# Classical Oracle Separation of QMA from the QCMA Hierarchy and Quantum Cryptography Even When P = QCMA

Alper Cakan*

**This work is in submission at STOC'27 and thus the full manuscript has been publicly posted on the conference website since November 3rd, 2026.**

**Abstract**

Inspired by the *Codeword-Hash Structure* problem recently introduced by Cakan (arXiV, 2026) to separate QMA from QCIP, we introduce a new problem called ***Codeword-Hash pairs in Noise*** (CHN) and prove that, in the query complexity model, it is in QMA but hard for the QCMA hierarchy $\mathsf{QCMAH} = \bigcup_{d\geq 1} \mathsf{QCMA}^{\mathsf{QCMA}^{\cdots^{\mathsf{QCMA}}}}$, the classical-witness quantum-verifier analogue of $\mathsf{PH} = \bigcup_{d\geq 1} \mathsf{NP}^{\mathsf{NP}^{\cdots^{\mathsf{NP}}}}$.

Using this problem, we show that there exists a classical oracle $\mathcal{O}$ such that

$$\mathsf{P}^{\mathcal{O}} = \mathsf{BQP}^{\mathcal{O}} = \mathsf{QCMA}^{\mathcal{O}} = \mathsf{QCMAH}^{\mathcal{O}} \subset \mathsf{QMA}^{\mathcal{O}} \qquad \text{but} \qquad \mathsf{QMA}^{\mathcal{O}} \not\subseteq \mathsf{QCMAH}^{\mathcal{O}},$$

thus showing that *quantum state* or *quantum proof advantage* could exist even if classical computers can efficiently solve any problem that can be quantumly verified with a classical proof.

Further, using our problem, we show that various quantum cryptographic primitives can exist even when P = QCMA and thus neither quantum-computable one-way functions nor (verifiable) quantum *computational* advantage exist. In more detail, we show that there exists a classical oracle $\mathcal{O}$ such that $\mathsf{P}^{\mathcal{O}} = \mathsf{BQP}^{\mathcal{O}} = \mathsf{QCMA}^{\mathcal{O}} = \mathsf{QCMAH}^{\mathcal{O}}$ but the following primitives exist relative to $\mathcal{O}$, answering an open question of Kretschmer, Qian, and Tal (STOC 2025).

- Pseudorandom states (PRS) with classical keys.
- Pseudorandom functions (PRF) with (copy-protected) quantum secret keys, and thus also pseudorandom unitaries (PRU) with quantum secret keys,
- Public-key encryption (PKE) with a (copy-protected) quantum decryption key, classical public keys and ciphertexts,
- Signature schemes with quantum signing keys and either classical verification keys and quantum signatures or quantum verification keys and classical signatures,
- Collusion-resistant public-key quantum money (PKQM) with classical verification keys.

**Keywords:** Quantum complexity, quantum cryptography, MicroCrypt.

*Carnegie Mellon University. alpercakan98@gmail.com.

# Contents


**1 Introduction** **4**
- 1.1 Our results . . . 5
- 1.2 Organization . . . 7

**2 Technical Overview** **8**
- 2.1 Model . . . 8
- 2.2 The CHN Problem and Our Query Complexity Result . . . 9
- 2.3 Complexity Proof Overview . . . 11
- 2.4 Cryptographic Oracle and Constructions . . . 17
- 2.5 Cryptography Proof Overview . . . 19
- 2.6 Detailed Organization . . . 25
- 2.7 Future Directions . . . 25
- 2.8 Related work . . . 26
- 2.9 AI Methodology . . . 26

**3 Preliminaries** **27**
- 3.1 Notation and Conventions . . . 27
- 3.2 Complexity . . . 28
- 3.3 Quantum Information . . . 31
- 3.4 Cryptography . . . 32

## I Technical Tools

**4 Codeword-Hash Pairs, Noise, and Incompressibility** **36**

**5 Deterministic-Classical-Query Simulation of Quantum Verifiers** **40**

**6 Removing Classical Witnesses by Switching** **44**

## II QMA versus QCMAH

**7 Codeword-Hash Pairs in Noise (CHN) Problem and BB-QMA Verification** **49**

**8 Depth Reduction for BB-QCMAH Machines** **51**

**9 BB-QCMAH Query Lower Bounds and QMA–QCMAH Oracle Separation** **55**

## III Technical Tools for Cryptography

**10 Simulating the Recursive Oracle** **57**

**11 Preparing Quantum States from Boolean Oracles** **64**

**12 Changing Preparation Targets** **66**

13 Encoded Functions with Quantum Keys 69

## IV Quantum Cryptography with P = QCMA

14 Reusable Keys for Coset Evaluation 72

15 Copy-Protected Pseudorandom Functions 75

16 Copy-Protected Public-Key Encryption 80

17 Signatures with Quantum Keys 85

18 Pseudorandom States 92

19 Quantum Cryptography Relative to an Oracle with P = QCMA 94

---

## Appendices

### Appendix I: Deferred Proofs

A Proofs of the Quantum Query and Encoding Bounds 103

B Proof of the Codeword-Hash Guarantees (Lemma 7) 105

C Correctness of the Signature Constructions 107

D Proof of the Ordered Switching Lemma (Lemma 25) 111

E Estimating Acceptance and Searching Classical Inputs 114

### Appendix II: Cryptographic Security Proofs

F Copy-Protection Definitions 116

G Security of Direct Coset Evaluation 117

H Proofs of the Signature Verification Bounds 126

I Changing the Oracle at a Hidden Key 128

### Appendix III: Proofs of Fixed-Oracle Separations

J Proof of the Fixed-Oracle Collapse and Separation (Theorem 11) 138

K Proof of the Fixed-Oracle Cryptography Theorem (Theorem 19) 142

# 1 Introduction

The QMA versus QCMA question asks whether quantum proofs are more powerful than classical ones, and has been a central question in quantum complexity since the introduction of these classes [AN02]. Aaronson and Kuperberg [AK07] gave the first evidence for such an advantage for quantum proofs by separating these classes relative to a quantum unitary oracle. After many attempts [FK18, NN24, LLPY24, BDK24, Zha25, LMY25, CGS26], recently the breakthrough work of Bostanci, Haferkamp, Nirkhe, and Zhandry [BHNZ26] finally showed the first *classical* oracle separation. Later, a simpler classical oracle separation was shown by Bostanci, Huang, and Vaikuntanathan [BHV26].

However, a fundamental question still remains open: whether QMA is contained in the QCMA hierarchy

$$\mathsf{QCMAH} = \bigcup_{d\geq 1} \underbrace{\mathsf{QCMA}^{\mathsf{QCMA}^{\cdots^{\mathsf{QCMA}}}}}_{d \text{ levels}},$$

the classical-witness quantum-verifier analogue of the polynomial hierarchy $\mathsf{PH} = \bigcup_{d\geq 1} \mathsf{NP}^{\mathsf{NP}^{\cdots^{\mathsf{NP}}}}$.

Even more importantly, whether QMA is larger even when BQP = QCMA. In other terms, can ***quantum state advantage*** or ***quantum proof advantage*** exist even if every problem that admits a classical proof can also be efficiently solved? Thus, we ask[1]:

> *Is* QMA *in the* QCMA *hierarchy? Can quantum state/proof advantage exist* (QMA $\not\subseteq$ QCMA) *even if* BQP = QCMA*, or even if* P = QCMA*?*

The languages[2] $L \in \mathsf{QMA}$ constructed in [BHNZ26] and [BHV26] both lie in AM, which is contained in the second level $\mathsf{QCMA}^{\mathsf{QCMA}}$ of the QCMA hierarchy. So, they do not obtain a QMA separation from QCMAH, or QMA advantage when BQP = QCMA.

Thus, this question was left open and indeed answering it requires a substantial strengthening of their results: Drawing a parallel to ***quantum computational advantage***[3], we note that an oracle separation of BQP from NP has been known since 2000 [Wat00], but a separation of BQP from PH, and separation of BQP from P when P = NP was obtained much later by the celebrated works of Raz and Tal [RT22] and Aaronson, Ingram, Kretschmer [AIK22], respectively.

**Cryptography** Aside from its complexity-theoretic interest, the question above also has an important connection to quantum cryptography. Initiated by Kretschmer [Kre21], the study of *MicroCrypt*, a fruitful major line of research in quantum cryptography [AQY22, MY22, BCQ23, KQST23, KT24, MY24, CGG24, BJ24, GMMY24, KQT24, BCN25, BMM+25], has shown evidence for the tantalizing possibility that quantum cryptography can exist even if there is no classical cryptography.

While earlier oracle separations in this area, including [Kre21], used quantum unitary oracles, Kretschmer, Qian, Sinha, and Tal [KQST23] finally showed a classical oracle relative to which P = NP yet single-copy pseudorandom states (PRS) exist. Later, Kretschmer, Qian, and Tal [KQT24] showed that there is a classical oracle relative to which P = NP but quantum-computable one-way

[1] Separating $\mathsf{QMA}^{\mathcal{O}}$ from $\mathsf{BQP}^{\mathcal{O}} = \mathsf{QCMA}^{\mathcal{O}}$ is stronger than just separating $\mathsf{QMA}^{\mathcal{O}}$ from $\mathsf{QCMAH}^{\mathcal{O}}$ since $\mathsf{BQP}^{\mathcal{O}} = \mathsf{QCMA}^{\mathcal{O}}$ implies $\mathsf{BQP}^{\mathcal{O}} = \mathsf{QCMAH}^{\mathcal{O}}$ for any oracle $\mathcal{O}$.

[2] Spectral Forrelation problem studied in [BHNZ26] is not necessarily in AM, although the language used in their oracle separation lies in AM because of the version of Spectral Forrelation needed in their QCMA lower-bound proof. In contrast, the problem in [BHV26] itself lies in AM.

[3] We mean efficiently verifiable quantum advantage without quantum memory.

functions (OWFs), public-key encryption (PKE), and multi-copy PRS can still exist, and left open the question of cryptography when $\mathsf{P} = \mathsf{QCMA}$.

Further, we note that the results of [KQT24] necessarily have that quantum computational advantage exists relative to their oracle: Indeed, observe that existence of quantum-computable one-way functions implies $\mathsf{BQP} \neq \mathsf{QCMA}$, and since they also have $\mathsf{P} = \mathsf{NP}$, their oracle world has $\mathsf{P} \neq \mathsf{BQP}$. Thus, we ask:

> *Can quantum cryptography exist even if* $\mathsf{P} = \mathsf{QCMA}$*, that is, without quantum-computable one-way functions (OWF) or quantum computational advantage?*

We believe this is an important question given that there have been many examples of quantum algorithms later being de-quantized [Tan19, CGL+20, KOW26], whereas there is more concrete evidence for quantum state advantage in the standard model [GKK+07, RK11, Mah18, CCHL22].

## 1.1 Our results

In this work we answer both of the questions affirmatively, through a new problem called ***Codeword-Hash pairs in Noise*** (CHN) that we introduce. Briefly, the problem is to decide if the constant-1 function have been planted on top of a string of sparse Bernoulli noise, at locations that are codewords and their symbol-wise hashes (i.e. Yamakawa–Zhandry hashes [YZ24]).

### 1.1.1 Complexity

We use the prefix BB- (for black-box) to distinguish classes in the query complexity model from the usual time-bounded classes. We first prove the following query complexity result for CHN[4].

**Theorem 1** (Informal, Theorems 8 and 10)**.** *In the query complexity model,*

$$\mathsf{CHN} \in \mathsf{BB\text{-}QMA} \setminus \mathsf{BB\text{-}QCMAH}.$$

Then, with a small extension of our query complexity argument, we construct a classical oracle relative to which QCMAH collapses to P while QMA remains strictly larger.

**Theorem 2** (Informal, Theorem 11)**.** *There exist a classical oracle* $\mathcal{O} : \{0,1\}^* \to \{0,1\}$ *and a unary language* $L$ *such that*

$$\mathsf{P}^{\mathcal{O}} = \mathsf{QCMA}^{\mathcal{O}} = \mathsf{QCMAH}^{\mathcal{O}} \subset \mathsf{QMA}^{\mathcal{O}}, \qquad L \in \mathsf{QMA}^{\mathcal{O}} \setminus \mathsf{QCMAH}^{\mathcal{O}}.$$

We note that our cryptography theorem below (with small caveats) subsumes this oracle separation theorem. In fact, we show the stronger separation where $\mathsf{PromiseP}^{\mathcal{O}} = \mathsf{PromiseQCMA}^{\mathcal{O}} = \mathsf{PromiseQCMAH}^{\mathcal{O}}$, but there is a language in $\mathsf{QMA}^{\mathcal{O}}$ that is hard even on average for $\mathsf{QCMAH}^{\mathcal{O}}$ machines (Corollary 3). However, note that they do not subsume the query complexity result (Theorem 1) regarding CHN.

[4]We define the problem CHN through two distributions (planted versus noise) over instances with overlapping supports, which means technically it is not even a promise problem. However, discarding the negligible sets of planted and noise samples on which the QMA verifier's respective acceptance and rejection guarantees fail yields an actual promise problem with the stated result.

**Quantum State/Proof Advantage** Our result shows that *quantum state advantage*, or *quantum proof advantage*, can exist even when classical computers can efficiently solve every problem that admits quantum verification with a classical witness ($\mathsf{P} = \mathsf{QCMA}$). While our problem itself has no direct connection to the Forrelation problem [Aar10, AA15], in some sense it can be thought of as a quantum-state analogue of it, since Forrelation has been used to show that *quantum computational advantage* can exist even when $\mathsf{P} = \mathsf{NP}$ [RT22, AIK22].

### 1.1.2 Cryptography

We use CHN to construct quantum cryptographic primitives in a world where $\mathsf{P} = \mathsf{QCMA} = \mathsf{QCMAH}$, and hence there is no quantum computational advantage, quantum-computable one-way functions or even efficiently verifiable one-way puzzles [CGG24]; answering an open question of Kretschmer, Qian, and Tal [KQT24].

**Theorem 3** (Informal, Theorem 19)**.** *There exists a classical oracle $\mathcal{O} : \{0,1\}^* \to \{0,1\}$ such that*

$$\mathsf{P}^{\mathcal{O}} = \mathsf{QCMA}^{\mathcal{O}} = \mathsf{QCMAH}^{\mathcal{O}},$$

*and relative to $\mathcal{O}$ the following primitives exist with correctness and security:*

- *Pseudorandom states (PRS) with classical keys, secure for any polynomial number of copies.*
- *Copy-protected public-key encryption (PKE) with quantum decryption keys and classical public keys and ciphertexts.*
- *Copy-protected pseudorandom functions (PRF) with quantum secret keys, pseudorandom against quantum evaluation queries.*
- *Signature schemes with quantum signing keys and either classical verification keys and quantum signatures or quantum verification keys and classical signatures.*

We emphasize that, unlike [KQT24], our PKE construction also satisfies *copy-protection*, a stronger security requirement that has been extensively studied [Aar09, ALL+20, GZ20, CLLZ22, CG25].

**Corollaries** Through the known black-box constructions in cryptography, we also obtain that the following exist relative to the same oracle:

- Collusion-resistant public-key quantum money (PKQM) with classical verification keys [AC13],
- Bounded-collusion-resistant copy-protected PKE and PRF schemes [CGK+26],
- Pseudorandom unitaries with a quantum key [MH25],
- (Inefficient-verifier) one-way puzzles [KT24],
- One-way state generators with a classical key [MY22],
- Signature scheme with quantum verification keys and classical signing keys and signatures, secure against a bounded number of signing queries [MY24].

We refer the reader to the Microcrypt Zoo [Sat] for more implications of these primitives.

**Quantum State Advantage in Cryptography** It is not hard to see that when $\mathsf{P} = \mathsf{QCMA}$, there is no quantum *computational* advantage: Any problem that can be solved efficiently by a quantum computer can also be solved efficiently classically, and any efficient quantum computational (even interactive) process that holds no quantum memory can be efficiently simulated by a classical computer[5]. Thus, we obtain cryptography from quantum state advantage even if there is no quantum computational advantage. Note that this is in contrast to the results of [KQST23, KQT24], which, as discussed, necessarily require quantum computational advantage ($\mathsf{P} \neq \mathsf{BQP}$).

Aside from [Kre21, KQST23, KQT24], very recently, Gulati, Khurana, and Tomer [GKT26] ruled out fully black-box constructions of quantum-computable one-way functions from PRUs. However, they do not show that PRUs can exist even if quantum-computable one-way functions do not exist, or if $\mathsf{P} = \mathsf{QCMA}$.[6]

## 1.2 Organization

In Section 2, we give a technical overview of the paper. See Section 2.6 for the detailed organization and Section 2.9 for the AI methodology.

In Section 2.7 we discuss some future directions and in Section 2.8 we discuss some more related work.

[5] There can be some efficient quantum (sampling) processes that can be *inefficiently* distinguished [Aar14, MSY25].

[6] See [BBF13, GKT26] for a discussion on the difference between fully black-box separations and oracle separations.

## 2 Technical Overview

In this section, we give a technical overview of our work, explaining our techniques and providing some intuition. Formal definitions, statements, and proofs can be found in the relevant sections.

**Complexity** In this overview, we focus on our query complexity[7] result about CHN to highlight the main techniques that also underlie our cryptographic results. Through (somewhat) routine techniques, such query complexity results can be converted to oracle separations (see, e.g., [RT22, AIK22, Zha25, BHNZ26, BHV26]). Indeed, our oracle separation showing that $\mathsf{QMA}^{\mathcal{O}}$ *is strictly larger than* $\mathsf{P}^{\mathcal{O}} = \mathsf{QCMAH}^{\mathcal{O}}$ *relative to some fixed oracle* $\mathcal{O}$ (Theorem 11), follows from this with small additions (which we provide for reference in Appendix J). As mentioned before, our cryptography results actually subsume this oracle separation, but not the query complexity result for CHN.

In Section 2.1, we briefly recall the model. Then Section 2.2 we introduce our problem CHN and state our query complexity result. We then give an overview of its proof in Section 2.3.

**Cryptography** We introduce our function encoding, PKE construction and collapse oracle in Section 2.4. We then give the proof overview in Section 2.5, first assuming that the required states are supplied and then explaining their preparation using classical oracles.

**Detailed Organization and Further Discussion** We give a detailed map of the paper in Section 2.6. Then we discuss future directions in Section 2.7, and review further related work in Section 2.8.

### 2.1 Model

We now briefly recall the query complexity model, the familiar readers may skip. See Section 3.2 for formal definitions.

**Algorithms and Machines** An algorithm is a classical or quantum procedure whose output may be probabilistic, whereas a machine specifies a partial decision function with values in $\{0, 1, \perp\}$. We call an algorithm or machine *query-efficient* if it makes polynomially many queries in total to its oracles, each on an *address* (also called a *query input*) of polynomial length. We allow unbounded computation outside oracle queries. Usually, we will use the word algorithm for query-efficient ones.

**Query Complexity Model** An algorithm or machine accesses its input as a finite Boolean oracle $\mathcal{O} : \{0,1\}^{p(\lambda)} \to \{0,1\}$, where $p$ is a fixed polynomial and $N = 2^{p(\lambda)}$ is size of the oracle's truth table. Resource bounds, such as the number of queries and the witness size, are measured in $\lambda$ rather than in $N$, so *polynomial* means polynomial in $\lambda$. We also consider oracles with domain $\mathcal{X}_\lambda$, writing $\mathcal{O} : \mathcal{X}_\lambda \to \{0,1\}$. Besides its input oracle, an algorithm or machine may have oracle access to another machine and receive a classical auxiliary input $z$ , supplied directly rather than through oracle access.[8] Unless otherwise specified, witnesses and auxiliary inputs have polynomial size. We use the prefix BB- (for *black-box*), as in BB-QMA and BB-QCMAH, to distinguish these query-model classes from their usual time-bounded counterparts.

[7] Also called *black-box* [BBC+98, AA11, RT22] or *oracle-input* [Zha25] model.

[8] The auxiliary input allows a higher-level machine to query a lower-level machine on different strings while both access the same input oracle.

**BB-QCMA and BB-QCMAH in Query Complexity Model** A BB-QCMA machine $M$ is specified by a query-efficient quantum algorithm $V$, called the *verifier* that additionally receives *classical witness* $w$, and is defined as

$$M^{\mathcal{O}}(z) = \begin{cases} 1, & \exists w : \Pr\left[V^{\mathcal{O}}(z,w) = 1\right] \geq 2/3, \\ 0, & \forall w : \Pr\left[V^{\mathcal{O}}(z,w) = 1\right] \leq 1/3, \\ \perp, & \text{otherwise.} \end{cases}$$

Following Agarwal and Ben-David [ABD26], we define the BB-QCMA hierarchy using *completions* of partial functions.[9] For $f : \{0,1\}^* \to \{0,1,\perp\}$, we define its set of Boolean completions by

$$\mathrm{Comp}(f) = \left\{ \widetilde{f} : \{0,1\}^* \to \{0,1\} \;\middle|\; \forall z,\ f(z) \neq \perp \implies \widetilde{f}(z) = f(z) \right\}.$$

Starting with $\mathsf{BB\text{-}QCMAH}_1 = \mathsf{BB\text{-}QCMA}$, for $d \geq 2$ a $\mathsf{BB\text{-}QCMAH}_d$ machine $M_d$ is specified by a query-efficient quantum verifier $V_d$ and a $\mathsf{BB\text{-}QCMAH}_{d-1}$ machine $M_{d-1}$, and is defined as

$$M_d^{\mathcal{O}}(z) = \begin{cases} 1, & \forall \widetilde{f} \in \mathrm{Comp}(M_{d-1}^{\mathcal{O}})\ \exists w : \Pr\left[V_d^{\mathcal{O},\widetilde{f}}(z,w) = 1\right] \geq 2/3, \\ 0, & \forall \widetilde{f} \in \mathrm{Comp}(M_{d-1}^{\mathcal{O}})\ \forall w : \Pr\left[V_d^{\mathcal{O},\widetilde{f}}(z,w) = 1\right] \leq 1/3, \\ \perp, & \text{otherwise.} \end{cases}$$

We define the BB-QCMA hierarchy BB-QCMAH as union of constant levels $d$,

$$\mathsf{BB\text{-}QCMAH} = \bigcup_{d \geq 1} \mathsf{BB\text{-}QCMAH}_d.$$

## 2.2 The CHN Problem and Our Query Complexity Result

**Starting point: CHS Problem** Our starting point is the *Codeword-Hash Structure* (CHS) problem recently introduced by Cakan [Cak26], to separate BB-QMA from BB-QCIP, which in turn is based on the *Code Intersection Subset Size* problem introduced by the elegant classical-oracle QMA–QCMA separation of Bostanci, Huang and Vaikuntanathan [BHV26]. Both problems build on the Yamakawa-Zhandry code intersection problem [YZ24] where we consider the symbol-wise hash function $H(a) = H_1(a_1)||\ldots||H_q(a_q)$ and a *suitable* linear code $C \in \Sigma^q$, and the task is to, for a given image $y \in \{0,1\}^q$, find a codeword $v \in C$ such that $H(v) = y$. Indeed, many QMA–QCMA separations rely on this same problem [LLPY24, BDK24, CCHY26]. In particular, [BHV26, CCHY26, Cak26] use the *biased* version introduced in [GGJL25].

More precisely, CHS is defined as follows. Let $C_\lambda \subseteq \Sigma_\lambda^{q_\lambda}$ be a suitable linear code, $q(\lambda) = \mathsf{poly}(\lambda)$ and we sample each value of the predicates $H_i : \Sigma_\lambda \to \{0,1\}$ independently as

$$H_i(a) \leftarrow \mathrm{Bernoulli}(\lambda^{-4}) \qquad (i \in [q_\lambda],\ a \in \Sigma_\lambda),$$

We then define the *codeword-hash relation* over $\mathcal{X}_\lambda = \Sigma_\lambda^{q_\lambda} \times \{0,1\}^\lambda$ by

$$\mathcal{P}_H = \left\{ (v,y) \in \mathcal{X}_\lambda : v \in C_\lambda, H(v) = H_1(v_1)\| \cdots \| H_{q_\lambda}(v_{q_\lambda}) = y \| 0^{q_\lambda - \lambda} \right\}. \tag{1}$$

Then, the CHS problem is to distinguish the *codeword-hash structure* distribution $\mathcal{O} = \mathbf{1}_{\mathcal{P}_H}$ (i.e. $\mathcal{O}(v,y) = 1$ iff $v \in C_\lambda$ and $H(v) = y\|0^{q_\lambda - \lambda}$), from the Bernoulli noise oracle $\mathcal{O} = B$, where $B(v,y) \leftarrow \mathrm{Bernoulli}(\delta)$ independently for $(v,y) \in C_\lambda \times \{0,1\}^\lambda, \delta = 2^{-\lceil \lambda/12 \rceil}$.

[9] See [ABD26] for a discussion on the choice of this definition for the BB-QCMA hierarchy.

**Quantum versus classical witnesses.** As observed by [Liu23], the code intersection problem satisfies a useful *equivocation* property: Using only the *predicate-preimage state* $|\Psi_H\rangle \propto \bigotimes_{i=1}^{q_\lambda} \left( \left( \sum_{a\in\Sigma_\lambda : H_i(a)=0} |a\rangle \right) \otimes \left( \sum_{a\in\Sigma_\lambda : H_i(a)=1} |a\rangle \right) \right)$, the decoding algorithm of [YZ24] can find a codeword $v \in C$ with $H(v) = y$ for any $y$, without querying $H$. Crucially, the same state $|\Psi_H\rangle$ works for any $y$ value. Based on this idea and a suitable choice of a code; [BHV26] shows that, using $|\Psi_H\rangle$ as the witness, we can check the codeword-hash structure in an input (such as $\mathcal{O} = \mathbf{1}_{\mathcal{P}_H}$ above), simply by sampling a random $y \leftarrow \{0,1\}^\lambda$ and running the decoder of [YZ24] for $y\|0^{q_\lambda - \lambda}$, and checking if the output $v$ satisfies $\mathcal{O}(v, y) = 1$. However, since a classical fixed witness $w$ can only encode a small amount of pairs $(v, y)$, they show that if the code satisfies sufficiently strong combinatorial properties, such structure cannot be checked reliably with classical witnesses, proving their QMA–QCMA separation. As we discuss later on, [Cak26] defines *contractive simultaneous incompressibility* property for oracles that codifies this notion, and extends this proof idea beyond classical witnesses/BB-QCMA to *classical-interactive proofs with quantum verifiers* (BB-QCIP).

**BB-coNP Attack for CHS** While CHS separates BB-QMA from BB-QCIP, unfortunately it is in BB-coNP $\subseteq$ BB-$(\mathsf{NP}^{\mathsf{NP}})$: To recognize the structured case, BB-coNP can simply check that no codeword $v$ is accepted for two distinct targets $y \neq y'$. Since $H(v)$ determines $y$, this holds for $\mathbf{1}_{\mathcal{P}_H}$, whereas $B$ accepts exponentially many targets for any fixed codeword with overwhelming probability. In fact, small tweaks to this problem will not help: If a noticeable relative gap remains in the number of accepted targets $N_{\mathcal{O}}(v) := |\{y \in \{0,1\}^\lambda : \mathcal{O}(v, y) = 1\}|$ for a fixed codeword $v$, we can use Stockmeyer's approximate-counting algorithm [Sto83] to approximate this count and decide the problem in BB-$(\mathsf{BPP}^{\mathsf{NP}}) \subseteq$ BB-QCMAH$_2$.

**Our Problem: CHN** To remove this counting gap, we use noise in both cases, introducing the problem *Codeword-Hash pairs in Noise* (CHN):

**The CHN problem.** $\beta = 2^{-\lceil\sqrt{\lambda}\rceil}$

| Planted (YES) distribution: $\mathsf{O}_{\mathsf{plant},\lambda}$ | Noise (NO) distribution: $\mathsf{O}_{\mathsf{noise},\lambda}$ |
|---|---|
| $B(v,y) \leftarrow \text{Bernoulli}(\beta) \quad (v,y) \in \mathcal{X}_\lambda$ | $B(v,y) \leftarrow \text{Bernoulli}(\beta) \quad (v,y) \in \mathcal{X}_\lambda$ |
| $H_i(u) \leftarrow \text{Bernoulli}(\lambda^{-4}) \quad i \in [q_\lambda], u \in \Sigma_\lambda$ | |
| $\mathcal{O}^{\mathsf{plant}}_{H,B}(v,y) := \begin{cases} 1, & (v,y) \in \mathcal{P}_H, \\ B(v,y), & \text{otherwise} \end{cases}$ | $\mathcal{O}^{\mathsf{noise}}_{B}(v,y) := B(v,y).$ |

We emphasize that the algorithms access only the sampled oracle $\mathcal{O}^{\mathsf{plant}}_{H,B}$ or $\mathcal{O}^{\mathsf{noise}}_{B}$, with no separate access to $H$ or $B$. With overwhelming probability, the noise contributes exponentially many accepted targets for a fixed codeword $v$ in either case, whereas $\mathcal{P}_H$ adds at most one. Thus, the additional pair has only an exponentially small relative effect on $N_{\mathcal{O}}(v)$, eliminating the noticeable gap used by the counting attack. Of course, merely eliminating the counting attack does not prove BB-QCMAH hardness. However, looking ahead, having noise in both cases will allow us to use a *switching lemma* [Hås86, Kel21] to remove witnesses one by one from depth-$d$ BB-QCMAH machines and reducing its depth, akin to some classical oracle separations regarding PH [Hås86, RST15]. But first, to make the switching lemmas from classical complexity applicable, we will *classicalize* our

quantum verifiers. More precisely, we will use the sparsity of the noise ($\beta = 2^{-\lceil\sqrt{\lambda}\rceil}$) and the properties of the codeword-hash relation to show *contractive simultaneous incompressibility*, and then use this to show how to simulate quantum verifiers using *deterministic-classical queries*. Finally, we note that even with our addition of the noise to the YES distribution, it is not too hard to prove that BB-QMA completeness and soundness still holds with the same verifier as [BHV26, Cak26].

**Our result.** We obtain the following query complexity result for CHN.

**Theorem 4** (Informal, Theorems 8 and 10)**.** *There is a* BB-QMA *machine that accepts planted samples and rejects noise samples, except on a negligible fraction from each distribution. For every constant $d$ and query-efficient* $\mathsf{BB\text{-}QCMAH}_d$ *machine $M_d$ satisfying $M_d^{\mathcal{O}}(1^\lambda) \neq \perp$ almost surely under both distributions,*

$$\left| \Pr_{\mathcal{O}\leftarrow \mathsf{O}_{\mathsf{plant},\lambda}} [M_d^{\mathcal{O}}(1^\lambda) = 1] - \Pr_{\mathcal{O}\leftarrow \mathsf{O}_{\mathsf{noise},\lambda}} [M_d^{\mathcal{O}}(1^\lambda) = 1] \right| \leq \mathsf{negl}(\lambda). \tag{2}$$

Although the distributions have overlapping supports, as mentioned beofre, removing the negligible exceptional sets where BB-QMA acceptance (for planted distribution) and rejection (for noise distribution) guarantee fails yields a promise problem in $\mathsf{BB\text{-}QMA} \setminus \mathsf{BB\text{-}QCMAH}$ in the query complexity model (Theorem 10).

### 2.3 Complexity Proof Overview

Throughout the whole proof, we fix the prefix predicates $(H_1, \ldots, H_\lambda)$ to any fixed one that satisfies the following: $H$ is a *good hash* (we can invert all $y$) with overwhelming probability over the sampling of the suffix predicates $H_{\lambda+1}, \ldots, H_q$ (Lemma 7). Indeed, the sampling of the prefix in both $\mathsf{O}_{\mathsf{plant},\lambda}, \mathsf{O}_{\mathsf{noise},\lambda}$ already satisfies this property with overwhelming probability. We implicitly allow all algorithms to depend on this choice.

Our proof of Theorem 4 consists of three main steps.

1. **Deterministic-classical-query simulation of BB-QCMA Verifier:** First, using the *contractive simultaneous incompressibility* property of our oracle distributions, we show the following: With a query-efficient simulator algorithm $\mathsf{Sim}_V^{\mathcal{O}}(z, w)$ that makes *deterministic-classical* queries (with randomized post-processing), we can simulate (for most oracles under either the planted or noise distribution) a BB-QCMA verifier's execution $V^{\mathcal{O}}(z, w)$ on any witness $w$ (Section 5).

2. **Removing the witness: DNF represention of the simulator and switching** Then, thanks to the fact that its queries are deterministic, classical and polynomially many, we represent the simulator's execution and the existential quantification of an accepting witness in a *small/polynomial width* disjunctive normal form (DNF) function of the oracle's bits. Note that even a single quantum query can require exponential-width DNF. Small-width DNF allows us to apply a *switching lemma* [Hås86, Kel21] to remove the existential quantification over witnesses. Thus, Step 1 then 2 *morally* turns the BB-QCMA machine into a BB-P machine, with a caveat that this machine will depend on some sections of the oracle. (Section 6).

3. **(Repeated) Depth reduction** Finally, this allows us to reduce the level/depth of a BB-QCMAH machine by one, by replacing the *deepest* BB-QCMA machine with a *witness-less* algorithm (Section 8). Then repeatedly applying the same steps (again, with caveats) completes our proof of Theorem 4, since any BB-QCMAH machine has constant number of levels (Section 9).

We note that switching lemmas are commonly used to remove witnesses or reduce depth [Hås86, RST15, Imp11, HN22, AIK22, ABD26]. However, apart from this high-level resemblance, our proof has no direct connection to the arguments in these works (see Section 2.8).

### 2.3.1 Deterministic-Classical-Query Simulation of BB-QCMA Verifiers

To bring ourselves to a setting where we can utilize classical complexity techniques, we first show how to simulate a BB-QCMA verifier $V$ using deterministic-classical queries. Our simulator will work simultaneously for all witnesses and for both oracle distributions. In fact, our simulation will work even when the description of $V$ depends on full descriptions of some suffix predicates $(H_i)_{i\in K}$ (where $K \subseteq \{\lambda+1,\dots,q\}$). As we see later on, this is crucial since our next step (*switching*) will introduce dependence of some of the suffix predicates, but for repeated depth reduction, we will need to apply simulation again even with such dependence.

Now we introduce some notation. For nonempty $I \subseteq \{\lambda+1,\dots,q_\lambda\}$, write

$$H_I = (H_i)_{i\in I}, \qquad \mathcal{P}_I = \{(v,y)\in\mathcal{X}_\lambda : v\in C_\lambda,\ H_i(v_i)=y_i\ (i\le\lambda),\ H_i(v_i)=0\ (i\in I)\}.$$

Set $\mathcal{P}_\varnothing = \mathcal{X}_\lambda$ and $\overline{K} = \{\lambda+1,\dots,q_\lambda\}\setminus K$.

We will prove the following simulation lemma[10].

**Lemma 1** (Informal, Theorem 5)**.** *Let $K \subseteq \{\lambda+1,\dots,q_\lambda\}$ satisfy $|\overline{K}| \ge \lfloor (q_\lambda-\lambda)/d\rfloor$ for a constant $d\ge 1$. Fix any $H_K$ and $B|_{\mathcal{X}_\lambda\setminus\mathcal{P}_K}$. For every verifier algorithm $V_{H_K,B|_{\mathcal{X}_\lambda\setminus\mathcal{P}_K}}$, there is a simulator $\mathsf{Sim}_{V_{H_K,B|_{\mathcal{X}_\lambda\setminus\mathcal{P}_K}}}$ making polynomially many deterministic-classical queries followed by randomized oracle-free computation, such that, with overwhelming probability over $H_{\overline{K}}$ and $B|_{\mathcal{P}_K}$,*

$$\left|\Pr\left[V^{\mathcal{O}}_{H_K,B|_{\mathcal{X}_\lambda\setminus\mathcal{P}_K}}(z,w)=1\right] - \Pr\left[\mathsf{Sim}^{\mathcal{O}}_{V_{H_K,B|_{\mathcal{X}_\lambda\setminus\mathcal{P}_K}}}(z,w)=1\right]\right| \le 1/100 \tag{3}$$

*simultaneously for every $z,w$ and both $\mathcal{O}\in\{\mathcal{O}^{\mathsf{plant}}_{H,B},\mathcal{O}^{\mathsf{noise}}_{B}\}$.*

The full theorem (Theorem 5) is more general, which is needed for our cryptography results later on.

**Contractive simultaneous incompressibility.** To prove our simulation lemma, we first recall the notion of *contractive simultaneous incompressibility*, from [Cak26].[11] Let $F$ be a set sampled according to some distribution and consider a $c(\lambda)$-bit classical witness $w$ that may depend arbitrarily on it (i.e. chosen by an unbounded sender who sees $F$). Then, given $w$, an unbounded receiver with no access to $F$ tries to output elements of $F$, and we allow the sender and this receiver to pre-share randomness independent of $F$. While the general version can have different parameters, in our version, we say $F$ satisfies *contractive simultaneous incompressibility* if every such sender and receiver, the probability that the receiver outputs

$$t_{\mathrm{inc}}(\lambda,c(\lambda)) = \left\lceil \frac{2(c(\lambda)+8\lambda^2)}{\sqrt{\lambda}} \right\rceil$$

distinct elements of $F$ is at most $2^{-8\lambda^2}$, for every polynomial $c(\lambda)$ and all sufficiently large $\lambda$.

[10] We note that a classical-query simulation lemma also appears in [Cak26] for the noise distribution and the *structure* distribution of CHS, but in a weaker form. In particular, it uses randomized queries (whereas deterministic queries is crucial for us) and it does not allow dependence of $V$ on sections of oracles (e.g. $H_K$), which is again crucial for us. Separately, a similar argument to our query-database size bounding approach used in our proof also appears in [BHV26] (and in a weaker form in [CGS26]).

[11] A similar notion is used implicitly in [BHV26], and a weaker version appears in [CGS26].

**Incompressibility in the planted and noise cases.** Using a suitable choice of the code $C$, we show that the set

$$F_K := \mathcal{P}_H \cup \{(v, y) \in \mathcal{P}_K : B(v, y) = 1\} \tag{4}$$

satisfies the above contractive simultaneous incompressibility bound, even after fixing $H_K$ and $B|_{\mathcal{X}_\lambda \setminus \mathcal{P}_K}$ (Corollary 1). This covers both distributions for our argument below, since $F_K$ contains every input in $\mathcal{P}_K$ at which either $\mathcal{O}^{\mathsf{plant}}_{H,B}$ or $\mathcal{O}^{\mathsf{noise}}_{B}$ answers 1.

**Our simulator.** Write $V = V_{H_K, B|_{\mathcal{X}_\lambda \setminus \mathcal{P}_K}}$, let $Q = \mathsf{poly}(\lambda)$ denote its number of queries, and let $t, m$ be polynomials (chosen later). For every set $E \subseteq \mathcal{P}_K$, define

$$\mathcal{O}_E(u) = \begin{cases} B(u), & u \notin \mathcal{P}_K, \\ \mathbf{1}_E(u), & u \in \mathcal{P}_K. \end{cases}$$

For $j \in [Q]$, let $\pi^{(j)}_{z,w,E}$ be the distribution defined as measuring the query input register for the $j$-th query during (purified) execution of $V^{\mathcal{O}_E}(z, w)$. Now independently sample $u^{(j,\ell)}_{z,w,E} \leftarrow \pi^{(j)}_{z,w,E}$ for $\ell \in [m], j \in [Q]$ and for every $z, w, E$ with $|E| < t$; and set $\mathbf{u}^{(j)}_{z,w,E} = (u^{(j,1)}_{z,w,E}, \ldots, u^{(j,m)}_{z,w,E})$. Thus, each list contains $m$ samples from the corresponding query, obtained from independent executions of $V^{\mathcal{O}_E}(z, w)$. We write $\mathbf{U}$ for the full collection of these samples.

Essentially, our simulator will try to estimate a database $D$ that records the addresses of the non-zero bits of our oracle, *that $V$ could query*. Intuitvely, we will be able to do so due to the simultaneous incompressibility property above, since the queries of $V$ with non-zero output will be concentrated in a small list.

$\underline{\mathsf{Sim}^{\mathcal{O}}_V(z, w)}$

1. Set $D = \varnothing$.
2. While $|D| < t$:
   - 2.1. For $j = 1, \ldots, Q$, consider $(u^{(j,\ell)}_{z,w,D})_{\ell \in [m]}$ in order, and query $\mathcal{O}(u)$ for each one that has $u \in \mathcal{P}_K \setminus D$. As soon as a query returns 1, add its input $u$ to $D$ and go back to Step 2.
   - 2.2. If all $Q$ lists have been completed without adding to $D$, go to Step 3.
3. Without further oracle queries, compute $p_D$, value of $\Pr\left[V^{\mathcal{O}_D}(z, w) = 1\right]$ with additive error at most $\lambda^{-2}$, and output Bernoulli$(p_D)$.

Note that $\mathsf{Sim}$ makes at most $tQm$ oracle queries.

**Showing $|D| < t$ by incompressibility.** We first show that the probability that the simulator reaches $|D| = t$ for any $(z, w)$ is negligible in total. To show this, we willl use our simulator as an encoding scheme for the incompressibility game. Consider the incompressibility game for $F = F_K$, with $(\mathbf{u}^{(j)}_{z,w,D})_{\substack{j \in [Q], z, w, D: \\ |D| < t}}$ as shared randomness, which is independent of $F$. Given $F$, the sender tries every $(z, w)$ and subset $S \subseteq F$, running the simulator while answering its queries with $\mathcal{O} = \mathbf{1}_S$. If a run reaches $|D| = t$, it sends the $c(\lambda)$-bit message

$$(z, w, k_1, \ldots, k_t), \qquad c(\lambda) = |z| + |w| + t \left\lceil \log_2(tQm) \right\rceil,$$

where $k_i \leq tQm$ is the query number of $\mathsf{Sim}$ at which the $i$th element was added to $D$ during the execution. Now the receiver runs the simulator, answering 1 exactly at these query numbers and also records the query input as $x_i$. It answers 0 elsewhere. Finally, the receiver outputs $x_1, \ldots, x_t$. Note that all $x_i$ lie in $F$ and are distinct, since inputs added to $D$ are never queried again. Thus, choosing polynomial $t$ with $t_{\mathrm{inc}}(\lambda, c(\lambda)) \leq t$, incompressibility bounds the probability of reaching $|D| = t$ by $2^{-8\lambda^2}$.

**Simulation Accuracy.** We now argue simulation accuracy as follows: Any non-zero set of addresses of our oracle will either be eventually added to the database $D$, or the verifier's query weight on that address set is small enough that not having it in the database makes a small difference. Let $D_{\mathrm{fin}}$ be the final database during execution of $\mathsf{Sim}$, and assume $|D_{\mathrm{fin}}| < t$ as above. Write $W(E)$ for the total query weight of $V^{\mathcal{O}_E}(z, w)$ on inputs where $\mathcal{O}_E$ and $\mathcal{O}$ differ, summed over all $Q$ queries. We claim that, with probability $1 - \mathsf{negl}(\lambda)$ over $\mathbf{U}$, we have $W(D_{\mathrm{fin}}) \leq \frac{1}{\lambda^2 Q}$, simultaneously for all $z, w$ and both oracles.

Then, by the hybrid bound ([BBBV97]), replacing $\mathcal{O}_{D_{\mathrm{fin}}}$ with $\mathcal{O}$ changes the verifier's acceptance probability by $O(1/\lambda)$. Our simulator outputs Bernoulli$(p_{D_{\mathrm{fin}}})$, where $p_{D_{\mathrm{fin}}}$ approximates the acceptance probability of $V^{\mathcal{O}_{D_{\mathrm{fin}}}}(z, w)$ within $\lambda^{-2}$, so its simulation error is $O(1/\lambda) + \lambda^{-2} < 1/100$ for sufficiently large $\lambda$.

Now to prove our claim, fix $z, w, \mathcal{O}$ and any database $E$ with $|E| < t$. Now observe that each sample $u^{(j,\ell)}_{z,w,E}$ is obtained by measuring the query input register at query $j$ in an independent execution of $V^{\mathcal{O}_E}(z, w)$. Thus, any input $u$ with query weight $a$ at this step is sampled with probability $a$. By independence, the probability that all $Qm$ samples avoid the inputs where $\mathcal{O}_E$ and $\mathcal{O}$ differ is at most $e^{-mW(E)}$.

Now note that for the simulator to terminate with $D_{\mathrm{fin}} = E$ for some $E$, it must be the case that during the final loop, all $Qm$ values $(u^{(j,\ell)}_{z,w,E})_{j\in[Q],\ell\in[m]}$ avoid the inputs where $\mathcal{O}_E$ and $\mathcal{O}$ differ (since we have already ruled out the other termination condition $|D_{\mathrm{fin}}| = t$). Otherwise, $\mathsf{Sim}$ would query an input $u \notin E$ with $\mathcal{O}(u) = 1$ and add it to $D$, and hence that would not be the last loop (recall that we only stop when there is no new additions). Thus, for each fixed $E$ with $|E| < t$, we have $\Pr_{\mathbf{U}}[D_{\mathrm{fin}} = E] \leq e^{-mW(E)}$. A union bound over all such $E$ with $W(E) > 1/(\lambda^2 Q)$, all $z, w$, and both oracles proves our claim about $W(D_{\mathrm{fin}})$, for sufficiently large polynomial $m$.

**Making $\mathsf{Sim}$ deterministic query.** We now fix a deterministic choice of $\mathbf{U} = (\mathbf{u}^{(j)}_{z,w,D})_{z,w,D,j}$. For any $\mathbf{U}$, let $\varepsilon(\mathbf{U})$ be the probability over $H_{\overline{K}}, B|_{\mathcal{P}_K}$ that acceptance error of $\mathsf{Sim}$ exceeds $1/100$ for some $z, w$ and either oracle. Choose the lexicographically first $\mathbf{U}^\star$ such that $\varepsilon(\mathbf{U}^\star) \leq \mathbb{E}_{\mathbf{U}}[\varepsilon(\mathbf{U})] \leq \mathsf{negl}(\lambda)$. Now we hardwire[12] this into our $\mathsf{Sim}$. Note that $\mathbf{U}^\star$ depends only on $V, H_K, B|_{\mathcal{X}_\lambda \setminus \mathcal{P}_K}$ that are already fixed; beyond that, it only depends on the distributions of our oracles and not actual sampled oracles.

### 2.3.2 Removing the witness: From deterministic-classical queries to DNFs and Switching

**Small-Width DNF Represention** Now observe that since $\mathsf{Sim}_{V_{H_K, B|_{\mathcal{X}_\lambda \setminus \mathcal{P}_K}}}$ (Lemma 1) makes at most $k = \mathsf{poly}(\lambda)$ *deterministic-classical* queries, for fixed $z, w$ its output depends only on $k$ oracle bits (beyond dependence on $H_K$, $B|_{\mathcal{X}_\lambda \setminus \mathcal{P}_K}$). Since this simulator is adaptive, the addresses of the bits can depend on the earlier queried bits, but still, for any auxiliary input $z$ and witness $w$,

[12] Recall that in our query complexity model, the only restriction is number of queries, so we can have such exponential length description.

whether $\mathsf{Sim}^{\mathcal{O}}_{V_{H_K,\,B|_{\mathcal{X}_\lambda\setminus\mathcal{P}_K}}}(z,w)$ accepts (with some probability[13] $p$) can be written as a disjunction of width-$k$ conjunctions of the oracle bits: Simply check if there is a *query transcript*[14] that matches the oracle and such that the simulator accepts with probability $p$. Note that this is in stark contrast to quantum-query algorithms, since even a single deterministic quantum query can depend on all of the oracle bits.

Further, the existence of an accepting witness can be written as another layer of disjunction. More precisely, for some query outputs $a_1,\dots,a_k$, let $p_{z,w,a}$ be the acceptance probability of $\mathsf{Sim}_{V_{H_K,\,B|_{\mathcal{X}_\lambda\setminus\mathcal{P}_K}}}(z,w)$ and denote its query inputs during this execution as $u_1^{z,w,a},\dots,u_k^{z,w,a}$. Then we have

$$f_{z,H_K,B|_{\mathcal{X}_\lambda\setminus\mathcal{P}_K}}(\mathcal{O}) := \mathbf{1}\left[\exists w : \Pr\left[\mathsf{Sim}^{\mathcal{O}}_{V_{H_K,\,B|_{\mathcal{X}_\lambda\setminus\mathcal{P}_K}}}(z,w)=1\right]\geq\tfrac{1}{2}\right] = \bigvee_{\substack{w,a:\\ p_{z,w,a}\geq 1/2}}\bigwedge_{j=1}^{k}[\mathcal{O}(u_j^{z,w,a})=a_j]. \tag{5}$$

**Removing the witness by switching.** While $f_{z,H_K,B|_{\mathcal{X}_\lambda\setminus\mathcal{P}_K}}$ has polynomial *width*, it still depends on exponentially many bits of the oracle $\mathcal{O}$. However, since it is a DNF, switching lemmas [Hås86, Kel21] can help us compute $f_{z,H_K,B|_{\mathcal{X}_\lambda\setminus\mathcal{P}_K}}$ using a polynomial number of classical-queries, after the oracle bits at all positions but a random small fraction are fixed (to independent random values). Note that here *small* means as a fraction of whole oracle table size $N$, but still superpolynomial in $\lambda$, so this is indeed non-trivial.

More precisely, Kelley's switching lemma [Kel21] applies to any fixed width-$k$ DNF and a random set $S$ of *free positions*, provided that it satisfies a *joint sparsity condition*: For an integer $Q\geq 1$ and $0<p\leq 1/(16k)$, every fixed set $A$ of at most $k+Q$ positions satisfies

$$\Pr_S[A\subseteq S]\leq p^{|A|}.$$

After independently and uniformly fixing the oracle bits outside our *free positions* $S$, with probability at least $1-2(8kp)^Q$, a classical-query deterministic algorithm whose description depends only on the position set $S$ and the fixed outside bits can compute the DNF value using fewer than $Q$ queries (for any oracle that has our fixed values outside $S$).

Since $\mathsf{Sim}_{V_{H_K,\,B|_{\mathcal{X}_\lambda\setminus\mathcal{P}_K}}}$ and thus $f_{z,H_K,B|_{\mathcal{X}_\lambda\setminus\mathcal{P}_K}}$ depend on $H_K$ and $B|_{\mathcal{X}_\lambda\setminus\mathcal{P}_K}$, we will consider $H_{\overline{K}}$ and $B|_{\mathcal{P}_K}$ for our switching lemma. Let $J\subseteq\{\lambda+1,\dots,q_\lambda\}\subseteq\overline{K}$ be such that $|J|\geq\lfloor(q_\lambda-\lambda)/d\rfloor$ for a constant $d\geq 1$. These will be our *still fresh suffix indices* that we will use for $S$ in the switching lemma. We consider any such $J$ instead of simply $\overline{K}$ so that we are only *spending* some of the remaining fresh suffix predicates: Observe that our switching lemma will introduce dependence on $H_J$, but in the depth reduction step later on we will need to apply it multiple times. Now observe that our oracles have the form needed for switching. $\mathcal{O}_B^{\mathsf{noise}}=B$ and $\mathcal{O}_{H,B}^{\mathsf{plant}}=B\vee\mathbf{1}_{\mathcal{P}_H}$ agree with $B$ on $\mathcal{P}_K\setminus\mathcal{P}_{K\cup J}$, and the fresh predicates $H_J$ give

$$\Pr_{H_J}[u_1,\dots,u_t\in\mathcal{P}_{K\cup J}]\leq 2^{-\Omega(\lambda t)}\qquad\text{for distinct } u_1,\dots,u_t\in\mathcal{P}_K,\quad t\leq 2^\lambda, \tag{6}$$

So, in the switching theorem above[15], we can use $S=\mathcal{P}_{K\cup J}$ as our free positions and randomly *fix* outputs of the other (i.e. $\mathcal{P}_K\setminus\mathcal{P}_{K\cup J}$) positions in $\mathcal{P}_K$ to these common $B$-values, recalling

[13] Due to randomized post-processing

[14] Simulator's adaptive queries and their answers

[15] Technically, Kelley's lemma in its original form only applies to independent uniform random fixing outside $S$, whereas our $B$ has biased bits. But the biased version of their lemma is an easy extension (see Theorem 7).

that the common answers on $\mathcal{P}_K \setminus \mathcal{P}_{K\cup J}$ are independent Bernoulli($\beta$) bits in both the planted and noise distributions. Thus, we obtain a classical-query deterministic algorithm $T$ with the same description for both the planted and noise oracles. This is precisely why adding noise to both the YES and NO distributions helps, as mentioned before.

**Lemma 2** (Informal, Theorem 7)**.** *Let $K, J \subseteq \{\lambda+1, \ldots, q_\lambda\}$ be disjoint, with $|J| \geq \lfloor (q_\lambda - \lambda)/d \rfloor$ for a constant $d \geq 1$. Fix any $H_K$ and $B|_{\mathcal{X}_\lambda \setminus \mathcal{P}_K}$, and fix the simulator $\mathsf{Sim}_{V_{H_K}, B|_{\mathcal{X}_\lambda \setminus \mathcal{P}_K}}$ from Lemma 1.*

*For any polynomially bounded integer $Q \geq 1$ there is a classical-query deterministic algorithm $T_{H_K, H_J, B|_{\mathcal{X}_\lambda \setminus \mathcal{P}_{K\cup J}}}$ taking no witness and making fewer than $Q$ queries, all in $\mathcal{P}_{K\cup J}$. For each fixed auxiliary input $z$, with probability at least $1 - 2^{-\Omega(Q\sqrt{\lambda})}$ over $H_J$ and $B|_{\mathcal{P}_K \setminus \mathcal{P}_{K\cup J}}$,*

$$\begin{aligned} T^{\mathcal{O}}_{H_K, H_J, B|_{\mathcal{X}_\lambda \setminus \mathcal{P}_{K\cup J}}}(z) &= f_z(\mathcal{O}) \\ &= \mathbf{1}\left[\exists w : \Pr\left[\mathsf{Sim}^{\mathcal{O}}_{V_{H_K}, B|_{\mathcal{X}_\lambda \setminus \mathcal{P}_K}}(z, w) = 1\right] \geq \tfrac{1}{2}\right] \end{aligned}$$

*for both $\mathcal{O} \in \{\mathcal{O}^{\mathsf{plant}}_{H,B}, \mathcal{O}^{\mathsf{noise}}_{B}\}$.*

*We emphasize that unlike the simulator $\mathsf{Sim}_{V_{H_K}, B|_{\mathcal{X}_\lambda \setminus \mathcal{P}_K}}$, $T_{H_K, H_J, B|_{\mathcal{X}_\lambda \setminus \mathcal{P}_{K\cup J}}}$ receives no witness.*

The full theorem (Theorem 7) is more general, which is needed for our cryptography results later on.

### 2.3.3 (Repeated) Depth reduction

**Warm-up: BB-QCMA Hardness** Observe that for the first level of BB-QCMAH, that is, for a fixed BB-QCMA machine $M_1$ with verifier $V$, we are already done. Since $M_1$ does not depend on the oracle, Lemma 1 with $K = \varnothing$ gives, except with negligible probability, $f_z(\mathcal{O}) = \mathbf{1}\left[\exists w : \Pr\left[\mathsf{Sim}^{\mathcal{O}}_{V}(z, w) = 1\right] \geq \frac{1}{2}\right] = M_1^{\mathcal{O}}(z)$ for both oracles $\mathcal{O} \in \{\mathcal{O}^{\mathsf{noise}}_{B}, \mathcal{O}^{\mathsf{plant}}_{H,B}\}$ whenever $M_1^{\mathcal{O}}(z) \neq \perp$. Using $J = \{\lambda+1, \ldots, q\}$ in Lemma 2 with $Q = 1$ and $z = 1^\lambda$ then gives a deterministic classical algorithm $T = T_{H, B|_{\mathcal{X}_\lambda \setminus \mathcal{P}_H}}$ that makes no queries and has the same description for both oracles and satisfies, except with negligible probability, $M_1^{\mathcal{O}}(1^\lambda) \neq \perp \implies T^{\mathcal{O}}(1^\lambda) = M_1^{\mathcal{O}}(1^\lambda)$ for both oracles. Therefore, $M_1$ cannot distinguish the oracles and we are done. For a general BB-QCMAH$_d$ machine, we will repeatedly do the same argument starting with the deepest BB-QCMA machine $M_1$, but more carefully: Note that already for a BB-QCMAH$_2$ machine defined by $V_2, M_1$, the above argument would be an issue since our replacement for $M_1$ depends on all of $H$, so we cannot apply another step of simulation or switching.

**General case (BB-QCMAH$_d$)** We now sketch the proof for the general case (see Theorem 10 and the full proofs in Sections 8 and 9.1). Consider a BB-QCMAH$_d$ machine $M_d$ with verifiers $V_1, \ldots, V_d$, where the deepest/bottom verifier $V_1$ queries only $\mathcal{O}$ and each $V_j$, for $j > 1$, also queries a Boolean completion of the lower machine $M^{\mathcal{O}}_{j-1}$ (defined by $V_1, \ldots, V_{j-1}$). Starting with the deepest level $M_1$, we will replace each machine by a classical-query deterministic algorithm $T_1$ taking no witness, then repeat for the next level. At each step, the replacement will agree with the non-$\perp$ outputs of the machine being replaced (except with negligible probability), which also preserves the higher levels' non-$\perp$ outputs by the Boolean-completion definition (Definition 4).

To leave fresh predicates for each step, partition the suffix indices into $d$ groups $J_1, \ldots, J_d$ whose sizes differ by at most one, and write

$$S_i = \bigcap_{j \leq i} \mathcal{P}_{J_j}, \qquad \mathcal{X}_\lambda = S_0 \supseteq S_1 \supseteq \cdots \supseteq S_d = \mathcal{P}_H.$$

At proof step $i$, $H_K$ and $B|_{\mathcal{X}_\lambda \setminus S_{i-1}}$ are already fixed. Let $K = J_1 \cup \cdots \cup J_{i-1}$. The remaining verifiers, from top to bottom, are $V_d, \ldots, V_i$. The bottom verifier $V_i$ queries $\mathcal{O}$ and, for $i > 1$, the previously constructed algorithm $T^{\mathcal{O}}_{i-1,\, H_K,\, B|_{\mathcal{X}_\lambda \setminus S_{i-1}}}$ (for $i = 1$, it queries only $\mathcal{O}$). We now remove $V_i$ and its witness. First, we incorporate $T_{i-1}$ into its code so that it only queries $\mathcal{O}$. Simulation (Lemma 1) gives us $\mathsf{Sim}_{i,\, H_K,\, B|_{\mathcal{X}_\lambda \setminus S_{i-1}}}$. For $i < d$, we then apply switching (Lemma 2) with $J = J_i$, choosing $Q = \mathsf{poly}(\lambda)$ large enough to union bound over all $2^{\mathsf{poly}(\lambda)}$ auxiliary inputs $z$. This gives us a classical-query deterministic algorithm $T_i = T_{i,\, H_K,\, H_{J_i},\, B|_{\mathcal{X}_\lambda \setminus S_i}}$ with the same description for both oracles, as guaranteed by Lemma 2. It takes no witness, makes fewer than $Q$ queries in $S_i$, and satisfies, except with negligible probability over $H_{\overline{K}}$ and $B|_{S_{i-1}}$, for all such $z$ and each $\mathcal{O} \in \{\mathcal{O}^{\mathsf{plant}}_{H,B}, \mathcal{O}^{\mathsf{noise}}_{B}\}$,

$$\exists w : \Pr\left[V_i^{\mathcal{O}, T^{\mathcal{O}}_{i-1}}(z, w) = 1\right] \geq \tfrac{2}{3} \quad \Longrightarrow \quad T_i^{\mathcal{O}}(z) = 1,$$

$$\forall w : \Pr\left[V_i^{\mathcal{O}, T^{\mathcal{O}}_{i-1}}(z, w) = 1\right] \leq \tfrac{1}{3} \quad \Longrightarrow \quad T_i^{\mathcal{O}}(z) = 0.$$

So, $T_i$ agrees with a Boolean completion of the current level-$i$ machine on every possible query, and so we can remove $V_i$ and its witness and continue with $V_{i+1}^{\mathcal{O}, T_i^{\mathcal{O}}}$ without changing the hierarchy's non-$\perp$ outputs (Definition 4).

Finally, for the last step $i = d$, take $Q = 1$, where only $z = 1^\lambda$ matters. The resulting $T_d$ makes no queries and has the same description for both oracles, so it outputs the same bit for both. Since our replacements preserve $M_d^{\mathcal{O}}(1^\lambda)$ whenever it is not $\perp$ (except with negligible probability), this proves Theorem 4.

### 2.4 Cryptographic Oracle and Constructions

We focus on our PKE result in this overview and refer the reader to the relevant sections for our other cryptography results.

We construct an oracle $\mathcal{O}$ such that $\mathsf{P}^{\mathcal{O}} = \mathsf{QCMA}^{\mathcal{O}}$ and there is a scheme $\mathsf{PKE}^{\mathcal{O}}$ with classical ciphertexts and public keys that satisfies the following strong guarantee called *copy-protection*: a QPT adversary $\mathcal{A}_0$ tries to split the quantum decryption key $|K\rangle$ into two registers $\mathsf{R}_1, \mathsf{R}_2$ for two noncommunicating adversaries $\mathcal{A}_1, \mathcal{A}_2$ that receive independent ciphertext challenges, and we require that their probability of both guessing their challenge bits is at most $1/2 + \mathsf{negl}(\lambda)$ (Definition 40).

**Intuition** On a very high level, our main idea is to use our problem $\mathsf{CHN}$ to encode a function $\mathfrak{F}$ (that provides us e.g. a copy-protected PKE construction) into an oracle $E$, and use our $\mathsf{BB\text{-}QMA}$ witness state $|\Psi_H\rangle$ to *unlock* it and achieve *functionality/correctness*. Then, using the tools underlying the $\mathsf{BB\text{-}QCMAH}$ hardness of $\mathsf{CHN}$, we will reduce attacks in presence of an oracle that collapses $\mathsf{QCMAH}$ to $P$, to QPT attacks with direct $\mathfrak{F}$ access. Indeed, [KQT24] uses the same idea, but instead with the Forrelation problem.

**Our Function Encoding and Decoding** To encode a function $\mathfrak{F}$, we modify $\mathsf{CHN}$ as follows: instead of planting the constant-1 function at codeword-hash pairs, we plant the outputs of $\mathfrak{F}$. More precisely, let $C_\kappa \subseteq \Sigma_\kappa^{q_\kappa}$ be our code and $\mathfrak{F} : \{0,1\}^\kappa \to \{0,1\}$ be the function to encode.[16] We

[16] Here $\kappa = \mathsf{poly}(\lambda)$ is the $\mathsf{CHN}$ parameter, chosen large enough to encode each function input in $\kappa$ bits.

sample $H$ as in CHN (Definition 14) and define $E = E_{H,B,\mathfrak{F}}$ on $(v, y) \in \Sigma_\kappa^{q_\kappa} \times \{0,1\}^\kappa$ by

$$E(v,y) = \begin{cases} \mathfrak{F}(y), & v \in C_\kappa \text{ and } H(v) = y \| 0^{q_\kappa - \kappa}, \\ B(v,y), & \text{otherwise,} \end{cases} \qquad \begin{array}{c} H \leftarrow \mathcal{H}_\kappa, \\ B(v,y) \overset{\text{ind}}{\leftarrow} \text{Bernoulli}(1/2). \end{array} \tag{7}$$

Now recall that the witness state $|\Psi_H\rangle$ lets us find, for any chosen $y \in \{0,1\}^\kappa$, a codeword $v \in C_\kappa$ satisfying $H(v) = y \| 0^{q_\kappa - \kappa}$ with overwhelming probability. Since $E(v, y)$ has the same output for every such $v$, we can obtain $\mathfrak{F}(y)$ for any $y$ without damaging $|\Psi_H\rangle$, up to negligible error (Section 13). We note that a similar encoding was used recently by [CCHY26], where they encode $H_i$ itself to achieve a ClonableQMA–QCMA separation.

For cryptography, we will use many independent instances of our problem, indexed by a serial number $s \in \{0,1\}^\lambda$, and write $E_s = E_{H_s,B_s,\mathfrak{F}_s}$. For now, let us assume that our constructions can *magically* obtain copies of $|\Omega\rangle = 2^{-\lambda/2} \sum_s |s\rangle |\Psi_{H_s}\rangle$. Measuring its serial register gives the witness state for a random $s$, and later we will show how to prepare this superposition with Prepare using classical Boolean oracles. A similar step was also used in [GMMY24], but with quantum unitary oracles and Haar-random states.

**Constructing PKE** Since we are trying to achieve a copy-protected PKE scheme, we start with the copy-protection construction of [ALL+20, CLLZ22], which uses *coset states* as decryption keys. For each serial number $s \in \{0,1\}^\lambda$, sample a uniform $2\lambda$-dimensional subspace $A_s \leq \mathbb{F}_2^{4\lambda}$, uniform shifts $a_s, a'_s \in \mathbb{F}_2^{4\lambda}$ and uniform Boolean functions $F_{s,0}, F_{s,1} : \{0,1\}^\lambda \to \{0,1\}$, and set $F_s = F_{s,0} \oplus F_{s,1}$. Then consider the function

$$C_s(x,c,\xi) = \begin{cases} F_{s,0}(x), & c = 0 \text{ and } \xi \in A_s + a_s, \\ F_{s,1}(x), & c = 1 \text{ and } \xi \in A_s^\perp + a'_s, \\ \perp, & \text{otherwise.} \end{cases}$$

As shown by [ALL+20, CLLZ22], using $|\chi_s\rangle = 2^{-\lambda} \sum_{\xi \in A_s} (-1)^{\langle a'_s, \xi \rangle} |\xi + a_s\rangle$, one can obtain $F_{s,c}(x)$ for any $c, x$ without damaging $|\chi_s\rangle$, and this can be used to decrypt ciphertexts $(u, m \oplus F_s(u))$.

The coset security argument of [ALL+20] gives copy-protection with direct oracle access to $C_s$, and Behera, Cakan, and Goyal [BCG26] give a simpler proof. For our goal, we will encode $C_s$ into $E_s$ and show that attacks with $E, N, A$ access can be converted to query-efficient attacks with direct $C_s$ access, at which point we invoke the coset security proof. Thus, at a high level, the coset state provides copy-protection security, while the CHN encoding lets us preserve it in the presence of the collapse oracle.

More precisely, we do the following.

- Encode the coset functions $(C_s)_s$ into the oracle $E(s, v, y) = E_s(v, y)$ using our function encoding above. Here $\mathfrak{F}_s$ returns the requested validity or value bit of $C_s$, so that its output $\perp$ also has a Boolean representation.
- PKE.Gen samples a quantum key $|K_s\rangle = |\Psi_{H_s}\rangle \otimes |\chi_s\rangle$ with public key $s$ by measuring the serial register of $|\Omega'\rangle = 2^{-\lambda/2} \sum_s |s\rangle |K_s\rangle$.
- PKE.Enc[17] creates $|\Gamma_s\rangle = 2^{-\lambda/2} \sum_u |u\rangle |F_s(u)\rangle$ and measures it to obtain $(u, F_s(u))$ with uniform $u$, then encrypts the message as $(u, m \oplus F_s(u))$.

[17] This sampling is necessary: if encryption were pseudodeterministic given explicit randomness $(s, m; r)$, then P = QCMA would let us invert it and break security.

Again, for now assume that we can magically obtain $|\Omega'\rangle$ and $|\Gamma_s\rangle$, which we will later prepare with $\mathsf{Prepare}$. Decryption correctness follows by combining our function decoding with the coset-state decryption described above. The Boolean oracles used by $\mathsf{Prepare}$, together with $E$, form $\mathcal{O}_{\mathsf{base}}$ (Definition 32).

**Adding the Collapse Oracle.** To obtain $\mathsf{P}^{\mathcal{O}} = \mathsf{QCMA}^{\mathcal{O}}$, we add two oracles $N, A$ to $\mathcal{O}_{\mathsf{base}}$. Morally, $N$ is an $\mathsf{NP}$ oracle, as in [KQT24], while $A$ is a $\mathsf{BQP}$ oracle that lets us simulate quantum verifiers. We build them from the ground up: suppose their answers on strings of length at most $n$ have already been defined, and write

$$\mathcal{O}_{\leq n} = (\mathcal{O}_{\mathsf{base}}|_{\leq n}, N_{\leq n}, A_{\leq n}).$$

We now define answers to queries of length $(n+2)^2$. Each such query specifies a circuit $M$ and input $x$, and $M$ may query only strings of length at most $n$, so $M^{\mathcal{O}_{\leq n}}$ uses only answers already constructed:

1. For $N$, interpret $M$ as a deterministic classical circuit and set $N(\langle M, x\rangle) = \mathbf{1}[\exists w \in \{0,1\}^{\leq n} : M^{\mathcal{O}_{\leq n}}(x, w) = 1]$.
2. For $A$, interpret $M$ as a quantum circuit and sample

$$A(\langle M, x\rangle, i) \overset{\text{ind}}{\leftarrow} \text{Bernoulli}\left(\frac{1 + \Pr\left[M^{\mathcal{O}_{\leq n}}(x) = 1\right]}{3}\right), \qquad i \in [n^4].$$

Each sampled answer remains fixed, and we set $\mathcal{O} = (\mathcal{O}_{\mathsf{base}}, N, A)$ (Definition 26).

**Showing $\mathsf{P}^{\mathcal{O}} = \mathsf{QCMA}^{\mathcal{O}}$.** Given a quantum verifier $V^{\mathcal{O}}(x, w)$ for a problem in $\mathsf{QCMA}^{\mathcal{O}}$, define the deterministic classical circuit

$$M^A_{\text{test}}(x, w) = \text{Maj}_{i \in [n^4]}\, A(\langle V, (x, w)\rangle, i).$$

This distinguishes the accepting and rejecting witnesses of $V^{\mathcal{O}}$ with overwhelming probability, so a single longer query $N(\langle M_{\text{test}}, x\rangle)$ decides whether there is an accepting witness. Thus $\mathsf{P}^{\mathcal{O}} = \mathsf{QCMA}^{\mathcal{O}}$ for almost every sampled oracle (Lemma 27).

## 2.5 Cryptography Proof Overview

We now give a high level overview of our proof.

1. **Simulating the encoded function.** First, assume that $|\Psi_H\rangle$ is supplied together with an auxiliary state $\rho$, such as the coset state. In this setting, we show that access to $E, N, A$ can be simulated using only $\mathfrak{F}$.

   **Lemma 3** (Informal, Theorem 14)**.** *Fix a query-efficient quantum algorithm $\mathcal{A}$ and Boolean oracle $\mathcal{O}_{\text{aux}}$. There is a query-efficient quantum simulator $\mathcal{S}$ such that*

$$\mathcal{A}^{\mathcal{O}_{\text{aux}}, E_{H,B,\mathfrak{F}}, N, A}(|\Psi_H\rangle, \rho) \approx \mathcal{S}^{\mathfrak{F}}(\rho). \tag{8}$$

2. **Preparing the states using classical oracles.** Next, we remove the assumption of supplied states by constructing Boolean oracles that prepare the witness and coset states, as well as the graph states used for encryption. To use the preceding simulation in this setting, we also show that sampling the preparation oracles for an exponentially close target remains undetectable, even with access to $N, A$.

**Lemma 4** (Informal, Lemma 17 and Theorem 13)**.** *For every $n$-qubit real-amplitude state $|\Omega\rangle$, there is a distribution on Boolean oracles $R, S$ and a uniform QPT algorithm* Prepare *such that, for sufficiently large $n$,*

$$\mathbb{E}_{R,S}\, \tfrac{1}{2}\|\mathsf{Prepare}^{R,S}(1^n) - |\Omega\rangle\langle\Omega|\,\|_1 \le 2^{-n}.$$

*Write $\mathcal{O}_\Omega = (\mathcal{O}_{\mathrm{aux}}, R, S, N, A)$ for the oracle obtained by sampling $R, S$ for $|\Omega\rangle$ and then sampling $N, A$, with fixed $\mathcal{O}_{\mathrm{aux}}$. For another real-amplitude state $\left|\widetilde{\Omega}\right\rangle$, $n \ge \lambda$ and every query-efficient quantum algorithm $\mathcal{A}$,*

$$\|\,|\Omega\rangle - \left|\widetilde{\Omega}\right\rangle\|_2 \le 2^{-\Omega(\lambda)} \quad \Longrightarrow \quad \mathcal{A}^{\mathcal{O}_\Omega}(\rho) \approx \mathcal{A}^{\mathcal{O}_{\widetilde{\Omega}}}(\rho),$$

*where $\rho$ and the challenger's operations may depend on both targets, but not on the sampled preparation or recursive answers.*

3. **Completing the PKE proof.** Finally, we keep the issued key and the rule for forming challenge ciphertexts unchanged, but sample the preparation oracles for new targets. This removes the preparation oracles' dependence on the corresponding secret information. Each new target is at distance $O(2^{-\lambda/2})$ from the original, so the preceding guarantee justifies this change of oracles. We can then apply encoded-function simulation and invoke the coset security proof of [BCG26], which bounds joint success on the independent ciphertext challenges by $1/2 + \mathsf{negl}(\lambda)$ (Definition 41 and Lemma 28).

### 2.5.1 Simulating Adversaries Using the Complexity Tools

To obtain the first guarantee, we simulate full-oracle queries by a classical-query deterministic algorithm $T_\eta^E$, whose classical description $\eta$ is independent of $\mathfrak{F}$, so that

$$\mathcal{A}^{\mathcal{O}_{\mathrm{aux}},E,N,A}(|\Psi_H\rangle, \rho) \approx \mathcal{A}^{T_\eta^E}(|\Psi_H\rangle, \rho).$$

The adversary may make quantum queries, so we evaluate $T_\eta^E$ coherently and leave its other operations unchanged. Thus it suffices to construct $T_\eta^E$ and then answer its remaining $E$ queries using $\mathfrak{F}$.

**Classical-query simulation and switching.** For this purpose, as in the complexity proof, partition the suffix predicates into constantly many groups $J_1, \ldots, J_m$. The encoding differs from $B$ only on $\mathcal{P}_H = \bigcap_j \mathcal{P}_{J_j}$, so at each recursive level we can use another group's incompressibility and switching guarantees. Moreover, only constantly many levels can reach an encoding input, because query lengths decrease by a square root at each level, whereas the adversary's query length is polynomial and encoding inputs have length at least $\lambda$. We can therefore choose enough groups to leave one unused at the end.

With these groups fixed, we now construct the answering algorithms in increasing query length. After substituting the algorithms for shorter queries into $M$, fix the predicates and noise used so far and write $M^Y$ for the resulting circuit, where $Y$ contains the remaining oracle answers. For an $N$ query, $M^Y$ is deterministic, so $\exists w : M^Y(x, w) = 1$ has a polynomial-width accepting-transcript DNF. The switching theorem therefore gives a polynomial-query deterministic algorithm for this condition (Theorem 7).

For an $A$ query, however, $M^Y$ is quantum, so we first estimate $p_M(Y) = \Pr\big[M^Y(x) = 1\big]$ using quantitative classical-query simulation (Theorem 5). We need exponentially small error, which can require exponentially many simulation queries. However, answers outside the chosen group's codeword-hash set are fixed, and contractive simultaneous incompressibility bounds the number of distinct queried inputs inside that set. We can therefore specify the algorithm using only polynomially many queries to $Y$. Crucially, both guarantees hold simultaneously for every choice of $Y$, and, as in the complexity proof, the resulting descriptions may depend on the predicates already used and the noise outside the current set, leaving the other groups available for subsequent levels.

**Keeping the recursive distribution exact.** The estimate alone does not yet give the required simulation, because replacing $p_M(Y)$ by $\widehat{p}_M(Y)$ in the sampling of $A$ could change a later existential answer. Thus we must preserve $A$'s exact conditional distribution throughout the recursion, and only afterwards argue that the adversary cannot detect the remaining difference.

To preserve this distribution, observe that a bit with bias $(1 + \widehat{p}_M(Y))/3$ can be sampled using a fair coin with probability $2/3$, and a Bernoulli$(\widehat{p}_M(Y))$ coin otherwise. We correct the fair part on an intersection of new independent random sets, chosen so that any input belongs to the intersection with probability $\delta$, where $\delta \geq \sup_Y |p_M(Y) - \widehat{p}_M(Y)|$ is exponentially small. Outside the intersection we use a fair bit, and inside it we sample with probability

$$\frac{1}{2} + \frac{p_M(Y) - \widehat{p}_M(Y)}{2\delta}.$$

Consequently, averaging over the intersection gives exactly the required bias $(1 + p_M(Y))/3$. Applying this sampling independently at each address preserves $A$'s conditional distribution given all shorter answers, while the algorithm for $N$ computes its answer exactly.

**The quantum query hybrid.** Now that the recursive answers have the correct distribution, we replace the auxiliary correction functions by uniform Boolean functions and include them in $\eta$, while leaving $E$ unchanged. To bound the effect, recall that each correction intersection has one random set that the recursive simulation has not used. The construction ensures that, after fixing $H, B, \mathfrak{F}, \eta$ and the supplied state and challenger information, every fixed input still belongs to such a remaining set with exponentially small probability. Since these choices determine the query states in the new experiment, the expected query weight on inputs where the answers differ is exponentially small. The quantum query hybrid therefore proves the desired simulation (Theorem 12).

**Obtaining direct function access.** It remains to answer the encoding queries. For fixed $\mathcal{O}_{\mathrm{aux}}$, $\eta$ depends only on $H, B$ and independent coins, so the simulator can sample these objects and prepare $|\Psi_H\rangle$ itself. It then runs $T_\eta^E$, answering an encoding query with $\mathfrak{F}(y)$ on $\mathcal{P}_H$ and with $B(v, y)$ elsewhere. Moreover, using this one sample at all stages preserves noncommunication and proves the encoded-function simulation guarantee (Lemma 3). In particular, setting $\rho = |\chi_s\rangle\langle\chi_s|$ reduces the supplied-key experiment to direct access to $C_s$, as required for the coset security proof.

### 2.5.2 Preparing the States Using Classical Oracles

So far, the states have been supplied directly. We now construct the classical oracles used by Prepare to obtain the key superposition $|\Omega'\rangle$ and graph state $|\Gamma_s\rangle$. More generally, we want to prepare an arbitrary real-amplitude state

$$|\Omega\rangle = \sum_{x \in \{0,1\}^n} a_x |x\rangle$$

using classical Boolean oracles. The target determines how the oracles are sampled, while the preparation algorithm receives only quantum query access to the sampled functions. In the security proof, we will replace these oracles by ones sampled for a slightly different target, so we also want close targets to give close probabilities for each oracle bit. Merely preparing the right state does not ensure this second property, as illustrated by a bit specifying whether $a_x \geq 1/2$, which can flip under an arbitrarily small change in $a_x$.

**Obtaining amplitudes by averaging Boolean answers.** A Boolean phase query multiplies each basis vector by $+1$ or $-1$, so it does not directly produce arbitrary amplitudes. However, averages of signs can approximate any number in $[-1, 1]$, suggesting that we obtain amplitudes by averaging Boolean answers.[18]

To use this averaging step with signs, let $S(x, y) \in \{0, 1\}$, where $y \in \{0, 1\}^{4n}$. Prepare $|+\rangle^{\otimes n} |+\rangle^{\otimes 4n}$, query $S$ in phase, and apply Hadamards to the last $4n$ qubits. Every $y$ contributes equally to the measurement outcome $0^{4n}$, so the remaining unnormalized vector on that outcome is

$$|\eta_S\rangle = 2^{-n/2} \sum_x \overline{\sigma}_x |x\rangle , \qquad \overline{\sigma}_x = 2^{-4n} \sum_y (-1)^{S(x,y)}.$$

Thus the task becomes choosing the signs so that their averages $\overline{\sigma}_x$ approximate the desired amplitudes.

At this point, we could assign signs using a deterministic threshold, but an individual answer could then jump when the target changes, just as in the example above. Instead, encode $a_x$ in the mean of a random sign, following the sampling used in [KQST23] to turn a distinguisher for quantum states into one for distributions of Boolean functions. In our case, we choose the sign means to equal the target amplitudes, sampling each $S(x, y)$ independently with

$$\Pr[S(x, y) = 1] = \frac{1 - a_x}{2}, \qquad \mathbb{E}[(-1)^{S(x,y)}] = a_x.$$

Now changing $a_x$ by $\varepsilon$ changes its bit probability by only $|\varepsilon|/2$. For example, a desired mean of $3/5$ gives probabilities $4/5$ and $1/5$ for signs $+1$ and $-1$. The oracle is sampled once and fixed, so the different $y$'s provide independent signs, rather than fresh randomness on repeated queries to one input.

Since each average $\overline{\sigma}_x$ has mean $a_x$ and uses $2^{4n}$ signs, we get

$$\mathbb{E}_S |\eta_S\rangle = 2^{-n/2} |\Omega\rangle , \qquad \mathbb{E}_S \big\| |\eta_S\rangle - 2^{-n/2} |\Omega\rangle \big\|_2^2 \leq 2^{-4n}.$$

Consequently, the root-mean-square error is exponentially smaller than the desired vector's norm $2^{-n/2}$, so normalization gives a state exponentially close to $|\Omega\rangle$ with overwhelming probability. Although this gives the right state, preparation is still inefficient because success has probability only about $2^{-n}$, the squared norm of $2^{-n/2} |\Omega\rangle$.

**Increasing the success probability.** To increase the success probability to about $p_{\text{succ}}$, we would instead like the successful vector to be $\sqrt{p_{\text{succ}}} |\Omega\rangle$, which the same calculation suggests obtaining by sampling

$$\Pr[S(x, y) = 1] = \frac{1 - \sqrt{2^n p_{\text{succ}}}\, a_x}{2}.$$

[18] A related averaging step is used in the state-preparation method of [SLSB19] to convert the fraction of inputs satisfying a Boolean condition into an amplitude.

This is a valid probability only if $\sqrt{2^n p_{\mathrm{succ}}}\,|a_x| \le 1$ for every $x$. Therefore, a target with small amplitudes permits larger $p_{\mathrm{succ}}$, but for $|\Omega\rangle = |0^n\rangle$ we still cannot exceed $p_{\mathrm{succ}} = 2^{-n}$. We therefore need to spread the amplitudes by a reversible transformation, prepare the transformed state, and then undo the transformation.

However, Hadamards alone cannot do this for every target, since they spread $|0^n\rangle$ uniformly but send $|+\rangle^{\otimes n}$ to $|0^n\rangle$. A related difficulty arises in state synthesis, where states with amplitudes $\pm 2^{-n/2}$ are easy to prepare but can have exponentially small overlap with the target. The construction of [GKW26] addresses this using random signs and Hadamards, and we use the same transformation to bound all amplitudes from above, as required by our sampling probabilities. Specifically, sample a uniform Boolean function $R : \{0,1\}^n \to \{0,1\}$ and define

$$U_R\,|z\rangle = (-1)^{R(z)}\,|z\rangle\,, \qquad |\alpha\rangle = \mathrm{H}^{\otimes n} U_R\,|\Omega\rangle = \sum_x \alpha_x\,|x\rangle\,.$$

Indeed, $\alpha_x = 2^{-n/2}\sum_z (-1)^{x\cdot z + R(z)} a_z$, so the random signs prevent the contributions from systematically adding with the same sign. Since $\sum_z a_z^2 = 1$, we have $\mathbb{E}_R \alpha_x^2 = 2^{-n}$, and applying Hoeffding's inequality followed by a union bound gives

$$\max_x |\alpha_x| \le \frac{5n}{2^{n/2+1}} \quad \text{except with probability } 2^{-n^2} \text{ for sufficiently large } n.$$

We can therefore take $p_{\mathrm{succ}} = 1/(25n^2)$, making every sign mean $\sqrt{2^n p_{\mathrm{succ}}}\,\alpha_x$ at most $1/2$ in absolute value. The resulting bit probabilities lie in $[1/4, 3/4]$, so each answer can be sampled using a fair coin half the time, supplying the uniform randomness needed for the later switching argument.[19]

**The preparation algorithm.** We can now put these ingredients together. For an $R$ satisfying the amplitude bound above, sample the bits of $S$ independently with

$$p_{\Omega,R}(x) := \Pr[S(x,y) = 1 \mid R] = \frac{1 - \sqrt{2^n p_{\mathrm{succ}}}\,\alpha_x}{2} \in [1/4, 3/4].$$

Run the averaging circuit with $S$ and, on outcome $0^{4n}$, apply $\mathrm{H}^{\otimes n}$ and then $U_R$ to the remaining qubits, and output them. The successful vector before these last operations has mean $\sqrt{p_{\mathrm{succ}}}\,|\alpha\rangle$ and expected squared error at most $2^{-4n}$, so, with overwhelming probability over $S$, normalization and the inverse transformation give a state exponentially close to $|\Omega\rangle$.

If the outcome is not $0^{4n}$, try again with fresh registers and the same fixed oracles for at most $500n^3$ attempts in total, outputting $|0^n\rangle$ if all fail. On the high-probability event just described, each attempt succeeds with probability at least $p_{\mathrm{succ}}/4 = 1/(100n^2)$, so the total failure probability is at most $e^{-5n}$. These steps use only $n$ and quantum queries to $R, S$, without an initial copy or a classical description of $|\Omega\rangle$.

Finally, for exceptional $R$, clip the displayed probability to $[1/4, 3/4]$ when defining $p_{\Omega,R}(x)$. This makes the sampling well defined for every $R$ without changing the argument on its high-probability event. Including the exceptional choices and the preparation error, the output $\rho_{R,S}$ satisfies

$$\mathbb{E}_{R,S} \frac{1}{2} \left\| \rho_{R,S} - |\Omega\rangle\langle\Omega| \right\|_1 \le 2^{-n}$$

for sufficiently large $n$ (Lemma 17).

[19] For $q \in [1/4, 3/4]$, a Bernoulli($q$) bit can be sampled using a fair coin with probability $1/2$, and a Bernoulli($2q - 1/2$) coin otherwise.

**Controlling the preparation-oracle probabilities.** Having established correctness, we return to the security requirement by bounding how much each bit's probability changes when the preparation oracles are sampled for close targets, despite the factor $\sqrt{2^n p_{\mathrm{succ}}}$ in the sign means. To show this, fix two targets $|\Omega\rangle, \left|\widetilde{\Omega}\right\rangle$ at distance $\delta = \| |\Omega\rangle - \left|\widetilde{\Omega}\right\rangle \|_2$ and sample the same $R$ for both, since its distribution does not depend on the target. Writing $\widetilde{\alpha}_x = \langle x| \mathrm{H}^{\otimes n} U_R \left|\widetilde{\Omega}\right\rangle$, their transformed difference is $\mathrm{H}^{\otimes n} U_R(|\Omega\rangle - \left|\widetilde{\Omega}\right\rangle)$, so applying the same concentration bound to this vector of norm $\delta$ gives

$$\max_x |\alpha_x - \widetilde{\alpha}_x| \le \frac{5n\delta}{2^{n/2+1}} \quad \text{except with probability } 2^{-n^2}.$$

Thus the increase in the sign means cancels the decrease in each amplitude difference, yielding

$$\max_x |p_{\Omega,R}(x) - p_{\widetilde{\Omega},R}(x)| \le \frac{\sqrt{2^n p_{\mathrm{succ}}}}{2} \max_x |\alpha_x - \widetilde{\alpha}_x| \le \frac{\delta}{4}.$$

Since clipping cannot increase this difference, the bound also applies to the fully defined sampling. Notice that we applied concentration to the difference of the targets, since bounding their amplitudes separately would give no factor of $\delta$.

We have therefore obtained both efficient preparation and a bound on every bit's change in probability when the preparation oracles are sampled for the other target, with independent bits conditional on $R$ (Lemma 17). This is a bound on individual probabilities, not on the statistical distance between the entire sampled functions.

**Adding $N, A$.** We now use this bound to show that replacing the preparation oracles by ones sampled for an exponentially close target remains undetectable with access to $N, A$. Since each $p_{\Omega,R}(x)$ lies in $[1/4, 3/4]$, its Bernoulli distribution contains a fair bit with probability $1/2$. As in the recursive simulation above, we put the small change of bias into this fair part on intersections of independent random sets, with exponentially small probability of containing any fixed input. Classical recursive simulation and the quantum query hybrid then bound the effect of this change. Thus sampling the preparation oracles for the other target and sampling $N, A$ accordingly is undetectable, even when the same supplied state $\rho$ depends on both targets (Theorem 13).

### 2.5.3 Completing the PKE Proof

We now combine the two guarantees to prove security of the actual scheme (Section 16.2). First, preparation correctness lets us give the challenger the ideal key $|K_s\rangle$ and ideal graph samples, while the adversary still accesses the full oracle. Its two challenge ciphertexts are $(u_i, m_{i,\beta_i} \oplus F_s(u_i))$, where $u_1, u_2$ and $\beta_1, \beta_2$ are independently uniform and the ciphertexts are given only after $\mathcal{A}_0$ outputs its message pairs and two systems.

Next, we change the public preparation oracles while keeping the issued key and the rule for forming challenge ciphertexts unchanged. First sample the key-preparation oracles for the target obtained from $|\Omega'\rangle$ by replacing $|K_s\rangle$ in its $s$ term by $|0\rangle$. Then resample $F_{s,0}, F_{s,1}$ at $u_1, u_2$ to obtain $G_0, G_1$ and sample the $s$ graph-preparation oracles for $2^{-\lambda/2} \sum_u |u\rangle |G_0(u) \oplus G_1(u)\rangle$. If $u_1 = u_2$, resample that input only once. The new targets differ from the originals in one key term and at most two graph terms, respectively, so their distances are $O(2^{-\lambda/2})$. Consequently, the preceding guarantee justifies each change of preparation oracles, with $N, A$ sampled accordingly, while the challenger continues using the original key and $F_s, u_i, \beta_i$ to form the ciphertexts.

We can now apply encoded-function simulation. Condition on $G_0, G_1$ and include the public preparation oracles, all $E_t$ with $t \neq s$, and the independent components for the other primitives

in $\mathcal{O}_{\text{aux}}$. These oracles are now independent of the issued coset, $H_s, B_s$ and the original challenge values, so the simulation gives direct $C_s$ access, reducing the adversary to the coset ciphertext experiment. Finally, the coset security bound (Lemma 28) gives joint success at most $1/2 + \mathsf{negl}(\lambda)$ and completes the proof.

## 2.6 Detailed Organization

### 2.6.1 Complexity

**Main Results** We define CHN and prove BB-QMA verification in Section 7, then establish depth reduction for BB-QCMAH in Section 8. Finally, Section 9 proves BB-QCMAH query hardness and constructs an oracle $\mathcal{O}$ with $\mathsf{P}^{\mathcal{O}} = \mathsf{QCMAH}^{\mathcal{O}} \subsetneq \mathsf{QMA}^{\mathcal{O}}$.

**Technical Tools** We introduce the codeword-hash relation and its incompressibility guarantees in Section 4, followed by deterministic-classical-query simulation of BB-QCMA verifiers in Section 5 and witness removal by switching in Section 6.

### 2.6.2 Cryptography

**Main Results** We give our cryptographic constructions in Sections 15 to 18. We then prove that these primitives coexist relative to one oracle $\mathcal{O}$ with $\mathsf{P}^{\mathcal{O}} = \mathsf{QCMAH}^{\mathcal{O}}$ in Section 19.

**Technical Tools** We prove classical simulation of the recursive oracle in Section 10, then construct Boolean oracles for state preparation in Section 11 and bound the effect of changing their target with full $N, A$ access in Section 12. Finally, Section 13 gives our function encoding, evaluation and simulation guarantees.

## 2.7 Future Directions

Our work leaves open some interesting questions.

**Pseudorandom unitaries (PRU) with a classical key when $\mathsf{P} = \mathsf{QCMA}$** Our PRUs with quantum keys follow from our PRF with quantum keys through the construction of Ma and Huang [MH25]. However, unlike classical-key PRFs, classical-key PRUs are not ruled out by $\mathsf{P} = \mathsf{QCMA}$, so one could seek a direct construction that does not go through PRFs. We leave this as an open problem, for which CHN and our techniques may be useful.

**Quantum Money and Copy-Protection without Obfuscation** A series of works have shown how to construct public-key quantum money (PKQM) [AC13, Zha19] and quantum copy-protection schemes for various cryptographic primitives and various classes of functionalities [ALL+20, CLLZ22, LLQZ22, CG24b, AB24, CG24a, CG25, ABH+26, KNSY26, BCG26]. These constructions rely on indistinguishability obfuscation (iO), and obtaining such primitives without iO remains an important question in quantum cryptography. Since our PKQM and copy-protection constructions survive without classical cryptography, they may suggest ways to obtain plain-model constructions without iO.

**Higher Levels of QCMA Hierarchy** The class QCMAH is the union of the constant levels of the QCMA hierarchy, which suffice to separate QMA from $\mathsf{P} = \mathsf{QCMA}$. Our proof may also extend to $\lambda^c$ levels for some fixed $c > 0$, and we leave this question for future work.

## 2.8 Related work

**MicroCrypt separations.** Oracle separations provide evidence that quantum cryptography can exist without classical one-way functions. Using quantum unitary oracles, Kretschmer [Kre21] obtained pseudorandom unitaries with $\mathsf{BQP} = \mathsf{QMA}$, while [GMMY24] obtained commitments and multiparty noninteractive key exchange with classical communication when $\mathsf{BQP} = \mathsf{QCMA}$, and quantum lightning relative to a separate oracle with $\mathsf{BQP} = \mathsf{QMA}$. Bartusek and Goldin [BG26] also obtained reusable unclonable encryption relative to a quantum unitary oracle with $\mathsf{BQP} = \mathsf{QMA}$.

Kretschmer, Qian, and Tal [KQT24] obtained quantum-computable, quantum-secure trapdoor one-way functions relative to a classical oracle with $\mathsf{P} = \mathsf{NP}$. Their result also gives pseudorandom unitaries, as well as public-key encryption, signatures and oblivious transfer with classical communication. Their PKE does not provide copy-protection and has classical keys, which would be impossible under our stronger collapse $\mathsf{P} = \mathsf{QCMA}$.

In a different direction, Gulati, Khurana, and Tomer [GKT26] rule out uniform fully black-box constructions of quantum-computable one-way functions from PRUs. However, this does not establish an oracle where PRUs exist but quantum-computable one-way functions do not, or where $\mathsf{P} = \mathsf{QCMA}$.

**Switching lemmas and oracle separations.** Switching lemmas have long been used to prove oracle separations by reducing circuit depth. Håstad [Hås86] used his switching lemma to prove a circuit depth hierarchy, yielding an oracle relative to which $\mathsf{PH}$ is infinite. Rossman, Servedio, and Tan [RST15] strengthened this to a random-oracle separation through an average-case depth hierarchy theorem, using random projections and a corresponding switching lemma.

On the quantum side, Aaronson, Ingram, and Kretschmer [AIK22] combine a random-restriction theorem for $\mathsf{BB\text{-}QMA}$ query algorithms with classical switching to show that $\mathsf{PP} \not\subseteq \mathsf{QMAH}$ relative to a random oracle. Whereas their argument allows quantum witnesses, Agarwal and Ben-David [ABD26] consider hierarchies with classical witnesses, developing a switching lemma for quantum query algorithms to show that both QCPH and $\mathsf{QCMAH}$ are infinite relative to a random oracle.

## 2.9 AI Methodology

We used ChatGPT-6 Astra Ultra as our sole AI tool. Our results were obtained by us without AI assistance except for the following: We used AI interactively to assist with obtaining our results regarding the classical-oracle state-preparation construction and its statistical analysis (Section 11). After our results were completed, we also used it to help choose some parameters and check correctness. We also interactively used it under our close direction for drafting and editing throughout the paper (except Sections 1 and 2 and the abstract), and we subsequently heavily edited and verified the text. We verified the whole paper and the results and take full responsibility for the full paper.

# 3 Preliminaries

In this section, we fix notation and give the definitions used for complexity, quantum information and cryptography.

## 3.1 Notation and Conventions

We use standard notation from cryptography and quantum information, referring to [Gol01, Gol04, NC10] for further details. The following conventions apply unless stated otherwise.

**Functions.** The security parameter is $\lambda$. A function $p$ is polynomially bounded if $p(\lambda) \leq \lambda^c$ for some constant $c$ and all sufficiently large $\lambda$, and we sometimes abbreviate "polynomially bounded" to "polynomial." We write $\mathsf{negl}(\lambda)$ for an unspecified nonnegative negligible function, meaning that

$$\forall c > 0 \ \exists \lambda_0 \ \forall \lambda \geq \lambda_0 : \qquad \mathsf{negl}(\lambda) \leq \lambda^{-c}.$$

A function $f$ is superlogarithmic if $f(\lambda) = \omega(\log \lambda)$, equivalently if $2^{-f(\lambda)}$ is negligible. Omitted parameter arguments are understood to be $\lambda$.

**Classical and quantum information.** We write classical keys and signatures as $\mathrm{pk}, \mathrm{sk}, \mathsf{vk}, \sigma$ and use $\mathsf{R}, \mathsf{R}_{\mathsf{sk}}, \mathsf{R}_{\mathsf{vk}}, \mathsf{R}_{\mathsf{sig}}$ for quantum registers, which may contain classical components and may be entangled with other registers. We use ket notation for pure states and density operators for mixed states, and $\mathsf{R} \leftarrow \rho$ means that $\mathsf{R}$ is initialized in state $\rho$. State sums are implicitly normalized unless explicitly described as unnormalized. We write $\mathrm{Haar}_n$ for the unitarily invariant distribution on pure $n$-qubit states.

The trace distance between states $\rho$ and $\sigma$ is $\frac{1}{2}\|\rho - \sigma\|_1$. In a trace-distance expression, an algorithm denotes its output density operator and a classical variable denotes its corresponding classical register, with all other randomness averaged. For a quantum channel $\Phi$ and any auxiliary system $W$, contractivity gives

$$\tfrac{1}{2} \left\| (\Phi \otimes \mathrm{id}_W)(\rho) - (\Phi \otimes \mathrm{id}_W)(\sigma) \right\|_1 \leq \tfrac{1}{2} \left\| \rho - \sigma \right\|_1 .$$

**Distributions.** We write $x \leftarrow S$ for uniform sampling from a finite set, $x \leftarrow \mathcal{D}$ for sampling from a distribution, and $x \leftarrow A(a)$ for a classical output of a randomized or quantum algorithm. We use the analogous notation $\mathsf{R} \leftarrow A(a)$ when the output is a quantum register.

**Algorithms and oracles.** The abbreviations $\mathsf{PPT}$ and $\mathsf{QPT}$ denote probabilistic and quantum polynomial-time algorithms. Cryptographic scheme algorithms are uniform $\mathsf{QPT}$, and an algorithm using a quantum key returns its updated register for later calls, which we leave implicit in the notation. We write $A^{\mathcal{O}}$ for an algorithm with access to the classical oracle $\mathcal{O}$, distinguishing classical and quantum access in the complexity model below. In cryptographic experiments, every algorithm and adversary has quantum access to the same oracle $\mathcal{O}$. If $\mathcal{O}$ is sampled, its sampling is independent of the adversary's description and is included in the experiment's probability. Cryptographic security is against $\mathsf{QPT}$ adversaries without additional quantum advice, with nonuniform classical circuit descriptions allowed where stated.

**Security games.** A security game is an experiment between a challenger and an adversary, whose output is a bit. We say that the adversary wins when the experiment outputs 1. A distinguishing experiment instead returns the adversary's output bit, and its advantage is the absolute difference between the two specified acceptance probabilities.

## 3.2 Complexity

The acceptance conditions for oracle machines are common to computational and query complexity, while their resource bounds differ. We first define these machines, including their behavior on inputs outside a promise, and then specify the classes in each model.

### 3.2.1 Oracle machines and the QCMA hierarchy

A quantum oracle verifier $V$ takes a classical input $z$ and a witness, has access to a Boolean oracle $\mathcal{O}$, and outputs one bit. Its oracle queries act as $|u\rangle|b\rangle \mapsto |u\rangle|b \oplus \mathcal{O}(u)\rangle$ (Equation (12)). A hierarchy verifier may also query a lower machine, as defined below. For now, we require finite computation but impose no efficiency bound. The witness length $\ell$ may depend on the input length or the parameter $\lambda$, as specified in each model. The probabilities below are over the verifier's execution with its input, witness, and oracles fixed.

We use the usual quantum and classical witness acceptance conditions [AN02], written as partial functions with values in $\{0, 1, \perp\}$.

**Definition 1** (QMA machine)**.** *Let $V$ be a quantum oracle verifier, and let $\rho$ range over density operators on $\ell$ qubits. We define its associated* QMA *machine by*

$$M^{\mathcal{O}}(z) = \begin{cases} 1, & \exists \rho : \Pr\left[V^{\mathcal{O}}(z, \rho) = 1\right] \geq 2/3, \\ 0, & \forall \rho : \Pr\left[V^{\mathcal{O}}(z, \rho) = 1\right] \leq 1/3, \\ \perp, & \textit{otherwise.} \end{cases}$$

**Definition 2** (QCMA machine)**.** *Let $V$ be a quantum oracle verifier, and let $w$ range over $\{0,1\}^{\ell}$. We define its associated* QCMA *machine by*

$$M^{\mathcal{O}}(z) = \begin{cases} 1, & \exists w : \Pr\left[V^{\mathcal{O}}(z, w) = 1\right] \geq 2/3, \\ 0, & \forall w : \Pr\left[V^{\mathcal{O}}(z, w) = 1\right] \leq 1/3, \\ \perp, & \textit{otherwise.} \end{cases}$$

Thus $\perp$ denotes the remaining inputs, on which neither acceptance condition holds. The witness may depend on $z$ and $\mathcal{O}$ in both definitions. To use such a partial machine as an oracle, we must specify what its oracle answers can be on these remaining inputs.

**Definition 3** (Boolean completion)**.** *A* Boolean completion *of $N : \{0,1\}^* \to \{0,1,\perp\}$ is a total function $\widetilde{N} : \{0,1\}^* \to \{0,1\}$ that agrees with every non-$\perp$ value of $N$. We write*

$$\mathrm{Comp}(N) = \left\{ \widetilde{N} : \{0,1\}^* \to \{0,1\} : \forall z,\ N(z) \neq \perp \implies \widetilde{N}(z) = N(z) \right\}. \tag{9}$$

Each completion assigns arbitrary bits where $N(z) = \perp$, but it is one fixed function throughout the computation, including coherent queries. Following Agarwal and Ben-David [ABD26], we require a hierarchy machine's answer to agree for every completion of its lower oracle.

**Definition 4** (QCMA hierarchy machine)**.** *A $\mathsf{QCMAH}_1$ machine is a* QCMA *machine. For $d \geq 2$, a $\mathsf{QCMAH}_d$ machine $M_d$ consists of a quantum oracle verifier $V_d$, a classical witness length $\ell$, and a $\mathsf{QCMAH}_{d-1}$ machine $M_{d-1}$. With $w \in \{0,1\}^{\ell}$, we define*

$$M_d^{\mathcal{O}}(z) = \begin{cases} 1, & \forall \widetilde{M}_{d-1} \in \mathrm{Comp}(M_{d-1}^{\mathcal{O}})\ \exists w : \Pr\left[V_d^{\mathcal{O}, \widetilde{M}_{d-1}}(z, w) = 1\right] \geq 2/3, \\ 0, & \forall \widetilde{M}_{d-1} \in \mathrm{Comp}(M_{d-1}^{\mathcal{O}})\ \forall w : \Pr\left[V_d^{\mathcal{O}, \widetilde{M}_{d-1}}(z, w) = 1\right] \leq 1/3, \\ \perp, & \textit{otherwise.} \end{cases} \tag{10}$$

In particular, the accepting witness may depend on the completion, whereas the answer must be the same for every completion. A depth-$d$ machine has verifiers $V_1, \ldots, V_d$, where $V_1$ queries only $\mathcal{O}$ and $V_i$ also queries a completion of $M_{i-1}^{\mathcal{O}}$ for $i > 1$. The lower machine receives the query string as its classical input and has access to the same base oracle $\mathcal{O}$.

For the replacement argument, it is useful to express the hierarchy condition using the partial QCMA machine $M$ associated with the two-oracle verifier $V_d$ (Definition 2). For $b \in \{0,1\}$,

$$M_d^{\mathcal{O}}(z) = b \quad \Longleftrightarrow \quad \forall \widetilde{M}_{d-1} \in \mathrm{Comp}(M_{d-1}^{\mathcal{O}}) : M^{\mathcal{O},\widetilde{M}_{d-1}}(z) = b. \tag{11}$$

### 3.2.2 Computational complexity

A promise problem is a pair $\Pi = (\Pi_{\mathsf{YES}}, \Pi_{\mathsf{NO}})$ of disjoint subsets of $\{0,1\}^*$. An algorithm must accept inputs in $\Pi_{\mathsf{YES}}$ and reject inputs in $\Pi_{\mathsf{NO}}$, with no requirement outside their union. A language $L \subseteq \{0,1\}^*$ corresponds to the total promise $(L, \{0,1\}^* \setminus L)$.

In this model, the oracle $\mathcal{O}$ is fixed and the explicit input is $x \in \{0,1\}^n$. A uniform quantum polynomial-time verifier is a family of polynomial-size quantum oracle circuits whose description on inputs of length $n$ is generated by a deterministic polynomial-time algorithm on $1^n$. The generator specifies the oracle gates without access to $\mathcal{O}$, and each oracle call counts as one step. Witness length, workspace, and query length are all polynomial in $n$.

For a fixed oracle $\mathcal{O}$, the classes $\mathsf{P}^{\mathcal{O}}$, $\mathsf{BPP}^{\mathcal{O}}$ and $\mathsf{BQP}^{\mathcal{O}}$ use uniform deterministic, randomized and quantum polynomial-time algorithms, respectively, with error at most $1/3$ for the latter two. The class $\mathsf{NP}^{\mathcal{O}}$ allows a polynomial-size classical witness checked by a deterministic polynomial-time verifier. We use PP for probabilistic polynomial time with a strict majority acceptance threshold, and PPT and QPT for probabilistic and quantum polynomial-time algorithms. Here "efficient" means polynomial-time.

We now give the witness conditions explicitly for both quantum verification classes [AN02].

**Definition 5** (QMA and QCMA)**.** *A promise problem $\Pi$ belongs to $\mathsf{QMA}^{\mathcal{O}}$ if there are a polynomial $p$ and a uniform QPT verifier $V$ such that*

$$\begin{aligned} x \in \Pi_{\mathsf{YES}} &\implies \exists \rho \text{ on } p(|x|) \text{ qubits} : \Pr\big[V^{\mathcal{O}}(x,\rho) = 1\big] \geq 2/3, \\ x \in \Pi_{\mathsf{NO}} &\implies \forall \rho \text{ on } p(|x|) \text{ qubits} : \Pr\big[V^{\mathcal{O}}(x,\rho) = 1\big] \leq 1/3. \end{aligned}$$

*It belongs to $\mathsf{QCMA}^{\mathcal{O}}$ if there are a polynomial $p$ and a uniform QPT verifier $V$ such that*

$$\begin{aligned} x \in \Pi_{\mathsf{YES}} &\implies \exists w \in \{0,1\}^{p(|x|)} : \Pr\big[V^{\mathcal{O}}(x,w) = 1\big] \geq 2/3, \\ x \in \Pi_{\mathsf{NO}} &\implies \forall w \in \{0,1\}^{p(|x|)} : \Pr\big[V^{\mathcal{O}}(x,w) = 1\big] \leq 1/3. \end{aligned}$$

The corresponding hierarchy uses these time and witness bounds at each level, together with the completion rule above [ABD26].

**Definition 6** (Computational QCMA hierarchy)**.** *For a fixed depth $d \geq 1$, a promise problem $\Pi$ belongs to $\mathsf{QCMAH}_d^{\mathcal{O}}$ if there is a hierarchy machine $M_d$ with uniform QPT verifiers $V_1, \ldots, V_d$ such that*

$$x \in \Pi_{\mathsf{YES}} \implies M_d^{\mathcal{O}}(x) = 1, \qquad x \in \Pi_{\mathsf{NO}} \implies M_d^{\mathcal{O}}(x) = 0.$$

*For each level $i$, there is a polynomial $p_i$ bounding the running time and witness length of $V_i$ on every explicit input $z$ by $p_i(|z|)$. These resource bounds hold for every base oracle and every total*

*Boolean lower oracle, whereas the acceptance conditions in Equation (10) quantify over completions of the specified lower machine. We define*

$$\mathsf{QCMAH}_1^{\mathcal{O}} = \mathsf{QCMA}^{\mathcal{O}}, \qquad \mathsf{QCMAH}^{\mathcal{O}} = \bigcup_{d \geq 1} \mathsf{QCMAH}_d^{\mathcal{O}},$$

*where $d$ is a constant independent of the input length.*

Without an oracle superscript, these classes have no base oracle. We apply the same definitions to languages using their total promises, and use the prefix Promise when emphasizing promise problems, as in $\mathsf{PromiseQCMA}^{\mathcal{O}}$.

### 3.2.3 Query complexity

In the query model, the input itself is a Boolean oracle $\mathcal{O} : \mathcal{X}_\lambda \to \{0,1\}$, where $\mathcal{X}_\lambda$ is finite and its elements have polynomial-length binary encodings. The size $N = |\mathcal{X}_\lambda|$ may be exponential, and resource bounds are measured in $\lambda$, not $N$. A classical algorithm queries individual inputs, whereas a quantum algorithm may query coherently (Equation (12)), with unrestricted finite computation between queries [ABD26].

An oracle promise problem is a family $\Pi = (\Pi_{\mathsf{YES},\lambda}, \Pi_{\mathsf{NO},\lambda})_{\lambda \geq 1}$ of disjoint sets of Boolean functions $\mathcal{X}_\lambda \to \{0,1\}$. We use a verifier family $(V_\lambda)_{\lambda \geq 1}$, whose description may depend on $\lambda$ but not on the input oracle. No uniformity or polynomial description-size bound is required. The prefix BB- denotes these black-box classes, distinguishing them from the time-bounded classes for a fixed oracle.

**Definition 7** (BB-QMA and BB-QCMA)**.** *An oracle promise problem $\Pi$ belongs to* BB-QMA *if there are polynomials $p, q$ and quantum query verifiers $V_\lambda$ using at most $q(\lambda)$ queries on every oracle and witness, such that*

$$\mathcal{O} \in \Pi_{\mathsf{YES},\lambda} \implies \exists \rho \text{ on } p(\lambda) \text{ qubits} : \Pr\left[V_\lambda^{\mathcal{O}}(1^\lambda, \rho) = 1\right] \geq 2/3,$$
$$\mathcal{O} \in \Pi_{\mathsf{NO},\lambda} \implies \forall \rho \text{ on } p(\lambda) \text{ qubits} : \Pr\left[V_\lambda^{\mathcal{O}}(1^\lambda, \rho) = 1\right] \leq 1/3.$$

*It belongs to* BB-QCMA *if there are such polynomials and verifiers with classical witnesses satisfying*

$$\mathcal{O} \in \Pi_{\mathsf{YES},\lambda} \implies \exists w \in \{0,1\}^{p(\lambda)} : \Pr\left[V_\lambda^{\mathcal{O}}(1^\lambda, w) = 1\right] \geq 2/3,$$
$$\mathcal{O} \in \Pi_{\mathsf{NO},\lambda} \implies \forall w \in \{0,1\}^{p(\lambda)} : \Pr\left[V_\lambda^{\mathcal{O}}(1^\lambda, w) = 1\right] \leq 1/3.$$

We use the same completion rule to define the query hierarchy, but bound queries and witnesses rather than running time [ABD26].

**Definition 8** (Black-box QCMA hierarchy)**.** *For a fixed depth $d \geq 1$, an oracle promise problem $\Pi$ belongs to* $\mathsf{BB\text{-}QCMAH}_d$ *if there is a family of hierarchy machines $M_{d,\lambda}$, with verifiers $V_{1,\lambda}, \ldots, V_{d,\lambda}$, such that*

$$\mathcal{O} \in \Pi_{\mathsf{YES},\lambda} \implies M_{d,\lambda}^{\mathcal{O}}(1^\lambda) = 1, \qquad \mathcal{O} \in \Pi_{\mathsf{NO},\lambda} \implies M_{d,\lambda}^{\mathcal{O}}(1^\lambda) = 0.$$

*There is a polynomial $p$ such that, starting from $1^\lambda$, every verifier makes at most $p(\lambda)$ queries, and all classical inputs to verifiers, witnesses, and oracle-query inputs have length at most $p(\lambda)$. These*

*bounds hold for every base oracle, every total Boolean lower oracle, and every witness, while finite local computation and description size are unrestricted. We define*

$$\mathsf{BB\text{-}QCMAH}_1 = \mathsf{BB\text{-}QCMA}, \qquad \mathsf{BB\text{-}QCMAH} = \bigcup_{d\geq 1} \mathsf{BB\text{-}QCMAH}_d,$$

*where $d$ is independent of $\lambda$.*

The same prefix is used for other query-model classes. For distributional results, we sample $\mathcal{O}$ and abbreviate $M^{\mathcal{O}}_{d,\lambda}(1^\lambda)$ to $M^{\mathcal{O}}_d$ when $\lambda$ is clear. Our quantitative statements also allow classical auxiliary inputs $z$, whose bounds are specified along with the query and witness bounds. Every lower machine receives its parent's query string $z$ and accesses the same base oracle $\mathcal{O}$, with the resource bounds holding even for total lower oracles that are not legal completions.

Unless a statement explicitly permits oracle-dependent descriptions, the verifier family is fixed for all input oracles, while a witness may depend on the oracle being verified. Bounds on additional advice or initial states are stated separately. For a full oracle, only the finite set of inputs accessible within the query-length bound matters. These query bounds do not imply polynomial running time.

## 3.3 Quantum Information

We specify quantum access to a classical oracle and recall the two query bounds used below, using the state and register notation fixed in Section 3.1.

### 3.3.1 Quantum oracle access and query bounds

A Boolean oracle is a function $\mathcal{O} : \mathcal{X} \to \{0,1\}$, where $\mathcal{X} \subseteq \{0,1\}^*$ is finite for the distributional lemmas and $\mathcal{X} = \{0,1\}^*$ for a full oracle. Quantum access to $\mathcal{O}$ means access to the unitary

$$|u\rangle\,|b\rangle \longmapsto |u\rangle\,|b \oplus \mathcal{O}(u)\rangle\,. \tag{12}$$

Oracles with finite output alphabets, such as $\{\bot, 0, 1\}$, use fixed encodings.[20]

As in the query model (Section 3.2.3), query bounds hold on every oracle. The query lemmas impose no bound on local computation or description length.

**Lemma 5** (Quantum query hybrid bound [BBBV97])**.** *Let $\mathcal{O}, \mathcal{O}'$ be two classical oracles that agree outside a set $E$. Fix a $Q$-query quantum algorithm $A$ and an initial state, which is the same in both executions. Let $p_j$ be the probability that measuring the input of query $j$ in the execution $A^{\mathcal{O}}$ gives an element of $E$. Then*

$$\tfrac{1}{2}\left\|A^{\mathcal{O}} - A^{\mathcal{O}'}\right\|_1 \leq 2\sum_{j=1}^{Q}\sqrt{p_j} \leq 2\sqrt{Q\sum_{j=1}^{Q} p_j}.$$

*The same bound holds for the joint output with any auxiliary system $W$ on which $A$ does not act. If both executions use the same purification of the initial state and the same intervening operations, their final pure states also satisfy*

$$\|\,|\Psi_{\mathcal{O}}\rangle - |\Psi_{\mathcal{O}'}\rangle\,\|_2^2 \leq 4Q\sum_{j=1}^{Q} p_j.$$

[20] Encodings of tuples and output symbols are implicit. They are polynomial-time computable, and malformed queries have a fixed answer.

To substitute a classical algorithm for an oracle, we use the following coherent implementation.

**Lemma 6** (Coherent classical computation)**.** *Let $T$ be a deterministic classical oracle algorithm that outputs one bit and makes at most $h$ queries on every auxiliary input and oracle. There is a quantum oracle algorithm using at most $2h$ queries that implements*

$$|z\rangle |b\rangle \longmapsto |z\rangle \left|b \oplus T^{\mathcal{O}}(z)\right\rangle$$

*exactly, including on coherent inputs. Its local computation and description may be unrestricted.*

The proof of Lemma 6 appears in Appendix A.

## 3.4 Cryptography

In this section, we recall pseudorandom functions, public-key encryption, signatures, and pseudorandom states. We state explicitly the quantum-key and repeated-use requirements used by our constructions.

### 3.4.1 Pseudorandom Functions

We use pseudorandomness against quantum queries [Zha12a]. For our constructions, we allow a reusable quantum key and an optional public serial supplied with each evaluation.

**Definition 9** (Pseudorandom Functions)**.** *A pseudorandom function (PRF) scheme is a family of functions $F_k : \{0,1\}^{m(\lambda)} \to \{0,1\}^{n(\lambda)}$ along with the following uniform $\mathsf{QPT}$ algorithms.*

- $\mathsf{Gen}^{\mathcal{O}}(1^\lambda)$ : *Takes the security parameter and samples an index $k$, then outputs a key register $\mathsf{R_{sk}}$ in state $\rho_k$ and an optional public serial $j$. The challenger also knows $k$, which need not be public.*
- $\mathsf{Eval}^{\mathcal{O}}(j, \mathsf{R_{sk}}, \mathsf{R_x}, \mathsf{R_b})$ : *Takes the serial, key, input and response registers, and evaluates $F_k$ in the response register, returning the updated key.*

*We require the following.*

**Correctness.** *The ideal evaluation applies*

$$U_{F_k} : |x, b\rangle \longmapsto |x, b \oplus F_k(x)\rangle$$

*and leaves the key unchanged. For every polynomial number of calls, the real and ideal executions have negligible trace distance, allowing coherent inputs entangled with arbitrary auxiliary systems, and intervening operations outside the key.*

**Security.** *Consider the following experiment between a challenger and a $\mathsf{QPT}$ adversary $\mathcal{A}$, with $b \in \{0,1\}$ specifying the challenge oracle.*

**Experiment** $\mathsf{PRF}^{\mathcal{O}}_{\mathcal{A}}(\lambda, b)$

1. *The challenger runs* $\mathsf{Gen}^{\mathcal{O}}(1^\lambda)$*, stores its index* $k$*, and samples an independent uniform function* $R : \{0,1\}^{m(\lambda)} \to \{0,1\}^{n(\lambda)}$*.*
2. *The challenger submits only* $j$ *to* $\mathcal{A}$*, with quantum access to* $F_k$ *if* $b = 0$ *and to* $R$ *if* $b = 1$*.*
3. *The adversary outputs a bit, which the experiment returns.*

*We require that for every* QPT *adversary* $\mathcal{A}$*,*

$$\left|\Pr\left[\mathsf{PRF}^{\mathcal{O}}_{\mathcal{A}}(\lambda, 0) = 1\right] - \Pr\left[\mathsf{PRF}^{\mathcal{O}}_{\mathcal{A}}(\lambda, 1) = 1\right]\right| \leq \mathsf{negl}(\lambda). \tag{13}$$

*For an ordinary classical-key PRF, the serial* $j$ *may be empty.*

Copy-protection for one-bit-output PRFs is defined in Definition 39.

### 3.4.2 Public-Key Encryption

We recall chosen-plaintext security for classical messages and ciphertexts, allowing the decryption key to be quantum and reused.

**Definition 10** (Public-Key Encryption)**.** *A public-key encryption scheme with message space* $\mathcal{M}_\lambda$ *consists of the following uniform* QPT *algorithms.*

- $\mathsf{Gen}^{\mathcal{O}}(1^\lambda)$ : *Takes the security parameter and outputs a classical public key* pk *and a secret-key register* $\mathsf{R}_{\mathsf{sk}}$*.*
- $\mathsf{Enc}^{\mathcal{O}}(\mathrm{pk}, m)$ : *Takes a public key* pk *and a message* $m \in \mathcal{M}_\lambda$*, and outputs a classical ciphertext* ct*.*
- $\mathsf{Dec}^{\mathcal{O}}(\mathsf{R}_{\mathsf{sk}}, \mathrm{ct})$ : *Takes a secret key and a ciphertext, and outputs a classical message* $\widehat{m}$*, returning the updated key.*

*We require the following.*

**Correctness.** *Over polynomially many honest encryptions and decryptions using one generated key, the probability of any incorrect decryption is negligible.*

**Security.** *Consider the following game between the challenger and an adversary* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$*.*

**Experiment** $\mathsf{CPA}^{\mathcal{O}}_{\mathcal{A}}(\lambda)$

1. *The challenger generates* $(\mathrm{pk}, \mathsf{R}_{\mathsf{sk}}) \leftarrow \mathsf{Gen}^{\mathcal{O}}(1^\lambda)$ *and submits* pk *to* $\mathcal{A}_0$*.*
2. *The adversary* $\mathcal{A}_0$ *outputs* $(m_0, m_1, \mathsf{R})$*, where* $m_0, m_1 \in \mathcal{M}_\lambda$ *and* $|m_0| = |m_1|$*.*
3. *The challenger samples* $b \leftarrow \{0,1\}$ *and* $\mathrm{ct} \leftarrow \mathsf{Enc}^{\mathcal{O}}(\mathrm{pk}, m_b)$*, and submits* ct *to* $\mathcal{A}_1$*.*
4. *The adversary outputs* $\widehat{b} \leftarrow \mathcal{A}^{\mathcal{O}}_1(\mathrm{ct}, \mathsf{R})$*.*
5. *The challenger outputs one if and only if* $\widehat{b} = b$*.*

*We require that for every* QPT *adversary* $\mathcal{A}$*,*

$$\left|\Pr\left[\mathsf{CPA}^{\mathcal{O}}_{\mathcal{A}}(\lambda) = 1\right] - \tfrac{1}{2}\right| \leq \mathsf{negl}(\lambda).$$

Copy-protection for bit encryption, with two independent ciphertext challenges after the split, is defined in Definition 40.

### 3.4.3 Digital Signature Schemes

We use adaptive existential unforgeability under classical chosen-message attacks, allowing quantum signing keys, verification keys, and signatures.

**Definition 11** (Digital Signature Schemes)**.** *A digital signature scheme with message space* $\{0,1\}^\lambda$ *consists of the following uniform* QPT *algorithms.*

- $\mathsf{Gen}^{\mathcal{O}}(1^\lambda)$ : *Takes the security parameter and outputs a verification key* $\mathsf{R}_{\mathsf{vk}}$ *and a signing key* $\mathsf{R}_{\mathsf{sk}}$*.*
- $\mathsf{Sign}^{\mathcal{O}}(\mathsf{R}_{\mathsf{sk}}, m)$ : *Takes a signing key and a message* $m \in \{0,1\}^\lambda$*, and outputs a signature register* $\mathsf{R}_{\mathsf{sig}}$*, returning the updated signing key.*
- $\mathsf{Ver}^{\mathcal{O}}(\mathsf{R}_{\mathsf{vk}}, m, \mathsf{R}_{\mathsf{sig}})$ : *Takes a verification key, a message and an alleged signature, and outputs an acceptance bit.*
- $\mathsf{VKGen}^{\mathcal{O}}(\mathsf{R}_{\mathsf{sk}})$ : *When verification keys are quantum, takes a signing key and outputs a fresh verification key, returning the updated signing key.*

*We require the following.*

**Correctness.** *Polynomially many honestly generated signatures are all accepted except with negligible probability, including adaptive messages and verification-key requests.*

**Security.** *Consider the following experiment between a challenger and an adversary* $\mathcal{A}$*.*

**Experiment** $\mathsf{Forge}^{\mathcal{O}}_{\mathsf{Sig},\mathcal{A}}(\lambda)$

1. *The challenger generates* $(\mathsf{R}_{\mathsf{vk}}, \mathsf{R}_{\mathsf{sk}}) \leftarrow \mathsf{Gen}^{\mathcal{O}}(1^\lambda)$*, submits* $\mathsf{R}_{\mathsf{vk}}$ *to* $\mathcal{A}$*, and initializes* $\mathcal{M} = \varnothing$*.*
2. *On each adaptive classical signing request* $m$*, the challenger runs* $\mathsf{Sign}^{\mathcal{O}}(\mathsf{R}_{\mathsf{sk}}, m)$*, keeps the updated signing key, returns only the signature and adds* $m$ *to* $\mathcal{M}$*. For a fresh quantum verification key, it runs* $\mathsf{VKGen}^{\mathcal{O}}(\mathsf{R}_{\mathsf{sk}})$*, keeps the updated signing key and returns only the new verification key.*
3. *The adversary outputs a message* $m$ *and an alleged signature* $\mathsf{R}_{\mathsf{sig}}$*.*
4. *The challenger outputs one if and only if* $m \in \{0,1\}^\lambda \setminus \mathcal{M}$ *and* $\mathsf{Ver}^{\mathcal{O}}(\mathsf{R}_{\mathsf{vk}}, m, \mathsf{R}_{\mathsf{sig}}) = 1$*, using a fresh honest verification key when it is quantum.*

*We require that for every* QPT *adversary* $\mathcal{A}$*,*

$$\Pr\left[\mathsf{Forge}^{\mathcal{O}}_{\mathsf{Sig},\mathcal{A}}(\lambda) = 1\right] \leq \mathsf{negl}(\lambda).$$

### 3.4.4 Pseudorandom Quantum States

We recall pseudorandom quantum states, introduced by Ji, Liu and Song [JLS18], using the generator formulation of Brakerski and Shmueli [BS20] for an output length $n(\lambda)$.

**Definition 12** (Pseudorandom Quantum States)**.** *A pseudorandom quantum state (PRS) generator consists of the following uniform* QPT *algorithms.*

- $\mathsf{KeyGen}^{\mathcal{O}}(1^\lambda)$ : *Takes the security parameter and outputs a classical key $k$.*
- $\mathsf{StateGen}^{\mathcal{O}}(1^\lambda, k)$ : *Takes the security parameter and a classical key $k$ in the support of $\mathsf{KeyGen}^{\mathcal{O}}(1^\lambda)$, and outputs an $n(\lambda)$-qubit state, which may be mixed.*

*We require the following.*

**Security.** *Consider the following experiment between a challenger and an adversary $\mathcal{A}$, for $b \in \{0,1\}$ and a polynomial number of copies $r = r(\lambda)$.*

**Experiment** $\mathsf{PRS}^{\mathcal{O}}_{\mathcal{A}}(\lambda, b, r)$

1. *If $b = 0$, the challenger samples $k \leftarrow \mathsf{KeyGen}^{\mathcal{O}}(1^\lambda)$, runs $\mathsf{StateGen}^{\mathcal{O}}(1^\lambda, k)$ independently $r$ times using this same key, and submits all $r$ outputs to $\mathcal{A}$. If $b = 1$, it samples one $|\psi\rangle \leftarrow \mathrm{Haar}_{n(\lambda)}$ independently of $\mathcal{O}$ and submits $|\psi\rangle^{\otimes r}$ to $\mathcal{A}$.*
2. *The adversary outputs a bit, which the experiment returns.*

*We require that for every $\mathsf{QPT}$ adversary $\mathcal{A}$ and every polynomial $r$,*

$$\left|\Pr\left[\mathsf{PRS}^{\mathcal{O}}_{\mathcal{A}}(\lambda, 0, r) = 1\right] - \Pr\left[\mathsf{PRS}^{\mathcal{O}}_{\mathcal{A}}(\lambda, 1, r) = 1\right]\right| \leq \mathsf{negl}(\lambda).$$

**Preparation accuracy of our construction.** Our construction additionally has prescribed pure target states $|\psi_k\rangle$. We measure its preparation error by

$$\epsilon_{\mathrm{prs}}(\lambda, \mathcal{O}) := \max_{k \in \mathrm{supp}(\mathsf{KeyGen}^{\mathcal{O}}(1^\lambda))} \frac{1}{2} \left\| \mathsf{StateGen}^{\mathcal{O}}(1^\lambda, k) - |\psi_k\rangle\langle\psi_k| \right\|_1 . \tag{14}$$

The fixed-oracle theorem (Theorem 19) gives $\epsilon_{\mathrm{prs}}(\lambda, \mathcal{O}) = \mathsf{negl}(\lambda)$ for almost every sampled $\mathcal{O}$. Writing $\rho_k$ for the output density matrix of $\mathsf{StateGen}^{\mathcal{O}}(1^\lambda, k)$, changing one tensor factor at a time gives

$$\frac{1}{2} \left\| \rho_k^{\otimes r} - (|\psi_k\rangle\langle\psi_k|)^{\otimes r} \right\|_1 \leq r \epsilon_{\mathrm{prs}}(\lambda, \mathcal{O}).$$

Thus using these ideal tensor powers in the security experiment changes any adversary's acceptance probability by a negligible amount.

# Part I
# Technical Tools

## 4 Codeword-Hash Pairs, Noise, and Incompressibility

In this section, we state the code and quantum decoding guarantees for codeword-hash pairs (Lemma 7). We then define contractive simultaneous incompressibility and prove it for codeword-hash pairs and sparse noise (Corollary 1).

### 4.1 Codeword-hash pairs and quantum decoding

The codeword-hash relation introduced by [YZ24] underlies our $\mathsf{CHN}$ problem. We use its *biased* version, introduced by [GGJL25] and later also used by [BHV26, Cak26].

**Definition 13** (Codeword-hash pairs)**.** *Let $C_\lambda \subseteq \Sigma_\lambda^{q_\lambda}$ be a linear code of length $q_\lambda$, where $q_\lambda > \lambda$, $q_\lambda = \mathsf{poly}(\lambda)$ and symbols have $b_\lambda = \mathsf{poly}(\lambda)$-bit encodings. Put $\mathcal{X}_\lambda = \Sigma_\lambda^{q_\lambda} \times \{0,1\}^\lambda$. For $H = (H_1, \ldots, H_{q_\lambda})$ with predicates $H_i : \Sigma_\lambda \to \{0,1\}$, we define the symbol-wise hash $H(v)$ and the set $\mathcal{P}_H$ of codeword-hash pairs by*

$$H(v) = H_1(v_1) \| \cdots \| H_{q_\lambda}(v_{q_\lambda}),$$
$$\mathcal{P}_H = \{(v, y) \in \mathcal{X}_\lambda : v \in C_\lambda,\ H(v) = y \| 0^{q_\lambda - \lambda}\}.$$

We keep the prefix predicates fixed and sample the suffix predicates from the following distribution.

**Definition 14** (Biased hash distribution)**.** *For fixed prefix predicates $H_1, \ldots, H_\lambda$, we define the biased hash distribution $\mathcal{H}_\lambda$ by independently sampling*

$$H_i(u) \leftarrow \mathrm{Bernoulli}(\lambda^{-4}) \qquad (i > \lambda,\ u \in \Sigma_\lambda).$$

*The choice of the fixed prefix predicates will be specified below.*

We next define the predicate-preimage state and the decoding property that we require of $H$.

**Definition 15** (Predicate-preimage state)**.** *Put $a_\lambda = 2q_\lambda b_\lambda$. When $H_i^{-1}(b)$ is nonempty for every $i \in [q_\lambda]$ and $b \in \{0,1\}$, we define the predicate-preimage state to be the normalized $a_\lambda$-qubit state*

$$|\Psi_H\rangle = \bigotimes_{i=1}^{q_\lambda} \bigotimes_{b \in \{0,1\}} \left( \sum_{u \in \Sigma_\lambda : H_i(u) = b} |u\rangle \right), \tag{15}$$

*with normalization factors suppressed. Otherwise, set $|\Psi_H\rangle = |0^{a_\lambda}\rangle$.*

The decoder prepares the public uniform superposition over $C_\lambda$ itself.

**Definition 16** (Good hashes)**.** *For an oracle-free quantum decoder $\mathsf{Decode}$, we call a hash $H$* good *if all preimages $H_i^{-1}(b)$ are nonempty and*

$$\forall y \in \{0,1\}^\lambda : \qquad \Pr_{v \leftarrow \mathsf{Decode}(y, |\Psi_H\rangle)}[(v, y) \in \mathcal{P}_H] \geq 1 - 2^{1-\lambda}.$$

*We denote the set of good hashes by $\mathsf{Good}_\lambda$.*

To use fresh predicates at successive levels of the hierarchy, we group the suffix constraints as follows.

**Definition 17** (Suffix-constraint sets)**.** *For an integer $d \geq 1$, partition the suffix coordinates into $d$ groups of sizes differing by at most one:*

$$\{\lambda+1,\dots,q_\lambda\} = J_1 \sqcup \cdots \sqcup J_d. \tag{16}$$

*We define the suffix-constraint sets $P_1(H),\dots,P_d(H)$ by*

$$P_j(H) = \{(v,y) \in \mathcal{X}_\lambda : v \in C_\lambda, H_1(v_1)\|\cdots\|H_\lambda(v_\lambda) = y,\ H_i(v_i) = 0 \text{ for all } i \in J_j\}. \tag{17}$$

We now state the code and decoding guarantees used below.

**Lemma 7** (Existence of suitable codes)**.** *There is a uniformly constructible family of linear codes $C_\lambda \subseteq \Sigma_\lambda^{q_\lambda}$ over efficiently represented alphabets, with polynomial-time encoding and membership testing, and a uniform polynomial-time oracle-free quantum decoder* Decode *returning a codeword or $\perp$, satisfying the following.*

1. Independent suffix constraints. *For every constant $d \geq 1$, all sufficiently large $\lambda$, and every choice of the fixed predicates $H_1,\dots,H_\lambda$, the sets $P_1(H),\dots,P_d(H)$ for $H \leftarrow \mathcal{H}_\lambda$ are mutually independent, satisfy $\mathcal{P}_H = \bigcap_{j=1}^d P_j(H)$, and obey*

$$\Pr_{H\leftarrow\mathcal{H}_\lambda}[u_1,\dots,u_t \in P_j(H)] \leq 2^{-\lambda t/(8d)} \tag{18}$$

   *for every $j \in [d]$ and all distinct $u_1,\dots,u_t \in \mathcal{X}_\lambda$ with $1 \leq t \leq 2^\lambda$.*

2. Good hashes. *The predicates $H_1,\dots,H_\lambda$ can be fixed so that, for all sufficiently large $\lambda$,*

$$\Pr_{H\leftarrow\mathcal{H}_\lambda}[H \in \mathsf{Good}_\lambda] \geq 1 - 2^{-\lambda}. \tag{19}$$

   *Neither this choice nor* Decode *depends on $d$.*

3. Code-state preparation. *The normalized uniform superposition over $C_\lambda$ can be prepared without oracle queries to trace-distance error $\epsilon$ in time* $\mathsf{poly}(\lambda, \log(1/\epsilon))$.

The proof of Lemma 7 appears in Appendix B, using the code and decoder of [BHV26]. Henceforth, we fix this code family and the prefix predicates from Lemma 7(2), unless another prefix is explicitly specified. The decoder needs neither the prefix description nor access to $H$, since its hash-dependent input is the state $|\Psi_H\rangle$.

## 4.2 Contractive simultaneous incompressibility

We recall simultaneous incompressibility from [Cak26]. Throughout this paper, we fix the error $\varepsilon_{\mathrm{inc}}(\lambda) = 2^{-8\lambda^2}$, independently of the message length.

**Definition 18** (Contractive simultaneous incompressibility)**.** *Let $F_\lambda \subseteq \mathcal{X}_\lambda$ be random, where $\mathcal{X}_\lambda \subseteq \{0,1\}^*$ is finite. Fix $\gamma : \mathbb{N} \to \mathbb{R}_{>0}$ with $\gamma(\lambda) \to \infty$ and define the recovery threshold*

$$t_{\mathrm{inc}}(\lambda, c) = \left\lceil \frac{c + 8\lambda^2}{\gamma(\lambda)} \right\rceil.$$

*The family* $(F_\lambda)_\lambda$ *satisfies* $\gamma$-contractive simultaneous incompressibility *if, for every polynomially bounded integer-valued* $c(\lambda) \geq 0$ *and all sufficiently large* $\lambda$*, the following holds. An unbounded sender receives* $F_\lambda$ *and sends* $w \leftarrow \mathsf{Enc}(F_\lambda, r) \in \{0,1\}^{c(\lambda)}$ *to an unbounded receiver* $G(w, r)$*, where* $r$ *is their shared random tape. The shared tape and the receiver's description, initial state and auxiliary oracles are jointly independent of* $F_\lambda$*, and the receiver has no access to* $F_\lambda$*. For every such sender and receiver,*

$$\Pr_{F_\lambda, r}[G(\mathsf{Enc}(F_\lambda, r), r) \text{ outputs } t_{\mathrm{inc}}(\lambda, c(\lambda)) \text{ distinct elements of } F_\lambda] \leq \varepsilon_{\mathrm{inc}}(\lambda). \tag{20}$$

The parameter $\gamma(\lambda)$ is independent of the message length. Here, *contractive* refers to the recovery bound, since $t_{\mathrm{inc}}(\lambda, c(\lambda)) = o(c(\lambda) + \lambda^2)$. For codeword-hash pairs, we obtain $\gamma(\lambda) = \Theta(\lambda)$, whereas sparse noise and its union with codeword-hash pairs give $\gamma(\lambda) = \Theta(\sqrt{\lambda})$ (Corollary 1). We prove these bounds using the following lemma, which verifies contractive simultaneous incompressibility from joint inclusion and remains valid after adding independent noise.

**Lemma 8** (Contractive simultaneous incompressibility and independent noise)**.** *Let* $F_\lambda \subseteq \mathcal{X}_\lambda$ *be random, where* $\mathcal{X}_\lambda \subseteq \{0,1\}^*$ *is finite. Suppose* $\rho(\lambda) > 0$ *tends to zero and, for all sufficiently large* $\lambda$ *and every* $1 \leq r \leq 2^\lambda$*,*

$$\Pr_{F_\lambda}[u_1, \ldots, u_r \in F_\lambda] \leq \rho(\lambda)^r \quad \text{for all distinct } u_1, \ldots, u_r \in \mathcal{X}_\lambda. \tag{21}$$

*Then* $(F_\lambda)_\lambda$ *satisfies* $\gamma$*-contractive simultaneous incompressibility for*

$$\gamma(\lambda) = \log_2(1/\rho(\lambda)).$$

*If* $B_\lambda \leftarrow \mathrm{Bernoulli}(\beta(\lambda))^{\mathcal{X}_\lambda}$ *is sampled independently of* $F_\lambda$ *and* $\rho(\lambda) + \beta(\lambda) \to 0$*, the same statement holds for* $F_\lambda \cup \mathrm{supp}(B_\lambda)$ *with* $\rho + \beta$ *in place of* $\rho$*.*

*Proof.* We suppress $\lambda$ in $F, B, \rho, \beta$. For $1 \leq t \leq 2^\lambda$ and distinct inputs $u_1, \ldots, u_t$, the noise bits satisfy $\Pr_B[u_1, \ldots, u_t \in \mathrm{supp}(B)] = \beta^t$. Assigning each input either to $F$ or to $\mathrm{supp}(B)$ and using their independence therefore gives

$$\Pr_{F,B}[u_1, \ldots, u_t \in F \cup \mathrm{supp}(B)] \leq \sum_{J \subseteq [t]} \rho^{|J|} \beta^{t-|J|} = (\rho + \beta)^t.$$

For either $F$ or $F \cup \mathrm{supp}(B)$, write $p = \rho$ or $p = \rho + \beta$, respectively. Fix a polynomially bounded $c = c(\lambda)$ and put $t = \lceil (c + 8\lambda^2)/\log_2(1/p) \rceil$. Since $p \leq 1/2$ for all sufficiently large $\lambda$, we have $1 \leq t \leq c + 8\lambda^2 + 1 \leq 2^\lambda$ eventually. Fix the shared tape $r$ and a message $w \in \{0,1\}^c$. The receiver's output is independent of the random set, so averaging the corresponding joint-inclusion bound over that output gives success probability at most $p^t$. Summing over the $2^c$ messages and averaging over $r$ bounds every sender's success probability by

$$2^c p^t \leq 2^{-8\lambda^2} = \varepsilon_{\mathrm{inc}}(\lambda).$$

The same calculation holds for every message length $c$ for which $t_{\mathrm{inc}}(\lambda, c) \leq 2^\lambda$. □

We now apply these bounds to the codeword-hash pairs and noise.

**Corollary 1** (Contractive simultaneous incompressibility of codeword-hash pairs and noise)**.** *Fix a constant* $d \geq 1$ *and any prefix predicates* $H_1, \ldots, H_\lambda$*, and independently sample*

$$H \leftarrow \mathcal{H}_\lambda, \qquad B \leftarrow \mathrm{Bernoulli}(\beta(\lambda))^{\mathcal{X}_\lambda}, \qquad \beta(\lambda) = 2^{-\lceil \sqrt{\lambda} \rceil}.$$

*Put* $\rho_d(\lambda) = 2^{-\lambda/(8d)}$. *For every* $j \in [d]$, *the families*

$$P_j(H), \qquad \mathrm{supp}(B), \qquad P_j(H) \cup \mathrm{supp}(B)$$

*satisfy contractive simultaneous incompressibility with parameters*

$$\gamma(\lambda) = \frac{\lambda}{8d}, \qquad \left\lceil \sqrt{\lambda} \right\rceil, \qquad \log_2 \frac{1}{\rho_d(\lambda) + \beta(\lambda)},$$

*respectively, and common error* $\varepsilon_{\mathrm{inc}}(\lambda) = 2^{-8\lambda^2}$. *The same holds with* $\mathcal{P}_H$ *in place of* $P_j(H)$, *setting* $d = 1$ *in these parameters.*

*Proof.* For $1 \le r \le 2^\lambda$ and any $r$ distinct inputs, Lemma 7(1) bounds their joint inclusion in $P_j(H)$ by $\rho_d(\lambda)^r$. Independence of the noise bits gives the bound $\beta(\lambda)^r$ for $\mathrm{supp}(B)$. Since $\rho_d(\lambda) + \beta(\lambda) \to 0$, Lemma 8 proves the three claims. Applying Lemma 7(1) with $d = 1$ gives the claims for $\mathcal{P}_H$. □

# 5 Deterministic-Classical-Query Simulation of Quantum Verifiers

In this section, we prove that contractive simultaneous incompressibility gives deterministic-classical-query simulation for quantum verifiers with classical witnesses (Theorem 5).

To state the simulation theorem, we define oracles that agree with a fixed function $C$ outside $F$.

**Definition 19** (Oracles agreeing outside a set)**.** *For a Boolean function $C : \mathcal{X} \to \{0,1\}$, a set $F \subseteq \mathcal{X}$ and $R : F \to \{0,1\}$, we define $\mathcal{O}_F^R$, which agrees with $C$ outside $F$, by*

$$\mathcal{O}_F^R(u) = \begin{cases} R(u), & u \in F, \\ C(u), & u \notin F. \end{cases}$$

We can now state the deterministic-classical-query simulation theorem.

**Theorem 5** (Deterministic-classical-query simulation from contractive simultaneous incompressibility)**.** *Let $\mathcal{X}, \mathcal{Z} \subseteq \{0,1\}^*$ be finite and nonempty. Fix $C : \mathcal{X} \to \{0,1\}$ and a one-bit $Q$-query quantum verifier $V(z,w)$, where $Q \geq 1$, $z \in \mathcal{Z}$ and $w \in \{0,1\}^\ell$, with a prescribed initial state for each $(z,w)$. For integers $t, L \geq 1$ and $0 < \eta < 1$, put*

$$\begin{aligned} c &= \ell + \lceil \log_2 |\mathcal{Z}| \rceil, \qquad Q_{\mathrm{sim}} = t^2 L, \\ m &= c + t\big(\lceil \log_2 Q_{\mathrm{sim}} \rceil + 1\big), \qquad \delta_{\mathrm{samp}} = 2^c (|\mathcal{X}| + 1)^t t e^{-\eta^2 L/(16Q^2)}. \end{aligned} \tag{22}$$

*Suppose $(F_\lambda)_\lambda$ satisfies $\gamma$-contractive simultaneous incompressibility (Definition 18), where $F = F_\lambda \subseteq \mathcal{X}$, $m$ is polynomially bounded and $m + 8\lambda^2 \leq \gamma(\lambda)t$. There is a deterministic-classical-query simulator $\mathsf{Sim}$ making at most $Q_{\mathrm{sim}}$ queries, followed by a randomized oracle-free computation producing one bit. Its description and local computation may be unrestricted. Writing $\mathcal{Q}_{z,w}^{\mathcal{O}}$ for its queried inputs, for all sufficiently large $\lambda$,*

$$\Pr_F \left[ \begin{array}{c} \forall z \in \mathcal{Z},\ w \in \{0,1\}^\ell,\ R : F \to \{0,1\} : \\ |\mathcal{Q}_{z,w}^{\mathcal{O}_F^R} \cap F| < t, \\ \left| \Pr\left[V^{\mathcal{O}_F^R}(z,w) = 1\right] - \Pr\left[\mathsf{Sim}^{\mathcal{O}_F^R}(z,w) = 1\right] \right| \leq \eta \end{array} \right] \geq 1 - 2^{-8\lambda^2} - \delta_{\mathrm{samp}}. \tag{23}$$

The proof of Theorem 5 appears in Section 5.1. The same query phase simulates quantum outputs in trace distance if oracle-free quantum computation is allowed after the queries.

## 5.1 Proof of deterministic-classical-query simulation

We now prove Theorem 5. Put $\alpha = \eta^2/(16Q^2)$ and $N = |\mathcal{X}|$. We may assume $L \geq 2Q$, since otherwise $\delta_{\mathrm{samp}} > 1$ and the probability bound is trivial. The simulator keeps a database $D$ of queried inputs whose answers differ from $C$. Since the answers are Boolean, $D$ determines these answers.

**Definition 20** (Database oracle)**.** *For a database $D \subseteq \mathcal{X}$, we define the database oracle by*

$$C_D(u) = C(u) \oplus \mathbf{1}[u \in D].$$

For each fixed $E \subseteq \mathcal{X}$ with $|E| < t$, let $\pi^{(j)}_{z,w,E}$ be the distribution of the measured input to query $j$ of $V^{C_E}(z,w)$, with output $\perp$ if the execution makes fewer than $j$ queries. Thus its total query weight on a set $A \subseteq \mathcal{X}$ is

$$W_{z,w,E}(A) = \sum_{j=1}^{Q} \pi^{(j)}_{z,w,E}(A). \tag{24}$$

Following the query-sampling arguments of [AIK22, Cak26], put $L_{\mathrm{qry}} = \lfloor L/Q \rfloor$ and independently sample

$$\mathbf{u}^{(j)}_{z,w,E} \leftarrow (\pi^{(j)}_{z,w,E})^{\otimes tL_{\mathrm{qry}}} \qquad (z \in \mathcal{Z},\ w \in \{0,1\}^{\ell},\ |E| < t,\ j \in [Q]). \tag{25}$$

These distributions depend only on $V, C, z, w, E$, so the whole collection is independent of $F$. Fixing it determines the learner below.

$\mathsf{Learn}^{\mathcal{O}}(z,w)$

Set $D = \varnothing$.

1. For $j = 1, \ldots, Q$, consider the inputs in $\mathbf{u}^{(j)}_{z,w,D}$ in order. For each $u \notin D \cup \{\perp\}$:
   1.1. Query $\mathcal{O}(u)$ to obtain $a$.
   1.2. If $a \neq C(u)$, add $u$ to $D$, then return $D$ if $|D| = t$ and restart Step 1 otherwise.
2. Return $D$ after completing all $Q$ sequences without adding an input.

Each restart adds an input to $D$, so the learner makes at most $t \cdot QtL_{\mathrm{qry}} \leq t^2 L = Q_{\mathrm{sim}}$ queries on every oracle. To produce its output bit, the simulator uses the rounded probabilities

$$b = \lceil \log_2(4/\eta) \rceil, \qquad p_{z,w,D} = 2^{-b} \left\lfloor 2^b \Pr\left[V^{C_D}(z,w) = 1\right] \right\rfloor,$$

which have error at most $\eta/4$ and are stored in its classical description.

$\mathsf{Sim}^{\mathcal{O}}(z,w)$

1. Run $\mathsf{Learn}^{\mathcal{O}}(z,w)$ to obtain $D$.
2. Output a bit sampled from Bernoulli$(p_{z,w,D})$.

We first state the sampling estimate that will bound the simulation error. It applies to every fixed set $F$, without an incompressibility assumption.

**Lemma 9** (Remaining query weight)**.** *Fix $F, E \subseteq \mathcal{X}$, $z \in \mathcal{Z}$ and $w \in \{0,1\}^{\ell}$, with $|E| < t$. For the samples in Equation (25), put*

$$S_i = \bigcup_{j=1}^{Q} \{u^{(j)}_{z,w,E,1}, \ldots, u^{(j)}_{z,w,E,iL_{\mathrm{qry}}}\} \cap \mathcal{X} \qquad (0 \leq i \leq t).$$

*Then*

$$\Pr_{\mathbf{u}}\left[|S_t \cap F| < t \ \wedge\ W_{z,w,E}\big(F \setminus (E \cup S_t)\big) > 2Q\alpha\right] \leq te^{-\alpha L}. \tag{26}$$

The proof of Lemma 9 appears in Section 5.2.

### 5.1.1 Applying contractive simultaneous incompressibility

We now bound the number of distinct elements of $F$ that the learner queries, using their query numbers and answers as an incompressibility message.

**Claim 1** (Few queries in the random set)**.** *For every fixed choice of the query samples in Equation (25),*

$$\Pr_F\Big[\exists z, w, R:\ \mathsf{Learn}^{\mathcal{O}_F^R}(z,w) \text{ queries } t \text{ distinct elements of } F\Big] \le 2^{-8\lambda^2}. \tag{27}$$

*Proof.* Fix the query samples. Given $F$, the sender finds $(z, w, R)$ for which the learner queries $t$ distinct elements of $F$ and sends

$$(z, w, (k_i, b_i)_{i=1}^t), \qquad m = c + t\big(\lceil \log_2 Q_{\mathrm{sim}} \rceil + 1\big),$$

where $k_i$ is the first query number of the $i$th distinct element of $F$, and $b_i$ is its answer. The receiver runs $\mathsf{Learn}(z, w)$ without oracle access, answering its $k$th query at $u$ by $b_i$ if $k = k_i$ and by $C(u)$ otherwise. After query $k_t$, it outputs the first $t_{\mathrm{inc}}(\lambda, m) \le t$ inputs queried at the marked positions, or $\perp$ if the learner stops earlier.

These answers reproduce the chosen execution through query $k_t$. Indeed, each newly queried element of $F$ has its first query marked, and a repeated query to such an element has answer $C(u)$, since the learner never queries inputs already added to $D$. Every other query lies outside $F$ and also has answer $C(u)$. The receiver therefore recovers $t_{\mathrm{inc}}(\lambda, m)$ distinct elements of $F$, so contractive simultaneous incompressibility gives Equation (27). □

**Fixing the query samples.** There are at most $2^c(N+1)^t$ triples $(z, w, E)$ with $|E| < t$. For every fixed $F$, the remaining-query-weight estimate (Lemma 9) and a union bound show that, except with probability $\delta_{\mathrm{samp}}$ over the query samples,

$$\forall z, w, E:\quad |S_t \cap F| < t \implies W_{z,w,E}\big(F \setminus (E \cup S_t)\big) \le 2Q\alpha. \tag{28}$$

Together with Claim 1, this bounds the probability of either failure over $F$ and the samples by $2^{-8\lambda^2} + \delta_{\mathrm{samp}}$. Consequently, some fixed collection of samples has failure probability at most this bound over $F$ alone. We use that collection in $\mathsf{Learn}$, so its queries are deterministic and its description depends only on the distribution of $F$, not on the sampled set.

**Bounding the simulation error.** Fix $F$ outside the exceptional event, fix $z, w, R$, and let $D = \mathsf{Learn}^{\mathcal{O}_F^R}(z, w)$ be the final database. Since the learner queries fewer than $t$ distinct elements of $F$ and $D \subseteq F$, it completes all sequences indexed by this final $D$. Each sampled input in $F$ either belongs to $D$ or is queried in this last pass, so its sampled set $S_t$ satisfies $|S_t \cap F| < t$. Moreover, $C_D$ and $\mathcal{O}_F^R$ agree on $D \cup S_t$ and outside $F$. The simultaneous sampling bound therefore gives

$$W_{z,w,D}\big(\{u : C_D(u) \ne \mathcal{O}_F^R(u)\}\big) \le 2Q\alpha. \tag{29}$$

Applying the quantum query hybrid bound (Lemma 5) to $V^{C_D}(z, w)$ now yields

$$\tfrac{1}{2}\left\| V^{\mathcal{O}_F^R}(z,w) - V^{C_D}(z,w) \right\|_1 \le 2\sqrt{Q \cdot 2Q\alpha} = \eta/\sqrt{2}.$$

Since $\mathsf{Sim}$ outputs Bernoulli$(p_{z,w,D})$, its simulation error is at most $\eta/\sqrt{2} + \eta/4 < \eta$. This holds simultaneously for every $z, w, R$ and proves Equation (23).

**Polynomial query bounds.** For codeword-hash pairs and noise, the recovery threshold has the form

$$t_{\mathrm{inc}}(\lambda, m) = \left\lceil \frac{m + 8\lambda^2}{\gamma(\lambda)} \right\rceil, \qquad \gamma(\lambda) = \Omega(\sqrt{\lambda}).$$

Suppose $Q, c, \log N, 1/\eta$ are polynomially bounded. Choose

$$t = c + \lambda^2, \qquad L = \left\lceil \alpha^{-1}\big(c \ln 2 + t \ln(N+1) + \ln t + 8\lambda^2 \ln 2\big) \right\rceil.$$

Then $Q_{\mathrm{sim}}$ is polynomial and $m = O(t \log \lambda)$, so $m + 8\lambda^2 \le \gamma(\lambda) t$ for sufficiently large $\lambda$. Thus $t_{\mathrm{inc}}(\lambda, m) \le t$ and $\delta_{\mathrm{samp}} \le 2^{-8\lambda^2}$, giving simulation error at most $\eta$ except with probability $2^{1-8\lambda^2}$.

## 5.2 Proof of the sampling estimate

We now prove Lemma 9. Group the samples into $t$ blocks, with block $i$ containing samples $(i-1)L_{\mathrm{qry}}+1, \ldots, iL_{\mathrm{qry}}$ from every query distribution. If $|S_t \cap F| < t$, some block adds no new element of $F$. For such a block $i$, the high-query-weight event implies

$$W_{z,w,E}\big(F \setminus (E \cup S_{i-1})\big) > 2Q\alpha,$$

since $S_{i-1} \subseteq S_t$. Conditional on the preceding blocks, put $w_j = \pi^{(j)}_{z,w,E}(F \setminus (E \cup S_{i-1}))$. The probability that this block misses the displayed set is

$$\prod_{j=1}^{Q} (1 - w_j)^{L_{\mathrm{qry}}} \le \exp\left( -L_{\mathrm{qry}} \sum_{j=1}^{Q} w_j \right) \le e^{-\alpha L},$$

where the last inequality uses $L_{\mathrm{qry}} \ge L/(2Q)$. Summing over the $t$ possible blocks proves the lemma.

# 6 Removing Classical Witnesses by Switching

In this section, we remove classical witnesses while preserving a quantum verifier's non-$\perp$ answers on a noise oracle and an oracle that differs only on a sparse random set (Theorem 6). The accepting-transcript identity and switching bound are also used in recursive-oracle simulation (Lemma 10 and Theorem 7).

We first specify the joint sampling of these oracles.

**Definition 21** (Oracles agreeing outside a sparse random set)**.** *Fix $c > 0$ and a finite nonempty domain $\mathcal{X} \subseteq \{0,1\}^*$, and put $b = \lceil \sqrt{\lambda} \rceil$, $\beta = 2^{-b}$ and $\rho = 2^{-c\lambda}$. We define $\mu$ to be any distribution of oracles agreeing outside a sparse random set, sampled as follows:*

1. *Sample $P \subseteq \mathcal{X}$ from any distribution satisfying*

$$\Pr_P[u_1, \ldots, u_t \in P] \le \rho^t \quad (1 \le t \le 2^\lambda)$$

   *for all distinct $u_1, \ldots, u_t \in \mathcal{X}$.*

2. *Independently sample $B(u) \leftarrow \mathrm{Bernoulli}(\beta)$ for each $u \in \mathcal{X}$.*

3. *Sample $\mathcal{O} : \mathcal{X} \to \{0,1\}$ arbitrarily subject to $\mathcal{O}|_{\mathcal{X}\setminus P} = B|_{\mathcal{X}\setminus P}$.*

For $(P, B, \mathcal{O}) \sim \mu$, we obtain the following replacement of a BB-QCMA machine by a classical-query deterministic algorithm.

**Theorem 6** (Removing classical witnesses)**.** *Let $\mu$ be a joint distribution of $(P, B, \mathcal{O})$ as specified (Definition 21) for a fixed constant $c > 0$. Let $\varnothing \ne \mathcal{Z} \subseteq \{0,1\}^*$ be finite and put $L = |\mathcal{Z}|$. Let $N_V$ be the partial BB-QCMA machine (Definition 2) associated with a fixed quantum query verifier $V(z, w)$, with auxiliary input $z \in \mathcal{Z}$ and witness $w \in \{0,1\}^\ell$. Suppose that $V$ makes at most $Q \ge 1$ queries on every oracle and that $Q, \ell, \log|\mathcal{X}|, \log L$ are polynomial in $\lambda$.*

*For every polynomially bounded integer $h \ge 1$, there is a mapping*

$$\mathsf{Replace}_{V,\mu,h} : (P, B|_{\mathcal{X}\setminus P}) \longmapsto T.$$

*For every auxiliary input and oracle, $T$ outputs a bit after fewer than $h$ deterministic-classical queries, all in $P$, and it outputs zero outside $\mathcal{Z}$. Its description and local computation may be unrestricted. Uniformly in $\mu$ and the fixed local computation of $V$, for all sufficiently large $\lambda$,*

$$\Pr_{(P,B,\mathcal{O})\leftarrow\mu}\left[\forall F \in \{B, \mathcal{O}\},\ z \in \mathcal{Z} : \quad N_V^F(z) \in \{T^F(z), \perp\}\right] \ge 1 - 2^{-\lambda^2} - 2L\,2^{-ch\sqrt{\lambda}/4}. \tag{30}$$

## 6.1 Proof of Theorem 6

We combine deterministic-classical-query simulation with two lemmas: acceptance of the simulator has a small-width DNF representation, and switching turns this DNF into a short classical query algorithm. We first state these two lemmas.

**Definition 22** (Accepting-transcript DNF)**.** *Let $\mathcal{X}, \mathcal{Z} \subseteq \{0,1\}^*$ be finite, let $\ell, m \ge 0$, and let $S(z, w)$ be an algorithm with auxiliary input $z \in \mathcal{Z}$ and witness $w \in \{0,1\}^\ell$. Suppose that, on every oracle $\mathcal{O} : \mathcal{X} \to \{0,1\}$, $S$ first makes at most $m$ deterministic-classical queries and then produces a bit without further oracle queries. For each $(z, w)$, let $\mathcal{A}_{z,w} \subseteq \bigcup_{t=0}^m \{0,1\}^t$ consist of the answer sequences $a$ on which the query phase terminates after exactly $|a|$ queries. For each such sequence*

*a, denote query $i$ by $u_i(z,w,a)$, which depends only on $z,w,a_{<i}$, and let $p_{z,w,a}$ be the acceptance probability of the final oracle-free computation. We define the accepting-transcript DNF $f_z$ by*

$$f_z(\mathcal{O}) = \bigvee_{\substack{w\in\{0,1\}^\ell,\ a\in\mathcal{A}_{z,w}\\ p_{z,w,a}\geq 1/2}} \bigwedge_{i=1}^{|a|} [\mathcal{O}(u_i(z,w,a)) = a_i]. \tag{31}$$

**Lemma 10** (Accepting transcripts)**.** *The accepting-transcript functions $f_z$ (Definition 22) are DNFs of width at most $m$ and satisfy, for every $z,\mathcal{O}$,*

$$f_z(\mathcal{O}) = 1 \quad\Longleftrightarrow\quad \exists w\in\{0,1\}^\ell : \Pr\big[S^{\mathcal{O}}(z,w)=1\big] \geq \tfrac{1}{2}. \tag{32}$$

*Moreover, let $M_V$ be the partial* QCMA *machine for a verifier $V$ with the same input and witness domains (Definition 2). For every oracle $\mathcal{O}$ satisfying*

$$\forall z\in\mathcal{Z}, w\in\{0,1\}^\ell : \quad \big|\Pr\big[V^{\mathcal{O}}(z,w)=1\big] - \Pr\big[S^{\mathcal{O}}(z,w)=1\big]\big| \leq \tfrac{1}{100}, \tag{33}$$

*the DNFs preserve every non-$\perp$ answer:*

$$\forall z\in\mathcal{Z} : \quad M_V^{\mathcal{O}}(z)\neq\perp \implies f_z(\mathcal{O}) = M_V^{\mathcal{O}}(z). \tag{34}$$

The proof of Lemma 10 appears in Section 6.2.1. We next state the switching bound for these DNFs.

**Definition 23** (Restricted Boolean function)**.** *Let $\mathcal{X}\subseteq\{0,1\}^*$ be finite, $f:\{0,1\}^{\mathcal{X}}\to\{0,1\}$, $P\subseteq\mathcal{X}$, and $U:\mathcal{X}\setminus P\to\{0,1\}$. We define the restriction $g^{P,U}$ of $f$ obtained by setting the answers outside $P$ to $U$ as follows: for $r:P\to\{0,1\}$, let*

$$\mathcal{O}_r(u) = \begin{cases} r(u), & u\in P,\\ U(u), & u\notin P, \end{cases} \qquad g^{P,U}(r) = f(\mathcal{O}_r).$$

Random restrictions are a standard tool for reducing classical circuit depth [Hås86]. Here the fixed answers are biased toward zero, and the decision tree must depend only on $P$ and $U$. We derive the following bound from Kelley's switching lemma [Kel21], which is independent of the number of DNF terms.

**Theorem 7** (Switching with sparse answers)**.** *Fix integers $b,k,h\geq 1$ and a DNF $f$ of width at most $k$ on a finite domain $\mathcal{X}\subseteq\{0,1\}^*$. There is a mapping*

$$\mathsf{Switch}_{f,h} : (P,U) \longmapsto T_{P,U}$$

*to deterministic classical algorithms that output one bit using fewer than $h$ queries to any oracle $r:P\to\{0,1\}$.*

*Suppose that $0<\rho^{1/b}\leq 1/(16bk)$ and that $P$ is random with*

$$\Pr_P[u_1,\dots,u_t\in P] \leq \rho^t \qquad (1\leq t\leq bk+h)$$

*for distinct $u_1,\dots,u_t\in\mathcal{X}$. Independently of $P$, sample*

$$B\leftarrow \mathrm{Bernoulli}(2^{-b})^{\mathcal{X}}, \qquad U = B|_{\mathcal{X}\setminus P}.$$

*The mapping and algorithms may have unrestricted local computation and description length. For the restricted function $g^{P,U}$ (Definition 23),*

$$\Pr_{P,B}\big[\forall r:P\to\{0,1\} : \quad T^r_{P,U} = g^{P,U}(r)\big] \geq 1 - 2(8bk\rho^{1/b})^h. \tag{35}$$

The proof of Theorem 7 appears in Section 6.2.2.

We now apply these lemmas to the deterministic-classical-query simulator to prove Theorem 6.

**Constructing the classical algorithm.** Use the parameters $b = \lceil\sqrt{\lambda}\rceil$, $\beta = 2^{-b}$ and $\rho = 2^{-c\lambda}$, and let $E = \mathrm{supp}(B) \cup P$. Choose the simulator $S$ from deterministic-classical-query simulation (Theorem 5) for the distribution of $E$, with the fixed function $C = 0$, accuracy $\eta = 1/100$, and parameters

$$c_V = \ell + \lceil \log_2 L \rceil, \qquad t = c_V + \lambda^2,$$
$$L_{\mathrm{sim}} = \left\lceil 16 \cdot 10^4 Q^2 \big(c_V \ln 2 + t \ln(|\mathcal{X}| + 1) + \ln t + 8\lambda^2 \ln 2\big) \right\rceil .$$

We verify the required incompressibility below. Write $k = t^2 L_{\mathrm{sim}} = \mathsf{poly}(\lambda)$ for its query bound and $f_z$ for its accepting-transcript DNF (Definition 22), both determined by $V$ and $\mu$.

Given $P$ and $U = B|_{\mathcal{X}\setminus P}$, construct one tree $\mathsf{Switch}_{f_z,h}(P,U)$ for each $z \in \mathcal{Z}$ and store these trees in the description of $T$.

$T^F(z)$

1. If $z \notin \mathcal{Z}$, output zero and terminate.
2. Start at the root of the stored tree $\mathsf{Switch}_{f_z,h}(P,U)$. At each internal node with query input $u \in P$, query $F(u)$ to obtain $a$, then follow the edge labelled $a$.
3. On reaching a leaf, return its bit and terminate.

This defines $\mathsf{Replace}_{V,\mu,h}$ using only $(P,U)$ from the sample.

**Approximating the verifier.** For distinct inputs, independence and the joint-inclusion bound give

$$\Pr[u_1, \ldots, u_t \in E] \le (\beta + \rho)^t.$$

The message length $m_{\mathrm{msg}} = c_V + t(\lceil \log_2 k \rceil + 1)$ is polynomial in $\lambda$. Since $k$ is polynomial, $t = c_V + \lambda^2$, and $\beta + \rho \le 2^{-\sqrt{\lambda}/2}$, we have

$$m_{\mathrm{msg}} + 8\lambda^2 \le t \log_2(1/(\beta + \rho))$$

for sufficiently large $\lambda$. Thus the incompressibility bound (Lemma 8) gives $t_{\mathrm{inc}}(\lambda, m_{\mathrm{msg}}) \le t$ for $E$. The choice of $L_{\mathrm{sim}}$ also bounds the sampling error in deterministic-classical-query simulation (Theorem 5) by $2^{-8\lambda^2}$. Consequently, except with probability $2^{-4\lambda^2}$ over $E$, $S$ has error at most $1/100$ simultaneously for every auxiliary input, witness and oracle supported in $E$. Both $B$ and $\mathcal{O}$ have support in $E$, so Lemma 10 gives

$$\forall F \in \{B, \mathcal{O}\}, z \in \mathcal{Z} : \quad N_V^F(z) \neq \perp \implies f_z(F) = N_V^F(z). \tag{36}$$

**Removing the witness.** Let $g_z^{P,U}$ be the restriction of $f_z$ obtained by fixing the answers outside $P$ to $U$ (Definition 23). Theorem 7 gives the query bound on every sample. Its correctness guarantee applies for sufficiently large $\lambda$, since $bk + h \le 2^\lambda$ and $\rho^{1/b} \le 1/(16bk)$. A union bound over $z \in \mathcal{Z}$ then shows that every returned tree computes $g_z^{P,U}$ on all assignments to $P$, except with probability $2L(8bk\rho^{1/b})^h$.

Since both $B$ and $\mathcal{O}$ equal $U$ outside $P$, successful switching gives

$$\forall F \in \{B, \mathcal{O}\},\ z \in \mathcal{Z} : \qquad T^F(z) = g_z^{P,U}(F|_P) = f_z(F).$$

Finally, $b \le 2\sqrt{\lambda}$ and $k$ is polynomial, so for sufficiently large $\lambda$,

$$8bk\rho^{1/b} = 8bk\, 2^{-c\lambda/b} \le 2^{-c\sqrt{\lambda}/4}.$$

Combining this bound with the displayed equality, Equation (36) and $2^{-4\lambda^2} \le 2^{-\lambda^2}$ proves Equation (30).

## 6.2 Proofs of the supporting lemmas

### 6.2.1 Accepting transcripts

We prove Lemma 10 by identifying the conjunction satisfied by each execution of the simulator.

*Proof.* For fixed $z, w, \mathcal{O}$, the conjunction indexed by $a$ holds exactly when $a$ is the answer sequence of the deterministic query phase, in which case $p_{z,w,a} = \Pr\big[S^{\mathcal{O}}(z,w) = 1\big]$. The disjunction over witnesses and sequences with $p_{z,w,a} \ge 1/2$ therefore gives Equation (32), with at most $m$ literals in each term.

To obtain agreement with $M_V$, apply the approximation in Equation (33) at the two QCMA thresholds:

$$M_V^{\mathcal{O}}(z) = 1 \implies \max_w \Pr\big[S^{\mathcal{O}}(z,w) = 1\big] \ge 2/3 - 1/100 > 1/2,$$
$$M_V^{\mathcal{O}}(z) = 0 \implies \max_w \Pr\big[S^{\mathcal{O}}(z,w) = 1\big] \le 1/3 + 1/100 < 1/2.$$

Equation (32) now gives Equation (34). □

### 6.2.2 Switching with sparse answers

We now prove switching with sparse answers (Theorem 7) using Kelley's switching lemma.

**Lemma 11** (Kelley's switching lemma [Kel21])**.** *Let $w, h \ge 1$ be integers, let $F$ be a width-$w$ DNF on a finite set $\mathcal{Y} \subseteq \{0,1\}^*$, and let $0 < p \le 1/(16w)$. Suppose a random set $P' \subseteq \mathcal{Y}$ satisfies*

$$\Pr_{P'}[S \subseteq P'] \le p^{|S|} \qquad (S \subseteq \mathcal{Y},\ |S| \le w + h).$$

*Independently sample $B' \leftarrow \mathrm{Bernoulli}(1/2)^{\mathcal{Y}}$ and put $U' = B'|_{\mathcal{Y}\setminus P'}$. Then*

$$\Pr_{P',B'}\Big[\mathrm{DT}(g^{P',U'}) \ge h\Big] \le 2(8wp)^h,$$

*where $g^{P',U'}$ is the restriction of $F$ (Definition 23) and* DT *denotes deterministic decision-tree depth.*

We first construct the returned decision tree, then use Kelley's lemma to bound the probability that it fails to compute the restriction.

Given $(P, U)$, the restricted function $g^{P,U}$ is determined without access to the unknown assignment $r : P \to \{0,1\}$. Define the mapping by the following finite search.

$\mathsf{Switch}_{f,h}(P, U)$

Fix an ordering of all decision trees of depth less than $h$ whose query inputs belong to $P$.

1. For each tree in order, enumerate all assignments $r : P \to \{0,1\}$ and compare its output with $f(\mathcal{O}_r)$, where $\mathcal{O}_r|_P = r$ and $\mathcal{O}_r|_{\mathcal{X}\setminus P} = U$. Return the first tree whose output equals $f(\mathcal{O}_r)$ for every $r$.
2. If the enumeration ends without finding such a tree, return the constant-zero tree.

The returned tree makes fewer than $h$ queries on every assignment. It remains to bound the probability that the search finds no tree computing $g^{P,U}$.

Sample a uniform Boolean function $B' : \mathcal{X} \times [b] \to \{0,1\}$ independently of $P$, and set

$$B(u) = \bigwedge_{j=1}^{b} B'(u,j), \qquad P' = P \times [b].$$

Substituting this conjunction for each variable of $f$ and distributing gives a DNF $f'$ of width at most $bk$: a positive literal contributes $b$ literals, while a negative literal becomes a disjunction of $b$ negative literals. To apply Kelley's lemma, we also need a joint-inclusion bound for $P'$. For $S \subseteq \mathcal{X} \times [b]$ with $|S| \le bk + h$, let $\pi(S) = \{u : (u,j) \in S \text{ for some } j\}$. Since $|\pi(S)| \ge |S|/b$, the joint-inclusion hypothesis gives

$$\Pr\big[S \subseteq P'\big] = \Pr[\pi(S) \subseteq P] \le \rho^{|\pi(S)|} \le (\rho^{1/b})^{|S|}.$$

Lemma 11 therefore applies with $w = bk$ and $p = \rho^{1/b}$, giving a decision tree of depth less than $h$ after the variables outside $P'$ are fixed according to $B'$, except with probability $2(8bk\rho^{1/b})^h$. Answering each query $(u,j) \in P \times [b]$ by $r(u)$ makes this tree compute $g^{P,U}(r)$ with fewer than $h$ queries, since the conjunction of the $b$ copies of $r(u)$ is again $r(u)$.

The search finds a correct tree of depth less than $h$ whenever one exists, so the bound above gives Equation (35).

When only one hierarchy level remains, we use the case $h = 1$: $T_{P,U}$ then makes no queries, so $g^{P,U}$ is constant on the successful event.

# Part II
# QMA versus QCMAH

## 7 Codeword-Hash Pairs in Noise (CHN) Problem and BB-QMA Verification

In this section, we define codeword-hash pairs in noise (CHN) and prove that it has a BB-QMA verifier using only three classical oracle queries (Theorem 8).

**Definition 24** (CHN oracle distributions)**.** *Let $C_\lambda \subseteq \Sigma_\lambda^{q_\lambda}$ be the linear code used for codeword-hash pairs (Definition 13), so oracle queries are pairs $(v, y) \in \mathcal{X}_\lambda = \Sigma_\lambda^{q_\lambda} \times \{0,1\}^\lambda$. Use the biased hash distribution $\mathcal{H}_\lambda$ (Definition 14), with prefix predicates fixed by Lemma 7(2). We define the* CHN *oracle distributions* $\mathsf{O}_{\mathsf{noise},\lambda}$ *and* $\mathsf{O}_{\mathsf{plant},\lambda}$ *by independently sampling*

$$H \leftarrow \mathcal{H}_\lambda, \qquad B(v,y) \overset{\text{ind}}{\leftarrow} \text{Bernoulli}(2^{-\lceil\sqrt{\lambda}\rceil}) \quad ((v,y) \in \mathcal{X}_\lambda), \tag{37}$$

*and returning the following Boolean oracles on $\mathcal{X}_\lambda$:*

1. Noise. *For $\mathsf{O}_{\mathsf{noise},\lambda}$, return the noise oracle $\mathcal{O}_B^{\mathsf{noise}}(v,y) := B(v,y)$.*
2. Planted pairs. *For $\mathsf{O}_{\mathsf{plant},\lambda}$, return the planted oracle*

$$\mathcal{O}_{H,B}^{\mathsf{plant}}(v,y) := \begin{cases} 1, & v \in C_\lambda \text{ and } H(v) = y \| 0^{q_\lambda - \lambda}, \\ B(v,y), & \text{otherwise} \end{cases}. \tag{38}$$

*When an oracle is sampled from either distribution, algorithms receive quantum query access to it without separate access to $H$ or $B$.*

We now give the verifier's acceptance bounds for the planted and noise distributions.

**Theorem 8** (BB-QMA verification)**.** *Let $a = a_\lambda$ be the predicate-preimage state length (Definition 15), and use the planted and noise* CHN *distributions $\mathsf{O}_{\mathsf{plant},\lambda}, \mathsf{O}_{\mathsf{noise},\lambda}$ (Definition 24). There is a uniform* QPT *verifier $V_{\mathsf{QMA}}$ using at most three classical oracle queries and a $3a$-qubit witness $\sigma$, such that, for all sufficiently large $\lambda$,*

$$\Pr_{\mathcal{O} \leftarrow \mathsf{O}_{\mathsf{plant},\lambda}} \left[ \exists \sigma : \Pr\left[ V_{\mathsf{QMA}}^{\mathcal{O}}(1^\lambda, \sigma) = 1 \right] \geq \frac{2}{3} \right] \geq 1 - 2^{-\lambda},$$

$$\Pr_{\mathcal{O} \leftarrow \mathsf{O}_{\mathsf{noise},\lambda}} \left[ \forall \sigma : \Pr\left[ V_{\mathsf{QMA}}^{\mathcal{O}}(1^\lambda, \sigma) = 1 \right] \leq \frac{1}{3} \right] \geq 1 - 6\sqrt{(a+1)2^{-\lambda}}.$$

### 7.1 Proof of BB-QMA verification

We now prove Theorem 8. The following bound will control acceptance under the noise distribution, even for an oracle-dependent witness.

**Lemma 12** (A random Boolean function)**.** *Let $\lambda, a \geq 1$ be integers. Fix an oracle-free quantum algorithm $A$ that receives an $a$-qubit density operator $\sigma$ and a classical input $y \in \{0,1\}^\lambda$ and outputs an element of $C_\lambda \cup \{\perp\}$ for the linear code $C_\lambda \subseteq \Sigma_\lambda^{q_\lambda}$ (Definition 13). For a uniform Boolean function $B : \Sigma_\lambda^{q_\lambda} \times \{0,1\}^\lambda \to \{0,1\}$,*

$$\Pr_B \left[ \exists \sigma : \Pr_{y \leftarrow \{0,1\}^\lambda,\, v \leftarrow A(\sigma,y)} [v \neq \perp \ \wedge \ B(v,y) = 1] > \frac{2}{3} \right] \leq 6\sqrt{(a+1)2^{-\lambda}}.$$

The proof of Lemma 12 appears in Appendix A.

**The verifier.** Using the oracle-free decoder $\mathsf{Decode}$ (Definition 16), define the following three-query verifier.

$V^{\mathcal{O}}_{\mathsf{QMA}}(1^\lambda, \mathsf{W}_1, \mathsf{W}_2, \mathsf{W}_3)$

The witness consists of three $a$-qubit registers.

1. For $j = 1, 2, 3$:
   1.1. Sample $y_j \leftarrow \{0,1\}^\lambda$ uniformly and run $v_j \leftarrow \mathsf{Decode}(y_j, \mathsf{W}_j)$.
   1.2. If $v_j = \perp$ or $\mathcal{O}(v_j, y_j) = 0$, output 0 and terminate.
2. Output 1.

**The planted distribution.** For every good hash $H \in \mathsf{Good}_\lambda$ (Definition 16),

$$\Pr\left[V^{\mathcal{O}^{\mathsf{plant}}_{H,B}}_{\mathsf{QMA}}(1^\lambda, |\Psi_H\rangle^{\otimes 3}) = 1\right] \geq 1 - 3 \cdot 2^{1-\lambda}.$$

Lemma 7(2) gives $\Pr_H[H \in \mathsf{Good}_\lambda] \geq 1 - 2^{-\lambda}$, proving the planted-distribution bound for sufficiently large $\lambda$.

**The noise distribution.** We couple the sparse noise oracle to a uniform Boolean oracle that is pointwise at least as large. Put $\beta = 2^{-\lceil\sqrt{\lambda}\rceil}$ and independently sample

$$U(u) \leftarrow \text{Bernoulli}(1/2), \qquad Z(u) \leftarrow \text{Bernoulli}(2\beta) \quad (u \in \mathcal{X}_\lambda),$$

and set $B(u) = U(u)Z(u)$. This gives a sample $B \leftarrow \mathsf{O}_{\mathsf{noise},\lambda}$ with $B \leq U$. For an $a$-qubit state $\tau$, write

$$p_B(\tau) = \Pr_{y \leftarrow \{0,1\}^\lambda,\, v \leftarrow \mathsf{Decode}(y,\tau)}[v \neq \perp \ \wedge \ B(v, y) = 1].$$

Because the decoder is oracle-free, $p_B(\tau) \leq p_U(\tau)$ for every $\tau$, and Lemma 12 gives

$$\Pr_B\left[\sup_\tau p_B(\tau) > 2/3\right] \leq \Pr_U\left[\sup_\tau p_U(\tau) > 2/3\right] \leq 6\sqrt{(a+1)2^{-\lambda}}.$$

To apply this one-check bound to all three checks, even when the witness registers are entangled, fix $B$ with $\sup_\tau p_B(\tau) \leq 2/3$. After conditioning on the preceding checks accepting, $\mathsf{W}_j$ still contains an $a$-qubit state, while the fresh challenge $y_j$ remains independent and uniform. Each conditional acceptance probability is therefore at most $2/3$, so

$$\Pr\left[V^{\mathcal{O}^{\mathsf{noise}}_B}_{\mathsf{QMA}}(1^\lambda, \sigma) = 1\right] \leq (2/3)^3 = 8/27 < 1/3$$

for every $\sigma$, including entangled witnesses, which proves the noise-distribution bound.

For $H \in \mathsf{Good}_\lambda$ (Definition 16), the predicate-preimage state $|\Psi_H\rangle$ (Definition 15) produces a valid pair outside any fixed set of oracle inputs $D \subseteq \mathcal{X}_\lambda$ with probability

$$\Pr_{\substack{y \leftarrow \{0,1\}^\lambda \\ v \leftarrow \mathsf{Decode}(y, |\Psi_H\rangle)}}[(v, y) \in \mathcal{P}_H \setminus D] \geq 1 - 2^{1-\lambda} - \frac{|D|}{2^\lambda}.$$

Indeed, at most $|D|$ challenges appear in $D$, and the decoder succeeds on every challenge with probability at least $1 - 2^{1-\lambda}$. The state $|\Psi_H\rangle$ depends on $H$, whereas Theorem 5 requires an oracle-independent initial state for each fixed classical auxiliary input and witness.

# 8 Depth Reduction for BB-QCMAH Machines

In this section, we prove that, except with negligible probability, a constant-depth BB-QCMAH machine cannot give different non-$\perp$ answers on two random oracles that differ only on the intersection of independent sparse sets (Theorem 9).

We first specify a common resource bound for all levels of the hierarchy.

**Definition 25** (Hierarchy resource bounds)**.** *A hierarchy has resource bound $p$ on a set $\mathcal{Z}$ of auxiliary inputs to the top machine if, starting from each $z \in \mathcal{Z}$, every verifier makes at most $p(\lambda)$ queries and all auxiliary inputs to verifiers, allowed witnesses, and oracle query inputs have length at most $p(\lambda)$. These bounds must hold on every base oracle and every choice of total Boolean lower oracles.*

With this resource convention, we can state the depth-reduction theorem.

**Theorem 9** (Iterated depth reduction)**.** *Fix constants $d \geq 1$, $c > 0$, and a polynomial $p$. Let $\varnothing \neq \mathcal{X} \subseteq \{0,1\}^{\leq p(\lambda)}$. Sample mutually independent sets $P_1, \ldots, P_d$ satisfying*

$$\Pr_{P_i}[u_1, \ldots, u_t \in P_i] \leq 2^{-c\lambda t} \qquad (1 \leq t \leq 2^\lambda)$$

*for every $i \in [d]$ and all distinct $u_1, \ldots, u_t \in \mathcal{X}$. Independently of these sets, sample*

$$B(u) \overset{\text{ind}}{\leftarrow} \text{Bernoulli}(2^{-\lceil\sqrt{\lambda}\rceil}) \quad (u \in \mathcal{X}), \qquad \Gamma = \bigcap_{i=1}^{d} P_i.$$

*Given $(P_1, \ldots, P_d, B)$, sample $\mathcal{O} : \mathcal{X} \to \{0,1\}$ arbitrarily subject to*

$$\mathcal{O}|_{\mathcal{X}\setminus\Gamma} = B|_{\mathcal{X}\setminus\Gamma} \quad \textit{almost surely.}$$

*Let $M_d$ be a fixed depth-$d$ hierarchy of quantum query verifiers, with a fixed nonempty set $\mathcal{Z} \subseteq \{0,1\}^*$ of at most $p(\lambda)$ auxiliary inputs to the top machine and resource bound $p$ on $\mathcal{Z}$ (Definition 25). There is a mapping*

$$(P_1, \ldots, P_d, B|_{\mathcal{X}\setminus\Gamma}) \longmapsto (c_z)_{z\in\mathcal{Z}} \in \{0,1\}^{\mathcal{Z}}$$

*whose outputs satisfy the following bound, uniformly in the joint distribution, finite domain and fixed local computation of $M_d$:*

$$\Pr_{P_1,\ldots,P_d,B,\mathcal{O}}\left[\forall z \in \mathcal{Z}, F \in \{B, \mathcal{O}\} : \quad M_d^F(z) \in \{c_z, \perp\}\right] \geq 1 - \varepsilon_{p,d,c}(\lambda), \qquad \varepsilon_{p,d,c}(\lambda) = 2^{-\Omega(\sqrt{\lambda})}. \tag{39}$$

## 8.1 Proof of iterated depth reduction

To prove Theorem 9, we first state two supporting lemmas. The first lets us substitute a classical query algorithm for a partial lower oracle while preserving every non-$\perp$ answer.

**Lemma 13** (Substituting a completion)**.** *Let $A(z, w)$ be a quantum query verifier with access to a base oracle and a Boolean lower oracle, let $N$ be a partial Boolean oracle machine, and let $M_{\text{comp}}$ be their completion-based composition (Definition 4). Fix a base oracle $\mathcal{O}$ and finite sets $\mathcal{U}, \mathcal{Z} \subseteq \{0,1\}^*$ such that, on auxiliary inputs $z \in \mathcal{U}$, every lower query of $A$ belongs to $\mathcal{Z}$, for every witness and*

*every choice of the two oracles. Suppose a classical-query deterministic algorithm $T$ outputs one bit on every auxiliary input and oracle and satisfies*

$$\forall u \in \mathcal{Z}: \qquad N^{\mathcal{O}}(u) \neq \perp \implies T^{\mathcal{O}}(u) = N^{\mathcal{O}}(u).$$

*Define $W^{\mathcal{O}}(z,w) = A^{\mathcal{O},T^{\mathcal{O}}}(z,w)$, and let $M_W$ be its* QCMA *machine. If $A$ and $T$ use at most $Q$ and $t$ queries on every oracle, then $W$ uses at most $Q(1+2t)$ queries and the same witness length as $A$, and*

$$\forall z \in \mathcal{U}: \qquad M^{\mathcal{O}}_{\mathrm{comp}}(z) \neq \perp \implies M^{\mathcal{O}}_{W}(z) = M^{\mathcal{O}}_{\mathrm{comp}}(z).$$

The proof of Lemma 13 appears in Section 8.2. Observe that this agreement also preserves non-$\perp$ answers higher in the hierarchy when all queries to the replaced machine lie in $\mathcal{U}$.

Theorem 6 supplies the classical algorithm needed for this substitution. The resulting hierarchy has bounded resources even when replacement fails, allowing us to iterate the reduction without conditioning on success.

**Lemma 14** (One-level depth reduction)**.** *Fix a constant $d \geq 2$, a constant $c > 0$, and a polynomial $p$. Let $\varnothing \neq \mathcal{X} \subseteq \{0,1\}^{\leq p(\lambda)}$ and sample $(P, B, \mathcal{O}) \leftarrow \mu$, where $\mu$ satisfies Definition 21 with this $c$ and domain. Let $M_d$ be a fixed depth-$d$ hierarchy of quantum query verifiers, with a finite set $\mathcal{Z} \subseteq \{0,1\}^*$ of auxiliary inputs to the top machine and resource bound $p$ on $\mathcal{Z}$ (Definition 25).*

*There is a mapping*

$$\mathsf{Reduce}_{M_d,\mu} : (P, B|_{\mathcal{X}\setminus P}) \longmapsto M'_{d-1}$$

*with unrestricted output description and local computation. Uniformly in $\mu$, $\mathcal{X}$ and the fixed local computations of $M_d$, it has the following properties:*

1. *For every input to the mapping, $M'_{d-1}$ is a depth-$(d-1)$ hierarchy of quantum query verifiers on domain $P$, with resource bound $p'(\lambda) = 5(p(\lambda)+1)^2$ on $\mathcal{Z}$.*

2. *For all sufficiently large $\lambda$,*

$$\Pr_{(P,B,\mathcal{O})\leftarrow\mu}\left[\begin{array}{c} \forall z \in \mathcal{Z}, \quad F \in \{B, \mathcal{O}\}: \\ M_d^F(z) \neq \perp \implies (M'_{d-1})^{F|_P}(z) = M_d^F(z) \end{array}\right] \geq 1 - \delta_{p,c}(\lambda), \qquad \delta_{p,c}(\lambda) = 2^{-\Omega(\sqrt{\lambda})}. \tag{40}$$

The proof of Lemma 14 appears in Section 8.2.1.

We now apply the one-level reduction repeatedly to prove Theorem 9.

**Constructing the output bits.** We define the mapping by induction on $d$. For $d = 1$, apply Theorem 6 with $P = P_1$, auxiliary input set $\mathcal{Z}$, and $h = 1$. Since the replacement makes no queries, its output on $z$ is a bit $c_z$ determined by $P_1, B|_{\mathcal{X}\setminus P_1}$. The error is $2^{-\Omega(\sqrt{\lambda})}$ because $|\mathcal{Z}|$ is polynomial.

For $d \geq 2$, let $\mu$ be the joint distribution of $(P_1, B, \mathcal{O})$ and construct the bits in two steps.

1. Apply one-level depth reduction (Lemma 14) with $P = P_1$, which is valid because $\Gamma \subseteq P_1$. Set

$$e = (P_1, B|_{\mathcal{X}\setminus P_1}), \qquad M'_{d-1,e} = \mathsf{Reduce}_{M_d,\mu}(e), \qquad P'_i = P_1 \cap P_i \quad (2 \leq i \leq d).$$

2. If $P_1 \neq \varnothing$, apply the inductive mapping to $M'_{d-1,e}$ on domain $P_1$, using $(P'_i)_{i=2}^d$ and $B|_{P_1\setminus\Gamma}$, and return its bits $(c_z)_{z\in\mathcal{Z}}$. If $P_1 = \varnothing$, return the non-$\perp$ answer of $M'_{d-1,e}$ on the empty oracle at each $z$, or zero when that answer is $\perp$.

The recursion decreases the depth, and its supplied answers together are exactly $B|_{\mathcal{X}\setminus\Gamma}$. Thus the construction terminates and has the required dependence on the sampled objects.

**Applying the induction.** We check the hypotheses for Step 2 under the conditional distribution given $e$. This conditioning fixes $M'_{d-1,e}$ and its domain $P_1$, and Lemma 14(1) gives the resource bound $p' = 5(p+1)^2$ for every $e$. The sets $P'_2, \dots, P'_d$ remain mutually independent and independent of $B|_{P_1}$, since $e$ fixes $P_1$ and only the bits of $B$ outside $P_1$. In particular,

$$B|_{P_1} \leftarrow \text{Bernoulli}(2^{-\lceil\sqrt{\lambda}\rceil})^{P_1}, \qquad \Pr\big[u_1, \dots, u_t \in P'_i \mid e\big] \le 2^{-c\lambda t}$$

for distinct $u_1, \dots, u_t \in P_1$, $1 \le t \le 2^\lambda$, and $2 \le i \le d$. Moreover, $\mathcal{O}|_{P_1} = B|_{P_1}$ outside $\bigcap_{i=2}^{d} P'_i$, which supplies the remaining induction hypothesis. For $P_1 = \varnothing$, the two reduced executions have the same empty oracle, so the chosen bits satisfy the conclusion exactly.

**Combining the errors.** Because the conditional error bound holds for every $e$ without conditioning on successful replacement, we may average over $e$ and add the first replacement's failure probability to obtain

$$\begin{aligned}
&\Pr\big[\exists z \in \mathcal{Z},\ F \in \{B, \mathcal{O}\} : M_d^F(z) \notin \{c_z, \perp\}\big] \\
&\quad \le \delta_{p,c}(\lambda) + \mathbb{E}_e \Pr\Big[\exists z, F : (M'_{d-1,e})^{F|_{P_1}}(z) \notin \{c_z, \perp\} \mid e\Big] \\
&\quad \le \delta_{p,c}(\lambda) + \varepsilon_{p',d-1,c}(\lambda) = 2^{-\Omega(\sqrt{\lambda})}.
\end{aligned}$$

The last equality uses that $d$ is constant, completing the proof.

## 8.2 Proofs of the supporting lemmas

We now prove Lemma 13, which justifies replacing a partial lower oracle by a classical query algorithm.

*Proof of Lemma 13.* Construct $W(z, w)$ by running $A(z, w)$, forwarding its queries to $\mathcal{O}$ and implementing each lower query with the coherent version of $T^{\mathcal{O}}$ from Lemma 6. This keeps the witness unchanged and uses at most $Q(1 + 2t)$ queries to $\mathcal{O}$.

For correctness, extend $T^{\mathcal{O}}|_{\mathcal{Z}}$ to a completion $\widetilde{N} \in \text{Comp}(N^{\mathcal{O}})$ by setting $\widetilde{N}(u) = N^{\mathcal{O}}(u)$ at every non-$\perp$ input outside $\mathcal{Z}$ and setting it to zero at the remaining inputs. Every lower query of $A$ belongs to $\mathcal{Z}$, so

$$W^{\mathcal{O}}(z, w) = A^{\mathcal{O}, \widetilde{N}}(z, w) \qquad (z \in \mathcal{U}).$$

If $M^{\mathcal{O}}_{\text{comp}}(z) \neq \perp$, it is the answer for every completion by Equation (11), and hence also for $\widetilde{N}$. This gives $M^{\mathcal{O}}_W(z) = M^{\mathcal{O}}_{\text{comp}}(z)$.

Since $M^{\mathcal{O}}_W$ preserves every non-$\perp$ answer of $M^{\mathcal{O}}_{\text{comp}}$ on $\mathcal{U}$, the restriction of any completion of $M^{\mathcal{O}}_W$ to $\mathcal{U}$ extends to a Boolean completion of $M^{\mathcal{O}}_{\text{comp}}$ (Equation (9)). Consequently, if a higher verifier queries $M_{\text{comp}}$ only on $\mathcal{U}$, every completion it receives after substitution is also a permitted completion before substitution on those inputs. Repeated application of Equation (11) therefore preserves the higher machines' non-$\perp$ answers as well. □

### 8.2.1 Removing one level

We now prove Lemma 14 by constructing the reduced hierarchy and checking its resource and correctness guarantees.

**Constructing the reduced hierarchy.** Let $V$ be the deepest verifier and $N_V$ its BB-QCMA machine, with witnesses padded to length $p(\lambda)$ by ignoring extra bits. Let $\mathcal{Z}_{\text{lower}} \subseteq \{0,1\}^{\leq p(\lambda)}$ contain every query string that can reach $N_V$ from $\mathcal{Z}$ under any base oracle, total Boolean lower oracles, and allowed witnesses, choosing this set independently of the sample. To absorb a union bound over $\mathcal{Z}_{\text{lower}}$, put $h = 1$ if this set is empty and otherwise

$$h = \lceil \log_2(2|\mathcal{Z}_{\text{lower}}|) \rceil \leq p(\lambda) + 2.$$

Write $U = B|_{\mathcal{X}\setminus P}$ and construct $M'_{d-1}$ as follows.

$\mathsf{Reduce}_{M_d,\mu}(P,U)$

1. If $\mathcal{Z}_{\text{lower}}$ is empty, set $T = 0$. Otherwise construct $T = \mathsf{Replace}_{V,\mu,h}(P,U)$ for the deepest verifier $V$.
2. Let $A$ be the parent verifier of $N_V$. Replace it by $W^F(z,w) = A^{F,T^F}(z,w)$, with the same classical witness. Remove $N_V$ from the hierarchy.
3. In every remaining verifier, answer a query $F(u)$ by $U(u)$ if $u \notin P$, and by a query to the given oracle on $P$ otherwise. Output this depth-$(d-1)$ hierarchy.

**Bounding the resources.** By Lemma 6, each query to $T$ uses at most $2h$ queries to $F$, so the modified parent uses at most

$$p(1+2h) \leq p(2p+5) \leq 5(p+1)^2$$

queries to $F$. Step 3 answers queries outside $P$ locally and changes neither witnesses nor query inputs, whose lengths remain at most $p(\lambda)$. Thus $p' = 5(p+1)^2$ bounds all stated query and length resources for every $(P,U)$, including samples on which replacement is incorrect.

**Preserving the answers.** If $\mathcal{Z}_{\text{lower}}$ is empty, no call to $N_V$ occurs and the substitution is exact. Otherwise, Theorem 6 bounds the probability that $T$ fails to preserve a non-$\perp$ answer of $N_V$ on either $B$ or $\mathcal{O}$ by

$$\begin{aligned} 2^{-\lambda^2} + 2|\mathcal{Z}_{\text{lower}}|\, 2^{-ch\sqrt{\lambda}/4} &\leq 2^{-\lambda^2} + 2^{-h(c\sqrt{\lambda}/4-1)} \\ &= 2^{-\Omega(\sqrt{\lambda})}, \end{aligned}$$

where the first inequality uses $2|\mathcal{Z}_{\text{lower}}| \leq 2^h$.

Whenever $T$ preserves these non-$\perp$ answers, Lemma 13 preserves all non-$\perp$ answers in Step 2. Step 3 implements queries to $B$ and $\mathcal{O}$ exactly, including on superpositions, because both equal $U$ outside $P$. This proves Equation (40).

# 9 BB-QCMAH Query Lower Bounds and QMA–QCMAH Oracle Separation

In this section, we prove that CHN is hard for BB-QCMAH machines (Theorem 10) and state the corresponding separation relative to a fixed classical oracle. Recall that CHN compares a random noise oracle $\mathcal{O}_B^{\mathsf{noise}} = B$ with the planted oracle $\mathcal{O}_{H,B}^{\mathsf{plant}} = B \vee \mathbf{1}_{\mathcal{P}_H}$ (Definition 24).

**Theorem 10** (BB-QCMAH indistinguishability)**.** *Let $d \geq 1$ be constant and let $M_d$ be a $\mathsf{BB\text{-}QCMAH}_d$ machine with a polynomial resource bound on $\{1^\lambda\}$ (Definition 25), evaluated on auxiliary input $1^\lambda$. For every fixed choice of prefix predicates $H_1, \ldots, H_\lambda$ in the biased hash distribution $\mathcal{H}_\lambda$ (Definition 14), independently sample*

$$H \leftarrow \mathcal{H}_\lambda, \qquad B(u) \overset{\text{ind}}{\leftarrow} \text{Bernoulli}(2^{-\lceil\sqrt{\lambda}\rceil}) \quad (u \in \mathcal{X}_\lambda),$$

*and put $\mathcal{O}_B^{\mathsf{noise}} = B$ and $\mathcal{O}_{H,B}^{\mathsf{plant}} = B \vee \mathbf{1}_{\mathcal{P}_H}$. There is a mapping*

$$(H, B|_{\mathcal{X}_\lambda \setminus \mathcal{P}_H}) \longmapsto c \in \{0,1\}$$

*such that*

$$\Pr_{H,B}\left[M_d^{\mathcal{O}_B^{\mathsf{noise}}}, M_d^{\mathcal{O}_{H,B}^{\mathsf{plant}}} \in \{c, \perp\}\right] \geq 1 - \varepsilon_{M_d}(\lambda), \qquad \varepsilon_{M_d}(\lambda) = 2^{-\Omega(\sqrt{\lambda})}. \tag{41}$$

*The bound is uniform in the prefix and in fixed auxiliary oracle answers independent of $H, B$. Therefore, for the planted and noise CHN distributions $\mathsf{O}_{\mathsf{plant},\lambda}, \mathsf{O}_{\mathsf{noise},\lambda}$ (Definition 24),*

$$\Pr_{\mathcal{O} \leftarrow \mathsf{O}_{\mathsf{plant},\lambda}}[M_d^{\mathcal{O}} = 1] + \Pr_{\mathcal{O} \leftarrow \mathsf{O}_{\mathsf{noise},\lambda}}[M_d^{\mathcal{O}} = 0] \leq 1 + \varepsilon_{M_d}(\lambda). \tag{42}$$

*If $M_d$ outputs a bit with probability 1 under both distributions, then*

$$\left|\Pr_{\mathcal{O} \leftarrow \mathsf{O}_{\mathsf{plant},\lambda}}[M_d^{\mathcal{O}} = 1] - \Pr_{\mathcal{O} \leftarrow \mathsf{O}_{\mathsf{noise},\lambda}}[M_d^{\mathcal{O}} = 1]\right| \leq \varepsilon_{M_d}(\lambda). \tag{43}$$

The proof of Theorem 10 appears in Section 9.1.

We also obtain a separation relative to a fixed classical oracle, including the promise collapse.

**Theorem 11** (QCMA collapse and QMA separation)**.** *There are a classical Boolean oracle $\mathcal{O}_{\mathsf{sep}}$ and a unary language $L$ such that*

$$\mathsf{P}^{\mathcal{O}_{\mathsf{sep}}} = \mathsf{BQP}^{\mathcal{O}_{\mathsf{sep}}} = \mathsf{QCMA}^{\mathcal{O}_{\mathsf{sep}}} = \mathsf{QCMAH}^{\mathcal{O}_{\mathsf{sep}}} \subsetneq \mathsf{QMA}^{\mathcal{O}_{\mathsf{sep}}}, \qquad L \in \mathsf{QMA}^{\mathcal{O}_{\mathsf{sep}}} \setminus \mathsf{QCMAH}^{\mathcal{O}_{\mathsf{sep}}},$$
$$\mathsf{PromiseP}^{\mathcal{O}_{\mathsf{sep}}} = \mathsf{PromiseQCMAH}^{\mathcal{O}_{\mathsf{sep}}}.$$

The proof of Theorem 11 appears in Appendix J.1. Note that the cryptographic oracle also gives a promise problem in QMA that is average-case hard for QCMAH (Corollary 3).

## 9.1 Proof of BB-QCMAH hardness

We now prove Theorem 10 by applying depth reduction to the independent groups of suffix predicates.

Apply the codeword-hash bound (Lemma 7(1)) with $d$ groups to the suffix-constraint sets $P_1(H), \ldots, P_d(H)$ (Definition 17). They are mutually independent, independent of $B$, intersect in $\mathcal{P}_H$, and satisfy the joint-inclusion bound $2^{-\lambda t/(8d)}$. Thus Theorem 9 applies with sparsity

exponent $1/(8d)$, intersection $\Gamma = \mathcal{P}_H$ and auxiliary input set $\{1^\lambda\}$ for the top machine, after incorporating the fixed auxiliary oracle answers into the verifiers' local computation. Its conclusion is

$$\Pr_{H,B}[M_d^{\mathcal{O}_B^{\mathsf{noise}}}, M_d^{\mathcal{O}_{H,B}^{\mathsf{plant}}} \in \{c, \perp\}] \geq 1 - 2^{-\Omega(\sqrt{\lambda})},$$

where the output bit $c$ depends only on the sets $P_i(H)$ and $B|_{\mathcal{X}_\lambda \setminus \mathcal{P}_H}$. Since $H$ determines those sets, this is precisely Equation (41).

To derive the probability bounds, let $\mathcal{E}$ be the event in Equation (41). For every sample,

$$\mathbf{1}[M_d^{\mathcal{O}_{H,B}^{\mathsf{plant}}} = 1] + \mathbf{1}[M_d^{\mathcal{O}_B^{\mathsf{noise}}} = 0] \leq 1 + \mathbf{1}[\neg\mathcal{E}],$$

so taking expectations gives Equation (42), and when both answers lie in $\{0, 1\}$ almost surely, their agreement on $\mathcal{E}$ also gives Equation (43).

# Part III
# Technical Tools for Cryptography

## 10 Simulating the Recursive Oracle

We simulate the recursive oracle with classical queries to its Boolean function $T$, while preserving auxiliary information about $T$.

**Definition 26** (The recursive oracle)**.** *For a Boolean oracle $\mathcal{O}_{\mathsf{base}}$ and functions $h_u : [0,1] \to [0,1]$, define the* recursive oracle $\mathcal{O} = (\mathcal{O}_{\mathsf{base}}, N, A)$ *by increasing query length, with $h_u(p) = (1+p)/3$ unless specified otherwise. For $a = \langle M, x\rangle \in \{0,1\}^n$, where $M$ has at most $n$ gates, witness bits and query length, the length-$(n+2)^2$ queries $\langle N, a\rangle$ and $u = \langle A, a, i\rangle$, $i \in [n^4]$, receive*

$$\begin{aligned} N(\langle M, x\rangle) &= \mathbf{1}[\exists w : M^{\mathcal{O}}(x, w) = 1], \\ A(\langle M, x\rangle, i) &\overset{\text{ind}}{\leftarrow} \text{Bernoulli}\big(h_u(\Pr[M^{\mathcal{O}}(x) = 1])\big)\,. \end{aligned} \tag{44}$$

*Here $M$ is deterministic for $N$ and quantum for $A$, and runs with the full oracle $\mathcal{O}$. Its queries have length at most $n$, so their answers have already been defined. Conditional on them, sample each $A$ bit once. Invalid queries receive zero, and $\mathcal{O}_{\mathsf{QCMA}} = (N, A)$.*

**Theorem 12** (Classical simulation of the recursive oracle)**.** *For every polynomial $L \geq \lambda$ there is a constant $m \geq 1$ such that the following holds for $c_0 > 0$ and sufficiently small $c > 0$. Fix $\mathcal{X} \subseteq \bigcup_{\ell=\lambda}^{L} \{0,1\}^{\ell}$, a Boolean oracle $\mathcal{O}_{\text{fix}}$ on $\mathcal{X}^c$, and functions $h_u$ satisfying*

$$2^{-c\lambda} \leq h_u(v) \leq 1 - 2^{-c\lambda}, \qquad |h_u(v) - h_u(w)| \leq |v - w| \quad (v, w \in [0,1]).$$

*Independently sample $G^{(1)}, \ldots, G^{(m)} \subseteq \mathcal{X}$ and $B \leftarrow \{0,1\}^{\mathcal{X}}$ with*

$$\Pr\Big[S \subseteq G^{(i)}\Big] \leq 2^{-c_0 \lambda |S|} \quad (|S| \leq 2^{\lambda}).$$

*Then sample any $(T, Z)$, where $Z$ is classical and $T : \mathcal{X} \to \{0,1\}$ satisfies*

$$T|_{(\bigcap_i G^{(i)})^c} = B|_{(\bigcap_i G^{(i)})^c}, \qquad \mathcal{O}_{\mathsf{base}}(u) = \begin{cases} T(u), & u \in \mathcal{X}, \\ \mathcal{O}_{\text{fix}}(u), & u \notin \mathcal{X}. \end{cases} \tag{45}$$

*Sample the recursive oracle $\mathcal{O} = (\mathcal{O}_{\mathsf{base}}, N, A)$ with means $h_u$ (Definition 26). Any supported recursive prefix below $\lambda^2$ may be fixed.*

*For every quantum protocol $\mathcal{A}_Z$ with $Z$-dependent input and operations, a common polynomial query bound and query length at most $L$, there is a classical-query deterministic algorithm $\mathsf{Answer}_E^T$ with*

$$E = \Phi(G^{(1)}, \ldots, G^{(m-1)}, B, R), \qquad R \perp (G^{(1)}, \ldots, G^{(m)}, B, T, Z). \tag{46}$$

*On inputs of length at most $L$, it answers $\mathcal{O}_{\mathsf{base}}$ queries exactly and makes polynomially many queries for every $T$. Uniformly over the fixed choices, its averaged joint output satisfies*

$$\tfrac{1}{2} \Big\| \mathbb{E}[|Z\rangle\langle Z| \otimes \mathcal{A}_Z^{\mathcal{O}}] - \mathbb{E}[|Z\rangle\langle Z| \otimes \mathcal{A}_Z^{\mathsf{Answer}_E^T}] \Big\|_1 \leq 2^{-\Omega(\lambda)}.$$

## 10.1 Proof

We now prove Theorem 12. Include $(G^{(1)}, \dots, G^{(m)}, B, T)$ in $Z$, since discarding these classical registers gives the stated conclusion. We first express $T$ using independent uniform Boolean functions, then construct its recursive extension while keeping one sparse set independent of the simulation.

Fix $d = m - 1$ large enough that $L < \lambda^{2^{d+1}}$, and choose

$$\rho_0 = 2^{-c_0\lambda}, \qquad \nu = \rho_0^{(256m)^{-d}}, \qquad 0 < c < \frac{c_0}{256(256m)^d}, \qquad b = 2^{-c\lambda}. \tag{47}$$

Thus $\nu^{1/256} \le b \le 1/4$ for sufficiently large $\lambda$.

**Coupling.** Sample independent uniform Boolean functions $B^{(1)}, \dots, B^{(m)}$. Define

$$G_{<m} = \bigcap_{i<m} G^{(i)}, \qquad j(u) = \min\{i < m : u \notin G^{(i)}\}, \qquad B(u) = \begin{cases} B^{(j(u))}(u), & u \notin G_{<m}, \\ B^{(m)}(u), & u \in G_{<m}. \end{cases}$$

Here $G_{<m} = \mathcal{X}$ when $m = 1$. Conditional on the groups, $B$ is uniform, so sample the original $T, Z$ from their prescribed conditional distribution given the groups and $B$, and define

$$T_0(u) = \begin{cases} T(u), & u \in G_{<m}, \\ B^{(m)}(u), & u \notin G_{<m}. \end{cases}$$

Thus $T(u) = B^{(j(u))}(u)$ when $u \notin G_{<m}$ and $T(u) = T_0(u)$ otherwise. This sampling preserves the original joint distribution, including $Z$.

Let $I$ specify the values of $\mathcal{O}_{\mathrm{fix}}$ through length $L$ and any fixed recursive prefix, and put $E_{\mathsf{base}} = (I, (G^{(i)}, B^{(i)})_{i<m})$. The following lemma constructs the recursive answers through functions $T_j$ that equal independent uniform functions $B_j$ outside sparse sets $G_j$. We will then remove these differences by a query hybrid.

**Lemma 15** (Recursive answer construction)**.** *The coupled sampling above extends to $E_d$ and functions $(G_j, B_j, T_j)_{j=0}^{d}$, with $(G_0, B_0) = (G^{(m)}, B^{(m)})$, and a deterministic algorithm $\mathsf{Rec}_{E_d}^{\mathbf{T}}$, where $\mathbf{T} = (T_0, \dots, T_d)$, such that*

$$\Pr[\mathsf{Bad}(E_d)] \le 3d\nu^4, \qquad \neg\mathsf{Bad}(E_d) \implies \mathsf{Rec}_{E_d}^{\mathbf{T}}(u) = \mathcal{O}(u) \quad (|u| \le L).$$

*The algorithm makes polynomially many queries for every $\mathbf{T}$ and returns zero on failure. The sampling preserves the original $(Z, \mathcal{O})$, and $E_d$ depends only on $E_{\mathsf{base}}$ and independent coins. Conditional on successful $E_d$, the sets $(G_j)_{j\ge1}$ and functions $(B_j)_{j\ge1}$ are mutually independent and independent of $(Z, B^{(m)})$, with uniform $B_j$ and*

$$T_j|_{G_j^c} = B_j|_{G_j^c}, \qquad \Pr[u \in G_j \mid E_d] \le \nu \quad (j \ge 1).$$

The proof of Lemma 15 appears in Section 10.1.2.

We first use this lemma to prove Theorem 12, then give the construction.

**Replacing $T_j$ by $B_j$ for $j \geq 1$.** Let $\mathsf{H}_0$ run the original protocol. Define $\mathsf{H}_1$ by answering $\mathcal{O}_{\mathsf{base}}$ queries with $T$ or $\mathcal{O}_{\mathrm{fix}}$, and $N, A$ queries with $\mathsf{Rec}_{E_d}^{\mathbf{T}}$. The latter algorithm returns zero on $\mathsf{Bad}$, so recursive answer construction (Lemma 15) gives

$$\tfrac{1}{2}\left\|\mathsf{H}_0 - \mathsf{H}_1\right\|_1 \leq 3d\nu^4.$$

Here and below the outputs include the original classical variable $Z$ and all quantum output systems. The same failure rule defines $\mathsf{H}_2$, which instead uses $B_j$ for $j \geq 1$ and leaves $T_0$ unchanged.

Conditional on successful $E_d$, $Z$, $B^{(m)}$ and the uniform Boolean functions $(B_j)_{j\geq 1}$, each new set $G_j$ still satisfies $\Pr\left[u \in G_j \mid E_d, Z, B^{(m)}, (B_\ell)_{\ell\geq 1}\right] \leq \nu$. In $\mathsf{H}_2$, these quantities determine every oracle answer: $T_0$ is determined by the original variables in $Z$ and $B^{(m)}$, and the other functions are the displayed $B_j$. They therefore determine the query weights. Writing $w(j,u)$ for the weight on address $(j,u)$ in an internal query, we have

$$\mathbb{E}\sum_{j\geq 1,u} w(j,u)\mathbf{1}[u \in G_j] \leq \nu\,\mathbb{E}\sum_{j\geq 1,u} w(j,u) \leq \nu.$$

For $Q$ original queries and at most $t_{\mathsf{sim}}$ classical queries per answer, coherent evaluation and uncomputation make at most $2Qt_{\mathsf{sim}}$ internal queries. Use $\mathsf{H}_2$ for the prefix at each changed query and apply the quantum query bound (Lemma 5), which gives

$$\tfrac{1}{2}\left\|\mathsf{H}_1 - \mathsf{H}_2\right\|_1 \leq 4Qt_{\mathsf{sim}}\sqrt{\nu}.$$

The two experiments coincide on failure. Their input and other operations may depend on $Z$, since the new sets are independent of that complete original sample under the stated conditioning.

We now express every remaining oracle call using $T$. By construction,

$$T_0(u) = \begin{cases} T(u), & u \in G_{<m}, \\ B^{(m)}(u), & u \notin G_{<m}. \end{cases}$$

Add $B^{(m)}$ and $(B_j)_{j\geq 1}$ to the classical description. On inputs longer than $L$, $\mathsf{Answer}_E^T$ returns zero. On a recursive query of length at most $L$, it runs $\mathsf{Rec}_{E_d}$ with oracles $T_0, B_1, \ldots, B_d$, answering each $T_0$ query by the displayed formula. On a query to $\mathcal{O}_{\mathsf{base}}$, it returns $T(u)$ for $u \in \mathcal{X}$ and $\mathcal{O}_{\mathrm{fix}}(u)$ otherwise. For fixed original groups and $B$, the conditional distribution of $(B^{(i)})_{i\in[m]}$ depends only on $G^{(1)}, \ldots, G^{(m-1)}, B$: at each $u$, its selected bit must equal $B(u)$, and the other bits are sampled independently from $\{0,1\}$. The sampled $T, Z$ do not affect this conditional distribution. All further coins in the construction are independent of the enlarged variable $Z$. The resulting description therefore has precisely the form $E = \Phi(G^{(1)}, \ldots, G^{(m-1)}, B, R)$ with $R \perp Z$.

Only oracle queries were replaced, so the protocol's other operations and party separation are preserved. Finally,

$$\mathbb{E}_Z \tfrac{1}{2}\left\|\mathbb{E}_{\mathcal{O}|Z}\mathcal{A}_Z^{\mathcal{O}} - \mathbb{E}_R\mathcal{A}_Z^{\mathsf{Answer}_E^T}\right\|_1 \leq 3d\nu^4 + 4Qt_{\mathsf{sim}}\sqrt{\nu} = 2^{-\Omega(\lambda)},$$

which proves the theorem.

### 10.1.1 One random restriction

Independently sample a random set $P \subseteq \mathcal{X}$ and a uniform Boolean function $C : \mathcal{X} \to \{0,1\}$, and write $C_P^Y(u) = Y(u)$ on $P$ and $C(u)$ elsewhere. Assume

$$\Pr_P[S \subseteq P] \le \rho^{|S|} \quad (|S| \le 2^\lambda), \qquad \rho = 2^{-c_{\mathrm{res}}\lambda}, \qquad \log(|\mathcal{X}|+1) = \mathsf{poly}(\lambda)$$

for a constant $c_{\mathrm{res}} > 0$. The next lemma constructs classical query algorithms that work for every $Y$. This simultaneous guarantee allows the later recursive construction to choose its remaining answers after the algorithms have been specified.

**Lemma 16** (One random restriction)**.** *Fix quantum circuits $W_u$ and deterministic verifiers $M_u(w)$ with polynomial query, index and witness bounds, and sample $(P, C)$ as above. For $\rho^{1/40} \le \epsilon \le 1/4$, there are classical-query deterministic algorithms $\tau_u, T_u$ and numbers $v_{u,\tau} \in [0,1]$, depending on $(P, C)$, where $\tau_u(Y)$ returns the inputs queried in $Y$ and their answers, or $\perp$. Both algorithms make polynomially many queries on every oracle, and except with probability $3\rho^4$ over $(P, C)$,*

$$\forall u, Y: \qquad |v_{u,\tau_u(Y)} - \Pr\left[W_u^{C_P^Y} = 1\right]| \le \epsilon, \qquad T_u^Y = \mathbf{1}[\exists w : M_u^{C_P^Y}(w) = 1].$$

We now prove Lemma 16, first estimating quantum acceptance and then computing the existential answers.

**Acceptance estimates.** If $\mathcal{X} = \varnothing$, evaluate the tests and existential predicates without queries, so assume $\mathcal{X} \neq \varnothing$. For quantum query bound $Q \ge 1$ and index length $s$, set $t = s + \lambda^2$ and put

$$M_{\mathrm{trial}} = \lceil \rho^{-1/10} \rceil, \qquad Q_{\mathrm{sim}} = t^2 M_{\mathrm{trial}}.$$

Apply deterministic-classical-query simulation (Theorem 5) using its simulator $\mathsf{Sim}$ with $F = P$, fixed function $C$, $V(u) = W_u$, no witness, and accuracy $\eta = \epsilon$. Its query phase is deterministic, so we implement that phase with access only to $Y$:

1. Initialize the partial function $\tau = \varnothing$ on $P$.
2. On a request $x \in P \setminus \operatorname{dom}\tau$, return $\perp$ if $|\operatorname{dom}\tau| = t - 1$, and otherwise query $Y(x)$ and set $\tau(x)$ to its answer. For every request $x$, supply $C(x)$ if $x \notin P$ and $\tau(x)$ otherwise.
3. Return $\tau$ when the query phase terminates.

For each partial function $\tau$ on $P$ with $|\operatorname{dom}\tau| < t$, run $\mathsf{Sim}(u)$ with answers $\tau(x)$ on $\operatorname{dom}\tau$ and $C(x)$ elsewhere, and let $v_{u,\tau}$ be its output-one probability. Put $v_{u,\perp} = 0$. This defines $\tau_u$ with fewer than $t$ queries to $Y$.

Observe that conditioning on $C$ leaves the joint-inclusion bound for $P$ unchanged. The message length $m_{\mathrm{msg}} = s + t(\lceil \log_2 Q_{\mathrm{sim}} \rceil + 1)$ is polynomial in $\lambda$. Since $\log_2 Q_{\mathrm{sim}} \le \frac{1}{10}\log_2(1/\rho) + O(\log\lambda)$ and $t = s + \lambda^2$, for sufficiently large $\lambda$ we have

$$m_{\mathrm{msg}} + 8\lambda^2 \le t \log_2(1/\rho).$$

The incompressibility bound (Lemma 8) therefore gives $t_{\mathrm{inc}}(\lambda, m_{\mathrm{msg}}) \le t$ for $P$, as required by deterministic-classical-query simulation (Theorem 5). On its good event, $\mathsf{Sim}$ meets fewer than $t$ distinct points of $P$, so our algorithm never returns $\perp$. Its returned $\tau$ reconstructs every answer in

this deterministic query phase, and hence the same final output distribution. The error is therefore at most $\epsilon$, simultaneously for every $u, Y$, while

$$\begin{aligned} Q_{\mathrm{sim}} &\le \mathsf{poly}(\lambda)\rho^{-1/10}, \\ \Pr_{P,C}\left[\begin{array}{c}\exists u, Y: \quad \tau_u(Y) = \perp \text{ or} \\ |v_{u,\tau_u(Y)} - \Pr\left[W_u^{C_P^Y} = 1\right]| > \epsilon\end{array}\right] &\le 2^{-8\lambda^2} + 2^s(|\mathcal{X}|+1)^t t e^{-\epsilon^2 M_{\mathrm{trial}}/(16Q^2)} \\ &\le \rho^4 + 2^{-4\lambda^2} \le 2\rho^4. \end{aligned}$$

Indeed, $\epsilon^2 M_{\mathrm{trial}}/(16Q^2) = 2^{\Omega(\lambda)}$, whereas the logarithm of $2^s(|\mathcal{X}|+1)^t t$ is polynomial in $\lambda$. Although $\mathsf{Sim}$ may make more than polynomially many queries, those outside $P$ use the fixed $C$, so only the fewer than $t$ queries in Step 2 count toward access to $Y$.

**Existential answers.** For each $u$, enumerate the accepting transcripts $\mathcal{T}_{u,w}$ of $M_u(w)$ and construct

$$f_u(\mathcal{O}) = \bigvee_w \bigvee_{\tau \in \mathcal{T}_{u,w}} \bigwedge_{(x,b)\in\tau} [\mathcal{O}(x) = b], \qquad T_u = \mathsf{Switch}_{f_u,t}(P, C|_{P^c}).$$

The accepting-transcript identity (Lemma 10) gives $f_u(\mathcal{O}) = \mathbf{1}[\exists w : M_u^{\mathcal{O}}(w) = 1]$, so running $T_u$ with answers from $Y$ will compute the desired predicate whenever the switching guarantee holds. Choose a polynomial width bound $k \ge 1$, which exists because $M_u$ has polynomial query complexity. The switching theorem (Theorem 7), applied with this $f_u$, bias parameter 1, and independent uniform Boolean function $C$, gives fewer than $t$ queries on every oracle and failure probability $2(8k\rho)^t$. For sufficiently large $\lambda$,

$$k + t \le 2^\lambda, \qquad \rho \le (16k)^{-1}, \qquad 2^{s+1}(8k\rho)^t \le \rho^4.$$

A union bound over $u$ and the acceptance-estimation failures proves the lemma.

### 10.1.2 Constructing the recursive answers

We now prove Lemma 15 by constructing each interval from the earlier answers, while keeping one independent group unexposed in every oracle function.

Put

$$\rho_i = \rho_0^{(256m)^{-(i-1)}} \quad (i \in [d+1]), \qquad \mathcal{I}_i = [\lambda^{2^i}, \lambda^{2^{i+1}}) \cap [0, L] \quad (i \in [d]),$$

so $\rho_{d+1} = \nu$. Sample $\mathcal{O}_{\mathrm{pre}}$, the $N, A$ answers below $\lambda^2$, conditionally on the prefix specified by $I$, and set $E_0 = (I, \mathcal{O}_{\mathrm{pre}})$. Since this prefix uses only $\mathcal{O}_{\mathrm{fix}}$ below $\lambda$ and independent coins, conditioning on $E_0$ preserves the independent groups and uniform Boolean functions.

For each valid $N$ or $A$ query at address $u$, write $\langle M_u, x_u\rangle$ for the padded encoding of its circuit and input. The circuit $M_u$ is deterministic for $N$ and quantum for $A$, and both query the full oracle $\mathcal{O}$. Address padding ensures that every such query reaches an earlier interval when $u \in \mathcal{I}_i$, so before constructing that interval we have $E_{i-1}$ and an algorithm $\mathsf{Rec}_{<i}$ satisfying, on success,

$$\mathsf{Rec}_{<i}^{\mathbf{T}}(u) = \mathcal{O}(u) \quad (|u| \le L,\ |u| < \lambda^{2^i}), \qquad \Pr[S \subseteq G \mid E_{i-1}] \le \rho_i^{|S|} \quad (|S| \le 2^\lambda)$$

for each of the $m - i + 1$ independent groups $G$ still available in every oracle function. All query counts are polynomial for every choice of $\mathbf{T}$. The sampling of $B^{(1)}, \dots, B^{(m)}, T$ above initializes these conditions.

**Step 1: Construct $T_u$ and $\tau_u$.** We first express the next interval's answers through classical queries to the groups that remain after this step. Number the groups of $V_j$ by $(G_j^{(r)}, B_j^{(r)})_{r\in[m-j]}$, with $V_0 = T$ and $(G_0^{(r)}, B_0^{(r)}) = (G^{(r)}, B^{(r)})$. Initialize $E_i = E_{i-1}$ and expose $(G_j^{(i-j)}, B_j^{(i-j)})$ for each $j < i$. Let $P$ contain $(j, u)$ exactly when $u$ belongs to that exposed group, and set $C(j, u)$ to the exposed function's value at $u$. Write $Y(j, u)$ for the value at $u$ of the Boolean function associated with the first still unexposed group omitting $u$, or the final correction bit if every remaining group contains $u$. The algorithm $\mathsf{Rec}_{<i}$ answers earlier exposed groups using $E_{i-1}$, so its remaining queries use $C_P^Y(j, u)$. Substitute $\mathsf{Rec}_{<i}$ into the circuits defining the $N, A$ answers in $\mathcal{I}_i$, coherently for the quantum circuits (Lemma 6), with their inputs $x_u$ fixed, then apply the one-restriction lemma (Lemma 16) with

$$\gamma_i = \rho_i^{1/128}, \qquad \epsilon_i = b\gamma_i/1000 \geq \rho_i^{1/40}.$$

The resulting circuits are fixed conditional on $E_{i-1}$ with polynomial query, index and witness bounds, and $(P, C)$ is independent of them with inclusion bound $\rho_i^{|S|}$. The displayed inequality follows from $b \geq \nu^{1/256} \geq \rho_i^{1/256}$ for sufficiently large $\lambda$. Thus the lemma supplies an existential algorithm $T_u$ at each $N$ address and an algorithm $\tau_u$ at each $A$ address, satisfying

$$T_u^Y = \mathbf{1}[\exists w : M_u^{\mathsf{Rec}_{<i}^{C_P^Y}}(x_u, w) = 1],$$
$$v_u(Y) = \Pr\left[M_u^{\mathsf{Rec}_{<i}^{C_P^Y}}(x_u) = 1\right], \qquad \sup_Y |v_u(Y) - v_{u,\tau_u(Y)}| \leq \epsilon_i.$$

Add $T_u, \tau_u$ to $E_i$. If this simultaneous conclusion fails, set $\mathsf{Bad} = 1$ and use the completion rule below instead of proceeding to Step 2.

**Step 2: Sample the $A$ answers.** Set $a = b/4$ and sample independently at each $A$ address $u$, defining $L_{u,\tau}$ for every possible output $\tau$, including $\perp$:

$$F_u \leftarrow \text{Bernoulli}(2a), \qquad U_u \leftarrow [0, 1], \qquad L_{u,\tau} = \mathbf{1}\left[U_u \leq \frac{h_u(v_{u,\tau}) - a}{1 - 2a}\right].$$

Add $F_u, (L_{u,\tau})_\tau$ to $E_i$. Define the new Boolean function $V_i$ using $m-i$ groups, each including every address independently with probability $\gamma_i^{1/(m-i)}$, and independent uniform Boolean functions. At each $u$, independently sample

$$C_i(u) \leftarrow \text{Bernoulli}(r_u), \qquad r_u = \frac{1}{2} + \frac{h_u(v_u(Y)) - h_u(v_{u,\tau_u(Y)})}{2a\gamma_i}, \tag{48}$$

where $Y$ gives the answers from the remaining groups in shorter intervals. Set $V_i(u)$ to the value at $u$ of the uniform Boolean function associated with the first group omitting $u$, or to $C_i(u)$ if every group contains $u$. To answer $u$, if $F_u = 0$, run $\tau_u$ with $Y$ and return $L_{u,\tau_u(Y)}$. If $F_u = 1$, query $V_i(u)$ and return its answer. At an $N$ address, run $T_u$.

Since $b \leq h_u(v) \leq 1 - b$, the thresholds defining $L_{u,\tau}$ lie in $[0, 1]$. Moreover, the estimate from Step 1 gives

$$|h_u(v_u(Y)) - h_u(v_{u,\tau_u(Y)})| \leq \epsilon_i, \qquad |r_u - 1/2| \leq \epsilon_i/(2a\gamma_i) = 1/500.$$

Conditional on $\mathcal{O}_{\mathsf{base}}$, $Y$ and $(T_u, \tau_u)_u$ from Step 1, the new coins give answer-one probability

$$h_u(v_{u,\tau_u(Y)}) + 2a\gamma_i(r_u - 1/2) = h_u(v_u(Y)).$$

Since distinct $u$ use independent coins and $v_u(Y)$ is determined by the actual shorter answers, averaging over $Y$ and $(T_u, \tau_u)_u$ gives precisely the required product of Bernoulli distributions.

**Step 3: Continue to the next interval.** Keep the exposed groups in $E_i$ and continue with the $m-i$ unexposed groups in each function's sampling, including those just sampled. Their inclusion bounds are at most $\rho_{i+1}^{|S|}$ because

$$\gamma_i^{1/(m-i)} = \rho_i^{1/(128(m-i))} \le \rho_{i+1}.$$

We now have the required number of independent groups with the next inclusion bound. Moreover, the algorithms and bits in $E_i$ were computed for every possible $Y$ before the actual shorter answers were used, so conditioning on $E_i$ preserves the independence of the unexposed groups and the uniform samples used to generate $C_i(u)$ with their $Y$-dependent means $r_u$. The construction can therefore continue to interval $i+1$.

### 10.1.3 The final algorithm and its distribution

Write $E = E_d$ and set $(G_j, B_j) = (G_j^{(m-j)}, B_j^{(m-j)})$. For $j \ge 1$, let $T_j$ equal $C_j$ on $G_j$ and $B_j$ elsewhere. The earlier groups belong to $E$, so define $\mathsf{Read}_{j,r,E}^{T_j}(u)$ as follows:

1. For $\ell = r, \dots, m-j-1$, if $u \notin G_j^{(\ell)}$, return $B_j^{(\ell)}(u)$.
2. Otherwise query $T_j(u)$ and return its answer.

This uses at most one oracle query. A query to $V_j$ uses $r = 1$. For $u \in \mathcal{I}_i$, run $T_u$ or $\tau_u$ by answering each $Y(j,v)$ query with $\mathsf{Read}_{j,i-j+1,E}^{T_j}(v)$. Denote these algorithms by $T_u^E, \tau_u^E$, whose only oracles are $(T_j)_j$. On input $u$, return zero if $|u| > L$, the recursive query is invalid, or $\mathsf{Bad}(E) = 1$. Otherwise return

$$\mathsf{Rec}_E^{\mathbf{T}}(u) = \begin{cases} \mathcal{O}_{\mathrm{fix}}(x), & u \text{ queries } \mathcal{O}_{\mathsf{base}} \text{ at } x \notin \mathcal{X}, \\ \mathsf{Read}_{0,1,E}^{T_0}(x), & u \text{ queries } \mathcal{O}_{\mathsf{base}} \text{ at } x \in \mathcal{X}, \\ \mathcal{O}_{\mathrm{pre}}(u), & u \text{ is an } N, A \text{ address below } \lambda^2, \\ (T_u^E)^{\mathbf{T}}, & u \text{ is an } N \text{ address in } \mathcal{I}_i, \\ L_{u,\tau_u^E(\mathbf{T})}, & u \text{ is an } A \text{ address in } \mathcal{I}_i,\ F_u = 0, \\ \mathsf{Read}_{i,1,E}^{T_i}(u), & u \text{ is an } A \text{ address in } \mathcal{I}_i,\ F_u = 1, \\ 0, & \text{otherwise.} \end{cases}$$

Every value other than a $T_j$ answer is specified by $E$, and the algorithm evaluates $L_{u,\tau_u^E(\mathbf{T})}$ without computing an acceptance probability. Nested calls reach strictly shorter intervals, so the algorithm terminates with polynomial query count for all $\mathbf{T}$, since $d$ is constant. Coherent classical computation (Lemma 6) implements its answer unitary with at most twice as many queries.

On failure, expose any remaining groups belonging to $E_{\mathsf{base}}$ and set $\mathsf{Bad}(E_d) = 1$. Unused pairs $(G_j, B_j)$ for $j \ge 1$ are sampled independently with $T_j = B_j$, while $(G_0, B_0) = (G^{(m)}, B^{(m)})$. In either case, sample every remaining recursive answer from its original conditional distribution. Thus the construction preserves $(Z, \mathcal{O})$, with $\Pr[\mathsf{Bad}] \le 3\sum_{i=1}^{d} \rho_i^4 \le 3d\nu^4$. The description $E_d$ uses only $E_{\mathsf{base}}$ and independent coins. Conditional on successful $E_d$, the newly sampled sets $(G_j)_{j\ge1}$ and functions $(B_j)_{j\ge1}$ are mutually independent and independent of $(Z, B^{(m)})$, every $B_j$ is uniform, $T_j|_{G_j^c} = B_j|_{G_j^c}$, and $\Pr[u \in G_j \mid E_d] \le \nu$. Indeed, $T_u, \tau_u$ are constructed before the actual $Y$ is used, and every $L_{u,\tau}$ is determined by the same $U_u$, which is independent of the unused groups.

# 11 Preparing Quantum States from Boolean Oracles

We prepare real-amplitude states from Boolean oracles. The state-preparation bound (Lemma 17) also controls the change in sampling probabilities when the target changes.

**Definition 27** (Preparation oracles)**.** *For an $n$-qubit real-amplitude state $|\Omega\rangle$, sample its* preparation oracles $(R, S) \leftarrow \mathcal{D}_{\mathrm{prep}}(|\Omega\rangle)$ *by*

$$\begin{aligned} R &\leftarrow \{f : \{0,1\}^n \to \{0,1\}\}, \qquad U_f |x\rangle = (-1)^{f(x)} |x\rangle, \\ p_{\Omega,R}(x) &= \frac{1}{2} - \frac{1}{2} \operatorname{clip}_{[-1/2,1/2]} \left( \frac{2^{n/2} \langle x| \mathrm{H}^{\otimes n} U_R |\Omega\rangle}{5n} \right) \in [1/4, 3/4], \\ S(x,y) &\leftarrow \mathrm{Bernoulli}(p_{\Omega,R}(x)) \quad \textit{independently for } (x,y) \in \{0,1\}^n \times \{0,1\}^{4n}. \end{aligned} \tag{49}$$

*Here $U_f$ is the phase oracle for $f$, and both $R$ and $S$ permit quantum queries.*

We use the preparation error bound $\epsilon_{\mathrm{prep}}(n) = 2^{-n}$.

**Lemma 17** (State preparation)**.** *A uniform* QPT *algorithm* $\mathsf{Prepare}^{R,S}(1^n)$ *outputs $\rho_{R,S}$ such that, for all $n$-qubit real-amplitude states $|\Omega\rangle, |\Omega^0\rangle$ and sufficiently large $n$,*

$$\begin{aligned} \mathbb{E}_{(R,S)\leftarrow\mathcal{D}_{\mathrm{prep}}(|\Omega\rangle)} \tfrac{1}{2} \left\| \rho_{R,S} - |\Omega\rangle\langle\Omega| \right\|_1 &\le \epsilon_{\mathrm{prep}}(n), \\ \Pr_R \left[ \max_x |p_{\Omega,R}(x) - p_{\Omega^0,R}(x)| > \tfrac{1}{4} \| |\Omega\rangle - |\Omega^0\rangle \|_2 \right] &\le 2^{-n^2}. \end{aligned}$$

The proof of Lemma 17 appears in Section 11.1.

For $r$ preparations with the same $(R, S)$, changing one tensor factor at a time gives

$$\mathbb{E}_{Z,R,S} \tfrac{1}{2} \left\| \rho_{R,S}^{\otimes r} - (|\Omega_Z\rangle\langle\Omega_Z|)^{\otimes r} \right\|_1 \le r \epsilon_{\mathrm{prep}}(n), \qquad (R,S) \mid Z \sim \mathcal{D}_{\mathrm{prep}}(|\Omega_Z\rangle). \tag{50}$$

Here $Z$ specifies the target. Tensoring both states with any jointly sampled density matrix $\sigma_W$ preserves the bound, since $\|\sigma_W \otimes X\|_1 = \|X\|_1$ pointwise.

## 11.1 Proof of state preparation

We now prove Lemma 17 using the following preparation algorithm. We identify its unnormalized output to bound both the error and the success probability.

**Preparing a state from Boolean oracles**

$\mathsf{Prepare}^{R,S}(1^n)$

1. Repeat at most $4(5n)^3$ times, using fresh registers:
   1.1. Prepare $U_S |+\rangle^{\otimes 5n}$, apply $\mathrm{H}^{\otimes 4n}$ to the last $4n$ qubits, and measure them.
   1.2. On outcome $0^{4n}$, apply $U_R \mathrm{H}^{\otimes n}$ to the remaining register and return it.
2. Return $|0^n\rangle$ if every attempt fails.

Put $m = 5n$. The following coefficient bound controls both the preparation error and a change of target: for every fixed real vector $|v\rangle$, Hoeffding's inequality and a union bound give

$$\Pr_R \left[ \max_x 2^{n/2} |\langle x| \mathrm{H}^{\otimes n} U_R |v\rangle| > \frac{m}{2} \| |v\rangle \|_2 \right] \le 2^{-n^2}, \tag{51}$$

because each term is a sum of independent signed coefficients whose squares sum to $\| |v\rangle \|_2^2$.

**Preparation error.** Apply the coefficient bound (Equation (51)) to $|v\rangle = |\Omega\rangle$, and let $\mathsf{Good}$ be the event that no coefficient is clipped. For fixed $(R, S)$, define the unnormalized vector

$$\bar{\sigma}_x = 2^{-4n} \sum_y (-1)^{S(x,y)}, \qquad |\eta\rangle = U_R \mathrm{H}^{\otimes n} \left( 2^{-n/2} \sum_x \bar{\sigma}_x |x\rangle \right).$$

The zero measurement outcome followed by $U_R \mathrm{H}^{\otimes n}$ gives this unnormalized vector, so each attempt succeeds with probability $\| |\eta\rangle \|_2^2$ and returns $|\widehat{\eta}\rangle = |\eta\rangle / \| |\eta\rangle \|_2$ on success. For $R \in \mathsf{Good}$ the clipping is inactive, so the means of the signs and independence of the bits of $S$ give

$$\mathbb{E}_S |\eta\rangle = |\Omega\rangle / m, \qquad \mathbb{E}_S \| |\eta\rangle - |\Omega\rangle / m \|_2^2 = 2^{-n} \sum_x \mathrm{Var}(\bar{\sigma}_x) \le 2^{-4n}. \tag{52}$$

To turn this estimate into an output bound, put $d = \| |\eta\rangle - |\Omega\rangle / m \|_2$. If $d \le 1/(2m)$, normalization and the triangle inequality bound the success probability and error, while independent attempts give the failure bound:

$$\| |\eta\rangle \|_2^2 \ge \frac{1}{4m^2}, \qquad \tfrac{1}{2} \| |\widehat{\eta}\rangle\langle\widehat{\eta}| - |\Omega\rangle\langle\Omega| \|_1 \le 2md, \qquad \Pr\big[\text{all } 4m^3 \text{ attempts fail}\big] \le e^{-m}.$$

Thus the output is within $2md + e^{-m}$ of the target when $d \le 1/(2m)$, and this bound also holds for larger $d$ because trace distance is at most $1 \le 2md$. Averaging and applying Cauchy–Schwarz to the second-moment estimate (Equation (52)) now gives

$$\mathbb{E}_{R,S} \tfrac{1}{2} \| \rho_{R,S} - |\Omega\rangle\langle\Omega| \|_1 \le 2m 2^{-2n} + e^{-m} + 2^{-n^2} \le \epsilon_{\mathrm{prep}}(n).$$

**Changing the target.** The same coefficient estimate also controls the change in the Boolean sampling probabilities. Write $p_x = p_{\Omega,R}(x)$ and $p_x^0 = p_{\Omega^0,R}(x)$. Since $|\operatorname{clip}_{[-1/2,1/2]}(a) - \operatorname{clip}_{[-1/2,1/2]}(b)| \le |a - b|$, applying the coefficient bound (Equation (51)) to $|\Delta\rangle = |\Omega\rangle - |\Omega^0\rangle$ gives

$$\Pr_R[\mathsf{Good}_\Delta] \ge 1 - 2^{-n^2}, \qquad \mathsf{Good}_\Delta := \left\{ \max_x |p_x - p_x^0| \le \tfrac{1}{4} \| |\Delta\rangle \|_2 \right\}. \tag{53}$$

This proves the claimed bound on the sampling probabilities.

# 12 Changing Preparation Targets

We show that exponentially close targets give indistinguishable preparation oracles even with quantum access to their recursive extension (Theorem 13).

**Theorem 13** (Changing a preparation target)**.** *Fix polynomial bounds on query count and length, and $c_* > 0$. For sufficiently small $c > 0$, fix a Boolean oracle $\mathcal{O}_{\mathrm{fix}}$, real-amplitude $n$-qubit states $|\Omega^0\rangle, |\Omega^1\rangle$, and functions $h_u$ satisfying*

$$n \geq \lambda, \qquad \| |\Omega^0\rangle - |\Omega^1\rangle \|_2 \leq 2^{-c_*\lambda},$$
$$2^{-c\lambda} \leq h_u(v) \leq 1 - 2^{-c\lambda}, \qquad |h_u(v) - h_u(w)| \leq |v - w| \quad (v, w \in [0,1]),$$

*and sample the preparation oracles (Definition 27) and their recursive extension with means $h_u$ (Definition 26):*

$$(R^e, S^e) \leftarrow \mathcal{D}_{\mathrm{prep}}(|\Omega^e\rangle), \qquad \mathcal{O}^e = (\mathcal{O}_{\mathrm{fix}}, R^e, S^e, N^e, A^e) \quad (e \in \{0,1\}).$$

*Any common supported $N, A$ prefix below $\lambda^2$ may be fixed.*

*For every quantum protocol $\mathcal{A}$ obeying the query bounds, allow unrestricted local computation and let its input density matrix $\rho$ and local channels depend on both targets and $\mathcal{O}_{\mathrm{fix}}$. Uniformly over these choices, the averaged outputs satisfy*

$$\omega_e = \mathbb{E}_{\mathcal{O}^e} \mathcal{A}^{\mathcal{O}^e}(\rho), \qquad \tfrac{1}{2} \|\omega_0 - \omega_1\|_1 \leq 2^{-\Omega(\lambda)}. \tag{54}$$

The proof of Theorem 13 appears in Section 12.1.

## 12.1 Proof of the output bound

We now prove Theorem 13. We first sample the two preparation oracles so that they can differ only on an intersection of sparse sets. Recursive simulation then removes the $N, A$ queries, after which a query hybrid bounds the effect of the remaining difference.

Fix the choices in the statement, and let $Q, L$ bound the protocol's query count and length. In each experiment, set the sampled base answers above length $L$ to zero, which changes neither the queried recursive answers nor the output. Choose $m$ for this $L$ from classical recursive simulation (Theorem 12), and put

$$0 < c_0 < c_*/m, \qquad \gamma = 2^{-mc_0\lambda}, \qquad \rho_0 = \gamma^{1/m}.$$

Each experiment returns the original protocol's full output and changes only its oracle queries.

**First hybrid: sample the preparation bits through sparse sets.** Let $\mathsf{H}_{e,0}$ sample $\mathcal{O}^e$ as in the target-change experiments (Theorem 13) and run $\mathcal{A}^{\mathcal{O}^e}(\rho)$, so its averaged output is $\omega_e$. In $\mathsf{H}_{e,1}$, first sample a uniform function $R$, then write $p^e_u = p_{\Omega^e,R}(x)$ for $u = (x, y)$ and sample independently at each such input

$$F_u \leftarrow \mathrm{Bernoulli}(1/2), \qquad C_u \leftarrow \mathrm{Bernoulli}(2p^0_u - 1/2),$$
$$\mathbf{1}[u \in P_i] \leftarrow \mathrm{Bernoulli}(\rho_0) \quad (i \in [m]), \qquad B(u) \leftarrow \mathrm{Bernoulli}(1/2),$$
$$Z_u \leftarrow \mathrm{Bernoulli}(q_u), \qquad q_u = \mathrm{clip}_{[0,1]}\left(\frac{1}{2} + \frac{2(p^1_u - p^0_u)}{\gamma}\right).$$

Write $P = \bigcap_{i=1}^{m} P_i$ and define

$$T^0 = B, \qquad T^1(u) = \begin{cases} Z_u, & u \in P, \\ B(u), & u \notin P, \end{cases}$$

$$\widehat{S}^e(u) = \begin{cases} C_u, & F_u = 0, \\ T^e(u), & F_u = 1. \end{cases}$$

Sample

$$\widehat{\mathcal{O}}^e = (\mathcal{O}_{\mathrm{fix}}, R, \widehat{S}^e, \widehat{N}^e, \widehat{A}^e)$$

with the fixed prefix and recursive means $h_u$, then run $\mathcal{A}^{\widehat{\mathcal{O}}^e}(\rho)$ and return its output.

The state-preparation bound (Lemma 17) gives

$$\Pr_R[\neg \mathsf{Good}] \le \eta := 2^{-n^2}, \qquad \mathsf{Good} \Longrightarrow \max_{u,e} |p_u^e - p_u^0| \le \tfrac{1}{4} \| \left|\Omega^1\right\rangle - \left|\Omega^0\right\rangle \|_2 \le \gamma/4.$$

$\widehat{S}^0$ has exactly the prescribed distribution for every $R$, since $\Pr\left[\widehat{S}^0(u) = 1 \mid R\right] = p_u^0$. On $\mathsf{Good}$, $q_u$ is unclipped and

$$\Pr\left[\widehat{S}^1(u) = 1 \mid R\right] = \frac{1}{2}(2p_u^0 - 1/2) + \frac{1}{2}\left(\frac{1-\gamma}{2} + \gamma q_u\right) = p_u^1.$$

Using the same preparation answers and recursive coins whenever the distributions agree therefore gives

$$\mathsf{H}_{0,0} = \mathsf{H}_{0,1}, \qquad \tfrac{1}{2} \|\mathsf{H}_{1,0} - \mathsf{H}_{1,1}\|_1 \le \eta.$$

**Second hybrid: simulate the recursive queries.** We now simulate the $N, A$ queries. Choose $c > 0$ from classical recursive simulation (Theorem 12) for $L, c_0$, and condition on

$$I = (R, (F_u, C_u)_u, \text{fixed recursive prefix}).$$

Restrict $T^e$ to the preparation inputs $\mathcal{X} = \{u = (x, y) : |u| \le L,\ F_u = 1\}$. Conditional on $I$, $\widehat{S}^e(u) = C_u$ for $|u| \le L$ outside $\mathcal{X}$, and all other base answers are fixed. The sets $P_i$ have inclusion bound $\rho_0^{|S|}$, and $B$ is uniform and independent of them. Thus classical recursive simulation applies, with an independent uniform choice $e \leftarrow \{0, 1\}$ included in its auxiliary variable $Z$. It supplies a classical-query deterministic algorithm $\mathsf{Answer}_E^{T^e}$ with polynomial query bound $t_{\mathsf{sim}}$ and

$$E = \Phi_I(P_1, \ldots, P_{m-1}, B, \xi), \qquad \xi \perp (e, (P_i)_i, B, (Z_u)_u) \mid I.$$

Let $\mathsf{H}_{e,2}$ replace every oracle query in $\mathsf{H}_{e,1}$ by this algorithm, coherently evaluated and uncomputed. For each $e$, the theorem gives

$$\tfrac{1}{2} \|\mathsf{H}_{e,1} - \mathsf{H}_{e,2}\|_1 \le \epsilon_{\mathrm{rec}} = 2^{-\Omega(\lambda)}.$$

Indeed, its joint bound includes $e$, and each value has probability $1/2$, so the conditional errors are at most twice that bound.

**Third hybrid: remove the remaining sparse change.** The first simulation already uses $\mathsf{Answer}_E^B$, since $T^0 = B$. Conditional on $I$, $(E, B)$ is independent of $(e, P_m)$, so we compare $\mathsf{H}_{1,2}$ directly with $\mathsf{H}_{0,2}$. The weight $w(u)$ of an internal query in $\mathsf{H}_{0,2}$ is independent of $P_m$, and $T^1 = B$ outside $P \subseteq P_m$. Thus

$$\mathbb{E} \sum_u w(u)\mathbf{1}[T^1(u) \neq B(u)] \leq \rho_0 \, \mathbb{E} \sum_u w(u) \leq \rho_0.$$

Use $\mathsf{H}_{0,2}$ for the prefix at each changed query. Coherent evaluation and uncomputation use at most $2Qt_{\mathsf{sim}}$ internal queries, so the quantum query bound (Lemma 5) gives

$$\tfrac{1}{2} \left\| \mathsf{H}_{1,2} - \mathsf{H}_{0,2} \right\|_1 \leq 4Qt_{\mathsf{sim}}\sqrt{\rho_0}.$$

Combining the transitions now proves

$$\tfrac{1}{2} \left\| \mathsf{H}_{0,0} - \mathsf{H}_{1,0} \right\|_1 \leq \eta + 2\epsilon_{\mathrm{rec}} + 4Qt_{\mathsf{sim}}\sqrt{\rho_0} = 2^{-\Omega(\lambda)}.$$

Only oracle queries were changed, so each party still applies its original channels, with no additional communication.

## 13 Encoded Functions with Quantum Keys

We evaluate encoded functions using predicate-preimage states and simulate their recursive oracle extension using direct function queries (Theorem 14).

**Definition 28** (Encoded function and evaluation)**.** *For $\mathfrak{F} : \{0,1\}^\kappa \to \{0,1\}$, a hash $H$ and $B : \mathcal{X}_\kappa \to \{0,1\}$, where $\mathcal{X}_\kappa = \Sigma_\kappa^{q_\kappa} \times \{0,1\}^\kappa$, define the encoded function on the code $C_\kappa \subseteq \Sigma_\kappa^{q_\kappa}$ by*

$$T_{H,B,\mathfrak{F}}(v,y) = \begin{cases} \mathfrak{F}(y), & v \in C_\kappa \text{ and } H(v) = y \| 0^{q_\kappa - \kappa}, \\ B(v,y), & \text{otherwise.} \end{cases}$$

*Its evaluation is $\mathsf{EvalHash}^T = U_{\mathsf{Decode}}^{-1} U_T U_{\mathsf{Decode}}$, where the decoder uses $|\Psi_H\rangle$ and leaves $y$ unchanged (Definition 15 and Lemma 7). Queries outside $\mathcal{X}_\kappa$, including $v = \perp$, receive zero.*

For the encoded-function oracle, fix a quantum protocol $\mathcal{A}$ with polynomial query count and length, $\mathcal{O}_{\text{aux}} : \{0,1\}^* \to \{0,1\}$, $K \subseteq J \subseteq \{0,1\}^{\leq \mathsf{poly}(\lambda)}$, and $\lambda \leq \kappa_j \leq \mathsf{poly}(\lambda)$. Sample a classical variable $\Theta$ containing $(\mathfrak{F}_j)_{j \in K}$, and for each $j \in J$ independently sample $H_j \leftarrow \mathcal{H}_{\kappa_j}$ and a uniform Boolean function $B_j : \mathcal{X}_{\kappa_j} \to \{0,1\}$. Define

$$\mathcal{O} = (\mathcal{O}_{\text{aux}}, (T_{H_j,B_j,\mathfrak{F}_j})_{j \in J}, N, A). \tag{55}$$

Here the other $\mathfrak{F}_j$ may depend arbitrarily on $\Theta, (H_j, B_j)_j$, and $N, A$ are the recursive extension (Definition 26). Queries $(j, v, y)$ return $T_{H_j,B_j,\mathfrak{F}_j}(v, y)$. A protocol opens $j \in K$ by a classical request for $\left|\Psi_{H_j}\right\rangle$, using the same hash on repeated requests. Initially no index is open. The querying parties know the current set $K_t$ of opened indices. The input and response channels may depend on $\Theta$, all $B_j$, and only the opened hashes.

**Theorem 14** (Evaluation and simulation of encoded functions)**.**

1. *For $T = T_{H,B,\mathfrak{F}}$ with $H \in \mathsf{Good}_\kappa$, let $\rho_q, \sigma_q$ be the joint outputs of $q$ calls to $\mathsf{EvalHash}^T$ or $\mathfrak{F}$, starting with $|\Psi_H\rangle$ and any other input, with identical intervening operations outside the key. Then*
$$\tfrac{1}{2} \|\rho_q - \sigma_q\|_1 \leq 2q\sqrt{\delta_{\text{dec}}(\kappa)}, \qquad \delta_{\text{dec}}(\kappa) = 2^{1-\kappa}. \tag{56}$$

2. *There is a fixed coherent query algorithm $\mathcal{S}_\eta^{(\mathfrak{F}_j)_{j \in K}}(K_t, \cdot)$ making polynomially many queries only to $\mathfrak{F}_j$ with $j \in K_t$, and answering unopened encodings with $B_j$. Its one classical sample $\eta \perp \Theta$ is jointly sampled with $(H_j, B_j)_j$. Replace each party's oracle queries in $\mathcal{A}$ by $\mathcal{S}_\eta(K_t, \cdot)$, keeping the original input and response channels and using the same $H_j$ for key requests. The averaged joint outputs $\rho_{\text{real}}, \rho_{\text{sim}}$, including $\Theta$, satisfy, uniformly for fixed query bounds,*
$$\tfrac{1}{2} \|\rho_{\text{real}} - \rho_{\text{sim}}\|_1 \leq 2^{-\Omega(\lambda)}. \tag{57}$$

Note that ideal evaluation leaves the key unchanged.

### 13.1 Proof of evaluation and simulation

We now prove Theorem 14. We first bound the error in replacing an encoded query by a direct function query, then construct one simulator for the entire protocol.

**Evaluation.** For $T = T_{H,B,\mathfrak{F}}$ and $H \in \mathsf{Good}_\kappa$, the decoder makes no oracle queries and satisfies

$$\forall y : \qquad \Pr[(\mathsf{Decode}(y, |\Psi_H\rangle), y) \notin \mathcal{P}_H] \le \delta_{\mathrm{dec}}(\kappa) \tag{58}$$

by the definition of good hashes (Definition 16). Purify the other input as $|\phi\rangle$, apply $U_{\mathsf{Decode}}$ to $|\Psi_H\rangle\,|\phi\rangle$, and let $\Pi$ project onto $\mathcal{P}_H$. Since decoding does not change $y$, the decoded state's weight outside $\Pi$ is at most $\delta_{\mathrm{dec}}(\kappa)$. The queries to $T$ and $\mathfrak{F}$ agree on $\Pi$, and the ideal query $U_{\mathfrak{F}}\,|y, b\rangle = |y, b \oplus \mathfrak{F}(y)\rangle$ commutes with $U_{\mathsf{Decode}}$. Reversing the decoding therefore gives

$$\|\mathsf{EvalHash}^T(|\Psi_H\rangle\,|\phi\rangle) - |\Psi_H\rangle\,U_{\mathfrak{F}}\,|\phi\rangle\,\|_2 \le 2\sqrt{\delta_{\mathrm{dec}}(\kappa)}.$$

The one-call estimate extends to repeated use by replacing calls in chronological order with $U_{\mathfrak{F}}$. Before the differing call, the ideal operations have left the key in $|\Psi_H\rangle$, so the same estimate applies. The remaining channels cannot increase trace distance, so the triangle inequality gives the stated $2q\sqrt{\delta_{\mathrm{dec}}(\kappa)}$ joint error.

**The query algorithm and experiments.** We construct the simulator by first answering recursive queries through the encodings, then removing the unopened encodings. Fix a polynomial bound $L$ on the encoded addresses and all protocol queries. Choose the constant number $m$ of groups from classical recursive simulation (Theorem 12). For each $j$, partition the suffix predicates into consecutive groups $J_{j,1}, \dots, J_{j,m}$ whose sizes differ by at most one, and put

$$P_{j,i} = \{(v, y) \in C_{\kappa_j} \times \{0,1\}^{\kappa_j} : H_{j,[\kappa_j]}(v) = y,\ H_{j,J_{j,i}}(v) = 0\} \qquad (i \in [m]).$$

These are the suffix-constraint sets (Definition 17). Conditional on the prefix predicates, the codeword-hash bound (Lemma 7, item 1) gives independent groups with, for distinct $u_1, \dots, u_t$,

$$\Pr[u_1, \dots, u_t \in P_{j,i}] \le 2^{-\kappa_j t/(8m)} \le 2^{-c_0 \lambda t} \qquad (t \le 2^\lambda)$$

with $c_0 = 1/(8m)$, since $\kappa_j \ge \lambda$. For the component addresses, put

$$P_i = \{(j, u) : j \in J,\ u \in P_{j,i}\}.$$

Independence across $j$ gives the same inclusion bound for $P_i$, and every such address has length at least $\kappa_j \ge \lambda$. Since each encoding $T_j = T_{H_j,B_j,\mathfrak{F}_j}$ equals $B_j$ outside $\bigcap_i P_{j,i}$, apply classical recursive simulation (Theorem 12) with fixed oracle $\mathcal{O}_{\mathrm{aux}}$ and $h_u(v) = (1+v)/3$, whose range and difference bound are $1/3 \le h_u(v) \le 2/3$ and $|h_u(v) - h_u(w)| = |v - w|/3$. Include $\Theta$, every original hash and noise function, and the prescribed channels in its auxiliary variable $Z$. The theorem supplies a deterministic polynomial-query algorithm $\mathsf{Answer}_E^{(T_j)_j}$, where $E$ uses only the prefix predicates, the first $m-1$ suffix groups, the original $B_j$ and independent coins. The algorithm answers queries to every original encoding $T_j$ exactly. Set $\eta = (E, (H_j, B_j)_j)$, which is independent of $\Theta$. We now define the simulator using this sample and direct access to the opened functions.

$\mathcal{S}_\eta^{(\mathfrak{F}_j)_{j\in K_t}}(K_t,u)$

Run $\mathsf{Answer}_E(u)$ reversibly, answering its encoding queries $(j,v,y)$, with $(v,y)\in\mathcal{X}_{\kappa_j}$, by

$$\widetilde{T}_{j,K_t}(v,y)=\begin{cases}\mathfrak{F}_j(y), & j\in K_t \text{ and } (v,y)\in\mathcal{P}_{H_j},\\ B_j(v,y), & \text{otherwise.}\end{cases}$$

XOR the output into the answer bit and reverse the computation, using the same function queries.

Membership in $\mathcal{P}_{H_j}$ and the values of $B_j$ are determined by $\eta$, so this algorithm makes polynomially many queries only to the opened $\mathfrak{F}_j$. On a classical request $j$, set $K_t=K_t\cup\{j\}$, apply the prescribed response channel, and supply $|\Psi_{H_j}\rangle$ using the same $H_j$ on repeated requests.

We analyze the following experiments, with the original sample, input and response channels.

1. In $\mathsf{H}_0$, answer oracle queries with $\mathcal{O}$.
2. In $\mathsf{H}_1$, answer them by coherent evaluation of $\mathsf{Answer}_E^{(T_j)_j}$.
3. In $\mathsf{H}_2$, answer them with $\mathcal{S}_\eta(K_t,\cdot)$.

Write $\mathsf{H}_b$ also for the averaged output density matrix, including $|\Theta\rangle\langle\Theta|$. The recursive simulation theorem preserves the full original sampling and gives

$$\tfrac{1}{2}\left\|\mathsf{H}_0-\mathsf{H}_1\right\|_1\le 2^{-\Omega(\lambda)}.$$

The simulator in $\mathsf{H}_2$ makes no original $N,A$ queries.

**Removing unopened encodings.** It remains to justify replacing the unopened encodings by $B_j$. For this purpose, condition on $\Theta$, the prefix predicates, all $B_j$, the first $m-1$ suffix groups, the coins defining $E$, and the opened hashes and request history. In $\mathsf{H}_2$, the input and response channels use only opened hashes, and every queried $\mathfrak{F}_j$ is contained in $\Theta$. Conditional on the prefix predicates, induction on the classical requests therefore leaves each unopened suffix $H_{j,J_{j,m}}$ with its original distribution, independent of the remaining conditioned information. The conditioned information determines the query density matrix. If $w_t(j,u)$ is the weight of an internal query on address $(j,u)$, the conditional inclusion bound for $P_{j,m}$ therefore gives

$$\mathbb{E}\sum_{j\notin K_t,u} w_t(j,u)\mathbf{1}[u\in P_{j,m}]\le 2^{-c_0\lambda}\mathbb{E}\sum_{j\notin K_t,u} w_t(j,u)\le 2^{-c_0\lambda}. \tag{59}$$

This total query-weight bound controls the whole protocol. Replace the internal queries of $\mathsf{H}_1$ by those of $\mathsf{H}_2$ in chronological order. Use $\mathsf{H}_2$ before the differing query and the same $\mathsf{H}_1$ continuation afterward, including future requests. The queried functions differ only on unopened last groups, so the quantum query bound (Lemma 5) and the estimate above bound each change by $2\,2^{-c_0\lambda/2}$, including all quantum systems. There are polynomially many internal queries, and their common continuation cannot increase trace distance, so

$$\tfrac{1}{2}\left\|\mathsf{H}_0-\mathsf{H}_2\right\|_1\le 2^{-\Omega(\lambda)}+\mathsf{poly}(\lambda)2^{-c_0\lambda/2}=2^{-\Omega(\lambda)}. \tag{60}$$

This proves the joint-output bound (Equation (57)). The estimates are uniform over the fixed choices for fixed query bounds, and each party uses its known $K_t$ for local query substitutions, so the simulation adds no communication.

# Part IV
# Quantum Cryptography with P = QCMA

## 14 Reusable Keys for Coset Evaluation

We construct reusable quantum keys for coset evaluation and prove their correctness (Lemma 18), using separate key families for PRF and encryption.

For a subspace $A \leq \mathbb{F}_2^d$ and shifts $a, b \in \mathbb{F}_2^d$, write

$$|A_{a,b}\rangle = |A|^{-1/2} \sum_{z \in A} (-1)^{\langle b,z \rangle} |z + a\rangle . \tag{61}$$

For Boolean functions $f_0, f_1$, define the two-basis oracle

$$C_{A,a,b,f_0,f_1}(x,c,z) = \begin{cases} f_0(x), & c = 0,\ z \in A + a, \\ f_1(x), & c = 1,\ z \in A^\perp + b, \\ \perp, & \text{otherwise.} \end{cases} \tag{62}$$

Its Boolean encoding is $\mathfrak{F}_C(x,c,z,j) = \mathrm{enc}(C(x,c,z))_j$ for $j \in \{0,1\}$, where $\mathrm{enc}(\perp) = 00$, $\mathrm{enc}(0) = 10$ and $\mathrm{enc}(1) = 11$, so $\mathsf{val}(C(x,c,z)) = \mathfrak{F}_C(x,c,z,1)$.

Throughout the constructions, put

$$\kappa = (n+4)^3, \qquad \kappa_M = (5a_\kappa + 7n + 20)^3,$$

where $a_\kappa$ is the predicate-preimage state length, and use the fixed prefix predicates of Lemma 7(2). Each oracle address specifies $n$, a component listed in that oracle's definition, and an input to that function, and the oracle returns its value. These addresses have fixed efficient injective encodings, padded to a common strictly increasing polynomial length $L_0(n)$, with zero on unused inputs.

The following definition specifies the key and its public encoding for any fixed functions and cosets. The PRF and encryption sections will give their sampling.

**Definition 29** (Functional keys and their encoding)**.** *For $s \in \{0,1\}^n$, let $F_{s,0}, F_{s,1} : \{0,1\}^n \to \{0,1\}$, $A_s \leq \mathbb{F}_2^{4n}$ with $\dim A_s = 2n$, and $a_s, b_s \in \mathbb{F}_2^{4n}$. We define the functional keys by*

$$|K_s\rangle = |\Psi_{H_s}\rangle\, |\chi_s\rangle , \qquad |\chi_s\rangle = |(A_s)_{a_s,b_s}\rangle , \qquad F_s = F_{s,0} \oplus F_{s,1}, \quad C_s = C_{A_s,a_s,b_s,F_{s,0},F_{s,1}},$$

*where $H_s \in \mathcal{H}_{\kappa_M}$ and $B_s : \mathcal{X}_{\kappa_M} \to \{0,1\}$ is a Boolean function. Their public encodings and indexed preparation target are*

$$T_s = T_{H_s,B_s,\mathfrak{F}_{C_s}}, \tag{63}$$

$$|\Omega_K\rangle = 2^{-n/2} \sum_s |s\rangle\, |K_s\rangle , \qquad n_K = 5n + a_{\kappa_M}, \qquad (R_K, S_K) \leftarrow \mathcal{D}_{\mathrm{prep}}(|\Omega_K\rangle). \tag{64}$$

*Here $T_{H,B,\mathfrak{F}}$ is the encoded function (Definition 28) and $\mathcal{D}_{\mathrm{prep}}$ samples the preparation oracles (Definition 27).*

We use the preparation error bound $\epsilon_{\mathrm{prep}}(m) = 2^{-m}$ and the decoding failure bound $\delta_{\mathrm{dec}}(\kappa) = 2^{1-\kappa}$.

**Lemma 18** (Functional-key correctness)**.** *There are uniform* QPT *algorithms* $\mathsf{GenKey}, \mathsf{Eval}$ *with the following guarantees. Fix the functions and keys* $(F_s, T_s, |K_s\rangle)_s$ *above, and sample* $(R_K, S_K) \leftarrow \mathcal{D}_{\mathrm{prep}}(|\Omega_K\rangle)$*. Writing* $\rho_{\mathrm{key}}^{R_K,S_K}$ *for the output density matrix of* $\mathsf{GenKey}^{R_K,S_K}(1^n)$*, we have*

$$\mathbb{E}_{R_K,S_K} \tfrac{1}{2} \| \rho_{\mathrm{key}}^{R_K,S_K} - 2^{-n} \sum_s |s\rangle\langle s| \otimes |K_s\rangle\langle K_s| \,\|_1 \le \epsilon_{\mathrm{prep}}(n_K).$$

*For* $H_s \in \mathsf{Good}_{\kappa_M}$*, let* $\rho_q, \sigma_q$ *be the joint outputs from* $|K_s\rangle\langle K_s| \otimes \sigma_{XBW}$ *after* $q$ *calls to* $\mathsf{Eval}$ *or to* $|x, b\rangle \mapsto |x, b \oplus F_s(x)\rangle$*, respectively, with the same intervening operations on* $XBW$*. Then*

$$\tfrac{1}{2}\|\rho_q - \sigma_q\|_1 \le 4q\sqrt{\delta_{\mathrm{dec}}(\kappa_M)}.$$

The proof of Lemma 18 appears in Section 14.1. Observe that ideal evaluation leaves the key unchanged, and the preparation bound also holds with conditional side information: for every density matrix $\eta$, $\|\eta \otimes X\|_1 = \|X\|_1$.

## 14.1 Proof of functional-key correctness

We specify the algorithms, then prove functional-key correctness (Lemma 18).

**Definition 30** (Encoded coset evaluation)**.** *We define encoded coset evaluation* $\mathsf{EvalKey}^{T_s}$ *by the four steps of* $\mathsf{Eval}$ *below, using* $\mathsf{EvalHash}$ *and the predicate-preimage state register to implement each value-bit query through* $T_s$ *(Definition 28). Write* $\mathsf{EvalCoset}^{C_s}$ *for its direct* $C_s$ *implementation.*

**Functional-key generation and evaluation**

$\underline{\mathsf{GenKey}(1^n)}$

Run $\mathsf{Prepare}^{R_K,S_K}(1^{n_K})$, measure the serial register, and return its outcome $s$ with the remaining register $\mathsf{R_{sk}}$.

$\underline{\mathsf{Eval}(s, \mathsf{R_{sk}}, x, b)}$

Write $\mathsf{R_{sk}}$ as a predicate-preimage state register and a coset register, and implement each value-bit query below by $\mathsf{EvalHash}^{T_s}$. Using the coset register as the query register $z$:

1. XOR $\mathsf{val}(C_s(x, 0, z))$ into $b$.
2. Apply $\mathrm{H}^{\otimes 4n}$ to the coset register.
3. XOR $\mathsf{val}(C_s(x, 1, z))$ into $b$.
4. Apply $\mathrm{H}^{\otimes 4n}$ again and return all registers.

If $b$ is omitted, initialize it to zero and measure the answer.

**Key generation.** Let $\tau^{R_K,S_K}$ be the output of $\mathsf{Prepare}^{R_K,S_K}(1^{n_K})$ and let $\mathcal{M}$ measure its serial register, so the ideal output is

$$\mathcal{M}(|\Omega_K\rangle\langle\Omega_K|) = 2^{-n} \sum_s |s\rangle\langle s| \otimes |K_s\rangle\langle K_s| \,.$$

For these fixed keys, the target $|\Omega_K\rangle$ is fixed and $(R_K, S_K) \leftarrow \mathcal{D}_{\mathrm{prep}}(|\Omega_K\rangle)$, so the state-preparation bound (Lemma 17) gives error at most $\epsilon_{\mathrm{prep}}(n_K)$. Since measurement cannot increase trace distance, we obtain

$$\mathbb{E}_{R_K,S_K} \tfrac{1}{2} \left\| \mathcal{M}(\tau^{R_K,S_K}) - \mathcal{M}(|\Omega_K\rangle\langle\Omega_K|) \right\|_1 \le \mathbb{E}_{R_K,S_K} \tfrac{1}{2} \left\| \tau^{R_K,S_K} - |\Omega_K\rangle\langle\Omega_K| \right\|_1 \le \epsilon_{\mathrm{prep}}(n_K).$$

The same estimate holds with conditional side information because $\|\sigma_W \otimes X\|_1 = \|X\|_1$ for every conditional density matrix $\sigma_W$. Averaging these conditional distances also gives the distance when the oracle and all sampled functions and hashes are included as classical outputs.

**Function evaluation.** Character orthogonality gives

$$\mathrm{H}^{\otimes 4n} |\chi_s\rangle = (-1)^{\langle a_s, b_s \rangle} \left| (A_s^{\perp})_{b_s, a_s} \right\rangle . \tag{65}$$

Let $U_{s,c}$ XOR $\mathsf{val}(C_s(x, c, z))$ into the answer register and write $H = \mathrm{H}^{\otimes 4n}$ on the coset register. The first query returns $F_{s,0}(x)$ throughout the support of $|\chi_s\rangle$, and the second returns $F_{s,1}(x)$ after the Hadamard transform, so

$$HU_{s,1}HU_{s,0} |\chi_s\rangle |x\rangle |b\rangle = |\chi_s\rangle |x\rangle |b \oplus F_s(x)\rangle .$$

This proves exact correctness with direct queries to $C_s$. For the encoded implementation, apply the repeated-evaluation bound (Theorem 14) to its $2q$ calls to $\mathsf{EvalHash}^{T_s}$, with ideal function $\mathfrak{F}_{C_s}$. Between these calls, the Hadamard transforms and the prescribed channels on $XBW$ act outside the predicate-preimage state register, so the theorem applies to the full joint input and gives

$$\tfrac{1}{2} \|\rho_q - \sigma_q\|_1 \leq 2(2q)\sqrt{\delta_{\mathrm{dec}}(\kappa_M)}. \tag{66}$$

The direct coset identity identifies $\sigma_q$ with the claimed ideal evaluations of $F_s$, and the bound includes the returned key and the auxiliary system $W$.

# 15 Copy-Protected Pseudorandom Functions

We construct a copy-protected PRF from reusable functional keys (Theorem 15), with the following public oracle.

**Definition 31** (PRF oracle)**.** *Independently at every parameter $n$, and independently for each $s \in \{0,1\}^n$, sample*

$$F_{s,0}, F_{s,1} \leftarrow \{f : \{0,1\}^n \to \{0,1\}\}, \qquad A_s \leftarrow \{A \le \mathbb{F}_2^{4n} : \dim A = 2n\},$$
$$a_s, b_s \leftarrow \mathbb{F}_2^{4n}, \qquad H_s \leftarrow \mathcal{H}_{\kappa_M}, \qquad B_s \leftarrow \{0,1\}^{\mathcal{X}_{\kappa_M}}.$$

*For the resulting keys and encodings $|K_s\rangle = |\Psi_{H_s}\rangle\,|(A_s)_{a_s,b_s}\rangle$ and $T_s = T_{H_s,B_s,\mathfrak{F}_{C_s}}$ (Definition 29), sample*

$$(R_K, S_K) \leftarrow \mathcal{D}_{\mathrm{prep}}\left(2^{-n/2}\sum_s |s\rangle\,|K_s\rangle\right).$$

*We define the PRF base oracle by $\mathcal{O}_{\mathsf{prf},n} = ((T_s)_s, R_K, S_K)$, and let $\mathcal{O}_{\mathsf{prf}}(n,u) = \mathcal{O}_{\mathsf{prf},n}(u)$, where $u$ specifies a component above and its input.*

Use the full oracle $\mathcal{O} = (\mathcal{O}_{\mathsf{base}}, N, A)$ with $\mathcal{O}_{\mathsf{base}} = (\mathcal{O}_{\mathsf{prf}}, \mathcal{O}_{\mathrm{rest}})$, where $\mathcal{O}_{\mathrm{rest}} : \{0,1\}^* \to \{0,1\}$ is sampled independently on disjoint addresses, and $N, A$ are sampled once over the entire base (Definition 26). All oracle access is coherent. At $n = \lambda$, define

$$F_s^{\mathrm{prf}}(x) = F_s(x) = F_{s,0}(x) \oplus F_{s,1}(x), \qquad \left|K_s^{\mathrm{prf}}\right\rangle = |K_s\rangle\,. \tag{67}$$

The public serial $s$ is the PRF function index (Definition 9).

## 15.1 Construction

The PRF uses its functional key to evaluate $F_s$, with all calls below using the PRF component of the oracle.

$\underline{\mathsf{PRF.Gen}^{\mathcal{O}}(1^\lambda)}$
1. Return $(s, \mathsf{R_{sk}}) \leftarrow \mathsf{GenKey}(1^\lambda)$.

$\underline{\mathsf{PRF.Eval}^{\mathcal{O}}(s, \mathsf{R_{sk}}, x, b)}$
1. Run $\mathsf{Eval}(s, \mathsf{R_{sk}}, x, b)$ with the caller's response register.
2. Return all registers.

The following theorem gives correctness, pseudorandomness and copy-protection for this construction.

**Theorem 15** (PRF security)**.** *For the PRF oracle $\mathcal{O}$ above and every independent base component $\mathcal{O}_{\mathrm{rest}}$, the uniform $\mathsf{QPT}$ construction $\mathsf{PRF}^{\mathcal{O}}$ satisfies the following guarantees, averaged over $\mathcal{O}$:*

- Correctness. *Polynomially many coherent evaluations have negligible joint error, returning the key for reuse.*
- Security. *The pseudorandomness and copy-protection guarantees in Equations (13) and (86) hold with error $2^{-\Omega(\lambda)}$ against every adversary with polynomial query count and length and unrestricted local computation.*

The proof of Theorem 15 appears in Section 15.2.

## 15.2 Proof of PRF security

We now prove correctness, pseudorandomness and copy-protection in Theorem 15.

**Correctness.** For $q$ coherent evaluations, compare the actual output with the ideal experiment

$$s \leftarrow \{0,1\}^{\lambda}, \qquad \mathsf{R_{sk}} \leftarrow |K_s\rangle\,, \qquad |x,b\rangle \longmapsto |x, b \oplus F_s(x)\rangle\,,$$

which leaves $\mathsf{R_{sk}}$ unchanged after each query. Functional-key correctness (Lemma 18) applies to arbitrary intervening operations outside the key, so its joint output, including the key, the sampled oracle and any entangled auxiliary system, differs from the actual output by at most

$$\epsilon_{\mathrm{prep}}(n_K) + 2^{-\kappa_M} + 4q\sqrt{\delta_{\mathrm{dec}}(\kappa_M)} = \mathsf{negl}(\lambda). \tag{68}$$

Here $2^{-\kappa_M}$ bounds the probability that $H_s$ is not good.

**Pseudorandomness.** Write $n = \lambda$ and fix the privacy adversary $\mathcal{A}$. All bounds below are uniform in the independently sampled additional base $\mathcal{O}_{\mathrm{rest}}$, which is included among the fixed choices in both security arguments. In both branches $b \in \{0,1\}$, sample an independent uniform function $R : \{0,1\}^n \to \{0,1\}$, and define $Q_0 = F_s^{\mathrm{prf}}$ and $Q_1 = R$. In experiment $\mathsf{P}_{i,b}$, write $\mathcal{O}^i = (\mathcal{O}^i_{\mathsf{base}}, N^i, A^i)$ for the public oracle supplied to $\mathcal{A}$. Every experiment returns the bit output by $\mathcal{A}(s)$, with coherent challenge access to $Q_b$:

$\mathsf{P}_{0,b}$ Sample $\mathcal{O}^0 = \mathcal{O}$ and $(s, \mathsf{R_{sk}}) \leftarrow \mathsf{PRF.Gen}^{\mathcal{O}^0}(1^n)$, discard $\mathsf{R_{sk}}$, and output $\mathcal{A}^{\mathcal{O}^0, Q_b}(s)$.

$\mathsf{P}_{1,b}$ Sample $\mathcal{O}^1 = \mathcal{O}$ and $s \leftarrow \{0,1\}^n$ independently, and output $\mathcal{A}^{\mathcal{O}^1, Q_b}(s)$.

$\mathsf{P}_{2,b}$ Sample all functions and hashes as above, but replace the key-preparation target by

$$\left|\Omega^0_K\right\rangle = 2^{-n/2}\left(\sum_{t \neq s} |t\rangle\,|K_t\rangle + |s\rangle\left|0^{n_K - n}\right\rangle\right).$$

Sample $(R^0_K, S^0_K) \leftarrow \mathcal{D}_{\mathrm{prep}}(\left|\Omega^0_K\right\rangle)$ and set

$$\mathcal{O}^2_{\mathsf{base}} = \mathcal{O}_{\mathsf{base}}[(R_K, S_K) \leftarrow (R^0_K, S^0_K)], \qquad \mathcal{O}^2 = (\mathcal{O}^2_{\mathsf{base}}, N^2, A^2),$$

where $N^2, A^2$ have the recursive distribution conditional on $\mathcal{O}^2_{\mathsf{base}}$. Output $\mathcal{A}^{\mathcal{O}^2, Q_b}(s)$.

$\mathsf{P}_{3,b}$ To construct one simulator for both branches, sample an independent analysis bit $\beta \leftarrow \{0,1\}$. Let $\mathcal{O}^{2,-s}_{\mathsf{base}}$ denote $\mathcal{O}^2_{\mathsf{base}}$ with $T_s$ omitted, and put

$$\Theta_{\mathrm{priv}} = (F_{s,0}, F_{s,1}, R, \beta),$$
$$G_{\mathrm{priv}} = (A_s, a_s, b_s, \mathcal{O}^{2,-s}_{\mathsf{base}}).$$

Apply encoded-function simulation (Theorem 14) once to the protocol with challenge oracle $Q_\beta$, using $J = \{s\}$ and $K = \varnothing$. For each fixed $G_{\mathrm{priv}}$ this supplies $\mathcal{S}_{G_{\mathrm{priv}}}$ and one classical sample $\eta_{\mathrm{priv}}$ independent of $\Theta_{\mathrm{priv}}$. In $\mathsf{P}_{3,b}$, condition on $\beta = b$ and output $\mathcal{S}^{Q_b}_{G_{\mathrm{priv}}}(s, \eta_{\mathrm{priv}})$. It runs $\mathcal{A}$, forwards challenge queries to $Q_b$, and computes every $\mathcal{O}^2$ answer by the coherent algorithm $\mathcal{S}_{\eta_{\mathrm{priv}}}(\varnothing, \cdot)$ with $T_s$ replaced by the original $B_s$. These answers use $\eta_{\mathrm{priv}}$ and no challenge-function query.

Only the adversary's challenge queries are forwarded to $Q_b$ in the last experiment. Write $p_{i,b} = \Pr[\mathsf{P}_{i,b} = 1]$.

Functional-key correctness (Lemma 18) allows us to replace the generated serial by a uniform one, giving $|p_{0,b} - p_{1,b}| \le \epsilon_{\mathrm{prep}}(n_K)$. The following claim removes the remaining dependence of the public oracle on the challenge function.

**Claim 2.** *For $b \in \{0,1\}$, the privacy experiments satisfy*

$$|p_{1,b} - p_{2,b}| + |p_{2,b} - p_{3,b}| \le 2^{-\Omega(n)}, \qquad p_{3,0} = p_{3,1}.$$

The proof of Claim 2 appears in Section 15.2.1.

The two final experiments have equal acceptance probability by the claim, so the triangle inequality gives

$$|p_{0,0} - p_{0,1}| \le \sum_{b=0}^{1} \sum_{i=0}^{2} |p_{i,b} - p_{i+1,b}| \le 2\epsilon_{\mathrm{prep}}(n_K) + 2^{-\Omega(n)} = 2^{-\Omega(n)}.$$

**Copy-protection.** Fix $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$. In $\mathsf{V}_0$, sample $\mathcal{O}^0 = \mathcal{O}$ and $(s, \mathsf{R}_{\mathsf{sk}}) \leftarrow \mathsf{PRF.Gen}^{\mathcal{O}^0}(1^n)$, then run

$$(\mathsf{R}_1, \mathsf{R}_2) \leftarrow \mathcal{A}_0^{\mathcal{O}^0}(s, \mathsf{R}_{\mathsf{sk}}), \qquad u_1, u_2 \leftarrow \{0,1\}^n, \qquad \widehat{f}_i \leftarrow \mathcal{A}_i^{\mathcal{O}^0}(u_i, \mathsf{R}_i).$$

The algorithms $\mathcal{A}_1, \mathcal{A}_2$ act on their possibly entangled registers without communication. In $\mathsf{V}_1$, sample $\mathcal{O}^1 = \mathcal{O}$ and uniform $s$ independently, substitute $\mathsf{R}_{\mathsf{sk}} = |K_s\rangle$, and make the same calls with $\mathcal{O}^1$. Both experiments return

$$\mathbf{1}[\widehat{f}_1 = F_s(u_1) \ \wedge \ \widehat{f}_2 = F_s(u_2)].$$

Key-generation correctness gives $|\Pr[\mathsf{V}_0 = 1] - \Pr[\mathsf{V}_1 = 1]| \le \epsilon_{\mathrm{prep}}(n_K)$.

We define the next experiments using an equivalent sampling for $\mathsf{V}_1$. Privately sample the inputs before invoking $\mathcal{A}_0$, without giving them to any party, and sample

$$u_1, u_2 \leftarrow \{0,1\}^n, \quad G_0, G_1 \leftarrow \{\{0,1\}^n \to \{0,1\}\},$$
$$b_{x,c} \leftarrow \{0,1\} \quad (x \in \{u_1, u_2\},\ c \in \{0,1\}), \qquad F_{s,c} = G_c[x \leftarrow b_{x,c} : x \in \{u_1, u_2\}].$$

There is one pair of bits per distinct input, so $F_{s,0}, F_{s,1}$ are independent uniform functions even when $u_1 = u_2$. Sample every other component of $\mathcal{O}$ with these functions. The next two experiments use this sampling, the original issued key and the same winning predicate:

$\mathsf{V}_2$ Sample $(R_K^0, S_K^0) \leftarrow \mathcal{D}_{\mathrm{prep}}(|\Omega_K^0\rangle)$, set $\mathcal{O}^2_{\mathsf{base}} = \mathcal{O}_{\mathsf{base}}[(R_K, S_K) \leftarrow (R_K^0, S_K^0)]$, and sample $(N^2, A^2)$ conditional on $\mathcal{O}^2_{\mathsf{base}}$. Run

$$(\mathsf{R}_1, \mathsf{R}_2) \leftarrow \mathcal{A}_0^{(\mathcal{O}^2_{\mathsf{base}}, N^2, A^2)}(s, |K_s\rangle), \qquad \widehat{f}_i \leftarrow \mathcal{A}_i^{(\mathcal{O}^2_{\mathsf{base}}, N^2, A^2)}(u_i, \mathsf{R}_i).$$

$\mathsf{V}_3$ Let $\mathcal{O}^{2,-s}_{\mathsf{base}}$ again omit $T_s$, and put

$$\Theta_{\mathrm{pir}} = (A_s, a_s, b_s, u_1, u_2, (b_{x,c})_{x,c}), \qquad G_{\mathrm{pir}} = (G_0, G_1, \mathcal{O}^{2,-s}_{\mathsf{base}}).$$

Apply encoded-function simulation (Theorem 14) to $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ for each fixed $G_{\mathrm{pir}}$, with $J = K = \{s\}$ and an initial request for $s$, obtaining local algorithms $(\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_2)$. Draw their one classical sample $\eta_{\mathrm{pir}}$ and run

$$(\mathsf{R}'_1, \mathsf{R}'_2) \leftarrow \mathcal{S}_0^{\mathfrak{F}_{C_s}}(s, |\chi_s\rangle, \eta_{\mathrm{pir}}), \qquad \widehat{f}_i \leftarrow \mathcal{S}_i^{\mathfrak{F}_{C_s}}(u_i, \mathsf{R}'_i).$$

Here $\mathsf{R}'_i$ includes the quantum state and classical information assigned to algorithm $i$. Each $\mathcal{S}_i$ answers every $(\mathcal{O}^2_{\mathsf{base}}, N^2, A^2)$ query by $\mathcal{S}_{\eta_{\mathrm{pir}}}(\{s\}, \cdot)$, using direct queries to $\mathfrak{F}_{C_s}$. The algorithms $\mathcal{S}_1, \mathcal{S}_2$ do not communicate.

We use the following bound to reduce the original adversary to one with direct coset-oracle access.

**Claim 3.** *The piracy experiments satisfy*

$$|\Pr[\mathsf{V}_1 = 1] - \Pr[\mathsf{V}_2 = 1]| + |\Pr[\mathsf{V}_2 = 1] - \Pr[\mathsf{V}_3 = 1]| \le 2^{-\Omega(n)}.$$

*Moreover, $\mathsf{V}_3$ is a direct coset function-value experiment (Definition 41) with $Q' = \mathsf{poly}(n)$ queries.*

The proof of Claim 3 appears in Section 15.2.1.

Applying the coset security bound (Lemma 28) and summing the three differences gives

$$\Pr[\mathsf{V}_0 = 1] \le \tfrac{1}{2} + O((Q'+1)2^{-n/2}) + \epsilon_{\mathrm{prep}}(n_K) + 2^{-\Omega(n)} = \tfrac{1}{2} + 2^{-\Omega(n)}.$$

### 15.2.1 Bounds for the PRF hybrids

We now prove Claim 2 and Claim 3. The privacy simulation opens no hash, whereas the piracy simulation opens the issued hash before the adversary separates its registers.

**Changing the key-preparation target.** For both $\mathsf{P}_{1,b} \to \mathsf{P}_{2,b}$ and $\mathsf{V}_1 \to \mathsf{V}_2$, fix $s$, the functions $F_{t,c}$, the hashes $H_t$, the coset descriptions $(A_t, a_t, b_t)$, and all base components except $(R_K, S_K)$. Also fix $R$ in the privacy experiment and $u_1, u_2$ in the piracy experiment. This determines the targets, the supplied inputs and the challenge operations, including $Q_b$ in the privacy experiment and the original $|K_s\rangle, F_s, u_i$ in the piracy experiment. Since

$$\| \, |\Omega_K\rangle - \left|\Omega^0_K\right\rangle \, \|_2 \le 2^{1-n/2},$$

the target-change bound (Theorem 13), with dimension $n_K$ and recursive mean $h(v) = (1+v)/3$, applies to each protocol. Averaging the fixed quantities gives

$$|p_{1,b} - p_{2,b}| \le 2^{-\Omega(n)}, \qquad |\Pr[\mathsf{V}_1 = 1] - \Pr[\mathsf{V}_2 = 1]| \le 2^{-\Omega(n)}.$$

**Privacy.** It remains to simulate $T_s$, the only base component still using the challenge functions. Conditional on $G_{\mathrm{priv}}$, the variables $\Theta_{\mathrm{priv}}, H_s, B_s$ are mutually independent: the new target substitutes $|0^{n_K - n}\rangle$ for $|K_s\rangle$, and all other primitive components were sampled independently. The single instance being simulated is $T_s = T_{H_s, B_s, \mathfrak{F}_{C_s}}$, and no predicate-preimage state is supplied. The challenge oracle depends on $\Theta_{\mathrm{priv}}$ but not $H_s$. Apply encoded-function simulation (Theorem 14) to the joint protocol, including $\beta$ in its classical output. The averaged joint-output bound implies, for either conditional branch,

$$|p_{2,b} - p_{3,b}| \le 2 \cdot 2^{-\Omega(n)} = 2^{-\Omega(n)}.$$

The one simulator sample and description are independent of $(F_{s,0}, F_{s,1}, R, \beta)$ conditional on $G_{\mathrm{priv}}$. Since the XOR of the two independent functions is uniform, $\mathsf{P}_{3,0}$ and $\mathsf{P}_{3,1}$ have the same distribution, completing Claim 2.

**Copy-protection.** The target $\left|\Omega_K^0\right\rangle$ substitutes $|0^{n_K-n}\rangle$ for $|K_s\rangle$, and every other primitive base is independent, so $\Theta_{\mathrm{pir}}, H_s, B_s$ are mutually independent conditional on $G_{\mathrm{pir}}$. Moreover, $(G_{\mathrm{pir}}, \Theta_{\mathrm{pir}})$ determines $\mathfrak{F}_{C_s}$, while the supplied coset state depends only on $\Theta_{\mathrm{pir}}$. The serial $s$ is opened before separation and is known to all three algorithms, so encoded-function simulation with $J = K = \{s\}$ gives

$$|\Pr[\mathsf{V}_2 = 1] - \Pr[\mathsf{V}_3 = 1]| \leq 2^{-\Omega(n)}.$$

Its local simulations make $Q' = \mathsf{poly}(n)$ direct queries and use no original $N, A$ oracle. After averaging $\mathcal{O}_{\mathsf{base}}^{2,-s}$, their classical sample and descriptions are independent of $\Theta_{\mathrm{pir}}$ conditional on $(G_0, G_1)$. Hence $\mathsf{V}_3$ is the direct coset function-value experiment (Definition 41), proving Claim 3.

# 16 Copy-Protected Public-Key Encryption

We construct copy-protected encryption with classical public keys and ciphertexts (Theorem 16). Its public oracle prepares quantum decryption keys and graph states for encryption.

**Definition 32** (Encryption oracle)**.** *Independently for every parameter $n$ and serial $s \in \{0,1\}^n$, sample*

$$F_{s,0}, F_{s,1} \leftarrow \{f : \{0,1\}^n \to \{0,1\}\}, \qquad A_s \leftarrow \{A \le \mathbb{F}_2^{4n} : \dim A = 2n\},$$
$$a_s, b_s \leftarrow \mathbb{F}_2^{4n}, \qquad H_s \leftarrow \mathcal{H}_{\kappa_M}, \qquad B_s \leftarrow \{0,1\}^{\mathcal{X}_{\kappa_M}}.$$

*Set $F_s = F_{s,0} \oplus F_{s,1}$ and $C_s = C_{A_s,a_s,b_s,F_{s,0},F_{s,1}}$, with functional keys and encodings*

$$|K_s\rangle = |\Psi_{H_s}\rangle\,|(A_s)_{a_s,b_s}\rangle\,, \qquad T_s = T_{H_s,B_s,\mathfrak{F}_{C_s}}.$$

*Conditional on these variables, independently sample one indexed-key preparation pair and one graph-preparation pair for each $s$:*

$$(R_K, S_K) \leftarrow \mathcal{D}_{\mathrm{prep}}\left(2^{-n/2}\sum_s |s\rangle\,|K_s\rangle\right),$$

$$|\Gamma_s\rangle = 2^{-n/2}\sum_{u\in\{0,1\}^n} |u\rangle\,|F_s(u)\rangle\,\left|0^{4n-1}\right\rangle, \tag{69}$$

$$(R_s^{\mathrm{graph}}, S_s^{\mathrm{graph}}) \leftarrow \mathcal{D}_{\mathrm{prep}}(|\Gamma_s\rangle). \tag{70}$$

*We define the encryption base oracle by*

$$\mathcal{O}_{\mathsf{pke},n} = ((T_s)_s, R_K, S_K, (R_s^{\mathrm{graph}}, S_s^{\mathrm{graph}})_s),$$

*and let $\mathcal{O}_{\mathsf{pke}}(n,u) = \mathcal{O}_{\mathsf{pke},n}(u)$, where $u$ specifies a component above and its input.*

Use the full oracle $\mathcal{O} = (\mathcal{O}_{\mathsf{base}}, N, A)$ with $\mathcal{O}_{\mathsf{base}} = (\mathcal{O}_{\mathsf{pke}}, \mathcal{O}_{\mathrm{rest}})$, where $\mathcal{O}_{\mathrm{rest}} : \{0,1\}^* \to \{0,1\}$ is sampled independently on disjoint addresses, and $N, A$ are sampled once over the entire base (Definition 26). The public key is $s$, and measuring $|\Gamma_s\rangle$ produces a uniform input and its pad without using the secret key $|K_s\rangle$.

## 16.1 Construction

Encryption samples an input and its pad from the graph state, while decryption uses the functional key to recover that pad. All calls below use the encryption component of the oracle.

$\underline{\mathsf{PKE.Gen}^{\mathcal{O}}(1^\lambda)}$

1. Run $(s, \mathsf{R_{sk}}) \leftarrow \mathsf{GenKey}(1^\lambda)$.
2. Return the public key $\mathrm{pk} = s$ and secret key $(s, \mathsf{R_{sk}})$.

$\underline{\mathsf{PKE.Enc}^{\mathcal{O}}(s,m)}$

1. Run $\mathsf{Prepare}^{R_s^{\mathrm{graph}}, S_s^{\mathrm{graph}}}(1^{5\lambda})$.
2. Measure the first $\lambda+1$ qubits to obtain $(u,b)$ and discard the remaining register.
3. Return $(u, m \oplus b)$.

$\mathsf{PKE.Dec}^{\mathcal{O}}((s, \mathsf{R_{sk}}), (u, c))$

1. Compute $b \leftarrow \mathsf{Eval}(s, \mathsf{R_{sk}}, u)$.
2. Return $c \oplus b$ together with the updated key.

We now state correctness and security for this encryption scheme.

**Theorem 16** (Encryption security)**.** *For the encryption oracle $\mathcal{O}$ above and every independent base component $\mathcal{O}_{\mathrm{rest}}$, the uniform $\mathsf{QPT}$ construction $\mathsf{PKE}^{\mathcal{O}}$ has classical public keys and ciphertexts and satisfies the following guarantees, averaged over $\mathcal{O}$:*

- Correctness. *Polynomially many encryptions and decryptions using one key have negligible joint error.*
- Security. *Against every adversary with polynomial query count and length and unrestricted local computation, the scheme is chosen-plaintext secure and satisfies, for the independent ciphertext challenges of Definition 40,*

$$\Pr_{\mathcal{O}}[\mathsf{Piracy}^{\mathcal{O}}_{\mathcal{A}}(\lambda) = 1] \le \tfrac{1}{2} + 2^{-\Omega(\lambda)}.$$

The proof of Theorem 16 appears in Section 16.2.

## 16.2 Proof of encryption security

We now prove Theorem 16 by reducing the encryption experiment to the direct coset ciphertext experiment.

**Correctness.** The ideal generation, encryption and decryption operations are

$$\begin{aligned} &s \leftarrow \{0,1\}^{\lambda}, \qquad && \mathsf{R_{sk}} \leftarrow |K_s\rangle\,, \\ &u \leftarrow \{0,1\}^{\lambda}, \qquad && \mathrm{ct} = (u, m \oplus F_s(u)), \\ &\widehat{m} = \mathrm{ct}_2 \oplus F_s(\mathrm{ct}_1) = m. \end{aligned}$$

Functional-key correctness (Lemma 18) bounds the key error. For the graph preparations, the state-preparation bound (Lemma 17) gives

$$\mathbb{E} \max_{s} \tfrac{1}{2} \left\| \mathsf{Prepare}^{R_s^{\mathrm{graph}}, S_s^{\mathrm{graph}}}(1^{5\lambda}) - |\Gamma_s\rangle\langle\Gamma_s| \right\|_1 \le 2^{\lambda} \epsilon_{\mathrm{prep}}(5\lambda) = 2^{-4\lambda}.$$

Applying these bounds sequentially shows that the joint output of $r$ encryptions and decryptions differs from the ideal output by at most

$$\epsilon_{\mathrm{prep}}(n_K) + r 2^{-4\lambda} + 2^{-\kappa_M} + 4r\sqrt{\delta_{\mathrm{dec}}(\kappa_M)} = \mathsf{negl}(\lambda). \tag{71}$$

This includes the key, the sampled oracle and any entangled auxiliary system, while $2^{-\kappa_M}$ bounds $\Pr[H_s \notin \mathsf{Good}_{\kappa_M}]$.

**The piracy experiments.** Fix $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ and write $n = \lambda$. The estimates are uniform in the independently sampled additional base $\mathcal{O}_{\mathrm{rest}}$, which is included in each $G$ below together with the other sampled base components. In $\mathsf{H}_0$, sample $\mathcal{O}^0 = \mathcal{O}$ and run

$$\begin{aligned}(s, (s, \mathsf{R}_{\mathsf{sk}})) &\leftarrow \mathsf{PKE.Gen}^{\mathcal{O}^0}(1^n),\\ ((m_{i,0}, m_{i,1})_{i=1}^2, \mathsf{R}_1, \mathsf{R}_2) &\leftarrow \mathcal{A}_0^{\mathcal{O}^0}(s, \mathsf{R}_{\mathsf{sk}}),\\ \beta_1, \beta_2 &\leftarrow \{0,1\}, \qquad \mathrm{ct}_i \leftarrow \mathsf{PKE.Enc}^{\mathcal{O}^0}(s, m_{i,\beta_i}),\\ \widehat{\beta}_i &\leftarrow \mathcal{A}_i^{\mathcal{O}^0}(\mathrm{ct}_i, \mathsf{R}_i).\end{aligned}$$

The two challenge bits and encryption coins are independent, and $\mathcal{A}_1, \mathcal{A}_2$ do not communicate. Return $\mathbf{1}[\widehat{\beta}_1 = \beta_1 \wedge \widehat{\beta}_2 = \beta_2]$. To define $\mathsf{H}_1$, sample $s \leftarrow \{0,1\}^n$ and privately sample

$$\begin{aligned}&(u_1, u_2, \beta_1, \beta_2) \leftarrow (\{0,1\}^n)^2 \times \{0,1\}^2,\\ &G_0, G_1 \leftarrow \{\{0,1\}^n \to \{0,1\}\}, \qquad b_{x,c} \leftarrow \{0,1\} \quad (x \in \{u_1, u_2\},\ c \in \{0,1\}),\\ &F_{s,c} = G_c[x \leftarrow b_{x,c} : x \in \{u_1, u_2\}], \qquad F_s = F_{s,0} \oplus F_{s,1}.\end{aligned}$$

Use these functions to sample $\mathcal{O}^1 = (\mathcal{O}^1_{\mathsf{base}}, N^1, A^1)$ and run

$$\begin{aligned}((m_{i,0}, m_{i,1})_{i=1}^2, \mathsf{R}_1, \mathsf{R}_2) &\leftarrow \mathcal{A}_0^{\mathcal{O}^1}(s, |K_s\rangle),\\ \mathrm{ct}_i &= (u_i, m_{i,\beta_i} \oplus F_s(u_i)),\\ \widehat{\beta}_i &\leftarrow \mathcal{A}_i^{\mathcal{O}^1}(\mathrm{ct}_i, \mathsf{R}_i).\end{aligned}$$

Return $\mathbf{1}[\widehat{\beta}_1 = \beta_1 \wedge \widehat{\beta}_2 = \beta_2]$. The algorithms $\mathcal{A}_1, \mathcal{A}_2$ run without communication and receive $\mathrm{ct}_i$ only after $\mathcal{A}_0$ has output its registers and message pairs. Because there is one fresh bit pair per distinct $u_i$, this is the ideal-key, ideal-encryption game even when $u_1 = u_2$.

Define the next three experiments by changing only the public oracle in this experiment. In each, the challenger uses the original $|K_s\rangle, F_s, u_i, \beta_i$ and the displayed winning predicate.

$\mathsf{H}_2$ Replace the key target by

$$\left|\Omega_K^0\right\rangle = 2^{-n/2}\left(\sum_{t \neq s} |t\rangle\,|K_t\rangle + |s\rangle\left|0^{n_K - n}\right\rangle\right),$$

sample $(R_K^0, S_K^0) \leftarrow \mathcal{D}_{\mathrm{prep}}(\left|\Omega_K^0\right\rangle)$, set $\mathcal{O}^2_{\mathsf{base}} = \mathcal{O}^1_{\mathsf{base}}[(R_K, S_K) \leftarrow (R_K^0, S_K^0)]$, and sample $N^2, A^2$ conditional on $\mathcal{O}^2_{\mathsf{base}}$. Run the displayed $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2$ calls with $\mathcal{O}^2 = (\mathcal{O}^2_{\mathsf{base}}, N^2, A^2)$ instead of $\mathcal{O}^1$.

$\mathsf{H}_3$ In addition, replace only the $s$ graph target by

$$\left|\Gamma_s^0\right\rangle = 2^{-n/2}\sum_u |u\rangle\,|G_0(u) \oplus G_1(u)\rangle\left|0^{4n-1}\right\rangle.$$

Sample $(R_s^0, S_s^0) \leftarrow \mathcal{D}_{\mathrm{prep}}(\left|\Gamma_s^0\right\rangle)$, set

$$\mathcal{O}^3_{\mathsf{base}} = \mathcal{O}^2_{\mathsf{base}}[(R_s^{\mathrm{graph}}, S_s^{\mathrm{graph}}) \leftarrow (R_s^0, S_s^0)],$$

and sample $N^3, A^3$ conditional on $\mathcal{O}^3_{\mathsf{base}}$. Run the same calls with $\mathcal{O}^3 = (\mathcal{O}^3_{\mathsf{base}}, N^3, A^3)$.

$\mathsf{H}_4$ Let $\mathcal{O}^{3,-s}_{\mathsf{base}}$ denote $\mathcal{O}^{3}_{\mathsf{base}}$ with $T_s$ omitted, and set

$$\Theta = (A_s, a_s, b_s, u_1, u_2, (b_{x,c})_{x,c}, \beta_1, \beta_2),$$
$$G = (G_0, G_1, \mathcal{O}^{3,-s}_{\mathsf{base}}).$$

Apply encoded-function simulation (Theorem 14) to $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ for each fixed $G$, with $J = K = \{s\}$ and an initial request for $s$, obtaining $(\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_2)$. Draw their one classical sample $\eta_G$ and run

$$((m_{i,0}, m_{i,1})_{i=1}^{2}, \mathsf{R}'_1, \mathsf{R}'_2) \leftarrow \mathcal{S}_0^{\mathfrak{F}_{C_s}}(s, |\chi_s\rangle, \eta_G),$$
$$\mathrm{ct}_i = (u_i, m_{i,\beta_i} \oplus F_s(u_i)),$$
$$\widehat{\beta}_i \leftarrow \mathcal{S}_i^{\mathfrak{F}_{C_s}}(\mathrm{ct}_i, \mathsf{R}'_i).$$

Each $\mathsf{R}'_i$ includes that algorithm's quantum state and classical information. The algorithms $\mathcal{S}_1, \mathcal{S}_2$ do not communicate.

Write $p_i = \Pr[\mathsf{H}_i = 1]$. Generation and two graph preparations, with the same full oracle supplied to the adversary, give

$$|p_0 - p_1| \le \epsilon_{\mathrm{prep}}(n_K) + 2 \cdot 2^{-4n}$$

by functional-key correctness and the graph-preparation bound above.

The following claim bounds the three changes to the public oracle.

**Claim 4.** *For the experiments just defined,*

$$|p_1 - p_2| + |p_2 - p_3| + |p_3 - p_4| \le 2^{-\Omega(n)}.$$

*In $\mathsf{H}_4$, the algorithms make $Q' = \mathsf{poly}(n)$ queries to $\mathfrak{F}_{C_s}$, and their descriptions and classical sample are independent of $\Theta$ conditional on $(G_0, G_1)$.*

The proof of Claim 4 appears in Section 16.2.1.

We can now apply the direct coset bound. The final algorithms have the coset state $|\chi_s\rangle$, the original ciphertext rule and independent challenge bits $\beta_1, \beta_2$, so Claim 4 makes $\mathsf{H}_4$ a direct coset ciphertext experiment (Definition 41). Its security bound (Lemma 28) gives

$$p_4 \le \tfrac{1}{2} + O((Q'+1)2^{-n/2}).$$

Summing the four adjacent differences proves copy-protection:

$$p_0 \le p_4 + \sum_{i=0}^{3} |p_i - p_{i+1}| \le \tfrac{1}{2} + O((Q'+1)2^{-n/2}) + \epsilon_{\mathrm{prep}}(n_K) + 2^{1-4n} + 2^{-\Omega(n)} = \tfrac{1}{2} + 2^{-\Omega(n)}.$$

**Chosen-plaintext security.** Given a CPA adversary $\mathcal{B} = (\mathcal{B}_0, \mathcal{B}_1)$, define the following three algorithms, all with oracle $\mathcal{O}$:

$$\begin{aligned}
\mathcal{A}_0(s, \mathsf{R}_{\mathsf{sk}}): &\quad (m_0, m_1, \mathsf{R}) \leftarrow \mathcal{B}_0(s),\\
&\quad \text{return } ((m_0, m_1), (0,1), \mathsf{R}, (s, \mathsf{R}_{\mathsf{sk}})),\\
\mathcal{A}_1(\mathrm{ct}_1, \mathsf{R}): &\quad \text{return } \mathcal{B}_1(\mathrm{ct}_1, \mathsf{R}),\\
\mathcal{A}_2(\mathrm{ct}_2, (s, \mathsf{R}_{\mathsf{sk}})): &\quad \text{return the plaintext from } \mathsf{PKE.Dec}((s, \mathsf{R}_{\mathsf{sk}}), \mathrm{ct}_2).
\end{aligned}$$

Algorithm $\mathcal{A}_1$ has exactly the CPA input distribution, while $\mathcal{A}_2$ decrypts an independent encryption of $\beta_2$ and fails with probability at most $\mathsf{negl}(n)$ by correctness. Hence

$$\Pr\left[\mathsf{CPA}^{\mathcal{O}}_{\mathcal{B}}(n) = 1\right] \le \Pr[\mathsf{H}_0 = 1] + \mathsf{negl}(n) \le \tfrac{1}{2} + \mathsf{negl}(n).$$

Applying this construction also to the complement of $\mathcal{B}_1$'s output bounds the absolute CPA advantage.

### 16.2.1 Bounds for the encryption hybrids

We now prove Claim 4, keeping the issued key, ciphertext maps and winning predicate fixed while changing the public oracle.

**Changing the preparation targets.** The two changes affect at most one key branch and two graph values, so

$$\| \, |\Omega_K\rangle - \left|\Omega_K^0\right\rangle \, \|_2 \le 2^{1-n/2}, \qquad \| \, |\Gamma_s\rangle - \left|\Gamma_s^0\right\rangle \, \|_2 \le 2\sqrt{2}\, 2^{-n/2}.$$

For the first change, fix $s, u_1, u_2, \beta_1, \beta_2$, all original functions and hashes, and every preparation pair except $(R_K, S_K)$. These choices determine both key targets, the issued density matrix $|K_s\rangle\langle K_s|$ and the maps producing $\mathrm{ct}_i$ from the message pairs output by $\mathcal{A}_0$. Apply the target-change bound (Theorem 13) to these targets and the original three algorithms, with dimension $n_K$ and recursive mean $h(v) = (1+v)/3$, to obtain

$$|p_1 - p_2| \le 2^{-\Omega(n)}.$$

For the graph change, fix the first changed pair $(R_K^0, S_K^0)$ as well, but sample the $s$ graph pair anew. The same theorem now applies in dimension $5n$ and gives

$$|p_2 - p_3| \le 2^{-\Omega(n)}.$$

Both estimates average over the fixed choices and use the original ciphertext rule, including independent $\beta_1, \beta_2$.

**Simulating the encoded functions.** Since $\left|\Omega_K^0\right\rangle$ replaces $|K_s\rangle$ by $|0^{n_K - n}\rangle$ and $\left|\Gamma_s^0\right\rangle$ uses only $G_0, G_1$, while all other primitive bases are independent, the variables $\Theta, H_s, B_s$ are mutually independent conditional on $G$. Also, $(G, \Theta)$ determines $\mathfrak{F}_{C_s}$, and $|\chi_s\rangle$ depends only on $\Theta$. The index $s$ is opened before the algorithms separate and remains known to each of them. These are the hypotheses of encoded-function simulation (Theorem 14) with $J = K = \{s\}$.

In $\mathsf{H}_4$, each $\mathcal{S}_i$ substitutes the coherent algorithm $\mathcal{S}_{\eta_G}(\{s\}, \cdot)$ for every $\mathcal{O}^3$ query of $\mathcal{A}_i$, using the one sample $\eta_G$ and direct queries to the Boolean encoding $\mathfrak{F}_{C_s}$ of $C_s = C_{A_s, a_s, b_s, F_{s,0}, F_{s,1}}$. All other answers and the master predicate-preimage state are computed from $\eta_G$. Thus $\mathcal{S}_i$ uses its assigned register $\mathsf{R}'_i$ and these same sampled functions, with no original $N, A$ access. The theorem applies to the original ciphertext maps and winning predicate, and therefore

$$|p_3 - p_4| \le 2^{-\Omega(n)}.$$

After averaging $\mathcal{O}^{3,-s}_{\mathsf{base}}$, the simulator's classical sample and descriptions are independent of $\Theta$ conditional on $(G_0, G_1)$, and its number of direct queries is $Q' = \mathsf{poly}(n)$. Together with the three displayed error bounds, this proves the claim.

# 17 Signatures with Quantum Keys

We construct two adaptively secure signature schemes with quantum signing keys (Theorem 17): $\mathsf{CSig}$ has classical verification keys and quantum signatures, whereas $\mathsf{QSig}$ has quantum verification keys and classical signatures.

## 17.1 Constructions

Both schemes use one signature-specific family, sampled independently of the PRF and encryption components. Put

$$n = \lambda, \qquad m \in \{0,1\}^n, \tag{72}$$

and use

$$\kappa = (n+4)^3, \qquad r = 3n, \qquad \kappa_M = (5a_\kappa + 7n + 20)^3. \tag{73}$$

For a hash $H = (H_i)_i$, let its preimage enumeration be

$$E_H = \mathrm{enc}\big((|H_i^{-1}(1)|, \pi_i, \pi_i^{-1})_i\big), \tag{74}$$

where $\pi_i$ lists $H_i^{-1}(1)$ followed by $H_i^{-1}(0)$, each in canonical order. This function permits coherent preparation of the predicate-preimage state, as shown in Appendix C.

**Definition 33** (Signature oracle)**.** *Independently for every parameter $n$ and serial $s \in \{0,1\}^n$, sample*

$$H_s \leftarrow \mathcal{H}_{\kappa_M}, \quad B_s \leftarrow \{0,1\}^{\mathcal{X}_{\kappa_M}}, \quad F_s^{\mathrm{sig}} \leftarrow \{f : \{0,1\}^n \to \{0,1\}^n\},$$
$$H_{s,t} \leftarrow \mathcal{H}_\kappa, \quad B_{s,t} \leftarrow \{0,1\}^{\mathcal{X}_\kappa} \quad \big(t \in \{(\mathsf{C}, m) : m \in \{0,1\}^n\} \cup \{\mathsf{V}\}\big).$$

*Write $H_{s,m} = H_{s,(\mathsf{C},m)}$, $H_s^V = H_{s,\mathsf{V}}$, and similarly for $B$. Define the signing and verification functions by*

$$P_s(0,m,i) = F_s^{\mathrm{sig}}(m)_i, \qquad P_s(1,t,u) = E_{H_{s,t}}(u),$$
$$G_s(m,z) = \mathbf{1}[z = F_s^{\mathrm{sig}}(m)]. \tag{75}$$

*Their encoded functions and the master signing state are*

$$T_s = T_{H_s, B_s, P_s}, \qquad \big|K_s^M\big\rangle = |\Psi_{H_s}\rangle\,, \tag{76}$$

$$T_{s,m} = T_{H_{s,m}, B_{s,m}, 1}, \qquad T_s^V = T_{H_s^V, B_s^V, G_s}. \tag{77}$$

*Conditional on these samples, define the master preparation target and sample its preparation pair:*

$$|\Omega_M\rangle = 2^{-n/2} \sum_s |s\rangle\,|\Psi_{H_s}\rangle\,, \qquad n_M = n + a_{\kappa_M}, \qquad (R_M, S_M) \leftarrow \mathcal{D}_{\mathrm{prep}}(|\Omega_M\rangle).$$

*We define the signature base oracle by*

$$\mathcal{O}_{\mathsf{sig},n} = ((T_s)_s, (T_{s,m})_{s,m}, (T_s^V)_s, R_M, S_M),$$

*and let $\mathcal{O}_{\mathsf{sig}}(n,u) = \mathcal{O}_{\mathsf{sig},n}(u)$, where $u$ specifies a component above and its input.*

Use $\mathcal{O} = (\mathcal{O}_{\mathsf{base}}, N, A)$ with $\mathcal{O}_{\mathsf{base}} = (\mathcal{O}_{\mathsf{sig}}, \mathcal{O}_{\mathrm{rest}})$, where $\mathcal{O}_{\mathrm{rest}} : \{0,1\}^* \to \{0,1\}$ is sampled independently on disjoint addresses, and one $N, A$ extension is sampled over the entire base (Definition 26). All oracle access is coherent.[21]

The subroutine $\mathsf{Read}(s, \mathsf{R_{sk}}, u, b)$ runs $\mathsf{EvalHash}^{T_s}$ on those registers. Using these encoded queries to $E_{H_{s,t}}$, $\mathsf{Issue}(s, \mathsf{R_{sk}}, t)$ prepares $\big|\Psi_{H_{s,t}}\big\rangle$ and returns the signing register for reuse (Appendix C.1).

[21] The displayed tuples have fixed injective binary encodings. We pad the oracle addresses to one polynomial length $L_0(n)$, as for the functional-key construction (Definition 29).

### 17.1.1 Classical verification keys

We define $\mathsf{CSig}$, whose signature is a predicate-preimage state reused by all verification checks.

**Definition 34** (Classical-verification-key signature oracle)**.** *The classical-verification-key signature oracle and its signature states are*

$$|\Sigma_{s,m}\rangle = \left|\Psi_{H_{s,m}}\right\rangle, \qquad \mathcal{O}_{\mathrm{csig},n} = (T_{s,m})_{s,m}. \tag{78}$$

*Here $T_{s,m}$ encodes the constant-one function (Equation (77)). The signing function $P_s$ supplies $E_{H_{s,m}}$, and no public preparation oracle supplies $|\Sigma_{s,m}\rangle$.*

$\underline{\mathsf{CSig.Gen}(1^\lambda)}$

1. Run $\mathsf{Prepare}^{R_M,S_M}(1^{n_M})$ and measure its serial register to obtain $s$, leaving the predicate-preimage state register $\mathsf{R_{sk}}$.
2. Return the verification key $\mathsf{vk} = s$ and signing key $(s, \mathsf{R_{sk}})$.

$\underline{\mathsf{CSig.Sign}((s, \mathsf{R_{sk}}), m)}$

1. Return $\mathsf{R_{sig}} \leftarrow \mathsf{Issue}(s, \mathsf{R_{sk}}, (\mathsf{C}, m))$, together with the updated signing register.

$\underline{\mathsf{CSig.Ver}(s, m, \mathsf{R_{sig}})}$

1. Reject incorrect lengths.
2. For $j = 1, \ldots, r$, choose fresh $y_j \leftarrow \{0,1\}^\kappa$ and run $\mathsf{EvalHash}^{T_{s,m}}$ on $\mathsf{R_{sig}}$, input $y_j$, and a zero response bit. Measure the response and reject unless it equals one. Discard the fresh private workspace and use the updated signature register for the next check.
3. Accept if all checks pass.

### 17.1.2 Quantum verification keys

We define $\mathsf{QSig}$, whose classical signatures are checked by encoded equality queries using a quantum verification key.

**Definition 35** (Quantum-verification-key signature oracle)**.** *The quantum-verification-key signature oracle and its verification states are*

$$\left|K_s^V\right\rangle = \left|\Psi_{H_s^V}\right\rangle, \qquad \mathcal{O}_{\mathrm{qsig},n} = (T_s^V)_s. \tag{79}$$

*Here $T_s^V$ encodes the equality test $G_s(m,z)$ (Equation (77)), and both schemes use the same signing function $P_s$.*

$\underline{\mathsf{QSig.Gen}(1^\lambda)}$

1. Run $\mathsf{Prepare}^{R_M,S_M}(1^{n_M})$ and measure its serial register to obtain $s$, leaving the predicate-preimage state register $\mathsf{R_{sk}}$.
2. Generate $(s, \mathsf{R_{vk}}) \leftarrow \mathsf{QSig.VKGen}((s, \mathsf{R_{sk}}))$.
3. Return the verification key $(s, \mathsf{R_{vk}})$ and signing key $(s, \mathsf{R_{sk}})$.

$\underline{\mathsf{QSig.VKGen}((s, \mathsf{R_{sk}}))}$

1. Run $\mathsf{R_{vk}} \leftarrow \mathsf{Issue}(s, \mathsf{R_{sk}}, \mathsf{V})$.
2. Return $(s, \mathsf{R_{vk}})$, together with the updated signing register.

$\underline{\mathsf{QSig.Sign}((s, \mathsf{R_{sk}}), m)}$

1. For $i \in [n]$, run $\mathsf{Read}(s, \mathsf{R_{sk}}, (0, m, i), 0)$ and measure the response bit to obtain $\sigma_i$.
2. Return $\sigma$ together with the updated signing register.

$\mathsf{QSig.Ver}((s, \mathsf{R_{vk}}), m, \sigma)$

1. Reject incorrect lengths.
2. Run $\mathsf{EvalHash}^{T_s^V}$ on $\mathsf{R_{vk}}$, input $(m, \sigma)$, and a zero response bit.
3. Measure that bit and accept iff it is one.

## 17.2 Security

We now state the common security guarantee for both signature schemes.

**Theorem 17** (Adaptive signature security)**.** *For the signature oracle $\mathcal{O}$ above and every independent base component $\mathcal{O}_{\text{rest}}$, both $\mathsf{CSig}^{\mathcal{O}}$ and $\mathsf{QSig}^{\mathcal{O}}$ are uniform $\mathsf{QPT}$ and satisfy the following guarantees, averaged over $\mathcal{O}$:*

- Correctness. *Polynomially many adaptive honest signing, verification-key, and verification calls have negligible joint error.*
- Adaptive unforgeability. *For either $\mathsf{Sig} \in \{\mathsf{CSig}, \mathsf{QSig}\}$ and every adversary with polynomial query count, query length and number of classical signing and verification-key requests, and unrestricted local computation,*

$$\Pr_{\mathcal{O}}[\mathsf{Forge}^{\mathcal{O}}_{\mathsf{Sig},\mathcal{A}}(\lambda) = 1] \leq 2^{-\Omega_{\mathcal{A}}(\lambda)}.$$

The proof of Theorem 17 appears in Section 17.3.

Combining these signatures with the encryption construction gives public-key quantum money.

**Corollary 2** (Public-key quantum money)**.** *For almost every cryptographic oracle $\mathcal{O}$ (Definition 37), there is a uniform $\mathsf{QPT}$ public-key quantum money scheme with classical verification keys, a reusable quantum minting key and reusable banknotes, secure against polynomial-size quantum counterfeiters with classical descriptions depending on $\mathcal{O}$ and no quantum advice for every polynomial number of adaptive minting requests.*

The proof of Corollary 2 appears in Appendix G.4.

## 17.3 Proof of adaptive signature security

We prove Theorem 17, using the following two bounds for verification with a random Boolean function and for predicting an unsigned function value. The joint output guarantee in Lemma 24 includes the full oracle and all responses to adaptive classical requests.

**Lemma 19** (Verification with a random function)**.** *Put $\kappa = (n+4)^3$ and $\tau_{\text{sig}}(\kappa) = 6\sqrt{(a_\kappa + 1)2^{-\kappa}}$. Let $V_B^r$ be the $r$ checks of $\mathsf{CSig.Ver}$ with $T_{s,m}$ replaced by $B$, setting $B(\perp, y) = 0$. For a uniform random Boolean function $B : \Sigma_\kappa^{q_\kappa} \times \{0,1\}^\kappa \to \{0,1\}$, with the maximum over all $a_\kappa$-qubit density matrices,*

$$\Pr_B\left[\max_\rho \Pr[V_B^r(\rho) = 1] > (2/3)^r\right] \leq \tau_{\text{sig}}(\kappa).$$

The proof of Lemma 19 appears in Appendix H.

**Lemma 20** (Adaptive equality queries)**.** *Let $F : \{0,1\}^n \to \{0,1\}^n$ be a uniform random function and $\mathsf{Eq}_F(m, z) = \mathbf{1}[z = F(m)]$. An algorithm with unrestricted local computation and an arbitrary auxiliary state independent of $F$ may make classical signing requests $m \mapsto F(m)$ and at most $Q$ quantum queries to $\mathsf{Eq}_F$. If $S$ is the set of signed messages and the algorithm outputs $(m, z)$, then*

$$\Pr_F[m \notin S \ \wedge \ z = F(m)] \leq 2(Q+1)2^{-n/2}.$$

The proof of Lemma 20 appears in Appendix H. We now prove adaptive security for both signature schemes.

**The forgery experiments.** Fix a forger $\mathcal{A}$ and write $n = \lambda$. The bounds below are uniform in the independently sampled additional base $\mathcal{O}_{\mathrm{rest}}$. Use $D = C, Q$ for the two schemes, writing $(\mathsf{Gen}_D, \mathsf{Sign}_D, \mathsf{Ver}_D)$ for their generation, signing and verification algorithms. In $\mathsf{S}_0^D$, sample $\mathcal{O}^0 = (\mathcal{O}^0_{\mathsf{base}}, N^0, A^0) = \mathcal{O}$ and run $(\mathsf{R}_{\mathsf{vk}}, \mathsf{R}_{\mathsf{sk}}) \leftarrow \mathsf{Gen}_D^{\mathcal{O}^0}(1^n)$. Initialize $S = \varnothing$ and run $\mathcal{A}^{\mathcal{O}^0}(\mathsf{R}_{\mathsf{vk}})$, answering each classical signing request $m$ by $\mathsf{Sign}_D^{\mathcal{O}^0}(\mathsf{R}_{\mathsf{sk}}, m)$, updating the signing register and setting $S = S \cup \{m\}$. For $D = Q$, answer verification-key requests by $\mathsf{QSig.VKGen}^{\mathcal{O}^0}(\mathsf{R}_{\mathsf{sk}})$ with the updated signing key. When $\mathcal{A}$ outputs $(m, \mathsf{R}_{\mathsf{sig}})$, return zero if $m \in S$, and otherwise return $\mathsf{Ver}_D^{\mathcal{O}^0}(\mathsf{R}_{\mathsf{vk}}, m, \mathsf{R}_{\mathsf{sig}})$, using a newly generated verification key for the final $Q$ test.

For $\mathsf{S}_1^D$, sample $\mathcal{O}^1 = \mathcal{O}$ and $s \leftarrow \{0,1\}^n$ independently, initialize $S = \varnothing$, and run $\mathcal{A}$ as follows:

1. Give $s$ to $\mathcal{A}$ in scheme $C$, or $(s, \left|\Psi_{H_s^V}\right\rangle)$ in scheme $Q$, with coherent access to $\mathcal{O}^1$ in either case.
2. On a classical signing request $m$, set $S = S \cup \{m\}$ and return $\left|\Psi_{H_{s,m}}\right\rangle$ in scheme $C$, or $F_s^{\mathrm{sig}}(m)$ in scheme $Q$. In scheme $Q$, answer each verification-key request with a fresh $\left|\Psi_{H_s^V}\right\rangle$ using the same sampled hash $H_s^V$.
3. On output $(m, \mathsf{R}_{\mathsf{sig}})$, reject if $m \in S$. Otherwise run $\mathsf{CSig.Ver}(s, m, \mathsf{R}_{\mathsf{sig}})$ in scheme $C$, or run $\mathsf{QSig.Ver}((s, \left|\Psi_{H_s^V}\right\rangle), m, \mathsf{R}_{\mathsf{sig}})$ with a fresh verification key in scheme $Q$, and return its acceptance bit.

Signature correctness (Lemma 24) applies jointly to all adaptively requested outputs and the fresh final verification key, including their correlations with $\mathcal{O}$. Thus, writing $p_i^D = \Pr[\mathsf{S}_i^D = 1]$, we have

$$|p_0^D - p_1^D| \le 2^{-\Omega(n)}.$$

In $\mathsf{S}_2^D$, use exactly this challenger, but replace the signing-state target and sample its oracle as follows:

$$\left|\Omega_M^0\right\rangle = 2^{-n/2}\left(\sum_{t \neq s} |t\rangle\left|\Psi_{H_t}\right\rangle + |s\rangle\left|0^{n_M - n}\right\rangle\right),$$

$$(R_M^0, S_M^0) \leftarrow \mathcal{D}_{\mathrm{prep}}(\left|\Omega_M^0\right\rangle), \qquad \mathcal{O}^2 = (\mathcal{O}^2_{\mathsf{base}}, N^2, A^2).$$

Set $\mathcal{O}^2_{\mathsf{base}} = \mathcal{O}^1_{\mathsf{base}}[(R_M, S_M) \leftarrow (R_M^0, S_M^0)]$ and sample $N^2, A^2$ conditional on $\mathcal{O}^2_{\mathsf{base}}$. Run $\mathcal{A}^{\mathcal{O}^2}$ with the same initial input and classical request rules as in $\mathsf{S}_1^D$, and give the final verifier $\mathcal{O}^2$ as well. Thus the signing and verification-key responses use the original functions and hashes in Steps 1–2.

**Quantum verification keys.** In $\mathsf{S}_3^Q$, keep $\mathcal{O}^3 = \mathcal{O}^2$ and run the same $\mathcal{A}^{\mathcal{O}^3}$ with the same signing and verification-key channels, replacing only its final test by

$$\mathbf{1}[m \notin S]\, G_s(m, \sigma), \qquad G_s(m, z) = \mathbf{1}[z = F_s^{\mathrm{sig}}(m)].$$

The final verifier in $\mathsf{S}_2^Q$ uses a fresh $\left|\Psi_{H_s^V}\right\rangle$, so encoded evaluation (Theorem 14) on the good-hash event, followed by averaging its complement, gives

$$|p_2^Q - p_3^Q| \le 2\sqrt{\delta_{\mathrm{dec}}(\kappa)} + 2^{-\kappa}.$$

We replace this test before simulating the public encodings.

Let $G_Q$ consist of the message hashes $(H_{s,m})_m$ and $\mathcal{O}^2_{\mathsf{base}}$ with $T_s, T_s^V$ omitted. For each fixed $G_Q$, encoded-function simulation (Theorem 14) supplies the algorithm $\mathcal{S}_{G_Q}$ used in the next experiment, with only the verification instance opened. Its hypotheses are checked below. Experiment $\mathsf{S}_4^Q$ runs it as follows:

1. Draw one classical sample $\eta_Q$ conditional on $G_Q$, initialize $S = \varnothing$, and run $\mathcal{S}^{G_s}_{G_Q}(s, \eta_Q)$. Inside this call, $\mathcal{S}_{G_Q}$ initializes $\mathcal{A}$ on $(s, \left|\Psi_{H_s^V}\right\rangle)$ using its sampled $H_s^V$.
2. On a signing request $m$, query the classical signing oracle for $F_s^{\mathrm{sig}}(m)$ to obtain $z$, set $S = S \cup \{m\}$, and return $z$. On a verification-key request, prepare another copy of $\left|\Psi_{H_s^V}\right\rangle$ from the same sampled $H_s^V$.
3. Answer the simulator's function queries by $G_s$. On the final $(m, \sigma)$, reject if $m \in S$ and otherwise return $G_s(m, \sigma)$.

**Classical verification keys.** Let $\mathcal{O}^{2,-\mathsf{C}}_{\mathsf{base}}$ denote $\mathcal{O}^2_{\mathsf{base}}$ with $T_s$ and all $T_{s,m}$ omitted, and set

$$G_C = (F_s^{\mathrm{sig}}, H_s^V, \mathcal{O}^{2,-\mathsf{C}}_{\mathsf{base}}).$$

For each fixed $G_C$, encoded-function simulation (Theorem 14) supplies $\mathcal{S}_{G_C}$, opening a message index precisely when it is requested for signing. The proof below checks this adaptive application. In $\mathsf{S}_3^C$, run this algorithm as follows:

1. Draw one classical sample $\eta_C$ conditional on $G_C$, initialize $S = \varnothing$, and run $\mathcal{S}_{G_C}(s, \eta_C)$, which initializes $\mathcal{A}$ on $s$.
2. On a signing request $m$, set $S = S \cup \{m\}$ and supply $\left|\Psi_{H_{s,m}}\right\rangle$ using the sampled $H_{s,m}$. Between requests, queries to the encodings use

$$T_s \mapsto B_s, \qquad T_{s,m} \mapsto \begin{cases} T_{H_{s,m},B_{s,m},1}, & m \in S, \\ B_{s,m}, & m \notin S. \end{cases}$$

   Every $\mathcal{O}^2$ query is answered by the query algorithm $\mathcal{S}_{\eta_C}(S, \cdot)$ using this one classical sample. Repeated signing requests use the same $H_{s,m}, B_{s,m}$.
3. On $(m, \mathsf{R}_{\mathsf{sig}})$, reject if $m \in S$. Otherwise run the $3n$ checks of $\mathsf{CSig.Ver}$ with $B_{s,m}$ and return their joint acceptance bit.

The following claim justifies the three public-oracle changes used by these experiments.

**Claim 5.** *For the fixed forger and either $D \in \{C, Q\}$,*

$$|p_1^D - p_2^D| + |p_3^Q - p_4^Q| + |p_2^C - p_3^C| \le 2^{-\Omega(n)}.$$

*The simulator in $\mathsf{S}_4^Q$ makes $Q' = \mathsf{poly}(n)$ queries only to $G_s$, in addition to the classical signing requests, and its classical sample and description are independent of $F_s^{\mathrm{sig}}$. In $\mathsf{S}_3^C$, every unsigned message is tested with its original uniform Boolean function $B_{s,m}$.*

The proof of Claim 5 appears in Section 17.3.1.

**Bounding the forgery probabilities.** In $\mathsf{S}_4^Q$, the simulator has classical signing access to $F_s^{\mathrm{sig}}$ and $Q' = \mathsf{poly}(n)$ equality queries, with an independent classical sample. Adaptive equality-query security (Lemma 20) therefore gives

$$p_4^Q \le 2(Q' + 1)2^{-n/2}.$$

Combining this with the adjacent bounds proves

$$p_0^Q \le p_4^Q + \sum_{i=0}^{3} |p_i^Q - p_{i+1}^Q| \le 2(Q'+1)2^{-n/2} + 2\sqrt{\delta_{\mathrm{dec}}(\kappa)} + 2^{-\kappa} + 2^{-\Omega(n)}.$$

Define the event

$$\mathsf{Bad}_{\mathrm{sig}} = \left\{ \exists m \in \{0,1\}^n : \max_{\rho} \Pr\left[ V^{3n}_{B_{s,m}}(\rho) = 1 \right] > (2/3)^{3n} \right\}.$$

Verification with a random function (Lemma 19) applies to each $B_{s,m}$, and a union bound gives

$$\Pr[\mathsf{Bad}_{\mathrm{sig}}] \le 2^n \tau_{\mathrm{sig}}(\kappa), \qquad p_3^C \le (2/3)^{3n} + 2^n \tau_{\mathrm{sig}}(\kappa).$$

The maximum ranges over every state depending on the selected function, so this estimate applies to the adaptively chosen forged message without conditioning on that choice. Therefore

$$p_0^C \le p_3^C + \sum_{i=0}^{2} |p_i^C - p_{i+1}^C| \le (2/3)^{3n} + 2^n \tau_{\mathrm{sig}}(\kappa) + 2^{-\Omega(n)}.$$

Both forgery bounds are $2^{-\Omega(n)}$.

### 17.3.1 Bounds for the signature hybrids

We now prove Claim 5. The preparation change keeps the original signing responses, and the two encoded simulations open precisely the hashes used in those responses.

**Changing the signing-state target.** Fix $s$, all sampled functions and hashes, and the preparation pairs other than $(R_M, S_M)$. They determine both targets and every request-response rule before the changed pair and recursive coins are sampled. The targets satisfy

$$\| \, |\Omega_M\rangle - \left|\Omega_M^0\right\rangle \, \|_2 \le 2^{1-n/2}.$$

Apply the target-change bound (Theorem 13) with dimension $n_M$, the initial state in Step 1, the response channels in Steps 2–3, and recursive mean $h(v) = (1+v)/3$, obtaining

$$|p_1^D - p_2^D| \le 2^{-\Omega(n)}.$$

The original signing and verification-key rules are the same in both experiments, so this bound includes adaptive requests.

**The quantum verification key.** The target $\left|\Omega_M^0\right\rangle$ replaces $|\Psi_{H_s}\rangle$ by $|0^{n_M - n}\rangle$, and all other primitive bases are independent, so the families

$$F_s^{\mathrm{sig}}, \quad (H_s, H_s^V), \quad (B_s, B_s^V) \qquad \text{are mutually independent conditional on } G_Q.$$

For each fixed $G_Q$, apply encoded-function simulation (Theorem 14) with $J = \{\mathsf{M}, \mathsf{V}\}$ and $K = \{\mathsf{V}\}$ for $T_s, T_s^V$, opening only the verification instance at the start and on later verification-key requests, with direct access only to $G_s$. The function $G_s$ depends on $F_s^{\mathrm{sig}}$, whereas the discarded master function may also use $H_s^V$ through $P_s(1, \mathsf{V}, u) = E_{H_s^V}(u)$, as allowed by the theorem. The opened verification index is known throughout this single-adversary interaction. Every $\mathcal{O}^2$ query

is answered by the coherent query algorithm with this opened set and the same classical sample $\eta_Q$. Its only direct queries are to $G_s$, and it has no original $N, A$ access. The unchanged request channels use $F_s^{\mathrm{sig}}$ and the opened verification hash, so the theorem gives

$$|p_3^Q - p_4^Q| \le 2^{-\Omega(n)}.$$

The sample and description are independent of $F_s^{\mathrm{sig}}$ conditional on $G_Q$, which is itself independent of that function. Averaging over $G_Q$ therefore establishes the equality-query experiment in the claim.

**Classical verification keys.** Conditional on $G_C$, the hashes are independent draws $H_s \leftarrow \mathcal{H}_{\kappa_M}$ and $H_{s,m} \leftarrow \mathcal{H}_\kappa$ for each $m$. The functions $B_s, (B_{s,m})_m$ are independent uniform Boolean functions, independent of those hashes. A signing response uses only its requested hash $H_{s,m}$, and these hashes occur elsewhere only in the discarded master function, through $P_s(1, (\mathsf{C}, m), u) = E_{H_{s,m}}(u)$. There is no public preparation pair for a message hash.

Apply encoded-function simulation (Theorem 14) with $J = \{\mathsf{M}\} \cup \{0,1\}^n$ and $K = \{0,1\}^n$, where $\mathsf{M}$ indexes $T_s$ and each message $m$ indexes $T_{s,m}$. Each signing request supplies a state using only its opened hash, so the adaptive input hypothesis holds and gives $|p_2^C - p_3^C| \le 2^{-\Omega(n)}$. In particular, the final test uses the original uniformly random Boolean function $B_{s,m}$ for every unsigned $m$.

The forger knows its signed-message set $S$, which is exactly the opened set used in $\mathcal{S}_{\eta_C}(S, \cdot)$. Thus no additional request information or oracle access is required.

# 18 Pseudorandom States

We construct PRS with classical keys and polynomial-copy security (Theorem 18), using binary-phase states as the preparation targets.

**Definition 36** (PRS oracle)**.** *Independently for every $n$ and $k \in \{0,1\}^n$, sample a uniform function $\Theta_k : \{0,1\}^{5n} \to \{0,1\}$ and then sample*

$$|\Omega_{\Theta_k}\rangle = 2^{-5n/2} \sum_{x \in \{0,1\}^{5n}} (-1)^{\Theta_k(x)} |x\rangle\,, \tag{80}$$

$$|\Omega_k\rangle = |\Omega_{\Theta_k}\rangle\,, \qquad (R_k^{\mathsf{prs}}, S_k^{\mathsf{prs}}) \leftarrow \mathcal{D}_{\mathrm{prep}}(|\Omega_k\rangle). \tag{81}$$

*We define the PRS base oracle by $\mathcal{O}_{\mathsf{prs},n} = ((R_k^{\mathsf{prs}}, S_k^{\mathsf{prs}}))_k$, and let $\mathcal{O}_{\mathsf{prs}}(n,u) = \mathcal{O}_{\mathsf{prs},n}(u)$, where $u$ specifies a component above and its input. Only the preparation pairs are oracle components.*

Use $\mathcal{O} = (\mathcal{O}_{\mathsf{base}}, N, A)$ with $\mathcal{O}_{\mathsf{base}} = (\mathcal{O}_{\mathsf{prs}}, \mathcal{O}_{\mathrm{rest}})$, where $\mathcal{O}_{\mathrm{rest}} : \{0,1\}^* \to \{0,1\}$ is sampled independently on disjoint addresses, and one $N, A$ extension is sampled over the entire base (Definition 26). In particular, direct access to the phase functions is unavailable.

## 18.1 Construction

For fixed $\mathcal{O}, k$, write $\rho_k^{\mathcal{O}}$ for the output density matrix of the following state-generation algorithm. Its $r$ calls use independent internal coins, so their joint output is $(\rho_k^{\mathcal{O}})^{\otimes r}$.

$\underline{\mathsf{KeyGen}(1^\lambda)}$

1. Return $k \leftarrow \{0,1\}^\lambda$.

$\underline{\mathsf{StateGen}^{\mathcal{O}}(1^\lambda, k)}$

1. Run $\mathsf{Prepare}^{R_k^{\mathsf{prs}}, S_k^{\mathsf{prs}}}(1^{5\lambda})$ and return the entire output.

All calls use the same hidden classical key $k$, and the following theorem establishes pseudorandomness for polynomially many copies.

**Theorem 18** (PRS security)**.** *Use the PRS oracle $\mathcal{O}$ above with any independent base component $\mathcal{O}_{\mathrm{rest}}$. The generator is uniform* QPT*. Sample $k \leftarrow \{0,1\}^\lambda$ and put $\rho_{k,r}^{\mathcal{O}} = (\rho_k^{\mathcal{O}})^{\otimes r}$. For every polynomial $r$ and quantum algorithm $\mathcal{A}$ with polynomial query count and length and unrestricted local computation,*

$$\left| \Pr_{\mathcal{O},k}[\mathcal{A}^{\mathcal{O}}(\rho_{k,r}^{\mathcal{O}}) = 1] - \Pr_{\mathcal{O},\psi \leftarrow \mathrm{Haar}_{5\lambda}}[\mathcal{A}^{\mathcal{O}}(|\psi\rangle^{\otimes r}) = 1] \right| \le 2^{-\Omega_{\mathcal{A}}(\lambda)}. \tag{82}$$

The proof of Theorem 18 appears in Section 18.2.

## 18.2 Proof of PRS security

We now prove Theorem 18 by changing the supplied state first to an independent binary-phase state and then to a Haar state.

Fix the additional base component. Each experiment below samples the full oracle $\mathcal{O} = (\mathcal{O}_{\mathsf{base}}, N, A)$ using the PRS preparation pairs (Definition 36), and changes only the input supplied to $\mathcal{A}^{\mathcal{O}}$:

$\mathsf{H}_0$ Sample one $k \leftarrow \{0,1\}^\lambda$, run $\mathsf{StateGen}^{\mathcal{O}}(1^\lambda, k)$ independently $r$ times, and output

$$\mathcal{A}^{\mathcal{O}}((\rho_k^{\mathcal{O}})^{\otimes r}).$$

$\mathsf{H}_1$ Sample $\Theta \leftarrow \{\{0,1\}^{5\lambda} \to \{0,1\}\}$ independently of $\mathcal{O}$, and output $\mathcal{A}^{\mathcal{O}}((|\Omega_\Theta\rangle\langle\Omega_\Theta|)^{\otimes r})$.

$\mathsf{H}_2$ Sample $\psi \leftarrow \mathrm{Haar}_{5\lambda}$ independently of $\mathcal{O}$, and output $\mathcal{A}^{\mathcal{O}}((|\psi\rangle\langle\psi|)^{\otimes r})$.

All $r$ outputs in $\mathsf{H}_0$ use the same key and preparation pair, while the other experiments use $r$ copies of one independent state. To justify the first change, apply the independent-input bound (Theorem 20) to the family $(\Theta_k)_{k\in\{0,1\}^\lambda}$, its preparation pairs, and the input $(\rho_k^{\mathcal{O}})^{\otimes r}$. Its hypotheses hold because these phase functions are independent and occur in $\mathcal{O}_{\mathsf{base}}$ only through their preparation pairs, while all other base components are independent. Its joint-output guarantee includes all $r$ copies and the full quantum oracle access, so

$$|\Pr[\mathsf{H}_0 = 1] - \Pr[\mathsf{H}_1 = 1]| \le 2^{-\Omega_{\mathcal{A}}(\lambda)}.$$

For the next step, the binary-phase moment bound [BS19] gives

$$\tfrac{1}{2}\left\|\mathbb{E}_\Theta[(|\Omega_\Theta\rangle\langle\Omega_\Theta|)^{\otimes r}] - \mathbb{E}_{\psi\leftarrow\mathrm{Haar}_{5\lambda}}[(|\psi\rangle\langle\psi|)^{\otimes r}]\right\|_1 \le O(r^2 2^{-5\lambda}). \tag{83}$$

The input state in $\mathsf{H}_1$ and $\mathsf{H}_2$ is independent of $\mathcal{O}$, so running $\mathcal{A}^{\mathcal{O}}$ and measuring its output is the same quantum channel applied to those two averaged states. Since quantum operations cannot increase trace distance,

$$|\Pr[\mathsf{H}_1 = 1] - \Pr[\mathsf{H}_2 = 1]| \le O(r^2 2^{-5\lambda}).$$

The triangle inequality now gives

$$\left|\Pr[\mathsf{H}_0 = 1] - \Pr[\mathsf{H}_2 = 1]\right| \le 2^{-\Omega_{\mathcal{A}}(\lambda)} + O(r^2 2^{-5\lambda}) = 2^{-\Omega_{\mathcal{A}}(\lambda)}.$$

The estimate is uniform in the fixed additional base component, so averaging over that component proves the theorem.

Theorem 19 gives the PRS guarantee (Definition 12) and bounds the preparation error (Equation (14)) simultaneously for every key.

# 19 Quantum Cryptography Relative to an Oracle with $\mathsf{P} = \mathsf{QCMA}$

In this section, we establish all cryptographic guarantees relative to one fixed oracle. Theorem 19 includes security against nonuniform classical circuit descriptions depending on that oracle, with no quantum advice.

The four independently sampled base oracles can now be combined before adding the recursive extension.

**Definition 37** (The cryptographic oracle)**.** *Independently sample the PRF, encryption, signature and PRS base oracles (Definitions 31 to 33 and 36). We define the cryptographic oracle by*

$$\mathcal{O}_{\mathsf{base}} = (\mathcal{O}_{\mathsf{prf}}, \mathcal{O}_{\mathsf{pke}}, \mathcal{O}_{\mathsf{sig}}, \mathcal{O}_{\mathsf{prs}}), \qquad \mathcal{O} = (\mathcal{O}_{\mathsf{base}}, N, A), \tag{84}$$

*where $\mathcal{O}_{\mathsf{base}}(c, u) = \mathcal{O}_c(u)$ for $c \in \{\mathsf{prf}, \mathsf{pke}, \mathsf{sig}, \mathsf{prs}\}$, with zero on unused addresses. One pair $N, A$ is sampled from this entire base (Definition 26).*

The following theorem makes the four guarantees simultaneous for a single fixed oracle.

**Theorem 19** (Cryptography for one fixed oracle)**.** *With probability one over the cryptographic oracle $\mathcal{O} = (\mathcal{O}_{\mathsf{base}}, N, A)$ (Definition 37), the following hold relative to $\mathcal{O}$. Every security guarantee is against polynomial-size quantum oracle circuit families with classical descriptions, allowing dependence on $\mathcal{O}$ but no quantum advice.*

- *The promise classes satisfy*

$$\mathsf{PromiseP}^{\mathcal{O}} = \mathsf{PromiseBQP}^{\mathcal{O}} = \mathsf{PromiseQCMA}^{\mathcal{O}} = \mathsf{PromiseQCMAH}^{\mathcal{O}}.$$

- *The PRF $\mathsf{PRF}^{\mathcal{O}}$ (Section 15) satisfies correctness, pseudorandomness and copy-protection as in Definitions 9 and 39.*
- *The encryption scheme $\mathsf{PKE}^{\mathcal{O}}$ (Section 16) satisfies correctness, chosen-plaintext security and copy-protection under independent ciphertext challenges, as in Definitions 10 and 40.*
- *The signature schemes $\mathsf{CSig}^{\mathcal{O}}$ and $\mathsf{QSig}^{\mathcal{O}}$ (Section 17) satisfy correctness and adaptive unforgeability as in Definition 11.*
- *The PRS scheme $\mathsf{PRS}^{\mathcal{O}}$ (Section 18) satisfies the PRS definition (Definition 12), with $\lambda$-bit keys and $5\lambda$-qubit outputs. Its preparation error (Equation (14)) is negligible uniformly over its keys, and its security holds for polynomially many same-key outputs.*

The proof of Theorem 19 appears in Appendix K. Observe that the uniform preparation bound also gives PRS security for polynomially many copies of the prescribed state $|\Omega_k\rangle$ (Equation (81)), by the tensor-product trace-distance bound.

Note that this also (trivially) implies that these primitives exist relative to a fixed oracle $\mathcal{O}$ such that $\mathsf{PromiseP}^{\mathcal{O}} = \mathsf{PromiseQCMAH}^{\mathcal{O}}$.

The encryption scheme also yields average-case hardness in $\mathsf{QMA}$.

**Corollary 3** (Average-case $\mathsf{QCMAH}$ hardness in $\mathsf{QMA}$)**.** *Fix an oracle $\mathcal{O}$ satisfying Theorem 19. There are a $\mathsf{QMA}$ promise problem $\Pi = (\Pi_0, \Pi_1)$[22] and a uniform $\mathsf{QPT}$ sampler $\mathsf{D}^{\mathcal{O}}(1^\lambda) \to (X, b)$ with classical output and uniform $b \in \{0, 1\}$ such that, for every constant $d$ and every uniform $\mathsf{QCMAH}_d$ machine $M_d$, counting $\perp$ answers as incorrect,*

$$\Pr_{X,b}[X \in \Pi_b] \geq 1 - \mathsf{negl}(\lambda), \qquad \Pr_{X,b}[M_d^{\mathcal{O}}(X) = b] \leq \tfrac{1}{2} + \mathsf{negl}_{M_d}(\lambda).$$

[22] We state the promise-problem version here. A total-language version can also be proved by adapting our lemmas.

The proof of Corollary 3 appears in Appendix K.2.

Through known black-box constructions, we obtain the following corollary.

**Corollary 4** (Further cryptographic primitives)**.** *For almost every cryptographic oracle $\mathcal{O}$ (Definition 37), the following primitives exist relative to $\mathcal{O}$ and satisfy correctness and security against polynomial-size quantum oracle circuit families with classical descriptions depending on $\mathcal{O}$ and no quantum advice:*

- *Collusion-resistant public-key quantum money with classical verification keys.*
- *For every polynomial $q$, $q$-bounded collusion-resistant copy-protected PKE and PRF schemes.*
- *Pseudorandom unitaries with a quantum key, secure under polynomially many forward oracle queries.*
- *One-way puzzles with possibly inefficient verification.*
- *One-way state generators with classical keys.*
- *For every polynomial $q$, signatures with quantum verification keys and classical signing keys and signatures, secure against $q(\lambda)$ adaptive signing queries.*

## References


[AA11] Scott Aaronson and Andris Ambainis. The need for structure in quantum speedups. In Bernard Chazelle, editor, *ICS 2011: 2nd Innovations in Computer Science*, pages 338–352, Tsinghua University, Beijing, China, January 7–9, 2011. Tsinghua University Press.

[AA15] Scott Aaronson and Andris Ambainis. Forrelation: A problem that optimally separates quantum from classical computing. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *47th Annual ACM Symposium on Theory of Computing*, pages 307–316, Portland, OR, USA, June 14–17, 2015. ACM Press.

[Aar09] Scott Aaronson. Quantum copy-protection and quantum money, 2009. CCC 2009. arXiv:1110.5353, version 1.

[Aar10] Scott Aaronson. BQP and the polynomial hierarchy, 2010. STOC 2010. Full version: arXiv:0910.4698, version 1.

[Aar12] Scott Aaronson. Impossibility of succinct quantum proofs for collision-freeness. *Quantum Information and Computation*, 12(1–2):21–28, 2012.

[Aar14] Scott Aaronson. The equivalence of sampling and searching. *Theory of Computing Systems*, 55(2):281–298, 2014.

[AB24] Prabhanjan Ananth and Amit Behera. A modular approach to unclonable cryptography. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology – CRYPTO 2024, Part VII*, volume 14926 of *Lecture Notes in Computer Science*, pages 3–37, Santa Barbara, CA, USA, August 18–22, 2024. Springer, Cham, Switzerland.

[ABD26] Avantika Agarwal and Shalev Ben-David. Oracle separations for the quantum-classical polynomial hierarchy. https://arxiv.org/abs/2410.19062v2, 2026. ITCS 2026. arXiv:2410.19062, version 2, 3 December 2025.

[ABH+26] Prabhanjan Ananth, Amit Behera, Zikuan Huang, Fuyuki Kitagawa, and Takashi Yamakawa. Copy-protection from unclonable puncturable obfuscation, revisited. In *Advances in Cryptology—EUROCRYPT 2026*, volume 16541 of *Lecture Notes in Computer Science*, pages 541–570. Springer, 2026.

[AC13] Scott Aaronson and Paul Christiano. Quantum money from hidden subspaces. *Theory of Computing*, 9(9):349–401, 2013.

[AIK22] Scott Aaronson, DeVon Ingram, and William Kretschmer. The acrobatics of BQP. In *37th Computational Complexity Conference (CCC 2022)*, volume 234 of *Leibniz International Proceedings in Informatics*, pages 20:1–20:17, 2022. Full version: arXiv:2111.10409, version 4.

[AK07] Scott Aaronson and Greg Kuperberg. Quantum versus classical proofs and advice. *Theory of Computing*, 3(7):129–157, 2007.

[ALL+20] Scott Aaronson, Jiahui Liu, Qipeng Liu, Mark Zhandry, and Ruizhe Zhang. New approaches for quantum copy-protection, 2020. arXiv:2004.09674, version 7, 16 October 2020.

[AN02] Dorit Aharonov and Tomer Naveh. Quantum NP—a survey, 2002.

[AQY22] Prabhanjan Ananth, Luowen Qian, and Henry Yuen. Cryptography from pseudorandom quantum states. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022, Part I*, volume 13507 of *Lecture Notes in Computer Science*, pages 208–236, Santa Barbara, CA, USA, August 15–18, 2022. Springer, Cham, Switzerland.

[BBBV97] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997.

[BBC+98] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. In *39th Annual Symposium on Foundations of Computer Science*, pages 352–361, Palo Alto, CA, USA, November 8–11, 1998. IEEE Computer Society Press.

[BBF13] Paul Baecher, Christina Brzuska, and Marc Fischlin. Notions of black-box reductions, revisited. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology – ASIACRYPT 2013, Part I*, volume 8269 of *Lecture Notes in Computer Science*, pages 296–315, Bengalore, India, December 1–5, 2013. Springer Berlin Heidelberg, Germany.

[BCG26] Amit Behera, Alper Cakan, and Vipul Goyal. Decisional monogamy-of-entanglement for coset states and applications to unclonable cryptography with correlated challenges, 2026. arXiv:2608.18841, version 2, 1 October 2026.

[BCN25] John Bostanci, Boyang Chen, and Barak Nehoran. Oracle separation between quantum commitments and quantum one-wayness, 2025. arXiv:2410.03358, version 3, 11 April 2025.

[BCQ23] Zvika Brakerski, Ran Canetti, and Luowen Qian. On the computational hardness needed for quantum cryptography. In Yael Tauman Kalai, editor, *ITCS 2023: 14th Innovations in Theoretical Computer Science Conference*, volume 251, pages 24:1–24:21, Cambridge, MA, USA, January 10–13, 2023. Leibniz International Proceedings in Informatics (LIPIcs).

[BDK24] Shalev Ben-David and Srijita Kundu. Oracle separation of QMA and QCMA with bounded adaptivity. In *51st International Colloquium on Automata, Languages, and Programming (ICALP 2024)*, volume 297 of *Leibniz International Proceedings in Informatics*, pages 21:1–21:18, 2024.

[BG26] James Bartusek and Eli Goldin. Unclonable encryption in the Haar random oracle model, 2026. CRYPTO 2026. Full version: arXiv:2603.11437, version 2, 15 June 2026.

[BHNZ26] John Bostanci, Jonas Haferkamp, Chinmay Nirkhe, and Mark Zhandry. Separating QMA from QCMA with a classical oracle. https://arxiv.org/abs/2511.09551v2, 2026. STOC 2026. arXiv:2511.09551, version 2.

[BHV26] John Bostanci, Andrew Huang, and Vinod Vaikuntanathan. Separating quantum and classical advice with good codes. https://arxiv.org/abs/2602.09385v2, 2026. arXiv:2602.09385, version 2.

[BJ24] Rishabh Batra and Rahul Jain. Commitments are equivalent to statistically-verifiable one-way state generators. In *65th Annual Symposium on Foundations of Computer Science*, pages 1178–1192, Chicago, IL, USA, October 27–30, 2024. IEEE Computer Society Press.

[BMM+25] Amit Behera, Giulio Malavolta, Tomoyuki Morimae, Tamer Mour, and Takashi Yamakawa. A new world in the depths of microcrypt: Separating OWSGs and quantum money from QEFID, 2025. EUROCRYPT 2025. Full version: arXiv:2410.03453, version 3, 5 February 2025.

[BS19] Zvika Brakerski and Omri Shmueli. (pseudo) random quantum states with binary phase, 2019. Version 2, 26 June 2019.

[BS20] Zvika Brakerski and Omri Shmueli. Scalable pseudorandom quantum states. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020, Part II*, volume 12171 of *Lecture Notes in Computer Science*, pages 417–440, Santa Barbara, CA, USA, August 17–21, 2020. Springer, Cham, Switzerland.

[Cak26] Alper Cakan. Separating QMA from QCIP with a classical oracle, or, the power of quantum proofs over classical interaction for quantum verifiers, 2026.

[CCHL22] Sitan Chen, Jordan Cotler, Hsin-Yuan Huang, and Jerry Li. Exponential separations between learning with and without quantum memory. In *62nd Annual Symposium on Foundations of Computer Science*, pages 574–585, Denver, CO, USA, February 7–10, 2022. IEEE Computer Society Press.

[CCHY26] Alper Cakan, Kai-Min Chung, Wei-Hsiang Hung, and Tzu-Yi Yang. Separating ClonableQMA and QCMA relative to a classical oracle, 2026. arXiv:2610.06740, version 1, 5 October 2026.

[CG24a] Alper Cakan and Vipul Goyal. Unclonable cryptography with unbounded collusions and impossibility of hyperefficient shadow tomography. In Elette Boyle and Mohammad Mahmoody, editors, *TCC 2024: 22nd Theory of Cryptography Conference, Part III*, volume 15366 of *Lecture Notes in Computer Science*, pages 225–256, Milan, Italy, December 2–6, 2024. Springer, Cham, Switzerland.

[CG24b] Andrea Coladangelo and Sam Gunn. How to use quantum indistinguishability obfuscation. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *56th Annual ACM Symposium on Theory of Computing*, pages 1003–1008, Vancouver, BC, Canada, June 24–28, 2024. ACM Press.

[CG25] Alper Cakan and Vipul Goyal. How to copy-protect all puncturable functionalities without conjectures: A unified solution to quantum protection. Cryptology ePrint Archive, Paper 2025/1197, 2025.

[CGG24] Kai-Min Chung, Eli Goldin, and Matthew Gray. On central primitives for quantum cryptography with classical communication. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology – CRYPTO 2024, Part VII*, volume 14926 of *Lecture Notes in Computer Science*, pages 215–248, Santa Barbara, CA, USA, August 18–22, 2024. Springer, Cham, Switzerland.

[CGK+26] Alper Cakan, Vipul Goyal, Fuyuki Kitagawa, Ryo Nishimaki, and Takashi Yamakawa. Multi-copy security in quantum cryptography and more. https://arxiv.org/abs/2510.12626v2, 2026.

[CGL+20] Nai-Hui Chia, András Gilyén, Tongyang Li, Han-Hsuan Lin, Ewin Tang, and Chunhao Wang. Sampling-based sublinear low-rank matrix arithmetic framework for dequantizing quantum machine learning. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *52nd Annual ACM Symposium on Theory of Computing*, pages 387–400, Chicago, IL, USA, June 22–26, 2020. ACM Press.

[CGS26] Alper Cakan, Vipul Goyal, and Omri Shmueli. Public-key quantum fire and key-fire from classical oracles. https://arxiv.org/abs/2504.16407v2, 2026. arXiv:2504.16407, version 2, 14 March 2026.

[CLLZ22] Andrea Coladangelo, Jiahui Liu, Qipeng Liu, and Mark Zhandry. Hidden cosets and applications to unclonable cryptography, 2022. arXiv:2107.05692, version 3, 14 July 2022.

[FK18] Bill Fefferman and Shelby Kimmel. Quantum vs. classical proofs and subset verification. In *43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018)*, volume 117 of *Leibniz International Proceedings in Informatics*, pages 22:1–22:23, 2018.

[GGJL25] Mika Göös, Tom Gur, Siddhartha Jain, and Jiawei Li. Quantum communication advantage in TFNP. In Michal Koucký and Nikhil Bansal, editors, *57th Annual ACM Symposium on Theory of Computing*, pages 1465–1475, Prague, Czechia, June 23–27, 2025. ACM Press.

[GKK+07] Dmitry Gavinsky, Julia Kempe, Iordanis Kerenidis, Ran Raz, and Ronald de Wolf. Exponential separations for one-way quantum communication complexity, with applications to cryptography. In David S. Johnson and Uriel Feige, editors, *39th Annual ACM Symposium on Theory of Computing*, pages 516–525, San Diego, CA, USA, June 11–13, 2007. ACM Press.

[GKT26] Aditya Gulati, Dakshita Khurana, and Kabir Tomer. How to build pseudorandom unitaries in microcrypt, 2026.

[GKW26] David Gosset, Robin Kothari, and Kewen Wu. Quantum state preparation with optimal T-count. *Quantum*, 10:2168, 2026. Full version: arXiv:2411.04790, version 3.

[GMMY24] Eli Goldin, Tomoyuki Morimae, Saachi Mutreja, and Takashi Yamakawa. CountCrypt: Quantum cryptography between QCMA and PP, 2024. arXiv:2410.14792, version 3, 6 October 2025.

[Gol01] Oded Goldreich. *Foundations of Cryptography*. Cambridge University Press, 2001.

[Gol04] Oded Goldreich. *Foundations of Cryptography*. Cambridge University Press, 2004.

[GZ20] Marios Georgiou and Mark Zhandry. Unclonable decryption keys. Cryptology ePrint Archive, Paper 2020/877, 2020. Revised 29 July 2020.

[Hås86] Johan Håstad. Computational limitations for small depth circuits, 1986. Book manuscript based on the author's MIT Ph.D. thesis.

[HN22] Shuichi Hirahara and Mikito Nanashima. Finding errorless pessiland in error-prone heuristica. In *37th Computational Complexity Conference (CCC 2022)*, volume 234 of *Leibniz International Proceedings in Informatics*, pages 25:1–25:28, 2022.

[Imp11] Russell Impagliazzo. Relativized separations of worst-case and average-case complexities for NP. In *2011 IEEE 26th Annual Conference on Computational Complexity*, pages 104–114, 2011.

[JLS18] Zhengfeng Ji, Yi-Kai Liu, and Fang Song. Pseudorandom quantum states. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part III*, volume 10993 of *Lecture Notes in Computer Science*, pages 126–152, Santa Barbara, CA, USA, August 19–23, 2018. Springer, Cham, Switzerland.

[Kel21] Zander Kelley. An improved derandomization of the switching lemma. https://eccc.weizmann.ac.il/report/2020/182/revision/1/download, 2021. Electronic Colloquium on Computational Complexity, TR20-182, revision 1.

[KNSY26] Fuyuki Kitagawa, Ryo Nishimaki, Yuki Shirakawa, and Takashi Yamakawa. Trapdoor functions with secure key leasing and copy protection, 2026.

[KOW26] Robin Kothari, Ryan O'Donnell, and Kewen Wu. No exponential quantum speedup for SIS$^\infty$ anymore. In *Proceedings of the 58th Annual ACM Symposium on Theory of Computing*, pages 101–105. ACM, 2026.

[KQST23] William Kretschmer, Luowen Qian, Makrand Sinha, and Avishay Tal. Quantum cryptography in algorithmica, 2023. arXiv:2212.00879, version 2, 2 May 2023.

[KQT24] William Kretschmer, Luowen Qian, and Avishay Tal. Quantum-computable one-way functions without one-way functions, 2024. arXiv:2411.02554, version 1, 4 November 2024.

[Kre21] William Kretschmer. Quantum pseudorandomness and classical complexity. In *16th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2021)*, volume 197 of *Leibniz International Proceedings in Informatics*, pages 2:1–2:20, 2021. Full version: arXiv:2103.09320, version 5, 18 September 2024.

[KT24] Dakshita Khurana and Kabir Tomer. Commitments from quantum one-wayness. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, 2024. Full version: arXiv:2310.11526, version 4.

[Liu23] Qipeng Liu. Non-uniformity and quantum advice in the quantum random oracle model. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023, Part I*, volume 14004 of *Lecture Notes in Computer Science*, pages 117–143, Lyon, France, April 23–27, 2023. Springer, Cham, Switzerland.

[LLPY24] Xingjian Li, Qipeng Liu, Angelos Pelecanos, and Takashi Yamakawa. Classical vs quantum advice and proofs under classically-accessible oracle. In Venkatesan Guruswami, editor, *ITCS 2024: 15th Innovations in Theoretical Computer Science Conference*, volume 287, pages 72:1–72:19, Berkeley, CA, USA, January 30 – February 2, 2024. Leibniz International Proceedings in Informatics (LIPIcs).

[LLQZ22] Jiahui Liu, Qipeng Liu, Luowen Qian, and Mark Zhandry. Collusion resistant copy-protection for watermarkable functionalities. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022: 20th Theory of Cryptography Conference, Part I*, volume 13747 of *Lecture Notes in Computer Science*, pages 294–323, Chicago, IL, USA, November 7–10, 2022. Springer, Cham, Switzerland.

[LMY25] Jiahui Liu, Saachi Mutreja, and Henry Yuen. QMA vs QCMA and pseudorandomness. In Michal Koucký and Nikhil Bansal, editors, *57th Annual ACM Symposium on Theory of Computing*, pages 1520–1531, Prague, Czechia, June 23–27, 2025. ACM Press.

[Mah18] Urmila Mahadev. Classical verification of quantum computations. In Mikkel Thorup, editor, *59th Annual Symposium on Foundations of Computer Science*, pages 259–267, Paris, France, October 7–9, 2018. IEEE Computer Society Press.

[MH25] Fermi Ma and Hsin-Yuan Huang. How to construct random unitaries. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pages 806–809. ACM, 2025.

[MSY25] Tomoyuki Morimae, Yuki Shirakawa, and Takashi Yamakawa. Cryptographic characterization of quantum advantage. In Michal Koucký and Nikhil Bansal, editors, *57th Annual ACM Symposium on Theory of Computing*, pages 1863–1874, Prague, Czechia, June 23–27, 2025. ACM Press.

[MY22] Tomoyuki Morimae and Takashi Yamakawa. Quantum commitments and signatures without one-way functions. In *Advances in Cryptology – CRYPTO 2022*, volume 13507 of *Lecture Notes in Computer Science*, pages 269–295. Springer, 2022.

[MY24] Tomoyuki Morimae and Takashi Yamakawa. One-wayness in quantum cryptography. In *19th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2024)*, volume 310 of *Leibniz International Proceedings in Informatics*, pages 4:1–4:21, 2024. Full version: arXiv:2210.03394, version 3.

[Nay99] Ashwin Nayak. Optimal lower bounds for quantum automata and random access codes. In *40th Annual Symposium on Foundations of Computer Science*, pages 369–376, 1999. Full version: arXiv:quant-ph/9904093, version 3.

[NC10] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

[NN24] Anand Natarajan and Chinmay Nirkhe. A distribution testing oracle separation between QMA and QCMA. *Quantum*, 8:1377, 2024.

[RK11] Oded Regev and Bo'az Klartag. Quantum one-way communication can be exponentially stronger than classical communication. In Lance Fortnow and Salil P. Vadhan, editors, *43rd Annual ACM Symposium on Theory of Computing*, pages 31–40, San Jose, CA, USA, June 6–8, 2011. ACM Press.

[RST15] Benjamin Rossman, Rocco A. Servedio, and Li-Yang Tan. An average-case depth hierarchy theorem for Boolean circuits. In Venkatesan Guruswami, editor, *56th Annual Symposium on Foundations of Computer Science*, pages 1030–1048, Berkeley, CA, USA, October 17–20, 2015. IEEE Computer Society Press.

[RT22] Ran Raz and Avishay Tal. Oracle separation of BQP and PH. *Journal of the ACM*, 69(4), 2022.

[Sat] Or Sattath. Microcrypt Zoo. https://sattath.github.io/microcrypt-zoo/. Accessed 23 October 2026.

[SLSB19] Yuval R. Sanders, Guang Hao Low, Artur Scherer, and Dominic W. Berry. Black-box quantum state preparation without arithmetic. *Physical Review Letters*, 122(2):020502, 2019. Full version: arXiv:1807.03206, version 2.

[Sto83] Larry J. Stockmeyer. The complexity of approximate counting (preliminary version). In *15th Annual ACM Symposium on Theory of Computing*, pages 118–126, Boston, MA, USA, April 25–27, 1983. ACM Press.

[Tan19] Ewin Tang. A quantum-inspired classical algorithm for recommendation systems. In Moses Charikar and Edith Cohen, editors, *51st Annual ACM Symposium on Theory of Computing*, pages 217–228, Phoenix, AZ, USA, June 23–26, 2019. ACM Press.

[Wat00] John Watrous. Succinct quantum proofs for properties of finite groups. In *41st Annual Symposium on Foundations of Computer Science*, pages 537–546, Redondo Beach, CA, USA, November 12–14, 2000. IEEE Computer Society Press.

[YZ24] Takashi Yamakawa and Mark Zhandry. Verifiable quantum advantage without structure. *Journal of the ACM*, 71(3), 2024.

[Zha12a] Mark Zhandry. How to construct quantum random functions, 2012. Electronic Colloquium on Computational Complexity, TR12-140, revision 2.

[Zha12b] Mark Zhandry. How to construct quantum random functions. In *53rd Annual Symposium on Foundations of Computer Science*, pages 679–687, New Brunswick, NJ, USA, October 20–23, 2012. IEEE Computer Society Press.

[Zha19] Mark Zhandry. Quantum lightning never strikes the same state twice. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019, Part III*, volume 11478 of *Lecture Notes in Computer Science*, pages 408–438, Darmstadt, Germany, May 19–23, 2019. Springer, Cham, Switzerland.

[Zha25] Mark Zhandry. Toward separating QMA from QCMA with a classical oracle. In Raghu Meka, editor, *ITCS 2025: 16th Innovations in Theoretical Computer Science Conference*, volume 325, pages 95:1–95:19, New York, NY, USA, January 7–10, 2025. Leibniz International Proceedings in Informatics (LIPIcs).

# Appendices

# Appendix I: Deferred Proofs

## A Proofs of the Quantum Query and Encoding Bounds

In this appendix, we prove the coherent implementation bound and the random-function bound used in BB-QMA soundness.

### A.1 Proof of coherent classical computation (Lemma 6)

We now prove the coherent implementation guarantee of Lemma 6.

*Proof of Lemma 6.* Run $T$ reversibly while storing its intermediate computation, XOR its output into the response bit, and run the computation backward. The working registers return to their initial state, so the operation is the required coherent map. Each of the at most $h$ oracle queries is made once in each direction, giving the bound $2h$. □

### A.2 Proof of the random-function bound (Lemma 12)

We prove Lemma 12, which bounds the probability that a random Boolean function admits a successful quantum witness, using the following random-access-code bound.

**Lemma 21** (Random access codes [Nay99])**.** *Let $x \mapsto \sigma_x$ encode $x \in \{0,1\}^N$ into $m$ qubits, and let $M_1, \ldots, M_N$ be binary measurements, where $N \geq 1$ and $m \geq 0$ are integers. Then*

$$\Pr_{x \leftarrow \{0,1\}^N,\, i \leftarrow [N]}[M_i(\sigma_x) = x_i] \leq \frac{1}{2} + \sqrt{\frac{m}{N}}.$$

The proof of Lemma 21 appears in Appendix A, after the proof of Lemma 12.

We now prove Lemma 12 by constructing a random access code from a witness accepted on a random Boolean function.

**Constructing the code.** Let $N = 2^\lambda$ and define

$$\mathcal{G} = \left\{ B : \exists \sigma, \Pr_{y \leftarrow \{0,1\}^\lambda,\, v \leftarrow A(\sigma, y)}[v \neq \perp \wedge B(v,y) = 1] > \frac{2}{3} \right\}, \qquad \gamma = \Pr_B[B \in \mathcal{G}].$$

For each $B \in \mathcal{G}$, choose an $a$-qubit state $\sigma_B$ whose acceptance probability exceeds $2/3$. To encode a uniform string using these states, independently sample

$$Z : \Sigma_\lambda^{q_\lambda} \times \{0,1\}^\lambda \to \{0,1\} \text{ uniformly}, \qquad x = (x_y)_{y \in \{0,1\}^\lambda} \leftarrow \{0,1\}^N,$$

and put

$$B(v,y) = Z(v,y) \oplus x_y.$$

For each fixed $x$, $B$ is uniform. Both the encoder and the recovery algorithm receive $Z$.

**Encoding $x$ and recovering $x_y$**

**Encoding.** Output $(1, \sigma_B)$ if $B \in \mathcal{G}$, and $(0, |0^a\rangle)$ otherwise.

**Recovery.** On input $y$ and an encoding $(e, \sigma)$:

1. If $e = 0$, output a uniform bit and terminate.
2. Run $v \leftarrow A(\sigma, y)$, then output a uniform bit if $v = \perp$ and $1 \oplus Z(v, y)$ otherwise.

**Bounding the recovery probability.** For $B \in \mathcal{G}$, every pair with $B(v, y) = 1$ recovers $x_y = 1 \oplus Z(v, y)$, giving success at least $2/3$ over $y$ and $A$, whereas for $B \notin \mathcal{G}$ recovery succeeds with probability $1/2$. Thus uniformity of $B$ gives average success over $Z, x$ at least

$$\frac{1-\gamma}{2} + \frac{2\gamma}{3} = \frac{1}{2} + \frac{\gamma}{6}.$$

Some fixed Boolean function $Z$ attains at least this average success. Fixing it makes the construction an $(a+1)$-qubit random access code for a uniform $N$-bit string, so Lemma 21 gives

$$\frac{1}{2} + \frac{\gamma}{6} \leq \Pr[\text{recovery is correct}] \leq \frac{1}{2} + \sqrt{\frac{a+1}{N}}.$$

Therefore $\gamma \leq 6\sqrt{(a+1)2^{-\lambda}}$, as required.

For completeness, we now derive the random access code bound from mutual information.

*Proof of Lemma 21.* Let $X$ be uniform on $\{0,1\}^N$, let $Q$ contain $\sigma_X$, and put $p_i = \Pr[M_i(\sigma_X) = X_i]$ and $p = N^{-1}\sum_i p_i$. Write $Y_i$ for the outcome of measurement $M_i$. Since measuring $Q$ cannot increase mutual information, Fano's inequality gives $I(X_i : Q) \geq I(X_i : Y_i) \geq 1 - h_2(p_i)$. Combining this with the $m$-qubit entropy bound and independence of the $X_i$, we obtain

$$m \geq I(X : Q) \geq \sum_{i=1}^{N} I(X_i : Q) \geq \sum_{i=1}^{N} (1 - h_2(p_i)) \geq N(1 - h_2(p)),$$

where $I$ denotes quantum mutual information and $h_2$ is binary entropy, with the last inequality following from concavity of $h_2$. Since $1 - h_2(p) \geq (p - 1/2)^2$, we conclude that $p \leq 1/2 + \sqrt{m/N}$. □

# B Proof of the Codeword-Hash Guarantees (Lemma 7)

In this appendix, we prove Lemma 7 using the multiplicity code of [BHV26].

## B.1 The multiplicity code

We first specify the code family whose properties are stated in Lemma 7. Let $q_\lambda$ be the smallest prime greater than $\lambda^5$, put $\Sigma_\lambda = \mathbb{F}_{q_\lambda}^{\lambda}$, and fix an ordering $\alpha_1, \dots, \alpha_{q_\lambda}$ of $\mathbb{F}_{q_\lambda}$. For $f \in \mathbb{F}_{q_\lambda}[T]$ with $\deg f < \lambda^3$, define

$$v(f)_i = \big([T^j]f(\alpha_i + T)\big)_{j=0}^{\lambda-1}, \qquad C_\lambda = \{(v(f)_1, \dots, v(f)_{q_\lambda}) : \deg f < \lambda^3\}.$$

This is the multiplicity code $\mathsf{Mult}_{\lambda,\mathbb{F}_{q_\lambda},\lambda^3}$ of [BHV26]. Its generator matrix and a left inverse are computable in polynomial time, giving efficient encoding and membership testing. Write $|C_\lambda\rangle = |C_\lambda|^{-1/2} \sum_{v \in C_\lambda} |v\rangle$ for the uniform codeword state. Preparing uniform field coefficients, applying the linear encoder, and uncomputing the coefficients with the left inverse prepares the uniform codeword state, with the approximation guarantee in Lemma 7. Encode each field element using $\lceil \log_2 q_\lambda \rceil$ bits, so $b_\lambda = \lambda \lceil \log_2 q_\lambda \rceil$ and $a_\lambda = 2q_\lambda \lambda \lceil \log_2 q_\lambda \rceil$.

## B.2 Expansion and decoding bounds

We use the following expansion and decoding bounds to establish the properties of this code family.

**Lemma 22** (Multiplicity-code expansion [BHV26])**.** *For all sufficiently large $\lambda$, every set $W \subseteq C_\lambda$ with $|W| \le 2^\lambda$ satisfies*

$$\big|\{(i, v_i) : i \in [q_\lambda],\ v \in W\}\big| \ge (q_\lambda - 4\lambda^4)|W|.$$

*Proof.* Apply the expansion bound of [BHV26] with degree bound $k = \lambda^3$, multiplicity $s = \lambda$ and set-size bound $K = 2^{\lambda+1}$. Since $\lambda^5 < q_\lambda \le 2\lambda^5$, its hypotheses $15 \le \lambda + 1 \le \lambda^3 \le q_\lambda$ hold, and its expansion factor is

$$q_\lambda - \frac{\lambda^3(\lambda+1)}{2}(q_\lambda 2^{\lambda+1})^{1/(\lambda+1)} = q_\lambda - \lambda^3(\lambda+1)q_\lambda^{1/(\lambda+1)} \ge q_\lambda - 4\lambda^4.$$

The formal derivatives in that bound are nonzero scalar multiples of the Hasse derivatives defining $C_\lambda$, so they count the same distinct coordinate–symbol pairs. □

The decoding bound below holds simultaneously for every target.

**Lemma 23** (Simultaneous biased-hash decoding)**.** *For the code $C_\lambda$ and predicate-preimage state $|\Psi_H\rangle$, independently sample $H_i(u) \leftarrow \mathrm{Bernoulli}(\lambda^{-4})$ for every $i \in [q_\lambda]$ and $u \in \Sigma_\lambda$. There is a uniform polynomial-time oracle-free quantum decoder $\mathsf{BiasYZ}$ such that, for all sufficiently large $\lambda$, with probability at least $1 - 2^{-\lambda}$ over $H$, every $H_i^{-1}(b)$ is nonempty and*

$$\forall y \in \{0,1\}^\lambda : \qquad \Pr_{v \leftarrow \mathsf{BiasYZ}(y, |\Psi_H\rangle \otimes |C_\lambda\rangle)}[(v, y) \in \mathcal{P}_H] \ge 1 - 2^{-\lambda}.$$

*Proof.* For this code and hash distribution, [BHV26, Corollary 5.24] gives the displayed simultaneous decoding guarantee. Its successful event excludes empty preimages, since the decoder returns $\perp$ if any preimage is empty ([BHV26, Definition 5.21 and Figure 1]). On this event its advice state is exactly $|\Psi_H\rangle \otimes |C_\lambda\rangle$, proving the claim. □

On input $(y, \sigma)$, the decoder $\mathsf{Decode}$ first prepares the public state $|C_\lambda\rangle$ to trace-distance error $2^{-\lambda}$ using the linear encoder above. It then runs $\mathsf{BiasYZ}$ on $y$ and the combined state, returning its output $v$ if $v \in C_\lambda$ and $\perp$ otherwise. The preparation error changes any acceptance probability by at most $2^{-\lambda}$, so the decoding error for $|\Psi_H\rangle$ is at most $2^{1-\lambda}$. This is the bound used in the definition of good hashes (Definition 16).

### B.3 Proof of the codeword-hash guarantees (Lemma 7)

We now prove the codeword-hash guarantees (Lemma 7).

**Bounding joint inclusion.** Fix a constant $d \geq 1$ and predicates $H_1, \ldots, H_\lambda$, and let $J_j$ and $P_j(H)$ be the suffix-coordinate groups and their constraint sets (Definition 17). For the joint-inclusion bound, fix $j \in [d]$ and distinct oracle inputs $u_r = (v_r, y_r)$, $r \in [t]$. If some $v_r \notin C_\lambda$, the joint-inclusion probability is zero, so we may assume that every $v_r$ is a codeword. If two inputs have the same codeword, their targets differ, so they cannot both belong to $P_j(H)$. We may therefore assume that the codewords are distinct. By Lemma 22, they contain at least $(q_\lambda - 4\lambda^4)t$ distinct coordinate–symbol pairs. We need only those in $J_j$, so write

$$S_j = \{(i, (v_r)_i) : i \in J_j,\ r \in [t]\}.$$

Removing coordinates outside $J_j$ deletes at most $(q_\lambda - |J_j|)t$ pairs from the expansion bound. Since the groups $J_1, \ldots, J_d$ partition the suffix coordinates and differ in size by at most one, this gives

$$|S_j| \geq (|J_j| - 4\lambda^4)t \geq \left(\frac{q_\lambda - \lambda}{d} - 1 - 4\lambda^4\right) t \geq \frac{\lambda^5 t}{2d}.$$

Simultaneous membership requires all predicate values indexed by $S_j$ to be zero. Under $H \leftarrow \mathcal{H}_\lambda$, these are independent bits of mean $\lambda^{-4}$, giving

$$\Pr[u_1, \ldots, u_t \in P_j(H)] \leq (1 - \lambda^{-4})^{|S_j|} \leq \exp\left(-\frac{\lambda t}{2d}\right) \leq 2^{-\lambda t/(8d)}.$$

With the prefix fixed, each $P_j(H)$ depends on a disjoint group of suffix predicates, giving mutual independence, while their intersection imposes every suffix constraint and is therefore $\mathcal{P}_H$.

**Fixing the prefix predicates.** When all predicate values are sampled independently with mean $\lambda^{-4}$, Lemma 23 and the definition of $\mathsf{Decode}$ give $\Pr_H[H \in \mathsf{Good}_\lambda] \geq 1 - 2^{-\lambda}$. Averaging over the first $\lambda$ predicates gives a fixed choice of $H_1, \ldots, H_\lambda$ for which

$$\Pr_{H \leftarrow \mathcal{H}_\lambda}[H \in \mathsf{Good}_\lambda] \geq 1 - 2^{-\lambda}.$$

The decoder is uniform and runs in polynomial time by Lemma 23, and neither it nor the chosen prefix depends on the partition into $d$ groups. Together with the code-state preparation in Appendix B.1, this proves all three parts of the lemma.

# C Correctness of the Signature Constructions

We prove joint correctness for the adaptive honest operations of CSig and QSig (Lemma 24), including the returned keys and signatures. Use the signature oracle $\mathcal{O} = (\mathcal{O}_{\mathsf{base}}, N, A)$ (Definition 33) with any independent additional base components. The ideal honest experiment starts with uniform $s \in \{0,1\}^n$ and signing state $|\Psi_{H_s}\rangle$, and returns

| | CSig | QSig |
|---|---|---|
| signature on $m$ | $\left\|\Psi_{H_{s,m}}\right\rangle$ | $F_s^{\mathrm{sig}}(m)$ |
| verification-key copy | $s$ | $(s, \left\|\Psi_{H_s^V}\right\rangle)$. |

Each ideal operation leaves its predicate-preimage state unchanged, and verification uses an exact query to the encoded function in place of EvalHash. Let $\Pi$ be generation followed by at most $q(n)$ adaptive honest signing, verification-key-generation and verification calls. For any conditional density matrix $\sigma_W^{\mathcal{O}}$, let $\Pi_{\mathrm{act}}^{\mathcal{O}}(\sigma_W^{\mathcal{O}})$ and $\Pi_{\mathrm{id}}^{\mathcal{O}}(\sigma_W^{\mathcal{O}})$ be the joint output density matrices obtained by these actual and ideal operations, and put

$$\Delta_\Pi(\mathcal{O}) = \tfrac{1}{2}\|\Pi_{\mathrm{act}}^{\mathcal{O}}(\sigma_W^{\mathcal{O}}) - \Pi_{\mathrm{id}}^{\mathcal{O}}(\sigma_W^{\mathcal{O}})\|_1.$$

Both outputs include $W$ and all keys and signatures. Thus averaging $\Delta_\Pi(\mathcal{O})$ also bounds the joint distance with the sampled oracle included as a classical output. The following lemma bounds this error simultaneously for all such honest executions.

**Lemma 24** (Signature correctness)**.** *There are a polynomial $p$ and sets $\mathcal{G}_n^{\mathrm{sig}}$ such that, for every polynomial $q \geq 1$, every honest execution $\Pi$ above and sufficiently large $n$,*

$$\Pr_{\mathcal{O}}[\mathcal{O} \notin \mathcal{G}_n^{\mathrm{sig}}] \leq 2^{-n},$$
$$\Delta_\Pi(\mathcal{O}) \leq q(n)p(n)2^{-n} \quad (\mathcal{O} \in \mathcal{G}_n^{\mathrm{sig}}).$$

The proof of Lemma 24 appears in Appendix C.1. Observe that the ideal experiment always accepts, so the same error bounds honest rejection, and averaging the good event gives $\mathbb{E}_{\mathcal{O}}\Delta_\Pi(\mathcal{O}) \leq q(n)p(n)2^{-n} + 2^{-n} = 2^{-\Omega(n)}$.

## C.1 Proof of signature correctness

We now prove Lemma 24. The signing state $|\Psi_{H_s}\rangle$ evaluates $P_s$ through $T_s = T_{H_s,B_s,P_s}$. For an issuance label $t$, the function $P_s(1,t,u) = E_{H_{s,t}}(u)$ supplies the counts and permutations

$$D = |\Sigma_\kappa|, \qquad k_i = |H_i^{-1}(1)|, \qquad I_{i,1} = [0,k_i), \quad I_{i,0} = [k_i, D),$$

where $H = H_{s,t}$ and $\pi_i : [0,D) \to \Sigma_\kappa$ lists $H_i^{-1}(1)$ before $H_i^{-1}(0)$ (Equation (74)). Thus $\pi_i$ maps a uniform rank in $I_{i,b}$ to a uniform preimage of $b$, while $\pi_i^{-1}$ erases the rank coherently. The following algorithms evaluate these functions and issue the resulting state.

**Reading signing values and issuing predicate-preimage states**

$\underline{\mathsf{Read}(s, \mathsf{R}_{\mathsf{sk}}, u, b)}$

Run $\mathsf{EvalHash}^{T_s}$ with the predicate-preimage state register, input $u$ and response $b$, and return all registers.

$\underline{\mathsf{Issue}(s, \mathsf{R}_{\mathsf{sk}}, t)}$

1. Use $\mathsf{Read}$ with zero response bits and measure those responses to obtain the counts $k_i$ for $H_{s,t}$. If a count is outside $[0, D]$ or either preimage is empty, return the default state $|0^{a_\kappa}\rangle$ and the signing register.
2. For each interval $I_{i,b} = [a, a + M)$, put $\ell = \lceil \log_2 M \rceil$. Prepare $2^{-\ell/2} \sum_{j=0}^{2^\ell - 1} |j\rangle$, measure whether $j < M$, and add $a$ on success. Retry a failed test polynomially many times, using gate accuracy that makes the total failure and approximation error at most $2^{-\kappa-1}$ for the intervals and encodings. Return the default state if an interval preparation fails.
3. Use $\mathsf{Read}$ to query $\pi_i$ and $\pi_i^{-1}$:

$$\sum_{j \in I_{i,b}} |j\rangle\,|0\rangle \longmapsto \sum_{j \in I_{i,b}} |j\rangle\,|\pi_i(j)\rangle \longmapsto \sum_{j \in I_{i,b}} |0\rangle\,|\pi_i(j)\rangle\,.$$

4. Return the prepared preimage registers and the signing register, discarding workspace.

We prove the joint guarantee by bounding issuance and verification at each fixed good oracle, then averaging over the oracle sample.

Issuance requires a joint bound for several encoded calls. We use the following bound for that operation.

**Claim 6.** *Fix $H_s \in \mathsf{Good}_{\kappa_M}$ and an issuance label $t$. Starting with $|\Psi_{H_s}\rangle$ and any auxiliary system $W$, let $\rho_{\mathrm{iss}}$ be the joint output of $\mathsf{Issue}(s, |\Psi_{H_s}\rangle, t)$, and let $\rho_{\mathrm{id}}$ instead append $|\Psi_{H_{s,t}}\rangle$ while leaving the signing state and $W$ unchanged. Then*

$$\tfrac{1}{2} \left\| \rho_{\mathrm{iss}} - \rho_{\mathrm{id}} \right\|_1 \le 2^{-\kappa}.$$

The proof of Claim 6 appears in Appendix C.1.1.

We first use this bound to prove Lemma 24.

**A simultaneous good-oracle event.** Let $\tau^{\mathcal{O}}$ be the output of $\mathsf{Prepare}^{R_M, S_M}(1^{n_M})$ and put

$$e_K(\mathcal{O}) = \tfrac{1}{2} \left\| \tau^{\mathcal{O}} - |\Omega_M\rangle\langle\Omega_M| \right\|_1 .$$

We now require all signing hashes and this preparation to be good:

$$\mathcal{H}_n^{\mathrm{sig}} = \{H_s \in \mathsf{Good}_{\kappa_M},\ H_{s,t} \in \mathsf{Good}_\kappa \text{ for every } s, t\},$$
$$\mathcal{G}_n^{\mathrm{sig}} = \mathcal{H}_n^{\mathrm{sig}} \cap \{e_K(\mathcal{O}) \le 2^{-n}\}.$$

The state-preparation bound (Lemma 17) gives $\mathbb{E}_{\mathcal{O}} e_K \le 2^{-n_M}$. There are $2^n$ master hashes and $2^{2n} + 2^n$ child hashes, so the good-hash bound (Equation (19)) and Markov's inequality imply

$$\Pr_{\mathcal{O}}[\mathcal{O} \notin \mathcal{G}_n^{\mathrm{sig}}] \le 2^{n - \kappa_M} + 2^{2n+1-\kappa} + 2^{n-n_M} \le 2^{-n}.$$

**The honest-operation hybrids.** Fix $\mathcal{O} \in \mathcal{G}_n^{\mathrm{sig}}$ and let $\mathsf{H}_0$ run the actual execution $\Pi$. In $\mathsf{H}_1$, change only indexed-key preparation in generation: prepare $|\Omega_M\rangle$, measure its serial, yielding $s$ and $|\Psi_{H_s}\rangle$. During the remaining execution, list in chronological order each issuance, signing-bit $\mathsf{Read}$ and verification $\mathsf{EvalHash}$ call, including the initial verification-key issuance in $\mathsf{QSig.Gen}$. Since verification uses at most $r = 3n$ checks, there are at most $T = (q+1) + qn + qr$ such operations. Append identity operations when necessary to use this common bound.

For $j = 1, \ldots, T$, define $\mathsf{H}_{j+1}$ from $\mathsf{H}_1$ by running the first $j$ operations ideally and all later operations as in the actual algorithms, with these ideal operations:

1. On issuance label $t$, return $\left|\Psi_{H_{s,t}}\right\rangle$, leaving the signing state unchanged.
2. On a signing-bit request $(m, i)$, return $F_s^{\mathrm{sig}}(m)_i$, leaving the signing state unchanged.
3. On a verification call for $T_{H,B,\mathfrak{F}}$, query $\mathfrak{F}$ directly and leave $|\Psi_H\rangle$ unchanged.

Each experiment uses the fixed $\mathcal{O}$, chooses its adaptive requests from its own preceding outputs, and returns all registers produced by $\Pi$ jointly with $W$. Write $\rho_j^{\mathcal{O}}$ for this output. The preparation error on $\mathcal{G}_n^{\mathrm{sig}}$ gives

$$\tfrac{1}{2}\left\|\rho_0^{\mathcal{O}} - \rho_1^{\mathcal{O}}\right\|_1 \le 2^{-n}.$$

Before the operation changed between $\mathsf{H}_j$ and $\mathsf{H}_{j+1}$, every honest operation is ideal. The master state therefore equals $|\Psi_{H_s}\rangle$, and each issued child state equals $\left|\Psi_{H_{s,t}}\right\rangle$ when used by that operation. The issuance bound (Claim 6) and encoded evaluation (Theorem 14) thus apply with respective errors

$$\epsilon(\mathsf{op}) = \begin{cases} 2^{-\kappa}, & \mathsf{op} \text{ is issuance,} \\ 2\sqrt{\delta_{\mathrm{dec}}(\kappa_M)}, & \mathsf{op} \text{ is a signing-bit query,} \\ 2\sqrt{\delta_{\mathrm{dec}}(\kappa)}, & \mathsf{op} \text{ is a verification query,} \\ 0, & \mathsf{op} \text{ is an identity operation.} \end{cases}$$

The common ideal prefix has the same distribution as in the fully ideal experiment. The identical remaining operations cannot increase trace distance, so averaging the next operation $\mathsf{op}_j$ gives

$$\tfrac{1}{2}\left\|\rho_j^{\mathcal{O}} - \rho_{j+1}^{\mathcal{O}}\right\|_1 \le \mathbb{E}_{\mathrm{ideal}}\epsilon(\mathsf{op}_j) \qquad (1 \le j \le T).$$

In $\mathsf{H}_{T+1}$, every $\mathsf{CSig}$ check queries the constant-one function, while every $\mathsf{QSig}$ check satisfies

$$G_s(m, F_s^{\mathrm{sig}}(m)) = 1.$$

Thus $\mathsf{H}_{T+1}$ is the ideal honest experiment and always accepts. Every execution contains at most $q+1$ issuances, $qn$ signing-bit queries and $qr$ verification queries, so summing the bounds gives

$$\begin{aligned} \Delta_\Pi(\mathcal{O}) &\le \sum_{j=0}^{T} \tfrac{1}{2}\left\|\rho_j^{\mathcal{O}} - \rho_{j+1}^{\mathcal{O}}\right\|_1 \\ &\le 2^{-n} + (q+1)2^{-\kappa} + 2qn\sqrt{\delta_{\mathrm{dec}}(\kappa_M)} + 2qr\sqrt{\delta_{\mathrm{dec}}(\kappa)} \le q(n)p(n)2^{-n}. \end{aligned}$$

The distance also bounds the probability of any honest rejection. Averaging the good-oracle event and its complement proves

$$\mathbb{E}_{\mathcal{O}}\Delta_\Pi(\mathcal{O}) \le q(n)p(n)2^{-n} + \Pr\left[\mathcal{O} \notin \mathcal{G}_n^{\mathrm{sig}}\right] = 2^{-\Omega(n)}.$$

### C.1.1 The issuance error

We now prove Claim 6. Replacing the encoded calls by exact queries to $P_s$ leaves only the interval preparation and permutation computations in $\mathsf{Issue}$ to analyze.

For a nonempty preimage, write

$$|I_{i,b}\rangle = |I_{i,b}|^{-1/2} \sum_{j \in I_{i,b}} |j\rangle\,, \qquad \left|H_i^{-1}(b)\right\rangle = |H_i^{-1}(b)|^{-1/2} \sum_{z \in H_i^{-1}(b)} |z\rangle\,.$$

Because $\pi_i : I_{i,b} \to H_i^{-1}(b)$ is a bijection, the forward and inverse permutation queries implement

$$|I_{i,b}\rangle\,|0\rangle \longmapsto |I_{i,b}|^{-1/2} \sum_{j \in I_{i,b}} |j\rangle\,|\pi_i(j)\rangle \longmapsto |0\rangle\,\left|H_i^{-1}(b)\right\rangle.$$

Taking the tensor product over all $i, b$ therefore gives exactly $\left|\Psi_{H_{s,t}}\right\rangle$, with the prescribed default state when a preimage is empty (Definition 15).

Fix $H_s \in \mathsf{Good}_{\kappa_M}$, and let $q_0(n)$ bound the encoded bit queries in one issuance. Define three experiments starting with the exact signing state $|\Psi_{H_s}\rangle$ and the same issuance label:

- $\mathsf{I}_0$ Run $\mathsf{Issue}(s, |\Psi_{H_s}\rangle\,, t)$ with $T_s$.
- $\mathsf{I}_1$ Run the same algorithm, replacing each $\mathsf{Read}(s, \mathsf{R}_{\mathsf{sk}}, u, b)$ by $|u, b\rangle \mapsto |u, b \oplus P_s(u)\rangle$ and leaving $\mathsf{R}_{\mathsf{sk}}$ unchanged.
- $\mathsf{I}_2$ Return $\left|\Psi_{H_{s,t}}\right\rangle\!\left\langle\Psi_{H_{s,t}}\right|$ together with the unchanged signing state and $W$.

Let $\rho_i$ be their joint outputs with $W$. To pass from $\mathsf{I}_0$ to $\mathsf{I}_1$, replace the $\mathsf{Read}$ calls in execution order. Every earlier replacement leaves $|\Psi_{H_s}\rangle$ unchanged, so encoded evaluation (Theorem 14) gives the first estimate below. For the second, each interval trial succeeds with probability $M/2^\ell \geq 1/2$, and the chosen trial count and gate accuracy bound the total error in interval preparation and encoding:

$$\tfrac{1}{2}\,\|\rho_0 - \rho_1\|_1 \leq 2q_0(n)\sqrt{\delta_{\mathrm{dec}}(\kappa_M)} \leq 2^{-\kappa-1}, \qquad \tfrac{1}{2}\,\|\rho_1 - \rho_2\|_1 \leq 2^{-\kappa-1}.$$

The default output on failure is included in $\rho_0, \rho_1$, so no conditioning on success is needed. Thus one issuance starting with $|\Psi_{H_s}\rangle$ differs from its ideal operation by at most $2^{-\kappa}$, including the signing state and $W$.

# D Proof of the Ordered Switching Lemma (Lemma 25)

We prove a switching lemma that returns the label of the first matching database and permits independent answers with different biases.

**Lemma 25** (Ordered switching)**.** *Fix positive integers $k, h$, a nonnegative integer $m$, a finite domain $\mathcal{X} \subseteq \{0,1\}^*$, and an ordered collection $\mathcal{C} = (D_1, \dots, D_m)$ of databases $D_i \subseteq \mathcal{X} \times \{0,1\}$, each with at most $k$ entries and at most one entry for each input. For each $D_i$, fix an order of its entries and an arbitrary label $l_i$. For fixed parameters $0 < \beta \le 1/2$ and $0 \le \rho \le \beta/(8k)$, let $P \subseteq \mathcal{X}$ be random, with*

$$\Pr_P[S \subseteq P] \le \rho^{|S|} \quad \text{for every } S \subseteq \mathcal{X} \text{ of size at most } h + k - 1.$$

*Independently of $P$, sample*

$$B(u) \overset{\text{ind}}{\leftarrow} \text{Bernoulli}(p_u) \quad (u \in \mathcal{X}), \qquad p_u \in [\beta, 1-\beta].$$

*To identify the first matching label, define*

$$j(\mathcal{O}) = \min\big(\{i \in [m] : \mathcal{O} \text{ agrees with } D_i\} \cup \{m+1\}\big), \qquad l_{m+1} = \perp.$$

*For $U = B|_{\mathcal{X}\setminus P}$, there is a fixed mapping $(P, U) \mapsto T_{P,U}$ with unrestricted local computation. Its output makes fewer than $h$ classical queries on every oracle, all in $P$, and satisfies*

$$\Pr_{P,B}\left[\forall \mathcal{O} \text{ agreeing with } U : \quad T_{P,U}^{\mathcal{O}} = l_{j(\mathcal{O})}\right] \ge 1 - 2(4k\rho/\beta)^h.$$

The proof of Lemma 25 appears in Appendix D.1. Observe that the correctness event depends only on $P, U$, since it requires agreement for every oracle extending $U$.

## D.1 Proof of ordered switching

We now prove Lemma 25. We construct an algorithm that tests the databases in order, stopping before its $h$th query, and bound the probability that this stopping rule changes any output.

**$T_{P,U}^{\mathcal{O}}$: first matching database**

Given $(P, U)$ and the ordered, labelled collection $\mathcal{C}$, the algorithm has classical access to $\mathcal{O}$. If it would make query $h$, it returns $\perp$ instead.

1. Initialize the answer database $D = \{(u, U(u)) : u \in \mathcal{X} \setminus P\}$.
2. Delete every $D_i$ containing an entry $(u, a)$ with $u \in \operatorname{dom} D$ and $D(u) \neq a$. If none remains, output $\perp$ and terminate. Otherwise, let $D_i$ be the first remaining database.
3. Set $Q = \operatorname{dom} D_i \setminus \operatorname{dom} D$. If $Q = \varnothing$, output $l_i$ and terminate. Otherwise, for each $u \in Q$ in the fixed entry order of $D_i$, query $\mathcal{O}(u)$ to obtain $a$ and set $D(u) = a$. Subject to the stopping rule, complete this batch even if an earlier answer contradicts $D_i$.
4. Repeat from Step 2.

To bound the probability of a long execution, we encode its queried positions and answers in a tuple.

**Definition 38** (Encoding a first-match execution)**.** *In an untruncated execution of $T_{P,U}$ that reaches query $h$, stop after the first batch containing that query. If the execution has $r$ batches and $s$ distinct queries, we define its* execution encoding *to be*

$$a = \big((s_i)_{i\le r}, (Q_i)_{i\le r}, (\alpha_i)_{i<r}\big), \qquad h \le s = \sum_{i=1}^{r} s_i \le h + k - 1,$$

*where $s_i$ is the positive length of batch $i$, $Q_i$ lists its queried positions in the selected database, and $\alpha_i$ lists its answers. The final batch's answers are omitted.*

For fixed $s$, there are at most $2^s k^s 2^s = (4k)^s$ encodings, accounting respectively for compositions of $s$, queried positions, and answers. The following bound does not include the database indices.

**Lemma 26** (Probability of a fixed execution encoding)**.** *Under the hypotheses of Lemma 25, fix $h \le s \le h + k - 1$ and an execution encoding $a$ of $s$ queries (Definition 38). Then*

$$\Pr_{P,B}[\exists \mathcal{O} \text{ agreeing with } U : \text{the untruncated execution has encoding } a] \le (\rho/\beta)^s.$$

The proof of Lemma 26 appears in Appendix D.2.

We now sum over the possible encodings to prove Lemma 25.

*Proof of Lemma 25.* The untruncated algorithm tests databases in order and returns $l_{j(\mathcal{O})}$, so truncation can fail only if some assignment to $P$ reaches query $h$. Counting encodings and applying Lemma 26 gives

$$\Pr[\text{some untruncated execution reaches query } h] \le \sum_{s=h}^{h+k-1} (4k\rho/\beta)^s \le 2(4k\rho/\beta)^h.$$

Outside this event, truncation changes no execution, proving simultaneous correctness on an event determined by $P, U$. □

## D.2 Bounding the probability of a long execution

We now prove Lemma 26 by recovering the queried inputs from an independently sampled Boolean function. Fix an execution encoding $a$ (Definition 38). We first define a decoder that uses a Boolean function $W$ to recover database indices, following Kelley's canonical switching argument [Kel21]. For a tuple $\ell = (\ell_1, \dots, \ell_r)$, let $S_{a,\ell}$ be the set of inputs at the positions $Q_i$ of $D_{\ell_i}$ specified by $a$.

$J_a(W)$

1. Enumerate the increasing tuples $1 \le \ell_1 < \cdots < \ell_r \le m$ in lexicographic order. Discard a tuple if a position in $Q_i$ is invalid for $D_{\ell_i}$ or if $|S_{a,\ell}| \ne s$.
2. For each remaining tuple, set $W' = W$ and process $i = 1, \dots, r$:
   - 2.1. If $W'(u) \ne b$ for some $(u, b) \in D_{\ell_i}$, discard this tuple and continue the enumeration in Step 1.
   - 2.2. If $i < r$, replace $W'(u)$ by the corresponding answer in $\alpha_i$ at each position of $Q_i$ in $D_{\ell_i}$.
3. Return the first tuple passing every test. If no tuple passes, return $\perp$.

The decoder uses only $W$ and the answers in $a$, so its selected inputs will be independent of $P$ whenever $W$ is. To obtain such a function while preserving the answers outside $P$, independently sample $X(u) \leftarrow \text{Bernoulli}(p_u)$ at each $u \in \mathcal{X}$ and set

$$W(u) = \begin{cases} X(u), & u \in P, \\ U(u), & u \notin P. \end{cases}$$

The next claim relates a long execution to the decoder's output.

**Claim 7.** *For fixed $P, U$, if some execution has encoding $a$, then*

$$\Pr_X[J_a(W) \neq \perp \ \wedge \ S_{a,J_a(W)} \subseteq P] \geq \beta^s.$$

*Proof.* Choose such an execution and write $\ell$ for its database indices. At batch $i$, the selected database $D_{\ell_i}$ agrees with all known answers, and its unknown entries are exactly those in $Q_i$. Consequently, the tests in Step 2 for $\ell$ require $s$ consistent literals, one at each input in $S_{a,\ell} \subseteq P$. Each literal holds independently with probability at least $\beta$.

When all these literals hold, $J_a(W) = \ell$. Indeed, a lexicographically earlier tuple first selects a database that the execution rejected using $U(u)$ or an answer from an earlier batch. The function $W'$ supplies that same answer when testing the earlier database, so the decoder rejects the earlier tuple as well. Thus all $s$ literals suffice to recover $S_{a,\ell}$, proving the claim. □

Now average the claim over $P, U$. Conditional on $P$, the bits of $W$ have the product distribution of $B$, so $W$ is independent of $P$. Whenever $J_a(W) \neq \perp$, its selected set $S_{a,J_a(W)}$ has size $s$ and is fixed by $W$, and hence the joint-inclusion hypothesis gives

$$\Pr_{P,W}[J_a(W) \neq \perp \ \wedge \ S_{a,J_a(W)} \subseteq P] \leq \rho^s.$$

Therefore the claim and the joint-inclusion bound give

$$\Pr_{P,B}[\text{some execution has encoding } a] \leq \beta^{-s} \Pr_{P,W}[J_a(W) \neq \perp \ \wedge \ S_{a,J_a(W)} \subseteq P] \leq (\rho/\beta)^s,$$

which proves Lemma 26.

# E Estimating Acceptance and Searching Classical Inputs

In this section, we use the recursive oracle $\mathcal{O} = (\mathcal{O}_{\mathsf{base}}, N, A)$ (Definition 26) to maximize acceptance over classical inputs and decide promise QCMA (Lemma 27).

For a valid length-$n$ description $a = \langle M, x\rangle$, where $M$ is a witnessless quantum circuit with classical input $x$, define

$$v_a = \Pr\big[M^{\mathcal{O}}(x) = 1\big], \qquad \widehat{v}_a = \frac{3}{n^4}\sum_{i=1}^{n^4} A(a,i) - 1, \tag{85}$$
$$\mathcal{E}_n = \{\forall \text{ valid } a \in \{0,1\}^n : |\widehat{v}_a - v_a| \le 3/n\}.$$

A test family is a polynomial-time map $a : \{0,1\}^\ell \to \{0,1\}^n$ to such padded pairs, with $n$, $\ell$ and evaluation time polynomial in the explicit input length.

**Lemma 27** (Estimation, search and collapse)**.** *Over any base distribution, there is a deterministic polynomial-time oracle algorithm* Search *such that, for every test family $a$,*

$$\Pr_{\mathcal{O}}[\neg\mathcal{E}_n] \le 2^{n+1}e^{-2n^2},$$
$$\mathcal{E}_n \quad\Longrightarrow\quad v_{a(\mathsf{Search}^{\mathcal{O}}(a))} \ge \max_{z\in\{0,1\}^\ell} v_{a(z)} - 6/n.$$

*Moreover, with probability one over $\mathcal{O}$,*

$$\mathsf{P}^{\mathcal{O}} = \mathsf{BQP}^{\mathcal{O}} = \mathsf{QCMA}^{\mathcal{O}} \qquad \textit{for languages and promise problems.}$$

The proof of Lemma 27 appears in Appendix E.1.

## E.1 Proof

We now prove Lemma 27. The recursive $N$ oracle lets a classical algorithm maximize the estimate $\widehat{v}_a$. A simultaneous bound on that estimate will then turn maximization into both approximate search and a decision procedure for promise QCMA.

**Step 1: Maximize the acceptance estimate.** Define

$$s(z) = \sum_{i=1}^{n^4} A(a(z), i), \qquad H(p,t) = \mathbf{1}[\exists z \in \{0,1\}^\ell : p \text{ is a prefix of } z,\ s(z) \ge t].$$

To compute $H(p,t)$, form the deterministic verifier $M_{\text{search}}$ with $a, n, \ell$ in its description. On ordinary input $(p,t)$ and witness $z \in \{0,1\}^\ell$, it rejects unless $p$ is a prefix of $z$, queries $A(a(z), i)$ for $i \in [n^4]$, and accepts exactly when their sum is at least $t$. Pad $\langle M_{\text{search}}, (p,t)\rangle$ to a polynomial length that bounds the verifier's size, witness length and query lengths, then query $N(\langle M_{\text{search}}, (p,t)\rangle)$. The recursive oracle (Definition 26) returns $H(p,t)$, since $M_{\text{search}}$ uses the same fixed $A$ as the definition of $s(z)$. The algorithm $\mathsf{Search}^{\mathcal{O}}(a)$ is then:

1. Binary-search $t \in \{0, \dots, n^4\}$ to find $s_* = \max\{t : H(\varnothing, t) = 1\}$.
2. Set $p = \varnothing$. While $|p| < \ell$, append 0 if $H(p0, s_*) = 1$, and 1 otherwise.
3. Return $z_* = p$.

By the definition of $s_*$, the initial prefix satisfies $H(\varnothing, s_*) = 1$. Each prefix step preserves this condition, since at least one of $p0, p1$ extends to a string whose sum is at least $s_*$. At termination the prefix itself is the returned string, so

$$s(z_*) = s_* = \max_{z \in \{0,1\}^\ell} s(z).$$

Thus $\mathsf{Search}$ uses $O(\log n + \ell)$ queries to $N$, each of polynomial length.

**Step 2: Bound the estimation error.** We now compare the maximized estimate with actual acceptance. Conditional on the shorter oracle determining $v_a$, the defining sampling of $A$ is

$$A(a, i) \overset{\text{ind}}{\leftarrow} \text{Bernoulli}((1 + v_a)/3), \qquad i \in [n^4].$$

Hoeffding's inequality followed by a union bound over the at most $2^n$ valid descriptions gives

$$\Pr_{\mathcal{O}}[|\widehat{v}_a - v_a| > 3/n] \le 2e^{-2n^2},$$
$$\Pr_{\mathcal{O}}[\neg \mathcal{E}_n] \le 2^{n+1} e^{-2n^2}.$$

On $\mathcal{E}_n$, let $z_{\text{opt}} \in \arg\max_z v_{a(z)}$. Since $\widehat{v}_{a(z)} = 3s(z)/n^4 - 1$, Step 1 implies

$$\begin{aligned} v_{a(z_*)} &\ge \widehat{v}_{a(z_*)} - 3/n \\ &\ge \widehat{v}_{a(z_{\text{opt}})} - 3/n \ge v_{a(z_{\text{opt}})} - 6/n. \end{aligned}$$

This proves the search bound.

**Step 3: Decide promise QCMA.** To obtain a decision procedure for every sufficiently padded description, observe that the failure bound in Step 2 is summable. The Borel–Cantelli lemma therefore gives $\mathcal{E}_n$ for all $n \ge n_0(\mathcal{O})$ with probability one. Fix such an oracle and a QCMA verifier $V$, viewed as a witnessless quantum circuit on ordinary input $(x, w)$. On input $x$, form the descriptions $a(x, w) = \langle V, (x, w) \rangle$ for classical witnesses $w$, padding all pairs to the same polynomial length

$$n \ge \max\{72, n_0(\mathcal{O})\}.$$

The finite constant $n_0(\mathcal{O})$ is hardwired into the resulting classical machine. On input $x$, use the $N$ query from Step 1 with the test family $a(x, \cdot)$ to output

$$H(\varnothing, \lfloor n^4/2 \rfloor + 1) = \mathbf{1}[\exists w : \widehat{v}_{a(x,w)} > 1/2].$$

On $\mathcal{E}_n$, the estimation bound gives

$$\begin{aligned} x \in L_{\text{yes}} &\implies \exists w : \widehat{v}_{a(x,w)} \ge 2/3 - 3/n > 1/2, \\ x \in L_{\text{no}} &\implies \forall w : \widehat{v}_{a(x,w)} \le 1/3 + 3/n < 1/2. \end{aligned}$$

Thus this deterministic polynomial-time algorithm is correct on both sides of the promise and returns a bit on every input. Therefore $\mathsf{QCMA}^{\mathcal{O}} \subseteq \mathsf{P}^{\mathcal{O}}$ for promise problems, and the inclusions $\mathsf{P}^{\mathcal{O}} \subseteq \mathsf{BQP}^{\mathcal{O}} \subseteq \mathsf{QCMA}^{\mathcal{O}}$ complete the proof.

# Appendix II: Cryptographic Security Proofs

## F Copy-Protection Definitions

We define copy-protection for one supplied key and two independent challenges. Every party has quantum access to the same oracle $\mathcal{O}$. After the split, the two recipients do not communicate, although their registers may be entangled.

**Definition 39** (Copy-protected PRFs)**.** *For a one-bit-output PRF (Definition 9), consider the following experiment.*

***PRF copy-protection***

1. Generate $(j, \mathsf{R_{sk}}) \leftarrow \mathsf{Gen}^{\mathcal{O}}(1^\lambda)$ with function index $k$, and run $(\mathsf{R}_1, \mathsf{R}_2) \leftarrow \mathcal{A}_0^{\mathcal{O}}(j, \mathsf{R_{sk}})$.
2. Independently sample $x_1, x_2 \leftarrow \{0,1\}^{m(\lambda)}$ and run $b_i \leftarrow \mathcal{A}_i^{\mathcal{O}}(x_i, \mathsf{R}_i)$ for $i = 1, 2$.
3. Return $\mathbf{1}[b_1 = F_k(x_1) \wedge b_2 = F_k(x_2)]$.

*The PRF is copy-protected if, for every* QPT $\mathcal{A}$*,*

$$\Pr[b_1 = F_k(x_1) \wedge b_2 = F_k(x_2)] \leq \tfrac{1}{2} + \mathsf{negl}(\lambda). \tag{86}$$

**Definition 40** (Copy-protected PKE)**.** *For a bit-encryption scheme (Definition 10), define the following experiment:*

$\mathsf{Piracy}_{\mathcal{A}}^{\mathcal{O}}(\lambda)$

1. Generate $(\mathrm{pk}, \mathsf{R_{sk}}) \leftarrow \mathsf{Gen}^{\mathcal{O}}(1^\lambda)$ and run
$$((m_{i,0}, m_{i,1})_{i=1}^2, \mathsf{R}_1, \mathsf{R}_2) \leftarrow \mathcal{A}_0^{\mathcal{O}}(\mathrm{pk}, \mathsf{R_{sk}}), \qquad m_{i,b} \in \{0,1\}.$$
2. Independently for $i = 1, 2$, sample $b_i \leftarrow \{0,1\}$ and $\mathrm{ct}_i \leftarrow \mathsf{Enc}^{\mathcal{O}}(\mathrm{pk}, m_{i,b_i})$. Run $\widehat{b}_i \leftarrow \mathcal{A}_i^{\mathcal{O}}(\mathrm{ct}_i, \mathsf{R}_i)$.
3. Return $\mathbf{1}[\widehat{b}_1 = b_1 \wedge \widehat{b}_2 = b_2]$.

*We call the scheme* copy-protected *if, for every* QPT $\mathcal{A}$*,*

$$\Pr\big[\mathsf{Piracy}_{\mathcal{A}}^{\mathcal{O}}(\lambda) = 1\big] \leq \tfrac{1}{2} + \mathsf{negl}(\lambda).$$

# G Security of Direct Coset Evaluation

In this section, we prove Lemma 28, the direct coset bound used in the PRF and encryption proofs (Sections 15.2 and 16.2). The bound applies to independent function-value and ciphertext challenges with access to the two-basis oracle (Equation (62)). We also prove the public-key quantum money guarantee (Corollary 2) in Appendix G.4.

We first define the two challenge experiments. The functions are sampled by replacing their values at the challenge inputs, which leaves them uniform while separating the fresh values from any auxiliary information held before the challenges.

**Definition 41** (Coset experiments)**.** *Fix $n \geq 1$. We define the* direct coset experiments *as follows.*

1. *Independently sample uniform Boolean functions $G_0, G_1 : \{0,1\}^n \to \{0,1\}$ and*

$$A \leftarrow \{V \leq \mathbb{F}_2^{4n} : \dim V = 2n\}, \quad a, b \leftarrow \mathbb{F}_2^{4n}, \quad x_1, x_2 \leftarrow \{0,1\}^n.$$

*Sample independent fair bits $(b_{i,0}, b_{i,1})$ for each distinct $x_i$, using the same pair when $x_1 = x_2$. Set*

$$F_c = G_c[x_i \leftarrow b_{i,c} : i \in \{1,2\}], \qquad F = F_0 \oplus F_1, \qquad C_F = C_{A,a,b,F_0,F_1}.$$

*Its query $(x, c, z)$ returns $F_c(x)$ when $z \in A + a$ for $c = 0$ or $z \in A^\perp + b$ for $c = 1$, and returns $\perp$ otherwise. Write $G = (G_0, G_1)$.*

2. *Fix algorithms $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ whose descriptions may depend on $G$, and give $\mathcal{A}_0$ the input density matrix $|A_{a,b}\rangle\langle A_{a,b}| \otimes \sigma_J^G$, where $\sigma_J^G$ is an arbitrary classical–quantum state depending only on $G$. For function-value challenges, run*

$$(\mathsf{R}_1, \mathsf{R}_2) \leftarrow \mathcal{A}_0^{C_F}(|A_{a,b}\rangle, \sigma_J^G), \qquad \widehat{f_i} \leftarrow \mathcal{A}_i^{C_F}(x_i, \mathsf{R}_i),$$

*and return $\mathbf{1}[\widehat{f_1} = F(x_1) \wedge \widehat{f_2} = F(x_2)]$.*

3. *For ciphertext challenges, instead run*

$$\begin{aligned} ((m_{i,0}, m_{i,1})_{i=1}^2, \mathsf{R}_1, \mathsf{R}_2) &\leftarrow \mathcal{A}_0^{C_F}(|A_{a,b}\rangle, \sigma_J^G), \\ \beta_1, \beta_2 \leftarrow \{0,1\}, \qquad \mathrm{ct}_i &= (x_i, m_{i,\beta_i} \oplus F(x_i)), \\ \widehat{\beta_i} &\leftarrow \mathcal{A}_i^{C_F}(\mathrm{ct}_i, \mathsf{R}_i), \end{aligned}$$

*and return $\mathbf{1}[\widehat{\beta_1} = \beta_1 \wedge \widehat{\beta_2} = \beta_2]$. The challenge bits are independent, $\mathsf{R}_1, \mathsf{R}_2$ may be entangled, and $\mathcal{A}_1, \mathcal{A}_2$ act without communication after $\mathcal{A}_0$ terminates. All three algorithms have unrestricted local computation.*

The following bound applies to both challenges.

**Lemma 28** (Security of the coset experiments)**.** *In either direct coset experiment (Definition 41), every adversary making at most $Q$ total quantum queries to $C_F$ satisfies*

$$\Pr[\textit{both challenges are answered correctly}] \leq \tfrac{1}{2} + O\big((Q+1)2^{-n/2}\big). \tag{87}$$

The proof of Lemma 28 appears in Appendix G.1.

## G.1 Proof of the direct coset bound

We prove Lemma 28 using two supporting lemmas: decisional coset monogamy with membership queries, and a reduction from fresh function values and ciphertexts to that game.

**Lemma 29** (Decisional monogamy with oracle access)**.** *Fix $n \geq 1$ and $Q \geq 0$, and independently sample*

$$A \leftarrow \{V \leq \mathbb{F}_2^{4n} : \dim V = 2n\}, \qquad a, b \leftarrow \mathbb{F}_2^{4n}.$$

*Let $M_0$ and $M_1$ test membership in $A + a$ and $A^\perp + b$. The algorithms have unrestricted local computation and may receive auxiliary information independent of $(A, a, b)$. Run $(\mathsf{R}_1, \mathsf{R}_2) \leftarrow \mathcal{A}_0^{M_0,M_1}(|A_{a,b}\rangle)$, then independently sample*

$$u \leftarrow A \setminus \{0\}, \qquad v \leftarrow A^\perp \setminus \{0\}, \qquad e = \langle u, b\rangle \oplus \langle v, a\rangle,$$

*and run $e_i \leftarrow \mathcal{A}_i^{M_0,M_1}(u, v, \mathsf{R}_i)$ without communication, optionally giving $A$ to $\mathcal{A}_1, \mathcal{A}_2$ after $\mathcal{A}_0$ terminates. Their registers may be entangled. If all three algorithms make at most $Q$ total queries, then*

$$\Pr_{A,a,b,u,v}[e_1 = e_2 = e] \leq \tfrac{1}{2} + O\big((Q+1)2^{-n/2}\big).$$

The proof of Lemma 29 appears in Appendix G.2.

The next lemma changes only the oracle used by $\mathcal{A}_0$, so that its queries reveal no fresh challenge values.

**Lemma 30** (Reduction from fresh values to coset monogamy)**.** *In the direct coset experiments (Definition 41), condition on $x_1 \neq x_2$, using the input state $|A_{a,b}\rangle\langle A_{a,b}| \otimes \sigma_J^G$. Run $\mathcal{A}_0$ with $C_G = C_{A,a,b,G_0,G_1}$ and $\mathcal{A}_1, \mathcal{A}_2$ with $C_F$. For each fixed $G = (G_0, G_1)$, any $Q$-query adversary induces a strategy in Lemma 29 using at most $4Q$ membership queries, with success equal to the adversary's in the function-value experiment and at least its success in the ciphertext experiment.*

The proof of Lemma 30 appears in Appendix G.3.

**Combining the two bounds.** We now prove Lemma 28 by a query hybrid. Put $N = 2^n$ and first condition on $x_1 \neq x_2$. Define

| | oracle for $\mathcal{A}_0$ | oracle for $\mathcal{A}_1, \mathcal{A}_2$ |
|---|---|---|
| $\mathsf{H}_0$ | $C_F$ | $C_F$ |
| $\mathsf{H}_1$ | $C_G$ | $C_F$. |

Both experiments use the conditional sampling and the displayed calls to $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2$ in the coset experiment (Definition 41), with the same input density matrix and winning predicate. Thus only $\mathcal{A}_0$'s oracle changes in $\mathsf{H}_1$, while $\mathcal{A}_1, \mathcal{A}_2$ still query $C_F$. Write $p_i = \Pr[\mathsf{H}_i = 1 \mid x_1 \neq x_2]$. For any query by $\mathcal{A}_0$ in $\mathsf{H}_1$, let $w_x$ be its weight on input $x$. Conditional on $(G, A, a, b)$, the query state, including $\sigma_J^G$, is independent of the uniform distinct inputs $x_i$, so

$$\mathbb{E}_{x_1 \neq x_2}[w_{x_1} + w_{x_2}] = \frac{2}{N} \sum_{x \in \{0,1\}^n} w_x \leq \frac{2}{N}.$$

Since the functions differ only at $x_1, x_2$, the query hybrid bound (Lemma 5) applies with expected query weight at most $2/N$ and gives

$$|p_0 - p_1| \leq 2Q\sqrt{2/N}.$$

Now the fresh-value reduction and decisional monogamy (Lemmas 29 and 30) give $p_1 \le 1/2 + O((Q+1)2^{-n/2})$. Finally,

$$\Pr[\text{both win}] \le p_0 + \Pr[x_1 = x_2] \le \tfrac{1}{2} + O((Q+1)2^{-n/2}) + N^{-1},$$

which proves the claimed bound for both challenge rules.

## G.2 Proof of decisional coset monogamy

We now prove Lemma 29, a quantitative adaptation of the oracle formulation in [BCG26, footnote 15].

**Replacing membership by larger sets.** Conditional on $A$, independently sample

$$L \leftarrow \{V \le A : \dim V = n\}, \qquad U \leftarrow \{V \ge A : \dim V = 3n\}.$$

Consider the three games

| | $M_0$ | $M_1$ |
|---|---|---|
| $\mathsf{H}_0$ | $\mathbf{1}[A + a]$ | $\mathbf{1}[A^\perp + b]$ |
| $\mathsf{H}_1$ | $\mathbf{1}[U + a]$ | $\mathbf{1}[A^\perp + b]$ |
| $\mathsf{H}_2$ | $\mathbf{1}[U + a]$ | $\mathbf{1}[L^\perp + b]$. |

Writing $(M_0^{(j)}, M_1^{(j)})$ for the displayed pair, game $\mathsf{H}_j$ runs

$$(\mathsf{R}_1, \mathsf{R}_2) \leftarrow \mathcal{A}_0^{M_0^{(j)}, M_1^{(j)}}(|A_{a,b}\rangle), \qquad e_i \leftarrow \mathcal{A}_i^{M_0^{(j)}, M_1^{(j)}}(u, v, \mathsf{R}_i),$$

where $(u, v)$ is sampled after the first call exactly as in the lemma. It returns $\mathbf{1}[e_1 = e_2 = e]$, using the original $e = \langle u, b\rangle \oplus \langle v, a\rangle$. For any point outside its original coset, the probability of entering the corresponding larger set is

$$\beta_n = \frac{1}{2^n + 1} \le 2^{-n}.$$

In $\mathsf{H}_0$, the query state is independent of $U$ conditional on $(A, a, b)$, so the query hybrid bound (Lemma 5) applied to the added points of $U + a$ gives

$$|\Pr[\mathsf{H}_0 = 1] - \Pr[\mathsf{H}_1 = 1]| \le 2Q\sqrt{\beta_n}.$$

Similarly, queries in $\mathsf{H}_1$ are independent of $L$ conditional on $(A, a, b, U)$, and the added points of $L^\perp + b$ therefore give

$$|\Pr[\mathsf{H}_1 = 1] - \Pr[\mathsf{H}_2 = 1]| \le 2Q\sqrt{\beta_n}.$$

The next claim bounds the game with both larger-set membership oracles.

**Claim 8.** *For $\mathsf{H}_2$ defined above,*

$$\Pr[\mathsf{H}_2 = 1] \le \tfrac{1}{2} + O(2^{-n}).$$

*The bound still holds when $A$ is given to the two later algorithms.*

The proof of Claim 8 appears in Appendix G.2.1.

Combining this bound with the two query hybrids gives

$$\begin{aligned}\Pr[\mathsf{H}_0 = 1] &\le \Pr[\mathsf{H}_2 = 1] + \sum_{i=0}^{1} |\Pr[\mathsf{H}_i = 1] - \Pr[\mathsf{H}_{i+1} = 1]| \\ &\le \tfrac{1}{2} + 4Q\sqrt{\beta_n} + O(2^{-n}) = \tfrac{1}{2} + O((Q+1)2^{-n/2}).\end{aligned}$$

### G.2.1 The larger-set game

We now prove Claim 8 by expressing the remaining hidden coset in $2n$ coordinates. Revealing the other coordinates makes both larger-set membership oracles computable.

Sample $\sigma \leftarrow U + a$ and $\tau \leftarrow L^{\perp} + b$, and give $(L, U, \sigma, \tau)$ to $\mathcal{A}_0$ and both later algorithms. Both larger-set membership oracles can now be evaluated from $U + a = U + \sigma$ and $L^{\perp} + b = L^{\perp} + \tau$. Choose a linear isomorphism $T$ determined by $(L, U)$ such that

$$\begin{aligned} T(L) &= \mathbb{F}_2^n \oplus 0 \oplus 0, \\ T(A) &= \mathbb{F}_2^n \oplus \bar{A} \oplus 0, \qquad \bar{A} \leftarrow \{V \le \mathbb{F}_2^{2n} : \dim V = n\}, \\ T(U) &= \mathbb{F}_2^n \oplus \mathbb{F}_2^{2n} \oplus 0, \end{aligned}$$

where the second equality defines $\bar{A}$. Write $U_T |z\rangle = |Tz\rangle$. Applying $U_T X^{\sigma} Z^{\tau}$ to the supplied state gives, up to global phase,

$$|+\rangle^{\otimes n} \otimes \left|\bar{A}_{\bar{a},\bar{b}}\right\rangle \otimes |0\rangle^{\otimes n}, \qquad \bar{a}, \bar{b} \leftarrow \mathbb{F}_2^{2n} \text{ independently.}$$

Writing $\bar{u}, \bar{v}$ for the middle coordinates of $Tu, T^{-\mathsf{T}}v$, respectively, decomposes the challenge as

$$e = \underbrace{\langle u, \tau\rangle \oplus \langle v, \sigma\rangle}_{\text{known after the challenge}} \oplus \langle \bar{u}, \bar{b}\rangle \oplus \langle \bar{v}, \bar{a}\rangle.$$

To apply the decisional bound, exclude the event $\mathsf{Bad} = \{\bar{u} = 0 \text{ or } \bar{v} = 0\}$, which has probability at most $2\beta_n$. Outside it, $\bar{u}, \bar{v}$ are independent uniform nonzero vectors of $\bar{A}, \bar{A}^{\perp}$. Define $(\mathcal{B}_0, \mathcal{B}_1, \mathcal{B}_2)$ for the residual game by the following calls:

1. Algorithm $\mathcal{B}_0$, on input $\left|\bar{A}_{\bar{a},\bar{b}}\right\rangle$, forms the three-factor state above and applies $(U_T X^{\sigma} Z^{\tau})^{-1}$. Sample independent uniform $r, t \in \{0,1\}^n$ for the two local simulators. Give the state to $\mathcal{A}_0$, answering its queries by membership in $U + \sigma$ and $L^{\perp} + \tau$. If $\mathcal{A}_0$ returns $(\mathsf{R}_1, \mathsf{R}_2)$, return $(\mathsf{R}_1, r, t)$ to $\mathcal{B}_1$ and $(\mathsf{R}_2, r, t)$ to $\mathcal{B}_2$.
2. Algorithm $\mathcal{B}_i$, on input $(\bar{u}, \bar{v}, \mathsf{R}_i, r, t)$, sets $u = T^{-1}(r, \bar{u}, 0)$ and $v = T^{\mathsf{T}}(0, \bar{v}, t)$, and runs $e_i \leftarrow \mathcal{A}_i^{\mathbf{1}[U+\sigma], \mathbf{1}[L^{\perp}+\tau]}(u, v, \mathsf{R}_i)$, then outputs $e_i \oplus \langle u, \tau\rangle \oplus \langle v, \sigma\rangle$.

Conditional on $(L, U, \sigma, \tau)$ and $\neg\mathsf{Bad}$, these operations give the original state and challenge distribution, as in the change-of-basis argument of [BCG26, proof of Theorem 22]. The outputs both equal the residual challenge bit exactly when both original outputs equal $e$. Thus [BCG26, Corollary 3] bounds the conditional success probability by $1/2 + O(2^{-n})$. The bound permits giving $A$ to $\mathcal{B}_1, \mathcal{B}_2$, since they can reconstruct it from $\bar{A}, L, U, T$ without either residual shift. Finally, $\Pr[\mathsf{Bad}] \le 2\beta_n \le 2^{1-n}$, so

$$\Pr[\mathsf{H}_2 = 1] \le \Pr[\mathsf{H}_2 = 1 \mid \neg\mathsf{Bad}] + \Pr[\mathsf{Bad}] \le \tfrac{1}{2} + O(2^{-n}).$$

## G.3 Proof of the fresh-value reduction

We now prove Lemma 30 for the direct coset experiments (Definition 41), by expressing the fresh function bits using the monogamy challenge and independent fair bits.

**Algorithm $\mathcal{B}_0$.** Fix $G$. On input $|A_{a,b}\rangle$, algorithm $\mathcal{B}_0^{M_0, M_1}$ prepares $\sigma_J^G$ and samples

$$(x_1, x_2) \leftarrow \{(x_1, x_2) \in (\{0,1\}^n)^2 : x_1 \ne x_2\}, \qquad (r_1, r_2, \zeta_1, \zeta_2) \leftarrow \{0,1\}^4.$$

Run $\mathcal{A}_0^{C_G}(|A_{a,b}\rangle, \sigma_J^G)$, implementing $C_G(x, c, z)$ by returning $G_c(x)$ if $M_c(z) = 1$ and $\perp$ otherwise, then uncomputing the membership test. If $\mathcal{A}_0$ outputs registers $\mathsf{R}_1, \mathsf{R}_2$ and, in the ciphertext experiment, message pairs, give $\mathcal{B}_i$ its register $\mathsf{R}_i$, both message pairs and $(x_1, x_2, r_1, r_2, \zeta_1, \zeta_2)$. These classical values specify the oracle answers below and are not given to $\mathcal{A}_i$.

**Algorithms $\mathcal{B}_1, \mathcal{B}_2$.** On receiving the monogamy challenge $(u, v)$, each $\mathcal{B}_i$ implements the oracle below for $\mathcal{A}_i$. To verify its distribution, define

$$b_{i,0} = \langle v, a\rangle \oplus r_i, \qquad b_{i,1} = \langle u, b\rangle \oplus r_i \oplus \zeta_i. \tag{88}$$

These values describe the simulated function rather than a computation of the unknown shifts $a, b$. The algorithm $\mathcal{B}_i^{M_0,M_1}$ answers each query of $\mathcal{A}_i$ by the following coherent computation.

**Answering a query of $\mathcal{A}_i$**

On query $(x, c, z)$, perform these steps coherently.

1. Test $z \in A + a$ when $c = 0$, or $z \in A^\perp + b$ when $c = 1$, returning $\perp$ if the test fails.
2. For $x \notin \{x_1, x_2\}$, return $G_c(x)$.
3. For $x = x_j$, return $\langle v, z\rangle \oplus r_j$ when $c = 0$, or $\langle u, z\rangle \oplus r_j \oplus \zeta_j$ when $c = 1$.

Uncompute the membership test after copying the answer.

This implements $C_F$ exactly because

$$z \in A + a \Longrightarrow \langle v, z\rangle = \langle v, a\rangle,$$
$$z \in A^\perp + b \Longrightarrow \langle u, z\rangle = \langle u, b\rangle.$$

Including uncomputation, each simulated query uses at most four membership queries. For every fixed coset and monogamy challenge, the independent fair bits $(r_i, \zeta_i)$ make the pairs $(b_{i,0}, b_{i,1})$ independent and uniform, with

$$F(x_i) = e \oplus \zeta_i, \qquad e = \langle v, a\rangle \oplus \langle u, b\rangle.$$

**Function-value challenges.** Algorithm $\mathcal{B}_i$ runs $\widehat{f_i} \leftarrow \mathcal{A}_i^{C_F}(x_i, \mathsf{R}_i)$ using the displayed oracle answers and outputs $\widehat{e_i} = \widehat{f_i} \oplus \zeta_i$. Since $F(x_i) = e \oplus \zeta_i$, the winning events agree:

$$\widehat{f_1} = F(x_1) \wedge \widehat{f_2} = F(x_2) \quad \Longleftrightarrow \quad \widehat{e_1} = \widehat{e_2} = e.$$

**Ciphertext challenges.** Each $\mathcal{B}_i$ has both message pairs output by $\mathcal{A}_0$. Suppose first that $m_{i,0} \neq m_{i,1}$ for both $i$. Independently, $\mathcal{B}_i$ samples $c_i \leftarrow \{0, 1\}$ and runs $\widehat{\beta_i} \leftarrow \mathcal{A}_i^{C_F}((x_i, c_i), \mathsf{R}_i)$ using the displayed oracle answers. The implied bits

$$\beta_i = c_i \oplus m_{i,0} \oplus F(x_i)$$

are independent and uniform conditional on the messages and $F$, and satisfy $c_i = m_{i,\beta_i} \oplus F(x_i)$. Thus these are exactly the required independent ciphertext challenges. Translate the outputs by

$$\widehat{e_i} = \widehat{\beta_i} \oplus m_{i,0} \oplus c_i \oplus \zeta_i,$$

so both translated outputs equal $e$ exactly when both ciphertext guesses are correct.

If either message pair is equal, $\mathcal{B}_1$ and $\mathcal{B}_2$ instead output the saved bit $r_1$. This bit is uniform and independent of $\mathcal{A}_0$'s output and the monogamy challenge, so the branch succeeds with probability $1/2$. In the original ciphertext experiment, the corresponding challenge bit is independent of the joint input and outputs of $\mathcal{A}_1, \mathcal{A}_2$, making their joint success at most $1/2$. The overall strategy therefore succeeds at least as often as the ciphertext adversary, using at most $4Q$ membership queries.

## G.4 Public-key quantum money

We prove Corollary 2 by authenticating a coset mini-scheme with $\mathsf{CSig}$. Repeating the encoded validity test is necessary because its off-preimage answers are random bits. Use the cryptographic oracle $\mathcal{O} = (\mathcal{O}_{\mathsf{base}}, N, A)$ with $\mathcal{O}_{\mathsf{base}} = (\mathcal{O}_{\mathsf{prf}}, \mathcal{O}_{\mathsf{pke}}, \mathcal{O}_{\mathsf{sig}}, \mathcal{O}_{\mathsf{prs}})$ (Definition 37). The mini-scheme uses the encryption encoding $T_s = T_{H_s,B_s,\mathfrak{F}_{C_s}}$ and key $|K_s\rangle = |\Psi_{H_s}\rangle\,|\chi_s\rangle$ (Definition 32).

**The coset mini-scheme**

Generation on input $1^n$

Run $(s, (s, \mathsf{R}_{\mathsf{note}})) \leftarrow \mathsf{PKE.Gen}(1^n)$ and return $(s, \mathsf{R}_{\mathsf{note}})$.

Verification of $(s, \mathsf{R}_{\mathsf{note}})$

Reject incorrect lengths and write $\mathsf{R}_{\mathsf{note}} = \mathsf{R}_{\mathsf{hash}}\mathsf{R}_{\mathsf{coset}}$ for its predicate-preimage state register and $4n$-qubit coset register. Put $r = 3n$.

1. For $c = 0, 1$, first apply $\mathrm{H}^{\otimes 4n}$ to $\mathsf{R}_{\mathsf{coset}}$ when $c = 1$, then run $r$ checks. In each check, independently sample $x \leftarrow \{0,1\}^n$, use $\mathsf{EvalHash}^{T_s}$ with key $\mathsf{R}_{\mathsf{hash}}$, input $(x, c, \mathsf{R}_{\mathsf{coset}}, 0)$ and a zero response bit, then measure that bit. Reject unless the bit is one.
2. Apply $\mathrm{H}^{\otimes 4n}$ to $\mathsf{R}_{\mathsf{coset}}$. Accept and return the updated note.

Each check uses fresh private workspace, which is discarded after uncomputing the decoder. Here $j = 0$ is the validity bit of the Boolean encoding $\mathfrak{F}_{C_s}$ (Equation (62)).

The mini-scheme counterfeiting game gives $(s, \mathsf{R}_{\mathsf{note}})$ to an adversary, which returns two registers. It wins if both pass verification under the same serial $s$. The adversary has full access to $\mathcal{O}$, polynomial query count and length, and unrestricted local computation. We use the following search bound for a single coset state.

**Lemma 31** (Coset search hardness, [CLLZ22])**.** *Independently sample $A \leftarrow \{V \le \mathbb{F}_2^{4n} : \dim V = 2n\}$ and $a, b \leftarrow \mathbb{F}_2^{4n}$, and let $M_0, M_1$ test membership in $A + a, A^{\perp} + b$. Every algorithm with unrestricted local computation and auxiliary information independent of $(A, a, b)$, given one copy of $|A_{a,b}\rangle$ and polynomially many quantum queries to $M_0, M_1$, satisfies*

$$\Pr_{A,a,b}[(v, w) \leftarrow \mathcal{A}^{M_0,M_1}(|A_{a,b}\rangle) : v \in A + a,\ w \in A^{\perp} + b] = \mathsf{negl}(n).$$

By the direct-product theorem of [CLLZ22], polynomially many queries have negligible success. Independent auxiliary information can be fixed before applying this uniform query bound.

The following claim supplies the mini-scheme guarantee needed by the authentication construction.

**Claim 9.** *Averaged over the full oracle $\mathcal{O}$, the coset mini-scheme has negligible joint correctness error for polynomially many verifications. Every counterfeiter with full access to $\mathcal{O}$, polynomial query count and length, and unrestricted local computation satisfies*

$$\Pr_{\mathcal{O}}[\textit{both output notes pass for the issued serial}] = \mathsf{negl}(n).$$

The proof of Claim 9 appears in Appendix G.4.1.

We now prove the public-key money guarantee (Corollary 2) using this mini-scheme and $\mathsf{CSig}$.

**Authenticating the serial.** Generate a $\mathsf{CSig}$ signing key as the quantum minting key and use its classical verification key as the money verification key. To mint, generate a mini-scheme note $(s, \mathsf{R}_{\mathsf{note}})$, sign $s$ with the current minting key, and return $(s, \mathsf{R}_{\mathsf{note}}, \mathsf{R}_{\mathsf{sig}})$, using the updated minting

key for subsequent calls. Verification runs $\mathsf{CSig.Ver}$ and the mini-scheme verifier on their respective registers, accepting only if both accept and returning the updated registers. Signature correctness (Theorem 17) and Claim 9 give reusable minting and banknotes.

The mini-scheme compiler of [AC13] applies by forwarding quantum signatures unchanged. Pad an adversary making at most $q$ adaptive minting requests to exactly $q$ requests, appending the extra honest notes to its output, and let $\ell = \mathsf{poly}(n)$ bound its number of outputs. Write $\mathsf{Count}$ for the number accepted in the joint verification experiment. Use the following reductions.

1. *Fresh serial.* Guess one of the $\ell$ output positions. Generate the mini-scheme notes honestly and obtain their signatures from the signing challenger. Return the guessed serial and its untouched signature register as the forgery.

2. *Repeated issued serial.* Guess one of the $q$ minting requests and a pair of output positions. Embed the mini-scheme challenge at that request, generate all other notes honestly, and sign with an independently generated $\mathsf{CSig}$ key. Return the two guessed mini-scheme registers without first verifying any output.

Let $\mathrm{Adv}_{\mathsf{sig}}$ and $\mathrm{Adv}_{\mathsf{mini}}$ be these reductions' success probabilities. Outside an issued-serial collision, the event $\mathsf{Count} > q$ contains either an accepted fresh serial or two accepted outputs with the same issued serial. Verification acts separately on each note, so the reductions' untouched outputs have the corresponding verification marginals. A union bound gives

$$\Pr[\mathsf{Count} > q] \le \ell \,\mathrm{Adv}_{\mathsf{sig}} + q\binom{\ell}{2}\,\mathrm{Adv}_{\mathsf{mini}} + q^2 2^{-n} + \mathsf{negl}(n).$$

Here $q^2 2^{-n}$ bounds collisions among the ideal uniform serials, and the negligible term accounts for replacing the honest generated notes by ideal notes and for errors on the appended honest notes. Both advantages are negligible by signature security (Theorem 17) and Claim 9.

**One fixed oracle.** For a polynomial circuit-size bound $q$, let $p_{D,\mathcal{O}}$ be the mini-scheme success probability of description $D$, and define the uniform quantum test $T^{\mathcal{O}}_{\mathrm{mini}}(1^n, D)$: generate one note, reject an invalid description, otherwise run $D$, verify both outputs, and return one iff both accept. This test has acceptance probability exactly $p_{D,\mathcal{O}}$.

Fix an inverse-polynomial $\epsilon > 0$. Pad the descriptions $D$ to one polynomial length, interpreting invalid strings as rejection, and encode each test by $a(D) = \langle T_{\mathrm{mini}}, (1^n, D)\rangle$ of common polynomial length $m \ge \max\{n, 6/\epsilon\}$. The adversary $\mathcal{A}_{\mathrm{sel}}$ first runs $\mathsf{Search}^{\mathcal{O}}(a)$, obtaining $D_*$. It then receives the challenge note, runs $D_*$ on it, and returns its outputs. The estimation and search bound (Lemma 27) gives

$$\mathbb{E}_{\mathcal{O}} \max_{|D|\le q(n)} p_{D,\mathcal{O}} \le \Pr_{\mathcal{O}}[\mathcal{A}_{\mathrm{sel}} \text{ wins}] + \epsilon + 2^{m+1} e^{-2m^2} \le \epsilon + \mathsf{negl}(n),$$

since $\mathcal{A}_{\mathrm{sel}}$ makes polynomially many queries and is covered by Claim 9. Since $\epsilon$ is arbitrary, $\delta_q(n) := \mathbb{E}_{\mathcal{O}} \max_{|D|\le q(n)} p_{D,\mathcal{O}}$ is negligible. Markov's inequality gives

$$\Pr_{\mathcal{O}}\left[\max_{|D|\le q(n)} p_{D,\mathcal{O}} > \sqrt{\delta_q(n)}\right] \le \sqrt{\delta_q(n)}.$$

This bound is summable, so Borel–Cantelli and a countable intersection over polynomial bounds $q$ give security for every polynomial-size classical description, including one depending on $\mathcal{O}$. Intersect this probability-one event with the fixed-oracle guarantees (Theorem 19). The authentication construction above then proves Corollary 2 for the same oracle, against nonuniform counterfeiters with no quantum advice.

### G.4.1 Correctness and security of the coset mini-scheme

We now prove Claim 9. The key is to turn two accepted notes into a primal and a dual coset vector without conditioning the counterfeiter's input on acceptance.

**Correctness and reuse.** On an ideal note $|K_s\rangle$, direct validity queries return one in both bases and preserve the state. For $H_s \in \mathsf{Good}_{\kappa_M}$, replace the $2r$ encoded calls one at a time by those direct calls. The coherent evaluation bound (Theorem 14) gives joint trace-distance error at most $4r\sqrt{\delta_{\mathrm{dec}}(\kappa_M)}$, including the acceptance bits and returned note. Averaging the bad-hash event, whose probability is at most $2^{-\kappa_M}$, and adding the functional-key preparation error (Lemma 18) gives negligible joint error for any polynomial number of verifications.

**Rejecting invalid coset components.** Let $p_{\mathrm{real}}$ be the probability that both forged notes pass. Replace key generation by $s \leftarrow \{0,1\}^n$ and the ideal note $|K_s\rangle$, and let $p_{\mathrm{ideal}}$ denote the resulting probability. Functional-key correctness (Lemma 18) gives $|p_{\mathrm{real}} - p_{\mathrm{ideal}}| \le \epsilon_{\mathrm{prep}}(n_K)$. We analyze this ideal experiment, in which $s$ is uniform and independent of the sampled oracle. Fix a serial $s$ and basis $c$, and let $\Pi_c$ project the coset register onto $A_s + a_s$ for $c = 0$ or $A_s^{\perp} + b_s$ for $c = 1$. Let $E_c$ be the acceptance effect of the $r = 3n$ checks after the indicated change of basis, so their acceptance probability on $\rho$ is $\mathrm{Tr}(E_c\rho)$. We need the following bound on acceptance outside the correct coset.

**Claim 10.** *Put $\alpha = (2/3)^r$ and $\tau_{\mathrm{money}}(n) = 6\sqrt{(a_{\kappa_M} + 4n + 1)2^{-n}}$. Except for an event of probability at most $2\tau_{\mathrm{money}}(n)$ over $B_s$, both acceptance effects satisfy*

$$E_c \le \Pi_c + \alpha I \qquad (c \in \{0,1\}). \tag{89}$$

*Each accepting channel preserves the ranges of $\Pi_c$ and $I - \Pi_c$.*

The proof of Claim 10 appears in Appendix G.4.2. These operator bounds hold for every forged state, including states entangled with other notes and depending on the sampled functions.

**From two accepted notes to coset search.** Consider the following search algorithm.

> **Search from a mini-scheme counterfeiter**
>
> On input $(s, |K_s\rangle)$, run the counterfeiter and the primal checks on both output notes. Abort if either group rejects. Otherwise measure the first coset register in the computational basis and the second in the Hadamard basis, and return their outcomes $(v, w)$.

No conditional state is supplied to the search algorithm.

Let $\rho$ be the joint output of a counterfeiter given one note. Run the first, primal-basis group of checks on both output notes, and write $\sigma$ for the unnormalized state when both groups accept. The preceding bound gives

$$\mathrm{Tr}[((I - \Pi_0) \otimes I)\sigma] \le \alpha.$$

The two notes' operations commute with each other. Thus, writing $\widehat{\Pi}_1 = \mathrm{H}^{\otimes 4n}\Pi_1\mathrm{H}^{\otimes 4n}$ and $\widehat{E}_1$ for the dual acceptance effect in the original basis, their joint verification probability is at most

$$\begin{aligned} \mathrm{Tr}[(\widehat{E}_1 \otimes \widehat{E}_1)\sigma] &\le \mathrm{Tr}[(I \otimes \widehat{E}_1)\sigma] \\ &\le \mathrm{Tr}[(I \otimes \widehat{\Pi}_1)\sigma] + \alpha \\ &\le \mathrm{Tr}[(\Pi_0 \otimes \widehat{\Pi}_1)\sigma] + 2\alpha. \end{aligned} \tag{90}$$

The last trace is exactly the unconditional success probability of the displayed search algorithm. Let $q_0$ be this success probability in the ideal experiment, averaged over the original oracle sample. The two exceptional noise events therefore give

$$p_{\text{ideal}} \le q_0 + 2\alpha + 2\tau_{\text{money}}(n).$$

**Removing the other oracle access.** We show that this search algorithm has negligible success even with the full cryptographic oracle. Use the ideal generation experiment fixed above and define the following experiments, each running the displayed search algorithm.

- $\mathsf{H}_0$ Use the original oracle and ideal note.
- $\mathsf{H}_1$ Replace only the indexed encryption-key target by

$$\left|\Omega_K^0\right\rangle = 2^{-n/2}\left(\sum_{t\neq s}|t\rangle\,|K_t\rangle + |s\rangle\left|0^{n_K-n}\right\rangle\right),$$

  sample its preparation pair from $\mathcal{D}_{\text{prep}}(\left|\Omega_K^0\right\rangle)$, and sample the one recursive extension conditional on the resulting full base. Continue to supply the original $|K_s\rangle$.
- $\mathsf{H}_2$ Apply encoded-function simulation (Theorem 14) to the entire preceding experiment for each fixed $G$ below, with $J = K = \{s\}$, an initial request for $s$, and direct oracle $\mathfrak{F}_{C_s}$.

Write $q_i$ for the search success probability in $\mathsf{H}_i$. For the first change, fix all functions, hashes and preparation pairs except the indexed encryption-key pair. Both targets and the initial density matrix are then fixed, and $\| |\Omega_K\rangle - \left|\Omega_K^0\right\rangle \|_2 \le 2^{1-n/2}$. The target-change bound (Theorem 13) applies in dimension $n_K$ to the displayed targets. Since $h(v) = (1+v)/3$ and the query bounds are polynomial, it gives

$$|q_0 - q_1| \le 2^{-\Omega(n)}.$$

Let $\mathcal{O}_{\mathsf{base}}^{1,-s}$ be the base in $\mathsf{H}_1$ with $T_s$ omitted, and put

$$G = (F_{s,0}, F_{s,1}, \mathcal{O}_{\mathsf{base}}^{1,-s}), \qquad \Theta = (A_s, a_s, b_s).$$

The indexed target replaces $|K_s\rangle$ by $|0^{n_K-n}\rangle$, and the unchanged graph pair depends only on $F_{s,0}, F_{s,1}$. All other primitive components were sampled independently. Hence $\Theta, H_s, B_s$ are independent conditional on $G$. The direct function $\mathfrak{F}_{C_s}$ is determined by $(G,\Theta)$, and the supplied coset state depends only on $\Theta$. For each fixed $G$, these are the sampling and request hypotheses of encoded-function simulation (Theorem 14). Averaging its joint-output bound, including $\Theta$, over $G$ gives

$$|q_1 - q_2| \le 2^{-\Omega(n)},$$

and a polynomial-query algorithm whose one classical sample $\eta_G$ is independent of $\Theta$ conditional on $G$. Each query to $\mathfrak{F}_{C_s}$ is implemented with two membership queries and the fixed functions $F_{s,0}, F_{s,1}$. Applying coset search hardness (Lemma 31) for each fixed $G$ therefore gives the bound $q_2 = \mathsf{negl}(n)$. Combining this with Equation (90) and the probability $2\tau_{\text{money}}(n)$ of bad noise gives

$$p_{\text{real}} \le \epsilon_{\text{prep}}(n_K) + 2\tau_{\text{money}}(n) + 2\alpha + q_2 + |q_0 - q_1| + |q_1 - q_2| = \mathsf{negl}(n).$$

Thus the mini-scheme has negligible counterfeiting probability.

### G.4.2 Rejecting invalid coset components

We now prove Claim 10. Fix the serial $s$ and a basis $c \in \{0,1\}$. We first bound a single random-function check for every input state, then apply that bound after each accepted check.

For a random Boolean function $B_s$, put

$$p_c(B_s) = \max_{\rho \text{ on } a_{\kappa_M}+4n \text{ qubits}} \Pr_{x \leftarrow \{0,1\}^n,\,(z,v)\leftarrow D_c(x,\rho)} [B_s(v,(x,c,z,0)) = 1],$$

where the oracle-free channel $D_c$ measures the coset register in the computational basis to obtain $z$, then runs $\mathsf{Decode}$ on the predicate-preimage state register with input $(x,c,z,0)$. Decoder failure is assigned answer zero. The random access encoding argument gives

$$\Pr_{B_s}[p_c(B_s) > 2/3] \le \tau_{\text{money}}(n). \tag{91}$$

To prove this estimate, sample independent uniform bits $Z_{x,z,v}$ and $w \leftarrow \{0,1\}^{2^n}$, and set

$$B_w(v,(x,c,z,0)) = Z_{x,z,v} \oplus w_x.$$

For each $w$, encode a state maximizing $p_c(B_w)$ together with the flag $\mathbf{1}[p_c(B_w) > 2/3]$. To recover $w_x$ when the flag is one, run $D_c$ and output $1 \oplus Z_{x,z,v}$ on decoder success. Output an independent fair bit if the flag is zero or decoding fails. Writing $\gamma = \Pr[p_c(B_s) > 2/3]$, the average recovery probability is at least $1/2+\gamma/6$. Fixing $Z$ to attain this average and applying Nayak's bound [Nay99] to $a_{\kappa_M} + 4n + 1$ qubits gives $\gamma \le \tau_{\text{money}}(n)$, proving Equation (91).

Fix $B_s$ outside the two exceptional events in Equation (91). On the range of $I - \Pi_c$, the encoded validity bit is zero on a correct preimage and equals $B_s$ otherwise. It is therefore pointwise at most the bit used to define $p_c(B_s)$. Each check has acceptance probability at most $2/3$ on this range. Moreover, each check leaves the coset basis value $z$ unchanged, because $\mathsf{EvalHash}$ uses it as an unchanged input. The accepting channel for $r$ checks consequently preserves the ranges of $\Pi_c$ and $I - \Pi_c$, and the operator $E_c$ giving acceptance probability $\mathrm{Tr}(E_c\rho)$ obeys

$$E_c \le \Pi_c + \alpha I, \qquad \alpha = (2/3)^r.$$

These inequalities hold for every forged state, including states entangled with other notes and depending on the sampled functions.

# H Proofs of the Signature Verification Bounds

In this section, we prove Lemmas 19 and 20, the two verification bounds used in Section 17.3 after removing the encoded signature oracles.

## H.1 Repeated verification with a random function

We now prove Lemma 19 using the random-function bound. Once $B$ is fixed outside its exceptional event, the same bound applies after every accepted check.

Set $B(\perp, y) = 0$ and define

$$p(B) = \max_{\rho \text{ on } a_\kappa \text{ qubits}} \Pr_{y\leftarrow\{0,1\}^\kappa,\, v\leftarrow \mathsf{Decode}(y,\rho)} [B(v,y) = 1].$$

The decoder is oracle-free and returns an element of $C_\kappa \cup \{\perp\}$. Applying the random-function bound (Lemma 12) with $\lambda = \kappa$ and $a = a_\kappa$ gives

$$\Pr_B[p(B) > 2/3] \le 6\sqrt{(a_\kappa + 1)2^{-\kappa}} = \tau_{\text{sig}}(\kappa).$$

**Repeated verification with fixed $B$.** Fix $B$ with $p(B) \le 2/3$. Uncomputing the decoder in $\mathsf{EvalHash}^B$ leaves its response bit unchanged, so each check, with its uniformly sampled $y$, has acceptance probability at most $2/3$ for every input state. This includes the state conditioned on earlier checks accepting, even for states entangled with an auxiliary system and depending on $B$. Therefore

$$\Pr[V_B^r(\rho) = 1] = \prod_{j=1}^{r} \Pr[\text{check } j \text{ accepts} \mid B, \text{checks } 1, \dots, j-1 \text{ accept}] \le (2/3)^r,$$

which proves the claimed bound outside an event of probability $\tau_{\mathrm{sig}}(\kappa)$.

## H.2 Predicting an unsigned value from equality queries

We now prove Lemma 20 by removing equality queries at messages that have not yet been signed. This must be done during the interaction, since fixing the final forged message could reveal information about the function. Recall that $F : \{0,1\}^n \to \{0,1\}^n$ is uniform and $\mathsf{Eq}_F(m, z) = \mathbf{1}[z = F(m)]$.

**The experiments.** For $0 \le j \le Q+1$, define $\mathsf{H}_j$ as follows. Sample $F \leftarrow \{\{0,1\}^n \to \{0,1\}^n\}$ and the adversary's auxiliary state independently. Write $\sigma_J$ for that auxiliary density matrix, initialize an empty database $D$, and run $\mathcal{A}(\sigma_J)$ with classical signing requests and coherent equality queries answered as follows:

1. On a classical signing request $m$, set $D(m) = F(m)$ if $m \notin \operatorname{dom} D$, then return $D(m)$.
2. Number equality queries consecutively and answer the first $j$ with $E_D$, and the rest with $\mathsf{Eq}_F$, where
$$E_D(m, z) = \begin{cases} \mathbf{1}[z = D(m)], & m \in \operatorname{dom} D, \\ 0, & m \notin \operatorname{dom} D. \end{cases}$$
3. On output $(m, z)$, reject if $m \in \operatorname{dom} D$. Otherwise make one more equality query under the same rule and return its answer.

The signed set is always $S = \operatorname{dom} D$. Thus $\mathsf{H}_0$ returns the forgery bit and $\mathsf{H}_{Q+1}$ always returns zero, while every experiment uses the same classical signing oracle and updates its database from its own requests.

**One equality query.** Consider the next query after a prefix that has used only $E_D$, and condition on its classical history $h$, including $D$. During this prefix, $E_D$ reveals no function value outside $D$, and a signing request adds only its requested value to $D$. Inducting on the interaction therefore shows that, conditional on $h$, the unused values $(F(m))_{m \notin \operatorname{dom} D}$ are still independent and uniform. If $w_{m,z}(h)$ is the next query's squared weight, then

$$\begin{aligned} \mathbb{E}_{F|h} \sum_{m \notin S, z} w_{m,z}(h) \mathbf{1}[F(m) = z] &= 2^{-n} \sum_{m \notin S, z} w_{m,z}(h) \\ &\le 2^{-n}. \end{aligned}$$

This includes the final test in Step 3. At the query where $\mathsf{H}_{j-1}$ and $\mathsf{H}_j$ differ, both have used only $E_D$ so far, and afterwards both use $\mathsf{Eq}_F$ with the same signing rule. That common continuation cannot increase trace distance, so the quantum query bound (Lemma 5) bounds each adjacent difference by $2\sqrt{2^{-n}}$. Summing over at most $Q+1$ queries gives

$$\Pr_F[m \notin S \wedge z = F(m)] \le \sum_{j=1}^{Q+1} |\Pr[\mathsf{H}_{j-1} = 1] - \Pr[\mathsf{H}_j = 1]| \le 2(Q+1)2^{-n/2}.$$

# I Changing the Oracle at a Hidden Key

We prove that polynomially many preparations at one hidden key can be replaced by copies of an independent binary-phase state, even with quantum access to the full recursive oracle (Theorem 20).

The PRS preparation oracle (Definition 36) samples

$$\Theta_k \leftarrow \{\{0,1\}^{5\lambda} \to \{0,1\}\}, \qquad |\Omega_{\Theta_k}\rangle = 2^{-5\lambda/2} \sum_{x\in\{0,1\}^{5\lambda}} (-1)^{\Theta_k(x)} |x\rangle, \qquad (R_k, S_k) \leftarrow \mathcal{D}_{\mathrm{prep}}(|\Omega_{\Theta_k}\rangle).$$

These samples are independent over $k \in \{0,1\}^\lambda$, and only $(R_k, S_k)$ are oracle components. The other base components form a Boolean oracle $\mathcal{O}_{\mathrm{aux}} : \{0,1\}^* \to \{0,1\}$, independent of this family. The cryptographic oracle (Definition 37) is $\mathcal{O} = (\mathcal{O}_{\mathrm{aux}}, (R_k, S_k)_k, N, A)$. Queries $(\mathrm{aux}, u)$, $(R, k, x)$ and $(S, k, x, y)$ return $\mathcal{O}_{\mathrm{aux}}(u)$, $R_k(x)$ and $S_k(x, y)$, respectively. Sample one recursive extension $(N, A)$ of these functions (Definition 26). For fixed $\mathcal{O}$, write $\rho_{R_k,S_k}$ for the output density matrix of $\mathsf{Prepare}^{R_k,S_k}(1^{5\lambda})$, so $r$ calls on separate input registers produce $\rho^{\mathcal{O}}_{k,r} = \rho^{\otimes r}_{R_k,S_k}$.

**Theorem 20** (Independent binary-phase input)**.** *Fix a family of density matrices $(\zeta_\theta)_\theta$ indexed only by the phase function. For polynomial $r$ and every quantum algorithm $\mathcal{A}$ with polynomial query count and length and unrestricted local computation,*

$$\tfrac{1}{2}\left\|\mathbb{E}_{\mathcal{O},k}[\mathcal{A}^{\mathcal{O}}(\rho^{\mathcal{O}}_{k,r} \otimes \zeta_{\Theta_k})] - \mathbb{E}_{\mathcal{O},\Theta}[\mathcal{A}^{\mathcal{O}}((|\Omega_\Theta\rangle\langle\Omega_\Theta|)^{\otimes r} \otimes \zeta_\Theta)]\right\|_1 \le 2^{-\Omega_{\mathcal{A}}(\lambda)}, \tag{92}$$

*where $k \leftarrow \{0,1\}^\lambda$ and $\Theta \leftarrow \{\{0,1\}^{5\lambda} \to \{0,1\}\}$ are independent of $\mathcal{O}$.*

The proof of Theorem 20 appears in Appendix I.1.

## I.1 Proof of the independent-input bound

We prove the independent-input bound (Theorem 20). Fix polynomial query bounds $Q, L$ with $L \ge \lambda$, and put

$$\sigma_\Theta = (|\Omega_\Theta\rangle\langle\Omega_\Theta|)^{\otimes r} \otimes \zeta_\Theta.$$

Every experiment below samples its oracle and input as specified, runs $\mathcal{A}$ with quantum access to the entire oracle, and returns $\mathcal{A}$'s output density matrix.

$\mathsf{H}_0$ Sample $\mathcal{O}$ and $k \leftarrow \{0,1\}^\lambda$, and supply $\rho^{\otimes r}_{R_k,S_k} \otimes \zeta_{\Theta_k}$, using $\mathsf{Prepare}^{R_k,S_k}$ for each of the $r$ output factors.

$\mathsf{H}_1$ Sample $\mathcal{O}, k$ as above and supply $\sigma_{\Theta_k}$ instead of $\rho^{\otimes r}_{R_k,S_k} \otimes \zeta_{\Theta_k}$.

$\mathsf{H}_2$ Sample $\mathcal{O}$ and an independent uniform $\Theta : \{0,1\}^{5\lambda} \to \{0,1\}$, and supply $\sigma_\Theta$.

For a joint sample $(\mathcal{O}, \mathcal{O}', \Theta)$ and a supported prefix $o = \mathcal{O}_{\le L}$, write

$$g_u(o, \theta) = \Pr\big[\mathcal{O}(u) \ne \mathcal{O}'(u) \mid \mathcal{O}_{\le L} = o, \Theta = \theta\big]. \tag{93}$$

The following coupling bounds this disagreement even after the old oracle and the supplied phase function are known.

**Lemma 32** (Conditional oracle disagreement)**.** *There is a joint sample $(\mathcal{O}, \mathcal{O}', \Theta)$ such that $(\mathcal{O}, \Theta)$ has the independent sampling in $\mathsf{H}_2$, $(\mathcal{O}', \Theta)$ has the sampling $(\mathcal{O}, \Theta_k)$ in $\mathsf{H}_1$, and*

$$\mathbb{E}_{o,\Theta} \max_{|u|\le L} g_u(o, \Theta) \le 2^{-\Omega(\lambda)}.$$

*The bound is uniform in the auxiliary oracle $\mathcal{O}_{\mathrm{aux}}$.*

The proof of Lemma 32 appears in Appendix I.2.

We now use conditional oracle disagreement to finish the main proof. For each fixed oracle and key, changing one tensor factor at a time gives

$$\tfrac{1}{2}\left\|\rho_{R_k,S_k}^{\otimes r}\otimes\zeta_{\Theta_k}-\sigma_{\Theta_k}\right\|_1\le r\tfrac{1}{2}\left\|\rho_{R_k,S_k}-|\Omega_{\Theta_k}\rangle\langle\Omega_{\Theta_k}|\right\|_1.$$

Conditional on $k,\Theta_k$, the preparation pair has distribution $\mathcal{D}_{\mathrm{prep}}(|\Omega_{\Theta_k}\rangle)$. Applying $\mathcal{A}^{\mathcal{O}}$ cannot increase trace distance, so the state-preparation bound (Lemma 17) gives

$$\tfrac{1}{2}\left\|\mathsf{H}_0-\mathsf{H}_1\right\|_1\le r2^{-5\lambda}.$$

For the second transition, use the coupling from conditional oracle disagreement (Lemma 32). Conditional on $(o,\Theta)$, the query states of $\mathcal{A}^o(\sigma_\Theta)$ are fixed, so each query has expected weight at most $\max_{|u|\le L} g_u(o,\Theta)$ on inputs where the two oracles differ. The quantum query bound (Lemma 5) therefore gives

$$\tfrac{1}{2}\left\|\mathsf{H}_1-\mathsf{H}_2\right\|_1\le 2Q\sqrt{\mathbb{E}_{o,\Theta}\max_{|u|\le L}g_u(o,\Theta)}=2^{-\Omega(\lambda)}. \tag{94}$$

The maximum allows queries to depend on every supplied copy and on $\zeta_\Theta$. Averaging $\mathcal{O}_{\mathrm{aux}}$ and adding the two errors proves the theorem, including its allowed auxiliary states.

## I.2 Proof of conditional oracle disagreement

We now prove Lemma 32 by fixing the auxiliary oracle $\mathcal{O}_{\mathrm{aux}}$ and constructing the joint sample whose conditional disagreement we will bound.

### I.2.1 Joint sampling of the recursive oracles

We replace the preparation pair at one hidden key and sample the new recursive answers jointly with the old answers. The extra randomness in this joint sampling will let us change the target while conditioning on the entire old oracle. Choose a constant $d\ge 1$ with $L<\lambda^{2^{d+1}}$, put $m=d+1$, and process interval $i\in[d]$ only at query lengths

$$\lambda^{2^i}\le|u|<\min\{L+1,\lambda^{2^{i+1}}\}.$$

Set

$$\rho_{P,i}=2^{-c_P\lambda(256m)^{-(i-1)}},\qquad \gamma_{P,i}=\rho_{P,i}^{1/128},\qquad t_i=\gamma_{P,i}/12\quad(i\in[m]).$$

The constant $c_P>0$ will be chosen below. The following definition specifies the distribution before we construct an implementation.

**Definition 42** (Joint oracle sample at a hidden key)**.** *We define the* joint oracle sample $(\mathcal{O},\mathcal{O}',\Theta)$ *as follows.*

1. *Sample independent $C_k=(\Theta_k,R_k^{\mathsf{prs}},S_k^{\mathsf{prs}})$ using the phase and preparation sampling above. Choose $k\leftarrow\{0,1\}^\lambda$ and an independent $C_k'$, and set $\Theta$ to its phase function. The old base uses the preparation pairs in $(C_j)_j$, and the new base replaces the pair at $k$ by that in $C_k'$.*
2. *Use the same recursive prefix below $\lambda^2$, then process query inputs in increasing length, computing each $N$ answer exactly from its own shorter oracle.*

3. *At an $A$ input in interval $i$ with old and new means $p_0, p_1$, sample the pair of bits from the matrix*

$$K_{t_i}(p_0,p_1) = M(p_0,p_1) + t_i J, \qquad J = \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix},$$
$$M(p_0,p_1) = \begin{pmatrix} 1-\max\{p_0,p_1\} & (p_1-p_0)_+ \\ (p_0-p_1)_+ & \min\{p_0,p_1\} \end{pmatrix}, \tag{95}$$

*with rows indexed by the old bit and columns by the new bit. Beyond length $L$, use any causal joint sampling with the prescribed marginals.*

Observe that the shared prefix uses only base inputs shorter than $\lambda$: a reserved input of length $(n+2)^2 < \lambda^2$ describes queries of length at most $n < \lambda$. At every later $A$ input, $p_0, p_1 \in [1/3, 2/3]$ and $0 < t_i \le 1/6$ make $K_{t_i}$ a positive probability matrix. Since each row and column of $J$ sums to zero, its marginals are Bernoulli with means $p_0, p_1$, so induction on query length gives the original recursive distribution for each oracle.

### I.2.2 An implementation that leaves the key hidden

We implement the joint sample using classical queries, so disagreement on any input can be charged to a query at the changed key or at a sparse set of recursive inputs.

For each of the $N_* = 2^\lambda$ keys, sample two independent components $C_{k,0}, C_{k,1}$. To select a component, independently sample $m$ sets $P_1, \dots, P_m$, including each key with probability $\rho_{P,1}$, and independent fair bits $B_1(k), \dots, B_m(k), Z(k)$. Set

$$X_k = \begin{cases} B_{j(k)}(k), & j(k) = \min\{i : k \notin P_i\} < \infty, \\ Z(k), & k \in P := \bigcap_i P_i, \end{cases} \qquad \Pr[k \in P] = p := \rho_{P,1}^m,$$

and use the preparation pair in $C_{k,X_k}$ in the old base. Conditional on the sets, the selectors are independent fair bits, so the selected and unused component arrays are independent of each other and of the sets. A preparation-oracle query at key $k$ reads $X_k$ and returns the requested bit of the pair in $C_{k,X_k}$. Its coherent implementation uses two selector queries, including uncomputation. This substitution preserves deterministic verification, polynomial query and witness bounds, and the shorter-query order because selector inputs have length $\lambda$.

Let $Z_{\mathsf{base}}$ denote the component arrays, selectors and the values of $\mathcal{O}_{\mathrm{aux}}$, and put $\nu_P = \rho_{P,m}$. We use the following implementation before selecting which $k \in P$ to replace, writing $(\mathcal{O}_0, \mathcal{O}_1) = (\mathcal{O}, \mathcal{O}')$.

**Lemma 33** (Classical implementation of coupled oracles)**.** *The joint oracle sample $(\mathcal{O}, \mathcal{O}', \Theta)$ (Definition 42) has an implementation with a classical description $E$ and a deterministic query algorithm $\mathsf{Answer}_E^{\mathbf{T}_e}$, where $\mathbf{T}_e = (T_{0,e}, \dots, T_{d,e})$ gives the unexposed selector answers and the successive intervals' $A$ answers. Averaging its coins, including failed stages, gives the original distribution of the old oracle and*

$$\mathcal{O} \perp \big((C_{k,1-X_k})_k, (P_i)_{i\in[m]}\big) \mid (C_{k,X_k})_k.$$

*When any $k \in P$ is replaced by its unused component,*

$$\mathcal{O}_e(u) = \mathsf{Answer}_E^{\mathbf{T}_e}(u) \quad (e \in \{0,1\},\ |u| \le L)$$

*outside an $E$-determined event $\mathsf{Fail}$ of probability at most $3d\nu_P^4$. Each execution uses at most $h = \mathsf{poly}(\lambda)$ queries for every choice of Boolean oracle functions, and changing that component does not change $E$. Write $T_j = T_{j,0}$ for the old oracle. On success, conditional on $E, Z_{\mathsf{base}}$ and the*

*earlier functions $T_{<j}$, each remaining Boolean function $T_j$, $j \geq 1$, is sampled independently at each input by*

$$G_j(u) \leftarrow \text{Bernoulli}(\pi_j), \qquad B_j(u) \leftarrow \text{Bernoulli}(1/2),$$
$$C_j(u) \leftarrow \text{Bernoulli}(c_{j,u}(T_{<j})), \qquad \pi_j \leq \nu_P,$$
$$T_j(u) = (1 - G_j(u))B_j(u) + G_j(u)C_j(u).$$

The proof of Lemma 33 appears in Appendix I.4.

### I.2.3 Disagreement conditional on the old oracle

We now select the changed key and bound each answer's probability of changing after all old answers have been fixed. Generate the old oracle using its classical implementation (Lemma 33), choose $k$ uniformly from $P$, and complete that implementation with the unused component at $k$ to obtain the new oracle. If $P = \varnothing$, choose $k \leftarrow \{0,1\}^\lambda$ and use the conditional rows of $K_{t_i}$ directly.

After averaging implementation coins, the old oracle is independent of the unused components and of the selector sets, while the selected and unused arrays are independent. Symmetry of the keys consequently gives

$$k \leftarrow \{0,1\}^\lambda, \qquad C'_k \perp (k, \mathcal{O}_{\mathsf{base}}, \mathcal{O}_{\leq L}), \qquad \Theta \perp \mathcal{O}.$$

At each later input the pair has matrix $K_{t_i}$, so this sampling gives the joint oracle sample $(\mathcal{O}, \mathcal{O}', \Theta)$ (Definition 42).

To bound a query chosen from the old oracle, we first check that conditioning on its actual answers does not reveal the remaining recursive sets.

**Claim 11** (Conditioning on the old answers)**.** *On success, for every $j \geq 1$ and input $w$,*

$$\Pr[G_j(w) = 1 \mid E, Z_{\mathsf{base}}, T_0, \ldots, T_d] \leq 4\nu_P.$$

*The bound also holds after averaging $Z_{\mathsf{base}}$, for an input chosen from $E$ and $T_0, \ldots, T_d$.*

*Proof.* Condition on $E, Z_{\mathsf{base}}, T_{<j}$. The classical implementation (Lemma 33) gives $\Pr[G_j(w) = 1] \leq \nu_P$. On $G_j(w) = 0$, the bit $T_j(w)$ is fair, so either value has probability at least $(1 - \nu_P)/2$. Bayes' rule therefore gives

$$\Pr[G_j(w) = 1 \mid E, Z_{\mathsf{base}}, T_{<j}, T_j(w) = x] \leq \frac{2\nu_P}{1 - \nu_P} \leq 4\nu_P.$$

Other inputs use independent pairs of bits, so conditioning on the whole function $T_j$ does not change this bound. Later correction probabilities depend on $G_j$ only through $T_j$, and hence conditioning on $T_{j+1}, \ldots, T_d$ also leaves the bound unchanged. It holds for every $w$ with these functions fixed, which allows $w$ to depend on them and permits averaging $Z_{\mathsf{base}}$. □

Since $|P|$ is binomial with mean $N_* p$, Chernoff's bound gives

$$\Pr[|P| < N_* p/2] \leq e^{-N_* p/8}.$$

Put $\mathsf{Bad} = \mathsf{Fail} \cup \{|P| < N_* p/2\}$ and $\xi = \Pr[\mathsf{Fail}] \leq 3d\nu_P^4$. Fix $E, T_0, \ldots, T_d$ and follow $\mathsf{Answer}_E(u)$ with the old answers. For a selector query, condition also on $P$: the changed key is uniform in $P$, so its probability of being queried is at most $1/|P| \leq 2/(N_* p)$ when $|P| \geq N_* p/2$. For a query to $T_j$, $j \geq 1$, instead use the conditional sparse-set bound (Claim 11), which bounds its probability of lying in $G_j$ by $4\nu_P$ conditional on $E, T_0, \ldots, T_d$, without fixing $P$.

Outside Bad, an answer can change only if one of the at most $h$ queries encounters one of these two events. Averaging the respective conditional bounds and taking a union bound therefore gives, for every supported $o = \mathcal{O}_{\le L}$,

$$\begin{aligned} \Pr\big[\mathcal{O}(u) \neq \mathcal{O}'(u) \mid \mathcal{O}_{\le L} = o\big] \le b(o) &:= \frac{2h}{N_* p} + 4h\nu_P + \Pr[\mathsf{Bad} \mid \mathcal{O}_{\le L} = o], \\ \mathbb{E}_o b(o) &\le \frac{2h}{N_* p} + 4h\nu_P + \xi + e^{-N_* p/8}. \end{aligned} \tag{96}$$

Choosing $mc_P < 1/2$ makes $\eta := \mathbb{E}_o b(o) = 2^{-\Omega(\lambda)}$. The same $b(o)$ works for every input $u$, and the last term covers the empty-set case as well.

### I.2.4 Queries chosen from copies of the target

Because $\Theta \perp \mathcal{O}$, the preceding estimate gives $\mathbb{E}_\Theta g_u(o, \Theta) \le b(o)$ for every $o, u$. We must make this bound simultaneous over query inputs that may depend on $\sigma_\Theta$. Put $D = 2^{5\lambda}$ and $H = \lfloor D^{3/4} \rfloor$. The following claim lets us pass from an average over phases to that simultaneous bound.

**Claim 12** (Changing the phase function)**.** *For sufficiently small $c_P > 0$, there is $\epsilon = 2^{-\Omega(\lambda)}$ such that, for every supported old prefix $o$, every $|u| \le L$, and all phase functions $\theta, \theta'$,*

$$d_H(\theta, \theta') \le H \quad \Longrightarrow \quad |g_u(o, \theta) - g_u(o, \theta')| \le \epsilon. \tag{97}$$

The proof of Claim 12 appears in Appendix I.3.

**Concentration over the phases.** For fixed $o, u$, Markov's inequality gives

$$S = \{\theta : g_u(o, \theta) \le 2b(o)\}, \qquad \Pr_\Theta[\Theta \in S] \ge \tfrac{1}{2}.$$

When $b(o) = 0$, nonnegativity gives $g_u(o, \Theta) = 0$ almost surely, so this conclusion also holds. The function $d_H(\Theta, S)$ changes by at most one when one bit changes. Bounded differences therefore implies

$$\frac{1}{2} \le \Pr[d_H(\Theta, S) = 0] \le \exp\left(-\frac{2(\mathbb{E} d_H(\Theta, S))^2}{D}\right), \qquad \mathbb{E} d_H(\Theta, S) \le \sqrt{D \ln 2/2}.$$

For sufficiently large $\lambda$, $H \ge 2\sqrt{D \ln 2/2}$, so a second application gives

$$\Pr[d_H(\Theta, S) > H] \le \delta := \exp\big(-H^2/(2D)\big).$$

By the phase-change bound (Claim 12), every phase function within distance $H$ of $S$ satisfies $g_u(o, \Theta) \le 2b(o) + \epsilon$. There are fewer than $2^{L+1}$ query inputs of length at most $L$, so a union bound and $g_u \le 1$ give

$$\mathbb{E}_{o,\Theta} \max_{|u| \le L} g_u(o, \Theta) \le 2\eta + \epsilon + 2^{L+1}\delta = 2^{-\Omega(\lambda)}. \tag{98}$$

The doubly exponential bound on $\delta$ allows every polynomial $L$.

The exact marginals of the coupled oracles and the bound on the conditional maximum prove the lemma.

## I.3 Proof of the phase-change bound

We now prove Claim 12. We identify the distribution of the new oracle conditional on the old oracle, so that the target-change bound applies with the old answers fixed.

**Conditional sampling of the new oracle.** Fix a supported old prefix $o$. For an $A$ input in interval $i$, let $x$ be its old answer and $p \in [1/3, 2/3]$ its old mean, both determined by $o$. For a new acceptance probability $v$, put $q(v) = (1+v)/3$. Dividing row $x$ of $K_{t_i}(p, q(v))$ by its sum gives the new conditional mean

$$\begin{aligned} h_1(v) &= \min\{1, q(v)/p\} - t_i/p, \\ h_0(v) &= \max\{0, (q(v)-p)/(1-p)\} + t_i/(1-p). \end{aligned} \tag{99}$$

Thus, conditional on $o$ and the new base, generate the new oracle in increasing query length, answering $N$ exactly from its shorter oracle and sampling each $A$ bit with mean $h_x(v)$. To verify this whole conditional distribution, factor each paired answer in a finite prefix as

$$K_{t_i}(p, q(v))_{x,y} = p^x (1-p)^{1-x} h_x(v)^y (1-h_x(v))^{1-y}.$$

Conditioning on $o$ cancels the old factors, whose product is its marginal probability. Old exact $N$ answers contribute one on the supported prefix, leaving precisely the new exact $N$ constraints. Since $\mathcal{O}$ is independent of the new component $C'_k$, its preparation pair still has distribution $(R, S) \leftarrow \mathcal{D}_{\mathrm{prep}}(|\Omega_\Theta\rangle)$ given $\Theta$. The shared prefix below $\lambda^2$ queries only base inputs shorter than $\lambda$, so it is the same for both oracles.

Since $p, 1-p \geq 1/3$ and $t_i \leq 1/12$, the conditional means satisfy

$$|h_x(v) - h_x(w)| \leq |v - w|, \qquad t_i \leq h_x(v) \leq 1 - t_i \quad (v, w \in [0,1]).$$

These are exactly the mean and margin hypotheses needed to change the new preparation target while keeping $o$ fixed.

**Changing the phase function.** Let $D = 2^{5\lambda}$ and $H = \lfloor D^{3/4} \rfloor$. For functions $\theta, \theta'$ with $d_H(\theta, \theta') \leq H$,

$$\| \, |\Omega_\theta\rangle - |\Omega_{\theta'}\rangle \, \|_2 = 2\sqrt{d_H(\theta, \theta')/D} \leq 2\sqrt{H/D} \leq 2^{-\lambda/9}$$

for sufficiently large $\lambda$. Fix $o$, the changed key $k$, and all base answers outside $k$. Define the test $\mathsf{T}_e$, for $e \in \{0,1\}$, by the following operations:

1. Sample the changed component with target $|\Omega_\theta\rangle$ if $e = 0$ and $|\Omega_{\theta'}\rangle$ if $e = 1$.
2. Sample its recursive extension with exact $N$ answers, the fixed shared prefix, and conditional means $h_x(v)$ (Equation (99)).
3. Query the new oracle at $u$, obtain $a$, and return $\mathbf{1}[a \neq o(u)]$.

Apply the target-change bound (Theorem 13) to the oracles sampled by $\mathsf{T}_0, \mathsf{T}_1$ and the one-query algorithm in Step 3, with $c_* = 1/9$ and $n = 5\lambda$. The conditional means satisfy the displayed difference and range bounds. If $c$ is the theorem's margin constant, choose $c_P$ so that $c_P/128 < c$ and $mc_P < 1/2$. Then

$$\min_i t_i = \frac{1}{12} 2^{-c_P \lambda/128} \geq 2^{-c\lambda}$$

for sufficiently large $\lambda$. The test uses the fixed bit $o(u)$ after its query. The theorem gives $|\Pr[\mathsf{T}_0 = 1] - \Pr[\mathsf{T}_1 = 1]| \leq \epsilon$ for $\epsilon = 2^{-\Omega(\lambda)}$, uniformly over the fixed quantities. The changed key and the base answers outside it, conditioned on $o$, are independent of the new phase function. Averaging these fixed choices therefore proves the phase-change bound (Equation (97)).

## I.4 Proof of the classical simulation of coupled oracles

We now prove Lemma 33 by constructing the classical query algorithm $\mathsf{Answer}_E$ and its common description $E$. At the beginning of interval $i$, the representation of each existing oracle function uses $m-i+1$ independent sets with joint inclusion bound $\rho_{P,i}^{|S|}$. The description $E$ specifies exposed sets and finite functions of query transcripts, and implements every earlier answer for arbitrary oracles for shorter inputs. The selector sampling establishes this invariant at the first interval. The following stages extend it.

**Classical query algorithms for the next interval.** For each valid $N$ or $A$ query at address $u$, write $\langle M_u, x_u\rangle$ for the padded encoding of its circuit and input, so $M_u$ is deterministic for $N$ and witnessless quantum for $A$. Address padding restricts both types of circuit to shorter queries to the full oracle. For each existing oracle function $j$, expose its next set $P_{i,j}$ and uniform Boolean function $B_{i,j}$, and define

$$P_i = \{(j,x) : x \in P_{i,j}\}, \qquad B_i(j,x) = B_{i,j}(x).$$

Independence of the exposed sets gives $P_i$ inclusion bound $\rho_{P,i}^{|S|}$. For each $A$ query, substitute the earlier answer algorithms and component selection into its defining circuit, giving

$$v_u(Y) = \Pr\left[M_u^{\mathsf{Answer}_{<i}^{(B_i)_{P_i}^{Y}}}(x_u) = 1\right].$$

For the sampled oracle, $Y(j,x)$ returns the bit of the first not-yet-exposed Boolean function whose corresponding set omits $x$. If every remaining set contains $x$, it returns the correction bit. The function $(B_i)_{P_i}^{Y}$ equals $Y$ on $P_i$ and $B_i$ elsewhere. The following query algorithms will be stored in $E$ before any actual shorter answers are used.

**Claim 13** (Query algorithms for one interval)**.** *Conditional on the preceding description, the exposed $(P_i, B_i)$ determine classical algorithms $T_u, \tau_u$ and numbers $\widehat{v}_{u,\tau} \in [0,1]$. The algorithms have polynomial query bounds on every oracle, and except with probability $3\rho_{P,i}^4$ over $(P_i, B_i)$, simultaneously for every address $u$ and every $Y$, $T_u^Y$ gives the exact existential answer at each $N$ address, while at each $A$ address*

$$|v_u(Y) - \widehat{v}_{u,\tau_u(Y)}| \le \gamma_{P,i}/4000.$$

The proof of Claim 13 appears in Appendix I.4.1.

Add these algorithms and numbers to $E$. If the simultaneous guarantee fails, use the completion rule below. Otherwise it applies to both sets of shorter answers $Y_0, Y_1$, allowing us to construct the paired $A$ answers next.

**Sampling the next $A$ answers.** For each $A$ input $u$ set

$$q_{u,\tau} = \frac{1+\widehat{v}_{u,\tau}}{3}, \qquad p_e = \frac{1+v_u(Y_e)}{3}, \qquad q_e = q_{u,\tau_u(Y_e)}, \qquad \delta_i = \frac{2\gamma_{P,i}}{3}.$$

The estimate gives $q_e \in [1/3, 2/3]$ and $\max_e |p_e - q_e| \le \delta_i/1000$. Independently at each $u$, sample $F_u \leftarrow \text{Bernoulli}(2/3)$ and $V_u \leftarrow [0,1]$, and put in $E$ the bit $F_u$ and the finite function $L_u(\tau) = \mathbf{1}[V_u \le \widehat{v}_{u,\tau}]$. For the new oracle function, sample $m-i$ independent sets of density $\gamma_{P,i}^{1/(m-i)}$ and

independent uniform Boolean functions. Independently at each input $u$, sample a correction pair $(C_{i,0}(u), C_{i,1}(u))$ from

$$R_i = D_{\text{fair}} + J/8 + \frac{M(p_0,p_1) - M(q_0,q_1)}{\delta_i}, \qquad D_{\text{fair}} = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix}. \tag{100}$$

To sample the old oracle before choosing the changed key, first sample only the old bit with mean $1/2 + (p_0 - q_0)/\delta_i$. Once the key and new shorter answers are chosen, sample the new bit from the row of $R_i$ indexed by the old bit, divided by that row's sum. Sample these correction pairs also where they will not be used. If $F_u = 0$, run $\tau_u$ with oracle $Y_e$ and output $L_u(\tau_u(Y_e))$. Otherwise, find the first new set that omits $u$ and return the value at $u$ of its corresponding uniform Boolean function, or return $C_{i,e}(u)$ if all new sets contain $u$. Compute each exact $N$ answer separately from its own shorter oracle.

To check the sampling, observe that using the same threshold $V_u$ for the two transcript estimates gives

$$\tfrac{1}{3} M(\widehat{v}_{u,\tau_u(Y_0)}, \widehat{v}_{u,\tau_u(Y_1)}) + \tfrac{2}{3} D_{\text{fair}} = M(q_0,q_1).$$

The event requiring a correction has probability $\delta_i$ and would contribute $\delta_i D_{\text{fair}}$ to this matrix. Replacing that contribution by $\delta_i R_i$ is valid because

$$D_{\text{fair}} + J/8 = \begin{pmatrix} 3/8 & 1/8 \\ 1/8 & 3/8 \end{pmatrix}, \qquad \|M(p_0,p_1) - M(q_0,q_1)\|_{\max} \le \delta_i/500,$$

so $R_i$ is a positive probability matrix. Its old marginal is the one used above, and $t_i/\delta_i = 1/8$ gives the required distribution

$$M(q_0,q_1) - \delta_i D_{\text{fair}} + \delta_i R_i = K_{t_i}(p_0,p_1).$$

In particular, sampling the old answer does not require the new component or its shorter oracle.

**Completing the construction and answering queries.** After this stage the representation of each earlier function has one fewer set, and the new function uses $m - i$ sets whose density satisfies

$$\gamma_{P,i}^{1/(m-i)} = \rho_{P,i}^{1/(128(m-i))} \le \rho_{P,i+1}.$$

The functions $L_u$ and estimates $\widehat{v}_{u,\tau}$ are defined for every possible partial function $\tau$ and for $\tau = \perp$, without using the remaining sets or correction bits. This remains true at later stages, whose algorithms use arbitrary oracle functions on shorter inputs. Conditional on $E$, the selected base and the earlier actual functions $T_{<j}$, the last sets and their fair and correction bits therefore have exactly the product distribution in the statement.

If a stage fails, set $\mathsf{Fail}$ in $E$, sample subsequent old answers with their prescribed means, and, after choosing the key, sample the new answers from the appropriate rows of $K_{t_i}$. Thus failure changes the implementation but never the prescribed oracle distribution. The union of the three failure events per interval has probability at most $3d\nu_P^4$.

For completeness, the final algorithm uses the following explicit oracle access to the function introduced at stage $j$. Number its sets and uniform Boolean functions by $(G_j^{(r)}, B_j^{(r)})_{r\in[m-j]}$, with those for $r < m - j$ in $E$. For $j \ge 1$, put $G_j = G_j^{(m-j)}$, $B_j = B_j^{(m-j)}$ and

$$T_{j,e}(u) = \begin{cases} C_{j,e}(u), & u \in G_j, \\ B_j(u), & u \notin G_j. \end{cases}$$

For $j = 0$, use $B_m$ outside $P_m$ and $Z$ inside $P_m$, flipping only the changed key for the new selector function $T_{0,1}$. Define $\mathsf{Read}_{j,r,E}^{T_{j,e}}(u)$ by two steps:

1. For $\ell = r, \dots, m-j-1$, return $B_j^{(\ell)}(u)$ at the first $\ell$ with $u \notin G_j^{(\ell)}$.

2. If every tested set contains $u$, query $T_{j,e}(u)$ and return its answer.

The first step uses only functions in $E$, and the second makes one oracle query. In an algorithm constructed for interval $i$, answer $Y(j,v)$ with $\mathsf{Read}^{T_{j,e}}_{j,i-j+1,E}(v)$. Let $T_u^E$ and $\tau_u^E$ be the resulting existential and transcript algorithms, whose only oracles are $(T_{j,e})_j$.

On input $u$, $\mathsf{Answer}_E^{\mathbf{T}_e}$ returns zero if $|u| > L$ or $\mathsf{Fail}$ is set. Otherwise it performs the applicable step:

1. For an $\mathcal{O}_{\mathrm{aux}}$ or shared-prefix input, return its bit in the corresponding fixed function in $E$.

2. For a preparation-oracle input with key $k$, compute $x = \mathsf{Read}^{T_{0,e}}_{0,1,E}(k)$ and return the requested bit of the pair in $C_{k,x}$.

3. For an $N$ input, execute $(T_u^E)^{\mathbf{T}_e}$ and return its output.

4. For an $A$ input in interval $i$, return $L_u(\tau_u^E(\mathbf{T}_e))$ if $F_u = 0$, and otherwise return $\mathsf{Read}^{T_{i,e}}_{i,1,E}(u)$.

Unused inputs and invalid $N, A$ encodings return zero. Strictly shorter recursive calls and constantly many intervals give termination and the polynomial query bound $h$ for every choice of its Boolean oracle functions.

Averaging the fresh implementation coins at each input gives old mean $p_0$ independently of unused components and the initial sets. Induction over query length proves the original conditional distribution of the entire old oracle, including failed stages. Finally, no exposed base set reveals a selector in $P$, and all later functions in $E$ are specified for every possible $\tau$, including $\perp$. Changing a selected component at any $k \in P$ therefore leaves $E$ unchanged, as required.

### I.4.1 Proof of the interval query algorithms

We now prove Claim 13, fixing the preceding description and the substituted circuits. The exposed restriction $(P_i, B_i)$ is independent of these circuits, with inclusion bound $\rho_{P,i}^{|S|}$ and independent uniform $B_i$.

Set $\epsilon = \gamma_{P,i}/4000$. Let $Q_i \geq 1$ bound the substituted query count, let $a_i$ bound the logarithm of the number of circuit indices, and let $\mathcal{X}_i$ be their query domain. Choose

$$s = a_i + \lambda^2, \qquad M_{\mathrm{trial}} = \lceil \rho_{P,i}^{-1/10} \rceil, \qquad Q_{\mathrm{sim}} = s^2 M_{\mathrm{trial}}.$$

Let $\mathsf{Sim}_u$ be the classical-query simulator (Theorem 5) for the circuit at $u$, with fixed function $B_i$, random set $P_i$, no witness and accuracy $\epsilon$. Implement its deterministic query phase using the partial function $\tau = \varnothing$ on $P_i$:

1. At a requested input $x \notin P_i$, supply $B_i(x)$. For $x \in \operatorname{dom}\tau$, supply $\tau(x)$.

2. At any other input, return $\perp$ if $|\operatorname{dom}\tau| = s-1$. Otherwise query $Y(x)$, set $\tau(x)$ to its answer, and continue.

3. Return $\tau$ when the query phase ends.

For each partial function $\tau$ on $P_i$ with fewer than $s$ inputs, run the same $\mathsf{Sim}_u$ with answers $\tau(x)$ on its domain and $B_i(x)$ elsewhere, and define $\widehat{v}_{u,\tau}$ to be its output-one probability. Put $\widehat{v}_{u,\perp} = 0$. This defines $\tau_u$ with fewer than $s$ queries to $Y$ on every execution.

Conditional on $B_i$, the joint-inclusion bound for $P_i$ is unchanged. The message length is

$$m_{\mathrm{msg},i} = a_i + s(\lceil \log_2 Q_{\mathrm{sim}} \rceil + 1) = \mathsf{poly}(\lambda).$$

Since $\log_2 Q_{\mathrm{sim}} \le \frac{1}{10}\log_2(1/\rho_{P,i}) + O(\log\lambda)$ and $s = a_i + \lambda^2$, for sufficiently large $\lambda$,

$$m_{\mathrm{msg},i} + 8\lambda^2 \le s\log_2(1/\rho_{P,i}).$$

Contractive simultaneous incompressibility (Lemma 8) therefore gives $t_{\mathrm{inc}}(\lambda, m_{\mathrm{msg},i}) \le s$ for $P_i$. Deterministic-classical-query simulation (Theorem 5), with index length $a_i$ and no witness, bounds the probability of any truncated or inaccurate execution by

$$2^{-8\lambda^2} + 2^{a_i}(|\mathcal{X}_i|+1)^s s e^{-\epsilon^2 M_{\mathrm{trial}}/(16Q_i^2)} \le \rho_{P,i}^4 + 2^{-4\lambda^2} \le 2\rho_{P,i}^4.$$

The second term is at most $2^{-4\lambda^2}$ because $\epsilon^2 M_{\mathrm{trial}}/(16Q_i^2) = 2^{\Omega(\lambda)}$, whereas the logarithm of its prefactor is polynomial. Outside this event, the returned $\tau$ determines every answer in the simulator's deterministic query phase, so its output probability is $\widehat{v}_{u,\tau_u(Y)}$. Consequently,

$$\sup_Y |v_u(Y) - \widehat{v}_{u,\tau_u(Y)}| \le \gamma_{P,i}/4000.$$

For an $N$ input $u$, let $f_u$ be the accepting-transcript DNF of $M_u(x_u, w)$ after the same substitutions (Lemma 10), whose width $k$ is polynomial. Define $T_u = \mathsf{Switch}_{f_u,s}(P_i, B_i|_{P_i^c})$, and evaluate it by forwarding its queries to $Y$. The switching theorem (Theorem 7) applies with bias parameter 1 because $B_i$ is independent and uniform. The same cutoff $s = a_i + \lambda^2$ satisfies, for sufficiently large $\lambda$,

$$k + s \le 2^\lambda, \qquad \rho_{P,i} \le (16k)^{-1}, \qquad 2^{a_i}\, 2(8k\rho_{P,i})^s \le \rho_{P,i}^4.$$

It therefore supplies exact $N$ answers simultaneously for every $u, Y$, with the stated failure probability. Both algorithms depend only on the exposed restriction and have polynomial query bounds also when they fail.

# Appendix III: Proofs of Fixed-Oracle Separations

## J Proof of the Fixed-Oracle Collapse and Separation (Theorem 11)

We prove the QMA separation and QCMA collapse relative to one fixed classical oracle.

### J.1 Proof of Theorem 11

We prove the fixed-oracle collapse and separation of Theorem 11.

The oracle $\mathcal{O}_{\mathsf{sep}}$ has inputs $\mathsf{base}(\lambda, v, y)$ of length $\ell_\lambda = (\lambda + c_0)^{10}$ and inputs $\mathsf{help}(a)$ of length $(|a| + 2)^2$, for a sufficiently large constant $c_0$. Here the query pairs range over $(v, y) \in \mathcal{X}_\lambda = \Sigma_\lambda^{q_\lambda} \times \{0,1\}^\lambda$ for the linear code $C_\lambda \subseteq \Sigma_\lambda^{q_\lambda}$ (Definition 13), and $a = \langle V, x\rangle$ is a padded description of a quantum verifier $V$ with ordinary input $x$ and classical witness $w$, using at most $|a|$ gates, witness bits and query length. Use injective polynomial-time encodings and zero on malformed inputs. Put

$$\mathcal{O}_{\mathsf{sep},\lambda}(v, y) = \mathcal{O}_{\mathsf{sep}}(\mathsf{base}(\lambda, v, y)), \qquad \alpha(\mathcal{O}) = \max_\sigma \Pr\Big[V_{\mathsf{QMA}}^{\mathcal{O}}(1^\lambda, \sigma) = 1\Big], \tag{101}$$

where $V_{\mathsf{QMA}}$ is the QMA verifier (Theorem 8). Require

$$\mathcal{O}_{\mathsf{sep}}(\mathsf{help}(\langle V, x\rangle)) = \begin{cases} 1, & \max_w \Pr\big[V^{\mathcal{O}_{\mathsf{sep}}}(x, w) = 1\big] \geq 2/3, \\ 0, & \max_w \Pr\big[V^{\mathcal{O}_{\mathsf{sep}}}(x, w) = 1\big] \leq 1/3, \end{cases} \tag{102}$$

with either answer allowed in the gap. These requirements depend only on shorter oracle inputs.

A finite prefix $\mathcal{O}_{\mathsf{sep},\leq N}$ is *admissible* if its QCMA answers satisfy Equation (102) and $\alpha(\mathcal{O}_{\mathsf{sep},\lambda}) \in [0, 1/3] \cup [2/3, 1]$ whenever $\ell_\lambda \leq N$.

The following extension lemma lets us diagonalize against a classical algorithm while preserving admissibility.

**Lemma 34** (An extension preserving QCMA answers)**.** *Let $\mathcal{O}_{\mathsf{sep},\leq N}$ be admissible and let $M$ be a deterministic polynomial-time oracle algorithm. For every sufficiently large $\lambda$ with $\ell_\lambda > N$, there is an admissible extension $\mathcal{O}'_{\mathsf{sep},\leq N'}$, with $N' \geq \ell_\lambda$ bounding $M$'s query lengths on $1^\lambda$ and all newly assigned base components other than $\lambda$ zero, such that*

$$M^{\mathcal{O}'_{\mathsf{sep}}}(1^\lambda) \neq \mathbf{1}[\alpha(\mathcal{O}'_{\mathsf{sep},\lambda}) \geq 2/3].$$

The proof of Lemma 34 appears in Appendix J.1, after the proof of Theorem 11.

We now apply the extension lemma to prove Theorem 11.

**Constructing the oracle.** Enumerate the deterministic oracle algorithms with explicit polynomial time bounds as $M_1, M_2, \ldots$, and start with the empty admissible prefix $\mathcal{O}_{\mathsf{sep}}^{(0)}$. At stage $i$, choose $\lambda_i$ large enough for Lemma 34 and with $\ell_{\lambda_i}$ beyond the domain of $\mathcal{O}_{\mathsf{sep}}^{(i-1)}$, then let $\mathcal{O}_{\mathsf{sep}}^{(i)}$ be an extension supplied by that lemma against $M_i$. Define the final oracle by

$$\mathcal{O}_{\mathsf{sep}}(u) = \mathcal{O}_{\mathsf{sep}}^{(i)}(u) \quad \text{for any stage } i \text{ whose prefix contains } u.$$

The prefixes extend one another and their lengths tend to infinity, so this defines every answer of $\mathcal{O}_{\mathsf{sep}}$. Each helper requirement depends only on shorter inputs, and hence remains satisfied after the stage at which its answer is fixed.

**Separating the language.** Admissibility gives

$$L = \{1^\lambda : \alpha(\mathcal{O}_{\mathsf{sep},\lambda}) \geq 2/3\} \in \mathsf{QMA}^{\mathcal{O}_{\mathsf{sep}}},$$

because every component has acceptance probability at most $1/3$ or at least $2/3$. On the other hand, the answers queried by $M_i(1^{\lambda_i})$ never change after stage $i$. Thus $M_i^{\mathcal{O}_{\mathsf{sep}}}$ errs on $1^{\lambda_i}$ and $L \notin \mathsf{P}^{\mathcal{O}_{\mathsf{sep}}}$.

**Collapsing the hierarchy.** For any $\mathsf{QCMA}$ verifier $V$ on input $x$, pad $\langle V, x\rangle$ to a polynomial length satisfying the bounds above, then query $\mathsf{help}(\langle V, x\rangle)$ and return its answer. The helper requirement (Equation (102)) makes this a deterministic polynomial-time solver for the verifier's promise problem. Hence $\mathsf{P}^{\mathcal{O}_{\mathsf{sep}}} = \mathsf{QCMA}^{\mathcal{O}_{\mathsf{sep}}}$ also for promise problems. Coherently substituting each lower level's deterministic solver into its parent gives $\mathsf{QCMAH}^{\mathcal{O}_{\mathsf{sep}}} = \mathsf{P}^{\mathcal{O}_{\mathsf{sep}}}$ by induction.

### J.1.1 Extending an admissible oracle

We now prove the extension lemma (Lemma 34).

**Assigning the shorter answers.** Choose a constant $d \geq 0$ so that $M$'s running time on $1^\lambda$ is less than $b_{d+1}$, where

$$b_i = \ell_\lambda^{2^i} \quad (0 \leq i \leq d+1), \qquad N' = b_{d+1} - 1.$$

Keep the existing prefix and set every new base component through $N'$ to zero except component $\lambda$. A helper input of length less than $b_1$ describes a circuit querying only inputs shorter than $b_0$, so these helper answers can be fixed in increasing length order using Equation (102), choosing zero in the gap and leaving every answer in the existing prefix unchanged. Let $T_0$ output these fixed answers without making oracle queries.

Independently sample $H \leftarrow \mathcal{H}_\lambda$ and $B \leftarrow \mathrm{Bernoulli}(2^{-\lceil\sqrt{\lambda}\rceil})^{\mathcal{X}_\lambda}$, and form the coupled target oracles

$$\mathcal{O}^{\mathsf{noise}} = B, \qquad \mathcal{O}^{\mathsf{plant}} = B \vee \mathbf{1}_{\mathcal{P}_H}$$

from Definition 24. Use $d+1$ suffix groups (Definition 17) and put

$$X_0 = \mathcal{X}_\lambda, \qquad X_i = \bigcap_{j \leq i} P_j(H), \qquad e_i = (P_1, \ldots, P_i, B|_{\mathcal{X}_\lambda \setminus X_i}).$$

Here $e_0$ is empty. We next construct total answer algorithms $T_i$ for helper inputs shorter than $b_{i+1}$, whose descriptions depend only on $e_i$ and whose target-oracle queries lie in $X_i$.

**Constructing the helper-answer algorithms.** For $i = 1, \ldots, d+1$, perform the following steps, with $F$ denoting the function $F : X_{i-1} \to \{0,1\}$ used in Step 1.

1. If $i \leq d$, decode $a = \langle V, x\rangle$ and set

$$W_i^F(\mathsf{help}(a), w) = V^{F, T_{i-1}^F}(x, w), \qquad \mathcal{Z}_i = \{z : b_i \leq |z| < b_{i+1}\},$$

rejecting malformed inputs. If $i = d+1$, instead take $\mathcal{Z}_{d+1} = \{1^\lambda\}$ and $W_{d+1}^F = M^{F, T_d^F}$, with no witness. In either case, answer each target query outside $X_{i-1}$ by the stored value $B(u)$ in $e_{i-1}$ and leave only queries in $X_{i-1}$ for the target oracle. All other base answers are the fixed answers chosen above.

2. Put $h_i = \lceil \log_2(2|\mathcal{Z}_i|) \rceil$. On the domain $X_{i-1}$, apply the mapping Replace of Theorem 6 to $W_i$, the remaining set $P_i \cap X_{i-1}$, and the fixed answers $B|_{X_{i-1} \setminus X_i}$, using the conditional distribution given $e_{i-1}$. Call its output $S_i$. If $X_{i-1} = \varnothing$, use the oracle-free completion that returns the non-$\perp$ answer of the BB-QCMA machine for $W_i$ and returns zero in its gap.

3. For $i \le d$, extend the earlier answers by setting, for every function $F : X_i \to \{0,1\}$ and valid helper input $z$,

$$T_i^F(z) = \begin{cases} T_{i-1}^{B|_{X_{i-1} \setminus X_i} \cup F}(z), & |z| < b_i, \\ S_i^F(z), & b_i \le |z| < b_{i+1}. \end{cases}$$

Set all other inputs to zero. In the first branch, answer each query $u \in X_{i-1} \setminus X_i$ with $B(u)$ and forward queries in $X_i$ to $F$. For $i = d+1$, set the output bit to $c = S_{d+1}(1^\lambda)$ and end the construction.

**Verifying each interval.** At interval $i \le d$, the padding gives $|a| < b_i$, so every helper query made by $V(x,w)$ is already covered by $T_{i-1}$. Coherent substitution (Lemma 6) therefore makes $W_i$ a quantum query verifier with polynomial resource bounds on every sample. The returned $S_i$ has fewer than $h_i$ queries, all in $X_i$. Substitution across constantly many intervals gives polynomial query and length bounds for every $T_i$, while Step 3 shows directly that its description depends only on $e_i$. Thus we can apply the same resource bound at every interval, including after an incorrect answer in an earlier interval.

For $i \in [d+1]$, let $N_{W_i}$ be the partial BB-QCMA machine associated with $W_i$. We need the event

$$\mathcal{E}_i = \left\{ \forall z \in \mathcal{Z}_i,\ F \in \{\mathcal{O}^{\mathsf{noise}}, \mathcal{O}^{\mathsf{plant}}\} : \quad N_{W_i}^{F|_{X_{i-1}}}(z) \in \{S_i^{F|_{X_i}}(z), \perp\} \right\}.$$

The following short claim supplies its conditional probability bound.

**Claim 14.** *For every $i \in [d+1]$, $\Pr[\neg \mathcal{E}_i \mid e_{i-1}] \le 2^{-\Omega(\sqrt{\lambda})}$, uniformly in $e_{i-1}$.*

*Proof.* Conditional on $e_{i-1}$, $W_i$ is fixed and

$$B|_{X_{i-1}} \leftarrow \mathrm{Bernoulli}(2^{-\lceil \sqrt{\lambda} \rceil})^{X_{i-1}} \quad \text{independently of } P_i \cap X_{i-1}.$$

The suffix-group bound (Lemma 7(1)) gives joint inclusion at most $2^{-\lambda t/(8(d+1))}$ for this remaining set, while $\mathcal{O}^{\mathsf{noise}}, \mathcal{O}^{\mathsf{plant}}$ agree outside it on $X_{i-1}$. Removing classical witnesses (Theorem 6) therefore applies in Step 2 with the displayed $h_i$, giving $\Pr[\neg \mathcal{E}_i \mid e_{i-1}] \le 2^{-\Omega(\sqrt{\lambda})}$. For an empty domain, $\mathcal{E}_i$ holds exactly. □

The claim holds without conditioning on earlier correctness, so averaging and summing over the constantly many intervals gives

$$\Pr\left[\bigcap_{i=1}^{d+1} \mathcal{E}_i\right] \ge 1 - \sum_{i=1}^{d+1} \mathbb{E}_{e_{i-1}} \Pr[\neg \mathcal{E}_i \mid e_{i-1}] \ge 1 - 2^{-\Omega(\sqrt{\lambda})}.$$

On this event, Step 3 preserves each earlier helper answer under either target oracle $F$, because $F = B$ on $X_{i-1} \setminus X_i$. A verifier described by a helper input in $[b_i, b_{i+1})$ queries only those earlier helpers, so $\mathcal{E}_i$ makes its new answer satisfy Equation (102). Induction on $i$ therefore shows that $T_d$ satisfies every helper requirement through $N'$ under both target oracles. For the terminal interval, $h_{d+1} = 1$, so $S_{d+1}$ makes no queries and both executions of $M$ output its bit $c$.

**Choosing the component on which $M$ errs.** The acceptance bounds for $\mathsf{QMA}$ verification (Theorem 8) give

$$\Pr_{H,B}[\alpha(\mathcal{O}^{\mathsf{plant}}) \geq 2/3 \wedge \alpha(\mathcal{O}^{\mathsf{noise}}) \leq 1/3] \geq 1 - 2^{-\Omega(\lambda)}.$$

For sufficiently large $\lambda$, there is therefore a sample satisfying both promises and all replacement events. Fix such a sample, set

$$\mathcal{O}'_{\mathsf{sep},\lambda} = \begin{cases} \mathcal{O}^{\mathsf{plant}}, & c = 0, \\ \mathcal{O}^{\mathsf{noise}}, & c = 1, \end{cases}$$

and assign every new helper answer through $N'$ using $T_d^{\mathcal{O}'_{\mathsf{sep},\lambda}|_{X_d}}$, with the fixed answers outside $X_d$ stored in its description. The construction agrees with the old prefix, satisfies every new helper requirement and gives the promised $\mathsf{QMA}$ value on every new base component, since the components other than $\lambda$ are zero. Hence $\mathcal{O}'_{\mathsf{sep},\leq N'}$ is admissible, and $M^{\mathcal{O}'_{\mathsf{sep}}}(1^\lambda) = c$ is the wrong answer for its target component, proving the lemma.

# K Proof of the Fixed-Oracle Cryptography Theorem (Theorem 19)

We now prove Theorem 19 by one common Markov–Borel–Cantelli argument.

**Correctness.** The cryptographic oracle $\mathcal{O} = (\mathcal{O}_{\mathsf{base}}, N, A)$ has independent PRF, encryption, signature and PRS base components, with one recursive extension (Definition 37). Together they have at most $2^{2\lambda+2}$ hashes and $2^{\lambda+2}$ preparation targets, each on at least $5\lambda$ qubits. By the good-hash and state-preparation bounds (Lemmas 7 and 17), Markov's inequality, and a union bound, the probability that some hash is bad or some preparation has trace-distance error greater than $2^{-\lambda}$ is at most

$$2^{2\lambda+2-\kappa} + 2^{\lambda+2}2^{\lambda}\epsilon_{\mathrm{prep}}(5\lambda) = 2^{2\lambda+2-\kappa} + 2^{-3\lambda+2}, \qquad \kappa = (\lambda+4)^3.$$

Since this bound is summable, Borel–Cantelli implies that all hashes and preparations are good at sufficiently large parameters for almost every oracle. Functional-key and signature correctness (Lemmas 18 and 24) then give correctness for all four constructions. The same preparation bound holds uniformly over the PRS keys, and replacing $r$ actual outputs by prescribed copies costs at most $r2^{-\lambda}$.

**Security for all classical descriptions.** For each primitive, condition on the other three independent base components and apply its security theorem with those components as $\mathcal{O}_{\mathrm{rest}}$. The theorem includes the single global $N, A$ extension, so averaging gives its security bound for the combined oracle. Fix one of the security games in Theorems 15 to 18, and a polynomial size bound $q$. For distinguishing games, use balanced guessing and allow an output complement. Write $p_{A,\mathcal{O}}$ for success probability and $p_0 \in \{0, 1/2\}$ for the security threshold, and put

$$\Delta_q(\mathcal{O}) := \left( \max_{|A|\le q(\lambda)} p_{A,\mathcal{O}} - p_0 \right)_+ .$$

The next claim replaces a best oracle-dependent description by a single adversary that selects its description using the oracle.

**Claim 15.** *For every inverse-polynomial $\epsilon > 0$, there is a uniform polynomial-query adversary $A_{\mathrm{sel}}$ such that*

$$\mathbb{E}_{\mathcal{O}}\Delta_q(\mathcal{O}) \le \mathbb{E}_{\mathcal{O}}(p_{A_{\mathrm{sel}},\mathcal{O}} - p_0) + \epsilon + \mathsf{negl}(\lambda).$$

*Selection takes place before using the game's inputs, and separated algorithms receive only their selected descriptions and assigned registers.*

The proof of Claim 15 appears in Appendix K.1.

The averaged security theorem applies to $A_{\mathrm{sel}}$ because its description is uniform and its queries have polynomial count and length. Thus $\mathbb{E}_{\mathcal{O}}p_{A_{\mathrm{sel}},\mathcal{O}} \le p_0 + \mathsf{negl}(\lambda)$, and Claim 15 gives

$$\mathbb{E}_{\mathcal{O}}\Delta_q(\mathcal{O}) \le \epsilon + \mathsf{negl}(\lambda).$$

Since $\epsilon$ is any inverse polynomial, this expectation is a negligible function $\delta_q(\lambda)$. As in [KQT24], Markov's inequality gives

$$\Pr_{\mathcal{O}}[\Delta_q(\mathcal{O}) > \sqrt{\delta_q(\lambda)}] \le \sqrt{\delta_q(\lambda)}, \qquad \sum_{\lambda} \sqrt{\delta_q(\lambda)} < \infty.$$

Borel–Cantelli and a countable intersection over integer polynomial bounds $q$, including copy and request bounds, give security for all nonuniform classical descriptions, even those depending on $\mathcal{O}$.

**The promise collapse.** Finally, the estimation, search and collapse lemma (Lemma 27) applies to this entire base distribution and gives

$$\mathsf{PromiseP}^{\mathcal{O}} = \mathsf{PromiseBQP}^{\mathcal{O}} = \mathsf{PromiseQCMA}^{\mathcal{O}}.$$

For a hierarchy machine, start at the lowest level and choose the total deterministic solver supplied for its promise problem. This solver is a Boolean completion of that level's partial function, so substitute it at every query of the next verifier and apply the collapse to the resulting QCMA promise problem. Repeat up to the top level. The completion convention ensures that each resulting solver agrees with every non-$\perp$ answer of the corresponding hierarchy machine, giving $\mathsf{PromiseQCMAH}^{\mathcal{O}} = \mathsf{PromiseP}^{\mathcal{O}}$. Intersect these probability-one events.

## K.1 Selecting an adversary description

We now prove Claim 15. We first implement the chosen security game as a quantum test, then use the recursive oracle to select a description with nearly maximal test acceptance.

**The game test.** We construct a uniform $\mathsf{QPT}$ test $T^{\mathcal{O}}_{\mathrm{game}}(1^\lambda, A)$, which runs the described adversary with full access to $\mathcal{O}$ and returns the game's success bit. The test samples the honest keys and independent randomness as specified below, and every adversarial algorithm has coherent access to the same $\mathcal{O}$:

- For PRF privacy, sample $\beta \leftarrow \{0,1\}$ and $(s, \mathsf{R_{sk}}) \leftarrow \mathsf{PRF.Gen}^{\mathcal{O}}(1^\lambda)$. Give $s$ to the adversary and answer its challenge queries with $\mathsf{PRF.Eval}^{\mathcal{O}}$ when $\beta = 0$, or an independent limited-independence function when $\beta = 1$. Return $\mathbf{1}[\widehat{\beta} = \beta]$.
- For PRF piracy, generate $(s, \mathsf{R_{sk}})$, privately sample $u_1, u_2 \leftarrow \{0,1\}^\lambda$, and obtain $f_i$ by measuring the response of $\mathsf{PRF.Eval}^{\mathcal{O}}(s, \mathsf{R_{sk}}, u_i, 0)$ sequentially. Run $(\mathsf{R}_1, \mathsf{R}_2) \leftarrow \mathcal{A}_0^{\mathcal{O}}(s, \mathsf{R_{sk}})$ on the resulting key, then $\widehat{f_i} \leftarrow \mathcal{A}_i^{\mathcal{O}}(u_i, \mathsf{R}_i)$ without communication, and return $\mathbf{1}[\widehat{f_1} = f_1 \wedge \widehat{f_2} = f_2]$.
- For encryption piracy, generate $(\mathrm{pk}, \mathsf{R_{sk}})$ and run

  $$((m_{i,0}, m_{i,1})_{i=1}^2, \mathsf{R}_1, \mathsf{R}_2) \leftarrow \mathcal{A}_0^{\mathcal{O}}(\mathrm{pk}, \mathsf{R_{sk}}).$$

  Independently sample $\beta_i \leftarrow \{0,1\}$ and $\mathrm{ct}_i \leftarrow \mathsf{PKE.Enc}^{\mathcal{O}}(\mathrm{pk}, m_{i,\beta_i})$, then run $\widehat{\beta}_i \leftarrow \mathcal{A}_i^{\mathcal{O}}(\mathrm{ct}_i, \mathsf{R}_i)$ without communication and return $\mathbf{1}[\widehat{\beta}_1 = \beta_1 \wedge \widehat{\beta}_2 = \beta_2]$. For CPA, instead run $(m_0, m_1, \mathsf{R}) \leftarrow \mathcal{B}_0^{\mathcal{O}}(\mathrm{pk})$, sample $\beta \leftarrow \{0,1\}$ and $\mathrm{ct} \leftarrow \mathsf{PKE.Enc}^{\mathcal{O}}(\mathrm{pk}, m_\beta)$, then run $\widehat{\beta} \leftarrow \mathcal{B}_1^{\mathcal{O}}(\mathrm{ct}, \mathsf{R})$ and return $\mathbf{1}[\widehat{\beta} = \beta]$.
- For signatures, generate $(\mathsf{R_{vk}}, \mathsf{R_{sk}})$, give $\mathsf{R_{vk}}$ to the forger, and initialize $S = \varnothing$. On a classical request $m$, run $\mathsf{Sign}^{\mathcal{O}}(\mathsf{R_{sk}}, m)$, return its signature and set $S = S \cup \{m\}$. Answer quantum verification-key requests with $\mathsf{QSig.VKGen}$. On a forgery $(m, \sigma)$, return $\mathbf{1}[m \notin S]\mathsf{Ver}^{\mathcal{O}}(\mathsf{R_{vk}}, m, \sigma)$, generating a fresh verification key for the final $\mathsf{QSig}$ test.
- For PRS, sample $\beta \leftarrow \{0,1\}$. If $\beta = 0$, sample one key $k$ and supply $r$ fresh outputs of $\mathsf{StateGen}^{\mathcal{O}}(1^\lambda, k)$. Otherwise sample one limited-independence Boolean function $f$ and supply $r$ copies of $2^{-5\lambda/2}\sum_x (-1)^{f(x)}|x\rangle$. Return $\mathbf{1}[\widehat{\beta} = \beta]$.

The independent functions and binary-phase states can be generated efficiently.[23] Replacing the two preliminary PRF evaluations by ideal evaluations leaves the key unchanged, so $u_1, u_2$ can then

[23] For $q$ quantum queries, use a fresh $2q$-wise independent function [Zha12b]. For $r$ Haar copies on $5\lambda$ qubits, prepare the normalized state $\sum_x (-1)^{f(x)}|x\rangle$ with one $2r$-wise independent Boolean $f$. The binary-phase moment bound [BS19] gives trace-distance error $O(r^2 2^{-5\lambda})$.

be sampled after $\mathcal{A}_0$ runs. Reusable correctness bounds this change jointly with the key, $f_1, f_2$ and subsequent coherent queries. Thus each test has the original winning predicate, up to the joint correctness and moment errors, giving uniformly over the descriptions,

$$\mathbb{E}_{\mathcal{O}} \max_{|A|\le q(\lambda)} \left|\Pr\left[T^{\mathcal{O}}_{\mathrm{game}}(1^\lambda, A) = 1\right] - p_{A,\mathcal{O}}\right| = \mathsf{negl}(\lambda).$$

Increase the size bound to include the following fixed strategy: abort when $p_0 = 0$, guess a fair bit in a distinguishing game, or give the key to one honest recipient while the other guesses in a piracy game. Write $A_{\mathrm{triv}}$ for this strategy. Correctness gives $\mathbb{E}_{\mathcal{O}}(p_0 - p_{A_{\mathrm{triv}},\mathcal{O}})_+ \le \mathsf{negl}(\lambda)$.

**Selecting and running a description.** Fix an inverse-polynomial $\epsilon > 0$ and pad adversary descriptions $z$ to length $\ell = \mathsf{poly}(\lambda)$. With the test $T_{\mathrm{game}}$ interpreting invalid $z$ as $A_{\mathrm{triv}}$, let $a(z) = \langle T_{\mathrm{game}}, (1^\lambda, z)\rangle$ encode the test and its ordinary input, padding all pairs to one polynomial length $n \ge \max\{\lambda, 6/\epsilon\}$. Define the following uniform adversary $A_{\mathrm{sel}}$, using $\mathsf{Search}^{\mathcal{O}}$ from the estimation, search and collapse lemma (Lemma 27):

1. Before using the game's inputs, run $\mathsf{Search}^{\mathcal{O}}(a)$ and denote its output by $z_*$.
2. Run the adversary described by $z_*$ on those inputs, forwarding its oracle queries and external responses. In a piracy game, perform selection before invoking $\mathcal{A}_0$, then give algorithm $i$ only its selected description and register.
3. Return the selected adversary's outputs at the prescribed times.

The selection algorithm queries the fixed oracles $N, A$, uses polynomially many queries of polynomial length, and never uses the current game's hidden randomness. Except with negligible probability, its selected test has acceptance probability within $6/n \le \epsilon$ of the best test. The test approximation and the bound for $A_{\mathrm{triv}}$ above then transfer this estimate to the true success probabilities. To finish the claim, write

$$e_{\mathrm{test}}(\mathcal{O}) = \max_{|A|\le q(\lambda)} \left|\Pr\left[T^{\mathcal{O}}_{\mathrm{game}}(1^\lambda, A) = 1\right] - p_{A,\mathcal{O}}\right|.$$

Except when selection fails, its additive test error is at most $\epsilon$, so

$$\max_{|A|\le q(\lambda)} p_{A,\mathcal{O}} \le p_{z_*,\mathcal{O}} + \epsilon + 2e_{\mathrm{test}}(\mathcal{O}).$$

Moreover, $(p_0 - \max_A p_{A,\mathcal{O}})_+ \le (p_0 - p_{A_{\mathrm{triv}},\mathcal{O}})_+$. Selection precedes the game inputs, so averaging $p_{z_*,\mathcal{O}}$ gives $p_{A_{\mathrm{sel}},\mathcal{O}}$. The expected test error, the expected deficit of $A_{\mathrm{triv}}$ and the selection-failure probability are negligible. Adding these three errors proves the displayed bound for $\mathbb{E}_{\mathcal{O}}\Delta_q(\mathcal{O})$.

## K.2 Proof of average-case QCMAH hardness in QMA (Corollary 3)

We now prove Corollary 3 using the encryption scheme $\mathsf{PKE}^{\mathcal{O}}$ (Section 16) for a fixed cryptographic oracle satisfying Theorem 19.

**The sampler and its promise problem.** Let $k(\lambda)$ bound the quantum secret-key length, including any classical secret information, and put $t = k + \lambda$. All algorithms below use the fixed oracle $\mathcal{O}$. The sampler chooses

$$(\mathrm{pk}, \mathsf{R}_{\mathsf{sk}}) \leftarrow \mathsf{PKE.Gen}(1^\lambda), \qquad m \leftarrow \{0,1\}^t, \quad b \leftarrow \{0,1\},$$
$$c_i \leftarrow \mathsf{PKE.Enc}(\mathrm{pk}, bm_i) \quad (i \in [t]), \qquad X = (1^\lambda, \mathrm{pk}, (c_i)_{i\in[t]}, m).$$

It discards the secret key and returns $(X, b)$. On input $X$, the verifier initializes a classical output string $z$ and uses its $k$-qubit witness as the decryption key. For $i = 1, \ldots, t$, it runs

$z_i \leftarrow \mathsf{PKE.Dec}(\mathsf{R}_{\mathsf{sk}}, c_i)$, using the updated key for the next call. After all decryptions, it accepts exactly when $z = m$. Let $a(X)$ be its maximum acceptance probability and set $\Pi_1 = \{X : a(X) \geq 2/3\}$ and $\Pi_0 = \{X : a(X) \leq 1/3\}$. Reusable correctness (Theorem 16 and Theorem 19) gives $\Pr[X \notin \Pi_1 \mid b = 1] \leq \mathsf{negl}(\lambda)$.

For $b = 0$, condition on $\mathrm{pk}, (c_i)_i$. Decryption produces its entire output before testing equality with the independent uniform $m$. For a witness density matrix $\rho$, write $\Pr[z = m] = \mathrm{Tr}(E_m \rho)$, where $E_m \geq 0$ and $\sum_m E_m \leq I$. Markov's inequality and $\|E_m\|_\infty \leq \mathrm{Tr}\, E_m$ give the quantum-witness guessing bound [Aar12]:

$$\Pr[X \notin \Pi_0 \mid b = 0] \leq 3\, 2^{-t} \sum_m \|E_m\|_\infty \leq 3\, 2^{-t}\, \mathrm{Tr}\, I = 3\, 2^{-\lambda}.$$

**Reducing prediction to chosen-plaintext security.** To prove hardness, let $\mathsf{H}_j$ sample a fresh key and uniform $m \in \{0,1\}^t$, encrypt $m_i$ for $i \leq j$ and 0 for $i > j$ with independent encryption randomness, and return $X$. Thus $\mathsf{H}_0$ and $\mathsf{H}_t$ are the conditional distributions of the sampler's $X$ given $b = 0$ and $b = 1$. For any polynomial-time bit predictor $D$, define a CPA adversary $(\mathcal{B}_0, \mathcal{B}_1)$ by the following algorithms:

1. Algorithm $\mathcal{B}_0^{\mathcal{O}}(\mathrm{pk})$ samples $j \leftarrow [t]$ and $m \leftarrow \{0,1\}^t$, submits the message pair $(0, m_j)$, and stores $(\mathrm{pk}, j, m)$.
2. Algorithm $\mathcal{B}_1^{\mathcal{O}}(c, \mathrm{pk}, j, m)$ sets $c_j = c$, independently encrypts $m_i$ for $i < j$ and 0 for $i > j$, forms $X = (1^\lambda, \mathrm{pk}, (c_i)_i, m)$, and outputs $D^{\mathcal{O}}(X)$.

For each $j$, the two CPA challenge bits give $\mathsf{H}_{j-1}$ and $\mathsf{H}_j$, so its success minus $1/2$ is

$$\frac{1}{2t} \sum_{j=1}^{t} \big(\Pr\big[D^{\mathcal{O}}(\mathsf{H}_j) = 1\big] - \Pr\big[D^{\mathcal{O}}(\mathsf{H}_{j-1}) = 1\big]\big) = \frac{\Pr_{X,b}[D^{\mathcal{O}}(X) = b] - 1/2}{t}.$$

Chosen-plaintext security in Theorem 19 therefore bounds the predictor's advantage by a negligible function. For a uniform $\mathsf{QCMAH}_d$ machine $M_d$, the promise collapse in that theorem supplies a total deterministic polynomial-time solver $D$ agreeing with $M_d$ wherever $M_d$ outputs a bit. Since $\perp$ answers count as incorrect, replacing $M_d$ by $D$ cannot decrease its prediction success, which proves the stated hardness.