Quantum State Obfuscation from Classical Oracles

James Bartusek* UC Berkeley
Zvika Brakerski† Weizmann Institute of Science
Vinod Vaikuntanathan‡ MIT

January 18, 2024

Abstract

A major unresolved question in quantum cryptography is whether it is possible to obfuscate arbitrary quantum computation. Indeed, there is much yet to understand about the feasibility of quantum obfuscation even in the classical oracle model, where one is given for free the ability to obfuscate any classical circuit.

In this work, we develop a new array of techniques that we use to construct a *quantum state obfuscator*, a powerful notion formalized recently by Coladangelo and Gunn (arXiv:2311.07794) in their pursuit of better software copy-protection schemes. Quantum state obfuscation refers to the task of compiling a *quantum program*, consisting of a quantum circuit C with a classical description and an auxiliary quantum state $|\psi\rangle$, into a functionally-equivalent *obfuscated* quantum program that hides as much as possible about C and $|\psi\rangle$. We prove the security of our obfuscator when applied to any pseudo-deterministic quantum program, i.e. one that computes a (nearly) deterministic classical input / classical output functionality. Our security proof is with respect to an efficient classical oracle, which may be heuristically instantiated using quantum-secure indistinguishability obfuscation for classical circuits.

Our result improves upon the recent work of Bartusek, Kitagawa, Nishimaki and Yamakawa (STOC 2023) who also showed how to obfuscate pseudo-deterministic quantum circuits in the classical oracle model, but only ones with a *completely classical* description. Furthermore, our result answers a question of Coladangelo and Gunn, who provide a construction of quantum state indistinguishability obfuscation with respect to a *quantum* oracle, but leave the existence of a concrete real-world candidate as an open problem. Indeed, our quantum state obfuscator together with Coladangelo-Gunn gives the first candidate realization of a "best-possible" copy-protection scheme for all polynomial-time functionalities.

Our techniques deviate significantly from previous works on quantum obfuscation. We develop several novel technical tools which we expect to be broadly useful in quantum cryptography. These tools include a publicly-verifiable, linearly-homomorphic quantum authentication scheme with classically-decodable ZX measurements (which we build from coset states), and a method for compiling any quantum circuit into a "linear + measurement" (LM) quantum program: an alternating sequence of CNOT operations and partial ZX measurements.

*bartusek.james@gmail.com
†zvika.brakerski@weizmann.ac.il
‡vinodv@csail.mit.edu

Contents

1 Introduction **1**
- 1.1 Our Results . . . 1
- 1.2 Related Work . . . 3

2 Technical Overview **4**
- 2.1 Quantum Authentication from Random Subspaces . . . 4
- 2.2 Linear + Measurement Quantum Programs . . . 7
- 2.3 Obfuscation Construction . . . 10
- 2.4 Discussion and Open Problems . . . 12

3 Preliminaries **13**
- 3.1 Quantum Background . . . 14
- 3.2 Useful Lemmas . . . 15
- 3.3 Signature Tokens . . . 18

4 Authentication Scheme **19**
- 4.1 Definitions . . . 19
- 4.2 Construction . . . 22
- 4.3 Security . . . 25

5 Linear + Measurement Quantum Programs **31**

6 Quantum State Obfuscation: Construction **40**

7 Quantum State Obfuscation: Security **45**
- 7.1 Proof Intuition . . . 45
- 7.2 Notation . . . 50
- 7.3 Main Theorem . . . 51
- 7.4 Inductive Argument . . . 60
- 7.5 Hardness of Mapping . . . 64

8 Acknowledgements **76**

1 Introduction

Whether white-box access to programs is more powerful than black-box access is a fundamental question in computer science. This question motivates the study of *program obfuscation*, the task of converting a given program $\mathcal{P}$ into an obfuscated program $\hat{\mathcal{P}}$ that preserves functionality (i.e. the input-output behavior of $\mathcal{P}$), yet hides all other information about $\mathcal{P}$. Indeed, the previous decades have produced exciting research in cryptography solving several foundational questions about program obfuscation, including: which notions of obfuscation are possible and which aren't [BGI+12, GGH+13]; which ones are useful and to what extent [SW14]; and of course, how to construct secure program obfuscators under well-founded hardness assumptions [JLS21, JLS22]. The study of program obfuscation has given us sophisticated cryptographic primitives such as witness encryption [GGSW13] and functional encryption [GGH+13], and it has shed new light on foundational complexity-theoretic questions [BPR15, ILW23].

Developments in quantum information, computation, and cryptography [Wie83, BB84, Sho94, AC12, BCM+18, Mah18b, JNV+20, Zha21] bring to fore a similarly foundational question:

Is it possible to obfuscate quantum computation, and to what extent?

Despite recent progress, there remains much to understand about the feasibility of quantum obfuscation, even in a world where one has the aid of a classical oracle, equivalently a world where ideal obfuscation of classical circuits comes for free.

The strongest known positive result on obfuscating quantum functionalities is from the recent work of Bartusek, Kitagawa, Nishimaki and Yamakawa [BKNY23], who showed how to obfuscate the class of *classically described* pseudo-deterministic quantum circuits in the classical oracle model, under the learning with errors (LWE) assumption. (For a description of other related work, see Section 1.2.) A deterministic quantum circuit is one that implements a classical, deterministic map $x \to Q(x)$. A *pseudo*-deterministic circuit is nearly deterministic, i.e. its output on every (classical) input x is some (classical) y with all but negligible probability, e.g. the Shor circuit for factoring. In the *classical oracle model*, the obfuscated circuit as well as the (possibly adversarial) evaluator have oracle access to an efficiently computable classical function sampled by the obfuscator.[1]

1.1 Our Results

In this work, we present a new array of techniques that we use to construct a *quantum state obfuscation* scheme, an efficient compiler that converts a *quantum program*, consisting of a quantum circuit C as well as an auxiliary quantum input $|\psi\rangle$, into an obfuscated quantum program that preserves functionality, but hides everything else. In a nutshell, our main result constructs an ideal quantum state obfuscator for pseudo-deterministic functionalities with respect to an efficient classical oracle. The class of circuits we obfuscate is strictly more general than those obfuscated by [BKNY23], which have a *purely classical description* (see Section 2.4 for a more detailed comparison). Moreover, our security is *unconditional*[2] while [BKNY23] additionally relied on the LWE assumption (in the classical oracle model).

[1] Importantly, the actual description of this function is not necessarily revealed to the evaluator / adversary.

[2] Technically, we only achieve unconditional security if we allow our oracles to use a true random oracle as a subroutine. If we insist on the efficiency of the oracles, and thus instantiate the random oracle with a pseudo-random function, then we must assume the existence of a quantum-secure one-way function.

Theorem 1.1 (Informal)**.** *There exists a quantum state obfuscator for the class of pseudo-deterministic quantum programs achieving the notion of ideal obfuscation (Definition 6.1) in the classical oracle model.*

The notion of quantum state obfuscation was recently conceptualized and defined by Coladangelo and Gunn [CG23] in their pursuit of better software copy-protection schemes [Aar09]. They show 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 $x \to Q(x)$ that produces an unclonable quantum program $(|\psi\rangle, C)$, then a quantum state obfuscation of Q is just as good a copy-protection scheme. Indeed, this follows from the fact that the obfuscation of Q is indistinguishable from an obfuscation of $(|\psi\rangle, C)$.

However, they leave 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 main result (Theorem 1.1) answers the open question from [CG23], providing a construction of quantum state obfuscation with respect to a classical oracle. The oracle can be heuristically instantiated using a candidate quantum-secure indistinguishability obfuscation for classical circuits (e.g. [GGH15, BGMZ18, CVW18, BDGM22, GP21, WW21]), thus providing the first concrete evidence that quantum state (indistinguishability) obfuscation is achievable. That is, while we don't currently have a proof in the plain model, it is plausible to conjecture the following.

Conjecture 1. *Instantiating our scheme with one of the above candidate quantum-secure indistinguishability obfuscators for classical circuits yields a secure quantum state indistinguishability obfuscator.*

Then, following Coladangelo-Gunn, we obtain the first candidate realization of a "best-possible" copy-protection scheme for all polynomial-time programs.

Theorem 1.2 (Following [CG23])**.** *Assuming the above conjecture, there exists a best-possible copy-protection scheme for all polynomial-time classical programs.*

We note that the prior work of [ALL+21] also obtains results on general-purpose copy-protection by working in the classical oracle model, and we describe the advantages of our approach (combined with [CG23]) in Section 2.4.

Our techniques deviate significantly from previous works on quantum obfuscation. For example, while previous work [BM22, BKNY23] builds from the idea of obfuscating the verifier for a classical verification of quantum computation scheme, we directly perform verifiable computation on an authenticated and encrypted input. Our construction is comfortingly reminiscent of Yao's classical garbled circuits construction [Yao86] (in fact, the free-XOR variant [KS08]) which has had a tremendous array of applications in cryptography. We give an overview of our techniques in Section 2, and provide more discussion on how they compare to prior work in Section 2.4.

On the classical oracle model. Before proceeding, a word is in order about achieving results in the classical oracle model. First, a result in this model can be interpreted as reducing the problem of obfuscating quantum functionalities to classical functionalities. Indeed, instantiating the (efficiently implementable) oracle by its indistinguishability obfuscation (iO) gives us a heuristically secure construction in the plain model. Furthermore, by the "best-possible" security guarantee of iO, if there exists a secure implementation of the oracle in the plain model, iO is one such.

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	VBB
[CG23]	Quantum	Quantum	Classical	Classical	Deterministic circuits	Quantum oracle model	iO
This work	Quantum	Quantum	Classical	Classical	(Pseudo)-Deterministic circuits	Classical oracle model	Ideal

Table 1: Summary of work on quantum obfuscation. 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 actually achieved in each work: iO is indistinguishability obfuscation, and VBB is virtual black-box obfuscation. We note that VBB and the stronger notion of ideal obfuscation are morally very similar, and [BKNY23] could also likely be shown to be an ideal obfuscator. Finally, we note that while achieving the notion of VBB/ideal obfuscation is only possible in the oracle model, the results that are in the *classical* oracle model yield heuristic candidates for iO in the plain model.

Secondly, results in the classical oracle model have historically been important harbingers of subsequent research that showed analogous results without the aid of an oracle. For example, quantum money [AC12] was first achieved in a classical oracle model before it was de-oracleized [Zha21]; and copy-protection for unlearnable programs was first achieved in a quantum oracle model [Aar09] before it was achieved in a classical oracle model [ALL+21] and later without oracles, for certain classes of functionalities (e.g. [CLLZ21, CMP22, LLQZ22, CG23]).

1.2 Related Work

Alagic and Fefferman [AF16] presented definitions for obfuscating quantum circuits (and obfuscating classical circuits using quantum states). Their results, as well as those of Alagic, Brakerski, Dulek and Schaffner [ABDS21] are negative. The latter, for example, shows 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 T gates. Bartusek and Malavolta [BM22] showed how to achieve indistinguishability obfuscation of null quantum circuits and, most related to our work, Bartusek, Kitagawa, Nishimaki and Yamakawa [BKNY23] showed how to obfuscate general pseudo-deterministic quantum circuits (with a classical description) in the classical oracle model. Finally, Coladangelo and Gunn [CG23], in a concurrent work, 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 our contribution, in Table 1.

2 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 2.1). Next, we discuss the notion of LM quantum programs, and describe a compiler that writes any quantum program as an LM quantum program (Section 2.2). 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 2.3), and mention a couple of key ideas behind proving security. We defer a more detailed proof overview to Section 7.1. We conclude with a discussion and open problems (Section 2.4).

2.1 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 [ABOEM17], and the trap code used for quantum one-time programs [BGS13] and zero-knowledge proofs for QMA [BJSW20].

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 $|0\rangle \to |S\rangle, |1\rangle \to |S+\Delta\rangle$.[3] 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

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

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 CNOT gates, which we refer to as a "linear"[4] 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.

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

[4] 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.

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 $\top$.

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

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

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 [ABOEM17]. 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 4 for the full details of our definitions, construction, and security proofs.

2.2 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".[5] 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. 3 (Page 34), 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,[6] but does support the decoding of Hadamard basis measurements.

[5] 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".

[6] 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.

Handling the T gate. As we show on the bottom left of Fig. 4 (Page 36), 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, taking inspiration from the "encrypted CNOT" operation introduced in [Mah18a],[7] 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. 4, 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. 4: 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 5.5, 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.

[7] 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 5.1, and provide an example diagram of an LM quantum program in Fig. 2.

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 5.2, 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 5.2 to the body.

It may be helpful to refer to Fig. 4 (our implementation of the T gate) and Fig. 2 (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. 4), 1st and 2nd wires of any H-gate circuit (Fig. 3) in the i'th layer, and 1st wire of the i'th T-gate circuit (Fig. 4).
- 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.

2.3 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. $(v_1, \ldots, v_i, w_i) \leftarrow \mathsf{Dec}_{k,\theta_i,L_i \ldots L_1}(\widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i)$.[8] 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 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 7.1. 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

[8] Recall the description of the decoding oracle Dec from Section 2.1. 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.

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 7.1. 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 7.1 that may provide some intuition for the actual proof.

2.4 Discussion and Open Problems

Comparison with [ALL+21]. In Section 1, we described an application of our construction to "best-possible" copy-protection, a notion recently introduced by [CG23]. However, it has already been shown by [ALL+21] that copy-protection for *all unlearnable functions* exists in the classical oracle model. So what is the advantage of our approach?

The catch with [ALL+21] is that it is also known that copy-protection for all unlearnable functions is *unachievable* in the plain model [ALP21]. Thus, one cannot expect to instantiate [ALL+21]'s construction in the plain model with, say, an indistinguishably obfuscator, and conjecture that their notion of security still holds. Moreover, it is unclear (to us) how one would prove that [ALL+21] or any other approach is a "best-possible" copy-protector without going through the intermediate notion of quantum state indistinguishability obfuscation.

On the other hand, there is no known impossibility for quantum state indistinguishability obfuscation in the plain model. Thus, one can conjecture that our approach (or some variant of it) will eventually yield a quantum state indistinguishability obfuscator in the plain model, which, by the results of [CG23], would immediately yield best-possible copy-protection for *any* functionality. Hence, we view our results (combined with [CG23]) as marking significant progress towards the long-standing goal of achieving general-purpose copy-protection from concrete cryptographic assumptions.

Comparison with [BKNY23]. As mentioned in Section 1, our techniques differ significantly from prior work on quantum obfuscation. To be more concrete, let's take the example of [BKNY23], who constructed obfuscation for classically-described pseudo-deterministic quantum functionalities.

At a high level, their approach was to encrypt the description of the circuit $Q \to \mathsf{Enc}(Q)$ using a blind quantum computation protocol (e.g. [Mah18a]) and prepare a classical oracle that is able to verify the result of computing $\mathsf{Enc}(Q) \to \mathsf{Enc}(Q(x))$, and decrypt the result if verification passes. But if the program $|Q\rangle$ is described quantumly, this approach completely breaks down, since the classical oracle cannot even *interpret* the *quantum* statement $\mathsf{Enc}(|Q\rangle) \to \mathsf{Enc}(Q(x))$ to be verified.

Our approach is to not only encrypt $|Q\rangle$ but to *authenticate* it as well. Then, we design oracles that are able to verify the blind computation *along the way*, as opposed to *all at once* at the end of the computation. That is, while both approaches make fundamental use of both blind and verifiable quantum computation, they differ significantly in execution.

Finally, it is worth mentioning that the [BKNY23] approach may prove to be advantageous in the case where we are only interested in obfuscating quantum circuits with a classical description, and desire to produce an obfuscated program that *also* has a classical description. Given the fact that quantum state obfuscation is a best-possible copy-protector, it is actually *inherent* that our obfuscated program includes (unclonable) quantum states, even when applied to a circuit with classical description. On the other hand, while the [BKNY23] approach as currently implemented also produces obfuscated programs with a quantum description, it is reasonable to hope that it could be de-quantized. Indeed, the only reason that [BKNY23]'s construction includes a quantum state is that two of their building blocks - signature tokens and Pauli functional commitments - are constructed using quantum keys. However, it is plausible that both of these building blocks could be instantiated with a classical key (e.g. using the ideas of [AGKZ20], though proving security may be difficult). We leave further exploration of these ideas to future work.

Open problems. We conclude this overview by mentioning a couple of natural open problems that are motivated by this work. Perhaps most obviously, can we obtain provable security in the plain model from quantum-secure indistinguishability obfuscation (or a different concrete and plausible assumption on classical obfuscators)? Besides representing major progress in our understanding of quantum obfuscation, answering this question would have significant implications for general-purpose software copy-protection (due to [CG23]), another area where positive results are difficult to come by.

Next, can we generalize beyond pseudo-deterministic programs? As a first step, can we obfuscate *sampling* circuits, i.e. those with classical inputs that produce a *distribution* over classical outputs? Interestingly, while [BKNY23]'s construction does not even satisfy correctness for (even classically-described) sampling circuits, our approach can plausibly be applied to sampling circuits, although it remains open to obtain any provable guarantees. Finally, the feasibility of obfuscating general-purpose circuits with quantum inputs and/or quantum outputs is still wide open, and remains a fascinating direction for future exploration, with potential applications beyond cryptography, e.g. to quantum complexity theory.

3 Preliminaries

Let λ denote the security parameter. We write $\mathrm{negl}(\cdot)$ to denote any *negligible* function, which is a function f such that for every constant $c \in \mathbb{N}$ there exists $N \in \mathbb{N}$ such that for all $n > N$,

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

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

denote the total variation distance. For a set S, we let $x \leftarrow S$ denote sampling a uniformly random element x from S. 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 that with probability $1-\delta$, sample from D_0 and with probability δ, sample from D_1.

3.1 Quantum Background

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

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

A family of quantum circuits is in general 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 apply a unitary that maps $|x\rangle\,|y\rangle \to |x\rangle\,|y \oplus \mathsf{F}(x)\rangle$. 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. For registers $\mathcal{X}, \mathcal{Y}$, the *partial trace* $\mathrm{Tr}^{\mathcal{Y}}$ is the unique operation from $\mathcal{X}, \mathcal{Y}$ to $\mathcal{X}$ such that for all $(\rho, \tau)^{\mathcal{X},\mathcal{Y}}$, $\mathrm{Tr}^{\mathcal{Y}}(\rho, \tau) = \mathrm{Tr}(\tau)\rho$. 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 3.1 (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 $\epsilon-$*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*

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

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 $\Pr[C(|x\rangle |\psi\rangle) \to y] \geq 1 - \epsilon(\lambda)$, and refer to Q as the map induced by $(|\psi\rangle, C)$*.*

3.2 Useful Lemmas

Lemma 3.2 (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 3.3 (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 3.4. *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*

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

Proof.

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

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

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

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

and the fourth is Jensen's inequality.

□

Lemma 3.5. *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*

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

Proof.

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

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 $\|\Gamma_k U^{O_k}\Pi_{k,i}\left|\psi_k\right\rangle\|$.

□

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 3.6. *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*

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

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 3.2, 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 3.2, it holds that $\mathsf{TD}(\mathcal{H}'_i, \mathcal{H}_{i+1}) \leq 2\sqrt{\epsilon}$.

Thus, the lemma follows by summing over the $2q$ hybrid switches.

□

3.3 Signature Tokens

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 3.7. *A signature token scheme* $(\mathsf{TokGen}, \mathsf{TokSign}, \mathsf{TokVer})$ *is* correct *if for any* $b \in \{0,1\}$,

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

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

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

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

Imported Theorem 3.9 ([BS16]). *There exists a signature token scheme that satisfies Definition 3.7 and Definition 3.8.*

4 Authentication Scheme

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.

4.1 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 4.1 (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 4.2 (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 4.3. *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 4.2, but will be convenient to use in our application to obfuscation.

Definition 4.4 (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 4.5 (Privacy). *For any n-qubit states $|\psi_0\rangle, |\psi_1\rangle$ and oracle-aided binary outcome projector D,*

$$\left| \Pr_{k \leftarrow \mathsf{Gen}(1^\lambda, n)} \left[1 \leftarrow D^{\mathsf{Ver}_{k,\cdot,\cdot}(\cdot)} \circ \mathsf{Enc}_k(|\psi_0\rangle) \right] - \Pr_{k \leftarrow \mathsf{Gen}(1^\lambda, n)} \left[1 \leftarrow D^{\mathsf{Ver}_{k,\cdot,\cdot}(\cdot)} \circ \mathsf{Enc}_k(|\psi_1\rangle) \right] \right| = 2^{-\Omega(\lambda)}.$$

4.2 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 $\widehat{S} \coloneqq S_\Delta^\perp \subset S^\perp$.

Let $\widehat{\Delta}$ be an arbitrary choice of a vector such that $S^\perp = \widehat{S} \cup (\widehat{S} + \widehat{\Delta})$, and define

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

Given a subspace S, define the state

$$|S\rangle \coloneqq \frac{1}{\sqrt{|S|}} \sum_{s\in S} |s\rangle \,,$$

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 $|1\rangle \to |S + \Delta\rangle$.

Theorem 4.6. *The QAS described in Figure 1 satisfies correctness (Definition 4.1).*

Proof. First, we show two key claims.

Claim 4.7. *For any S and Δ, it holds that $H^{\otimes 2\lambda+1} E_{\widehat{S},\widehat{\Delta}} = E_{S,\Delta} H$.*

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 $k \coloneqq (S, \Delta, x, z)$.
- $\mathsf{Enc}_k = X^x Z^z E_{S,\Delta}^{\otimes n}$.
- $\mathsf{LinEval}_L(\mathcal{C})$: 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 1: Our construction of a publicly-verifiable linearly-homomorphic QAS with classically-decodable ZX measurements.

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 4.8. *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 $\{|b_1, b_2\rangle\}_{b_1,b_2\in\{0,1\}}$.

$$\begin{aligned} &\mathsf{CNOT}^{\otimes 2\lambda+1} E_{S,\Delta}^{\otimes 2} |b_1, b_2\rangle \\ &= \mathsf{CNOT}^{\otimes 2\lambda+1} |S + b_1 \cdot \Delta\rangle |S + b_2 \cdot \Delta\rangle \\ &= \frac{1}{2^\lambda} \mathsf{CNOT}^{\otimes 2\lambda+1} \sum_{s_1 \in S} |s_1 + b_1 \cdot \Delta\rangle \sum_{s_2 \in S} |s_2 + b_2 \cdot \Delta\rangle \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2^\lambda} \sum_{s_1 \in S} |s_1 + b_1 \cdot \Delta\rangle \sum_{s_2 \in S} |(s_1 + s_2) + (b_1 + b_2) \cdot \Delta\rangle \\
&= |S + b_1 \cdot \Delta\rangle \, |S + (b_1 + b_2) \cdot \Delta\rangle \\
&= E_{S,\Delta}^{\otimes 2} \mathsf{CNOT} \, |b_1, b_2\rangle \, .
\end{aligned}
$$

□

Now, define measurements $M'_{\theta,f}, \widetilde{M}'_{\theta,f,k,L}$ so that

$$M_{\theta,f} = H^{\Phi_1} M'_{\theta,f} H^{\Phi_1}, \quad \text{and} \quad \widetilde{M}_{\theta,f,k,L} = H^{\widetilde{\Phi}_1} X^{x_{L,\Phi_0}, z_{L,\Phi_1}} \widetilde{M}'_{\theta,f,k,L} X^{x_{L,\Phi_0}, z_{L,\Phi_1}} H^{\widetilde{\Phi}_1}.$$

To be concrete,

$$M'_{\theta,f} \coloneqq \left\{ \sum_{m : f(m_\Phi) = y} |m\rangle\langle m| \right\}_y ,$$

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,

$$
\begin{aligned}
&\mathsf{LinEval}_L^\dagger \widetilde{M}_{\theta,f,k,L} \mathsf{LinEval}_L \mathsf{Enc}_k \\
&= \mathsf{LinEval}_L^\dagger \widetilde{M}_{\theta,f,k,L} \mathsf{LinEval}_L X^x Z^z E_{S,\Delta}^{\otimes n} \\
&= \mathsf{LinEval}_L^\dagger \widetilde{M}_{\theta,f,k,L} X^{x_L} Z^{z_L} \mathsf{LinEval}_L E_{S,\Delta}^{\otimes n} \\
&= \mathsf{LinEval}_L^\dagger \widetilde{M}_{\theta,f,k,L} X^{x_L} Z^{z_L} E_{S,\Delta}^{\otimes n} L && \text{(Claim 4.8)} \\
&= \mathsf{LinEval}_L^\dagger H^{\widetilde{\Phi}_1} X^{x_{L,\Phi_0} z_{L,\Phi_1}} \widetilde{M}'_{\theta,f,k,L} X^{x_{L,\Phi_0}, z_{L,\Phi_1}} H^{\widetilde{\Phi}_1} X^{x_L} Z^{z_L} E_{S,\Delta}^{\otimes n} L \\
&= \mathsf{LinEval}_L^\dagger H^{\widetilde{\Phi}_1} X^{x_{L,\Phi_0} z_{L,\Phi_1}} \widetilde{M}'_{\theta,f,k,L} X^{x_{L,\Phi_\perp}} Z^{z_{L,\Phi_\perp}, z_{L,\Phi_0}, x_{L,\Phi_1}} H^{\widetilde{\Phi}_1} E_{S,\Delta}^{\otimes n} L \\
&= \mathsf{LinEval}_L^\dagger H^{\widetilde{\Phi}_1} X^{x_{L,\Phi_\perp}, x_{L,\Phi_0}, z_{L,\Phi_1}} Z^{z_{L,\Phi_\perp}, z_{L,\Phi_0}, x_{L,\Phi_1}} \widetilde{M}'_{\theta,f,k,L} H^{\widetilde{\Phi}_1} E_{S,\Delta}^{\otimes n} L \\
&= \mathsf{LinEval}_L^\dagger X^{x_L} Z^{z_L} H^{\widetilde{\Phi}_1} \widetilde{M}'_{\theta,f,k,L} H^{\widetilde{\Phi}_1} E_{S,\Delta}^{\otimes n} L \\
&= \mathsf{LinEval}_L^\dagger X^{x_L} Z^{z_L} H^{\widetilde{\Phi}_1} \widetilde{M}'_{\theta,f,k,L} \left(E_{S,\Delta}^{\otimes |\Phi_\perp \cup \Phi_0|} \otimes E_{\widehat{S},\widehat{\Delta}}^{\otimes |\Phi_1|} \right) H^{\Phi_1} L && \text{(Claim 4.7)} \\
&= \mathsf{LinEval}_L^\dagger X^{x_L} Z^{z_L} H^{\widetilde{\Phi}_1} \left(E_{S,\Delta}^{\otimes |\Phi_\perp \cup \Phi_0|} \otimes E_{\widehat{S},\widehat{\Delta}}^{\otimes |\Phi_1|} \right) M'_{\theta,f} H^{\Phi_1} L \\
&= \mathsf{LinEval}_L^\dagger X^{x_L} Z^{z_L} E_{S,\Delta}^{\otimes n} H^{\Phi_1} M'_{\theta,f} H^{\Phi_1} L
\end{aligned}
$$

$$= X^x Z^z \mathsf{LinEval}_L^\dagger E_{S,\Delta}^{\otimes n} M_{\theta,f} L$$
$$= X^x Z^z E_{S,\Delta}^{\otimes n} L^\dagger M_{\theta,f} L$$
$$= \mathsf{Enc}_k L^\dagger M_{\theta,f} L.$$

□

4.3 Security

Theorem 4.9. *The QAS described in Figure 1 satisfies security (Definition 4.2).*

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 4.10. *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 3.6 (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 4.10 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 4.11. *Fix any $R, \widehat{R}, x^{\mathsf{co}(R)}, z^{\mathsf{co}(\widehat{R})}$. Then for any L, θ, f, and unitary A,[9] there exists an $\epsilon = \epsilon(\lambda) \in [0,1]$ such that for any $|\psi\rangle$,*

$$\left\{ y : \begin{array}{r} \left(S, \Delta, x^R, z^{\widehat{R}}\right) \leftarrow \mathsf{Gen}'_{R,\widehat{R}}(1^\lambda, n) \\ x := x^R + x^{\mathsf{co}(R)}, z := z^{\widehat{R}} + z^{\mathsf{co}(\widehat{R})} \\ k := (S, \Delta, x, z) \\ y \leftarrow \widetilde{M}_{\theta,f,k,L} \circ A \circ \mathsf{Enc}_k(|\psi\rangle) \end{array} \right\} \approx_{2^{-\Omega(\lambda)}} (1-\epsilon)\{y : y \leftarrow M_{\theta,f} \circ L(|\psi\rangle)\} + \epsilon\{\perp\}.$$

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

$$\left(S, \Delta, x^R, z^{\widehat{R}}\right) \leftarrow \mathsf{Gen}'_{R,\widehat{R}}(1^\lambda, n)$$

[9] As noted in Remark 4.3, 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.

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 3.3).

Next, we introduce some more notation.

- For each $y \in \mathsf{range}(f)$, define

$$V_y := \bigcup_{m:f(m_\Phi)=y} \left(\bigotimes_{i\in\Phi_0} (S + m_i \cdot \Delta) \bigotimes_{i\in\Phi_1} \left(\widehat{S} + m_i \cdot \widehat{\Delta}\right) \right),$$

  and define

$$V_\perp := \{0,1\}^{(2\lambda+1)n} \setminus \bigcup_{y\in\mathsf{range}(f)} V_y.$$

  For $y \in \mathsf{range}(f) \cup \{\perp\}$, define $|V_y\rangle := \sum_{v\in V_y} |v\rangle$.

- Define the unitary $B := A \circ \mathsf{LinEval}_L^\dagger$. 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 $\mathsf{Mx}[|\phi\rangle] := |\phi\rangle\langle\phi|$.

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

$$\mathcal{P} := \left\{ X^x Z^z : x = (x_1, \ldots, x_n), z = (z_1, \ldots, z_n) \in \{0,1\}^{(2\lambda+1)n} \right\},$$

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 $y \in \mathsf{range}(f) \cup \{\perp\}$ 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

$$\begin{aligned}
&\sum_{y \in \mathsf{range}(f) \cup \{\perp\}} |y\rangle\langle y| \frac{1}{|\mathcal{K}_L|} \sum_{(S,\Delta,x_L,z_L)} \langle V_y| \mathsf{Mx}\left[H^{\widetilde{\Phi}_1} Z^{z_L} X^{x_L} \left(\sum_{P \in \mathcal{P}_{C,\widehat{C}}} \alpha_P P\right) X^{x_L} Z^{z_L} E_{S,\Delta}^{\otimes n} L\,|\psi\rangle\right] |V_y\rangle \\
&\quad + \epsilon_\perp\, |\perp\rangle\langle\perp| \\
&= \sum_{y \in \mathsf{range}(f) \cup \{\perp\}} |y\rangle\langle y| \sum_{P_0, P_1 \in \mathcal{P}_{C,\widehat{C}}} \alpha_{P_0} \alpha_{P_1}^* \frac{1}{|\mathcal{K}_L|} \\
&\qquad \left(\sum_{(S,\Delta,x_L,z_L)} \langle V_y|\, H^{\widetilde{\Phi}_1} Z^{z_L} X^{x_L} P_0 X^{x_L} Z^{z_L} \mathsf{Mx}\left[E_{S,\Delta}^{\otimes n} L\,|\psi\rangle\right] Z^{z_L} X^{x_L} P_1^\dagger X^{x_L} Z^{z_L} H^{\widetilde{\Phi}_1} |V_y\rangle\right) \\
&\quad + \epsilon_\perp\, |\perp\rangle\langle\perp|
\end{aligned}$$

Now, we are finally ready to apply the the Pauli twirl over affine subspaces (Lemma 3.3). 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 3.3, all the cross-terms $P_0 \neq P_1$ are killed in the above expression for $\mathcal{D}$, which we can now write as

$$\begin{aligned}
&\sum_{y \in \mathsf{range}(f) \cup \{\perp\}} |y\rangle\langle y| \sum_{P \in \mathcal{P}_{C,\widehat{C}}} \alpha_P \alpha_P^* \frac{1}{|\mathcal{K}_L|} \left(\sum_{(S,\Delta,x_L,z_L)} \langle V_y| \mathsf{Mx}\left[H^{\widetilde{\Phi}_1} Z^{z_L} X^{x_L} P X^{x_L} Z^{z_L} E_{S,\Delta}^{\otimes n} L\,|\psi\rangle\right] |V_y\rangle\right) \\
&\quad + \epsilon_\perp\, |\perp\rangle\langle\perp|\,,
\end{aligned}$$

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 $P \in \mathcal{P}_{C,\widehat{C}} \setminus \mathcal{I}$,

$$\begin{aligned}
&\sum_{y \in \mathsf{range}(f) \cup \{\perp\}} \frac{1}{|\mathcal{K}_L|} \sum_{(S,\Delta,x_L,z_L)} \langle V_y| \mathsf{Mx}\left[H^{\widetilde{\Phi}_1} Z^{z_L} X^{x_L} P X^{x_L} Z^{z_L} E_{S,\Delta}^{\otimes n} L\,|\psi\rangle\right] |V_y\rangle \\
&\leq \frac{|S_\Delta \setminus \widehat{R}_\perp|}{|R \setminus \widehat{R}_\perp|} + \frac{|\widehat{S}_{\widehat{\Delta}} \setminus R_\perp|}{|\widehat{R} \setminus R_\perp|} = 2 \cdot \frac{2^{\lambda+1} - 2^\lambda}{2^{3\lambda/2+1} - 2^\lambda} = 2^{-\Omega(\lambda)}.
\end{aligned}$$

Thus, $\mathcal{D}$ is within $2^{-\Omega(\lambda)}$ total variation distance of

$$
\begin{aligned}
&(1-\epsilon_\perp)\sum_y |y\rangle\langle y| \frac{1}{|\mathcal{K}_L|}\sum_{(S,\Delta,x_L,z_L)} \langle V_y|\,\mathsf{Mx}\left[H^{\widetilde{\Phi}_1} Z^{z_L} X^{x_L} \mathcal{I} X^{x_L} Z^{z_L} E_{S,\Delta}^{\otimes n} L\,|\psi\rangle\right]|V_y\rangle + \epsilon_\perp\,|\perp\rangle\langle\perp| \\
&= (1-\epsilon_\perp)\sum_y |y\rangle\langle y| \frac{1}{|\mathcal{K}_L|}\sum_{(S,\Delta,x_L,z_L)} \langle V_y|\,\mathsf{Mx}\left[H^{\widetilde{\Phi}_1} E_{S,\Delta}^{\otimes n} L\,|\psi\rangle\right]|V_y\rangle + \epsilon_\perp\,|\perp\rangle\langle\perp| \\
&= (1-\epsilon_\perp)\sum_y |y\rangle\langle y| \left(\sum_{m:f(m)=y}\langle m|\right)\mathsf{Mx}\left[H^{\Phi_1} L\,|\psi\rangle\right]\left(\sum_{m:f(m)=y}|m\rangle\right) + \epsilon_\perp\,|\perp\rangle\langle\perp| \\
&= (1-\epsilon_\perp)\{y : y \leftarrow M_{\theta,f}\circ L(|\psi\rangle)\} + \epsilon_\perp\{\perp\},
\end{aligned}
$$

which completes the proof.

□

□

Theorem 4.12. *The QAS described in Figure 1 satisfies privacy (Definition 4.5).*

Proof. First, recalling the definitions of $\mathsf{Gen}', \mathsf{Ver}'$ in the proof of Theorem 4.9, and applying the oracle indistinguishability argued during the proof of Claim 4.10, 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 Linear + Measurement Quantum Programs

In this section, we show that any quantum program with classical input and output (Definition 3.1) 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 4.1. 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.

We first formally define LM quantum programs, and accompany this with a diagram in Fig. 2.

Definition 5.1 (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 4.1, 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* $\mathsf{LMEval}\left(x, |\psi\rangle, \{L_i\}_{i\in[t+1]}, \{\theta_i\}_{i\in[t+1]}, \{f_i\}_{i\in[t]}, g\right) \to y$ *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. 2 is indeed an example of such an LM quantum program.

Definition 5.2 (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*.*

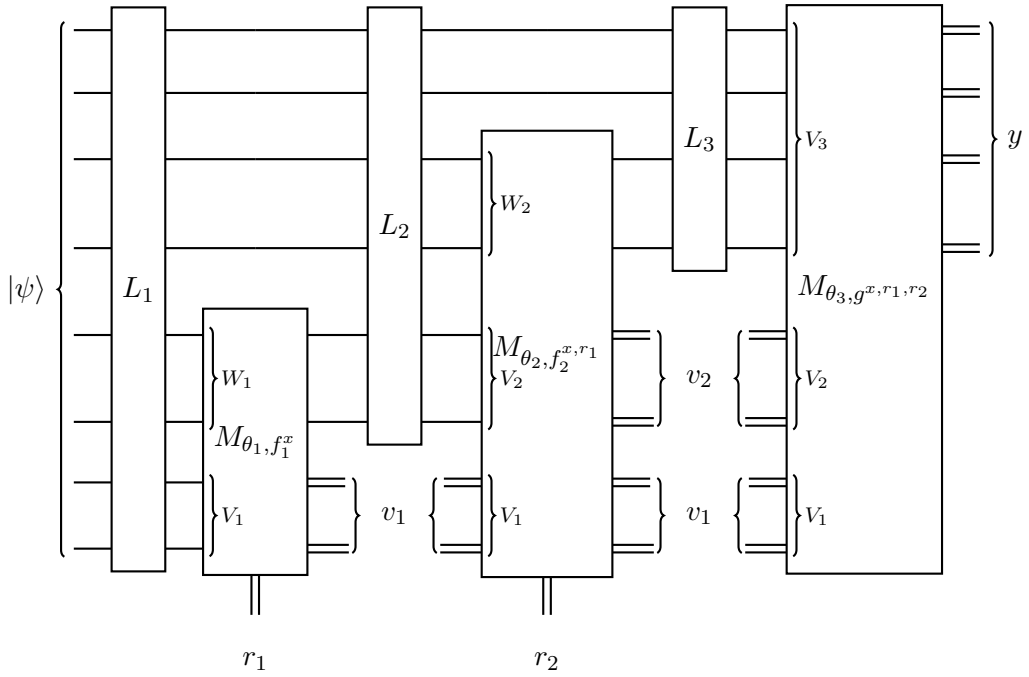

Figure 2: Diagram of an LM quantum program. We use some non-standard quantum circuit notation, so we provide some explanation. Each partial ZX measurement $M_{\theta_1, f_1^{(\cdot)}}, M_{\theta_2, f_2^{(\cdot)}}, M_{\theta_3, g^{(\cdot)}}$ is applied to the wires coming from the left of the corresponding 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.

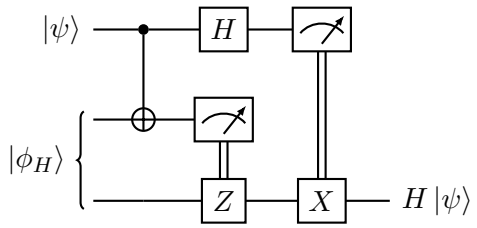

Figure 3: Implementation of the H gate with the H-magic state $|\phi_H\rangle \propto |00\rangle + |01\rangle + |10\rangle - |11\rangle$.

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 7.1).

Theorem 5.3. *Any quantum program $(|\psi\rangle, C)$ (Definition 3.1) 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 $(|\psi\rangle, C)$.*

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 3. For completeness, we show that the circuit indeed implements the Hadamard gate.

Claim 5.4. *The circuit in Figure 3 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 4. 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 5.5. *The bottom right circuit in Figure 4 implements the T gate.*

Proof. Write $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$. Applying the first CNOT yields

$$\begin{aligned}
&\alpha\,|00\rangle + e^{i\pi/4}\beta\,|01\rangle + \beta\,|10\rangle + e^{i\pi/4}\alpha\,|11\rangle \\
&= |0\rangle\,(\alpha\,|0\rangle + e^{i\pi/4}\beta\,|1\rangle) + |1\rangle\,(\beta\,|0\rangle + e^{i\pi/4}\alpha\,|1\rangle).
\end{aligned}$$

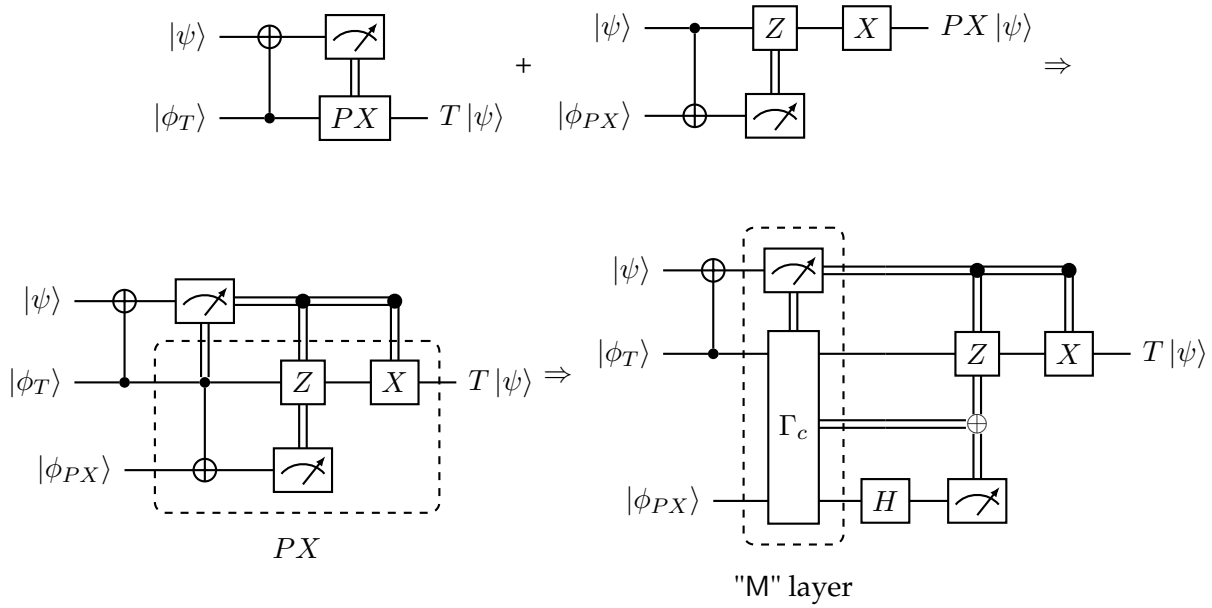

Figure 4: Implementation of the T gate. First, we combine an implementation of the T gate using the T-magic state $|\phi_T\rangle \propto |0\rangle + e^{i\pi/4}|1\rangle$ (upper left) with an implementation of the PX gate using the PX-magic state $|\phi_{PX}\rangle \propto i|0\rangle + |1\rangle$ (upper 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 5.6 (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 5.1), 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 5.7. *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 5.1).*

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 5.4 and Claim 5.5 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. 3.
  - 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. 4.
  - 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 5.2) are fulfilled.

□

6 Quantum State Obfuscation: Construction

In this section, we define quantum state obfuscation, describe our construction, and show that it is correct. We will prove security in the following section.

Definition 6.1 (Quantum State Obfuscation)**.** *For any $\epsilon = \epsilon(\lambda)$, let $\mathcal{C}_\epsilon$ be the set of families of ϵ-pseudo-deterministic quantum programs (Definition 3.1), 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 $(\mathsf{QObf}, \mathsf{QEval})$ with the following syntax.*

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

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

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

We define two notions of security with respect to some pseudo-deterministic parameter $\epsilon = \epsilon(\lambda)$.

- ***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 $\{D_\lambda\}_{\lambda\in\mathbb{N}}$,*

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

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

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

Remark 6.2 (Classical Oracle Model)**.** *In this work, we construct quantum state obfusation in the* classical oracle model*. In this model, we allow* QObf *to additionally output the description of a classical deterministic functionality O, and both* QEval *and the adversary A_λ 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 state $|\widetilde{\psi}\rangle$.*

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 4.
- Signature token $(\mathsf{TokGen}, \mathsf{TokSign}, \mathsf{TokVer})$, defined in Section 3.3.
- 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

$$\left\{ \left|\psi^{\mathsf{unv}}_{\lambda,n,m,m'}\right\rangle, C^{\mathsf{unv}}_{\lambda,n,m,m'} \right\}_{\lambda\in\mathbb{N}}$$

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,

$$C^{\mathsf{unv}}_{\lambda,n,m,m'}\left(|x\rangle |C_\lambda\rangle |\phi_\lambda\rangle \left|\psi^{\mathsf{unv}}_{\lambda,n,m,m'}\right\rangle\right) = C_\lambda\left(|x\rangle |\phi_\lambda\rangle\right).$$

By Theorem 5.3, for any (n, m, m') family $\{|\phi_\lambda\rangle, C_\lambda\}_{\lambda\in\mathbb{N}}$, we can write each quantum program

$$|C_\lambda\rangle |\phi_\lambda\rangle \left|\psi^{\mathsf{unv}}_{\lambda,n,m,m'}\right\rangle, C^{\mathsf{unv}}_{\lambda,n,m,m'}$$

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 5.2), where we have dropped the indexing by λ to reduce notational clutter. Note that Theorem 5.3 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 5.3.

The construction is given in Figure 5, 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 4.1 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 6.3. *The scheme described in Fig. 5 is a quantum state obfuscator that satisfies correctness (Definition 6.1).*

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

$$\Pr\left[y^* = y : \begin{array}{c} |\widetilde{\psi}\rangle, O \leftarrow \mathsf{QObf}(1^\lambda, |\psi\rangle) \\ y^* \leftarrow \mathsf{QEval}^O\left(x, |\widetilde{\psi}\rangle\right) \end{array}\right] = 1 - \mathrm{negl}(\lambda).$$

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 $\perp$.

Quantum State Obfuscation

$\mathsf{QObf}\left(1^\lambda, |\psi\rangle\right)$:

- 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 $\mathsf{TokVer}(\mathsf{vk}, x, \sigma_x) = \perp$.
  - 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 $\mathsf{TokVer}(\mathsf{vk}, x, \sigma_x) = \perp$.
  - 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 $|\widetilde{\psi}\rangle = |\psi_k\rangle |\mathsf{sk}\rangle, O = (\mathsf{F}_1, \ldots, \mathsf{F}_t, \mathsf{G})$.

$\mathsf{QEval}^O\left(x, |\widetilde{\psi}\rangle\right)$:

- Parse $|\widetilde{\psi}\rangle = |\psi_k\rangle |\mathsf{sk}\rangle, O = (\mathsf{F}_1, \ldots, \mathsf{F}_t, \mathsf{G})$.
- 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 5: Construction of quantum state obfuscation.

  - 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 $\bot$ if the result is $\bot$.
- Compute $(\cdot, r_i) = f_i^{x, r_1, \ldots, r_{i-1}}(v_1, \ldots, v_i, w_i)$.
- 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 $\bot$ if the result is $\bot$.
- 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 3.7).
- $\mathcal{H}_2 \approx_{\mathrm{negl}(\lambda)} \mathcal{H}_3$: 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 4.1). 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 4.1) 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 $((v_2, r_2), \mathcal{C}) \leftarrow \widetilde{M}_{\theta_2, f_2^{x, r_1}, k, L_2 L_1} \circ \mathsf{LinEval}_{L_2 L_1}(\mathcal{C})$.

– ...
– 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 $((v_t, r_t), \mathcal{M}) \leftarrow L_1^\dagger \ldots L_t^\dagger \circ M_{\theta_t, f_t^{x,r_1,\ldots,r_{t-1}}} \circ L_t \ldots L_1(\mathcal{M})$.
– Compute output $y \leftarrow M_{\theta_{t+1}, g^{x,r_1,\ldots,r_t}} \circ L_{t+1} \ldots L_1(\mathcal{M})$.

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 $((v_t, r_t), \mathcal{M}) \leftarrow M_{\theta_t, f_t^{x,r_1,\ldots,r_{t-1}}} \circ L_t(\mathcal{M})$.
– Compute output $y \leftarrow M_{\theta_{t+1}, g^{x,r_1,\ldots,r_t}} \circ L_{t+1}(\mathcal{M})$.

□

7 Quantum State Obfuscation: Security

In this section, we prove that the scheme described in Fig. 5 is an ideal quantum state obfuscator in the classical oracle model. We provide some intuition in Section 7.1 and set up some notation in Section 7.2 before coming to the formal proof in Section 7.3 - Section 7.5.

7.1 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 $\mathsf{F}_1, \ldots, \mathsf{F}_t, \mathsf{G}$ 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 4.5) 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 2.3), 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 $\mathsf{F}_1, \ldots, \mathsf{F}_t, \mathsf{G}$, the adversary cannot prepare a "bad" query $(x^*, \sigma_{x^*}, \widetilde{v}_1, \ldots, \widetilde{v}_{t+1}, \ell_1, \ldots, \ell_t)$ with the property that

$$\mathsf{G}(x^*, \sigma_{x^*}, \widetilde{v}_1, \ldots, \widetilde{v}_{t+1}, \ell_1, \ldots, \ell_t) \notin \{Q(x^*), \bot\}.$$

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 $|\widetilde{\psi}[x^*, v_1, r_1]\rangle := \mathsf{Enc}_k(\Pi[x^*, v_1, r_1] \, |\psi\rangle)$ 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 7.4.

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 5.2). 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 $|\widetilde{\psi}[w]\rangle \coloneqq \mathsf{Enc}_k(\Pi[w] \, |\psi\rangle)$, 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 7.3, and also in lower-level hybrids in Section 7.5.

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 2.3, 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 2.3, 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 $|{+}{+}\rangle$:

$$\mathbb{E}\left[\left\| \left(\Pi[S + \Delta + x] \otimes |{+}{+}\rangle\langle{+}{+}|\right) A^O \left(X^x Z^z |S\rangle\right) |{+}{+}\rangle \right\|^2\right] = \text{negl}(\lambda).$$

  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,

$$X^x Z^z |S\rangle = \frac{1}{\sqrt{2}} H^{\otimes 2\lambda+1} X^z Z^x |\widehat{S}\rangle + \frac{1}{\sqrt{2}} H^{\otimes 2\lambda+1} X^z Z^x |\widehat{S} + \widehat{\Delta}\rangle \coloneqq |\widetilde{+}\rangle + |\widetilde{-}\rangle .$$

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 7.5, and in particular in the proof of Lemma 7.25.

7.2 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 5.1 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 5.2) 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 4.1 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 4.1 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

$$\Pi[\{w_i^*\}_{i\in S}] := \bigotimes_{i\in S} \Pi[w_i^*], \quad \Pi[\neg\{w_i^*\}_{i\in S}] := \sum_{i\in S} \Pi[\neg w_i^*].$$

7.3 Main Theorem

Theorem 7.1. *For any $\epsilon = \epsilon(\lambda) = \mathrm{negl}(\lambda) \cdot 2^{-2m(\lambda)}$, the scheme described in Figure 5 is a quantum state obfuscator that satisfies ideal obfuscation (Definition 6.1) for ϵ-pseudo-deterministic families of quantum programs.*

Remark 7.2. *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 [CG23] 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[10] adversary A and distinguisher D, and define $D[A]$ to be the procedure that runs A, feeds its output to D, and then runs D to produce a binary-valued outcome. Thus, we can write the "real" obfuscation experiment as

$$\Pr\left[1 \leftarrow D\left(A\left(\mathsf{QObf}\left(1^\lambda, |\psi\rangle\right)\right)\right)\right] = \mathop{\mathbb{E}}_{(|\widetilde{\psi}\rangle, O)\leftarrow \mathsf{QObf}(1^\lambda,|\psi\rangle)}\left[\left\| D[A]^O |\widetilde{\psi}\rangle \right\|^2\right].$$

Now, we will consider a sequence of hybrid distributions over (state, oracle) pairs $(|\widetilde{\psi}\rangle, O)$, beginning with the real distribution $\mathsf{QObf}_0 := \mathsf{QObf}$, and ending with a fully simulated distribution QObf_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 7.1, 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 $w_i^* \in \{0,1\}^{|W_i|}$.

$\mathsf{QObf}_1(1^\lambda, |\psi\rangle)$:

- 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 QObf_0.
- Define G as in QObf_0.
- Output $|\widetilde{\psi}\rangle = |\psi_k\rangle\,|\mathsf{sk}\rangle, O = (\mathsf{F}_1, \ldots, \mathsf{F}_t, \mathsf{G})$.

$\mathsf{QObf}_2(1^\lambda, |\psi\rangle)$:

- 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 QObf_0.
- Define the function GSim $(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_{t+1}, \ell_1, \ldots, \ell_t)$:

[10] 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 adversary.

- 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 $|\widetilde{\psi}\rangle = |\psi_k\rangle |\mathsf{sk}\rangle, O = (\mathsf{F}_1, \ldots, \mathsf{F}_t, \mathsf{GSim})$.

$\mathsf{QObf}_3(1^\lambda, |\psi\rangle)$:

- 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 $\ell_i \coloneqq H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \ell_1, \ldots, \ell_{i-1}, r_i)$, and output $(\widetilde{v}_i, \ell_i)$.
- Define GSim as in QObf_2.
- Output $|\widetilde{\psi}\rangle = |\psi_k\rangle |\mathsf{sk}\rangle, O = (\mathsf{F}_1[\cdot], \ldots, \mathsf{F}_t[\cdot], \mathsf{GSim})$.

$\mathsf{QObf}_4(1^\lambda, |\psi\rangle)$:

- 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 $\ell_i \coloneqq H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \ell_1, \ldots, \ell_{i-1}, r_i)$, and output $(\widetilde{v}_i, \ell_i)$.
- Define GSim as in QObf_2.
- Output $|\widetilde{\psi}\rangle = |\psi_k\rangle |\mathsf{sk}\rangle, O = (\mathsf{F}_1[\cdot], \ldots, \mathsf{F}_t[\cdot], \mathsf{GSim})$.

$\mathsf{QObf}_5(1^\lambda, |\psi\rangle)$:

- 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 $\bot$ if $\mathsf{TokVer}(\mathsf{vk}, x, \sigma_x) = \bot$.
  - For each $\iota \in [i-1]$, let
$$\ell_{\iota,0} = H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 0), \quad \ell_{\iota,1} = H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_\iota, \ell_1, \ldots, \ell_{\iota-1}, 1),$$
    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 $\ell_i \coloneqq H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \ell_1, \ldots, \ell_{i-1}, 0)$, and output $(\widetilde{v}_i, \ell_i)$.
- Define GSim as in QObf_2.
- Output $|\widetilde{\psi}\rangle = |\psi_k\rangle\,|\mathsf{sk}\rangle\,, O = (\mathsf{FSim}_1, \ldots, \mathsf{FSim}_t, \mathsf{GSim})$.

$\mathsf{QObf}_6(1^\lambda)$:

- 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 QObf_4.
- Define GSim as in QObf_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* QObf_2, where the F_i oracles are not yet simulated.

Claim 7.3. *For any QPQ unitary U and any $w^* = \{w_i^*\}_{i\in[t]}$, it holds that*

$$\mathbb{E}\left[\left\|\Pi[\neg w^*]U^O\Pi[w^*]\,|\widetilde{\psi}\rangle\right\|^2 : |\widetilde{\psi}\rangle\,, O \leftarrow \mathsf{QObf}_5(1^\lambda, |\psi\rangle)\right] = 2^{-\Omega(\kappa)}.$$

Proof. The key point is that in QObf_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 4.9), and in particular that it satisfies Definition 4.4 (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 $\perp$ if the result is $\perp$.
- 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.

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 $\mathsf{QObf}_3, \mathsf{QObf}_4$, and QObf_5 in the case where the W wires of the input state have been collapsed to outcome w^*.

Claim 7.4. *For any (unbounded) distinguisher D and any $w^* = \{w_i^*\}_{i \in [t]}$, it holds that*

$$\begin{aligned}
&\mathbb{E}\left[\left\| D^{\mathsf{F}_1[w^*], \ldots, \mathsf{GSim}}\left(k, \Pi[w^*]\left|\widetilde{\psi}\right\rangle\right)\right\|^2 : \left|\widetilde{\psi}\right\rangle, (\mathsf{F}_1[\cdot], \ldots, \mathsf{F}_t[\cdot], \mathsf{GSim}) \leftarrow \mathsf{QObf}_4(1^\lambda, |\psi\rangle)\right] \\
&= \mathbb{E}\left[\left\| D^{\mathsf{FSim}_1, \ldots, \mathsf{GSim}}\left(k, \Pi[w^*]\left|\widetilde{\psi}\right\rangle\right)\right\|^2 : |\psi\rangle, (\mathsf{FSim}_1, \ldots, \mathsf{FSim}_t, \mathsf{GSim}) \leftarrow \mathsf{QObf}_5(1^\lambda, |\psi\rangle)\right],
\end{aligned}$$

where D's input includes the key k sampled by $\mathsf{QObf}_4, \mathsf{QObf}_5$.

Proof. In QObf_4, we have that

$$\mathsf{F}_i[w^*](x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1}) \in \{H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \ell_1, \ldots, \ell_{i-1}, R[k, w^*](\widetilde{v}_1, \ldots, \widetilde{v}_i)), \perp\},$$

while in QObf_5, we have that

$$\mathsf{FSim}_i[w^*](x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1}) \in \{H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \ell_1, \ldots, \ell_{i-1}, 0), \perp\}.$$

Both implementations of the oracles will always output $\perp$ 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-$\perp$ 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 7.5. *For any QPQ distinguisher D and any $w^* = \{w_i^*\}_{i \in [t]}$, it holds that*

$$\begin{aligned}
&\Bigg| \mathbb{E}\left[\left\| D^{\mathsf{F}_1[w^*], \ldots, \mathsf{GSim}}\left(k, \Pi[w^*]\left|\widetilde{\psi}\right\rangle\right)\right\|^2 : \left|\widetilde{\psi}\right\rangle, (\mathsf{F}_1[\cdot], \ldots, \mathsf{F}_t[\cdot], \mathsf{GSim}) \leftarrow \mathsf{QObf}_3(1^\lambda, |\psi\rangle)\right] \\
&- \mathbb{E}\left[\left\| D^{\mathsf{F}_1[w^*], \ldots, \mathsf{GSim}}\left(k, \Pi[w^*]\left|\widetilde{\psi}\right\rangle\right)\right\|^2 : \left|\widetilde{\psi}\right\rangle, (\mathsf{F}_1[\cdot], \ldots, \mathsf{F}_t[\cdot], \mathsf{GSim}) \leftarrow \mathsf{QObf}_4(1^\lambda, |\psi\rangle)\right] \Bigg| = 2^{-\Omega(\kappa)},
\end{aligned}$$

where D's input includes the key k sampled by $\mathsf{QObf}_3, \mathsf{QObf}_4$.

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 QObf_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 3.6 (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 QObf_5 and the indistinguishability of QObf_3 and QObf_5 - to show the indistinguishability of QObf_2 and QObf_3 (in the case where the W wires are collapsed to some w^*).

Claim 7.6. *For any QPQ distinguisher D and any $w^* = \{w_i^*\}_{i\in[t]}$, it holds that*

$$\Big| \mathbb{E}\left[\left\| D^{\mathsf{F}_1,\ldots,\mathsf{GSim}}\left(k, \Pi[w^*]\left|\widetilde{\psi}\right\rangle\right) \right\|^2 : \left|\widetilde{\psi}\right\rangle, (\mathsf{F}_1, \ldots, \mathsf{F}_t, \mathsf{GSim}) \leftarrow \mathsf{QObf}_2(1^\lambda, |\psi\rangle) \right]$$
$$- \mathbb{E}\left[\left\| D^{\mathsf{F}_1[w^*],\ldots,\mathsf{GSim}}\left(k, \Pi[w^*]\left|\widetilde{\psi}\right\rangle\right) \right\|^2 : \left|\widetilde{\psi}\right\rangle, (\mathsf{F}_1[\cdot], \ldots, \mathsf{F}_t[\cdot], \mathsf{GSim}) \leftarrow \mathsf{QObf}_3(1^\lambda, |\psi\rangle) \right] \Big| = 2^{-\Omega(\kappa)},$$

where D's input includes the key k sampled by $\mathsf{QObf}_2, \mathsf{QObf}_3$.

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 QObf_3 except with probability $2^{-\Omega(\kappa)}$. The claim follows by applying Lemma 3.6 (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 QObf_2.

Corollary 7.7. *For any QPQ unitary U, and any $w^* = \{w_i^*\}_{i\in[t]}$, it holds that*

$$\mathbb{E}\left[\left\| \Pi[\neg w_i^*] U^O \Pi[w^*] \left|\widetilde{\psi}\right\rangle \right\|^2 : \left|\widetilde{\psi}\right\rangle, O \leftarrow \mathsf{QObf}_2(1^\lambda, |\psi\rangle) \right] = 2^{-\Omega(\kappa).}$$

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:

$$\mathcal{H}_0 = \mathbb{E}\left[\left\| D[A]^{\mathsf{F}_1,\ldots,\mathsf{G}}\left|\widetilde{\psi}\right\rangle \right\|^2 : \left|\widetilde{\psi}\right\rangle, (\mathsf{F}_1, \ldots, \mathsf{F}_t, \mathsf{G}) \leftarrow \mathsf{QObf}_0(1^\lambda, |\psi\rangle) \right]$$

- Replace PRF with random oracle:

$$\mathcal{H}_1 = \mathbb{E}\left[\left\| D[A]^{\mathsf{F}_1,\ldots,\mathsf{G}}\left|\widetilde{\psi}\right\rangle \right\|^2 : \left|\widetilde{\psi}\right\rangle, (\mathsf{F}_1, \ldots, \mathsf{F}_t, \mathsf{G}) \leftarrow \mathsf{QObf}_1(1^\lambda, |\psi\rangle) \right]$$

- Simulate the G oracle:

$$\mathcal{H}_2 = \mathbb{E}\left[\left\|D[A]^{\mathsf{F}_1,\ldots,\mathsf{GSim}}\left|\widetilde{\psi}\right\rangle\right\|^2 : \left|\widetilde{\psi}\right\rangle, (\mathsf{F}_1,\ldots,\mathsf{F}_t,\mathsf{GSim}) \leftarrow \mathsf{QObf}_2(1^\lambda,|\psi\rangle)\right]$$

- Split $\left|\widetilde{\psi}\right\rangle$ into orthogonal components:

$$\mathcal{H}_3 = \mathbb{E}\left[\sum_{w^*}\left\|D[A]^{\mathsf{F}_1,\ldots,\mathsf{GSim}}\Pi[w^*]\left|\widetilde{\psi}\right\rangle\right\|^2 : \left|\widetilde{\psi}\right\rangle, (\mathsf{F}_1,\ldots,\mathsf{F}_t,\mathsf{GSim}) \leftarrow \mathsf{QObf}_2(1^\lambda,|\psi\rangle)\right].$$

- Hard-code the w^* measurement results:

$$\mathcal{H}_4 = \mathbb{E}\left[\sum_{w^*}\left\|D[A]^{\mathsf{F}_1[w^*],\ldots,\mathsf{GSim}}\Pi[w^*]\left|\widetilde{\psi}\right\rangle\right\|^2 : \left|\widetilde{\psi}\right\rangle, (\mathsf{F}_1[\cdot],\ldots,\mathsf{F}_t[\cdot],\mathsf{GSim}) \leftarrow \mathsf{QObf}_3(1^\lambda,|\psi\rangle)\right].$$

- Simulate the $\mathsf{F}_1,\ldots,\mathsf{F}_t$ oracles:

$$\mathcal{H}_5 = \mathbb{E}\left[\sum_{w^*}\left\|D[A]^{\mathsf{FSim}_1,\ldots,\mathsf{GSim}}\Pi[w^*]\left|\widetilde{\psi}\right\rangle\right\|^2 : \left|\widetilde{\psi}\right\rangle, (\mathsf{FSim}_1,\ldots,\mathsf{FSim}_t,\mathsf{GSim}) \leftarrow \mathsf{QObf}_5(1^\lambda,|\psi\rangle)\right].$$

- Put $\left|\widetilde{\psi}\right\rangle$ back together:

$$\mathcal{H}_6 = \mathbb{E}\left[\left\|D[A]^{\mathsf{FSim}_1,\ldots,\mathsf{GSim}}\left|\widetilde{\psi}\right\rangle\right\|^2 : \left|\widetilde{\psi}\right\rangle, (\mathsf{FSim}_1,\ldots,\mathsf{FSim}_t,\mathsf{GSim}) \leftarrow \mathsf{QObf}_5(1^\lambda,|\psi\rangle)\right].$$

- Simulate the state:

$$\mathcal{H}_7 = \mathbb{E}\left[\left\|D[A]^{\mathsf{FSim}_1,\ldots,\mathsf{GSim}}\left|\widetilde{\psi}\right\rangle\right\|^2 : \left|\widetilde{\psi}\right\rangle, (\mathsf{FSim}_1,\ldots,\mathsf{FSim}_t,\mathsf{GSim}) \leftarrow \mathsf{QObf}_6(1^\lambda)\right].$$

Now, we are ready to define the simulator $\mathsf{Sim}^Q(1^\lambda,n,m,m')$ for our obfuscation scheme:

- Sample $\left|\widetilde{\psi}\right\rangle, (\mathsf{FSim}_1,\ldots,\mathsf{FSim}_t,\mathsf{GSim}) \leftarrow \mathsf{QObf}_6(1^\lambda)$.
- 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 7.8. $|\mathcal{H}_0 - \mathcal{H}_1| = \mathrm{negl}(\lambda)$.

Proof. This follows directly from the security of the PRF against quantum superposition-query attacks. □

Claim 7.9. $|\mathcal{H}_1 - \mathcal{H}_2| = \mathrm{negl}(\lambda)$.

Proof. Suppose that we sample $|\widetilde{\psi}\rangle, (\mathsf{F}_1, \ldots, \mathsf{F}_t, \mathsf{G}) \leftarrow \mathsf{QObf}_1(1^\lambda, |\psi\rangle)$, and then define the set

$$B = \{(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_{t+1}, \ell_1, \ldots, \ell_t) : \mathsf{G}(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_{t+1}, \ell_1, \ldots, \ell_t) \notin \{Q(x), \bot\}\}.$$

Observe that the only difference between QObf_1 and QObf_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 3.6 (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

$$\mathbb{E}\left[\left\|\Pi[B]U^{\mathsf{F}_1,\ldots,\mathsf{GSim}}|\widetilde{\psi}\rangle\right\|^2 : |\widetilde{\psi}\rangle, (\mathsf{F}_1, \ldots, \mathsf{F}_t, \mathsf{GSim}) \leftarrow \mathsf{QObf}_2(1^\lambda, |\psi\rangle)\right] = \delta(\lambda)$$

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

$$\mathbb{E}\left[\left\|\Pi\left[B[x^*]\right]U^{\mathsf{F}_1,\ldots,\mathsf{F}_t,\mathsf{GSim}}|\widetilde{\psi}\rangle\right\|^2 : |\widetilde{\psi}\rangle, (\mathsf{F}_1, \ldots, \mathsf{F}_t, \mathsf{GSim}) \leftarrow \mathsf{QObf}_2(1^\lambda, |\psi\rangle)\right] \geq \delta(\lambda) \cdot 2^{-m}.$$

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

$$|\psi'_{x^*}\rangle \coloneqq \mathsf{Coh\text{-}LMEval}[x^*]^\dagger |Q(x^*)\rangle\langle Q(x^*)| \mathsf{Coh\text{-}LMEval}[x^*] |\psi\rangle, \quad |\psi_{x^*}\rangle \coloneqq \frac{|\psi'_{x^*}\rangle}{\| |\psi'_{x^*}\rangle \|}.$$

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 3.2), there is some $\delta'(\lambda) = \mathsf{non\text{-}negl}(\lambda)$ such that

$$\begin{aligned}&\mathbb{E}\left[\left\|\Pi[B[x^*]]U^{\mathsf{F}_1,\ldots,\mathsf{GSim}}|\widetilde{\psi}_{x^*}\rangle\right\|^2 : |\widetilde{\psi}_{x^*}\rangle, (\mathsf{F}_1, \ldots, \mathsf{F}_t, \mathsf{GSim}) \leftarrow \mathsf{QObf}_2(1^\lambda, |\psi_{x^*}\rangle)\right] \\ &\geq \delta'(\lambda) \cdot 2^{-m} \geq \delta'(\lambda) \cdot 2^{-n}.\end{aligned}$$

However, this violates Lemma 7.16 with $\tau = 0$, which is proven in the following section. Indeed, plugging in $\kappa = n^4$, the lemma states that, for some constant c,

$$\begin{aligned}&\mathbb{E}\left[\left\|\Pi[B[x^*]]U^{\mathsf{F}_1,\ldots,\mathsf{GSim}}|\widetilde{\psi}_{x^*}\rangle\right\|^2 : |\widetilde{\psi}_{x^*}\rangle, (\mathsf{F}_1, \ldots, \mathsf{F}_t, \mathsf{GSim}) \leftarrow \mathsf{QObf}_2(1^\lambda, |\psi_{x^*}\rangle)\right] \\ &= 2^{3n(t+1)-cn^4} = 2^{-\Omega(n^2)} < \delta'(\lambda) \cdot 2^{-n},\end{aligned}$$

which gives us the contradiction. □

Claim 7.10. $|\mathcal{H}_2 - \mathcal{H}_3| \leq 2^{-\Omega(\kappa)}$.

Proof.

$$
\begin{aligned}
&|\mathcal{H}_2 - \mathcal{H}_3| \\
&\leq 2^n \cdot \left[\sum_{w^* \neq w'^*} \mathbb{E}\left[\left\| \Pi[w'^*] D_1^{\mathsf{F}_1,\ldots,\mathsf{GSim}} \Pi[w^*] \left|\widetilde{\psi}\right\rangle \right\|^2 : \left|\widetilde{\psi}\right\rangle, (\mathsf{F}_1,\ldots,\mathsf{F}_t,\mathsf{GSim}) \leftarrow \mathsf{QObf}_2(1^\lambda, |\psi\rangle) \right] \right]^{1/2} \\
&\leq 2^n \cdot \left[2^n \cdot \sum_{w^*} \mathbb{E}\left[\left\| \Pi[\neg w^*] D_1^{\mathsf{F}_1,\ldots,\mathsf{GSim}} \Pi[w^*] \left|\widetilde{\psi}\right\rangle \right\|^2 : \left|\widetilde{\psi}\right\rangle, (\mathsf{F}_1,\ldots,\mathsf{F}_t,\mathsf{GSim}) \leftarrow \mathsf{QObf}_2(1^\lambda, |\psi\rangle) \right] \right]^{1/2} \\
&\leq 2^{2n} \cdot 2^{-\Omega(\kappa)} = 2^{-\Omega(\kappa)},
\end{aligned}
$$

where the first inequality follows from Lemma 3.4 (which is an application of Cauchy-Schwarz) and the third follows from Corollary 7.7 proven above.

□

Claim 7.11. $|\mathcal{H}_3 - \mathcal{H}_4| = 2^{-\Omega(\kappa)}$.

Proof. This follows from Corollary 7.7 proven above and Lemma 3.6 (which is a standard oracle hybrid argument). □

Claim 7.12. $|\mathcal{H}_4 - \mathcal{H}_5| = 2^{-\Omega(\kappa)}$.

Proof. This follows by combining Claim 7.5 and Claim 7.4 proven above. □

Claim 7.13. $|\mathcal{H}_5 - \mathcal{H}_6| \leq 2^{-\Omega(\kappa)}$.

Proof.

$$
\begin{aligned}
&|\mathcal{H}_5 - \mathcal{H}_6| \\
&\leq 2^n \cdot \left[\sum_{w^* \neq w'^*} \mathbb{E}\left[\left\| \Pi[w'^*] D_1^{\mathsf{FSim}_1,\ldots,\mathsf{GSim}} \Pi[w^*] \left|\widetilde{\psi}\right\rangle \right\|^2 : \left|\widetilde{\psi}\right\rangle, \mathsf{F}_1,\ldots,\mathsf{F}_t,\mathsf{GSim} \leftarrow \mathsf{QObf}_5(1^\lambda, |\psi\rangle) \right] \right]^{1/2} \\
&\leq 2^n \cdot \left[2^n \cdot \sum_{w^*} \mathbb{E}\left[\left\| \Pi[\neg w^*] D_1^{\mathsf{FSim}_1,\ldots,\mathsf{GSim}} \Pi[w^*] \left|\widetilde{\psi}\right\rangle \right\|^2 : \left|\widetilde{\psi}\right\rangle, \mathsf{F}_1,\ldots,\mathsf{F}_t,\mathsf{GSim} \leftarrow \mathsf{QObf}_5(1^\lambda, |\psi\rangle) \right] \right]^{1/2} \\
&\leq 2^{2n} \cdot 2^{-\Omega(\kappa)} = 2^{-\Omega(\kappa)},
\end{aligned}
$$

where the first inequality follows from Lemma 3.4 (which is an application of Cauchy-Schwarz) and the third follows from Claim 7.3 proven above.

□

Claim 7.14. $|\mathcal{H}_6 - \mathcal{H}_7| \leq 2^{-\Omega(\kappa)}$.

Proof. This follows from privacy of the authentication scheme (Theorem 4.12), since the oracles in $\mathcal{H}_6$ can be implemented with oracle access to $\mathsf{Ver}_{k,\cdot,\cdot}(\cdot)$ rather than $\mathsf{Dec}_{k,\cdot,\cdot}(\cdot)$. □

□

7.4 Inductive Argument

In this section, we give an inductive proof of Lemma 7.16, which was required by Claim 7.9 above. First, we describe a variant of QObf 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 QObf.
  - 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 QObf_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 $\mathsf{Ver}_{k,L_{t+1}\dots L_1,\theta_{t+1}[V_1,\dots,V_\tau]}(\widetilde{v}_1,\dots,\widetilde{v}_\tau)=\bot$.
- Compute $(v_{\tau+1},\dots,v_{t+1}) = \mathsf{Dec}_{k,L_{t+1}\dots L_1,\theta_{t+1}[V_{\tau+1},\dots,V_{t+1}]}(\widetilde{v}_{\tau+1},\dots,\widetilde{v}_{t+1})$, and output $\bot$ if the result is $\bot$.
- Output $g^{x,r_1^*,\dots,r_\tau^*,r_{\tau+1},\dots,r_t}(v_1^*,\dots,v_\tau^*,v_{\tau+1},\dots,v_{t+1})$.

- Let $C[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]}]$ be the set that includes, for any $i\in[\tau]$, all $(\sigma_{x^*},\widetilde{v}_1,\dots,\widetilde{v}_i,\widetilde{w}_i,\ell_1,\dots,\ell_{i-1})$ such that

$$\begin{aligned}&\mathsf{F}_i[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau-1]}](x^*,\sigma_{x^*},\widetilde{v}_1,\dots,\widetilde{v}_i,\widetilde{w}_i,\ell_1,\dots,\ell_{i-1})\\ &\neq \mathsf{F}_i[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]}](x^*,\sigma_{x^*},\widetilde{v}_1,\dots,\widetilde{v}_i,\widetilde{w}_i,\ell_1,\dots,\ell_{i-1}),\end{aligned}$$

  and all $(\sigma_{x^*},\widetilde{v}_1,\dots,\widetilde{v}_{t+1},\ell_1,\dots,\ell_t)$ such that

$$(\sigma_{x^*},\widetilde{v}_1,\dots,\widetilde{v}_{t+1},\ell_1,\dots,\ell_t)\in B[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau-1]}]\setminus B[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]}].$$

- Output $|\widetilde{\psi}\rangle = |\psi_k\rangle\,|\mathsf{sk}\rangle\,, O=(\mathsf{F}_1[\cdot],\dots,\mathsf{F}_t[\cdot],\mathsf{GSim})\,, B[\cdot], C[\cdot]$.

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 7.9 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 QObf_2 augmented with the set $B[x^*]$ as defined in the proof of Claim 7.9 (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 $\left|\psi[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]}]\right\rangle$ as defined in Section 7.2.

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 $\left|\psi[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]}]\right\rangle$ 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 7.24.

Lemma 7.15. *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*

$$\left| \mathbb{E}\left[\left\| \Pi\left[B\left[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau-1]}\right]\right] U^{\mathsf{F}_1[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau-1]}],\ldots,\mathsf{F}_t[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau-1]}],\mathsf{GSim}} |\widetilde{\psi}\rangle \right\|^2 \right] \right.$$
$$\left. - \mathbb{E}\left[\left\| \Pi\left[B\left[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}\right]\right] U^{\mathsf{F}_1[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]}],\ldots,\mathsf{F}_t[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]}],\mathsf{GSim}} |\widetilde{\psi}\rangle \right\|^2 \right] \right| \leq 2^{-\Omega(\kappa)},$$

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 3.6 (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

$$\Pi[\neg(v^*_\tau, r^*_\tau)] := \Pi[\neg r^*_\tau] + \Pi[\neg v^*_\tau].$$

Finally, given the sampled signature token verification key vk, define the projector onto valid signatures of x^*:

$$\Pi[x^*, \mathsf{vk}] := \sum_{\sigma:\mathsf{TokVer}(\mathsf{vk},x^*,\sigma)=\top} |\sigma\rangle\langle\sigma|,$$

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 7.24 proven in the next section. This completes the proof. □

Now, we show the main lemma of the section.

Lemma 7.16. *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 7.15. For the inductive step, suppose that Lemma 7.16 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 7.16 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 3.5 (an application of Cauchy-Schwarz).
- The second inequality follows from Lemma 7.15 proven above.
- The third inequality follows from Lemma 7.16 for τ (the induction hypothesis).
- The final inequality follows because the two summands can each be bounded by the quantity $2^{3n(t+1-(\tau-1))-c\kappa-1}$.

□

7.5 Hardness of Mapping

Our final step is to prove the "hardness of mapping" lemma, Lemma 7.24, that was required for Lemma 7.15 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 QObf_5 from Section 7.3.

$\mathsf{ParSim}_1(1^\lambda, |\psi\rangle)$:

- 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 the function $\mathsf{F}_i[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}, \{w_\iota^*\}_{\iota\in[\tau+1,t]}](x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \ldots, \ell_{i-1})$:
  - 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 QObf.
  - 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 $\mathsf{Ver}_{k,L_i\dots L_1,\theta_i}(\widetilde{v}_1,\dots,\widetilde{v}_i,\widetilde{w}_i) = \bot$.
  - Set $\ell_i := H(x,\sigma_x,\widetilde{v}_1,\dots,\widetilde{v}_i,\ell_1,\dots,\ell_{i-1},r_i^*)$, and output $(\ell_i,\widetilde{v}_i)$.
- If $i > \tau$:
  - For each $\iota \in [\tau+1, i-1]$, let

$$\ell_{\iota,0} = H(x,\sigma_x,\widetilde{v}_1,\dots,\widetilde{v}_\iota,\ell_1,\dots,\ell_{\iota-1},0),\ \ \ell_{\iota,1} = H(x,\sigma_x,\widetilde{v}_1,\dots,\widetilde{v}_\iota,\ell_1,\dots,\ell_{\iota-1},1),$$

    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 $\mathsf{Ver}_{k,L_i\dots L_1,\theta_i[V_1,\dots,V_\tau,W_i]}(\widetilde{v}_1,\dots,\widetilde{v}_\tau,\widetilde{w}_i) = \bot$.
  - Compute $(v_{\tau+1},\dots,v_i) = \mathsf{Dec}_{k,L_i\dots L_1,\theta_i[V_{\tau+1},\dots,V_i]}(\widetilde{v}_{\tau+1},\dots,\widetilde{v}_i)$, and output $\bot$ if the result is $\bot$.
  - Compute $(\cdot,r_i) = f_i^{x,r_1^*,\dots,r_\tau^*,r_{\tau+1},\dots,r_{i-1}}(v_1^*,\dots,v_\tau^*,v_{\tau+1},\dots,v_i,w_i^*)$.
  - Set $\ell_i := H(x,\sigma_x,\widetilde{v}_1,\dots,\widetilde{v}_i,\ell_1,\dots,\ell_{i-1},r_i)$, and output $(\ell_i,\widetilde{v}_i)$.

- Define GSim as in QObf_2.
- Output $|\widetilde{\psi}\rangle = |\psi_k\rangle\,|\mathsf{sk}\rangle\,, O = (\mathsf{F}_1[\cdot],\dots,\mathsf{F}_t[\cdot],\mathsf{GSim})$.

$\mathsf{ParSim}_2(1^\lambda,|\psi\rangle)$:

- 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 the function $\mathsf{F}_i[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]},\{w_\iota^*\}_{\iota\in[\tau+1,t]}](x,\sigma_x,\widetilde{v}_1,\dots,\widetilde{v}_i,\widetilde{w}_i,\ell_1,\dots,\ell_{i-1})$:
  - If $x \neq x^*$, output $\mathsf{F}_i\,(x,\sigma_x,\widetilde{v}_1,\dots,\widetilde{v}_i,\widetilde{w}_i,\ell_1,\dots,\ell_{i-1})$, where F_i is defined as in QObf.
  - Output $\bot$ if $\mathsf{TokVer}(\mathsf{vk},x,\sigma_x) = \bot$.
  - For each $\iota \in [i-1]$, let

$$\ell_{\iota,0} = H(x,\sigma_x,\widetilde{v}_1,\dots,\widetilde{v}_\iota,\ell_1,\dots,\ell_{\iota-1},0),\ \ \ell_{\iota,1} = H(x,\sigma_x,\widetilde{v}_1,\dots,\widetilde{v}_\iota,\ell_1,\dots,\ell_{\iota-1},1),$$

    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 $\mathsf{Ver}_{k,L_i\dots L_1,\theta_i}(\widetilde{v}_1,\dots,\widetilde{v}_i,\widetilde{w}_i) = \bot$.
    - Set $\ell_i := H(x,\sigma_x,\widetilde{v}_1,\dots,\widetilde{v}_i,\ell_1,\dots,\ell_{i-1},r_i^*)$, and output $(\ell_i,\widetilde{v}_i)$.
  - If $i > \tau$:
    - Output $\bot$ if $\mathsf{Ver}_{k,L_i\dots L_1,\theta_i[V_1,\dots,V_\tau,W_i]}(\widetilde{v}_1,\dots,\widetilde{v}_\tau,\widetilde{w}_i) = \bot$.
    - Compute $(v_{\tau+1},\dots,v_i) = \mathsf{Dec}_{k,L_i\dots L_1,\theta_i[V_{\tau+1},\dots,V_i]}(\widetilde{v}_{\tau+1},\dots,\widetilde{v}_i)$, and output $\bot$ if the result is $\bot$.
    - For $\iota \in [\tau+1,i]$, compute $(\cdot,r_\iota) = f_\iota^{x,r_1^*,\dots,r_\tau^*,r_{\tau+1},\dots,r_{\iota-1}}(v_1^*,\dots,v_\tau^*,v_{\tau+1},\dots,v_\iota,w_\iota^*)$.
    - Set $\ell_i := H(x,\sigma_x,\widetilde{v}_1,\dots,\widetilde{v}_i,\ell_1,\dots,\ell_{i-1},r_i)$, and output $(\ell_i,\widetilde{v}_i)$.

- Define GSim as in QObf_2.
- Output $|\widetilde{\psi}\rangle = |\psi_k\rangle\,|\mathsf{sk}\rangle\,, O = (\mathsf{F}_1[\cdot],\dots,\mathsf{F}_t[\cdot],\mathsf{GSim})$.

$\mathsf{ParSim}_3(1^\lambda,|\psi\rangle)$:

- 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 QObf.
  - 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 $\ell_i \coloneqq H(x, \sigma_x, \widetilde{v}_1, \ldots, \widetilde{v}_i, \ell_1, \ldots, \ell_{i-1}, 0)$, and output $(\widetilde{v}_i, \ell_i)$.
- Define GSim as in QObf_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 QObf_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 QObf_4.
- Define GSim as in QObf_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 QObf_5.
- Define GSim as in QObf_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 7.24) 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 7.24. 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 7.17. *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

$$\mathbb{E}\left[\left\|\Pi[\neg(v_\tau^*,r_\tau^*)]U^O\Pi[\{w_\iota^*\}_{\iota\in[\tau+1,t]}]\left|\widetilde{\psi}\right\rangle\right\|^2 : \left|\widetilde{\psi}\right\rangle, O \leftarrow \mathsf{ParSim}_5(1^\lambda, |\psi[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]}]\rangle)\right] = 2^{-\Omega(\kappa)},$$

where $\Pi[\neg(v_\tau^*,r_\tau^*)]$ is defined in the proof of Lemma 7.15.

Proof. The key point is that in QObf_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 4.9), and in particular that it satisfies Definition 4.4 (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 $\mathsf{ParSim}_3,\mathsf{ParSim}_4$, and ParSim_5 in the case when all of the W wires have been collapsed to some value w^*.

Claim 7.18. *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*

$$\begin{aligned}&\mathbb{E}\left[\left\|D^{\mathsf{FSim}_1[x^*,w^*],\ldots,\mathsf{GSim}}\left(k,\Pi[w^*]\left|\widetilde{\psi}\right\rangle\right)\right\|^2 : \left|\widetilde{\psi}\right\rangle,(\mathsf{FSim}_1[\cdot],\ldots,\mathsf{GSim}) \leftarrow \mathsf{ParSim}_4(1^\lambda,|\psi[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]}]\rangle)\right]\\ &=\mathbb{E}\left[\left\|D^{\mathsf{FSim}_1,\ldots,\mathsf{GSim}}\left(k,\Pi[w^*]\left|\widetilde{\psi}\right\rangle\right)\right\|^2 : \left|\widetilde{\psi}\right\rangle,(\mathsf{FSim}_1,\ldots,\mathsf{GSim}) \leftarrow \mathsf{ParSim}_5(1^\lambda,|\psi[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]}]\rangle)\right].\end{aligned}$$

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 7.4, except that here we are only switching oracle inputs on $x\neq x^*$ rather than all x. □

Claim 7.19. *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*

$$\begin{aligned}&\Bigg|\mathbb{E}\left[\left\|D^{\mathsf{FSim}_1[x^*],\ldots,\mathsf{GSim}}\left(k,\Pi[w^*]\left|\widetilde{\psi}\right\rangle\right)\right\|^2 : \left|\widetilde{\psi}\right\rangle,(\mathsf{FSim}_1[\cdot],\ldots,\mathsf{GSim}) \leftarrow \mathsf{ParSim}_3(1^\lambda,|\psi[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]}]\rangle)\right]\\ &-\mathbb{E}\left[\left\|D^{\mathsf{FSim}_1[x^*,w^*],\ldots,\mathsf{GSim}}\left(k,\Pi[w^*]\left|\widetilde{\psi}\right\rangle\right)\right\|^2 : \left|\widetilde{\psi}\right\rangle,(\mathsf{FSim}_1[\cdot],\ldots,\mathsf{GSim}) \leftarrow \mathsf{ParSim}_4(1^\lambda,|\psi[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]}]\rangle)\right]\Bigg|\\ &=2^{-\Omega(\kappa)},\end{aligned}$$

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 7.5, 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 7.20. *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*

$$\mathbb{E}\left[\left\|\Pi[\neg\{w_i^*\}_{i\in[\tau+1,t]}]U^{\mathsf{FSim}_1[x^*],\ldots,\mathsf{GSim}}\Pi[\{w_i^*\}_{i\in[\tau+1,t]}]\,|\widetilde{\psi}\rangle\,\right\|^2\right] = 2^{-\Omega(\kappa)},$$

where the expectation is over

$$|\widetilde{\psi}\rangle\,, (\mathsf{FSim}_1[\cdot],\ldots,\mathsf{GSim}) \leftarrow \mathsf{ParSim}_3(1^\lambda,\, |\psi[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]}]\rangle).$$

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,

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathsf{ParSim}_3}\left[\left\|\Pi[\neg\{w_i^*\}_{i\in[\tau+1,t]}]U^{\mathsf{FSim}_1[x^*],\ldots,\mathsf{GSim}}\Pi[\{w_i^*\}_{i\in[\tau+1,t]}]\,|\widetilde{\psi}\rangle\,\right\|^2\right]\\
&\le 2^n\cdot\sum_{\{w_\iota^*\}_{\iota\in[\tau]}}\mathop{\mathbb{E}}_{\mathsf{ParSim}_3}\left[\left\|\Pi[\neg\{w_i^*\}_{i\in[\tau+1,t]}]U^{\mathsf{FSim}_1[x^*],\ldots,\mathsf{GSim}}\Pi[w^*]\,|\widetilde{\psi}\rangle\,\right\|^2\right]\\
&\le 2^n\cdot\sum_{\{w_\iota^*\}_{\iota\in[\tau]}}\mathop{\mathbb{E}}_{\mathsf{ParSim}_5}\left[\left\|\Pi[\neg\{w_i^*\}_{i\in[\tau+1,t]}]U^{\mathsf{FSim}_1,\ldots,\mathsf{GSim}}\Pi[w^*]\,|\widetilde{\psi}\rangle\,\right\|^2\right] + 2^n\cdot 2^{-\Omega(\kappa)}\\
&= 2^{-\Omega(\kappa)}
\end{aligned}$$

where

- The first inequality follows from Lemma 3.5 (an application of Cauchy-Schwarz).
- The second inequality follows by combining Claim 7.19 and Claim 7.18 proven above.
- The third inequality follows from Claim 7.17 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 $i \le \tau$, 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 7.21. *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 7.22. *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 3.6 (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 7.23. *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 3.6 (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 7.22 and Claim 7.21, and the second inequality follows from Claim 7.20, all proven above. □

Now, we prove the main lemma of this section. The proof of Lemma 7.24 combines some claims proven above with Lemma 7.25 that follows. Lemma 7.25 is a similar claim but with respect to ParSim_3 instead of ParSim_0.

Lemma 7.24. *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 7.15, 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 3.5 (an application of Cauchy-Schwarz).
- The second inequality follows from combining Claim 7.23, Claim 7.22, and Claim 7.21 proven above.
- The third inequality follows from Lemma 7.25 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 7.1.

Lemma 7.25. *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*

$$\mathbb{E}\left[\left\| \left(\Pi[x^*, \mathsf{vk}] \otimes \Pi[\neg(v_\tau^*, r_\tau^*)]\right) U^{\mathsf{FSim}_1[x^*],\ldots,\mathsf{GSim}} \Pi[\{w_\iota^*\}_{\iota\in[\tau+1,t]}] |\widetilde{\psi}\rangle \right\|^2\right] = 2^{-\Omega(\kappa)},$$

where the expectation is over

$$|\widetilde{\psi}\rangle, (\mathsf{FSim}_1[\cdot], \ldots, \mathsf{FSim}_t[\cdot], \mathsf{GSim}) \leftarrow \mathsf{ParSim}_3(1^\lambda, |\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]\rangle).$$

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 $\mathcal{D} := \{\mathcal{D}_a\}_{a\in\{0,1\}^K}$, 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 $|+_H\rangle^{\mathcal{D}} := |+_\kappa\rangle^{\otimes\{\mathcal{D}_a\}_{a\in\{0,1\}^K}}$ to be the uniform superposition over all random oracles H. Finally, let $A[\neq x^*] := \{a \in \{0,1\}^K : a = (x, \cdot) \text{ for } x \neq x^*\}$ 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 $(x, \dots)$ for some $x \neq x^*$.

Claim 7.26. *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*

$$\mathbb{E}\left[\left\| \left(\Pi[\neg(v_\tau^*, r_\tau^*)] \otimes |+_\kappa\rangle\langle+_\kappa|^{\otimes\{\mathcal{D}_a\}_{a\in A[\neq x^*]}}\right) U^{\mathsf{FSim}_1[x^*],\dots,\mathsf{GSim}} |\widetilde{\psi}^*\rangle |+_H\rangle^{\mathcal{D}} \right\|^2\right] = 2^{-\Omega(\kappa)},$$

where the expectation is over

$$|\widetilde{\psi}\rangle, (\mathsf{FSim}_1[\cdot], \dots, \mathsf{FSim}_t[\cdot], \mathsf{GSim}) \leftarrow \mathsf{ParSim}_3(1^\lambda, |\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]\rangle),$$

and

$$|\widetilde{\psi}^*\rangle := \Pi[\{w_\iota^*\}_{\iota\in[\tau+1,t]}] |\widetilde{\psi}\rangle .$$

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 $\mathsf{FSim}_i[x^*][z](x, \sigma_x, \widetilde{v}_1, \dots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \dots, \ell_{i-1})$:
  - If $z = \emptyset$, output $\mathsf{FSim}_i(x, \sigma_x, \widetilde{v}_1, \dots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \dots, \ell_{i-1})$, where FSim_i is defined as in QObf_5.
  - Otherwise, if $z = w^*$:
    - If $x = x^*$, output $\mathsf{FSim}_i(x^*, \sigma_{x^*}, \widetilde{v}_1, \dots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \dots, \ell_{i-1})$.
    - If $x \neq x^*$, output $\mathsf{FSim}_i[w^*](x, \sigma_x, \widetilde{v}_1, \dots, \widetilde{v}_i, \widetilde{w}_i, \ell_1, \dots, \ell_{i-1})$, where $\mathsf{FSim}_i[w^*]$ is defined as in QObf_4.
- Define GSim as in QObf_2.
- Output $|\widetilde{\psi}\rangle = |\psi_k\rangle |\mathsf{sk}\rangle, O = (\mathsf{FSim}_1[x^*][\cdot], \dots, \mathsf{FSim}_t[x^*][\cdot], \mathsf{GSim})$.

Indeed, observe that

$$|\widetilde{\psi}\rangle, (\mathsf{FSim}_1[x^*][w^*], \dots, \mathsf{FSim}_t[x^*][w^*], \mathsf{GSim}) \leftarrow \mathsf{Sim}[x^*](1^\lambda, |\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]\rangle)$$

is equivalent to

$$|\widetilde{\psi}\rangle, (\mathsf{FSim}_1[x^*, w^*], \dots, \mathsf{FSim}_t[x^*, w^*], \mathsf{GSim}) \leftarrow \mathsf{ParSim}_4(1^\lambda, |\psi[x^*, \{v_\iota^*, r_\iota^*\}_{\iota\in[\tau]}]\rangle),$$

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 7.3:

$R[k,w^*](x,\widetilde{v}_1,\ldots,\widetilde{v}_i)$:

- 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 $\bot$, then output $\bot$.
- 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.

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 7.27. *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 7.28. *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 7.19 proven above.
- The following equality is by definition of $\mathsf{Sim}[x^*]$.
- The following equality is Fact 7.27.
- The following equality is Fact 7.28.
- The following equality is by definition of $\mathsf{Sim}[x^*]$.
- The final inequality follows from Claim 7.17 proven above.

□

Now, assume for contradiction that the lemma is false. Combined with the fact that Claim 7.26 is true, this implies that

$$\mathbb{E}\left[\left\| \left(\Pi[x^*,\mathsf{vk}] \otimes \left(\mathcal{I} - |+_\kappa\rangle\langle+_\kappa|^{\otimes\{\mathcal{D}_a\}_{a\in A[\neq x^*]}}\right)\right) U^{\mathsf{FSim}_1[x^*],\ldots,\mathsf{GSim}} \left|\widetilde{\psi}^*\right\rangle |+_H\rangle^{\mathcal{D}} \right\|^2\right] = 2^{-o(\kappa)},$$

where the expectation is over

$$\left|\widetilde{\psi}\right\rangle, (\mathsf{FSim}_1[\cdot],\ldots,\mathsf{FSim}_t[\cdot],\mathsf{GSim}) \leftarrow \mathsf{ParSim}_3(1^\lambda, |\psi[x^*,\{v_\iota^*,r_\iota^*\}_{\iota\in[\tau]}]\rangle).$$

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 $\left|\widetilde{\psi}^*\right\rangle, |+_H\rangle$, and $(\mathsf{FSim}_1[\cdot],\ldots,\mathsf{FSim}_t[\cdot],\mathsf{GSim})$ as in the description of Lemma 7.25, and run $U^{\mathsf{FSim}_1[\cdot],\ldots,\mathsf{GSim}} \left|\widetilde{\psi}^*\right\rangle |+_H\rangle$, 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 3.8). 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.

□

8 Acknowledgements

We thank Sam Gunn for correspondence throughout this project, including several illuminating discussions.

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.

[ABOEM17] Dorit Aharonov, Michael Ben-Or, Elad Eban, and Urmila Mahadev. Interactive proofs for quantum computations, 2017.

[AC12] Scott Aaronson and Paul Christiano. Quantum money from hidden subspaces. In Howard J. Karloff and Toniann Pitassi, editors, *44th ACM STOC*, pages 41–60. ACM Press, May 2012.

[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*, page 255–268. Association for Computing Machinery, New York, NY, USA, 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.

[ALP21] Prabhanjan Ananth and Rolando L. La Placa. Secure software leasing. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021*, pages 501–530, Cham, 2021. Springer International Publishing.

[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.

[BB84] C. H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing*, page 175, India, 1984.

[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.

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

[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.

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

[BK21] Anne Broadbent and Raza Ali Kazmi. Constructions for quantum indistinguishability obfuscation. In Patrick Longa and Carla Ràfols, editors, *Progress in Cryptology - LATINCRYPT 2021 - 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.

[BPR15] Nir Bitansky, Omer Paneth, and Alon Rosen. On the cryptographic hardness of finding a Nash equilibrium. In Venkatesan Guruswami, editor, *56th FOCS*, pages 1480–1498. IEEE Computer Society Press, October 2015.

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

[CG23] Andrea Coladangelo and Sam Gunn. How to use quantum indistinguishability obfuscation, 2023.

[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.

[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.

[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. In Omar Fawzi and Michael Walter, editors, *18th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2023)*, volume 266 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 2:1–2:23, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[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 *54th FOCS*, pages 40–49. IEEE Computer Society Press, October 2013.

[GGH15] Craig Gentry, Sergey Gorbunov, and Shai Halevi. Graph-induced multilinear maps from lattices. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 498–527. Springer, Heidelberg, March 2015.

[GGSW13] Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 467–476. ACM, 2013.

[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.

[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.

[ILW23] Rahul Ilango, Jiatu Li, and R. Ryan Williams. Indistinguishability obfuscation, range avoidance, and bounded arithmetic. 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 1076–1089. ACM, 2023.

[JLS21] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In Samir Khuller and Virginia Vassilevska Williams,

editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 60–73. ACM, 2021.

[JLS22] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from LPN over $\mathbb {F}_p$, dlin, and prgs in nc^0. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part I*, volume 13275 of *Lecture Notes in Computer Science*, pages 670–699. Springer, 2022.

[JNV+20] Zhengfeng Ji, Anand Natarajan, Thomas Vidick, John Wright, and Henry Yuen. Mip*=re. *CoRR*, abs/2001.04383, 2020.

[KS08] Vladimir Kolesnikov and Thomas Schneider. Improved garbled circuit: Free XOR gates and applications. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations*, volume 5126 of *Lecture Notes in Computer Science*, pages 486–498. Springer, 2008.

[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.

[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 Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 259–267. IEEE Computer Society, 2018.

[Sho94] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994*, pages 124–134. IEEE Computer Society, 1994.

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In David B. Shmoys, editor, *46th ACM STOC*, pages 475–484. ACM Press, May / June 2014.

[Wie83] Stephen Wiesner. Conjugate coding. *SIGACT News*, 15(1):78–88, 1983.

[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 *Advances in Cryptology – EUROCRYPT 2021: 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17–21, 2021, Proceedings, Part III*, page 127–156, Berlin, Heidelberg, 2021. Springer-Verlag.

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*, pages 162–167. IEEE Computer Society, 1986.

[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.