H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 0),\ \ \ell_{\iota,1} = H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 1),$$

    and output $\bot$ if $\ell_{\iota,0} = \ell_{\iota,1}$ or $\ell_\iota \notin \{\ell_{\iota,0}, \ell_{\iota,1}\}$. Otherwise, let r_ι be such that $\ell_\iota = \ell_{\iota,r_\iota}$.
  - Compute $(v_1, \ldots, v_i, \cdot) = \mathsf{Dec}_{k,L_i\ldots L_1,\theta_i}(\widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i)$ and output $\bot$ if the result is $\bot$.
  - Compute $(\cdot, r_i) = f_i^{x,r_1,\ldots,r_{i-1}}(v_1, \ldots, v_i, w_i^*)$.
  - Set $\ell_i := H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \ell_1, \ldots, \ell_{i-1}, r_i)$, and output $(\widetilde{v}_i, \ell_i)$.
- Define GSim as in QSObf_2.
- Output $|\widetilde{\psi}\rangle = |\psi_k\rangle\,|\mathsf{sk}\rangle\,, O = (\mathsf{F}_1[\cdot],\ldots,\mathsf{F}_t[\cdot],\mathsf{GSim})$.

$\mathsf{QSObf}_4(1^\lambda, |\psi\rangle)$:

- Sample $k \leftarrow \mathsf{Gen}(1^\kappa, n)$, and compute $|\psi_k\rangle = \mathsf{Enc}_k(|\psi\rangle)$.
- Sample a signature token $(\mathsf{vk}, |\mathsf{sk}\rangle) \leftarrow \mathsf{TokGen}(1^\kappa)$.
- Let $H : \{0,1\}^* \to \{0,1\}^\kappa$ be a random oracle.
- For each $i \in [t]$, define the function $\mathsf{F}_i[w^*]\,(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1})$:
  - Output $\bot$ if $\mathsf{TokVer}(\mathsf{vk}, x, \sigma_x) = \bot$.
  - For each $\iota \in [i-1]$, let

    $$\ell_{\iota,0} = H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 0),\ \ \ell_{\iota,1} = H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 1),$$

    and output $\bot$ if $\ell_{\iota,0} = \ell_{\iota,1}$ or $\ell_\iota \notin \{\ell_{\iota,0}, \ell_{\iota,1}\}$. ~~Otherwise, let r_ι be such that $\ell_\iota = \ell_{\iota,r_\iota}$.~~
  - Compute $(v_1, \ldots, v_i, \cdot) = \mathsf{Dec}_{k,L_i\ldots L_1,\theta_i}(\widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i)$ and output $\bot$ if the result is $\bot$.
  - For $\iota \in [i]$, compute $(\cdot, r_\iota) = f_\iota^{x,r_1,\ldots,r_{\iota-1}}(v_1, \ldots, v_\iota, w_\iota^*)$.
  - Set $\ell_i := H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \ell_1, \ldots, \ell_{i-1}, r_i)$, and output $(\widetilde{v}_i, \ell_i)$.
- Define GSim as in QSObf_2.
- Output $|\widetilde{\psi}\rangle = |\psi_k\rangle\,|\mathsf{sk}\rangle\,, O = (\mathsf{F}_1[\cdot],\ldots,\mathsf{F}_t[\cdot],\mathsf{GSim})$.

$\mathsf{QSObf}_5(1^\lambda, |\psi\rangle)$:

- Sample $k \leftarrow \mathsf{Gen}(1^\kappa, n)$, and compute $|\psi_k\rangle = \mathsf{Enc}_k(|\psi\rangle)$.
- Sample a signature token $(\mathsf{vk}, |\mathsf{sk}\rangle) \leftarrow \mathsf{TokGen}(1^\kappa)$.
- Let $H : \{0,1\}^* \to \{0,1\}^\kappa$ be a random oracle.
- For each $i \in [t]$, define the function $\mathsf{FSim}_i\,(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1})$:
  - Output $\perp$ if $\mathsf{TokVer}(\mathsf{vk}, x, \sigma_x) = \perp$.
  - For each $\iota \in [i-1]$, let
    $$\ell_{\iota,0} = H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 0), \;\; \ell_{\iota,1} = H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 1),$$
    and output $\perp$ if $\ell_{\iota,0} = \ell_{\iota,1}$ or $\ell_\iota \notin \{\ell_{\iota,0}, \ell_{\iota,1}\}$.
  - Output $\perp$ if $\mathsf{Ver}_{k, L_i \ldots L_1, \theta_i}(\widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i) = \perp$.
  - Set $\ell_i := H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \ell_1, \ldots, \ell_{i-1}, 0)$, and output $(\widetilde{v}_i, \ell_i)$.
- Define GSim as in QSObf_2.
- Output $|\widetilde{\psi}\rangle = |\psi_k\rangle\,|\mathsf{sk}\rangle\,, O = (\mathsf{FSim}_1, \ldots, \mathsf{FSim}_t, \mathsf{GSim})\,.$

$\mathsf{QSObf}_6(1^\lambda)$:

- Sample $k \leftarrow \mathsf{Gen}(1^\kappa, n)$, and compute $|\psi_k\rangle = \mathsf{Enc}_k(|0^n\rangle)$.
- Sample a signature token $(\mathsf{vk}, |\mathsf{sk}\rangle) \leftarrow \mathsf{TokGen}(1^\kappa)$.
- Let $H : \{0,1\}^* \to \{0,1\}^\kappa$ be a random oracle.
- Define $\mathsf{FSim}_1, \ldots, \mathsf{FSim}_t$ as in QSObf_4.
- Define GSim as in QSObf_2.
- Output $|\widetilde{\psi}\rangle = |\psi_k\rangle\,|\mathsf{sk}\rangle\,, O = (\mathsf{FSim}_1, \ldots, \mathsf{FSim}_t, \mathsf{GSim})\,.$

Later, we will use these distributions to define a sequence of hybrids starting with the real obfuscation experiment and ending with the simulated obfuscation experiment. But first, we will establish several claims about these distributions that will be useful while arguing indistinguishability of the hybrids.

This first claim establishes that, once the oracles are simulated, no adversary can map a state whose W wires have been collapsed to outcome w^* onto the support of a state with different outcomes $w \neq w^*$. At the end of this sequence of claims, we will have established that this property holds *even in* QSObf_2, where the F_i oracles are not yet simulated.

Claim 5.23. *For any QPQ (quantum polynomial-query) unitary U and any $w^* = \{w_i^*\}_{i\in[t]}$, it holds that*

$$\mathbb{E}\left[\left\|\Pi[\neg w^*]U^O\Pi[w^*]\,|\widetilde{\psi}\rangle\right\|^2 : |\widetilde{\psi}\rangle\,, O \leftarrow \mathsf{QSObf}_5(1^\lambda, |\psi\rangle)\right] = 2^{-\Omega(\kappa)}.$$

Proof. The key point is that in QSObf_5, none of the oracles $\mathsf{FSim}_1, \ldots, \mathsf{FSim}_t, \mathsf{GSim}$ require access to the the decryption oracle $\mathsf{Dec}_{k,\cdot,\cdot}(\cdot)$ for the authentication scheme. Rather, they

can be implemented just given access to the verification oracle $\mathsf{Ver}_{k,\cdot,\cdot}(\cdot)$. Now, since the W wires are authenticated, the hardness of mapping from the support of w^* to w for any $w \neq w^*$ follows directly from the security of the authentication scheme (Theorem 5.9), and in particular that it satisfies Definition 5.4 (mapping security). □

In the proof of the next two claims, we will use the following notation. Fix a key k and $w^* = \{w_i^*\}_{i\in[t]}$, and define the following function.

$R[k, w^*] : (x, \widetilde{v}_1, \ldots, \widetilde{v}_i) \to r_i$

- Compute $(v_1, \ldots, v_i) = \mathsf{Dec}_{k,L_i \ldots L_1, \theta_i[V_1,\ldots,V_i]}(\widetilde{v}_1, \ldots, \widetilde{v}_i)$ and output $\bot$ if the result is $\bot$.
- For $\iota \in [i]$, compute $(\cdot, r_\iota) = f_\iota^{x, r_1, \ldots, r_{\iota-1}}(v_1, \ldots, v_\iota, w_\iota^*)$.
- Output r_i.

This function determines the bit r_i when the w^* outcomes have been hard-coded into the oracles, and we will use it when showing indistinguishability between $\mathsf{QSObf}_3, \mathsf{QSObf}_4$, and QSObf_5 in the case where the W wires of the input state have been collapsed to outcome w^*.

Claim 5.24. *For any (unbounded) distinguisher D and any $w^* = \{w_i^*\}_{i\in[t]}$, it holds that*

$$\mathbb{E}\left[\left\|D^{\mathsf{F}_1[w^*],\ldots,\mathsf{GSim}}\left(k, \Pi[w^*]\left|\widetilde{\psi}\right\rangle\right)\right\|^2 : \left|\widetilde{\psi}\right\rangle, (\mathsf{F}_1[\cdot], \ldots, \mathsf{F}_t[\cdot], \mathsf{GSim}) \leftarrow \mathsf{QSObf}_4(1^\lambda, |\psi\rangle)\right]$$
$$= \mathbb{E}\left[\left\|D^{\mathsf{FSim}_1,\ldots,\mathsf{GSim}}\left(k, \Pi[w^*]\left|\widetilde{\psi}\right\rangle\right)\right\|^2 : |\psi\rangle, (\mathsf{FSim}_1, \ldots, \mathsf{FSim}_t, \mathsf{GSim}) \leftarrow \mathsf{QSObf}_5(1^\lambda, |\psi\rangle)\right],$$

where D's input includes the key k sampled by $\mathsf{QSObf}_4, \mathsf{QSObf}_5$.

Proof. In QSObf_4, we have that

$$\mathsf{F}_i[w^*](x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1}) \in \{H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \ell_1, \ldots, \ell_{i-1}, R[k, w^*](\widetilde{v}_1, \ldots, \widetilde{v}_i)), \bot\},$$

while in QSObf_5, we have that

$$\mathsf{FSim}_i[w^*](x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1}) \in \{H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \ell_1, \ldots, \ell_{i-1}, 0), \bot\}.$$

Both implementations of the oracles will always output $\bot$ on the same set of inputs, since this is true of $\mathsf{Dec}_{k,\cdot,\cdot}(\cdot)$ and $\mathsf{Ver}_{k,\cdot,\cdot}(\cdot)$ by definition. Finally, their non-$\bot$ answers are identically distributed over the randomness of the random oracle H, since each $(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1})$ fixes a single choice of bit $R[k, w^*](\widetilde{v}_1, \ldots, \widetilde{v}_i) \in \{0, 1\}$, and for any $(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1})$,

$$H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1}, 0) \quad \text{and} \quad H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1}, 1)$$

are identically distributed (each is a uniformly random string). □

Claim 5.25. *For any QPQ distinguisher D and any $w^* = \{w_i^*\}_{i\in[t]}$, it holds that*

$$\Big|\mathbb{E}\left[\left\|D^{\mathsf{F}_1[w^*],\ldots,\mathsf{GSim}}\left(k,\Pi[w^*]\left|\widetilde{\psi}\right\rangle\right)\right\|^2 : \left|\widetilde{\psi}\right\rangle,(\mathsf{F}_1[\cdot],\ldots,\mathsf{F}_t[\cdot],\mathsf{GSim})\leftarrow\mathsf{QSObf}_3(1^\lambda,|\psi\rangle)\right]$$
$$-\mathbb{E}\left[\left\|D^{\mathsf{F}_1[w^*],\ldots,\mathsf{GSim}}\left(k,\Pi[w^*]\left|\widetilde{\psi}\right\rangle\right)\right\|^2 : \left|\widetilde{\psi}\right\rangle,(\mathsf{F}_1[\cdot],\ldots,\mathsf{F}_t[\cdot],\mathsf{GSim})\leftarrow\mathsf{QSObf}_4(1^\lambda,|\psi\rangle)\right]\Big| = 2^{-\Omega(\kappa)},$$

where D's input includes the key k sampled by $\mathsf{QSObf}_3, \mathsf{QSObf}_4$.

Proof. Observe that the oracles $\mathsf{F}_i[w^*]$ in these experiments are identical except for on inputs

$$(x,\sigma_x,\widetilde{v}_1,\ldots,\widetilde{v}_i,\widetilde{w}_i,\ell_1,\ldots,\ell_{i-1})$$

such that there exists an $\iota\in[i-1]$ with

$$\ell_\iota = H(x,\sigma_x,\widetilde{v}_1,\ldots,\widetilde{v}_\iota,\ell_1,\ldots,\ell_{\iota-1},1-R[k,w^*](\widetilde{v}_1,\ldots,\widetilde{v}_\iota)).$$

However, the oracles $\mathsf{F}_i[w^*]$ in QSObf_4 are defined to never output such an ℓ_ι, and thus such an input can only be guessed with probability $2^{-\kappa}$ over the randomness of H. The claim follows by applying Lemma 2.8 (a standard oracle hybrid argument). □

Next, we combine what we have shown so far - the hardness of mapping between $\Pi[w^*]$ and $\Pi[\neg w^*]$ in QSObf_5 and the indistinguishability of QSObf_3 and QSObf_5 - to show the indistinguishability of QSObf_2 and QSObf_3 (in the case where the W wires are collapsed to some w^*).

Claim 5.26. *For any QPQ distinguisher D and any $w^* = \{w_i^*\}_{i\in[t]}$, it holds that*

$$\Big|\mathbb{E}\left[\left\|D^{\mathsf{F}_1,\ldots,\mathsf{GSim}}\left(k,\Pi[w^*]\left|\widetilde{\psi}\right\rangle\right)\right\|^2 : \left|\widetilde{\psi}\right\rangle,(\mathsf{F}_1,\ldots,\mathsf{F}_t,\mathsf{GSim})\leftarrow\mathsf{QSObf}_2(1^\lambda,|\psi\rangle)\right]$$
$$-\mathbb{E}\left[\left\|D^{\mathsf{F}_1[w^*],\ldots,\mathsf{GSim}}\left(k,\Pi[w^*]\left|\widetilde{\psi}\right\rangle\right)\right\|^2 : \left|\widetilde{\psi}\right\rangle,(\mathsf{F}_1[\cdot],\ldots,\mathsf{F}_t[\cdot],\mathsf{GSim})\leftarrow\mathsf{QSObf}_3(1^\lambda,|\psi\rangle)\right]\Big| = 2^{-\Omega(\kappa)},$$

where D's input includes the key k sampled by $\mathsf{QSObf}_2, \mathsf{QSObf}_3$.

Proof. Observe that the oracles $\mathsf{F}_i, \mathsf{F}_i[w^*]$ in these experiments are identical except for on inputs

$$(x,\sigma_x,\widetilde{v}_1,\ldots,\widetilde{v}_i,\widetilde{w}_i,\ell_1,\ldots,\ell_{i-1})$$

such that $\widetilde{w}_i \in P[\neg w_i^*]$. By combining the previous three claims, we see that no QPQ adversary can find such a $\widetilde{w}_i$ in QSObf_3 except with probability $2^{-\Omega(\kappa)}$. The claim follows by applying Lemma 2.8 (a standard oracle hybrid argument). □

Next, we state a direct corollary of these four claims, which is the hardness of mapping between $\Pi[w^*]$ and $\Pi[\neg w^*]$ even in QSObf_2.

Corollary 5.27. *For any QPQ unitary U, and any $w^* = \{w_i^*\}_{i\in[t]}$, it holds that*

$$\mathbb{E}\left[\left\|\Pi[\neg w_i^*]U^O\Pi[w^*]\left|\widetilde{\psi}\right\rangle\right\|^2 : \left|\widetilde{\psi}\right\rangle, O \leftarrow \mathsf{QSObf}_2(1^\lambda, |\psi\rangle)\right] = 2^{-\Omega(\kappa)}.$$

Finally, we consider a sequence of hybrids beginning with the real obfuscation experiment as described at the beginning of the proof, and ending with the simulated experiment using a simulator that we define below.

- The real experiment:

$$\mathcal{H}_0 = \mathbb{E}\left[\left\|D[A]^{\mathsf{F}_1,\ldots,\mathsf{G}}\left|\widetilde{\psi}\right\rangle\right\|^2 : \left|\widetilde{\psi}\right\rangle, (\mathsf{F}_1,\ldots,\mathsf{F}_t,\mathsf{G}) \leftarrow \mathsf{QSObf}_0(1^\lambda, |\psi\rangle)\right]$$

- Replace PRF with random oracle:

$$\mathcal{H}_1 = \mathbb{E}\left[\left\|D[A]^{\mathsf{F}_1,\ldots,\mathsf{G}}\left|\widetilde{\psi}\right\rangle\right\|^2 : \left|\widetilde{\psi}\right\rangle, (\mathsf{F}_1,\ldots,\mathsf{F}_t,\mathsf{G}) \leftarrow \mathsf{QSObf}_1(1^\lambda, |\psi\rangle)\right]$$

- Simulate the G oracle:

$$\mathcal{H}_2 = \mathbb{E}\left[\left\|D[A]^{\mathsf{F}_1,\ldots,\mathsf{GSim}}\left|\widetilde{\psi}\right\rangle\right\|^2 : \left|\widetilde{\psi}\right\rangle, (\mathsf{F}_1,\ldots,\mathsf{F}_t,\mathsf{GSim}) \leftarrow \mathsf{QSObf}_2(1^\lambda, |\psi\rangle)\right]$$

- Split $|\widetilde{\psi}\rangle$ into orthogonal components:

$$\mathcal{H}_3 = \mathbb{E}\left[\sum_{w^*}\left\|D[A]^{\mathsf{F}_1,\ldots,\mathsf{GSim}}\Pi[w^*]\left|\widetilde{\psi}\right\rangle\right\|^2 : \left|\widetilde{\psi}\right\rangle, (\mathsf{F}_1,\ldots,\mathsf{F}_t,\mathsf{GSim}) \leftarrow \mathsf{QSObf}_2(1^\lambda, |\psi\rangle)\right].$$

- Hard-code the w^* measurement results:

$$\mathcal{H}_4 = \mathbb{E}\left[\sum_{w^*}\left\|D[A]^{\mathsf{F}_1[w^*],\ldots,\mathsf{GSim}}\Pi[w^*]\left|\widetilde{\psi}\right\rangle\right\|^2 : \left|\widetilde{\psi}\right\rangle, (\mathsf{F}_1[\cdot],\ldots,\mathsf{F}_t[\cdot],\mathsf{GSim}) \leftarrow \mathsf{QSObf}_3(1^\lambda, |\psi\rangle)\right].$$

- Simulate the $\mathsf{F}_1,\ldots,\mathsf{F}_t$ oracles:

$$\mathcal{H}_5 = \mathbb{E}\left[\sum_{w^*}\left\|D[A]^{\mathsf{FSim}_1,\ldots,\mathsf{GSim}}\Pi[w^*]\left|\widetilde{\psi}\right\rangle\right\|^2 : \left|\widetilde{\psi}\right\rangle, (\mathsf{FSim}_1,\ldots,\mathsf{FSim}_t,\mathsf{GSim}) \leftarrow \mathsf{QSObf}_5(1^\lambda, |\psi\rangle)\right].$$

- Put $|\widetilde{\psi}\rangle$ back together:

$$\mathcal{H}_6 = \mathbb{E}\left[\left\|D[A]^{\mathsf{FSim}_1,\ldots,\mathsf{GSim}}\left|\widetilde{\psi}\right\rangle\right\|^2 : \left|\widetilde{\psi}\right\rangle, (\mathsf{FSim}_1,\ldots,\mathsf{FSim}_t,\mathsf{GSim}) \leftarrow \mathsf{QSObf}_5(1^\lambda, |\psi\rangle)\right].$$

- Simulate the state:

$$\mathcal{H}_7 = \mathbb{E}\left[\left\|D[A]^{\mathsf{FSim}_1,\ldots,\mathsf{GSim}}\left|\widetilde{\psi}\right\rangle\right\|^2 : \left|\widetilde{\psi}\right\rangle, (\mathsf{FSim}_1,\ldots,\mathsf{FSim}_t,\mathsf{GSim}) \leftarrow \mathsf{QSObf}_6(1^\lambda)\right].$$

Now, we are ready to define the simulator $\mathsf{Sim}^Q(1^\lambda, n, m, m')$ for our obfuscation scheme:

- Sample $\left|\widetilde{\psi}\right\rangle, (\mathsf{FSim}_1,\ldots,\mathsf{FSim}_t,\mathsf{GSim}) \leftarrow \mathsf{QSObf}_6(1^\lambda)$.
- Output $A^{\mathsf{FSim}_1,\ldots,\mathsf{FSim}_t,\mathsf{GSim}}\left|\widetilde{\psi}\right\rangle$.

Thus, the simulated experiment is exactly

$$\Pr\left[1 \leftarrow D\left(\mathsf{Sim}^Q\left(1^\lambda, n, m, m'\right)\right)\right] = \mathcal{H}_7.$$

The following set of claims then completes the proof.

Claim 5.28. $|\mathcal{H}_0 - \mathcal{H}_1| = \mathrm{negl}(\lambda)$.

Proof. This follows directly from the security of the PRF against quantum superposition-query attacks. □

Claim 5.29. $|\mathcal{H}_1 - \mathcal{H}_2| = \mathrm{negl}(\lambda)$.

Proof. Suppose that we sample $\left|\widetilde{\psi}\right\rangle, (\mathsf{F}_1,\ldots,\mathsf{F}_t,\mathsf{G}) \leftarrow \mathsf{QSObf}_1(1^\lambda, |\psi\rangle)$, and then define the set

$$B = \left\{(x, \sigma_x, \widetilde{v}_1,\ldots,\widetilde{v}_{t+1}, \ell_1,\ldots,\ell_t) : \mathsf{G}(x, \sigma_x, \widetilde{v}_1,\ldots,\widetilde{v}_{t+1}, \ell_1,\ldots,\ell_t) \notin \{Q(x), \perp\}\right\}.$$

Observe that the only difference between QSObf_1 and QSObf_2 is the definition of the oracles $\mathsf{G}, \mathsf{GSim}$, and that these oracles are identical outside of the set B. Suppose for contradiction that the claim is false. Then by Lemma 2.8 (which is a standard oracle hybrid argument), there must exist an adversary that can find an input on which G and GSim differ with non-negligible probability. That is, there exists a QPQ unitary U such that

$$\mathbb{E}\left[\left\|\Pi[B]U^{\mathsf{F}_1,\ldots,\mathsf{GSim}}\left|\widetilde{\psi}\right\rangle\right\|^2 : \left|\widetilde{\psi}\right\rangle, (\mathsf{F}_1,\ldots,\mathsf{F}_t,\mathsf{GSim}) \leftarrow \mathsf{QSObf}_2(1^\lambda, |\psi\rangle)\right] = \delta(\lambda)$$

for some $\delta(\lambda) = \mathsf{non\text{-}negl}(\lambda)$. Now, for each input x, define

$$B[x] \coloneqq \{(x, \cdot) : (x, \cdot) \in B\}.$$

Then by a union bound, there must exist *some* $x^* \in \{0,1\}^m$ such that

$$\mathbb{E}\left[\left\|\Pi\left[B[x^*]\right]U^{\mathsf{F}_1,\ldots,\mathsf{F}_t,\mathsf{GSim}}\left|\widetilde{\psi}\right\rangle\right\|^2 : \left|\widetilde{\psi}\right\rangle, (\mathsf{F}_1,\ldots,\mathsf{F}_t,\mathsf{GSim}) \leftarrow \mathsf{QSObf}_2(1^\lambda, |\psi\rangle)\right] \geq \delta(\lambda)\cdot 2^{-m}.$$

Next, define $\mathsf{Coh\text{-}LMEval}[x^*]$ to be the unitary that coherently applies the evaluation procedure $\mathsf{LMEval}(x^*, \cdot, \{L_i\}_{i\in[t+1]}, \{\theta_i\}_{i\in[t+1]}, \{f_i\}_{i\in[t]}, g)$ for the LM quantum program that we are obfuscating, and define

$$|\psi'_{x^*}\rangle := \mathsf{Coh\text{-}LMEval}[x^*]^\dagger |Q(x^*)\rangle\langle Q(x^*)| \mathsf{Coh\text{-}LMEval}[x^*] |\psi\rangle, \quad |\psi_{x^*}\rangle := \frac{|\psi'_{x^*}\rangle}{\| |\psi'_{x^*}\rangle \|}.$$

That is, $|\psi_{x^*}\rangle$ is the result of running the LM quantum program coherently on input x^* and post-selecting on obtaining the "correct" output $Q(x^*)$. By the ϵ-pseudo-determinism of Q and Gentle Measurement (Lemma 2.3), there is some $\delta'(\lambda) = \mathsf{non\text{-}negl}(\lambda)$ such that

$$\begin{aligned}&\mathbb{E}\left[\left\|\Pi[B[x^*]]U^{\mathsf{F}_1,\ldots,\mathsf{GSim}} |\widetilde{\psi}_{x^*}\rangle\right\|^2 : |\widetilde{\psi}_{x^*}\rangle, (\mathsf{F}_1,\ldots,\mathsf{F}_t,\mathsf{GSim}) \leftarrow \mathsf{QSObf}_2(1^\lambda, |\psi_{x^*}\rangle)\right]\\ &\geq \delta'(\lambda)\cdot 2^{-m} \geq \delta'(\lambda)\cdot 2^{-n}.\end{aligned}$$

However, this violates Lemma 5.36 with $\tau = 0$, which is proven in the following section. Indeed, plugging in $\kappa = n^4$, the lemma states that, for some constant c,

$$\begin{aligned}&\mathbb{E}\left[\left\|\Pi[B[x^*]]U^{\mathsf{F}_1,\ldots,\mathsf{GSim}} |\widetilde{\psi}_{x^*}\rangle\right\|^2 : |\widetilde{\psi}_{x^*}\rangle, (\mathsf{F}_1,\ldots,\mathsf{F}_t,\mathsf{GSim}) \leftarrow \mathsf{QSObf}_2(1^\lambda, |\psi_{x^*}\rangle)\right]\\ &= 2^{3n(t+1)-cn^4} = 2^{-\Omega(n^2)} < \delta'(\lambda)\cdot 2^{-n},\end{aligned}$$

which gives us the contradiction. □

Claim 5.30. $|\mathcal{H}_2 - \mathcal{H}_3| \leq 2^{-\Omega(\kappa)}$.

Proof.

$$\begin{aligned}&|\mathcal{H}_2 - \mathcal{H}_3|\\ &\leq 2^n \cdot \left[\sum_{w^*\neq w'^*} \mathbb{E}\left[\left\|\Pi[w'^*]D_1^{\mathsf{F}_1,\ldots,\mathsf{GSim}}\Pi[w^*] |\widetilde{\psi}\rangle\right\|^2 : |\widetilde{\psi}\rangle, (\mathsf{F}_1,\ldots,\mathsf{F}_t,\mathsf{GSim}) \leftarrow \mathsf{QSObf}_2(1^\lambda, |\psi\rangle)\right]\right]^{1/2}\\ &\leq 2^n \cdot \left[2^n \cdot \sum_{w^*} \mathbb{E}\left[\left\|\Pi[\neg w^*]D_1^{\mathsf{F}_1,\ldots,\mathsf{GSim}}\Pi[w^*] |\widetilde{\psi}\rangle\right\|^2 : |\widetilde{\psi}\rangle, (\mathsf{F}_1,\ldots,\mathsf{F}_t,\mathsf{GSim}) \leftarrow \mathsf{QSObf}_2(1^\lambda, |\psi\rangle)\right]\right]^{1/2}\\ &\leq 2^{2n}\cdot 2^{-\Omega(\kappa)} = 2^{-\Omega(\kappa)},\end{aligned}$$

where the first inequality follows from Lemma 2.6 (which is an application of Cauchy-Schwarz) and the third follows from Corollary 5.27 proven above.

□

Claim 5.31. $|\mathcal{H}_3 - \mathcal{H}_4| = 2^{-\Omega(\kappa)}$.

Proof. This follows from Corollary 5.27 proven above and Lemma 2.8 (which is a standard oracle hybrid argument). □

Claim 5.32. $|\mathcal{H}_4 - \mathcal{H}_5| = 2^{-\Omega(\kappa)}$.

Proof. This follows by combining Claim 5.25 and Claim 5.24 proven above. □

Claim 5.33. $|\mathcal{H}_5 - \mathcal{H}_6| \leq 2^{-\Omega(\kappa)}$.

Proof.

$$
\begin{aligned}
&|\mathcal{H}_5 - \mathcal{H}_6| \\
&\leq 2^n \cdot \left[\sum_{w^* \neq w'^*} \mathbb{E}\left[\left\| \Pi[w'^*] D_1^{\mathsf{FSim}_1,\ldots,\mathsf{GSim}} \Pi[w^*] \left|\widetilde{\psi}\right\rangle \right\|^2 : \left|\widetilde{\psi}\right\rangle, \mathsf{F}_1, \ldots, \mathsf{F}_t, \mathsf{GSim} \leftarrow \mathsf{QSObf}_5(1^\lambda, |\psi\rangle) \right] \right]^{1/2} \\
&\leq 2^n \cdot \left[2^n \cdot \sum_{w^*} \mathbb{E}\left[\left\| \Pi[\neg w^*] D_1^{\mathsf{FSim}_1,\ldots,\mathsf{GSim}} \Pi[w^*] \left|\widetilde{\psi}\right\rangle \right\|^2 : \left|\widetilde{\psi}\right\rangle, \mathsf{F}_1, \ldots, \mathsf{F}_t, \mathsf{GSim} \leftarrow \mathsf{QSObf}_5(1^\lambda, |\psi\rangle) \right] \right]^{1/2} \\
&\leq 2^{2n} \cdot 2^{-\Omega(\kappa)} = 2^{-\Omega(\kappa)},
\end{aligned}
$$

where the first inequality follows from Lemma 2.6 (which is an application of Cauchy-Schwarz) and the third follows from Claim 5.23 proven above.

□

Claim 5.34. $|\mathcal{H}_6 - \mathcal{H}_7| \leq 2^{-\Omega(\kappa)}$.

Proof. This follows from privacy of the authentication scheme (Theorem 5.12), since the oracles in $\mathcal{H}_6$ can be implemented with oracle access to $\mathsf{Ver}_{k,\cdot,\cdot}(\cdot)$ rather than $\mathsf{Dec}_{k,\cdot,\cdot}(\cdot)$. □

□

5.4.5 Inductive argument

In this section, we give an inductive proof of Lemma 5.36, which was required by Claim 5.29 above. First, we describe a variant of QSObf that we call ParMeas, which supports hard-coding an input x^* and the first τ partial measurement results $\{v_\iota^*, r_\iota^*\}_{\iota \in [\tau]}$ into the oracles $\mathsf{F}_1, \ldots, \mathsf{F}_t$. When these results are hard-coded, the inputs $\widetilde{v}_1, \ldots, \widetilde{v}_\tau, \ell_1, \ldots, \ell_\tau$ are merely verified rather than decoded by the oracles $\mathsf{F}_1, \ldots, \mathsf{F}_t$, and the hard-coded results $\{v_\iota^*, r_\iota^*\}_{\iota \in [\tau]}$ are used in place of the decoded results.

We define the distribution to include two additional outputs:

- The set $B[x^*, \{v_\iota^*, r_\iota^*\}_{\iota \in [\tau]}]$ contains the "bad" set of inputs that verify properly but decode to an incorrect output $Q(x^*)$, when using the hard-coded values $\{v_\iota^*, r_\iota^*\}_{\iota \in [\tau]}$.
- The set $C[x^*, \{v_\iota^*, r_\iota^*\}_{\iota \in [\tau]}]$ contains the set of inputs on which the oracles would differ had the latest measurement (v_τ^*, r_τ^*) not been hard-coded.

$\mathsf{ParMeas}(1^\lambda, |\psi\rangle)$:

- Sample $k \leftarrow \mathsf{Gen}(1^\kappa, n)$, and compute $|\psi_k\rangle = \mathsf{Enc}_k(|\psi\rangle)$.
- Sample a signature token $(\mathsf{vk}, |\mathsf{sk}\rangle) \leftarrow \mathsf{TokGen}(1^\kappa)$.
- Let $H : \{0,1\}^* \to \{0,1\}^\kappa$ be a random oracle.
- For each $i \in [t]$, define the function $\mathsf{F}_i[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}](x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1})$:
  - If $x \neq x^*$, output $\mathsf{F}_i\,(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1})$, where F_i is defined as in QSObf.
  - Output $\bot$ if $\mathsf{TokVer}(\mathsf{vk}, x, \sigma_x) = \bot$.
  - For each $\iota \in [\min\{\tau, i-1\}]$, let

    $$\ell_{\iota,0} = H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 0), \;\; \ell_{\iota,1} = H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 1),$$

    and output $\bot$ if $\ell_{\iota,0} = \ell_{\iota,1}$ or $\ell_\iota \notin \{\ell_{\iota,0}, \ell_{\iota,1}\}$.
  - If $i \leq \tau$:
    - Output $\bot$ if $\mathsf{Ver}_{k,L_i\ldots L_1,\theta_i}(\widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i) = \bot$.
    - Set $\ell_i \coloneqq H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \ell_1, \ldots, \ell_{i-1}, r_i^*)$, and output $(\ell_i, \widetilde{v}_i)$.
  - If $i > \tau$:
    - For each $\iota \in [\tau+1, i-1]$, let

      $$\ell_{\iota,0} = H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 0), \;\; \ell_{\iota,1} = H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 1),$$

      and output $\bot$ if $\ell_{\iota,0} = \ell_{\iota,1}$ or $\ell_\iota \notin \{\ell_{\iota,0}, \ell_{\iota,1}\}$. Otherwise, let r_ι be such that $\ell_\iota = \ell_{\iota,r_\iota}$.
    - Output $\bot$ if $\mathsf{Ver}_{k,L_i\ldots L_1,\theta_i[V_1,\ldots,V_\tau]}(\widetilde{v}_1, \ldots, \widetilde{v}_\tau) = \bot$.
    - Compute $(v_{\tau+1}, \ldots, v_i, w_i) = \mathsf{Dec}_{k,L_i\ldots L_1,\theta_i[V_{\tau+1},\ldots,V_i,W_i]}(\widetilde{v}_{\tau+1}, \ldots, \widetilde{v}_i, \widetilde{w}_i)$, and output $\bot$ if the result is $\bot$.
    - Compute $(\cdot, r_i) = f_i^{x, r_1^*, \ldots, r_\tau^*, r_{\tau+1}, \ldots, r_{i-1}}(v_1^*, \ldots, v_\tau^*, v_{\tau+1}, \ldots, v_i, w_i)$.
    - Set $\ell_i \coloneqq H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \ell_1, \ldots, \ell_{i-1}, r_i)$, and output $(\ell_i, \widetilde{v}_i)$.
- Define GSim as in QSObf_2.
- Let $B[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]$ be the set of $(\sigma_{x^*}, \widetilde{v}_1, \ldots, \widetilde{v}_{t+1}, \ell_1, \ldots, \ell_t)$ such that the output of the following procedure is $\notin \{Q(x), \bot\}$:
  - Output $\bot$ if $\mathsf{TokVer}(\mathsf{vk}, x, \sigma_x) = \bot$.
  - For each $\iota \in [t]$, let

    $$\ell_{\iota,0} = H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 0), \;\; \ell_{\iota,1} = H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 1),$$

    and output $\bot$ if $\ell_{\iota,0} = \ell_{\iota,1}$ or $\ell_\iota \notin \{\ell_{\iota,0}, \ell_{\iota,1}\}$. If $\iota > \tau$, let r_ι be such that $\ell_\iota = \ell_{\iota,r_\iota}$.
  - Output $\bot$ if $\mathsf{Ver}_{k,L_{t+1}\ldots L_1,\theta_{t+1}[V_1,\ldots,V_\tau]}(\widetilde{v}_1, \ldots, \widetilde{v}_\tau) = \bot$.
  - Compute $(v_{\tau+1}, \ldots, v_{t+1}) = \mathsf{Dec}_{k,L_{t+1}\ldots L_1,\theta_{t+1}[V_{\tau+1},\ldots,V_{t+1}]}(\widetilde{v}_{\tau+1}, \ldots, \widetilde{v}_{t+1})$, and output $\bot$ if the result is $\bot$.
  - Output $g^{x, r_1^*, \ldots, r_\tau^*, r_{\tau+1}, \ldots, r_t}(v_1^*, \ldots, v_\tau^*, v_{\tau+1}, \ldots, v_{t+1})$.

- Let $C[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]$ be the set that includes, for any $i \in [\tau]$, all $(\sigma_{x^*}, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1})$ such that

$$\begin{aligned}&\mathsf{F}_i[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau-1]}](x^*, \sigma_{x^*}, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1})\\ &\neq \mathsf{F}_i[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}](x^*, \sigma_{x^*}, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1}),\end{aligned}$$

  and all $(\sigma_{x^*}, \widetilde{v}_1, \ldots, \widetilde{v}_{t+1}, \ell_1, \ldots, \ell_t)$ such that

$$(\sigma_{x^*}, \widetilde{v}_1, \ldots, \widetilde{v}_{t+1}, \ell_1, \ldots, \ell_t) \in B[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau-1]}] \setminus B[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}].$$

- Output $|\widetilde{\psi}\rangle = |\psi_k\rangle\,|\mathsf{sk}\rangle\,, O = (\mathsf{F}_1[\cdot], \ldots, \mathsf{F}_t[\cdot], \mathsf{GSim})\,, B[\cdot], C[\cdot]$.

We now make a few remarks.

- Throughout the remainder of the proof, we will be working with *fully-deterministic* LM quantum programs for a given input x^*, i.e.

$$\Pr\left[\mathsf{LMEval}(x^*, |\psi\rangle\,, \{L_i\}_{i\in[t+1]}, \{\theta_i\}_{i\in[t+1]}, \{f_i\}_{i\in[t+1]}, g) \to Q(x^*)\right] = 1.$$

  Indeed, recall that we performed a post-selection on the correct outcome $Q(x^*)$ during the proof of Claim 5.29 above.

- Whenever we reference an input x^* and partial measurement results $\{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}$, we always mean measurement results that occur with *non-zero* probability, i.e. they are in the support of the partial evaluation of $|\psi\rangle$ on input x^*.

- For $\tau = 0$ (i.e. no partial measurements), the above distribution $\mathsf{ParMeas}$ is identical to QSObf_2 augmented with the set $B[x^*]$ as defined in the proof of Claim 5.29 (and $C[x^*]$ is undefined in this case since it requires $\tau \geq 1$).

- For any input x^* and full set of measurement results $\{v_\iota^*, r_\iota^*\}_{\iota\in[t+1]}$, the set $B[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[t+1]}]$ is empty, by virtue of the fact that these measurement results occur with non-zero probability, and the program outputs $Q(x^*)$ with probability 1.

- In the remainder of the proof, we will make use of the notation $|\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]\rangle$ as defined in Section 5.4.3.

Before proving the main inductive lemma of this section, we show the following statement, which essentially says that it is hard to find an element of

$$B[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau-1]}] \setminus B[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]$$

given the authenticated version of $|\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]\rangle$ and the oracles with x^* and $\{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}$ hard-coded. The meat of this proof is actually deferred to the following section, in which we prove the "hardness of mapping" lemma, Lemma 5.44.

Lemma 5.35. *For any input x^*, $\tau \in [1, \ldots, t+1]$, measurement results $\{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}$, and QPQ unitary U, it holds that*

$$\Big| \mathbb{E}\left[\left\| \Pi \left[B\left[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau-1]}\right]\right] U^{\mathsf{F}_1[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau-1]}],\ldots,\mathsf{F}_t[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau-1]}],\mathsf{GSim}} |\widetilde{\psi}\rangle \right\|^2 \right]$$
$$- \mathbb{E}\left[\left\| \Pi \left[B\left[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}\right]\right] U^{\mathsf{F}_1[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]}],\ldots,\mathsf{F}_t[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]}],\mathsf{GSim}} |\widetilde{\psi}\rangle \right\|^2 \right] \Big| \leq 2^{-\Omega(\kappa)},$$

where both expectations are over

$$|\widetilde{\psi}\rangle, (\mathsf{F}_1[\cdot], \mathsf{F}_t[\cdot], \mathsf{GSim}), B[\cdot], C[\cdot] \leftarrow \mathsf{ParMeas}(1^\lambda, |\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]\rangle).$$

Proof. Note that the set $C[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]$ includes all elements of

$$B[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau-1]}] \setminus B[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]$$

and all inputs on which $\mathsf{F}_i[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau-1]}]$ and $\mathsf{F}_i[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]$ differ for any $i \in [t]$. Thus, by Lemma 2.8 (a standard oracle hybrid argument) it suffices to show that

$$\mathbb{E}\left[\left\| \Pi \left[C\left[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}\right]\right] U^{\mathsf{F}_1[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]}],\ldots,\mathsf{F}_t[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]}],\mathsf{GSim}} |\widetilde{\psi}\rangle \right\|^2 \right] \leq 2^{-\Omega(\kappa)},$$

where the expectation is over

$$|\widetilde{\psi}\rangle, (\mathsf{F}_1[\cdot], \ldots, \mathsf{F}_t[\cdot], \mathsf{GSim}), B[\cdot], C[\cdot] \leftarrow \mathsf{ParMeas}(1^\lambda, |\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]\rangle).$$

Now we consider all possible elements of the set $C[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]$. By inspecting the definition, we see that any element of $C[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]$ must fall into one of the following categories:

- An input to $\mathsf{F}_\tau[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]$ that contains a sub-string $(\widetilde{v}_\tau, \widetilde{w}_\tau)$ such that

  $$f_\tau^{x^*, r_1^*, \ldots, r_{\tau-1}^*}(v_1^*, \ldots, v_{\tau-1}^*, \mathsf{Dec}_{k,L_\tau\ldots L_1,\theta_\tau[V_\tau,W_\tau]}(\widetilde{v}_\tau, \widetilde{w}_\tau)) = (\cdot, 1 - r_\tau^*),$$

  where $\mathsf{Dec}_{k,L_\tau\ldots L_1,\theta_\tau[V_\tau,W_\tau]}(\widetilde{v}_\tau, \widetilde{w}_\tau) = (v_\tau, w_\tau) \neq \bot$.

- An input to $\mathsf{F}_i[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]$ for $i > \tau$ or an element of

  $$B[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau-1]}] \setminus B[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]$$

  that contains either:

  - $\widetilde{v}_\tau$ such that $\mathsf{Dec}_{k,L_\tau\ldots L_1,\theta_\tau[V_\tau]}(\widetilde{v}_\tau) \notin \{v_\tau^*, \bot\}$, or
  - ℓ_τ such that $\ell_\tau = H(\ldots, 1 - r_\tau^*)$.

First, notice that by definition, the oracles $\mathsf{F}_1[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}], \ldots, \mathsf{F}_t[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}], \mathsf{GSim}$ never output a label $H(\ldots, 1 - r_\tau^*)$. Thus, U can only successfully guess an ℓ_τ such that $\ell_\tau = H(\ldots, 1 - r_\tau^*)$ with probability $2^{-\kappa}$ over the randomness of the random oracle.

Now, define $\Pi[\neg r_\tau^*]$ to be the projection onto strings $(\widetilde{v}_\tau, \widetilde{w}_\tau)$ such that

$$f_\tau^{x^*, r_1^*, \ldots, r_{\tau-1}^*}(v_1^*, \ldots, v_{\tau-1}^*, \mathsf{Dec}_{k, L_\tau \ldots L_1, \theta_\tau[V_\tau, W_\tau]}(\widetilde{v}_\tau, \widetilde{w}_\tau)) = (\cdot, 1 - r_\tau^*),$$

where $\mathsf{Dec}_{k, L_\tau \ldots L_1, \theta_\tau[V_\tau, W_\tau]}(\widetilde{v}_\tau, \widetilde{w}_\tau) = (v_\tau, w_\tau) \neq \bot$, and define $\Pi[\neg v_\tau^*]$ to be the projection onto strings $\widetilde{v}_\tau$ such that

$$\mathsf{Dec}_{k, L_\tau \ldots L_1, \theta_\tau[V_\tau]}(\widetilde{v}_\tau) \notin \{v_\tau^*, \bot\}.$$

Then, define

$$\Pi[\neg(v_\tau^*, r_\tau^*)] \coloneqq \Pi[\neg r_\tau^*] + \Pi[\neg v_\tau^*].$$

Finally, given the sampled signature token verification key vk, define the projector onto valid signatures of x^*:

$$\Pi[x^*, \mathsf{vk}] \coloneqq \sum_{\sigma : \mathsf{TokVer}(\mathsf{vk}, x^*, \sigma) = \top} |\sigma\rangle\langle\sigma| \,,$$

and note that any element of $C\left[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}\right]$ must include a valid signature of x^*.

Now, by the preceding observations, we have that

$$\begin{aligned}
&\mathbb{E}\left[\left\|\Pi\left[C\left[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}\right]\right] U^{\mathsf{F}_1[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}], \ldots, \mathsf{F}_t[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}], \mathsf{GSim}} |\widetilde{\psi}\rangle\right\|^2\right] \\
&\leq \mathbb{E}\left[\left\| \left(\Pi[x^*, \mathsf{vk}] \otimes \Pi[\neg(v_\tau^*, r_\tau^*)]\right) U^{\mathsf{F}_1[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}], \ldots, \mathsf{F}_t[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}], \mathsf{GSim}} |\widetilde{\psi}\rangle\right\|^2\right] + 2^{-\Omega(\kappa)} \\
&\leq 2^{-\Omega(\kappa)},
\end{aligned}$$

where the first inequality is due to the observation that ℓ_τ such that $\ell_\tau = H(\ldots, 1 - r_\tau^*)$ can only be guessed with probability $2^{-\kappa}$, and the second inequality is Lemma 5.44 proven in the next section. This completes the proof. □

Now, we show the main lemma of the section.

Lemma 5.36. *There exist a constant $c > 0$ such that for any input x^*, $\tau \in [0, \ldots, t+1]$, measurement results $\{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}$, and QPQ unitary U, it holds that*

$$\mathbb{E}\left[\left\|\Pi\left[B\left[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}\right]\right] U^{\mathsf{F}_1[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}], \ldots, \mathsf{F}_t[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}], \mathsf{GSim}} |\widetilde{\psi}\rangle\right\|^2\right] \leq 2^{3n(t+1-\tau) - c\kappa},$$

where the expectation is over

$$|\widetilde{\psi}\rangle, (\mathsf{F}_1[\cdot], \ldots, \mathsf{F}_t[\cdot], \mathsf{GSim}), B[\cdot], C[\cdot] \leftarrow \mathsf{ParMeas}(1^\lambda, |\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]\rangle).$$

Proof. We will show this by induction on τ, starting with $\tau = t+1$ and ending with $\tau = 0$. The base case ($\tau = t+1$) is trivial because the set $B[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[t+1]}]$ is empty, as noted above.

Now, we will let c be a constant such that $2^{-c\kappa}$ is an upper bound on the expression in the statement of Lemma 5.35. For the inductive step, suppose that Lemma 5.36 holds for some $\tau \in [t+1]$. Consider any x^* and measurement results $\{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}$, and define the following two distributions over $|\widetilde{\psi}\rangle, (\mathsf{F}_1[\cdot], \ldots, \mathsf{F}_t[\cdot], \mathsf{GSim}), B[\cdot]$ (dropping the set $C[\cdot]$ since we don't need it for this proof):

- $\mathcal{D}$: $\mathsf{ParMeas}(1^\lambda, |\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau-1]}]\rangle)$.
- $\mathcal{D}[v_\tau^*, r_\tau^*]$: $\mathsf{ParMeas}(1^\lambda, |\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]\rangle)$.

To show that Lemma 5.36 holds for $\tau - 1$, we have that

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{|\widetilde{\psi}\rangle,(\mathsf{F}_1[\cdot],\ldots,\mathsf{F}_t[\cdot],\mathsf{GSim}),B[\cdot]\leftarrow\mathcal{D}} \left[\left\|\Pi\left[B\left[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau-1]}\right]\right] U^{\mathsf{F}_1[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau-1]}],\ldots,\mathsf{GSim}} |\widetilde{\psi}\rangle\right\|^2\right] \\
&\leq 2^n \cdot \sum_{v_\tau^*, r_\tau^*} \mathop{\mathbb{E}}_{\substack{|\widetilde{\psi}\rangle,(\mathsf{F}_1[\cdot],\ldots,\mathsf{F}_t[\cdot],\mathsf{GSim}),\\ B[\cdot]\leftarrow\mathcal{D}[v_\tau^*,r_\tau^*]}} \left[\left\|\Pi\left[B\left[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau-1]}\right]\right] U^{\mathsf{F}_1[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau-1]}],\ldots,\mathsf{GSim}} |\widetilde{\psi}\rangle\right\|^2\right] \\
&\leq 2^n \cdot \sum_{v_\tau^*, r_\tau^*} \mathop{\mathbb{E}}_{\substack{|\widetilde{\psi}\rangle,(\mathsf{F}_1[\cdot],\ldots,\mathsf{F}_t[\cdot],\mathsf{GSim}),\\ B[\cdot]\leftarrow\mathcal{D}[v_\tau^*,r_\tau^*]}} \left[\left\|\Pi\left[B\left[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}\right]\right] U^{\mathsf{F}_1[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]}],\ldots,\mathsf{GSim}} |\widetilde{\psi}\rangle\right\|^2\right] + 2^{2n-c\kappa} \\
&\leq 2^{2n+3n(t+1-\tau)-c\kappa} + 2^{2n-c\kappa} \\
&\leq 2^{3n(t+1-(\tau-1))-c\kappa-n} + 2^{2n-c\kappa} \\
&\leq 2^{3n(t+1-(\tau-1))-c\kappa},
\end{aligned}
$$

where

- The first inequality follows from Lemma 2.7 (an application of Cauchy-Schwarz).
- The second inequality follows from Lemma 5.35 proven above.
- The third inequality follows from Lemma 5.36 for τ (the induction hypothesis).
- The final inequality follows because the two summands can each be bounded by the quantity $2^{3n(t+1-(\tau-1))-c\kappa-1}$.

□

5.4.6 Hardness of mapping

Our final step is to prove the "hardness of mapping" lemma, Lemma 5.44, that was required for Lemma 5.35 above. But first, we will need to introduce some "partial simulation" hybrid distributions ParSim_i. We let $\mathsf{ParSim}_0 = \mathsf{ParMeas}$ as defined in the preceding section. We will change the distribution gradually until it corresponds to a simulated distribution, equivalent to QSObf_5 from Section 5.4.4.

$\mathsf{ParSim}_1(1^\lambda, |\psi\rangle)$:

- Sample $k \leftarrow \mathsf{Gen}(1^\kappa, n)$, and compute $|\psi_k\rangle = \mathsf{Enc}_k(|\psi\rangle)$.
- Sample a signature token $(\mathsf{vk}, |\mathsf{sk}\rangle) \leftarrow \mathsf{TokGen}(1^\kappa)$.
- Let $H : \{0,1\}^* \to \{0,1\}^\kappa$ be a random oracle.
- For each $i \in [t]$, define $\mathsf{F}_i[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}, \{w_\iota^*\}_{\iota\in[\tau+1,t]}](x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1})$:
  - If $x \neq x^*$, output $\mathsf{F}_i\,(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1})$, where F_i is defined as in QSObf.
  - Output $\bot$ if $\mathsf{TokVer}(\mathsf{vk}, x, \sigma_x) = \bot$.
  - For each $\iota \in [\min\{\tau, i-1\}]$, let

    $$\ell_{\iota,0} = H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 0), \;\; \ell_{\iota,1} = H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 1),$$

    and output $\bot$ if $\ell_{\iota,0} = \ell_{\iota,1}$ or $\ell_\iota \notin \{\ell_{\iota,0}, \ell_{\iota,1}\}$.
  - If $i \leq \tau$:
    - Output $\bot$ if $\mathsf{Ver}_{k,L_i\ldots L_1,\theta_i}(\widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i) = \bot$.
    - Set $\ell_i := H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \ell_1, \ldots, \ell_{i-1}, r_i^*)$, and output $(\ell_i, \widetilde{v}_i)$.
  - If $i > \tau$:
    - For each $\iota \in [\tau+1, i-1]$, let

      $$\ell_{\iota,0} = H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 0), \;\; \ell_{\iota,1} = H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 1),$$

      and output $\bot$ if $\ell_{\iota,0} = \ell_{\iota,1}$ or $\ell_\iota \notin \{\ell_{\iota,0}, \ell_{\iota,1}\}$. Otherwise, let r_ι be such that $\ell_\iota = \ell_{\iota,r_\iota}$.
    - Output $\bot$ if $\mathsf{Ver}_{k,L_i\ldots L_1,\theta_i[V_1,\ldots,V_\tau,W_i]}(\widetilde{v}_1, \ldots, \widetilde{v}_\tau, \widetilde{w}_i) = \bot$.
    - Compute $(v_{\tau+1}, \ldots, v_i) = \mathsf{Dec}_{k,L_i\ldots L_1,\theta_i[V_{\tau+1},\ldots,V_i]}(\widetilde{v}_{\tau+1}, \ldots, \widetilde{v}_i)$, and output $\bot$ if the result is $\bot$.
    - Compute $(\cdot, r_i) = f_i^{x,r_1^*,\ldots,r_\tau^*,r_{\tau+1},\ldots,r_{i-1}}(v_1^*, \ldots, v_\tau^*, v_{\tau+1}, \ldots, v_i, w_i^*)$.
    - Set $\ell_i := H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \ell_1, \ldots, \ell_{i-1}, r_i)$, and output $(\ell_i, \widetilde{v}_i)$.
- Define GSim as in QSObf_2.
- Output $|\widetilde{\psi}\rangle = |\psi_k\rangle\,|\mathsf{sk}\rangle\,, O = (\mathsf{F}_1[\cdot], \ldots, \mathsf{F}_t[\cdot], \mathsf{GSim})\,.$

$\mathsf{ParSim}_2(1^\lambda, |\psi\rangle)$:

- Sample $k \leftarrow \mathsf{Gen}(1^\kappa, n)$, and compute $|\psi_k\rangle = \mathsf{Enc}_k(|\psi\rangle)$.
- Sample a signature token $(\mathsf{vk}, |\mathsf{sk}\rangle) \leftarrow \mathsf{TokGen}(1^\kappa)$.
- Let $H : \{0,1\}^* \to \{0,1\}^\kappa$ be a random oracle.

- For each $i \in [t]$, define $\mathsf{F}_i[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}, \{w_\iota^*\}_{\iota\in[\tau+1,t]}](x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1})$:
  - If $x \neq x^*$, output $\mathsf{F}_i\,(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1})$, where F_i is defined as in QSObf.
  - Output $\bot$ if $\mathsf{TokVer}(\mathsf{vk}, x, \sigma_x) = \bot$.
  - For each $\iota \in [i-1]$, let

    $$\ell_{\iota,0} = H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 0),\ \ \ell_{\iota,1} = H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 1),$$

    and output $\bot$ if $\ell_{\iota,0} = \ell_{\iota,1}$ or $\ell_\iota \notin \{\ell_{\iota,0}, \ell_{\iota,1}\}$.
  - If $i \leq \tau$:
    - Output $\bot$ if $\mathsf{Ver}_{k,L_i\ldots L_1,\theta_i}(\widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i) = \bot$.
    - Set $\ell_i := H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \ell_1, \ldots, \ell_{i-1}, r_i^*)$, and output $(\ell_i, \widetilde{v}_i)$.
  - If $i > \tau$:
    - Output $\bot$ if $\mathsf{Ver}_{k,L_i\ldots L_1,\theta_i[V_1,\ldots,V_\tau,W_i]}(\widetilde{v}_1, \ldots, \widetilde{v}_\tau, \widetilde{w}_i) = \bot$.
    - Compute $(v_{\tau+1}, \ldots, v_i) = \mathsf{Dec}_{k,L_i\ldots L_1,\theta_i[V_{\tau+1},\ldots,V_i]}(\widetilde{v}_{\tau+1}, \ldots, \widetilde{v}_i)$, and output $\bot$ if the result is $\bot$.
    - For $\iota \in [\tau+1, i]$, compute $(\cdot, r_\iota) = f_\iota^{x, r_1^*, \ldots, r_\tau^*, r_{\tau+1}, \ldots, r_{\iota-1}}(v_1^*, \ldots, v_\tau^*, v_{\tau+1}, \ldots, v_\iota, w_\iota^*)$.
    - Set $\ell_i := H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \ell_1, \ldots, \ell_{i-1}, r_i)$, and output $(\ell_i, \widetilde{v}_i)$.
- Define GSim as in QSObf_2.
- Output $|\widetilde{\psi}\rangle = |\psi_k\rangle\,|\mathsf{sk}\rangle\,, O = (\mathsf{F}_1[\cdot], \ldots, \mathsf{F}_t[\cdot], \mathsf{GSim})\,.$

$\mathsf{ParSim}_3(1^\lambda, |\psi\rangle)$:

- Sample $k \leftarrow \mathsf{Gen}(1^\kappa, n)$, and compute $|\psi_k\rangle = \mathsf{Enc}_k(|\psi\rangle)$.
- Sample a signature token $(\mathsf{vk}, |\mathsf{sk}\rangle) \leftarrow \mathsf{TokGen}(1^\kappa)$.
- Let $H : \{0,1\}^* \to \{0,1\}^\kappa$ be a random oracle.
- For each $i \in [t]$, define the function $\mathsf{FSim}_i[x^*](x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1})$:
  - If $x \neq x^*$, output $\mathsf{F}_i\,(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1})$, where F_i is defined as in QSObf.
  - Output $\bot$ if $\mathsf{TokVer}(\mathsf{vk}, x, \sigma_x) = \bot$.
  - For each $\iota \in [i-1]$, let

    $$\ell_{\iota,0} = H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 0),\ \ \ell_{\iota,1} = H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 1),$$

    and output $\bot$ if $\ell_{\iota,0} = \ell_{\iota,1}$ or $\ell_\iota \notin \{\ell_{\iota,0}, \ell_{\iota,1}\}$.
  - Output $\bot$ if $\mathsf{Ver}_{k,L_i\ldots L_1,\theta_i}(\widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i) = \bot$.
  - Set $\ell_i := H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \ell_1, \ldots, \ell_{i-1}, 0)$, and output $(\widetilde{v}_i, \ell_i)$.
- Define GSim as in QSObf_2.
- Output $|\widetilde{\psi}\rangle = |\psi_k\rangle\,|\mathsf{sk}\rangle\,, O = (\mathsf{FSim}_1[\cdot], \ldots, \mathsf{FSim}_t[\cdot], \mathsf{GSim})\,.$

$\mathsf{ParSim}_4(1^\lambda, |\psi\rangle)$:

- Sample $k \leftarrow \mathsf{Gen}(1^\kappa, n)$, and compute $|\psi_k\rangle = \mathsf{Enc}_k(|\psi\rangle)$.

- Sample a signature token $(\mathsf{vk}, |\mathsf{sk}\rangle) \leftarrow \mathsf{TokGen}(1^\kappa)$.
- Let $H : \{0,1\}^* \to \{0,1\}^\kappa$ be a random oracle.
- For each $i \in [t]$, define the function $\mathsf{FSim}_i[x^*, w^*](x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1})$:
  - If $x = x^*$, output $\mathsf{FSim}_i(x^*, \sigma_{x^*}, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1})$, where FSim_i is defined as in QSObf_5.
  - If $x \neq x^*$, output $\mathsf{FSim}_i[w^*](x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1})$, where $\mathsf{FSim}_i[w^*]$ is defined as in QSObf_4.
- Define GSim as in QSObf_2.
- Output $|\widetilde{\psi}\rangle = |\psi_k\rangle\,|\mathsf{sk}\rangle\,, O = (\mathsf{FSim}_1[\cdot], \ldots, \mathsf{FSim}_t[\cdot], \mathsf{GSim})\,.$

$\mathsf{ParSim}_5(1^\lambda, |\psi\rangle)$:

- Sample $k \leftarrow \mathsf{Gen}(1^\kappa, n)$, and compute $|\psi_k\rangle = \mathsf{Enc}_k(|\psi\rangle)$.
- Sample a signature token $(\mathsf{vk}, |\mathsf{sk}\rangle) \leftarrow \mathsf{TokGen}(1^\kappa)$.
- Let $H : \{0,1\}^* \to \{0,1\}^\kappa$ be a random oracle.
- For each $i \in [t]$, define the function FSim_i as in QSObf_5.
- Define GSim as in QSObf_2.
- Output $|\widetilde{\psi}\rangle = |\psi_k\rangle\,|\mathsf{sk}\rangle\,, O = (\mathsf{FSim}_1, \ldots, \mathsf{FSim}_t, \mathsf{GSim})\,.$

Next, before proving the main lemma (Lemma 5.44) of this section, which involves $\mathsf{ParMeas}_0$, we prove several claims about these hybrid distributions, which will allow us to use the properties of simulated distributions when proving Lemma 5.44. We will essentially "work backwards" from ParSim_5 to ParSim_0 in order to show the sequence of indistinguishability claims we will need. Whenever we write $\{w_i^*\}_{i\in[S]}$ for some set $S \subseteq [t]$, we parse each $w_i^* \in \{0,1\}^{|W_i|}$.

First, we confirm that the mapping hardness claims we'll need hold in the fully simulated case of ParSim_5.

Claim 5.37. *For any QPQ unitary U, input x^*, partial measurement results $\{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}$, and $w^* = \{w_i^*\}_{i\in[t]}$, it holds that*

$$\mathbb{E}\left[\left\|\Pi[\neg w^*]U^O\Pi[w^*]\,|\widetilde{\psi}\rangle\,\right\|^2 : |\widetilde{\psi}\rangle\,, O \leftarrow \mathsf{ParSim}_5(1^\lambda, |\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]\rangle)\right] = 2^{-\Omega(\kappa)}$$

and

$$\mathbb{E}\left[\left\|\Pi[\neg(v_\tau^*, r_\tau^*)]U^O\Pi[\{w_\iota^*\}_{\iota\in[\tau+1,t]}]\,|\widetilde{\psi}\rangle\,\right\|^2 : |\widetilde{\psi}\rangle\,, O \leftarrow \mathsf{ParSim}_5(1^\lambda, |\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]\rangle)\right] = 2^{-\Omega(\kappa)},$$

where $\Pi[\neg(v_\tau^*, r_\tau^*)]$ is defined in the proof of Lemma 5.35.

Proof. The key point is that in QSObf_5, none of the oracles $\mathsf{FSim}_1, \ldots, \mathsf{FSim}_t, \mathsf{GSim}$ require access to the decryption oracle $\mathsf{Dec}_{k,\cdot,\cdot}(\cdot)$ for the authentication scheme. Rather, they can

be implemented just given access to the verification oracle $\mathsf{Ver}_{k,\cdot,\cdot}(\cdot)$. Thus, these claims follow directly from the security of the authentication scheme (Theorem 5.9), and in particular that it satisfies Definition 5.4 (mapping security). Note that in the second claim, we only collapse the wires $\{W_i\}_{i\in[\tau+1,t]}$, that is, the W wires *after* level τ, which are disjoint from (V_τ, W_τ). The claim could have been trivially false if we had collapsed the wires in (V_τ, W_τ), in particular to an outcome different from (v^*_τ, r^*_τ). □

The next two claims establish the indistinguishability of ParSim_3, ParSim_4, and ParSim_5 in the case when all of the W wires have been collapsed to some value w^*.

Claim 5.38. *For any (unbounded) distinguisher D, input x^*, partial measurement results $\{v^*_\iota, r^*_\iota\}_{\iota\in[\tau]}$, and $w^* = \{w^*_i\}_{i\in[t]}$, it holds that*

$$\mathbb{E}\left[\left\|D^{\mathsf{FSim}_1[x^*,w^*],\ldots,\mathsf{GSim}}\left(k,\Pi[w^*]\left|\widetilde{\psi}\right\rangle\right)\right\|^2 : \begin{array}{l}\left|\widetilde{\psi}\right\rangle,(\mathsf{FSim}_1[\cdot],\ldots,\mathsf{GSim})\\ \leftarrow \mathsf{ParSim}_4(1^\lambda,|\psi[x^*,\{v^*_\iota,r^*_\iota\}_{\iota\in[\tau]}]\rangle)\end{array}\right]$$
$$=\mathbb{E}\left[\left\|D^{\mathsf{FSim}_1,\ldots,\mathsf{GSim}}\left(k,\Pi[w^*]\left|\widetilde{\psi}\right\rangle\right)\right\|^2 : \begin{array}{l}\left|\widetilde{\psi}\right\rangle,(\mathsf{FSim}_1,\ldots,\mathsf{GSim})\\ \leftarrow \mathsf{ParSim}_5(1^\lambda,|\psi[x^*,\{v^*_\iota,r^*_\iota\}_{\iota\in[\tau]}]\rangle)\end{array}\right].$$

where D's input includes the key k sampled by $\mathsf{ParSim}_4, \mathsf{ParSim}_5$.

Proof. The proof is exactly the same as the proof of Claim 5.24, except that here we are only switching oracle inputs on $x \neq x^*$ rather than all x. □

Claim 5.39. *For any QPQ distinguisher D, input x^*, partial measurement results $\{v^*_\iota, r^*_\iota\}_{\iota\in[\tau]}$, and $w^* = \{w^*_i\}_{i\in[t]}$, it holds that*

$$\left|\mathbb{E}\left[\left\|D^{\mathsf{FSim}_1[x^*],\ldots,\mathsf{GSim}}\left(k,\Pi[w^*]\left|\widetilde{\psi}\right\rangle\right)\right\|^2 : \begin{array}{l}\left|\widetilde{\psi}\right\rangle,(\mathsf{FSim}_1[\cdot],\ldots,\mathsf{GSim})\\ \leftarrow \mathsf{ParSim}_3(1^\lambda,|\psi[x^*,\{v^*_\iota,r^*_\iota\}_{\iota\in[\tau]}]\rangle)\end{array}\right]\right.$$
$$\left.-\mathbb{E}\left[\left\|D^{\mathsf{FSim}_1[x^*,w^*],\ldots,\mathsf{GSim}}\left(k,\Pi[w^*]\left|\widetilde{\psi}\right\rangle\right)\right\|^2 : \begin{array}{l}\left|\widetilde{\psi}\right\rangle,(\mathsf{FSim}_1[\cdot],\ldots,\mathsf{GSim})\\ \leftarrow \mathsf{ParSim}_4(1^\lambda,|\psi[x^*,\{v^*_\iota,r^*_\iota\}_{\iota\in[\tau]}]\rangle)\end{array}\right]\right|$$
$$=2^{-\Omega(\kappa)},$$

where D's input includes the key k sampled by $\mathsf{ParSim}_3, \mathsf{ParSim}_4$.

Proof. The proof is exacly the same as the proof of Claim 5.25, except that here we are only switching oracle inputs on $x \neq x^*$ rather than all x. □

This next claim shows that in ParSim_3, it is hard to map wires $W_{\tau+1},\ldots,W_t$ collapsed to $w^*_{\tau+1},\ldots,w^*_t$ to an outcome different from $w^*_{\tau+1},\ldots,w^*_t$.

Claim 5.40. *For any QPQ unitary U, input x^*, partial measurement results $\{v^*_\iota, r^*_\iota\}_{\iota\in[\tau]}$, and $\{w^*_i\}_{i\in[\tau+1,t]}$, it holds that*

$$\mathbb{E}\left[\left\|\Pi[\neg\{w^*_i\}_{i\in[\tau+1,t]}]U^{\mathsf{FSim}_1[x^*],\ldots,\mathsf{GSim}}\Pi[\{w^*_i\}_{i\in[\tau+1,t]}]\left|\widetilde{\psi}\right\rangle\right\|^2\right]=2^{-\Omega(\kappa)},$$

where the expectation is over

$$\left|\widetilde{\psi}\right\rangle,(\mathsf{FSim}_1[\cdot],\ldots,\mathsf{GSim})\leftarrow\mathsf{ParSim}_3(1^\lambda,|\psi[x^*,\{v^*_\iota,r^*_\iota\}_{\iota\in[\tau]}]\rangle).$$

Proof. We write $\mathbb{E}_{\mathsf{ParSim}_3}$ as shorthand for the expectation over

$$|\widetilde{\psi}\rangle, (\mathsf{FSim}_1[\cdot], \ldots, \mathsf{GSim}) \leftarrow \mathsf{ParSim}_3(1^\lambda, |\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]\rangle),$$

we write $\mathbb{E}_{\mathsf{ParSim}_5}$ as shorthand for the expectation over

$$|\widetilde{\psi}\rangle, (\mathsf{FSim}_1, \ldots, \mathsf{GSim}) \leftarrow \mathsf{ParSim}_5(1^\lambda, |\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]\rangle),$$

and, given $\{w_\iota^*\}_{\iota\in[\tau]}, \{w_\iota^*\}_{\iota\in[\tau+1,t]}$, we write $w^* = \{w_\iota^*\}_{\iota\in[\tau]} \cup \{w_\iota^*\}_{\iota\in[\tau+1,t]}$. Then,

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathsf{ParSim}_3}\left[\left\|\Pi[\neg\{w_i^*\}_{i\in[\tau+1,t]}]U^{\mathsf{FSim}_1[x^*],\ldots,\mathsf{GSim}}\Pi[\{w_i^*\}_{i\in[\tau+1,t]}]\,|\widetilde{\psi}\rangle\right\|^2\right]\\
&\leq 2^n\cdot\sum_{\{w_\iota^*\}_{\iota\in[\tau]}}\mathop{\mathbb{E}}_{\mathsf{ParSim}_3}\left[\left\|\Pi[\neg\{w_i^*\}_{i\in[\tau+1,t]}]U^{\mathsf{FSim}_1[x^*],\ldots,\mathsf{GSim}}\Pi[w^*]\,|\widetilde{\psi}\rangle\right\|^2\right]\\
&\leq 2^n\cdot\sum_{\{w_\iota^*\}_{\iota\in[\tau]}}\mathop{\mathbb{E}}_{\mathsf{ParSim}_5}\left[\left\|\Pi[\neg\{w_i^*\}_{i\in[\tau+1,t]}]U^{\mathsf{FSim}_1,\ldots,\mathsf{GSim}}\Pi[w^*]\,|\widetilde{\psi}\rangle\right\|^2\right]+2^n\cdot 2^{-\Omega(\kappa)}\\
&=2^{-\Omega(\kappa)}
\end{aligned}$$

where

- The first inequality follows from Lemma 2.7 (an application of Cauchy-Schwarz).
- The second inequality follows by combining Claim 5.39 and Claim 5.38 proven above.
- The third inequality follows from Claim 5.37 proven above.

□

In the proof of the next two claims, we will use the following notation. Fix a key k, input x^*, partial measurement results $\{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}$, and $\{w_\iota^*\}_{i\in[\tau+1,t]}$, and define the following function:

$R[k, x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}, \{w_\iota^*\}_{\iota\in[\tau+1,t]}] : (\widetilde{v}_1, \ldots, \widetilde{v}_i) \to r_i$

- If $i \leq \tau$, output r_i^*.
- Otherwise, compute $(v_{\tau+1}, \ldots, v_i) = \mathsf{Dec}_{k,L_i\ldots L_1,\theta_i[V_{\tau+1},\ldots,V_i]}(\widetilde{v}_{\tau+1}, \ldots, \widetilde{v}_i)$, and output $\perp$ if the result is $\perp$.
- For $\iota \in [\tau+1, i]$, compute $(\cdot, r_\iota) = f_\iota^{x^*, r_1^*, \ldots, r_\tau^*, r_{\tau+1}, \ldots, r_{\iota-1}}(v_1^*, \ldots, v_\tau^*, v_{\tau+1}, \ldots, v_\iota, w_\iota^*)$.
- Output r_i.

This function determines the bit r_i when x^*, $\{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}$, and $\{w_\iota^*\}_{\iota\in[\tau+1,t]}$ have been hard-coded into the oracles in ParSim_2. We will use it when showing indistinguishability between $\mathsf{ParSim}_1, \mathsf{ParSim}_2$, and ParSim_3 in the case when the $W_{\tau+1}, \ldots, W_t$ wires of the input state have been collapsed to outcome $w_{\tau+1}^*, \ldots, w_t^*$.

Claim 5.41. *For any (unbounded) distinguisher D, input x^*, partial measurement results $\{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}$, and $\{w_\iota^*\}_{i\in[\tau+1,t]}$, it holds that*

$$\begin{aligned}&\mathbb{E}\left[\left\|D^{\mathsf{F}_1[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]},\{w_\iota^*\}_{\iota\in[\tau+1,t]},\ldots,\mathsf{GSim}}\left(k,\mathsf{vk},\Pi[\{w_\iota^*\}_{\iota\in[\tau+1,t]}]\,|\widetilde{\psi}\rangle\right)\right\|^2\right]\\&=\mathbb{E}\left[\left\|D^{\mathsf{FSim}_1[x^*],\ldots,\mathsf{GSim}}\left(k,\mathsf{vk},\Pi[\{w_\iota^*\}_{\iota\in[\tau+1,t]}]\,|\widetilde{\psi}\rangle\right)\right\|^2\right],\end{aligned}$$

where the first expectation is over

$$|\widetilde{\psi}\rangle, (\mathsf{F}_1[\cdot], \ldots, \mathsf{F}_t[\cdot], \mathsf{GSim}) \leftarrow \mathsf{ParSim}_2(1^\lambda, |\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]\rangle),$$

the second expectation is over

$$|\widetilde{\psi}\rangle, (\mathsf{FSim}_1[\cdot], \ldots, \mathsf{FSim}_t[\cdot], \mathsf{GSim}) \leftarrow \mathsf{ParSim}_3(1^\lambda, |\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]\rangle),$$

and D's input includes the keys k, vk sampled by $\mathsf{ParSim}_2, \mathsf{ParSim}_3$.

Proof. In ParSim_2, we have that for inputs that begin with x^*,

$$\begin{aligned}&\mathsf{F}_i[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}, \{w_\iota^*\}_{\iota\in[\tau+1,t]}](x^*, \sigma_{x^*}, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1})\\&\in \{H(x^*, \sigma_{x^*}, \widetilde{v}_1, \ldots, \widetilde{v}_i, \ell_1, \ldots, \ell_{i-1}, R[k, x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}, \{w_\iota^*\}_{\iota\in[\tau+1,t]}](\widetilde{v}_1, \ldots, \widetilde{v}_i)), \bot\},\end{aligned}$$

while in ParSim_3, we have that for inputs that begin with x^*,

$$\begin{aligned}&\mathsf{FSim}_i[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}, \{w_\iota^*\}_{\iota\in[\tau+1,t]}](x^*, \sigma_{x^*}, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1})\\&\in \{H(x^*, \sigma_{x^*}, \widetilde{v}_1, \ldots, \widetilde{v}_i, \ell_1, \ldots, \ell_{i-1}, 0), \bot\}.\end{aligned}$$

Both implementations of the oracles are identical on inputs $x \neq x^*$, and both will always output $\bot$ on the same set of inputs, since this is true of $\mathsf{Dec}_{k,\cdot,\cdot}(\cdot)$ and $\mathsf{Ver}_{k,\cdot,\cdot}(\cdot)$ by definition. Finally, their non-$\bot$ answers on inputs that begin with x^* are identically distributed over the randomness of the random oracle H, since each $(x^*, \sigma_{x^*}, \widetilde{v}_1, \ldots, \widetilde{v}_i, \ell_1, \ldots, \ell_{i-1})$ fixes a single choice of bit

$$R[k, x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}, \{w_\iota^*\}_{\iota\in[\tau+1,t]}](\widetilde{v}_1, \ldots, \widetilde{v}_i).$$

Indeed, for any $(x^*, \sigma_{x^*}, \widetilde{v}_1, \ldots, \widetilde{v}_i, \ell_1, \ldots, \ell_{i-1})$,

$$H(x^*, \sigma_{x^*}, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1}, 0) \quad \text{and} \quad H(x^*, \sigma_{x^*}, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1}, 1)$$

are identically distributed (each is a uniformly random string).

□

Claim 5.42. *For any QPQ distinguisher D, input x^*, partial measurement results $\{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}$, and $\{w_\iota^*\}_{i\in[\tau+1,t]}$, it holds that*

$$\left| \mathbb{E}\left[\left\| D^{\mathsf{F}_1[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]},\{w_\iota^*\}_{\iota\in[\tau+1,t]],\ldots,\mathsf{GSim}}\left(k,\mathsf{vk},\Pi[\{w_\iota^*\}_{\iota\in[\tau+1,t]}]\,|\widetilde{\psi}\rangle\right)\right\|^2\right] \right.$$
$$\left. - \mathbb{E}\left[\left\| D^{\mathsf{F}_1[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]},\{w_\iota^*\}_{\iota\in[\tau+1,t]],\ldots,\mathsf{GSim}}\left(k,\mathsf{vk},\Pi[\{w_\iota^*\}_{\iota\in[\tau+1,t]}]\,|\widetilde{\psi}\rangle\right)\right\|^2\right] \right| = 2^{-\Omega(\kappa)},$$

where the first expectation is over

$$|\widetilde{\psi}\rangle\,, (\mathsf{F}_1[\cdot],\ldots,\mathsf{F}_t[\cdot],\mathsf{GSim}) \leftarrow \mathsf{ParSim}_1(1^\lambda, |\psi[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]}]\rangle),$$

and the second expectation is over

$$|\widetilde{\psi}\rangle\,, (\mathsf{F}_1[\cdot],\ldots,\mathsf{F}_t[\cdot],\mathsf{GSim}) \leftarrow \mathsf{ParSim}_2(1^\lambda, |\psi[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]}]\rangle),$$

and D's input includes the keys k, vk sampled by $\mathsf{ParSim}_1, \mathsf{ParSim}_2$.

Proof. Observe that the oracles $\mathsf{F}_i[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]},\{w_\iota^*\}_{\iota\in[\tau+1,t]}]$ in these experiments are identical except for on inputs

$$(x^*, \sigma_{x^*}, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1})$$

such that there exists an $\iota \in [i-1]$ with

$$\ell_\iota = H(x^*, \sigma_{x^*}, \widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 1 - R[k, x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}, \{w_\iota^*\}_{\iota\in[\tau+1,t]}](\widetilde{v}_1, \ldots, \widetilde{v}_\iota)).$$

However, the oracles $\mathsf{F}_i[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]},\{w_\iota^*\}_{\iota\in[\tau+1,t]}]$ in ParSim_2 are defined to never output such an ℓ_ι, and thus such an input can only be guessed with probability $2^{-\kappa}$ over the randomness of H. The claim follows by applying Lemma 2.8 (a standard oracle hybrid argument). □

Now, in our final claim before the main lemma of this section, we combine what we have shown so far to establish the indistinguishability of ParSim_0 and ParSim_1 in the case when the $W_{\tau+1}, \ldots, W_t$ wires of the input state have been collapsed to some outcome $w_{\tau+1}^*, \ldots, w_t^*$.

Claim 5.43. *For any QPQ distinguisher D, input x^*, partial measurement results $\{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}$, and $\{w_\iota^*\}_{i\in[\tau+1,t]}$, it holds that*

$$\left| \mathbb{E}\left[\left\| D^{\mathsf{F}_1[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]],\ldots,\mathsf{GSim}}\left(k,\mathsf{vk},\Pi[\{w_\iota^*\}_{\iota\in[\tau+1,t]}]\,|\widetilde{\psi}\rangle\right)\right\|^2\right] \right.$$
$$\left. - \mathbb{E}\left[\left\| D^{\mathsf{F}_1[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]},\{w_\iota^*\}_{\iota\in[\tau+1,t]],\ldots,\mathsf{GSim}}\left(k,\mathsf{vk},\Pi[\{w_\iota^*\}_{\iota\in[\tau+1,t]}]\,|\widetilde{\psi}\rangle\right)\right\|^2\right] \right| = 2^{-\Omega(\kappa)},$$

where the first expectation is over

$$|\widetilde{\psi}\rangle\,, (\mathsf{F}_1[\cdot], \ldots, \mathsf{F}_t[\cdot], \mathsf{GSim}) \leftarrow \mathsf{ParSim}_0(1^\lambda, |\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]\rangle),$$

the second expectation is over

$$|\widetilde{\psi}\rangle\,, (\mathsf{F}_1[\cdot], \ldots, \mathsf{F}_t[\cdot], \mathsf{GSim}) \leftarrow \mathsf{ParSim}_1(1^\lambda, |\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]\rangle),$$

and D's input includes the keys k, vk sampled by $\mathsf{ParSim}_0, \mathsf{ParSim}_1$.

Proof. Observe that the oracles in these experiments are identical except for on inputs that include a $\widetilde{w}_i \in P[\neg w_i^*]$ for some $i \in [\tau+1, t]$. Thus, by Lemma 2.8 (a standard oracle hybrid argument), it suffices to show that for any QPQ unitary U,

$$\mathbb{E}\left[\Pi[\neg\{w_\iota^*\}_{i\in[\tau+1,t]}]U^{\mathsf{F}_1[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]},\{w_\iota^*\}_{\iota\in[\tau+1,t]}],\ldots,\mathsf{GSim}}\Pi[\{w_\iota^*\}_{i\in[\tau+1,t]}]\,|\widetilde{\psi}\rangle\right] = 2^{-\Omega(\kappa)},$$

where the expectation is over

$$|\widetilde{\psi}\rangle\,, (\mathsf{F}_1[\cdot], \ldots, \mathsf{F}_t[\cdot], \mathsf{GSim}) \leftarrow \mathsf{ParSim}_1(1^\lambda, |\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]\rangle).$$

Write $\mathbb{E}_{\mathsf{ParSim}_1}$ as shorthand for the expectation over

$$|\widetilde{\psi}\rangle\,, (\mathsf{F}_1[\cdot], \ldots, \mathsf{FSim}_t[\cdot], \mathsf{GSim}) \leftarrow \mathsf{ParSim}_1(1^\lambda, |\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]\rangle),$$

and write $\mathbb{E}_{\mathsf{ParSim}_3}$ as shorthand for the expectation over

$$|\widetilde{\psi}\rangle\,, (\mathsf{FSim}_1[\cdot], \ldots, \mathsf{FSim}_t[\cdot], \mathsf{GSim}) \leftarrow \mathsf{ParSim}_3(1^\lambda, |\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]\rangle).$$

Then,

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathsf{ParSim}_1}\left[\Pi[\neg\{w_\iota^*\}_{i\in[\tau+1,t]}]U^{\mathsf{F}_1[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]},\{w_\iota^*\}_{\iota\in[\tau+1,t]}],\ldots,\mathsf{GSim}}\Pi[\{w_\iota^*\}_{i\in[\tau+1,t]}]\,|\widetilde{\psi}\rangle\right]\\
&\leq \mathop{\mathbb{E}}_{\mathsf{ParSim}_3}\left[\Pi[\neg\{w_\iota^*\}_{i\in[\tau+1,t]}]U^{\mathsf{FSim}_1[x^*],\ldots,\mathsf{GSim}}\Pi[\{w_\iota^*\}_{i\in[\tau+1,t]}]\,|\widetilde{\psi}\rangle\right] + 2^{-\Omega(\kappa)}\\
&\leq 2^{-\Omega(\kappa)},
\end{aligned}$$

where the first inequality follows by combining Claim 5.42 and Claim 5.41, and the second inequality follows from Claim 5.40, all proven above. □

Now, we prove the main lemma of this section. The proof of Lemma 5.44 combines some claims proven above with Lemma 5.45 that follows. Lemma 5.45 is a similar claim but with respect to ParSim_3 instead of ParSim_0.

Lemma 5.44. *For any QPQ unitary U, input x^*, and partial measurement results $\{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}$, it holds that*

$$\mathbb{E}\left[\left\|\left(\Pi[x^*, \mathsf{vk}] \otimes \Pi[\neg(v_\tau^*, r_\tau^*)]\right) U^{\mathsf{F}_1[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]}],\ldots,\mathsf{GSim}}\,|\widetilde{\psi}\rangle\right\|^2\right] = 2^{-\Omega(\kappa)},$$

where the projectors are defined as in the proof of Lemma 5.35, and the expectation is over

$$|\widetilde{\psi}\rangle\,, (\mathsf{F}_1[\cdot], \ldots, \mathsf{F}_t[\cdot], \mathsf{GSim}) \leftarrow \mathsf{ParMeas}(1^\lambda, |\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]\rangle),$$

which, recall, is defined to be the same as

$$|\widetilde{\psi}\rangle\,, (\mathsf{F}_1[\cdot], \ldots, \mathsf{F}_t[\cdot], \mathsf{GSim}) \leftarrow \mathsf{ParSim}_0(1^\lambda, |\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]\rangle).$$

Proof. Write $\mathbb{E}_{\mathsf{ParSim}_0}$ as shorthand for the expectation over

$$|\widetilde{\psi}\rangle\,, (\mathsf{F}_1[\cdot], \ldots, \mathsf{F}_t[\cdot], \mathsf{GSim}) \leftarrow \mathsf{ParSim}_1(1^\lambda, |\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]\rangle),$$

and write $\mathbb{E}_{\mathsf{ParSim}_3}$ as shorthard for the expectation over

$$|\widetilde{\psi}\rangle\,, (\mathsf{FSim}_1[\cdot], \ldots, \mathsf{FSim}_t[\cdot], \mathsf{GSim}) \leftarrow \mathsf{ParSim}_3(1^\lambda, |\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]\rangle).$$

Then,

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathsf{ParSim}_0}\left[\left\| \left(\Pi[x^*, \mathsf{vk}] \otimes \Pi[\neg(v_\tau^*, r_\tau^*)]\right) U^{\mathsf{F}_1[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]}],\ldots,\mathsf{GSim}} |\widetilde{\psi}\rangle \right\|^2\right] \\
&\leq 2^n \cdot \sum_{\{w_\iota^*\}_{\iota\in[\tau+1,t]}} \mathop{\mathbb{E}}_{\mathsf{ParSim}_0}\left[\left\| \left(\Pi[x^*, \mathsf{vk}] \otimes \Pi[\neg(v_\tau^*, r_\tau^*)]\right) U^{\mathsf{F}_1[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]}],\ldots,\mathsf{GSim}} \Pi[\{w_\iota^*\}_{\iota\in[\tau+1,t]}] |\widetilde{\psi}\rangle \right\|^2\right] \\
&\leq 2^n \cdot \sum_{\{w_\iota^*\}_{\iota\in[\tau+1,t]}} \mathop{\mathbb{E}}_{\mathsf{ParSim}_3}\left[\left\| \left(\Pi[x^*, \mathsf{vk}] \otimes \Pi[\neg(v_\tau^*, r_\tau^*)]\right) U^{\mathsf{FSim}_1[x^*],\ldots,\mathsf{GSim}} \Pi[\{w_\iota^*\}_{\iota\in[\tau+1,t]}] |\widetilde{\psi}\rangle \right\|^2\right] + 2^{-\Omega(\kappa)} \\
&\leq 2^{-\Omega(\kappa)},
\end{aligned}$$

where

- The first inequality follows from Lemma 2.7 (an application of Cauchy-Schwarz).
- The second inequality follows from combining Claim 5.43, Claim 5.42, and Claim 5.41 proven above.
- The third inequality follows from Lemma 5.45 proven below.

□

We finally complete the proof of security of our obfuscation scheme with the following lemma. An overview of the techniques involved in this proof, which include extraction via a purified random oracle, is given in Section 5.4.2.

Lemma 5.45. *For any QPQ unitary U, input x^*, partial measurement results $\{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}$, and $\{w_\iota^*\}_{\iota\in[\tau+1,t]}$, it holds that*

$$\mathbb{E}\left[\left\| \left(\Pi[x^*, \mathsf{vk}] \otimes \Pi[\neg(v_\tau^*, r_\tau^*)]\right) U^{\mathsf{FSim}_1[x^*],\ldots,\mathsf{GSim}} \Pi[\{w_\iota^*\}_{\iota\in[\tau+1,t]}] \left|\widetilde{\psi}\right\rangle \right\|^2\right] = 2^{-\Omega(\kappa)},$$

where the expectation is over

$$\left|\widetilde{\psi}\right\rangle, (\mathsf{FSim}_1[\cdot], \ldots, \mathsf{FSim}_t[\cdot], \mathsf{GSim}) \leftarrow \mathsf{ParSim}_3(1^\lambda, |\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]\rangle).$$

Proof. Recall that the oracles $\mathsf{FSim}_1[x^*], \ldots, \mathsf{FSim}_t[x^*], \mathsf{GSim}$ output by ParSim_3 internally make use of a random oracle $H : \{0,1\}^* \to \{0,1\}^\kappa$, and let $K = \mathrm{poly}(\lambda)$ be an upper bound on the length of strings that H takes as input.

We will purify the random oracle H, introducing an oracle register $\mathcal{D} \coloneqq \{\mathcal{D}_a\}_{a\in\{0,1\}^K}$, where each $\mathcal{D}_a$ is a κ-qubit register. Define $|+_\kappa\rangle$ to be the uniform superposition over all κ-bit strings, and define $|+_H\rangle^{\mathcal{D}} \coloneqq |+_\kappa\rangle^{\otimes\{\mathcal{D}_a\}_{a\in\{0,1\}^K}}$ to be the uniform superposition over all random oracles H. Finally, let $A[\neq x^*] \coloneqq \{a \in \{0,1\}^K : a = (x, \cdot) \text{ for } x \neq x^*\}$ be the set of all random oracle inputs / sub-registers of $\mathcal{D}$ that do not start with x^*.

Now, the purified random oracle begins by initializing $\mathcal{D}$ to the state $|+_H\rangle$. Each time a query to H is made on input register $\mathcal{A}$ and output register $\mathcal{B}$, we apply a unitary defined by the map

$$|a\rangle^{\mathcal{A}} |b\rangle^{\mathcal{B}} |H\rangle^{\mathcal{D}} \to |a\rangle \left(\mathsf{CNOT}^{\otimes\kappa}\right)^{\mathcal{D}_a,\mathcal{B}} |b\rangle^{\mathcal{B}} |H\rangle^{\mathcal{D}}.$$

For the rest of this proof, we will implement oracle queries to H using this purified procedure, and explicitly introduce the register $\mathcal{D}$, initialized to $|+_H\rangle$, in our expressions.

The central claim we need is the following, which shows that is is hard to map onto $\Pi[\neg(v_\tau^*, r_\tau^*)]$ *without disturbing* the random oracle registers $\{\mathcal{D}_a\}_{a\in A[\neq x^*]}$. In other words, at any point at which the adversary's state has some overlap with $\Pi[\neg(v_\tau^*, r_\tau^*)]$, it *must* be the case that the adversary currently holds some information about a random oracle output at $(x, \ldots)$ for some $x \neq x^*$.

Claim 5.46. *For any QPQ unitary U, input x^*, partial measurement results $\{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}$, and $\{w_\iota^*\}_{\iota\in[\tau+1,t]}$, it holds that*

$$\mathbb{E}\left[\left\| \left(\Pi[\neg(v_\tau^*, r_\tau^*)] \otimes |+_\kappa\rangle\langle+_\kappa|^{\otimes\{\mathcal{D}_a\}_{a\in A[\neq x^*]}}\right) U^{\mathsf{FSim}_1[x^*],\ldots,\mathsf{GSim}} \left|\widetilde{\psi}^*\right\rangle |+_H\rangle^{\mathcal{D}} \right\|^2\right] = 2^{-\Omega(\kappa)},$$

where the expectation is over

$$\left|\widetilde{\psi}\right\rangle, (\mathsf{FSim}_1[\cdot], \ldots, \mathsf{FSim}_t[\cdot], \mathsf{GSim}) \leftarrow \mathsf{ParSim}_3(1^\lambda, |\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]\rangle),$$

and

$$\left|\widetilde{\psi}^*\right\rangle \coloneqq \Pi[\{w_\iota^*\}_{\iota\in[\tau+1,t]}] \left|\widetilde{\psi}\right\rangle.$$

Proof. First, consider the following distribution, which essentially toggles between ParSim_4 and ParSim_5.

$\mathsf{Sim}[x^*](1^\lambda, |\psi\rangle)$:

- Sample $k \leftarrow \mathsf{Gen}(1^\kappa, n)$, and compute $|\psi_k\rangle = \mathsf{Enc}_k(|\psi\rangle)$.
- Sample a signature token $(\mathsf{vk}, |\mathsf{sk}\rangle) \leftarrow \mathsf{TokGen}(1^\kappa)$.
- Let $H : \{0,1\}^* \to \{0,1\}^\kappa$ be a random oracle.
- For each $i \in [t]$, define the function $\mathsf{FSim}_i[x^*][z](x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1})$:
  - If $z = \emptyset$, output $\mathsf{FSim}_i(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1})$, where FSim_i is defined as in QSObf_5.
  - Otherwise, if $z = w^*$:
    - If $x = x^*$, output $\mathsf{FSim}_i(x^*, \sigma_{x^*}, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1})$.
    - If $x \neq x^*$, output $\mathsf{FSim}_i[w^*](x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1})$, where $\mathsf{FSim}_i[w^*]$ is defined as in QSObf_4.
- Define GSim as in QSObf_2.
- Output $|\widetilde{\psi}\rangle = |\psi_k\rangle\,|\mathsf{sk}\rangle\,, O = (\mathsf{FSim}_1[x^*][\cdot], \ldots, \mathsf{FSim}_t[x^*][\cdot], \mathsf{GSim})\,.$

Indeed, observe that

$$|\widetilde{\psi}\rangle\,, (\mathsf{FSim}_1[x^*][w^*], \ldots, \mathsf{FSim}_t[x^*][w^*], \mathsf{GSim}) \leftarrow \mathsf{Sim}[x^*](1^\lambda, |\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]\rangle)$$

is equivalent to

$$|\widetilde{\psi}\rangle\,, (\mathsf{FSim}_1[x^*, w^*], \ldots, \mathsf{FSim}_t[x^*, w^*], \mathsf{GSim}) \leftarrow \mathsf{ParSim}_4(1^\lambda, |\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]\rangle),$$

while

$$|\widetilde{\psi}\rangle\,, (\mathsf{FSim}_1[x^*][\emptyset], \ldots, \mathsf{FSim}_t[x^*][\emptyset], \mathsf{GSim}) \leftarrow \mathsf{Sim}[x^*](1^\lambda, |\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]\rangle)$$

is equivalent to

$$|\widetilde{\psi}\rangle\,, (\mathsf{FSim}_1, \ldots, \mathsf{FSim}_t, \mathsf{GSim}) \leftarrow \mathsf{ParSim}_5(1^\lambda, |\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]\rangle).$$

Next, recall the definition of $R[k, w^*]$ from Section 5.4.4:

$R[k, w^*](x, \widetilde{v}_1, \ldots, \widetilde{v}_i)$:

- Compute $(v_1, \ldots, v_i) = \mathsf{Dec}_{k, L_i \ldots L_1, \theta_i[V_1, \ldots, V_i]}(\widetilde{v}_1, \ldots, \widetilde{v}_i)$. If the result is $\perp$, then output $\perp$.
- For $\iota \in [i]$, compute $(\cdot, r_\iota) = f_\iota^{x, r_1, \ldots, r_{\iota-1}}(v_1, \ldots, v_\iota, w_\iota^*)$.
- Output r_i.

Now, for any $w^* = \{w^*_\iota\}_{\iota\in[t]}$, define a unitary $\Sigma[w^*]$ that permutes the registers $A[\neq x^*]$ according to the rule that, for $x \neq x^*$, swaps

$$(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1}, 0) \quad \text{and} \quad (x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1}, 1)$$

whenever $R[k, w^*](x, \widetilde{v}_1, \ldots, \widetilde{v}_i) = 1$. By definition of $\mathsf{Sim}[x^*]$, we have the following fact.

Fact 5.47. *For any unitary U and*

$$|\widetilde{\psi}\rangle, (\mathsf{FSim}_1[x^*][\cdot], \ldots, \mathsf{FSim}_t[x^*][\cdot], \mathsf{GSim}) \in \mathsf{Sim}[x^*](1^\lambda, |\psi[x^*, \{v^*_\iota, r^*_\iota\}_{\iota\in[\tau]}]\rangle),$$

it holds that

$$U^{\mathsf{FSim}_1[x^*][w^*],\ldots,\mathsf{GSim}} |\widetilde{\psi}^*\rangle |+_H\rangle^{\mathcal{D}} = \Sigma[w^*] U^{\mathsf{FSim}_1[x^*][\emptyset],\ldots,\mathsf{GSim}} \Sigma[w^*] |\widetilde{\psi}^*\rangle |+_H\rangle^{\mathcal{D}},$$

where the unitary $\Sigma[w^*]$ is applied to registers $\{\mathcal{D}_a\}_{a\in A[\neq x^*]}$.

We will also use the following immediate fact.

Fact 5.48. *For any w^*,*

$$\Sigma[w^*] |+_\kappa\rangle^{\otimes\{\mathcal{D}_a\}_{a\in A[\neq x^*]}} = |+_\kappa\rangle^{\otimes\{\mathcal{D}_a\}_{a\in A[\neq x^*]}}.$$

Now, write $\mathbb{E}_{\mathsf{ParSim}_3}$ as shorthand for the expectation over

$$|\widetilde{\psi}\rangle, (\mathsf{FSim}_1[\cdot], \ldots, \mathsf{FSim}_t[\cdot], \mathsf{GSim}) \leftarrow \mathsf{ParSim}_3(1^\lambda, |\psi[x^*, \{v^*_\iota, r^*_\iota\}_{\iota\in[\tau]}]\rangle),$$

write $\mathbb{E}_{\mathsf{ParSim}_4}$ as shorthand for the expectation over

$$|\widetilde{\psi}\rangle, (\mathsf{FSim}_1[\cdot], \ldots, \mathsf{FSim}_t[\cdot], \mathsf{GSim}) \leftarrow \mathsf{ParSim}_4(1^\lambda, |\psi[x^*, \{v^*_\iota, r^*_\iota\}_{\iota\in[\tau]}]\rangle),$$

write $\mathbb{E}_{\mathsf{ParSim}_5}$ as shorthand for the expectation over

$$|\widetilde{\psi}\rangle, (\mathsf{FSim}_1, \ldots, \mathsf{FSim}_t, \mathsf{GSim}) \leftarrow \mathsf{ParSim}_5(1^\lambda, |\psi[x^*, \{v^*_\iota, r^*_\iota\}_{\iota\in[\tau]}]\rangle),$$

and write $\mathbb{E}_{\mathsf{Sim}[x^*]}$ as shorthand for

$$|\widetilde{\psi}\rangle, (\mathsf{FSim}_1[x^*][\cdot], \ldots, \mathsf{FSim}_t[x^*][\cdot], \mathsf{GSim}) \leftarrow \mathsf{Sim}[x^*](1^\lambda, |\psi[x^*, \{v^*_\iota, r^*_\iota\}_{\iota\in[\tau]}]\rangle).$$

Then,

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{\mathsf{ParSim}_3}\left[\left\|\left(\Pi[\neg(v^*_\tau,r^*_\tau)]\otimes|+_\kappa\rangle\langle+_\kappa|^{\otimes\{\mathcal{D}_a\}_{a\in A[\neq x^*]}}\right)U^{\mathsf{FSim}_1[x^*],\ldots,\mathsf{GSim}}|\widetilde{\psi}^*\rangle\,|+_H\rangle^{\mathcal{D}}\right\|^2\right]\\
&=\mathop{\mathbb{E}}_{\mathsf{ParSim}_3}\left[\left\|\sum_{w^*}\left(\Pi[\neg(v^*_\tau,r^*_\tau)]\otimes|+_\kappa\rangle\langle+_\kappa|^{\otimes\{\mathcal{D}_a\}_{a\in A[\neq x^*]}}\right)U^{\mathsf{FSim}_1[x^*],\ldots,\mathsf{GSim}}\Pi[w^*]\,|\widetilde{\psi}^*\rangle\,|+_H\rangle^{\mathcal{D}}\right\|^2\right]\\
&=\mathop{\mathbb{E}}_{\mathsf{ParSim}_3}\left[\left\|\Pi[\neg(v^*_\tau,r^*_\tau)]\sum_{w^*}\left(|+_\kappa\rangle\langle+_\kappa|^{\otimes\{\mathcal{D}_a\}_{a\in A[\neq x^*]}}\right)U^{\mathsf{FSim}_1[x^*],\ldots,\mathsf{GSim}}\Pi[w^*]\,|\widetilde{\psi}^*\rangle\,|+_H\rangle^{\mathcal{D}}\right\|^2\right]\\
&\le\mathop{\mathbb{E}}_{\mathsf{ParSim}_4}\left[\left\|\Pi[\neg(v^*_\tau,r^*_\tau)]\sum_{w^*}\left(|+_\kappa\rangle\langle+_\kappa|^{\otimes\{\mathcal{D}_a\}_{a\in A[\neq x^*]}}\right)U^{\mathsf{FSim}_1[x^*,w^*],\ldots,\mathsf{GSim}}\Pi[w^*]\,|\widetilde{\psi}^*\rangle\,|+_H\rangle^{\mathcal{D}}\right\|^2\right]+2^n\cdot2^{-\Omega(\kappa)}\\
&=\mathop{\mathbb{E}}_{\mathsf{Sim}[x^*]}\left[\left\|\Pi[\neg(v^*_\tau,r^*_\tau)]\sum_{w^*}\left(|+_\kappa\rangle\langle+_\kappa|^{\otimes\{\mathcal{D}_a\}_{a\in A[\neq x^*]}}\right)U^{\mathsf{FSim}_1[x^*][w^*],\ldots,\mathsf{GSim}}\Pi[w^*]\,|\widetilde{\psi}^*\rangle\,|+_H\rangle^{\mathcal{D}}\right\|^2\right]+2^{-\Omega(\kappa)}\\
&=\mathop{\mathbb{E}}_{\mathsf{Sim}[x^*]}\left[\left\|\Pi[\neg(v^*_\tau,r^*_\tau)]\sum_{w^*}\left(|+_\kappa\rangle\langle+_\kappa|^{\otimes\{\mathcal{D}_a\}_{a\in A[\neq x^*]}}\right)\Sigma[w^*]U^{\mathsf{FSim}_1[x^*][\emptyset],\ldots,\mathsf{GSim}}\Sigma[w^*]\Pi[w^*]\,|\widetilde{\psi}^*\rangle\,|+_H\rangle^{\mathcal{D}}\right\|^2\right]\\
&\quad+2^{-\Omega(\kappa)}\\
&=\mathop{\mathbb{E}}_{\mathsf{Sim}[x^*]}\left[\left\|\Pi[\neg(v^*_\tau,r^*_\tau)]\sum_{w^*}\left(|+_\kappa\rangle\langle+_\kappa|^{\otimes\{\mathcal{D}_a\}_{a\in A[\neq x^*]}}\right)U^{\mathsf{FSim}_1[x^*][\emptyset],\ldots,\mathsf{GSim}}\Pi[w^*]\,|\widetilde{\psi}^*\rangle\,|+_H\rangle^{\mathcal{D}}\right\|^2\right]+2^{-\Omega(\kappa)}\\
&=\mathop{\mathbb{E}}_{\mathsf{ParSim}_5}\left[\left\|\Pi[\neg(v^*_\tau,r^*_\tau)]\sum_{w^*}\left(|+_\kappa\rangle\langle+_\kappa|^{\otimes\{\mathcal{D}_a\}_{a\in A[\neq x^*]}}\right)U^{\mathsf{FSim}_1,\ldots,\mathsf{GSim}}\Pi[w^*]\,|\widetilde{\psi}^*\rangle\,|+_H\rangle^{\mathcal{D}}\right\|^2\right]+2^{-\Omega(\kappa)}\\
&=\mathop{\mathbb{E}}_{\mathsf{ParSim}_5}\left[\left\|\left(\Pi[\neg(v^*_\tau,r^*_\tau)]\otimes|+_\kappa\rangle\langle+_\kappa|^{\otimes\{\mathcal{D}_a\}_{a\in A[\neq x^*]}}\right)U^{\mathsf{FSim}_1,\ldots,\mathsf{GSim}}|\widetilde{\psi}^*\rangle\,|+_H\rangle^{\mathcal{D}}\right\|^2\right]+2^{-\Omega(\kappa)}\\
&\le 2^{-\Omega(\kappa)},
\end{aligned}
$$

where

- The first inequality follows from Claim 5.39 proven above.
- The following equality is by definition of $\mathsf{Sim}[x^*]$.
- The following equality is Fact 5.47.
- The following equality is Fact 5.48.
- The following equality is by definition of $\mathsf{Sim}[x^*]$.
- The final inequality follows from Claim 5.37 proven above.

□

Now, assume for contradiction that the lemma is false. Combined with the fact that Claim 5.46 is true, this implies that

$$\mathbb{E}\left[\left\|\left(\Pi[x^*,\mathsf{vk}]\otimes\left(\mathcal{I}-|+_\kappa\rangle\langle+_\kappa|^{\otimes\{\mathcal{D}_a\}_{a\in A[\neq x^*]}}\right)\right)U^{\mathsf{FSim}_1[x^*],\ldots,\mathsf{GSim}}\,|\widetilde{\psi}^*\rangle\,|+_H\rangle^{\mathcal{D}}\right\|^2\right]=2^{-o(\kappa)},$$

where the expectation is over

$$|\widetilde{\psi}\rangle\,,(\mathsf{FSim}_1[\cdot],\ldots,\mathsf{FSim}_t[\cdot],\mathsf{GSim})\leftarrow\mathsf{ParSim}_3(1^\lambda,|\psi[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]}]\rangle).$$

However, this would violate the security of the signature token scheme, which we now show. Consider the following reduction that takes as input the signing key $|\mathsf{sk}\rangle$ for a signature token scheme, and has oracle access to $\mathsf{TokVer}[\mathsf{vk}]$.

- Prepare $|\widetilde{\psi}^*\rangle\,,|+_H\rangle$, and $(\mathsf{FSim}_1[\cdot],\ldots,\mathsf{FSim}_t[\cdot],\mathsf{GSim})$ as in the description of Lemma 5.45, and run $U^{\mathsf{FSim}_1[\cdot],\ldots,\mathsf{GSim}}\,|\widetilde{\psi}^*\rangle\,|+_H\rangle$, except that TokVer queries are computed by forwarding them to $\mathsf{TokVer}[\mathsf{vk}]$.
- Measure the final state of U in the standard basis, and parse the outcome as $(\sigma_{x^*},\cdot)$.
- Measure the final state on registers $\{\mathcal{D}_a\}_{a\in A[\neq x^*]}$ in the Hadamard basis. If any register $\mathcal{D}_a$ gives a result other than 0^κ, then parse $a=(x,\sigma_x,\cdot)$ for some $x\neq x^*$.
- Output (σ_{x^*},σ_x).

Then, by the definition of $\Pi[x^*,\mathsf{vk}]$ and the fact that the random oracle H is only ever queried on inputs that begin with (x,σ_x) such that $\mathsf{TokVer}(\mathsf{vk},x,\sigma_x)=\top$, we have that with probability $2^{-o(\kappa)}$, $\mathsf{TokVer}(\mathsf{vk},x^*,\sigma_{x^*})=\mathsf{TokVer}(\mathsf{vk},x,\sigma_x)=\top$, which violates security of the signature token scheme (Definition 2.21). Indeed, note that the signature token scheme is secure against $\mathrm{poly}(\lambda)$ query bounded adversaries that otherwise have unlimited time and space, which is satisfied by the reduction given above.

□

References

[Aar09] Scott Aaronson. Quantum copy-protection and quantum money. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity, CCC 2009, Paris, France, 15-18 July 2009*, pages 229–242. IEEE Computer Society, 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, *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I*, volume 12825 of *Lecture Notes in Computer Science*, pages 497–525. Springer, 2021.

[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. Noninteractive 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.

[AF16] Gorjan Alagic and Bill Fefferman. On quantum obfuscation. *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.

[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, *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I*, volume 12825 of *Lecture Notes in Computer Science*, pages 526–555. Springer, 2021.

[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.

[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.

[BBV24] James Bartusek, Zvika Brakerski, and Vinod Vaikuntanathan. Quantum state obfuscation from classical oracles. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, page 1009–1017, New York, NY, USA, 2024. Association for Computing Machinery.

[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+18] 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. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 320–331. IEEE Computer Society, 2018.

[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+01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 1–18. Springer, Heidelberg, August 2001.

[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.

[BGS13] Anne Broadbent, Gus Gutoski, and Douglas Stebila. Quantum one-time programs - (extended abstract). In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 344–360. Springer, Heidelberg, August 2013.

[BJSW16] Anne Broadbent, Zhengfeng Ji, Fang Song, and John Watrous. Zero-knowledge proof systems for qma. *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 31–40, 2016.

[BK21] Anne Broadbent and Raza Ali Kazmi. Constructions for quantum indistinguishability obfuscation. In Patrick Longa and Carla Ràfols, editors, *Progress in Cryptology - LATINCRYPT 2021 - 7th International Conference on Cryptology and Information Security in Latin America, Bogotá, Colombia, October 6-8, 2021, Proceedings*, volume 12912 of *Lecture Notes in Computer Science*, pages 24–43. Springer, 2021.

[BKNY23] James Bartusek, Fuyuki Kitagawa, Ryo Nishimaki, and Takashi Yamakawa. Obfuscation of pseudo-deterministic quantum circuits. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 1567–1578. ACM, 2023.

[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.

[BOCG+06] Michael Ben-Or, Claude Crépeau, Daniel Gottesman, Avinatan Hassidim, and Adam Smith. Secure multiparty quantum computation with (only) a strict honest majority. pages 249 – 260, 11 2006.

[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.

[BZ14] Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 480–499. Springer, Heidelberg, August 2014.

[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.

[CG24] Andrea Coladangelo and Sam Gunn. How to use quantum indistinguishability obfuscation. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, page 1003–1008, New York, NY, USA, 2024. Association for Computing Machinery.

[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, *Advances in Cryptology – CRYPTO 2021*, pages 556–584, Cham, 2021. Springer International Publishing.

[CLTV15] Ran Canetti, Huijia Lin, Stefano Tessaro, and Vinod Vaikuntanathan. Obfuscation of probabilistic circuits and applications. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 468–497. Springer, Heidelberg, March 2015.

[CMP22] Andrea Coladangelo, Christian Majenz, and Alexander Poremba. Quantum copy-protection of compute-and-compare programs in the quantum random oracle model, 2022.

[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.

[DS18] Yfke Dulek and Florian Speelman. Quantum ciphertext authentication and key recycling with the trap code, 2018.

[DS23] Marcel Dall'Agnol and Nicholas Spooner. On the Necessity of Collapsing for Post-Quantum and Quantum Commitments. *Leibniz Int. Proc. Inf.*, 266:2:1–2:23, 2023.

[GGH+13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 40–49, 2013.

[GGSW13] Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 467–476. ACM Press, June 2013.

[GMR85] S Goldwasser, S Micali, and C Rackoff. The knowledge complexity of interactive proof-systems. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, STOC '85, page 291–304, New York, NY, USA, 1985. Association for Computing Machinery.

[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.

[GR07] Shafi Goldwasser and Guy N. Rothblum. On best-possible obfuscation. In Salil P. Vadhan, editor, *TCC 2007*, volume 4392 of *LNCS*, pages 194–213. Springer, Heidelberg, February 2007.

[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.

[GYZ17] Sumegha Garg, Henry Yuen, and Mark Zhandry. New security notions and feasibility results for authentication of quantum data. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 342–371. Springer, Heidelberg, August 2017.

[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.

[LLQZ22] Jiahui Liu, Qipeng Liu, Luowen Qian, and Mark Zhandry. Collusion resistant copy-protection for watermarkable functionalities. In Eike Kiltz and Vinod Vaikuntanathan, editors, *Theory of Cryptography - 20th International Conference, TCC 2022, Chicago, IL, USA, November 7-10, 2022, Proceedings, Part I*, volume 13747 of *Lecture Notes in Computer Science*, pages 294–323. Springer, 2022.

[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.

[Mah18a] Urmila Mahadev. Classical homomorphic encryption for quantum circuits. In Mikkel Thorup, editor, *59th FOCS*, pages 332–338. IEEE Computer Society Press, October 2018.

[Mah18b] Urmila Mahadev. Classical verification of quantum computations. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 259–267, 2018.

[RSA78] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, feb 1978.

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

[Sho97] 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.

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: Deniable encryption, and more. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '14, page 475–484, New York, NY, USA, 2014. Association for Computing Machinery.

[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.

[WZ17] Daniel Wichs and Giorgos Zirdelis. Obfuscating compute-and-compare programs under lwe. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 600–611, 2017.

[Yao82] Andrew C. Yao. Protocols for secure computations. In *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*, pages 160–164, 1982.

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

[Zha19] Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 239–268. Springer, Heidelberg, August 2019.

[Zha21] Mark Zhandry. Quantum lightning never strikes the same state twice. or: Quantum money from cryptographic assumptions. *J. Cryptol.*, 34(1):6, 2021.