Quantum copy-protection of compute-and-compare programs in the quantum random oracle model

Andrea Coladangelo* Christian Majenz† Alexander Poremba‡

September 30, 2020

Abstract

Copy-protection allows a software distributor to encode a program in such a way that it can be evaluated on any input, yet it cannot be "pirated" – a notion that is impossible to achieve in a classical setting. Aaronson (CCC 2009) initiated the formal study of quantum copy-protection schemes, and speculated that quantum cryptography could offer a solution to the problem thanks to the quantum no-cloning theorem.

In this work, we introduce a quantum copy-protection scheme for a large class of evasive functions known as "compute-and-compare programs" – a more expressive generalization of point functions. A compute-and-compare program $\mathsf{CC}[f, y]$ is specified by a function f and a string y within its range: on input x, $\mathsf{CC}[f, y]$ outputs 1, if $f(x) = y$, and 0 otherwise. We prove that our scheme achieves non-trivial security against fully malicious adversaries in the quantum random oracle model (QROM), which makes it the first copy-protection scheme to enjoy any level of provable security in a standard cryptographic model. As a complementary result, we show that the same scheme fulfils a weaker notion of software protection, called "secure software leasing", introduced very recently by Ananth and La Placa (eprint 2020), with a standard security bound in the QROM, i.e. guaranteeing negligible adversarial advantage.

*UC Berkeley, USA. andrea.coladangelo@gmail.com

†QuSoft and Centrum Wiskunde & Informatica, Amsterdam. christian.majenz@cwi.nl

‡Computing and Mathematical Sciences, Caltech, USA. aporemba@caltech.edu

Contents

1 Introduction

1.1 Copy-protection

Copy-protection captures the following cryptographic task. A vendor wishes to encode a program in such a way that a user who receives the encoded program is able to run it on arbitrary inputs. However, the recipient should not be able to create functionally equivalent "pirated" copies of the original program. More concretely, no user should be able to process the encoded program so as to split it into two parts, each of which allows for the evaluation of the function implemented by the original program. Copy-protection of any kind is trivially impossible to achieve classically. This is because any information that the user receives can simply be copied. In the quantum realm, however, the no-cloning theorem prevents any naive copying strategy from working unconditionally, and copy-protection seems, at least in principle, possible. The key question then becomes: *Is it possible to encode the functionality of a program in a quantum state while at the same time preserving the no-cloning property?*

To be precise, we are not satisfied with preventing an adversary from copying the state that encodes the program (this is certainly a necessary condition), but we also require that there is no other way for a (computationally bounded) user to process the state into two parts (not necessarily a copy of the original) so as to allow each half to recover the input-output behaviour of the encoded program.

Quantum copy-protection was first formally considered by Aaronson [Aar09]. One of the first observations there is that families of *learnable* functions cannot be copy-protected: access to a copy-protected program, and hence its input-output behaviour, allows one to recover a classical description of the program itself, which can be copied. In [Aar09], Aaronson provides some formal definitions and constructions of copy-protection schemes. More precisely, Aaronson describes:

- A provably secure scheme to copy-protect any family of efficiently computable functions which is not quantumly learnable, assuming a *quantum* oracle implementing a certain family of unitaries.
- Two candidate schemes to copy-protect point functions in the plain model, although neither of the two has a proof of security.

In recent work [ALZ20], Aaronson, Liu and Zhang provide a scheme to copy-protect any family of efficiently computable functions which is not quantumly learnable, assuming access to a *classical* oracle, i.e. an oracle (which can be queried in superposition) that implements a classical function. We emphasize, however, that this classical function is dependent on the function that one wishes to copy-protect. In particular, none of the oracles that these schemes rely upon have candidate realizations in the plain model, and some, in particular, are impossible to realize. For example, the classical oracle used in the scheme from [ALZ20] can be used to construct an ideal obfuscator for the function f that is being copy-protected. Such an ideal obfuscator is impossible to realize unless the class of circuits which are obfuscated is learnable [BGI+12]. Some of the questions left open in [Aar09] and [ALZ20] are:

(i) Does there exist a scheme to copy-protect any non-trivial family of functions (the simplest example being point functions) which we can prove secure in the plain model under some standard assumption? What about larger classes of programs?

(ii) Can such a scheme exist that does not involve multi-qubit entanglement?

On the negative side, aside from the impossibility of copy protecting families of learnable functions, it has remained an open question to determine whether a more general impossibility result applies. In a recent result, Ananth and La Placa [ALP20] prove that a universal copy-protection scheme cannot exist, assuming the quantum hardness of the learning with errors problem [Reg05] and the existence of quantum fully homomorphic encryption.

1.2 Our contributions

In this work, we approach copy-protection from the positive side. Our main result is a copy-protection scheme for compute-and-compare programs, for which we prove non-trivial security in the quantum random oracle model. By non-trivial security we mean, informally, that a query-bounded adversary fails at pirating with at least some constant probability (which is approximately 10^{-4} for our scheme).

A desirable feature of our scheme is that the copy-protected program does not involve multi-qubit entanglement – in fact it only involves BB84 states and computational and Hadamard basis measurements. This is in contrast to previous candidate schemes for point functions in [Aar09], whose security is only conjectured, and which employ highly entangled states. The simple structure of the copy-protected program is advantageous for, e.g., error-corrected storage of the copy-protected program. We point out, however, that in a practical implementation of our scheme, where the oracle is replaced by a hash function, evaluation of the copy-protected program on an input requires *coherently* computing the hash function in an auxiliary register. This operation requires universal quantum computation.

Our scheme is not in the plain model, and hence does not fully resolve questions (i) and (ii). The (quantum) random oracle model, however, enjoys widespread acceptance and popularity in (post-quantum/quantum) cryptography, and many schemes designed for, and deployed in, practical applications enjoy provable security in that model only.

Our security definition is essentially analogous to the original definition in [Aar09] but differs more significantly from the more recent definition in [ALZ20], which is weaker. In Section 3.1, we give a more detailed comparison of our definition with the ones in [Aar09] and [ALZ20].

Our techniques and construction are inspired by recent work on *unclonable encryption* by Broadbent and Lord [BL19]. The main technical ingredient on which their construction relies are *monogamy of entanglement games*, introduced and studied extensively in [TFKW13], which they combine with an adaption of the one-way-to-hiding (O2H) lemma of [Unr15] for a security analysis in the quantum random oracle model. In a nutshell, (a special case of) the latter lemma allows one to upper bound the probability that an algorithm outputs $H(x)$, where H is a random oracle and x is any string in the domain, in terms of the probability that the algorithm "queries" at x at some point during its execution. The adaption of [BL19] extends the applicability of the O2H lemma to a setting that involves *two players*, and upper bounds the probability that the two (possibly entangled) players *simultaneously* guess $H(x)$ by the probability that they both query at x at some point during the execution of their respective strategies.

Our main technical contribution in this work is that we augment the analysis of this "simultaneous one-way-to-hiding" lemma by a search-to-decision reduction. We give an informal description of this contribution at the end of the next subsection. Along the way, we generalize the analysis of Broadbent and Lord to certain random oracles with non-uniform distributions.

1.2.1 A sketch of our copy-protection scheme

We start by describing a scheme to copy-protect point functions, which is the crux of this work. Subsequently, we describe how to extend this to compute-and-compare programs. Our scheme is inspired by Broadbent and Lord's unclonable encryption scheme [BL19], which is itself rooted in Wiesner's conjugate coding scheme [Wie83].

Let $\lambda \in \mathbb{N}$. The main idea is that it is possible to "hide" a string $v \in \{0,1\}^\lambda$ by making λ uniformly random choices of basis (either computational or Hadamard) which we denote by $\theta \in \{0,1\}^\lambda$, and then encoding each bit of v in either the computational or Hadamard basis, according to θ. Formally, this amounts to preparing the following quantum state on λ qubits:

$$|\Psi\rangle = \bigotimes_{i=1}^{\lambda} |v_i^{\theta_i}\rangle\,,$$

where $|b^s\rangle = H^s\,|b\rangle$, for $b, s \in \{0,1\}$. Given the string of basis choices θ and the state $|\Psi\rangle$, one is able to "decrypt" and recover the string v by measuring each qubit of $|\Psi\rangle$ in the basis specified by θ. We can bootstrap this idea to copy-protect point functions as follows. Let P_y be a point function with marked input $y \in \{0,1\}^\lambda$, i.e.

$$P_y(x) = \begin{cases} 1 & \text{if } x = y\,, \\ 0 & \text{if } x \neq y\,. \end{cases}$$

Our scheme rests on the following simple idea: we interpret the string y as the basis choice that "encrypts" a uniformly random string v. The copy-protected version of P_y then consists of the state $|\Psi\rangle$ together with some classical information that enables an evaluator to "recognize" v. One can take the latter information to be $H(v)$, for some hash function H (or a uniformly random function H, if one works in the random oracle model). Then, to evaluate the program on some input x, the evaluator attempts to "decrypt" using x as the basis choice, i.e. applies Hadamards $H^x = H^{x_1} \otimes \cdots \otimes H^{x_\lambda}$ to $|\Psi\rangle$, followed by a measurement in the computational basis. Let $v' \in \{0,1\}^\lambda$ be the outcome of this measurement. The evaluator checks that $H(v') = H(v)$, and outputs 1 if so, 0 otherwise.[1] Except for a minor modification which we highlight in the next paragraph, this is our scheme (described in detail in Construction 1). We will now informally discuss the correctness and the copy-protection property of this scheme.

Correctness. Informally, the scheme is correct since "decrypting" using y will result in the correct string v with certainty, whereas if one tries to "decrypt" using $x \neq y$, the outcome will most likely be a string $v' \neq v$ with $H(v') \neq H(v)$ (provided H has a large enough range). There is a slight issue with this approach, namely that an x which is, for instance, equal to y everywhere

[1] To ensure that the quantumly copy-protected program can be reused, v' is not actually measured, rather, the check $H(v') = H(v)$ is performed coherently (only the result of the check is measured). For details, see Section 4.

except for a single bit, will result in an honest evaluator outputting 1 (i.e the incorrect output) with probability $\frac{1}{2}$. To circumvent this, instead of having y itself be the choice of basis for the encoding, we have the latter be $G(y)$, where G is hash function whose range is sufficiently larger than the domain[2]. For the rest of the discussion in this section we will omit G for ease of exposition.

Copy-protection. The copy-protection property crucially leverages the following property: it is impossible for any pirate who has $|\Psi\rangle = \bigotimes_{i=1}^{\lambda} |v_i^{y_i}\rangle$ but does not know y, to produce a state on two registers AB such that two "freeloaders" Alice and Bob with access to registers A and B respectively, *as well as* access to y, can simultaneously recover v. Note that the latter property crucially holds even when both Alice and Bob are *simultaneously* receiving y. This property is essentially a consequence of the *monogamy of entanglement*. At a high level, one can consider a purification of the state received by the pirate (when averaging over the choice of v and y). Let C be the purifying register, which we can think of as being held by the vendor. Since the pirate does not have access to the register containing the choice of basis y, it is possible to argue that the only way for the state of register A to allow for recovery of v (with high probability over the basis choice) is if A is (close to) maximally entangled with C. The same argument applies to B. Hence, the monogamy of entanglement prevents Alice and Bob from recovering v simultaneously. This property is captured formally in [TFKW13], via the study of *monogamy of entanglement games* (Section 2.2). In particular, a rephrasing of the results of [TFKW13] is that, for any (unbounded) strategy of the pirate, and Alice and Bob, the probability that both Alice and Bob are able to output v is exponentially small. Unfortunately, our proof of security does not immediately follow from the above observations, mainly due to the fact that the encoded program also consists of the classical string $H(v)$, which further complicates the matter.

Before we expand on the technical hurdles we encounter when proving the copy-protection property of our scheme, let us first define security in a bit more detail. Informally, a quantum copy-protection scheme is *secure* for a family of circuits $\mathcal{C}$ (as well as a distribution D over $\mathcal{C}$) if no adversary consisting of a triple of quantum polynomial time algorithms $(\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2)$, a "pirate" $\mathcal{P}$ and two "freeloaders" $\mathcal{F}_1$ and $\mathcal{F}_2$, can win with certainty at the following game:

- The pirate $\mathcal{P}$ receives a copy-protected program ρ_C from the challenger (where the program $C \in \mathcal{C}$ is sampled from D). $\mathcal{P}$ then creates a bipartite state on registers A and B, and sends A to $\mathcal{F}_1$ and B to $\mathcal{F}_2$.
- The challenger samples a pair (x_1, x_2) of inputs to C from a suitable distribution (which is allowed to depend on C), and sends x_1 to $\mathcal{F}_1$ and x_2 to $\mathcal{F}_2$.
- $\mathcal{F}_1$ and $\mathcal{F}_2$, who are not allowed to communicate, return bits b_1 and b_2 respectively.
- $(\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2)$ win if $b_1 = C(x_1)$ and $b_2 = C(x_2)$.

The crux in proving that our scheme satisfies this security definition is in arguing that a strategy that performs well enough in the security game must be such that $\mathcal{F}_1$ and $\mathcal{F}_2$ are simultaneously querying the oracle H at v with significant probability (at some point during their executions). This would allow to construct a strategy that simultaneously "extracts" v, and thus breaks the monogamy of entanglement property.

Unruh's one-way-to-hiding (O2H) lemma [Unr15] is the standard tool to argue the above. One variant of the O2H lemma provides an upper bound on the probability that an adversary distinguishes $H(v)$ from a uniformly random string in the co-domain of H, in terms of the probability that such an adversary queries the oracle at v. However, in our security proof, this analysis needs to be augmented: we need to account for possibly *entangled* strategies that attempt to distinguish $H(v)$ from a uniformly random string.

- The main technical contribution of [BL19] is an important step in this direction. There, the authors bound the probability of entangled parties simultaneously guessing $H(v)$ in terms of the probability that the two parties simultaneously query the oracle at v.

[2]Such a hash function can, e.g., be obtained using the sponge construction as in SHA3, but by extending the so-called squeezing phase.

- The above is not entirely sufficient for our purpose: in our security game, the freeloaders are not asked to guess $H(v)$, rather they are only required to return a single bit. Note that an adversary who wins our security game (with high probability) *must* be able to distinguish $H(v)$ from a uniformly random string. This is because an adversary who receives a uniformly random string instead of $H(v)$ cannot do better than random guessing in the security game. Thus, we need to provide a *search-to-decision* reduction: an adversary such that the freeloaders are both able to distinguish $H(v)$ from a uniformly random string (sufficiently well) can be used to construct an adversary such that the freeloaders simultaneously extract v. This is our main technical contribution (captured by Lemmas 18, 19 and 20).

Naively, the difficulties encountered when attempting simultaneous extraction might seem somewhat surprising given the fact that the problem is solved by a straightforward union bound in the classical setting. Indeed, if two algorithms both distinguish $H(v)$ from a uniformly random string with probability at least $3/4 + \epsilon$, for some $\epsilon > 0$, the probability that the respective query transcripts contain v is at least $1/2 + 2\varepsilon$ for each of them. This guarantees that both transcripts contain v simultaneously with probability at least 4ε. For quantum queries, however, there exists no transcript, and measuring a query for extraction disturbs the run of the algorithms. But the extraction probability is too small for a union bound even in the classical case, as extraction requires choosing a query at random, which suppresses the success probability by a factor of $O(q^\alpha)$, where q is the total number of oracle queries, and $\alpha = -1$ in the classical and $\alpha = -2$ in the quantum case.

The dependence of the simple classical technique on query transcripts, and the lack of such in the quantum case, in some sense capture the essence of a major difficulty we encounter. Let us therefore elaborate a bit further by noting that the classical technique mentioned above crucially depends on *separate* query transcripts for the two players. This implies that it depends on information that can be elusive in the quantum case, in a very strong sense. Suppose e.g. the two players share an entangled state $(|0\rangle|1\rangle + |1\rangle|0\rangle)/\sqrt{2}$. They proceed by making a single query each, on input x_i controlled on their part of the entangled state being in state i, for two arbitrary inputs x_0, x_1. The final state is then

$$|\psi\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle |x_0\rangle |H(x_0)\rangle |1\rangle |x_1\rangle |H(x_1)\rangle + |1\rangle |x_1\rangle |H(x_1)\rangle |0\rangle |x_0\rangle |H(x_0)\rangle \right),$$

where the first, and the last, three registers are held by the first, and the second, player, respectively. At this point, the inputs x_0 and x_1 have both been queried with certainty, but the information of who has queried which of the two is, in fact, *distributed quantum* information in the hand of the players, precluding any third-party knowledge about it due to the no-cloning theorem. The difference between global and individual query transcripts also becomes evident in the recently developed superposition oracle framework [Zha19]. In the described example, the fact that both x_0 and x_1 have been queried by somebody can be recovered using the superposition oracle framework. More generally, the superposition oracle framework can be used as a replacement for query transcripts, sometimes in a quite straight-forward manner (see e.g. [Zha19, BHH+19, AMRS20, CMSZ19, HI19]). However, the described example illustrates that recording *which input* was queried *at which interface* is incompatible with the correctness of any quantum-accessible random oracle simulation.

1.2.2 Extension to compute-and-compare programs.

The copy-protection scheme for point functions we described in the previous section can be straightforwardly extended to the more general class of compute-and-compare programs [WZ17, GKW17]. By definition, a compute-and-compare program $\mathsf{CC}[f, y]$ is specified by an efficiently computable function $f : \{0,1\}^n \to \{0,1\}^m$ and a string $y \in \{0,1\}^m$ in its range, where

$$\mathsf{CC}[f, y](x) = \begin{cases} 1 & \text{if } f(x) = y\,, \\ 0 & \text{if } f(x) \neq y\,. \end{cases}$$

Point functions are a special case of compute-and-compare programs where the function f is the identity map.

In Section 5, we show how to copy-protect $\mathsf{CC}[f, y]$ in the following simple way: the copy-protected program consists of (a description of a circuit computing) f in the clear, together with a copy-protected version of the point function with marked input y. The intuition is that it is enough to copy-protect the marked input y in order to render $\mathsf{CC}[f, y]$ unclonable. At first, it might seem surprising that one can give f in the clear while preserving unclonability, as the encoded program now leaks significantly more information than its input/output behavior alone.

However, at a second thought, it is in fact quite natural that one can render a functionality "unclonable" by just making some sufficiently important component of it unclonable. In fact, it is straightforward to show that copy-protection security of the extended construction reduces to security of the original point function scheme.

1.2.3 Secure software leasing

On top of proving the impossibility of a general copy-protection scheme for all unlearnable functions, Ananth and La Placa introduce in [ALP20] a weaker notion of copy-protection, which they call "secure software leasing" (SSL). The sense in which the latter is weaker than copy-protection is that one assumes that the freeloaders $\mathcal{F}_1$ and $\mathcal{F}_2$ (now a single adversary) are limited to performing the honest evaluation procedure only. Rather than emphasizing the impossibility of simultaneous evaluation on inputs chosen by a challenger, SSL captures the essence of quantum copy-protection in the following scenario. An authority (the lessor) wishes to lease a copy ρ_C of a classical circuit $C \in \mathcal{C}$ to a user (the lessee) who is supposed to return back ρ_C at a later point in time, as specified by the lease agreement. Once the supposed copy is returned and verified by the lessor, the security property requires that the adversary can no longer compute C. More formally, no adversary should be able to produce a (possibly entangled) quantum state such that:

- One half of the state is deemed valid by the lessor, once it is returned.
- The other half can be used to honestly evaluate C on every input of the adversary's choosing.

Surprisingly, Ananth and La Placa were able to show in [ALP20] that a general SSL scheme is also impossible, despite having weaker security requirements compared to copy-protection. On the positive side, the authors describe an SSL scheme for general evasive circuits assuming the existence of subspace-hiding obfuscators [Zha17] and the quantum hardness of the learning with errors problem [Reg05]. Because subspace-hiding obfuscators are currently only known to exist under indistinguishability obfuscation [Zha17, SW13], the same applies to the security of the scheme proposed in [ALP20].

Our work: SSL revisited. As we mentioned earlier, the original definition of secure software leasing in [ALP20] is a weaker version of copy-protection in the following two ways:

- The lessor performs a prescribed verification procedure on a register returned by the lessee.
- The lessee is required to perform the honest evaluation procedure with respect to any post-verification registers in the lessee's possession.

We revisit the notion of secure software leasing from a similar perspective as in our copy-protection definition. Our main contributions are the following. First, we introduce a new and intuitive SSL definition (Section 6) by means of a cryptographic security game which does not limit the adversary to performing the honest evaluation on any post-verification registers.[3] Our definition remains faithful to the idea of SSL, while at the same time offering a stronger security guarantee. Second, we show that our definition of security is achievable with a standard negligible security bound in the quantum random oracle model for the class of compute-and-compare programs (Section 6). Our SSL scheme (Construction 4) is virtually equivalent to our copy-protection scheme, but is adapted to the syntax of SSL. Informally, any SSL scheme $(\mathsf{SSL.Gen}, \mathsf{SSL.Lease}, \mathsf{SSL.Eval}, \mathsf{SSL.Verify})$ according to our definition should satisfy the following property. After receiving a leased copy of a classical circuit C, denoted by ρ_C (and generated using SSL.Lease), and a circuit for SSL.Eval, no adversary should be able to produce a (possibly entangled) quantum state σ on two registers R_1 and R_2 such that:

[3] The SSL definition in [ALP20] is not "operational" and cannot be directly phrased as a security game.

- $\mathsf{SSL.Verify}$ deems the contents of register R_1 of $\sigma_{\mathsf{R}_1\mathsf{R}_2}$ to be valid, and
- the adversary can predict the output of circuit C (on challenge inputs chosen by the lessor) using an arbitrary measurement of the post-verification state in register R_2.

In Section 6 we show the following key property about our SSL scheme in Construction 4: once a leased copy is successfully returned to the lessor, no adversary can distinguish the marked input of a compute-and-compare program from a random (non-marked) input with probability better than 1/2, except for a negligible advantage (in the security parameter). Our scheme can thus be thought of as having perfect security for a natural choice of "input challenge distribution". The result follows from a standard application of the O2H lemma and a custom "uncertainty relation" variant of the monogamy of entanglement property which appeared in a work of Unruh [Unr15]. The latter appears in similar contexts in the quantum key-distribution literature. Note that the technical complications arising in the proof of security of our original copy-protection scheme do not appear in the SSL security proof. Crucially, this is because we can leverage the fact that the lessor is performing a prescribed verification procedure.

1.3 Related work

Obfuscation. Copy-protection is related to program obfuscation [BGI+12, AF16], although the extent to which they relate to each other is still unclear. It seems plausible that some degree of "obfuscation" is necessary in order to prevent a pirate from copying a program. A natural question is whether there exist schemes that satisfy both notions simultaneously.

We give an affirmative answer to this question by showing that our quantum copy-protection scheme for point functions (Construction 1) also satisfies the notion of quantum virtual-black box (VBB) obfuscation [AF16]. This results in the first provably secure scheme which is simultaneously a quantum copy-protection scheme as well as a quantum obfuscation scheme, in the quantum random oracle model. The main idea is the following: any computationally bounded adversary cannot distinguish between $\left((\bigotimes_i |v_i^{\theta_i}\rangle), H(v)\right)$ and $\left((\bigotimes_i |v_i^{\theta_i}\rangle), z\right)$, where z is a uniformly random string of the same size as $H(v)$, unless the adversary queries the oracle at v, which can only happen with negligible probability. Notice that, when averaging over H, v, θ and z, the state $\left((\bigotimes_i |v_i^{\theta_i}\rangle), z\right)$ is maximally mixed. Thus, the copy-protected program is computationally indistinguishable from a maximally mixed state. We give a detailed proof in Section 4.2.

Conversely, there exist, of course, program obfuscation schemes which are not copy-protection schemes (since obfuscators for certain classes of programs are possible classically, while copy-protection schemes are impossible). Moreover, Aaronson [Aar09] previously observed that any family of *learnable* programs cannot be copy-protected: access to a copy-protected program (and hence its input-output behaviour) allows the user to recover the classical description of the program itself, which can be copied. But for precisely the same reason, *learnable* programs can be VBB obfuscated by definition.

Finally, we also provide a missing piece that allows us to separate the notions of quantum copy-protection and quantum VBB obfuscation (Figure 1). Namely, we show that there exists a provably secure scheme (Construction 2) to copy-protect compute-and-compare programs (in the quantum random oracle model) that does not, in general, satisfy the notion of quantum VBB obfuscation (this follows easily in the case when f comes from an ensemble that is not VBB obfuscatable). Our construction suggests that it is sometimes safe to publish parts of the program that is being copy-protected in the clear, as long as some crucial components are still "obfuscated".

In conclusion, we find that obfuscation and copy-protection are, in general, incomparable functionalities. We point out, however, that obfuscators have so far been employed in all proposed constructions of quantum copy-protection schemes, such as in [Aar09, ALZ20], as well as in our scheme of Construction 1 (note that publishing $H(v)$ for a random oracle H is an ideal obfuscator for the point function with marked input v).

Unclonable encryption. This cryptographic functionality was formalized recently by Broadbent and Lord [BL19], and informally introduced earlier by Gottesman [Got03]. In an unclonable encryption scheme, one encrypts a *classical* message in a *quantum* ciphertext, in such a way that the latter cannot be processed and *split* into two parts, each of which, together with a classical

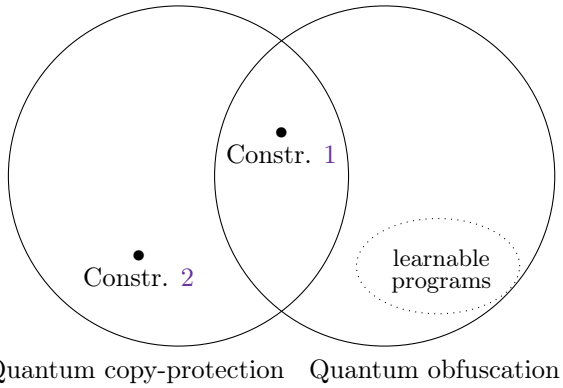

Figure 1: **Separation between quantum copy-protection and quantum VBB obfuscation**. Construction 1 features a quantum copy-protection scheme for point functions which satisfies both notions, while our second scheme for compute-and-compare programs in Construction 2 does not satisfy the notion of quantum VBB obfuscation.

secret key, enables decryption. The setting of unclonable encryption is very similar to that of copy-protection, the main difference being that there is no "functionality" associated to the quantum ciphertext, other than it being used for recovering the encrypted message. As we mentioned earlier, our copy-protection scheme is inspired by the unclonable encryption scheme in [BL19], and our analysis extends some of the techniques developed there.

Revocable quantum timed-release encryption. Timed-release encryption (also known as time-lock puzzles) is an encryption scheme that allows a recipient to decrypt only after a specified amount time, say T, has passed. Unruh [Unr15] gave the first quantum timed-release encryption scheme that is "revocable" in the sense that a user can return the timed-release encryption before time T, thereby losing all access to the data. It is easy to see that this notion is impossible to achieve classically for precisely the same reason copy-protection is impossible: any adversary can simply generate copies of the classical ciphertext or source code, respectively. From a technical point of view, revocable quantum timed-release encryption shares many similarities with the notion of "secure software leasing" in [ALP20]. Besides the fact that the former encodes a *plaintext* and the latter encodes a *program*, the security property essentially remains the same: once a quantum state is returned and successfully verified, the user is supposed to lose all relevant information. Our proof of security for the SSL scheme in Construction 3 is inspired by Unruh's proof for revocable one-way timed-release quantum encryption in [Unr15].

1.4 Open questions

Our work is the first to construct a copy-protection scheme in a standard cryptographic model (the QROM) with non-trivial security against malicious adversaries. It leaves several questions open. The most pressing ones are the following.

- First of all, our security guarantee is very weak: it only ensures that no adversary can win with probability more than $1 - \delta$, for a very small constant δ (approximately 10^{-4}). Is it possible to boost the security to ensure negligible advantage? (The main technical hurdle seems to be the factor of 9 in the bound of Lemma 19.)
- Another open problem concerns upgrading security in a different way, by providing the pirate with k copy-protected copies of a program and asking them to satisfy $k + 1$ freeloaders. We believe that our scheme can achieve such a notion, but with a security that becomes worse as k grows. Providing a scheme where security does not depend on k is an interesting open question.

- Finally, is it possible to remove the requirement of a random oracle, and to achieve a scheme with non-trivial security against malicious adversary in the plain model? We think that this would require fundamentally different techniques.

1.5 Acknowledgements

The authors thank Anne Broadbent, Yfke Dulek, Jiahui Liu, Fermi Ma, Christian Schaffner, Umesh Vazirani and Thomas Vidick for helpful discussions. In particular, the authors thank Thomas Vidick for valuable feedback on an earlier draft of this manuscript, and Umesh Vazirani for suggesting the extension of our scheme to compute-and-compare programs. The authors are grateful for the support of the Simons Institute for the Theory of Computing, where part of this research was carried out. A.C was funded by the Simons Institute for the Theory of Computing, and is currently funded by DOD. C.M. was funded by a NWO VENI grant (Project No. VI.Veni.192.159). A.P. is partially supported by AFOSR YIP award number FA9550- 16-1-0495, the Institute for Quantum Information and Matter (an NSF Physics Frontiers Center; NSF Grant PHY-1733907) and the Kortschak Scholars program.

2 Preliminaries

2.1 Notation and background

Basic notation. For $n, m \in \mathbb{N}$, we denote the set of all functions $f : \{0,1\}^n \to \{0,1\}^m$, as $\mathsf{Bool}(n,m)$. The notation $x \leftarrow \{0,1\}^n$ denotes sampling an element x uniformly at random in $\{0,1\}^n$, whereas $x \leftarrow D$ denotes sampling of an element x from a distribution D. We denote the expectation value of a random variable X by $\mathbb{E}[X] = \sum_x x \Pr[X = x]$. We call a non-negative real-valued function $\mu : \mathbb{N} \to \mathbb{R}^+$ negligible if $\mu(n) = o(1/p(n))$, for every polynomial $p(n)$. The min-entropy of a random variable X is defined as $\mathbf{H}_{\min}(X) = -\log(\max_x \Pr[X = x])$. The conditional min-entropy of a random variable X conditioned on a correlated random variable Y is defined as $\mathbf{H}_{\min}(X|Y) = -\log\left(\mathbb{E}_{y\leftarrow Y}\left[\max_x \Pr[X = x|Y = y]\right]\right)$.

Quantum computation. A comprehensive introduction to quantum computation and quantum information can be found in [NC11] and [Wil13]. We denote a finite-dimensional complex Hilbert space by $\mathcal{H}$, and we use subscripts to distinguish registers. For example $\mathcal{H}_{\mathsf{R}}$ is the Hilbert space of register R. The Euclidean norm over a finite-dimensional complex Hilbert space $\mathcal{H}$ is denoted as $\|\cdot\|$. A quantum register over the Hilbert space $\mathcal{H} = \mathbb{C}^2$ is called a *qubit*. For $n \in \mathbb{N}$, we refer to quantum registers over the Hilbert space $\mathcal{H} = \left(\mathbb{C}^2\right)^{\otimes n}$ as n-qubit states. We use the word "quantum state" to refer to both pure states (unit vectors $|\psi\rangle \in \mathcal{H}$) and density matrices (positive semidefinite matrices ρ of unit trace in the space of density matrices $\mathcal{D}(\mathcal{H})$). When n is clear from the context, we denote by $|\phi^+\rangle_{\mathsf{AB}}$ the maximally entangled n-qubit Einstein-Podolsky-Rosen (EPR) [EPR35] state on the Hilbert space $\mathcal{H}_{\mathsf{A}} \otimes \mathcal{H}_{\mathsf{B}}$:

$$|\phi^+\rangle_{\mathsf{AB}} = \frac{1}{\sqrt{2^n}} \sum_{v\in\{0,1\}^n} |v\rangle_{\mathsf{A}} \otimes |v\rangle_{\mathsf{B}}\,.$$

A POVM is a finite set $\{M_i\}$ of positive semidefinite matrices with the property that $\sum_i M_i = \mathbb{1}$, where $\mathbb{1}$ is the identity matrix. The trace distance between two states ρ and σ is defined by $\mathsf{TD}(\rho,\sigma) = \frac{1}{2}\|\rho - \sigma\|_1$. A quantum channel is a linear map $\Phi : \mathcal{H}_{\mathsf{A}} \to \mathcal{H}_{\mathsf{B}}$ between two Hilbert spaces $\mathcal{H}_{\mathsf{A}}$ and $\mathcal{H}_{\mathsf{B}}$. We say that a channel Φ is completely positive if, for any ancilla register of dimension n, the induced map $\mathbb{1}_n \otimes \Phi$ is positive, and we call it trace-preserving if it holds that $\mathrm{Tr}[\Phi(\rho)] = \mathrm{Tr}[\rho]$, for all $\rho \in \mathcal{D}(\mathcal{H}_{\mathsf{A}})$. A quantum channel that is both completely positive and trace-preserving is called a quantum CPTP channel. Let $\rho \in \mathcal{D}(\mathcal{H}_{\mathsf{A}} \otimes \mathcal{H}_{\mathsf{B}})$ be a bipartite state. Then, the min-entropy of A conditioned on system B is given by:

$$\mathbf{H}_{\min}(\mathsf{A}|\mathsf{B})_\rho = -\inf_{\sigma\in\mathcal{D}(\mathcal{H}_{\mathsf{B}})} D_{\max}\left(\rho_{\mathsf{AB}} \,||\, \mathbb{1}_{\mathsf{A}} \otimes \sigma_{\mathsf{B}}\right),$$

where $D_{\max}(\rho \,||\, \sigma) = \inf_\lambda\{\lambda : \rho \leq 2^\lambda \sigma\}$ is the max-relative entropy. A *classical-quantum* state $\rho \in \mathcal{D}(\mathcal{H}_{\mathsf{X}} \otimes \mathcal{H}_{\mathsf{B}})$ is a quantum state on registers X and B that depends on a classical variable X in system X. If the variable X is distributed according to P_X, the state ρ can be expressed as

$$\rho_{\mathsf{XB}} = \sum_x P_X(x) \left|x\right\rangle \left\langle x\right|_{\mathsf{X}} \otimes \rho^x_{\mathsf{B}}.$$

Let $p_g(\mathsf{X}|\mathsf{B}) = \sum_x P_X(x) \mathrm{Tr}[\Lambda_x \rho^x_{\mathsf{B}}]$ be the optimal guessing probability for the variable X when using an optimal measurement strategy $\{\Lambda_x\}$ that maximizes the expression. The quantity can then be expressed in terms of the min-entropy via $p_g(\mathsf{X}|\mathsf{B}) = 2^{-\mathbf{H}_{\min}(\mathsf{X}|\mathsf{B})}$ [KRS09]. By a polynomial-time *quantum algorithm* (or QPT algorithm) we mean a polynomial-time uniform family of quantum circuits, where each circuit in the circuit family is described by a sequence of unitary gates and measurements. A quantum algorithm may, in general, receive (mixed) quantum states as inputs and produce (mixed) quantum states as outputs. We oftentimes restrict QPT algorithms implicitly; for example, if we write $\Pr[\mathcal{A}(1^\lambda) = 1]$ for a QPT algorithm $\mathcal{A}$, it is implicit that $\mathcal{A}$ is a QPT algorithm that output a single classical bit.

Definition 1 (Indistinguishability of ensembles of quantum states, [Wat06])**.** *Let $p : \mathbb{N} \to \mathbb{N}$ be a polynomially bounded function, and let ρ_λ and σ_λ be $p(\lambda)$-qubit quantum states. We say that $\{\rho_\lambda\}_{\lambda\in\mathbb{N}}$ and $\{\sigma_\lambda\}_{\lambda\in\mathbb{N}}$ are quantum computationally indistinguishable ensembles of quantum states, denoted by $\rho_\lambda \approx_c \sigma_\lambda$, if, for any QPT distinguisher $\mathcal{D}$ with single-bit output, any polynomially bounded $q : \mathbb{N} \to \mathbb{N}$, any family of $q(\lambda)$-qubit auxiliary states $\{\nu_\lambda\}_{\lambda\in\mathbb{N}}$, and every $\lambda \in \mathbb{N}$,*

$$\left|\Pr[\mathcal{D}(\rho_\lambda \otimes \nu_\lambda) = 1] - \Pr[\mathcal{D}(\sigma_\lambda \otimes \nu_\lambda) = 1]\right| \leq \mathsf{negl}(\lambda)\,.$$

We also make use of the following standard results:

Lemma 1 (Ricochet property)**.** *Let $n \in \mathbb{N}$ and $M \in \mathbb{C}^{2^n \times 2^n}$ be any matrix, and let $\left|\phi^+\right\rangle_{\mathsf{AB}}$ an EPR state on registers A and B on n qubits. Then,*

$$(M_{\mathsf{A}} \otimes \mathbb{1}_{\mathsf{B}}) \left|\phi^+\right\rangle_{\mathsf{AB}} = (\mathbb{1}_{\mathsf{A}} \otimes M^T_{\mathsf{B}}) \left|\phi^+\right\rangle_{\mathsf{AB}}\,.$$

Lemma 2 (Gentle Measurement Lemma, [Win99, Aar05])**.** *Let ρ be a density matrix, let Λ, where $0 \leq \Lambda \leq \mathbb{1}$, be an element of a POVM such that $\mathrm{Tr}[\Lambda\rho] \geq 1 - \epsilon$, for some $\epsilon > 0$ and let $\tilde{\rho} = \Lambda\rho\Lambda$. Then:*

$$\mathsf{TD}(\rho, \tilde{\rho}) \leq \sqrt{\epsilon}.$$

And a slight variation of Lemma 2.

Lemma 3 (Closeness to ideal states, [Unr15])**.** *Let ρ be a density matrix, let Π be a projector, and let $\epsilon = \mathrm{Tr}[(\mathbb{1} - \Pi)\,\rho]$. Then, there exists an ideal state ρ^{id} such that:*

- $\mathsf{TD}(\rho, \rho^{\mathsf{id}}) \leq \sqrt{\epsilon}$
- *ρ^{id} is a mixture in the image of Π, i.e. $\rho^{\mathsf{id}} = \sum_i p_i \left|\psi_i\right\rangle \left\langle\psi_i\right|$ is a normalized state with $\left|\psi_i\right\rangle \in \mathsf{im}(\Pi)$, $\sum_i p_i = 1$ and $p_i \geq 0$, for all i.*

2.2 Monogamy of entanglement games

For a detailed introduction to monogamy-of-entanglement games, we refer the reader to [TFKW13], where they were introduced and studied extensively. In this section, we limit ourselves to introducing a version of a monogamy-of-entanglement game that suffices for our purpose. Let $\lambda \in \mathbb{N}$. The game is between a challenger and an adversary, specified by a triple of interactive quantum machines $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ (for a formal definition of interactive quantum machine we refer the reader to [Unr12]). For brevity, we use the notation $\left|x^\theta\right\rangle = H^\theta \left|x\right\rangle$, where $H^\theta = H^{\theta_1} \otimes \ldots \otimes H^{\theta_\lambda}$ and $\theta, x \in \{0,1\}^\lambda$. The game takes place as follows:

- The challenger samples $x, \theta \leftarrow \{0,1\}^\lambda$ and sends the state $\left|x^\theta\right\rangle$ to $\mathcal{A}$.
- $\mathcal{A}$ sends a quantum register to $\mathcal{B}$ and one to $\mathcal{C}$.
- The challenger sends θ to both $\mathcal{B}$ and $\mathcal{C}$.
- $\mathcal{B}$ and $\mathcal{C}$ return strings x' and x'' to the challenger.

$\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ are not allowed to communicate other than where specified by the game. $\mathcal{A}, \mathcal{B}, \mathcal{C}$ win if $x = x' = x''$.

The following lemma, from [TFKW13], upper bounds the winning probability of an adversary in the game. As stated in the form below, this lemma appears in [BL19].

Lemma 4. *Let $\lambda \in \mathbb{N}$. For any Hilbert spaces $\mathcal{H}_{\mathsf{B}}$ and $\mathcal{H}_{\mathsf{C}}$, any families of* POVMs *on these Hilbert spaces respectively,*

$$\left\{\left\{B_\theta^x\right\}_{x\in\{0,1\}^\lambda}\right\}_{\theta\in\{0,1\}^\lambda} \quad \text{and} \quad \left\{\left\{C_\theta^x\right\}_{x\in\{0,1\}^\lambda}\right\}_{\theta\in\{0,1\}^\lambda},$$

and any CPTP *map $\Phi : \mathcal{D}\left((\mathbb{C}^2)^{\otimes\lambda}\right) \to \mathcal{D}(\mathcal{H}_{\mathsf{B}} \otimes \mathcal{H}_{\mathsf{C}})$, we have:*

$$\mathbb{E}_{\theta\in\{0,1\}^\lambda}\mathbb{E}_{x\in\{0,1\}^\lambda}\mathrm{Tr}\left[B_\theta^x \otimes C_\theta^x\, \Phi\left(|x^\theta\rangle\langle x^\theta|\right)\right] \leq \left(\frac{1}{2} + \frac{1}{2\sqrt{2}}\right)^\lambda \tag{1}$$

It is natural to wonder whether a similar guarantee holds also when x is sampled from a distribution with high min-entropy, which is not necessarily uniform. This follows as a corollary of Lemma 4.

Corollary 1. *Let $\lambda \in \mathbb{N}$. Let X be a random variable over $\{0,1\}^\lambda$ with min-entropy $\mathbf{H}_{\min}(X) \geq h$. For any Hilbert spaces $\mathcal{H}_{\mathsf{B}}$ and $\mathcal{H}_{\mathsf{C}}$, any family of* POVMs *on these Hilbert spaces respectively,*

$$\left\{\left\{B_\theta^x\right\}_{x\in\{0,1\}^\lambda}\right\}_{\theta\in\{0,1\}^\lambda} \quad \text{and} \quad \left\{\left\{C_\theta^x\right\}_{x\in\{0,1\}^\lambda}\right\}_{\theta\in\{0,1\}^\lambda},$$

and any CPTP *map $\Phi : \mathcal{D}\left((\mathbb{C}^2)^{\otimes\lambda}\right) \to \mathcal{D}(\mathcal{H}_{\mathsf{B}} \otimes \mathcal{H}_{\mathsf{C}})$, we have:*

$$\sum_{x\in\{0,1\}^\lambda} \Pr[X = x]\, \mathbb{E}_\theta \mathrm{Tr}\left[B_\theta^x \otimes C_\theta^x\, \Phi\left(|x^\theta\rangle\langle x^\theta|\right)\right] \leq 2^{-h}\left(1 + \frac{1}{\sqrt{2}}\right)^\lambda. \tag{2}$$

Proof.

$$\begin{aligned}\sum_{x\in\{0,1\}^\lambda} \Pr[X = x]\, \mathbb{E}_\theta \mathrm{Tr}\left[B_\theta^x \otimes C_\theta^x\, \Phi\left(|x^\theta\rangle\langle x^\theta|\right)\right] &\leq 2^{-h}2^\lambda \mathbb{E}_x \mathbb{E}_\theta \mathrm{Tr}\left[B_\theta^x \otimes C_\theta^x\, \Phi\left(|x^\theta\rangle\langle x^\theta|\right)\right] \\ &\leq 2^{-h+\lambda}\left(\frac{1}{2} + \frac{1}{2\sqrt{2}}\right)^\lambda = 2^{-h}\left(1 + \frac{1}{\sqrt{2}}\right)^\lambda.\end{aligned} \tag{3}$$

Note that the second inequality follows from Lemma 4. □

In particular, notice that when $h > \frac{4}{5}\lambda$, the RHS of (3) is less than 0.981^λ, and thus negligible.

2.3 The quantum random oracle model

Oracles with quantum access have been studied extensively, for example in [BBBV97, BDF+11]. We say that a quantum algorithm $\mathcal{A}$ has oracle access to a classical function $H : \{0,1\}^\lambda \to \{0,1\}^m$, denoted by $\mathcal{A}^H$, if $\mathcal{A}$ is allowed to use a unitary gate O^H at unit cost in time. The unitary O^H acts as follows on the computational basis states of a Hilbert space $\mathcal{H}_{\mathsf{A}} \otimes \mathcal{H}_{\mathsf{B}}$ of $\lambda + m$ qubits:

$$O^H : \quad |x\rangle_{\mathsf{A}} \otimes |y\rangle_{\mathsf{B}} \longrightarrow |x\rangle_{\mathsf{A}} \otimes |y \oplus H(x)\rangle_{\mathsf{B}},$$

where the operation $\oplus$ denotes bit-wise addition modulo 2. In general, we can model the interaction of a quantum algorithm that makes q queries to an oracle H as $(UO^H)^q$, i.e. alternating unitary computations and queries to the oracle H, where U is some operator acting on $\mathcal{H}_{\mathsf{A}} \otimes \mathcal{H}_{\mathsf{B}} \otimes \mathcal{H}_{\mathsf{C}}$, where $\mathcal{H}_{\mathsf{C}}$ is some auxiliary Hilbert space [BBBV97, BDF+11, Unr15][4]. We call a (possibly super-polynomial-time) quantum algorithm $\mathcal{A}$ with access to an oracle O *query-bounded* if $\mathcal{A}$ makes at most polynomially many (in the size of its input) queries to O. The *random oracle* model refers to

[4] We can chose the algorithm's unitaries between oracle calls to be all the same by introducing a "clock register" that keeps track of the number of oracle calls made so far.

a setting in which the function $H : \{0,1\}^\lambda \to \{0,1\}^m$ is sampled uniformly at random. Random oracles play an important role in cryptography as models for cryptographic hash functions in the so-called random oracle model (ROM) [BR93]. For post-quantum and quantum cryptography, random oracles modelling hash functions need to be *quantum* accessible (i.e. accessible as a unitary gate, and thus in superposition), resulting in what is known as the quantum random oracle model (QROM) [BDF+11]. Despite being uninstantiable in principle [CGH04, ES20], modeling hash functions in the (Q)ROM is considered a standard assumption in cryptography.

2.3.1 Some technical lemmas

Recall that we denote by $\mathrm{Bool}(\lambda, m)$ the set of functions from $\{0,1\}^\lambda$ to $\{0,1\}^m$.

Lemma 5. *Let $f : \mathrm{Bool}(\lambda, m) \to \mathbb{R}$, and $x \in \{0,1\}^\lambda$. For $H \in \mathrm{Bool}(\lambda, m)$ and $y \in \{0,1\}^m$, let $H_{x,y} \in \mathrm{Bool}(\lambda, m)$ be such that*

$$H_{x,y}(s) = \begin{cases} H(s) & \text{if } s \neq x\,, \\ y & \text{if } s = x\,. \end{cases}$$

Then,

$$\mathbb{E}_H f(H) = \mathbb{E}_H \mathbb{E}_y f(H_{x,y})\,.$$

Proof. The proof is straightforward, and can be found in Lemma 19 of [BL19]. □

The following is a technical lemma about a quantum adversary not being able to distinguish between samples from $H(U_\lambda)$ and from U_m, even when given oracle access to H, where the function $H : \{0,1\}^\lambda \to \{0,1\}^m$ is sampled uniformly at random.

Consider the following game between a challenger and a quantum adversary $\mathcal{A}$, specified by $\lambda, m \in \mathbb{N}$, and a distribution X over $\{0,1\}^\lambda$,

- The challenger samples a uniformly random function $H : \{0,1\}^\lambda \to \{0,1\}^m$ and $b \leftarrow \{0,1\}$.
- If $b = 0$: the challenger samples $x \leftarrow X$, sends $H(x)$ to $\mathcal{A}$.
  If $b = 1$: the challenger samples uniformly $z \leftarrow \{0,1\}^m$, sends z to $\mathcal{A}$.
- $\mathcal{A}$ additionally gets oracle access to H. $\mathcal{A}$ returns a bit b' to the challenger.

$\mathcal{A}$ wins if $b = b'$. Let $\mathsf{Distinguish}(\mathcal{A}, \lambda, m, X)$ be a random variable for the outcome of the game.

Lemma 6. *For any adversary $\mathcal{A}$ making q oracle queries, any family of distributions $\{X_\lambda : \lambda \in \mathbb{N}\}$ where for all λ, X_λ is a distribution over $\{0,1\}^\lambda$, for any polynomially bounded function $m : \mathbb{N} \to \mathbb{N}$, there exists a negligible function μ such that, for any $\lambda \in \mathbb{N}$, the following holds:*

$$\Pr[\mathsf{Distinguish}(\mathcal{A}, \lambda, m(\lambda), X_\lambda) = 1] \leq \frac{1}{2} + (3q+2)qM + \mu(\lambda)\,,$$

where M is a quantity that is negligible in λ if $2^{-\mathbf{H}_{\min}(X_\lambda)/2}$ is negligible in λ.

Corollary 2. *For any query-bounded adversary $\mathcal{A}$, any $\epsilon > 0$, any family of distributions $\{X_\lambda : \lambda \in \mathbb{N}\}$, where X_λ is a distribution over $\{0,1\}^\lambda$ with $\mathbf{H}_{\min}(X_\lambda) > \lambda^\epsilon$ for every λ, for any polynomially bounded function $m : \mathbb{N} \to \mathbb{N}$, there exists a negligible function μ such that, for any $\lambda \in \mathbb{N}$, the following holds:*

$$\Pr[\mathsf{Distinguish}(\mathcal{A}, \lambda, m(\lambda), X_\lambda) = 1] \leq \frac{1}{2} + \mu(\lambda)\,.$$

The key step in the proof of Lemma 6 is captured by the one-way-to-hiding lemma [Unr15, AHU19][5]. We restate it here following our notation (and provide a proof in the Appendix A.1 for completeness). Informally, the lemma gives an upper bound on an adversary's advantage (when given access to a uniformly random function $H : \{0,1\}^n \to \{0,1\}^m$) at distinguishing between a sample drawn from $H(U_n)$ and a sample drawn from U_m. The upper bound is in terms of the probability that the adversary queries the oracle at the pre-image of the sample at some point

[5] While additional improved variants of the one-way to hiding lemma were developed [BHH+19, KSS+20], any of them suffices for our asymptotic analysis.

during its execution. Equivalently, given two oracles that are identical except on a single input (or more generally on a subset of the inputs), the advantage of an adversary at distinguishing the two oracles is bounded above in terms of the probability that the adversary queries at the differing point (or at a point in the subset where they differ) at some point during its execution.

Lemma 7. *Let $\lambda, m \in \mathbb{N}$. For any $q \in \mathbb{N}$, any unitaries U, any family of states $\{|\psi_x\rangle\}_{x\in\mathcal{X}}$, any complete pair of orthogonal projectors (Π^0, Π^1) and any distribution X on $\{0,1\}^\lambda$, it holds that:*

$$\frac{1}{2}\mathbb{E}_H\mathbb{E}_{x\leftarrow X}\|\Pi^0(UO^H)^q\left(|H(x)\rangle\otimes|\psi_x\rangle\right)\|^2+\frac{1}{2}\mathbb{E}_H\mathbb{E}_{z\leftarrow\{0,1\}^m}\|\Pi^1(UO^H)^q\left(|z\rangle\otimes|\psi_x\rangle\right)\|^2 \leq \frac{1}{2}+(3q+2)qM\,, \tag{4}$$

where O^H is the oracle unitary for $H:\{0,1\}^\lambda\to\{0,1\}^m$, and M is given by

$$\begin{aligned} M =& \frac{1}{2}\,\mathbb{E}_H\mathbb{E}_{x\leftarrow X}\mathbb{E}_{z\leftarrow\{0,1\}^m}\mathbb{E}_k\|\,|x\rangle\langle x|\,(UO^{H_{x,z}})^k\,|z\rangle\otimes|\psi_x\rangle\,\| \\ &+\frac{1}{2}\,\mathbb{E}_H\mathbb{E}_{x\leftarrow X}\mathbb{E}_{z\leftarrow\{0,1\}^m}\mathbb{E}_k\|\,|x\rangle\langle x|\,(UO^{H})^k\,|z\rangle\otimes|\psi_x\rangle\,\|\,. \end{aligned} \tag{5}$$

Moreover, M is negligible if and only if the second term in M is negligible.

The lemma holds also when the states $|\psi_x\rangle$ are not necessarily pure (but we write them as pure states for ease of notation).

Proof of Lemma 6. Without loss of generality, let $\mathcal{A}$ be specified by a unitary U, the oracle unitary O^H and a measurement given by projectors Π_0 and $\Pi_1 = \mathbb{1}-\Pi_0$, so that the unitary part of $\mathcal{A}$'s algorithm is $(UO^H)^q$, where q the number of oracle queries made by $\mathcal{A}$. Then, $\mathcal{A}$'s winning probability is precisely given by,

$$\begin{aligned} &\Pr[\mathsf{Distinguish}(\mathcal{A},\lambda,m,X_\lambda)=1] \\ &=\frac{1}{2}\mathbb{E}_H\mathbb{E}_{x\leftarrow X_\lambda}\|\Pi^0(UO^H)^q\,|H(x)\rangle\,\|^2+\frac{1}{2}\mathbb{E}_H\mathbb{E}_{z\leftarrow\{0,1\}^m}\|\Pi^1(UO^H)^q\,|z\rangle\,\|^2\,, \end{aligned} \tag{6}$$

where we omit writing ancilla qubits initialized in the zero state that $(UO^H)^q$ might be acting on.

Then, by Lemma 7, we have

$$\Pr[\mathsf{Distinguish}(\mathcal{A},\lambda,m,X_\lambda)=1]\le\frac{1}{2}+(3q+2)q\,M \tag{7}$$

where M is the quantity given by

$$\begin{aligned} M =& \frac{1}{2}\,\mathbb{E}_H\mathbb{E}_{x\leftarrow X_\lambda}\mathbb{E}_{z\leftarrow\{0,1\}^m}\mathbb{E}_k\|\,|x\rangle\langle x|\,(UO^{H_{x,z}})^k\,|z\rangle\,\| \\ &+\frac{1}{2}\,\mathbb{E}_H\mathbb{E}_{x\leftarrow X_\lambda}\mathbb{E}_{z\leftarrow\{0,1\}^m}\mathbb{E}_k\|\,|x\rangle\langle x|\,(UO^{H})^k\,|z\rangle\,\|\,. \end{aligned} \tag{8}$$

Moreover, by Lemma 7, M is negligible if and only if the second term,

$$\mathbb{E}_H\mathbb{E}_{x\leftarrow X_\lambda}\mathbb{E}_{z\leftarrow\{0,1\}^m}\mathbb{E}_k\|\,|x\rangle\langle x|\,(UO^{H})^k\,|z\rangle\,\|,$$

is negligible. Hence, it suffices to bound the above term. Notice that for any fixed H, z and k,

$$\begin{aligned} &\mathbb{E}_{x\leftarrow X_\lambda}\|\,|x\rangle\langle x|\,(UO^{H})^k\,|z\rangle\,\| \\ &\qquad\le\sqrt{\mathbb{E}_{x\leftarrow X_\lambda}\|\,|x\rangle\langle x|\,(UO^{H})^k\,|z\rangle\,\|^2} \\ &\qquad\le 2^{-\mathbf{H}_{\min}(X_\lambda)/2}\,. \end{aligned} \tag{9}$$

where the first inequality follows from Jensen's inequality (for concave functions), and the second inequality uses the fact that the state $(UO^H)^k\,|z\rangle$ does not depend on x, and hence the quantity under the square root is bounded above by the optimal probability of correctly predicting a sample from X, which is, by definition, $2^{-\mathbf{H}_{\min}(X)}$. Therefore, M is negligible so long as $2^{-\mathbf{H}_{\min}(X_\lambda)}$ is negligible. □

Finally, we define the notion of indistinguishability of ensembles of quantum states *in the QROM*. This is similar to Definition 1.

Definition 2 (Indistinguishability of ensembles of quantum states in the QROM)**.** *Let $m : \mathbb{N} \to \mathbb{N}$ and $p : \mathbb{N} \to \mathbb{N}$ be polynomially bounded functions, and let ρ_λ^H and σ_λ^H be $p(\lambda)$-qubit states, for $H \in \mathsf{Bool}(\lambda, m(\lambda))$. We say that $\{\rho_\lambda^H\}_{\lambda\in\mathbb{N},H\in\mathsf{Bool}(\lambda,m(\lambda))}$ and $\{\sigma_\lambda^H\}_{\lambda\in\mathbb{N},H\in\mathsf{Bool}(\lambda,m(\lambda))}$ are quantum computationally indistinguishable ensembles of quantum states, denoted by $\rho_\lambda^H \approx_c \sigma_\lambda^H$, if, for any QPT distinguisher $\mathcal{D}^H$ with single-bit output, any polynomially bounded $q : \mathbb{N} \to \mathbb{N}$, any family of $q(\lambda)$-qubit auxiliary states $\{\nu_\lambda\}_{\lambda\in\mathbb{N}}$, and every $\lambda \in \mathbb{N}$,*

$$\mathbb{E}_H \left| \Pr[\mathcal{D}^H(\rho_\lambda^H \otimes \nu_\lambda) = 1] - \Pr[\mathcal{D}^H(\sigma_\lambda^H \otimes \nu_\lambda) = 1] \right| \leq \mathsf{negl}(\lambda)\,.$$

3 Quantum copy-protection

Except for some slight differences, our definition of a secure copy-protection scheme is identical to the notion in [Aar09]. We elaborate on the differences in Section 3.1.

Definition 3 (Quantum copy-protection scheme)**.** *Let $\mathcal{C}$ be a family of classical circuits with a single bit output. A quantum copy-protection (CP) scheme for $\mathcal{C}$ is a pair of QPT algorithms $(\mathsf{CP.Protect}, \mathsf{CP.Eval})$ with the following properties:*

- $\mathsf{CP.Protect}$ *takes as input a security parameter $\lambda \in \mathbb{N}$ and a classical circuit $C \in \mathcal{C}$, and outputs a quantum state ρ.*
- $\mathsf{CP.Eval}$ *takes as input a quantum state ρ and a string x, and outputs a single bit.*

We say that the scheme is correct if, for any $\lambda \in \mathbb{N}$, $C \in \mathcal{C}$, and any input string x to C:

$$\Pr[\mathsf{CP.Eval}(\rho, x) = C(x) : \rho \leftarrow \mathsf{CP.Protect}(1^\lambda, C)] \geq 1 - \mathsf{negl}(\lambda).$$

(note that we think of $\mathsf{CP.Protect}$ as a deterministic procedure which outputs a mixed state; so the probability in the above expression comes from $\mathsf{CP.Eval}$.)

Two remarks are in order about Definition 3.

- For ease of exposition, we define copy-protection for circuits outputting a single bit. It is straightforward to generalize the above definition to circuits with multi-bit output.
- While our definition only requires the ability to compute the circuit C on a single input using the copy-protection state ρ, polynomially many evaluations are possible by delaying any measurements in $\mathsf{CP.Eval}$, and uncomputing after copying the output bit to recover the approximate original state ρ.

We define security in terms of a game between a challenger and an adversary consisting of a triple of QPT algorithms $\mathcal{A} = (\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2)$ - a "pirate" $\mathcal{P}$ and two "freeloaders" $\mathcal{F}_1$ and $\mathcal{F}_2$. The game is specified by a security parameter λ, a distribution D_λ over circuits in $\mathcal{C}$, an ensemble $\{D_C\}_{C\in\mathcal{C}}$ where D_C is a distribution over pairs of inputs to $C \in \mathcal{C}$. We refer to $\{D_\lambda\}_{\lambda\in\mathbb{N}}$ as the *program ensemble*, and to $\{D_C\}_{C\in\mathcal{C}}$ as the *input challenge ensemble*. The security game proceeds as follows.

- The challenger samples $C \leftarrow D_\lambda$ and sends $\rho \leftarrow \mathsf{CP.Protect}(1^\lambda, C)$ to $\mathcal{P}$.
- $\mathcal{P}$ creates a state on registers A and B, and sends A to $\mathcal{F}_1$ and B to $\mathcal{F}_2$.
- (*input challenge phase:*) The challenger samples $(x_1, x_2) \leftarrow D_C$ and sends x_1 to $\mathcal{F}_1$ and x_2 to $\mathcal{F}_2$. ($\mathcal{F}_1$ and $\mathcal{F}_2$ are not allowed to communicate).
- $\mathcal{F}_1$ and $\mathcal{F}_2$ each return bits b_1 and b_2 to the challenger.

$\mathcal{A} = (\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2)$ win if $b_1 = C(x_1)$ and $b_2 = C(x_2)$. Let $\mathsf{PiratingGame}(\lambda, \mathcal{P}, \mathcal{F}_1, \mathcal{F}_2, D_\lambda, \{D_C\})$ denote the Boolean random variable for whether the game is won or not.

Before defining security, we define $p^{\mathrm{triv}}_{D_\lambda,\{D_C\}_{C\in\mathcal{C}}}$ to be the winning probability that is trivially possible due to correctness: the pirate forwards the copy-protected program to one of the freeloaders, and leaves the other one with guessing as his best option. Formally, let $\hat{D}_C$ be the

induced distribution of winning answer pairs, and let $\hat{D}_{C,i}$, for $i \in \{1,2\}$ be its marginals. We define the optimal guessing probability of any of the two freeloaders,

$$p^{\text{triv}}_{D_\lambda,\{D_C\}_{C\in\mathcal{C}}} = \max_{i\in\{1,2\}} \max_{b\in\{0,1\}} \mathbb{E}_{C\leftarrow D_\lambda} \hat{D}_{C,i}(b).$$

We define security as follows:

Definition 4 (Security)**.** *A quantum copy-protection scheme for a family of circuits $\mathcal{C}$ is said to be δ-secure with respect to the ensemble $\{D_\lambda\}_{\lambda\in\mathbb{N}}$ of distributions over circuits in $\mathcal{C}$, and with respect to the ensemble $\{D_C\}_{C\in\mathcal{C}}$, where D_C is a distribution over pairs of inputs to program $C \in \mathcal{C}$, if for any* QPT *adversary $(\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2)$, any $\lambda \in \mathbb{N}$,*

$$\Pr[\mathit{PiratingGame}(\lambda, \mathcal{P}, \mathcal{F}_1, \mathcal{F}_2, D_\lambda, \{D_C\}) = 1] \leq 1 - \delta(\lambda) + \mathsf{negl}(\lambda)\,.$$

If $\delta(\lambda) = 1 - p^{\text{triv}}_{D_\lambda,\{D_C\}_{C\in\mathcal{C}}}$, we simply say that the copy-protection scheme is secure.

Two remarks are in order about the above definition.

- The definition can be generalized by quantifying over all challenge distributions. The acceptable adversarial winning probability then needs to be related to the optimal guessing probability for challenges drawn from the distribution. We refrain from such a generalization for ease of exposition.
- We follow Aaronson [Aar05] in that the parameter δ quantifies *security*, not *insecurity*. We decided in favor of this convention to maintain coherence with the previous literature on quantum copy protection, despite the fact that in cryptography, the ϵ in "ϵ-secure" traditionally quantifies adversarial advantage which is a measure of *in*security.

3.1 Comparison with existing definitions of copy-protection

Two security definitions for copy-protection schemes have been considered previously, in [Aar09] and [ALZ20]. Our definition is very similar to the original security definition of [Aar09]. The only difference is the following. In [Aar09], a scheme is δ-secure if for any bounded adversary who tries to create $k+1$ programs upon receiving k copy-protected copies the average number of input challenges answered correctly is $k(1+\delta)$. In contrast, in our definition, a scheme is δ-secure if, for any bounded adversary, the probability that *all* $k+1$ challenges are answered correctly is at most $1-\delta$. It is straightforward to see that these two choices are equivalent up to a factor of k in the security. In the present work, we focus on the case of $k = 1$.

Our definition, and the definition in [Aar09], differ more substantially from the recent definition in [ALZ20]. The definition in [ALZ20] is framed in terms of a security game between a challenger and a pirate, who receives k copy-protected copies of a program (each created independently) and returns to the challenger a state ρ on $k+1$ registers, together with evaluation circuits C_i, for $i \in [k+1]$. The pirate loses if the challenger is able to find an input x such that $C_i(\rho_i, x) \neq f(x)$ for some i.

Such a game can equivalently be recast as a game between a challenger, a pirate, and $k+1$ freeloaders: the only difference with our security game is that in [ALZ20] the challenger can be thought of as being *adaptive*, meaning that it can look at the states ρ_i in order to find a challenge input x on which some freeloader fails. Since this challenger has more power, this results in a weaker security definition. In particular, a scheme which is δ-secure with respect to our security definition, is also δ-secure with respect to the definition in [ALZ20]: the challenger from our definition yields a simple challenger for [ALZ20].

One additional point about the definition in [ALZ20], which makes it a little unnatural, is that it is not operational: the challenger might output an x such that $C_i(\rho_i; x) \neq f(x)$, but she might have destroyed the state ρ_i in the process, and is thus unable to verify this condition on her own.

4 Quantum copy-protection of point functions

In this section, we present a quantum copy-protection scheme for point functions, and prove its security in the quantum random oracle model (QROM).

In what follows, we consider a copy-protection scheme for the class of point functions P_y with marked input $y \in \{0,1\}^n$. Note that, for simplicity, we hand y to the copy-protection algorithm as an input, rather than as a circuit for the point function P_y itself.

Remark 1. *In the following description, the size of the marked input n and the security parameter λ are distinct variables (as they should be in principle). However, the security guarantee that we will prove is with respect to ensembles of programs for which $n = \lambda$. When copy-protecting an ensemble of programs, the level of security cannot be independent of the size of the inputs to the programs, since a pirate with access to the copy-protected program can always determine the whole truth table of the program in time 2^n.*

Recall that for $v, \theta \in \{0,1\}^\lambda$, we use the notation $|v^\theta\rangle = H^\theta |v\rangle$, where $H^\theta = H^{\theta_1} \otimes \ldots \otimes H^{\theta_\lambda}$ is the Hadamard operator on λ many qubits. Our construction is the following:

Construction 1 (Copy-protection scheme for point functions)**.** *Let λ be the security parameter, and let $G : \{0,1\}^n \to \{0,1\}^{m(\lambda)}$ and $H : \{0,1\}^{m(\lambda)} \to \{0,1\}^\lambda$ be hash functions, where $m(\lambda) > \lambda$, and n is the input size of the point function. We define* $(\mathsf{CP.Protect}, \mathsf{CP.Eval})$ *as follows:*

- $\mathsf{CP.Protect}(1^\lambda, y)$*: Takes as input a security parameter λ and a point function P_y, succinctly specified by the marked input y (of size n).*
  - *Set $\theta = G(y)$.*
  - *Sample $v \leftarrow \{0,1\}^{m(\lambda)}$ uniformly at random. Let $z = H(v)$.*
  - *Output $(|v^\theta\rangle, z)$.*
- $\mathsf{CP.Eval}(1^\lambda, (\rho, z); x)$*: Takes as input a security parameter λ, an alleged copy-protected program (ρ, z), and a string $x \in \{0,1\}^n$ (the input on which the program is to be evaluated).*
  - *Set $\theta' = G(x)$.*
  - *Apply the Hadamard operator $H^{\theta'}$ to ρ. Append $n+1$ ancillary qubits, all in state $|0\rangle$, and compute the hash function H with input ρ into the first n of them (possibly making use of additional ancillary qubits). Then, coherently measure whether the first n ancilla qubits are in state $|z\rangle$, recording the result in the last ancilla qubit, uncompute the hash function H and undo the Hadamards $H^{\theta'}$. Finally, measure the last ancilla qubit to obtain a bit b as output.*

In what follows, we will model both G and H in Construction 1 as random oracles on the the appropriate domain and co-domains, i.e. we operate in the quantum random oracle model (QROM). Before stating our main theorem about the security of Construction 1, we define the following class of distributions over programs.

- $\mathcal{D}_{\mathsf{PF\text{-}UNP}}$. The class of *unpredictable point function distributions* $\mathcal{D}_{\mathsf{PF\text{-}UNP}}$ consists of ensembles $D = \{D_\lambda\}$ where D_λ is a distribution over point functions on $\{0,1\}^\lambda$ such that $P_y \leftarrow D_\lambda$ satisfies $\mathbf{H}_{\min}(y) \geq \lambda^\epsilon$ for some $\epsilon > 0$.

We also define the following class of distributions over input challenges.

- $\mathcal{D}_{\mathsf{PF\text{-}Chall}}$. An ensemble $D = \{D_y\}$, where each D_y is a distribution over $\{0,1\}^{|y|} \times \{0,1\}^{|y|}$, belongs to the class $\mathcal{D}_{\mathsf{PF\text{-}Chall}}$ if there exists an efficiently sampleable family $\{X_\lambda\}$ of distributions over $\{0,1\}^\lambda$ with $\mathbf{H}_{\min}(X_\lambda) \geq \lambda^\epsilon$, for some $\epsilon > 0$, such that D_y is the following distribution (where $\lambda = |y|$):
  - with probability $1/3$, sample $x \leftarrow X_\lambda$, and output (x, y).
  - with probability $1/3$, sample $x \leftarrow X_\lambda$, and output (y, x).
  - with probability $1/3$, sample $x, x' \leftarrow X_\lambda$, and output (x, x').

  We say the ensemble D is *specified* by the ensemble X_λ.

We finally define two classes of distributions over pairs of programs and challenges.

- $\mathcal{D}_{\mathsf{PF\text{-}pairs\text{-}stat}}$. This consists of pairs of ensembles $\big(D = \{D_\lambda\}, D' = \{D'_y\}\big)$ such that:
  - $D \in \mathcal{D}_{\mathsf{PF\text{-}UNP}}$ and $D' \in \mathcal{D}_{\mathsf{PF\text{-}Chall}}$.
  - Let D' be specified by the family $\{X_\lambda\}$, and denote by $\mathsf{MarkedInput}(D_\lambda)$ the following distribution over $\{0,1\}^\lambda$: sample $P_y \leftarrow D_\lambda$, and ouput y. We require the families $\{X_\lambda\}$ and $\{\mathsf{MarkedInput}(D_\lambda)\}$ to be statistically indistinguishable.

- $\mathcal{D}_{\mathsf{PF\text{-}pairs\text{-}comp}}$. This is defined in the same way as $\mathcal{D}_{\mathsf{PF\text{-}pairs\text{-}stat}}$, except that we only require $\{X_\lambda\}$ and $\{\mathsf{MarkedInput}(D_\lambda)\}$ to be *computationally* indistinguishable.

We are ready to state our main theorem about security of Construction 1.

Theorem 2. *There exists a constant $\delta^* > 0$ such that, the scheme of Construction 1, with $m(\lambda) > 5\lambda$, is a δ^*-secure quantum copy-protection scheme for point functions with respect to any pair of ensembles $(D, D') \in \mathcal{D}_{\mathsf{PF\text{-}pairs\text{-}stat}}$ ($\in \mathcal{D}_{\mathsf{PF\text{-}pairs\text{-}comp}}$), against query-bounded (computationally bounded) adversaries in the quantum random oracle model (assuming the existence of quantum-secure one-way functions).*

We emphasize that quantum-secure one-way functions are only needed for the computational version of Theorem 2.

Correctness of Construction 1 is immediate to verify, and makes use the fact that G is a random oracle with a range that is sufficiently larger than its domain. The next section is devoted to proving security.

4.1 Proof of security

We assume that $(D, D') \in \mathcal{D}_{\mathsf{PF\text{-}pairs\text{-}stat}}$ (the case of $(D, D') \in \mathcal{D}_{\mathsf{PF\text{-}pairs\text{-}comp}}$ is analogous except for a slight difference in the proof of Lemma 10, which we will point out). Moreover, following the notation for ensembles in $(D, D') \in \mathcal{D}_{\mathsf{PF\text{-}pairs\text{-}stat}}$, we let D' be specified by an efficiently sampleable family $\{X_\lambda\}$ of distributions over $\{0,1\}^\lambda$. As discussed in Remark 1, we assume that $n = \lambda$ in Construction 1.

We will prove Theorem 2 through a sequence of hybrids.

H_0: This is the security game $\mathsf{PiratingGame}$ for the copy-protection scheme of Construction 1.

- The challenger samples a point function $P_y \leftarrow D_\lambda$ with $y \in \{0,1\}^\lambda$, and sends the state $(|v^\theta\rangle, z) \leftarrow \mathsf{CP.Protect}(1^\lambda, y)$, where $\theta = G(y)$ and $z = H(v)$, to the pirate $\mathcal{P}$.
- $\mathcal{P}$ creates a state on registers A and B, and sends A to $\mathcal{F}_1$ and B to $\mathcal{F}_2$.
- (*input challenge phase:*) The challenger samples $(x_1, x_2) \leftarrow D'_y$ and sends x_1 to $\mathcal{F}_1$ and x_2 to $\mathcal{F}_2$. ($\mathcal{F}_1$ and $\mathcal{F}_2$ are not allowed to communicate).
- $\mathcal{F}_1$ and $\mathcal{F}_2$ return b_1 and b_2, respectively, and win if $b_1 = P_y(x_1)$ and $b_2 = P_y(x_2)$.

H_1: Same as H_0, except that in the input challenge phase the challenger samples $(x_1, x_2) \leftarrow D'_y$. Then, it sends $G(x_1)$ and $G(x_2)$ to $\mathcal{F}_1$ and $\mathcal{F}_2$ respectively (instead of sending x_1 and x_2 directly).

H_2: Same as H_1, except for the following. During the input challenge phase, the challenger samples $(x_1, x_2) \leftarrow D'_y$. Then, for $i \in \{1,2\}$, if $x_i \neq y$, the challenger samples $\theta'_i \leftarrow \{0,1\}^{m(\lambda)}$, and sends θ'_i to $\mathcal{F}_i$ instead of $G(x_i)$.

H_3: Same as H_2, except that in the first step of the security game, the challenger samples $\theta \leftarrow \{0,1\}^{m(\lambda)}$ (as opposed to sampling $P_y \leftarrow D_\lambda$ and setting $\theta = G(y)$). Then, in the input challenge phase, if $x_i = y$, the challenger sends $\theta_i := \theta$ to $\mathcal{F}_i$, and sends a uniformly random θ'_i otherwise.

H_4: Same as H_3, except that the challenger samples $z \leftarrow \{0,1\}^\lambda$ instead of choosing $z = H(v)$.

H_5: Same as H_4, except the pirate gets the challenge inputs θ_1 and θ_2 together with the copy-protected program.

In the rest of the section, we say that x is a 0-input (1-input) to a boolean function f, if $f(x) = 0$ ($f(x) = 1$). We prove the following two lemmas, which together give Theorem 2.

Lemma 8. *For any adversary $\mathcal{A}$,*

$$\Pr[\mathcal{A} \textit{ wins } H_5] \leq \frac{1}{3}.$$

In the rest of the section, for any function $g : [0,1] \to [0,1]$, we use the notation

$$p(H_i) > g(p(H_j)) - \mathsf{negl}(\lambda)$$

as a shorthand for the following: for any adversary $\mathcal{A}$ for H_j, there exists an adversary $\mathcal{A}'$ for H_i and a negligible function μ such that

$$\Pr[\mathcal{A}' \text{ wins } H_i] > g(\Pr[\mathcal{A} \text{ wins } H_j]) - \mu(\lambda)\,.$$

Informally, one can think of $p(H_i)$ the optimal winning probability in hybrid H_i (up to negligible functions in the security parameter).

Lemma 9. *There exists a constant $\delta^* > 0$ such that*

$$p(H_5) > p(H_0) - 2/3 + \delta^* - \mathsf{negl}(\lambda)\,. \tag{10}$$

Lemmas 8 and 9 immediately imply that, for any adversary $\mathcal{A}$ for H_0,

$$\Pr[\mathcal{A} \text{ wins } H_0] < 1 - \delta^* + \mathsf{negl}(\lambda)\,,$$

which gives Theorem 2.

Proof of Lemma 8. Denote by x_1 and x_2 the inputs sampled by the challenger. There are three cases: x_1 and x_2 are both 0-inputs; x_1 is a 0-input and x_2 is a 1-input; x_1 is a 1-input and x_2 is a 0-input. We argue that the density matrix ρ that is handed to the pirate in all three cases is a maximally mixed state. More precisely, when $P_y(x_1) = 0, P_y(x_2) = 0$, the state ρ that the pirate receives is the following, which is completely independent of the oracle H:

$$\begin{aligned}&\mathbb{E}_{v,\theta,\theta',\theta'',z} \left|v^\theta\right\rangle \left\langle v^\theta\right| \otimes \left|\theta'\right\rangle \left\langle\theta'\right| \otimes \left|\theta''\right\rangle \left\langle\theta''\right| \otimes \left|z\right\rangle \left\langle z\right| \\ &= \frac{\mathbb{1}}{2^{3m+\lambda}}\,,\end{aligned} \tag{11}$$

When $P_y(x_1) = 0, P_y(x_2) = 1$, the state ρ that the pirate receives is again completely independent of the oracle, and is the following:

$$\begin{aligned}&\mathbb{E}_{v,\theta,\theta',z} \left|v^\theta\right\rangle \left\langle v^\theta\right| \otimes \left|\theta'\right\rangle \left\langle\theta'\right| \otimes \left|\theta\right\rangle \left\langle\theta\right| \otimes \left|z\right\rangle \left\langle z\right| \\ &= \frac{\mathbb{1}}{2^{3m+\lambda}}\,,\end{aligned} \tag{12}$$

where crucially that the state is still maximally mixed.

The third case is analogous to the second. Thus it is impossible, even for an unbounded pirate to distinguish the three cases with any advantage over random guessing. □

We prove Lemma 9 by keeping track of how the optimal winning probability changes across hybrids. We break down the proof into the following lemmas.

Lemma 10. $p(H_1) \geq p(H_0) - \mathsf{negl}(\lambda)$.

Lemma 11. $|p(H_1) - p(H_2)| = \mathsf{negl}(\lambda)$.

Lemma 12. $|p(H_2) - p(H_3)| = \mathsf{negl}(\lambda)$.

Lemma 13. $p(H_4) > p(H_3) - 2/3 + \delta^* - \mathsf{negl}(\lambda)$, *for some constant* $\delta^* > 0$.

Lemma 14. $p(H_5) \geq p(H_4)$.

Lemma 9 follows immediately from Lemmas 10-14. The crux is Lemma 13.

Proof of Lemma 10. Let $\mathcal{A} = (\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2)$ be a query-bounded adversary that wins with probability p in H_0. We will construct an adversary $\mathcal{A}'$ that wins with probability at least $p - \mathsf{negl}(\lambda)$ in hybrid H_1. The adversary $\mathcal{A}' = (\mathcal{P}', \mathcal{F}_1', \mathcal{F}_2')$ is the following:

- Upon receiving $(|\Psi\rangle, z)$ from the challenger, where $|\Psi\rangle = |v^{G(y)}\rangle$ for some string $y \in \{0,1\}^\lambda$, $\mathcal{P}'$ samples a uniformly random function $\hat{G} : \{0,1\}^\lambda \to \{0,1\}^{m(\lambda)}$ (in the computational version of the argument, $\mathcal{P}'$ samples instead a function $\hat{G}$ from a quantum-secure pseudo-random function (PRF) family. Note that a quantum-secure PRF family can be constructed from any quantum-secure one-way function [Zha12]). Subsequently, $\mathcal{P}'$ runs $\mathcal{P}$ on input $(|\Psi\rangle, z)$, using $\hat{G}$ to respond to queries to G. $\mathcal{P}'$ forwards the two registers A and B output by $\mathcal{P}$ to $\mathcal{F}_1'$ and $\mathcal{F}_2'$, respectively, together with a description of $\hat{G}$.
- $\mathcal{F}_1'$ and $\mathcal{F}_2'$ receive $w_1 = G(x_1)$ and $w_2 = G(x_2)$ respectively from the challenger, for some x_1 and x_2. $\mathcal{F}_1'$ and $\mathcal{F}_2'$ sample $x_1', x_2' \leftarrow X_\lambda$. $\mathcal{F}_1'$ runs $\mathcal{F}_1$ on input (x_1', A) and responds to oracle queries to G using $\hat{G}_{x_1', w_1}$, where

$$\hat{G}_{x_1', w_1}(x) = \begin{cases} \hat{G}(x) & \text{if } x \neq x', \\ w_1 & \text{if } x = x_1'. \end{cases}$$

  $\mathcal{F}_1'$ returns to the challenger the output of $\mathcal{F}_1$. $\mathcal{F}_2'$ proceeds analogously.

We claim that the success probability of $(\mathcal{P}', \mathcal{F}_1', \mathcal{F}_2')$ is at least $p - \mathsf{negl}(\lambda)$. We leave the full details of the proof to Appendix A.2, as these are lengthy but not particularly enlightening. Here, why provide some intuition about the reduction. The crucial observation is that there is nothing special about the the marked input y in H_0 other than the fact that the oracle G maps it to the correct basis choice used by the challenger. What $\mathcal{P}'$ does (without ever seeing y) is come up with x_1' and x_2', and then reprogram the oracle so that x_1' and x_2' map to the (correct or incorrect) basis choices he received as part of the H_1 game. Crucially, x_1 and x_2 are not sampled uniformly at random, but from the distribution X_λ which is statistically indistinguishable from the distribution of marked inputs from which y is sampled (by definition of the challenge distribution $\mathcal{D}_{\mathsf{PF\text{-}pairs\text{-}stat}}$). This makes the distribution of the game played by the invoked adversary $(\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2)$ essentially the same as in the game H_0. The only difference between the two games is that they involve slightly different oracles, respectively $\hat{G}$ and G. Replacing one with the other can be done without affecting the winning probability by more than a negligible amount as long as P does not query at x_1' and x_2' with non-negligible probability, which is the case.

We point out that the proof of this lemma is the only step in the proof of Theorem 2 in which one uses the indistinguishability condition in the definition of $\mathcal{D}_{\mathsf{PF\text{-}pairs\text{-}stat}}$ (and of $\mathcal{D}_{\mathsf{PF\text{-}pairs\text{-}comp}}$ for the computational version of the result). This completes the proof of Lemma 10. We point out that the first step in the proof of the computational version of Lemma 10 is to argue that using a pseudorandom function (PRF) instead of a uniformly random function $\hat{G}$ does not change the success probability of $(\mathcal{P}', \mathcal{F}_1', \mathcal{F}_2')$ more than negligibly, which is straightforward. The rest of the proof for the version of Theorem 2 with $\mathcal{D}_{\mathsf{PF\text{-}pairs\text{-}comp}}$ is analogous to the proof for the version with $\mathcal{D}_{\mathsf{PF\text{-}pairs\text{-}stat}}$. □

Proof of Lemma 11. Suppose for a contradiction that the lemma is false. Let $\mathcal{A} = (\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2)$ be a query-bounded adversary that wins H_1 with probability noticeably higher than H_2 (the reverse case being similar). We construct a query-bounded adversary $\mathcal{A}'$ that distinguishes samples from the distributions $G(X_\lambda)$ and $U_{m(\lambda)}$, which cannot exist by Corollary 2. We assume here that $\mathcal{A}'$ has access to two samples from either $G(X_\lambda)$ or $U_{m(\lambda)}$ (this assumption is justified by the equivalence of single-sample and polynomially many-sample distinguishing tasks for efficiently sampleable distributions). $\mathcal{A}'$ receives as input two samples $\theta_1, \theta_2 \in \{0,1\}^{m(\lambda)}$. $\mathcal{A}'$ then simulates the challenger in a copy-protection game of hybrid H_1 with $\mathcal{A}$, in the following way: in the input challenge phase, it samples $(x_1, x_2) \leftarrow D_y'$. For each $i \in \{1,2\}$, if $x_i \neq y$, $\mathcal{A}'$ sends θ_i to $\mathcal{F}_i$ (otherwise sends $G(y)$). If $\mathcal{A}$ wins the game, $\mathcal{A}'$ guesses that the sample was from $G(X_\lambda)$, otherwise that it was from $U_{m(\lambda)}$. □

Proof of Lemma 12. The proof is similar to the previous lemma. Suppose for a contradiction that the lemma is false. Let $\mathcal{A} = (\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2)$ be a query-bounded adversary that wins H_2 with probability noticeably higher than H_3 (the reverse case being similar). We construct a query-bounded adversary $\mathcal{A}'$ that distinguishes samples from the distributions $G(X_\lambda)$ and $U_{m(\lambda)}$, which cannot exist by Corollary 2. $\mathcal{A}'$ receives as input a challenge $\theta \in \{0,1\}^{m(\lambda)}$. $\mathcal{A}'$ then simulates the challenger in a copy-protection game of hybrid H_2 with $\mathcal{A}$ as follows. In the input challenge

phase, it samples $(x_1, x_2) \leftarrow D'_y$. For each $i \in \{1,2\}$, if $x_i = y$, $\mathcal{A}'$ sends θ to $\mathcal{F}_i$. If $\mathcal{A}$ wins the game, $\mathcal{A}'$ guesses that the sample was from $G(X_\lambda)$, otherwise that it was from $U_{m(\lambda)}$. □

Proof of Lemma 14. Given an adversary $\mathcal{A}$ that wins with probability p in H_4, the adversary $\mathcal{A}'$ for H_5 which acts identically to $\mathcal{A}$ (i.e. $\mathcal{P}$ ignores the additional inputs θ_1 and θ_2 received at the start, and simply forwards them to $\mathcal{F}_1$ and $\mathcal{F}_2$ respectively) clearly wins with probability p.

□

We move to the crux of the proof, Lemma 13. We break down the proof of Lemma 13 into a few technical lemmas.

Lemma 15.

$$\Big|\Pr\Big[(\mathcal{P},\mathcal{F}_1,\mathcal{F}_2)\ \textit{win}\ H_4 \,|\, x_1\ \textit{is a 0-input and}\ x_2\ \textit{is a 0-input}\Big] - \Pr\Big[(\mathcal{P},\mathcal{F}_1,\mathcal{F}_2)\ \textit{win}\ H_3 \,|\, x_1\ \textit{is a 0-input and}\ x_2\ \textit{is a 0-input}\Big]\Big| = \mathsf{negl}(\lambda) \quad (13)$$

Proof. Notice that the task of distinguishing H_3 and H_4 is precisely amenable to the one-way-to-hiding lemma (in its form of Lemma 7). By an application Lemma 7, it is sufficient to show that the probability that an adversary, with access to all of the information received in H_4 (including the input phase challenges, which in this case are independent of the correct basis choice), queries the oracle at v is negligible. The latter is true, since an adversary violating this straightforwardly gives an adversary which contradicts Lemma 4. □

The next lemma is the crucial step in the proof of Lemma 13.

Lemma 16. *For any bounded strategy $(\mathcal{P},\mathcal{F}_1,\mathcal{F}_2)$, there exists a negligible function μ such that, for all λ, there exists an $i \in \{1,2\}$ such that:*

$$\Big|\Pr\big[\mathcal{F}_i\ \textit{returns 1 in}\ H_4 \,|\, x_i\ \textit{is a 1-input}\ \big] - \Pr\big[\mathcal{F}_i\ \textit{returns 1 in}\ H_3 \,|\, x_i\ \textit{is a 1-input}\ \big]\Big| < 1 - \epsilon^* + \mu(\lambda)\,, \quad (14)$$

where we can take $\epsilon^* = 10^{-4}$.

The proof of Lemma 16 is fairly involved, and constitutes the main technical work in this paper. We break the proof down into two parts:

- First, we show that an adversary violating Lemma 16 can be used to construct an adversary that wins at a certain r-fold parallel repetition of an appropriate distinguishing game (where one should think of r as a small constant), which we call G_r (Lemma 18).
- Next, we prove a technical lemma (Lemma 19, a generalization of a similar lemma in [BL19]), which allows us to upper bound the probability of a query-bounded adversary winnning in G_r (Lemma 20). Such an upper bound relies on Lemma 19 as well as on properties of monogamy of entanglement games.

We discuss informally below, after the description of the game G_r, the reason why we consider the r-fold parallel repetition of a natural distinguishing game.

Let $\epsilon > 0$. Suppose for a contradiction that there exists a strategy $(\mathcal{P},\mathcal{F}_1,\mathcal{F}_2)$ such that, for all negligible functions μ, there exists some λ such that, for all $i \in \{1,2\}$,

$$\Big|\Pr[\mathcal{F}_i \text{ returns 1 in } H_4|x_i \text{ is a 1-input }] - \Pr[\mathcal{F}_i \text{ returns 1 in } H_3|x_i \text{ is a 1-input }]\Big| \geq 1 - \epsilon + \mu(\lambda)\,. \quad (15)$$

Notice, in particular, that the above straightforwardly implies that, for all negligible functions μ, there exist *infinitely many* λ's such that, for all $i \in \{1,2\}$,

$$\Big|\Pr[\mathcal{F}_i \text{ returns 1 in } H_4|x_i \text{ is a 1-input }] - \Pr[\mathcal{F}_i \text{ returns 1 in } H_3|x_i \text{ is a 1-input }]\Big| \geq 1 - \epsilon + \mu(\lambda)\,. \quad (16)$$

Let $r \in \mathbb{N}$. Given such $(\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2)$, we will construct an adversary $(\mathcal{P}', \mathcal{F}_1', \mathcal{F}_2')$ that succeeds at the following game G_r between a challenger and an adversary $(\mathcal{P}', \mathcal{F}_1', \mathcal{F}_2')$ with probability at least $1 - g(\epsilon)$, for some continuous, monotonically increasing function $g(\epsilon)$ such that $g(0) = 0$. Given a function H, we denote by $H_{x,y}$ the function such that $H_{x,y}(x') = H(x')$ for all $x' \neq x$, and $H_{x,y}(x) = y$. We describe game G_r as follows:

(i) The challenger samples $\theta, v \leftarrow \{0,1\}^{m(\lambda)}$, $z' \leftarrow \{0,1\}^{\lambda}$, and $w \leftarrow \{0,1\}^r$ as well as a random oracle $H : \{0,1\}^{m(\lambda)} \to \{0,1\}^{\lambda}$. The challenger then sends $(|v^{\theta}\rangle, z)$ with $z = H(v)$ to $\mathcal{P}'$ who gets oracle access to $H_1, .., H_r$, where

$$H_i = \begin{cases} H & \text{if } w_i = 0\,, \\ H_{v,z'} & \text{if } w_i = 1\,. \end{cases}$$

(ii) $\mathcal{P}'$ creates a bipartite state σ, and sends its two subsystems to $\mathcal{F}_1'$ and $\mathcal{F}_2'$, respectively.

(iii) The challenger sends θ to both $\mathcal{F}_1'$ and $\mathcal{F}_2'$.

(iv) $\mathcal{F}_1'$ and $\mathcal{F}_2'$ (who cannot communicate) return w' and w'' in $\{0,1\}^r$ respectively to the challenger.

$(\mathcal{P}', \mathcal{F}_1', \mathcal{F}_2')$ win if $w' = w'' = w$.

It might seem mysterious why we consider such r-fold parallel repetition of the more natural distinguishing game with $r = 1$. The reason will become more apparent later in the proof. For now, informally the reason is the following: in order to complete the proof of security, we will need to employ a decision version of a one-way-to-hiding lemma for entangled parties. This does not straightforwardly follow from the search version in [BL19], because of a factor of 9 loss in the security compared to the non-entangled version the lemma. The parallel repetition allows to overcome the security loss, and to obtain a non-trivial security statement.

The reduction we use to construct $(\mathcal{P}', \mathcal{F}_1', \mathcal{F}_2')$ winning at game G_r uses the so-called "Gentle Measurement Lemma" or "Almost As Good As New Lemma" [Win99, Aar05]: if a measurement succeeds with high probability, then the post-measurement state conditioned on success, is close to the initial state. Specifically, we need the following lemma, which is a consequence of the aforementioned lemmas.

Lemma 17. *Let $r \in \mathbb{N}$, let $|\psi\rangle$ to be a unit vector, and let $\{\Pi_i\}_{i\in[r]}$ be such that for every $i \in [r]$:*

$$\|\Pi_i |\psi\rangle\|_2^2 \geq 1 - \epsilon.$$

Then, it follows that:

$$\|\Pi_r \Pi_{r-1} ... \Pi_1 |\psi\rangle\|_2^2 \geq 1 - 2r\sqrt{\epsilon}.$$

Proof. Let $|\delta_i\rangle = (\mathbb{1} - \Pi_i)|\psi\rangle$. We prove by induction that

$$\Pi_\ell \Pi_{\ell-1} ... \Pi_1 |\psi\rangle = |\psi\rangle + |\eta_\ell\rangle\,,$$

for some vector $|\eta_\ell\rangle$ such that $\| |\eta_\ell\rangle \|_2 \leq \ell\sqrt{\epsilon}$. The statement clearly holds for $\ell = 0$. Assuming it holds up to $\ell - 1$, we get

$$\begin{aligned} \Pi_\ell \Pi_{\ell-1} ... \Pi_1 |\psi\rangle &= \Pi_\ell \left(|\psi\rangle + |\eta_{\ell-1}\rangle\right) \\ &= |\psi\rangle - |\delta_\ell\rangle + \Pi_\ell |\eta_{\ell-1}\rangle \\ &=: |\psi\rangle + |\eta_\ell\rangle\,, \end{aligned}$$

where $\| |\eta_\ell\rangle \|_2 \leq \ell\sqrt{\epsilon}$ by the triangle inequality, the fact that projectors have unit operator norm, and the inductive hypothesis. One more application of the triangle inequality shows that:

$$\begin{aligned} \|\Pi_r \Pi_{r-1} ... \Pi_1 |\psi\rangle\|_2^2 &= \| |\psi\rangle + |\eta_r\rangle \|_2^2 \\ &\geq \left(1 - r\sqrt{\epsilon}\right)^2 \\ &\geq 1 - 2r\sqrt{\varepsilon}. \end{aligned}$$

□

We apply the above lemma to construct a strategy that wins at G_r.

Lemma 18. *Suppose $(\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2)$ satisfies (16) for some $\epsilon > 0$. Then, there exists $(\mathcal{P}', \mathcal{F}_1', \mathcal{F}_2')$ such that, for any negligible function ν, there exist infinitely many λ, such that $(\mathcal{P}', \mathcal{F}_1', \mathcal{F}_2')$ wins game G_r with security parameter λ with probability at least $1 - 4r\sqrt{\epsilon} + \nu(\lambda)$.*

Proof. Note that we can equivalently recast the game in hybrid H_3 as a game in which $\mathcal{P}$ receives a uniformly random string z (instead of $H(v)$), but then the oracle is reprogrammed at v, so that $\mathcal{P}$ has oracle access to $H_{v,z}$ instead of H, where $H_{v,z}(x) = H(x)$ if $x \neq v$, and $H(v) = z$.

Let q be the number of queries $\mathcal{P}$ makes. Without loss of generality, $\mathcal{P}$'s strategy is specified by a unitary U and takes the form $(UO^H)^q$.

Notice that

$$\mathbb{E}_H \mathbb{E}_v \mathbb{E}_\theta \mathbb{E}_z \mathbb{E}_{k\in[q]} \left\| |v\rangle \langle v| \left(UO^{H_{v,z}}\right)^k |v^\theta\rangle \otimes |z\rangle \right\|^2 = \mathsf{negl}(\lambda)\,. \tag{17}$$

Suppose for a contradiction that the latter was not the case, then the pirate could recover the classical string v with non-negligible probability and send v to both $\mathcal{F}_1$ and $\mathcal{F}_2$. This would give a strategy that wins the monogamy game (more precisely the variant of Lemma 4).

By an application of a suitable variation of the one-way-to-hiding lemma (Lemma 7), the global state after $\mathcal{P}$'s action, including the challenger's registers, is negligibly close in hybrids H_3 and H_4, i.e. there exists a negligible function ν', such that

$$\begin{aligned}\Big\| &\mathbb{E}_H \mathbb{E}_v \mathbb{E}_\theta \mathbb{E}_z |H\rangle\langle H| \otimes |v\rangle\langle v| \otimes |\theta\rangle\langle\theta| \otimes \left((UO^{H_{v,z}})^q |v^\theta\rangle\langle v^\theta| \otimes |z\rangle\langle z| \left((UO^{H_{v,z}})^q \right)^\dagger \right) \\ &- \mathbb{E}_H \mathbb{E}_v \mathbb{E}_\theta \mathbb{E}_z |H\rangle\langle H| \otimes |v\rangle\langle v| \otimes |\theta\rangle\langle\theta| \otimes \left((UO^{H})^q |v^\theta\rangle\langle v^\theta| \otimes |z\rangle\langle z| \left((UO^{H})^q \right)^\dagger \right) \Big\|_1 = \nu'(\lambda)\,.\end{aligned} \tag{18}$$

Now, by hypothesis, for any negligible function μ, for all $i \in \{1,2\}$ there exist infinitely many λ such that $(\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2)$ satisfies expression (16). In particular, we will use this hypothesis for a choice of μ sufficiently large (with respect to the negligible functions ν and ν').

We will also assume that, in (16), it is

$$\Pr[\mathcal{F}_i \text{ returns 1 in } H_4 | x_i \text{ is a 1-input }] > \Pr[\mathcal{F}_i \text{ returns 1 in } H_3 | x_i \text{ is a 1-input }]\,,$$

for infinitely many such λ's (the reverse case being similar). Let Λ_{good} be the corresponding set of such λ's.

We will construct an adversary $(\mathcal{P}', \mathcal{F}_1', \mathcal{F}_2')$ for the game G_r with security parameter $\lambda \in \Lambda_{\mathsf{good}}$, which wins with probability at least $1 - 4r\sqrt{\epsilon} + \nu(\lambda)$. The adversary $(\mathcal{P}', \mathcal{F}_1', \mathcal{F}_2')$ is as follows:

- $\mathcal{P}'$ receives ρ and z from the challenger and runs $\mathcal{P}$ on input ρ and z and using oracle H_1 to answer any query by $\mathcal{P}$ (although any other H_i would be fine). Let $\tilde{\rho}$ (a bipartite state) be the output. $\mathcal{P}'$ sends the first subsystem to $\mathcal{F}_1'$ and the second subsystem to $\mathcal{F}_2'$.
- $\mathcal{F}_1'$ and $\mathcal{F}_2'$ receive θ from the challenger. For $i \in [r]$, they do the following:
  - Let $b \in \{1,2\}$, $\mathcal{F}_b'$ runs $\mathcal{F}_b$ on input the system received from $\mathcal{P}'$ and θ, using oracle H_i. Let $w_i^{(b)}$ be the outcome. $\mathcal{F}_b'$ runs the inverse of the computation run by $\mathcal{F}_b$ before the measurement.
- $\mathcal{F}_b'$ returns the string $(w_1^{(b)}, \ldots, w_r^{(b)})$ to the challenger.

For $i \in [r]$, let U_i^b be the unitary implemented by $\mathcal{F}_b$ when run with oracle H_i, and let Π_i^b be the projector corresponding to the correct outcome in the final measurement. Define the projectors $\widetilde{\Pi}_i^b = \left(U_i^b\right)^\dagger \Pi_i^b U_i^b$. Crucially, Equation (18), together with the hypothesis of Lemma 18 (used with a sufficiently large negligible function μ), imply the following:

- If $|\psi\rangle$ is a purification of the state $\tilde{\rho}$ returned by $\mathcal{P}$ to $\mathcal{P}'$, then, for all $b \in \{1,2\}$, $i \in [r]$:

$$\| \widetilde{\Pi}_i^b |\psi\rangle \|^2 \geq 1 - \epsilon + \mu'(\lambda) \tag{19}$$

for a sufficiently large negligible function μ', and for infinitely many λ.

Applying the variant of the "Gentle Measurement Lemma" from Lemma 17 implies that $\mathcal{F}'_b$ succeeds with probability at least $1-2r\sqrt{\epsilon}+\frac{1}{2}\nu(\lambda)$. By a union bound, we conclude that $(\mathcal{P}', \mathcal{F}'_1, \mathcal{F}'_2)$ win with probability at least $1-4r\sqrt{\epsilon}+\nu(\lambda)$, for infinitely many λ, as desired. □

Next, our goal is to upper bound the probability that a query-bounded adversary $(\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2)$ wins G_r by the probability that $\mathcal{F}_1$ and $\mathcal{F}_2$ simultaneously query the oracle at point v. This is captured by the following technical lemma, which we specialize to our exact case in the subsequent corollary. Before stating the lemma, we introduce some notation. Let $F : \mathcal{X} \to \mathcal{Y}$. Let $\underline{v} \in \mathcal{X}^r$ be such that all $\underline{v}_i$ are distinct, for $i \in [r]$, and let $\underline{z} \in \mathcal{Y}^r$. We denote by $F_{\underline{v},\underline{z}}$ the function,

$$F_{\underline{v},\underline{z}}(x) = \begin{cases} F(x) & \text{if } x \neq \underline{v}_i \text{ for all } i\,, \\ \underline{z}_i & \text{if } x = \underline{v}_i\,. \end{cases}$$

Lemma 19. *Let $\mathcal{X}, \mathcal{Y}, \mathcal{R}$ be sets of binary strings. Let $r \in \mathbb{N}$. Let F be a function-valued random variable, where each function is from $\mathcal{X}$ to $\mathcal{Y}$. Let $\mathcal{Z} \subseteq \mathcal{Y}^r$. For any $\underline{v} \in \mathcal{X}^r$, any injective function $g : \mathcal{Z} \to \mathcal{R}$, any distribution Z on $\mathcal{Z}$, any unitaries $U_{\mathsf{B}}, U_{\mathsf{C}}$, any bipartite state $|\psi\rangle$, any $q_{\mathsf{B}}, q_{\mathsf{C}} \in \mathbb{N}$, the following holds:*

$$\mathbb{E}_F \mathbb{E}_{\underline{z} \leftarrow Z} \left\| \Pi^{g(\underline{z})} \left(U_{\mathsf{B}} O_{\mathsf{B}}^{F_{\underline{v},\underline{z}}} \right)^{q_{\mathsf{B}}} \otimes \left(U_{\mathsf{C}} O_{\mathsf{C}}^{F_{\underline{v},\underline{z}}} \right)^{q_{\mathsf{C}}} |\psi\rangle \right\|^2 \tag{20}$$

$$\leq 9p_{\max} + \mathsf{poly}(q_{\mathsf{B}}, q_{\mathsf{C}})\sqrt{M}\,, \tag{21}$$

where $\Pi^w = \Pi^w_{\mathsf{B}} \otimes \Pi^w_{\mathsf{C}}$, $p_{\max} = \max_{r \in \mathcal{R}} \Pr[g(\underline{z}) = r : \underline{z} \leftarrow Z]$, and

$$M = \mathbb{E}_k \mathbb{E}_l \mathbb{E}_F \mathbb{E}_{\underline{z} \leftarrow \mathcal{Z}} \left\| (P_{\underline{v}} \otimes P_{\underline{v}}) \left(U_{\mathsf{B}} O_{\mathsf{B}}^{F_{\underline{v},\underline{z}}})^k \otimes (U_{\mathsf{C}} O_{\mathsf{C}}^{F_{\underline{v},\underline{z}}})^l \right) |\psi\rangle \right\|^2,$$

where $P_{\underline{v}} = \sum_{i=1}^r |\underline{v}_i\rangle \langle \underline{v}_i|$.

Proof. The following proof is similar to the proof of Lemma 21 in [BL19], but extended to a slightly more general setting.

For $\mathsf{L} \in \{\mathsf{B}, \mathsf{C}\}$, let $V_{\mathsf{L}}^F = \left(U_{\mathsf{L}} O_{\mathsf{L}}^F (\mathbb{1} - P_{\underline{v}})\right)^{q_{\mathsf{L}}}$ and $W_{\mathsf{L}}^F = (U_{\mathsf{L}} O_{\mathsf{L}}^F)^{q_{\mathsf{L}}} - V_{\mathsf{L}}^F$.

Fix $\underline{v} \in \mathcal{X}^r$, a set $\mathcal{Z} \subseteq \mathcal{Y}^r$ and $\underline{z} \in \mathcal{Z}$. We have the following.

$$\begin{aligned}
&\|\Pi^{g(\underline{z})}) \left(\left(U_{\mathsf{B}} O_{\mathsf{B}}^{F_{\underline{v},\underline{z}}} \right)^{q_{\mathsf{B}}} \otimes \left(U_{\mathsf{C}} O_{\mathsf{C}}^{F_{\underline{v},\underline{z}}} \right)^{q_{\mathsf{C}}} \right) |\psi\rangle \|^2 \\
&= \|\Pi^{g(\underline{z})} \left(\left(U_{\mathsf{B}} O_{\mathsf{B}}^{F_{\underline{v},\underline{z}}} \right)^{q_{\mathsf{B}}} \otimes V_{\mathsf{C}}^{F_{\underline{v},\underline{z}}} + V_{\mathsf{B}}^{F_{\underline{v},\underline{z}}} \otimes W_{\mathsf{C}}^{F_{\underline{v},\underline{z}}} + W_{\mathsf{B}}^{F_{\underline{v},\underline{z}}} \otimes W_{\mathsf{C}}^{F_{\underline{v},\underline{z}}} \right) |\psi\rangle \|^2 && (22) \\
&\leq \|\Pi^{g(\underline{z})} \left(\left(U_{\mathsf{B}} O_{\mathsf{B}}^{F_{\underline{v},\underline{z}}} \right)^{q_{\mathsf{B}}} \otimes V_{\mathsf{C}}^{F_{\underline{v},\underline{z}}} + V_{\mathsf{B}}^{F_{\underline{v},\underline{z}}} \otimes W_{\mathsf{C}}^{F_{\underline{v},\underline{z}}} \right) |\psi\rangle \|^2 \\
&+ (3q_{\mathsf{B}}q_{\mathsf{C}} + 2) q_{\mathsf{B}} q_{\mathsf{C}}\, \mathbb{E}_k \mathbb{E}_l \left\| P_{\underline{v}} \otimes P_{\underline{v}} \left(\left(U_{\mathsf{B}} O_{\mathsf{B}}^{F_{\underline{v},\underline{z}}} \right)^k \otimes \left(U_{\mathsf{C}} O_{\mathsf{C}}^{F_{\underline{v},\underline{z}}} \right)^l \right) |\psi\rangle \right\|, && (23)
\end{aligned}$$

where the inequality follows from a triangle inequality together with a bound used in [BL19] (more precisely, Lemma 18 in [BL19]). Applying Jensen's inequality and the definition of M, we obtain

$$\begin{aligned}
&\mathbb{E}_F \mathbb{E}_{\underline{z} \leftarrow \mathcal{Z}} \|\Pi^{g(\underline{z})} \left(\left(U_{\mathsf{B}} O_{\mathsf{B}}^{F_{\underline{v},\underline{z}}} \right)^{q_{\mathsf{B}}} \otimes \left(U_{\mathsf{C}} O_{\mathsf{C}}^{F_{\underline{v},\underline{z}}} \right)^{q_{\mathsf{C}}} \right) |\psi\rangle \|^2 \\
&\leq \mathbb{E}_F \mathbb{E}_{\underline{z} \leftarrow \mathcal{Z}} \|\Pi^{g(\underline{z})} \left(\left(U_{\mathsf{B}} O_{\mathsf{B}}^{F_{\underline{v},\underline{z}}} \right)^{q_{\mathsf{B}}} \otimes V_{\mathsf{C}}^{F_{\underline{v},\underline{z}}} + V_{\mathsf{B}}^{F_{\underline{v},\underline{z}}} \otimes W_{\mathsf{C}}^{F_{\underline{v},\underline{z}}} \right) |\psi\rangle \|^2 && (24) \\
&+ (3q_{\mathsf{B}}q_{\mathsf{C}} + 2) q_{\mathsf{B}} q_{\mathsf{C}} \sqrt{M} && (25)
\end{aligned}$$

We will show that, for any fixed F,

$$\mathbb{E}_{\underline{z} \leftarrow \mathcal{Z}} \|\Pi^{g(\underline{z})} \left(\left(U_{\mathsf{B}} O_{\mathsf{B}}^{F_{\underline{v},\underline{z}}} \right)^{q_{\mathsf{B}}} \otimes V_{\mathsf{C}}^{F_{\underline{v},\underline{z}}} + V_{\mathsf{B}}^{F_{\underline{v},\underline{z}}} \otimes W_{\mathsf{C}}^{F_{\underline{v},\underline{z}}} \right) |\psi\rangle \|^2 \leq 9p_{\max}\,.$$

Let

$$\alpha = \mathbb{E}_{\underline{z} \leftarrow \mathcal{Z}} \|\Pi^{g(\underline{z})} \left(\left(U_{\mathsf{B}} O_{\mathsf{B}}^{F_{\underline{v},\underline{z}}} \right)^{q_{\mathsf{B}}} \otimes V_{\mathsf{C}}^{F_{\underline{v},\underline{z}}} \right) |\psi\rangle \|^2,$$

and

$$\beta = \mathbb{E}_{\underline{z}\leftarrow\mathcal{Z}} \|\Pi^{g(\underline{z})} \left(V_{\mathsf{B}}^{F_{\underline{v}},\underline{z}} \otimes W_{\mathsf{C}}^{F_{\underline{v}},\underline{z}}\right) |\psi\rangle \|^2 .$$

Applying triangle inequalities, and using that the $\Pi^{g(\underline{z})}$ are orthogonal projectors, we get

$$\mathbb{E}_{\underline{z}\leftarrow\mathcal{Z}} \|\Pi^{g(\underline{z})} \left(\left(U_{\mathsf{B}} O_{\mathsf{B}}^{F_{\underline{v}},\underline{z}}\right)^{q_{\mathsf{B}}} \otimes V_{\mathsf{C}}^{F_{\underline{v}},\underline{z}} + V_{\mathsf{B}}^{F_{\underline{v}},\underline{z}} \otimes W_{C}^{F_{\underline{v}},\underline{z}}\right) |\psi\rangle \|^2 \leq \alpha + \beta + 2\sqrt{\alpha\beta} \tag{26}$$

Now, notice that $V_{\mathsf{B}}^{F_{\underline{v}},\underline{z}}$ and $V_{\mathsf{C}}^{F_{\underline{v}},\underline{z}}$ do not depend on $\underline{z}$, since they always project on the support of $\mathbb{1} - P_{\underline{v}}$. Using standard properties of the operator norm, we get

$$\begin{aligned}
\alpha &= \mathbb{E}_{\underline{z}\leftarrow\mathcal{Z}} \|\Pi^{g(\underline{z})} \left(\left(U_{\mathsf{B}} O_{\mathsf{B}}^{F_{\underline{v}},\underline{z}}\right)^{q_{\mathsf{B}}} \otimes V_{\mathsf{C}}^{F_{\underline{v}},\underline{z}}\right) |\psi\rangle \|^2 && (27)\\
&\leq \mathbb{E}_{\underline{z}\leftarrow\mathcal{Z}} \|(\mathbb{1}_{\mathsf{B}} \otimes \Pi_{\mathsf{C}}^{g(\underline{z})}) \left(\mathbb{1}_{\mathsf{B}} \otimes V_{\mathsf{C}}^{F_{\underline{v}},\underline{z}}\right) |\psi\rangle \|^2 && (28)\\
&\leq \langle\psi| \mathbb{1}_{\mathsf{B}} \otimes \left(\left(V_{\mathsf{C}}^{F_{\underline{v}},\underline{z}}\right)^{\dagger} \left(\mathbb{E}_{z\leftarrow\mathcal{Z}} \Pi_{C}^{g(\underline{z})}\right) V_{\mathsf{C}}^{F_{\underline{v}},\underline{z}}\right) |\psi\rangle && (29)\\
&\leq p_{\max} , && (30)
\end{aligned}$$

where the last line follows because $\mathbb{E}_{\underline{z}\leftarrow\mathcal{Z}} \Pi_{\mathsf{C}}^{g(\underline{z})} \leq p_{\max} \mathbb{1}$.

Similarly, one obtains $\beta \leq 4p_{\max}$, where the factor of 4 is due to the fact that $W_{\mathsf{C}}^{F_{\underline{v}},\underline{z}}$ is not unitary in general, and so we upper bound it by using

$$\|W_{\mathsf{C}}^{F_{\underline{v}},\underline{z}}\|_\infty \leq \| \left(U_{\mathsf{C}} O_{\mathsf{C}}^{F_{\underline{v}},\underline{z}}\right)^{q_{\mathsf{C}}} \|_\infty + \|V_{\mathsf{C}}^{F_{\underline{v}},\underline{z}}\|_\infty \leq 2 .$$

It follows that $\alpha + \beta + 2\sqrt{\alpha\beta} \leq 9p_{\max}$, as desired. □

We specialize Lemma 19 to our exact case in the following corollary. Before stating the corollary, let's fix some notation. For functions $H_1, \ldots, H_r$, we define a "combined" oracle unitary $O^{H_1,\ldots,H_r}$, acting on a "control" register, a "query" register, and an auxiliary register, as follows:

$$O^{H_1,\ldots,H_r} |i\rangle |x\rangle |0\rangle = |i\rangle O^{H_i}(|x\rangle |0\rangle) .$$

Corollary 3. *Let $r \in \mathbb{N}$. Let $v \in \{0,1\}^{m(\lambda)}$ and $z, z' \in \{0,1\}^{\lambda}$. For all unitaries $U_{\mathsf{B}}, U_{\mathsf{C}}$, for all bipartite states $|\psi\rangle$, the following holds:*

$$\mathbb{E}_H \mathbb{E}_{w\leftarrow\{0,1\}^r} \|\Pi^w \left(U_{\mathsf{B}} O_{\mathsf{B}}^{H_{v,z}^{w_1},\ldots,H_{v,z}^{w_r}}\right)^{q_{\mathsf{B}}} \otimes \left(U_{\mathsf{C}} O_{\mathsf{C}}^{H_{v,z}^{w_1},\ldots,H_{v,z}^{w_r}}\right)^{q_{\mathsf{C}}} |\psi\rangle \|^2 \tag{31}$$

$$\leq \frac{9}{2^r} + \mathsf{poly}(q_{\mathsf{B}}, q_{\mathsf{C}})\sqrt{M}, \tag{32}$$

where $\Pi^w = \Pi_{\mathsf{B}}^w \otimes \Pi_{\mathsf{C}}^w$ is a projector, where $H : \{0,1\}^{m(\lambda)} \to \{0,1\}^{\lambda}$ and, for $b \in \{0,1\}$,

$$H_{v,z}^b := \begin{cases} H_{v,z} & \text{if } b = 0 , \\ H_{v,z'} & \text{if } b = 1 , \end{cases}$$

and

$$M = \mathbb{E}_k \mathbb{E}_l \mathbb{E}_H \mathbb{E}_w \| |v\rangle \langle v| \otimes |v\rangle \langle v| \, (U_{\mathsf{B}} O_{\mathsf{B}}^{H_{v,z}^{w_1},\ldots,H_{v,z}^{w_r}})^k \otimes (U_{\mathsf{C}} O_{\mathsf{C}}^{H_{v,z}^{w_1},\ldots,H_{v,z}^{w_r}})^l |\psi\rangle \|^2 .$$

Proof. We apply Lemma 19 with

- $\mathcal{R} = \{0,1\}^r$, $\mathcal{X} = [r] \times \{0,1\}^{m(\lambda)}$, $\mathcal{Y} = \{0,1\}^{\lambda}$.
- $\mathcal{Z} = \{(z_1, \ldots, z_r) : z_i \in \{z, z'\}\}$, and Z the uniform distribution over $\mathcal{Z}$,
- $g : \mathcal{Z} \to \mathcal{R}$ such that $g(z_1, \ldots, z_r) = w$ where $w_i = 0$ if $z_i = z$ and $w_i = 1$ if $z_i = z'$.
- $F : \mathcal{X} \to \mathcal{Y}$ is such that $F(i, x) = (i, H(x))$.
- $\underline{v} = \big((1, v), \ldots, (r, v)\big)$.

□

Lemma 20. *For all query-bounded adversaries $\mathcal{A}$ there exists a negligible function μ such that $\mathcal{A}$ wins game G_r with probability at most $\frac{9}{2^r} + \mu(\lambda)$.*

Proof. Suppose for a contradiction that there existed an adversary $(\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2)$, and a polynomial $p > 0$, such that $(\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2)$ wins game G_r with probability greater than $\frac{9}{2^r} + 1/p(\lambda)$ for infinitely many λ's. We show that this implies an adversary that wins at the following variant of the monogamy game of Section 2.2. Let $\lambda, \lambda' \in \mathbb{N}$. For clarity, we will denote the adversary in this monogamy game as the triple $(\mathcal{P}', \mathcal{F}_1', \mathcal{F}_2')$.

- The challenger samples uniformly $\theta, v \leftarrow \{0,1\}^\lambda$ as in the original game. The challenger picks a uniformly random function $H : \{0,1\}^\lambda \to \{0,1\}^{\lambda'}$ and sends $H(v)$ to $\mathcal{P}'$.
- Additionally, the challenger samples uniformly $w \leftarrow \{0,1\}^r$, and $z \leftarrow \{0,1\}^{\lambda'}$. $\mathcal{P}$, $\mathcal{F}_1$ and $\mathcal{F}_2$ get oracle access to $H_{v,z}^{w_i}$, for $i \in [r]$, where

$$H_{v,z}^b := \begin{cases} H & \text{if } b = 0\,, \\ H_{v,z} & \text{if } b = 1\,. \end{cases}$$

- The rest proceeds as in the original monogamy game.

We claim that as long as $\lambda' < \frac{1}{5}\lambda$, any adversary wins with at most negligible probability in the game. The reason is that as long as $\lambda' < \frac{1}{5}\lambda$, we have $\mathbf{H}_{\min}(V|\mathsf{E}) > \frac{4}{5}\lambda$, where V is the random variable for the string v, and E represents all classical and quantum information that the adversary gets. Thus, Corollary 1 applies.

Next, we are ready to construct an adversary $(\mathcal{P}', \mathcal{F}_1', \mathcal{F}_2')$ for the above monogamy game from an adversary $(\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2)$ winning game G_r with probability $> \frac{9}{2^r} + 1/p(\lambda)$ for some λ. $(\mathcal{P}', \mathcal{F}_1', \mathcal{F}_2')$ is as follows:

- $\mathcal{P}'$ runs $\mathcal{P}$ on the state received from the challenger. The output is a state on two registers: $\mathcal{P}'$ sends the first half to $\mathcal{F}_1'$ and the second half to $\mathcal{F}_2'$.
- Let q_1 and q_2 be the number of oracle queries performed by the $\mathcal{F}_1$ and $\mathcal{F}_2$ algorithms. $\mathcal{F}_1'$ and $\mathcal{F}_2'$ respectively pick uniformly $k \leftarrow [q_1]$ and $l \leftarrow [q_2]$. Then, $\mathcal{F}_1'$ and $\mathcal{F}_2'$ respectively run the $\mathcal{F}_1$ and $\mathcal{F}_2$ algorithms for k and l queries, using oracle access to $H_{v,z}^{w_i}$, for $i \in [r]$. Finally, $\mathcal{F}_1'$ and $\mathcal{F}_2'$ measure the respective oracle registers and return their outcomes to the challenger.

By Corollary 3, the outcome returned by $\mathcal{F}_1'$ and $\mathcal{F}_2'$ is v with inverse-polynomial probability. □

Proof of Lemma 16. Combining lemmas 18 and 20, we deduce that, for any $r \geq 4$, the statement of Lemma 16 must hold for any ϵ^* such that

$$1 - 4r\sqrt{\epsilon^*} > \frac{10}{2^r}\,.$$

Taking $r = 5$, we deduce that the statement of Lemma 16 holds for any $\epsilon^* \leq 10^{-4}$. □

Proof of Lemma 13. Let $(\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2)$ be a strategy that wins with probability p in H_3. We will argue that $(\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2)$ wins with probability at least $p - \frac{2}{3} + \delta^* - \mathsf{negl}(\lambda)$ in H_4, where $\delta^* > 0$. When considering the winning probability of a strategy, we can divide the analysis into three cases bases on what type of inputs the challenger is challenging $\mathcal{F}_1$ and $\mathcal{F}_2$ on. Let $\theta \in \{0,1\}^{m(\lambda)}$ denote the basis choice used to encode a string $v \in \{0,1\}^{m(\lambda)}$. Then, in the input challenge phase, $\mathcal{F}_1$ and $\mathcal{F}_2$ receive basis choices θ_1 and θ_2, respectively, according to the following distribution:

(i) with probability $\frac{1}{3}$, the challenger picks uniformly random $\theta', \theta'' \in \{0,1\}^{m(\lambda)}$, sends θ' to $\mathcal{F}_1$ and θ'' to $\mathcal{F}_2$.

(ii) with probability $\frac{1}{3}$, the challenger picks a uniformly random $\theta' \in \{0,1\}^{m(\lambda)}$, sends θ' to $\mathcal{F}_1$ and the correct basis choice θ to $\mathcal{F}_2$.

(iii) with probability $\frac{1}{3}$, the challenger picks a uniformly random $\theta' \in \{0,1\}^\lambda$, sends the correct basis choice θ to $\mathcal{F}_1$ and θ' to $\mathcal{F}_2$.

Conditioned on case (i), we argue that the winning probabilities of $\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2$ in H_3 and in H_4 are negligibly close, i.e. there exists a negligible function ν such that:

$$\Big| \Pr\Big[(\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2) \text{ win in } H_4 \,|\, \theta_1 \text{ is random and } \theta_2 \text{ is random}\Big] \\ - \Pr\Big[(\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2) \text{ win in } H_3 \,|\, \theta_1 \text{ is random and } \theta_2 \text{ is random}\Big]\Big| \leq \nu(\lambda). \tag{33}$$

This follows from a similar argument made earlier. By an application of the one-way-to-hiding lemma, one can bound the advantage in distinguishing H_3 from H_4 by the probability of querying at the encoded point v. An adversary that queries with non-negligible probability at v straightforwardly yields an adversary that wins with non-negligible probability at the monogamy of entanglement game of Lemma 4. This implies that the winning probability of $(\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2)$ in H_4 is at least $p - \frac{2}{3} - \nu(\lambda)$.

What we will argue next is that in fact the lower bound is $p - \frac{2}{3} + \delta^* - \mathsf{negl}(\lambda)$, for some $\delta^* > 0$. We appeal to Lemma 16, by which there exists $\epsilon^* > 0$ (where we can take $\epsilon^* = 10^{-4}$), and a negligible function μ, such that for all λ, there exists an $i \in \{1, 2\}$ such that $\mathcal{F}_i$ satisfies

$$\Big| \Pr[\mathcal{F}_i \text{ returns 1 in } H_4 \,|\, \theta_i \text{ is correct }] \\ - \Pr[\mathcal{F}_i \text{ returns 1 in } H_3 \,|\, \theta_i \text{ is correct}]\Big| < 1 - \epsilon^* + \mu(\lambda)\,. \tag{34}$$

Fix a λ, and assume (34) holds for $i = 1$ for this particular λ (the other case being analogous).

We consider the two cases:

(a)
$$\Pr[(\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2) \text{ win in } H_3 \,|\, \theta_1 \text{ is correct and } \theta_2 \text{ is random}] > 1 - \frac{\epsilon^*}{3}\,. \tag{35}$$

(b)
$$\Pr[(\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2) \text{ win in } H_3 \,|\, \theta_1 \text{ is a correct and } \theta_2 \text{ is random}] \leq 1 - \frac{\epsilon^*}{3}\,. \tag{36}$$

We start by assuming (a). Then, Equation (34) implies that

$$\Pr[\mathcal{F}_1 \text{ returns 1 in } H_4 \,|\, \theta_1 \text{ is correct}] > \frac{2\epsilon^*}{3} - \mu(\lambda)\,, \tag{37}$$

where we are using the fact that the correlation generated by $\mathcal{F}_1$ and $\mathcal{F}_2$ is non-signalling to remove the condition of θ_2 being random.

By the same argument used for case (i), notice that

$$|\Pr[\mathcal{F}_2 \text{ returns 0 in } H_3 \,|\, \theta_2 \text{ is random }] - \Pr[\mathcal{F}_2 \text{ returns 0 in } H_4 \,|\, \theta_2 \text{ is random }]| \leq \mathsf{negl}(\lambda)\,.$$

Then, Equation (35) implies

$$\Pr[\mathcal{F}_2 \text{ returns 0 in } H_4 \,|\, \theta_2 \text{ is random}] > 1 - \frac{\epsilon^*}{3} - \mathsf{negl}(\lambda)\,, \tag{38}$$

where we again use the fact that the correlation generated by $\mathcal{F}_1$ and $\mathcal{F}_2$ is non-signalling.

Combining (37) and (38), and using a union bound implies that

$$\Pr[(\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2) \text{ win in } H_4 \,|\, \theta_1 \text{ is correct and } \theta_2 \text{ is random}] > \frac{\epsilon^*}{3} - \mathsf{negl}(\lambda)\,. \tag{39}$$

Thus, Equations (35) and (39) trivially imply that

$$\Pr[(\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2) \text{ win in } H_4 \,|\, \theta_1 \text{ is correct and } \theta_2 \text{ is random}] \\ > \Pr[(\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2) \text{ win in } H_3 \,|\, \theta_1 \text{ is correct and } \theta_2 \text{ is random}] - \left(1 - \frac{\epsilon^*}{3}\right) - \mathsf{negl}(\lambda)\,. \tag{40}$$

Now, equations (33) and (40) imply:

$$
\begin{aligned}
&\Pr[(\mathcal{P},\mathcal{F}_1,\mathcal{F}_2) \text{ win in } H_4]\\
&= \frac{1}{3}\Pr\left[(\mathcal{P},\mathcal{F}_1,\mathcal{F}_2) \text{ win in } H_4 \,|\, \theta_1 \text{ is random and } \theta_2 \text{ is random}\right]\\
&+ \frac{1}{3}\Pr\left[(\mathcal{P},\mathcal{F}_1,\mathcal{F}_2) \text{ win in } H_4 \,|\, \theta_1 \text{ is correct and } \theta_2 \text{ is random}\right]\\
&+ \frac{1}{3}\Pr\left[(\mathcal{P},\mathcal{F}_1,\mathcal{F}_2) \text{ win in } H_4 \,|\, \theta_1 \text{ is random and } \theta_2 \text{ is correct}\right]\\
&> \frac{1}{3}\Pr\left[(\mathcal{P},\mathcal{F}_1,\mathcal{F}_2) \text{ win in } H_3 \,|\, \theta_1 \text{ is random and } \theta_2 \text{ is random}\right]\\
&+ \frac{1}{3}\left(\Pr\left[(\mathcal{P},\mathcal{F}_1,\mathcal{F}_2) \text{ win in } H_3 \,|\, \theta_1 \text{ is correct and } \theta_2 \text{ is a random}\right] - 1 + \frac{\epsilon^*}{3}\right)\\
&+ \frac{1}{3}\left(\Pr\left[(\mathcal{P},\mathcal{F}_1,\mathcal{F}_2) \text{ win in } H_3 \,|\, \theta_1 \text{ is random and } \theta_2 \text{ is correct}\right] - 1\right) - \mathsf{negl}(\lambda)\\
&= p - \frac{2}{3} + \frac{\epsilon^*}{9} - \mathsf{negl}(\lambda)\,. \qquad (41)
\end{aligned}
$$

Now, instead, we assume (b), i.e.

$$
\Pr\left[(\mathcal{P},\mathcal{F}_1,\mathcal{F}_2) \text{ win in } H_3 \,|\, \theta_1 \text{ is correct and } \theta_2 \text{ is random}\right] \leq 1 - \frac{\epsilon^*}{3}\,. \qquad (42)
$$

Then, we immediately have

$$
\begin{aligned}
&\Pr\left[(\mathcal{P},\mathcal{F}_1,\mathcal{F}_2) \text{ win in } H_4 \,|\, \theta_1 \text{ is correct and } \theta_2 \text{ is random}\right]\\
&> \Pr\left[(\mathcal{P},\mathcal{F}_1,\mathcal{F}_2) \text{ win in } H_3 \,|\, \theta_1 \text{ is correct and } \theta_2 \text{ is random}\right] - \left(1 - \frac{\epsilon^*}{3}\right)\,. \qquad (43)
\end{aligned}
$$

A similar calculation to Equation (41) gives

$$
\Pr\left[(\mathcal{P},\mathcal{F}_1,\mathcal{F}_2) \text{ win in } H_4\right] > p - \frac{2}{3} + \frac{\epsilon^*}{9} - \mathsf{negl}(\lambda)\,. \qquad (44)
$$

This gives the desired bound with $\delta^* = \frac{\epsilon^*}{9}$.

□

This concludes the proof of Lemma 13, and hence of Theorem 2.

4.2 Proof of quantum virtual black-box obfuscation

In this section, we show that our quantum copy-protection scheme for point functions is also a quantum virtual black-box (VBB) obfuscator [AF16]. In particular, we will show that the algorithm CP.Protect from Construction 1 satisfies a notion of obfuscation called *distributional indistinguishability*, which for evasive classes of circuits is equivalent to VBB obfuscation [WZ17] (it is straightforward to see that distributional indistinguishability implies VBB - the reverse implication requires slightly more work). CP.Protect is already functionality preserving, this follows from the definition of a copy-protection scheme. All that is left to show is security. In what follows, we assume that a program has an associated set of parameters $P.\mathsf{params}$ (input size, output size, circuit size) which we are not required to hide.

Definition 5 (Distributional indistinguishability)**.** *An obfuscator* Obf *for the class of distributions $\mathcal{D}$ over programs $\mathcal{C}$, satisfies distributional indistinguishability if there exists a* QPT *simulator* Sim*, such that for every distribution ensemble $D = \{D_\lambda\} \in \mathcal{D}$, we have*

$$
\mathbb{E}_{P\leftarrow D_\lambda}\mathsf{Obf}(1^\lambda, P) \approx_c \mathsf{Sim}(1^\lambda, P.\mathsf{params})\,. \qquad (45)
$$

(where the notation "$\approx_c$" was introduced in Definition 1.) We remark that the notion of indistinguishability of ensembles of quantum states (Definition 1) already accounts for auxiliary quantum information in the two ensembles.

Distributional indistinguishability relative to any oracle is analogous to Definition 5, except that the algorithms Obf and Sim are quantum oracle algorithms, and the notation "$\approx_c$" refers to Definition 2.

Theorem 3. *The* QPT *algorithm* $\mathsf{CP.Protect}$ *from Construction 1 satisfies distributional indistinguishability in the QROM for the class of distributions* $\mathcal{D}_{\mathsf{PF\text{-}UNP}}$.

Proof. We define the following simulator Sim:

- Sim takes as input 1^λ, an auxiliary state ν, and outputs the state: $\frac{\mathbb{1}}{2^{m(\lambda)+\lambda}} \otimes \nu$, where the first factor is the maximally mixed state on $m(\lambda)+\lambda$ qubits.

Let

$$\rho_{\lambda,G,H} := \mathbb{E}_{P_y \leftarrow D_\lambda} \mathsf{CP.Protect}^{G,H}(1^\lambda, P_y) = \mathbb{E}_{P_y \leftarrow D_\lambda} \mathbb{E}_v \left|v^{G(y)}\right\rangle \left\langle v^{G(y)}\right| \otimes \left|H(v)\right\rangle \left\langle H(v)\right| ,$$

and

$$\sigma_{\lambda,H} := \mathbb{E}_\theta \mathbb{E}_v \left|v^\theta\right\rangle \left\langle v^\theta\right| \otimes \left|H(v)\right\rangle \left\langle H(v)\right| .$$

By an argument analogous to that of the proof of Lemma 11, it holds that, for any computationally bounded oracle adversary $\mathcal{A}$, and any auxiliary state ν_λ,

$$\mathbb{E}_{G,H} \left| \Pr[\mathcal{A}^{G,H}(\rho_{\lambda,G,H} \otimes \nu_\lambda) = 1] - \Pr[\mathcal{A}^{G,H}(\sigma_{\lambda,H} \otimes \nu_\lambda) = 1] \right| = \mathsf{negl}(\lambda) .$$

Next, we replace the state $\sigma_{\lambda,H}$ with the state $\sigma'_\lambda := \mathbb{E}_\theta \mathbb{E}_v \mathbb{E}_z \left|v^\theta\right\rangle \left\langle v^\theta\right| \otimes \left|z\right\rangle \left\langle z\right|$. We argue that for any query bounded adversary $\mathcal{A}$, the following holds:

$$\mathbb{E}_H \left| \Pr[\mathcal{A}^H(\sigma_{\lambda,H} \otimes \nu_\lambda) = 1] - \Pr[\mathcal{A}^H(\sigma'_\lambda \otimes \nu_\lambda) = 1] \right| = \mathsf{negl}(\lambda) . \tag{46}$$

Let $\mathcal{A}$ be a distinguisher making q queries. Without loss of generality, let $\mathcal{A}$ be specified by the unitary $(UO^H)^q$, for some unitary U.

We apply one-way-to-hiding (Lemma 7) to deduce that the LHS of (46) is negligible if the quantity

$$\mathbb{E}_H \mathbb{E}_v \mathbb{E}_\theta \mathbb{E}_{z \leftarrow \{0,1\}^m} \mathbb{E}_k \mathrm{Tr}\Big[\left|v\right\rangle \left\langle v\right| (UO^H)^k \left(\left|v^\theta\right\rangle \left\langle v^\theta\right| \otimes \left|z\right\rangle \left\langle z\right| \otimes \nu_\lambda \right) (UO^H)^k \Big]$$

is negligible.

Suppose for a contradiction the latter is not negligible. Then, we can construct an adversary that wins at the monogamy of entanglement game of Lemma 4. The reduction is straightforward: the adversary for the monogamy of entanglement game prepares the auxiliary state ν_λ, and runs $\mathcal{A}$ (by simulating an oracle) to extract v. Then sends v to both $\mathcal{B}$ and $\mathcal{C}$ (using the notation for the monogamy game of Lemma 4).

The conclusion of the theorem follows by observing that σ'_λ is the maximally mixed state. □

5 Extension to compute-and-compare programs

In this section, we show that a quantum copy-protection scheme for point functions, which is secure with respect to the appropriate program and challenge ensembles, implies a quantum copy-protection scheme for compute-and-compare programs with the same level of security.

The idea is simple: to copy-protect the compute-and-compare program $\mathsf{CC}[f, y]$, we copy-protect the point function P_y, and give f in the clear. By copy-protecting P_y we are copy-protecting the portion of the compute-and-compare program which checks equality with y. The intuition is that this is sufficient to make the functionality unclonable because its output is not already determined by f. More generally, one might suspect that, for copy-protecting a function $F = f_1 \circ f_2 ... \circ f_\ell$, it should be sufficient to copy-protect any of the functions f_i that is sufficiently non-constant *within its context*.

Let $(\mathsf{CP\text{-}PF.Protect}, \mathsf{CP\text{-}PF.Eval})$ be a copy-protection scheme for point functions.

Construction 2 (Copy-protection scheme for compute-and-compare programs)**.** *The copy-protection scheme for compute-and-compare programs* $(\mathsf{CP\text{-}CC.Protect}, \mathsf{CP\text{-}CC.Eval})$ *is defined as follows:*

- $\mathsf{CP\text{-}CC.Protect}(1^\lambda, (f, y))$*: Takes as input a security parameter* λ *and a compute-and-compare program* $\mathsf{CC}[f, y]$*, specified succinctly by* f *and* y*. Then,*
  - *Let* $\rho = \mathsf{CP\text{-}PF.Protect}(\lambda, y))$*.*
  - *Output* (f, ρ)*.*
- $\mathsf{CP\text{-}CC.Eval}(1^\lambda, (f, \rho); x)$*: Takes as input a security parameter* λ*, an alleged copy-protected program* (f, ρ)*, and a string* $x \in \{0,1\}^n$ *(where* n *is the size of the inputs to* f*). Then,*
  - *Compute* $y' = f(x)$*.*
  - *Let* $b \leftarrow \mathsf{CP\text{-}PF.Eval}(\rho; y')$*. Output* b*.*

Recall the definition of the class of distributions over point functions $\mathcal{D}_{\mathsf{PF\text{-}UNP}}$ from Section 4. We define a related class of distributions over compute-and-compare programs.

- $\mathcal{D}_{\mathsf{CC\text{-}UNP}}$. We refer to this class as the class of *unpredictable compute-and-compare programs*. This consists of ensembles $D = \{D_\lambda\}$ where D_λ is a distribution over compute-and-compare programs such that $\mathsf{CC}[f, y] \leftarrow D_\lambda$ satisfies $\mathbf{H}_{\min}(y|f) \geq \lambda^\epsilon$ for some $\epsilon > 0$, and where the input length of f is λ and the output length is bounded by some polynomial $t(\lambda)$.

We also define the following class of distributions over input challenges:

- $\mathcal{D}_{\mathsf{CC\text{-}Chall}}$. An ensemble $D = \{D_{f,y}\}$, where each $D_{f,y}$ is a distribution over pairs of elements in the domain of f, belongs to the class $\mathcal{D}_{\mathsf{CC\text{-}Chall}}$ if there exists an efficiently sampleable family $\{X_\lambda\}$ of distributions over $\{0,1\}^\lambda$ with $\mathbf{H}_{\min}(X_\lambda) \geq \lambda^\epsilon$, for some $\epsilon > 0$, and an efficiently sampleable family $\{Z_{f,y}\}$, where $Z_{f,y}$ is a distribution over the set $f^{-1}(y)$, such that $D_{f,y}$ is the following distribution (where λ is the size of inputs to f):
  - With probability 1/3, sample $z \leftarrow Z_{f,y}$ and $x \leftarrow X_\lambda$, and output (x, z).
  - With probability 1/3, sample $z \leftarrow Z_{f,y}$ and $x \leftarrow X_\lambda$, and output (z, x).
  - With probability 1/3, sample $x, x' \leftarrow X_\lambda$, and output (x, x').

  We say the ensemble D is *specified* by the families $\{X_\lambda\}$ and $\{Z_{f,y}\}$.

Just like in the point function case, we also define two classes of distributions over pairs of programs and challenges.

- $\mathcal{D}_{\mathsf{CC\text{-}pairs\text{-}stat}}$. This consists of pairs of ensembles $\left(D = \{D_\lambda\}, D' = \{D'_{f,y}\}\right)$ where $D \in \mathcal{D}_{\mathsf{CC\text{-}UNP}}$ and $D' \in \mathcal{D}_{\mathsf{CC\text{-}Chall}}$ satisfying the following. Let D' be parametrized by the families $\{X_\lambda\}$ and $\{Z_{f,y}\}$ (following the notation introduced above), and denote by $\mathsf{MarkedInput}\,(D_\lambda, \{Z_{f,y}\})$ the distribution over $\{0,1\}^\lambda$ induced by D_λ and $\{Z_{f,y}\}$, i.e.:
  - Sample $(f, y) \leftarrow D_\lambda$, then $z \leftarrow Z_{f,y}$.

  For any fixed f_* with domain $\{0,1\}^\lambda$ such that (f_*, y_*) is in the support of D_λ for some y_*, denote by $\mathsf{MarkedInput}(D_\lambda, \{Z_{f,y}\})|_{f_*}$, the distribution $\mathsf{MarkedInput}(D_\lambda, \{Z_{f,y}\})$ conditioned on D_λ sampling f_*. Then, we require that, for any sequence $\{f_*^{(\lambda)}\}$ (where, for all λ, $(f_*^{(\lambda)}, y_*)$ is in the support of D_λ for some y_*), the families $\{X_\lambda\}$ and $\{\mathsf{MarkedInput}(D_\lambda, \{Z_{f,y}\})|_{f_*^{(\lambda)}}\}$ are statistically indistinguishable.
- $\mathcal{D}_{\mathsf{CC\text{-}pairs\text{-}comp}}$. This is defined in the same way as $\mathcal{D}_{\mathsf{CC\text{-}pairs\text{-}stat}}$, except that we only require $\{X_\lambda\}$ and $\{\mathsf{MarkedInput}(D_\lambda, \{Z_{f,y}\})|_{f_*^{(\lambda)}}\}$ to be *computationally* indistinguishable.

Theorem 4. *Let* $(\mathsf{CP\text{-}PF.Protect}, \mathsf{CP\text{-}PF.Eval})$ *be a copy-protection scheme for point functions that is* δ*-secure with respect to all pairs* $(D, D') \in \mathcal{D}_{\mathsf{PF\text{-}pairs\text{-}stat}}$ *(*$\in \mathcal{D}_{\mathsf{PF\text{-}pairs\text{-}comp}}$*). Then, the scheme of Construction 2, instantiated with* $(\mathsf{CP\text{-}PF.Protect}, \mathsf{CP\text{-}PF.Eval})$*, is a* δ*-secure copy-protection scheme for compute-and-compare programs with respect to all pairs* $(D, D') \in \mathcal{D}_{\mathsf{CC\text{-}pairs\text{-}stat}}$ *(*$\in \mathcal{D}_{\mathsf{CC\text{-}pairs\text{-}comp}}$*). The same conclusion holds relative to any oracle, i.e. when all algorithms have access to the same oracle, with respect to query-bounded (computationally bounded) adversaries.*

Proof. We prove the theorem for the case of $(\{D_\lambda\}, \{D_{f,y}\}) \in \mathcal{D}_{\mathsf{CC\text{-}pairs\text{-}stat}}$ (the case of $(\{D_\lambda\}, \{D_{f,y}\}) \in \mathcal{D}_{\mathsf{CC\text{-}pairs\text{-}comp}}$ being virtually identical). Let $t(\lambda)$ be the length of strings in the range of f's sampled from D_λ. Let the ensemble $\{D_{f,y}\}$ be specified by $\{Z_{f,y}\}$ and $\{X_\lambda\}$ (using the notation introduced above for ensembles in $\mathcal{D}_{\mathsf{CC\text{-}Chall}}$).

Let $\mathcal{A} = (\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2)$ be an adversary for the compute-and-compare copy-protection scheme of Construction 2 with respect to ensembles $\{D_\lambda\}$ and $\{D_{f,y}\}$. Suppose $\mathcal{A}$ wins with probability $p(\lambda) > 0$. It follows that for each λ there exists $f_*^{(\lambda)}$ such that $(f_*^{(\lambda)}, y)$ is in the support of D_λ for some y, and the probability that $\mathcal{A}$ wins is at least $p(\lambda)$, conditioned on $f^{(\lambda)}$ being sampled.

We will construct an adversary $\mathcal{A}'$ that wins with probability $p(\lambda) - \mathsf{negl}(\lambda)$ in the point function security game with respect to $\{D'_{t(\lambda)}\}$ and $\{D'_y\}$, defined as follows:

- $D'_{t(\lambda)}$: sample $x \leftarrow X_\lambda$ and output the point function $P_{f_*^{(\lambda)}(x)}$.
- D'_y: sample $(x, x') \leftarrow D_{f_*^{(\lambda)}, y}$ and output $(f_*^{(\lambda)}(x), f_*^{(\lambda)}(x'))$.

The adversary $\mathcal{A}' = (\mathcal{P}', \mathcal{F}'_1, \mathcal{F}'_2)$ then acts as follows:

- $\mathcal{P}'$ receives as input a state ρ. Then, $\mathcal{P}'$ provides $(f_*^{(\lambda)}, \rho)$ as input to $\mathcal{P}$. Let A_1 and A_2 be the registers returned by $\mathcal{P}$. $\mathcal{P}'$ forwards A_1 and A_2 to $\mathcal{F}'_1$ and $\mathcal{F}'_2$ respectively.
- Upon receiving a challenge x_i, $\mathcal{F}'_i$ samples $x'_i \leftarrow Z_{f,x_i}$. $\mathcal{F}'_i$ then runs $\mathcal{F}_i$ on input x'_i and the register A_i. Let b_i be the output returned by $\mathcal{F}_i$. $\mathcal{F}'_i$ returns b_i to the challenger.

It is straightforward to check that the game "simulated" by $(\mathcal{P}', \mathcal{F}'_1, \mathcal{F}'_2)$ for $(\mathcal{P}, \mathcal{F}_1, \mathcal{F}_2)$ is statistically indistinguishable from a security game with respect to $\{D_\lambda\}$ and $\{D_{f,y}\}$, conditioned on $f_*^{(\lambda)}$. Thus, we deduce, by hypothesis, that $\mathcal{F}_1$ and $\mathcal{F}_2$ both return the correct bits with probability at least $p(\lambda) - \mathsf{negl}(\lambda)$, and thus $(\mathcal{P}', \mathcal{F}'_1, \mathcal{F}'_2)$ wins with probability at least $p(\lambda) - \mathsf{negl}(\lambda)$. Crucially, note that $\left(\{D'_{t(\lambda)}\}, \{D'_y\}\right) \in \mathcal{D}_{\mathsf{PF\text{-}pairs\text{-}stat}}$. It follows that if the point function copy-protection scheme is δ-secure, then the compute-and-compare copy-protection scheme must also be δ-secure. It is easy to check that all steps of the proof also hold relative to an oracle. □

Corollary 4. *Let $\delta^* > 0$ be the constant from Theorem 2. There exists a δ^*-secure copy-protection scheme for compute-and-compare programs with respect to ensembles $(D, D') \in \mathcal{D}_{\mathsf{PF\text{-}pairs\text{-}stat}}$ ($\in \mathcal{D}_{\mathsf{PF\text{-}pairs\text{-}comp}}$) against query-bounded adversaries (computationally bounded adversaries, assuming the existence of quantum-secure one-way functions).*

Proof. Combining Theorem 2 with Theorem 4 immediately gives this corollary. □

6 Secure software leasing

Recall that the level of security achieved by our copy-protection scheme in Theorem 2 is far from ideal. In this section, we show that our construction (augmented with a verification routine) satisfies a weaker notion of copy-protection called "secure software leasing" (SSL), introduced in [ALP20], but with a standard level of security, i.e. the adversarial success probability is negligible in the security parameter. We formalize the notion of SSL in the following section.

Definition 6 (Secure software leasing)**.** *Let $\mathcal{C}$ be a family of classical circuits with a single bit output. A secure software leasing* (SSL) *scheme* (SSL.Gen, SSL.Lease, SSL.Eval, SSL.Verify) *consists of the following* QPT *algorithms:*

- $\mathsf{SSL.Gen}(1^\lambda)$ *takes as input the security parameter λ and outputs a secret key* sk.
- $\mathsf{SSL.Lease}(\mathsf{sk}, C)$ *takes as input a secret key* sk *and a* $\mathsf{poly}(\lambda)$*-sized circuit $C \in \mathcal{C}$ with input size n, and outputs a quantum state ρ_C.*
- $\mathsf{SSL.Eval}(x, \rho_C)$ *takes a string x as input to C together with a state ρ_C, and outputs a bit and a post-evaluation state $\tilde{\rho}_C$.*
- $\mathsf{SSL.Verify}(\mathsf{sk}, C, \sigma)$ *takes as input the secret key* sk*, the circuit $C \in \mathcal{C}$ and a state σ, and outputs 1, if σ is a valid lease state for C, and 0 otherwise.*

Above, we have defined the algorithms SSL.Eval and SSL.Verify with quantum states as inputs. To formalize security, however, we need to provide as inputs to them parts of quantum states supported on several registers.

The following definitions introduce further properties of secure software leasing schemes.

Definition 7 (Correctness)**.** *An* SSL *scheme* (SSL.Gen, SSL.Lease, SSL.Eval, SSL.Verify) *is called correct for a family of circuits $\mathcal{C}$ if, for all $C \in \mathcal{C}$, there exists a negligible function* negl *satisfying,*

- *Correctness of evaluation:*

$$\forall x \in \{0,1\}^n : \quad \Pr\left[\mathsf{SSL.Eval}(x, \rho_C) = C(x)\right] \geq 1 - \mathsf{negl}(\lambda).$$

- *Correctness of verification:*

$$\Pr\left[\mathsf{SSL.Verify}(\mathsf{sk}, C, \rho_C) = 1\right] \geq 1 - \mathsf{negl}(\lambda).$$

Security is defined in terms of a security game between a lessor and an adversary (the lessee). Informally, any secure software leasing (SSL) scheme should satisfy the following key property. After receiving a leased copy of C denoted by ρ_C (generated using SSL.Lease), the adversary should not be able to produce a quantum state σ on registers R_1 and R_2 such that:

- SSL.Verify deems the contents of register R_1 of $\sigma_{\mathsf{R}_1\mathsf{R}_2}$ to be valid, once it is returned.
- The adversary can still compute C (on inputs chosen by the lessor) from the post-measurement state in register R_2 given by $\sigma^*_{\mathsf{R}_2} \propto \mathrm{Tr}_{\mathsf{R}_1}\left[\Pi_1\left[\left(\mathsf{SSL.Verify}(\cdot)_{\mathsf{R}_1} \otimes \mathbb{1}_{\mathsf{R}_2}\right)\sigma_{\mathsf{R}_1\mathsf{R}_2}\right]\right]$.

Let $\mathcal{A}$ denote the adversary. As in the case of copy protection, security is defined for a security parameter λ, a distribution D_λ over circuits C from $\mathcal{C}$, and for a family of distributions $\{D_C\}_{C\in\mathcal{C}}$ over inputs to C. We formalize the security in terms of the following game:

- The lessor samples a $\mathsf{poly}(\lambda)$-sized circuit $C \leftarrow D_\lambda$ and runs $\mathsf{SSL.Gen}(1^\lambda)$, followed by $\mathsf{SSL.Lease}(\mathsf{sk}, C)$ that takes as input the secret key sk and the circuit $C \in \mathcal{C}$, and outputs a quantum state ρ_C. The lessor then sends ρ_C (together with the circuit SSL.Eval for ρ_C) to the adversary $\mathcal{A}$.
- $\mathcal{A}$ outputs a (possibly entangled) state σ on two registers R_1 and R_2, and then sends the first register R_1 to the lessor.
- For verification, the lessor runs SSL.Verify on input the secret key sk, the circuit $C \in \mathcal{C}$ and the register R_1 of the state $\sigma_{\mathsf{R}_1\mathsf{R}_2}$. If SSL.Verify accepts, the lessor outputs $\mathsf{ok} = 1$ and lets the game continue, otherwise, the lessor outputs $\mathsf{ok} = 0$ and $\mathcal{A}$ loses.
- The lessor samples $x \leftarrow \mathcal{D}_C$, and sends x to the adversary.
- $\mathcal{A}$ responds with a bit b. If $b = C(x)$, the lessor outputs 1. Otherwise, the lessor outputs 0.

Further, we let $\mathsf{SSLGame}(\lambda, \mathcal{A}, D_\lambda, \{D_C\})$ denote a Boolean variable that is equal to 1, if the above security game is won, and 0 otherwise.

As in the case of full copy protection, we define the trivial winning probability, which in the case of SSL is just the straightforward guessing probability for the answer to the challenge,

$$p^{\mathrm{triv},\mathsf{SSL}}_{D_\lambda,\{D_C\}_{C\in\mathcal{C}}} = \max_{b\in\{0,1\}} \mathbb{E}_{C\leftarrow D_\lambda} \hat{D}_C(b), \tag{47}$$

where $\hat{D}_C(b)$ is the probability that the correct answer to a challenge sampled from D_C is b.

Definition 8 (Security)**.** *A secure software leasing* (SSL) *scheme for a family of classical circuits $\mathcal{C} = \{C_\lambda\}_{\lambda\in\mathbb{N}}$ is said to be δ-secure with respect to the ensemble $D = \{D_\lambda\}_{\lambda\in\mathbb{N}}$ of distributions over circuits in $\mathcal{C}$, and with respect to the ensemble $\{D_C\}_{C\in\mathcal{C}}$, where D_C is a distribution over challenge inputs to program C, if for any $\lambda \in \mathbb{N}$ and any* QPT *adversary $\mathcal{A}$,*

$$\Pr[\mathsf{SSLGame}(\lambda, \mathcal{A}, D_\lambda, \{D_C\}) = 1] \leq 1 - \delta(\lambda) + \mathsf{negl}(\lambda)\,.$$

If $\delta(\lambda) = 1 - p^{\mathrm{triv},\mathsf{SSL}}_{D_\lambda,\{D_C\}_{C\in\mathcal{C}}}$, we simply call the scheme secure.

We refer to $D = \{D_\lambda\}_{\lambda\in\mathbb{N}}$ as the *program ensemble*, and to $\{D_C\}_{C\in\mathcal{C}}$ as the *input challenge ensemble.*

6.1 Secure software leasing for point functions

We first consider a version of the scheme for point functions in Construction 1, slightly adapted to the syntax of secure software leasing. Then, in Section 6.3, we extend the scheme to the class of compute-and-compare programs. Again, for simplicity, we hand the marked input $y \in \{0,1\}^n$ to the leasing algorithm as an input, rather than a circuit for the point function P_y itself. We also omit the procedure $\mathsf{SSL.Gen}$ as we do not require it in our construction.

Construction 3 (SSL scheme for point functions)**.** *Let λ be the security parameter, and let $H : \{0,1\}^{m(\lambda)} \to \{0,1\}^{\lambda}$ and $G : \{0,1\}^n \to \{0,1\}^{m(\lambda)}$ be hash functions, where $m(\lambda) \geq \lambda$. Consider the following secure software leasing (SSL) scheme ($\mathsf{SSL.Lease}, \mathsf{SSL.Eval}, \mathsf{SSL.Verify}$) for point functions P_y with marked input $y \in \{0,1\}^n$:*

- $\mathsf{SSL.Lease}(1^\lambda, y)$*: Takes as input a security parameter λ and a point function P_y, succinctly specified by the marked input y (of size n)*
  - *Set $\theta = G(y)$.*
  - *Sample $v \leftarrow \{0,1\}^{m(\lambda)}$ uniformly at random and let $z = H(v)$.*
  - *Output $(|v^\theta\rangle, z)$.*
- $\mathsf{SSL.Eval}(1^\lambda, (\rho, z); x)$*: Takes as input a security parameter λ, a program (ρ, z), and a string $x \in \{0,1\}^n$ (the input on which the program is to be evaluated).*
  - *Set $\theta' = G(x)$.*
  - *Apply Hadamards $H^{\theta'} = H^{\theta'_1} \otimes \cdots \otimes H^{\theta'_\lambda}$ to ρ. Append $n+1$ ancillary qubits, all in state $|0\rangle$, and compute the hash function H with input ρ into the first n of them (possibly making use of additional ancillary qubits). Then, coherently measure whether the first n ancilla qubits are in state $|z\rangle$, recording the result in the last ancilla qubit, uncompute the hash function H and undo the Hadamards $H^{\theta'}$. Finally, measure the last ancilla qubit to obtain a bit b as output.*
- $\mathsf{SSL.Verify}(1^\lambda, y, z, \sigma)$*: Apply H^θ to the input state σ, where $\theta = G(y)$, and measure in the standard basis. Output 1 if the result is v such that $H(v) = z$, and 0 otherwise.*

The correctness property of Construction 3 according to Definition 7 is immediate to verify. Before stating our main theorem on the security of Construction 3, we recall the following class of distributions over point functions, which was defined in Section 4.

- $\mathcal{D}_{\mathsf{PF\text{-}UNP}}$. The class of *unpredictable point function distributions* $\mathcal{D}_{\mathsf{PF\text{-}UNP}}$ consists of ensembles $D = \{D_\lambda\}$ where D_λ is a distribution over point functions on $\{0,1\}^\lambda$ such that $P_y \leftarrow D_\lambda$ satisfies $\mathbf{H}_{\min}(y) \geq \lambda^\epsilon$ for some $\epsilon > 0$.

We also define the following class of distributions over input challenges.

- $\mathcal{D}_{\mathsf{PF\text{-}Chall\text{-}SSL}}$. An ensemble $D = \{D_y\}$, where each D_y is a distribution over $\{0,1\}^{|y|}$, belongs to the class $\mathcal{D}_{\mathsf{PF\text{-}Chall\text{-}SSL}}$ if there exists an efficiently sampleable family $\{X_\lambda\}$ of distributions over $\{0,1\}^\lambda$ with $\mathbf{H}_{\min}(X_\lambda) \geq \lambda^\epsilon$, for some $\epsilon > 0$, such that D_y is the following distribution (where $\lambda = |y|$):
  - with probability $1/2$, output y.
  - with probability $1/2$, sample $x \leftarrow X_\lambda$, and output x.

  We say the ensemble D is *specified* by the ensemble X_λ.

We finally define two classes of distributions over pairs of programs and challenges.

- $\mathcal{D}_{\mathsf{PF\text{-}pairs\text{-}stat\text{-}SSL}}$. This consists of pairs of ensembles $\left(D = \{D_\lambda\}, D' = \{D'_y\}\right)$ where $D \in \mathcal{D}_{\mathsf{PF\text{-}UNP}}$ and $D' \in \mathcal{D}_{\mathsf{PF\text{-}Chall\text{-}SSL}}$ satisfying the following. Let D' be parametrized by the family $\{X_\lambda\}$ (following the notation introduced above), and denote by $\mathsf{MarkedInput}(D_\lambda)$ the distribution over marked points in $\{0,1\}^\lambda$ induced by D_λ. Then, the families $\{X_\lambda\}$ and $\{\mathsf{MarkedInput}(D_\lambda)\}$ are statistically indistinguishable.
- $\mathcal{D}_{\mathsf{PF\text{-}pairs\text{-}comp\text{-}SSL}}$. This is defined in the same way as $\mathcal{D}_{\mathsf{PF\text{-}pairs\text{-}stat\text{-}SSL}}$, except that we only require $\{X_\lambda\}$ and $\{\mathsf{MarkedInput}(D_\lambda)\}$ to be *computationally* indistinguishable.

Let us now conclude Section 6 with our main result on the security of Construction 3.

Theorem 5. *The scheme of Construction 3, with $m(\lambda) = \mathsf{poly}(\lambda)$, is a secure software leasing scheme for point functions with respect to any pair of ensembles $(D, D') \in \mathcal{D}_{\mathsf{PF\text{-}pairs\text{-}stat\text{-}SSL}}$ ($\in \mathcal{D}_{\mathsf{PF\text{-}pairs\text{-}comp\text{-}SSL}}$), against query-bounded (computationally bounded) adversaries in the quantum random oracle model.*

Theorem 5 implies that, once a leased copy is successfully returned to the lessor, no adversary can distinguish the marked input of a point function from a random (non-marked) input with probability better than 1/2, except for a negligible advantage (in the parameter λ).

We give a proof of Theorem 5 in the next section.

6.2 Proof of security

To prove the theorem, we rely on a few technical results.

Lemma 21. *Let $\alpha \in \mathbb{C}^n$ and $A_1, \ldots, A_n \in \mathbb{C}^{m\times m}$. Then, it holds that*

$$\mathrm{Tr}\Big[\sum_{i=1}^{n} \alpha_i A_i\Big] \;\leq\; \|\alpha\|_1 \cdot \sum_{i=1}^{n} |\mathrm{Tr}[A_i]|.$$

Proof. Using the Cauchy-Schwarz inequality, we have

$$\mathrm{Tr}\Big[\sum_{i=1}^{n} \alpha_i A_i\Big] = \sum_{i=1}^{n} \alpha_i \mathrm{Tr}\big[A_i\big] \leq \sqrt{\sum_{i=1}^{n} |\alpha_i|^2} \cdot \sqrt{\sum_{i=1}^{n} |\mathrm{Tr}\big[A_i\big]|^2}.$$

The claim follows from the norm inequality $\|x\|_2 \leq \|x\|_1$, for all $x \in \mathbb{C}^n$. □

Lemma 22. *Let $0 \leq \Pi \leq \mathbb{1}$ and let ρ and σ be states such that $\mathsf{TD}(\rho, \sigma) \leq \gamma$. Then,*

$$\mathrm{Tr}[\Pi\rho] - \gamma \leq \mathrm{Tr}[\Pi\sigma] \leq \mathrm{Tr}[\Pi\rho] + \gamma$$

Proof. By the standard identity $\mathsf{TD}(\sigma, \rho) = \max_{0\leq\Lambda\leq\mathbb{1}} \mathrm{Tr}[\Lambda(\sigma - \rho)]$, it follows that:

$$\begin{aligned}
\mathrm{Tr}[\Pi\sigma] &= \mathrm{Tr}[\Pi\rho] + \mathrm{Tr}[\Pi(\sigma - \rho)] \\
&\leq \mathrm{Tr}[\Pi\rho] + \max_{0\leq\Lambda\leq\mathbb{1}} \mathrm{Tr}[\Lambda(\sigma - \rho)] \\
&= \mathrm{Tr}[\Pi\rho] + \mathsf{TD}(\sigma, \rho) \\
&\leq \mathrm{Tr}[\Pi\rho] + \gamma.
\end{aligned}$$

The other inequality can be shown by reversing the role of ρ and σ. □

Lemma 23 ([Unr15], Lemma 18)**.** *Let $\theta \in \{0,1\}^m$ and define $\Pi^{\mathsf{eq}}_\theta = \sum_{v\in\{0,1\}^m} H^\theta |v\rangle \langle v| H^\theta \otimes H^\theta |v\rangle \langle v| H^\theta$ (i.e the projector that checks if two registers yield the same outcome if measured in the H^θ basis). Then, the following is true for every $t \in [m]$. For any approximate EPR state,*

$$|\phi^+_{ab}\rangle = \frac{1}{\sqrt{2^m}} \sum_{v\in\{0,1\}^m} |v\rangle \otimes X^a Z^b |v\rangle,$$

where $a, b \in \{0,1\}^m$ have Hamming weight at most t, it follows that:

- $\Pi^{\mathsf{eq}}_\theta |\phi^+_{ab}\rangle = |\phi^+_{ab}\rangle$ *holds if and only if for all $i \in [m]$:*

$$(\theta_i = 0 \wedge a_i = 0) \vee (\theta_i = 1 \wedge b_i = 0).$$

- $\Pi^{\mathsf{eq}}_\theta |\phi^+_{ab}\rangle = 0$ *holds for all other cases.*

We also rely on the next lemma which is based on a result by Unruh [Unr15, Lemma 15]. To state the lemma, we define the projector onto the subspace spanned by EPR-pairs in registers XY with up to $t \in \mathbb{N}$ single-qubit Pauli operators applied to register Y:

$$\Pi_t^{\mathsf{EPR}} = \sum_{\substack{a,b\in\{0,1\}^m \\ w(a),w(b)\leq t}} |\phi_{ab}^+\rangle\langle\phi_{ab}^+|, \qquad |\phi_{ab}^+\rangle = \frac{1}{\sqrt{2^m}} \sum_{v\in\{0,1\}^m} |v\rangle \otimes X^a Z^b |v\rangle,$$

where $w(a), w(b)$ denote the Hamming weights of the strings a and b. Since $\{|\phi_{ab}^+\rangle : a,b \in \{0,1\}^m\}$ forms an orthogonal basis of XY, any state ρ such that $(\Pi_t^{\mathsf{EPR}} \otimes \mathbb{1}_{\mathsf{R}})\rho_{\mathsf{XYR}} = \rho_{\mathsf{XYR}}$ on registers X, Y and R can be written as follows (where a, b of weight greater than t have probability zero):

$$\rho_{\mathsf{XYR}} = \sum_{\substack{a,b\in\{0,1\}^m \\ w(a),w(b)\leq t}} p_{ab} \left(|\phi_{ab}^+\rangle\langle\phi_{ab}^+|_{\mathsf{XY}} \otimes \sigma_{\mathsf{R}}^{a,b}\right), \tag{48}$$

for some arbitrary states $\sigma^{a,b}$ and indices $a, b \in \{0,1\}^m$. We show the following lemma:

Lemma 24 (Monogamy uncertainty relation)**.** *Fix a parameter $t \in \mathbb{N}$ and string $\theta \in \{0,1\}^m$. Let ρ be a density matrix on registers X, Y and R such that $(\Pi_t^{\mathsf{EPR}} \otimes \mathbb{1}_{\mathsf{R}})\rho_{\mathsf{XYR}} = \rho_{\mathsf{XYR}}$. Let $\{\Pi_{v'}\}_{v'\in\{0,1\}^m}$ be an arbitrary complete set of orthogonal projectors on register R and measure according to the set $\{H^\theta|v\rangle\langle v|_{\mathsf{X}} H^\theta \otimes \mathbb{1}_{\mathsf{Y}} \otimes \Pi_{v'\mathsf{R}}\}_{v'\in\{0,1\}^m}$. Then,*

$$\Pr[v' = v] = \sum_{v\in\{0,1\}^m} \mathrm{Tr}\left[\left(H^\theta|v\rangle\langle v|_{\mathsf{X}} H^\theta \otimes \mathbb{1}_{\mathsf{Y}} \otimes \Pi_{v\mathsf{R}}\right)\rho_{\mathsf{XYR}}\right] \leq 2^{-m}(m+1)^{2t}.$$

In other words, the min-entropy of the random variable V (with outcome v) given register R is at least $\mathbf{H}_{\min}(V|\mathsf{R}) \geq m - 2t\log(m+1)$.

Proof. For brevity, we define a family of projectors $\{\Lambda_u^\theta\}_u$ acting on registers X and Y, where

$$\Lambda_u^\theta = \left(H^\theta|u\rangle\langle u|_{\mathsf{X}} H^\theta \otimes \mathbb{1}_{\mathsf{Y}}\right).$$

Let T be the set of all possible indices of weight less or equal than t. Now, using decomposition (48), we can bound the success probability of measuring $v' = v$ using the information in the ancilla register R as follows:

$$\begin{aligned}
\Pr[v' = v] &= \sum_{v\in\{0,1\}^m} \mathrm{Tr}\Big[\Big(H^\theta|v\rangle\langle v|_{\mathsf{X}} H^\theta \otimes \mathbb{1}_{\mathsf{Y}} \otimes \Pi_{v\mathsf{R}}\Big)\rho_{\mathsf{XYR}}\Big] \\
&= \sum_{v\in\{0,1\}^m} \mathrm{Tr}\Big[\sum_{\substack{a,b\in\{0,1\}^m \\ w(a),w(b)\leq t}} p_{ab}\left(\Lambda_v^\theta|\phi_{ab}^+\rangle\langle\phi_{ab}^+|_{\mathsf{XY}}\Lambda_v^\theta\right) \otimes \left(\Pi_v\sigma_{\mathsf{R}}^{a,b}\right)\Big] && \text{(by def.)} \\
&\leq \sum_{v\in\{0,1\}^m} \Big(\sum_{\substack{a,b\in\{0,1\}^m \\ w(a),w(b)\leq t}} p_{ab}\Big) \cdot \Big(\sum_{\substack{a,b\in\{0,1\}^m \\ w(a),w(b)\leq t}} \|\Lambda_v^\theta|\phi_{ab}^+\rangle\|^2 \cdot \mathrm{Tr}\big[\Pi_v\sigma_{\mathsf{R}}^{a,b}\big]\Big) && \text{(Lem. 21)} \\
&= \sum_{v\in\{0,1\}^m} \sum_{\substack{a,b\in\{0,1\}^m \\ w(a),w(b)\leq t}} \|H^\theta|v\rangle\langle v|_{\mathsf{X}} H^\theta \otimes \mathbb{1}_{\mathsf{Y}}|\phi_{ab}^+\rangle\|^2 \cdot \mathrm{Tr}\big[\Pi_v\sigma_{\mathsf{R}}^{a,b}\big] && \text{(by def.)} \\
&= \sum_{v\in\{0,1\}^m} \sum_{\substack{a,b\in\{0,1\}^m \\ w(a),w(b)\leq t}} \|H^\theta|v\rangle\langle v|_{\mathsf{X}} H^\theta \otimes X^a Z_{\mathsf{Y}}^b|\phi^+\rangle\|^2 \cdot \mathrm{Tr}\big[\Pi_v\sigma_{\mathsf{R}}^{a,b}\big] \\
&= \sum_{v\in\{0,1\}^m} \sum_{\substack{a,b\in\{0,1\}^m \\ w(a),w(b)\leq t}} \|H^\theta \otimes X^a Z^b H^\theta\big(|v\rangle\langle v|_{\mathsf{X}} \otimes \mathbb{1}_{\mathsf{Y}}\big)|\phi^+\rangle\|^2 \cdot \mathrm{Tr}\big[\Pi_v\sigma_{\mathsf{R}}^{a,b}\big] && \text{(Lem. 1)} \\
&= \sum_{v\in\{0,1\}^m} \sum_{\substack{a,b\in\{0,1\}^m \\ w(a),w(b)\leq t}} \frac{\mathrm{Tr}\big[\Pi_v\sigma_{\mathsf{R}}^{a,b}\big]}{2^m} = \sum_{\substack{a,b\in\{0,1\}^m \\ w(a),w(b)\leq t}} \frac{\mathrm{Tr}\big[\sigma_{\mathsf{R}}^{a,b}\big]}{2^m} = \frac{|T|}{2^m},
\end{aligned}$$

where in the second-to-last step we used the completeness property that $\sum_v \Pi_v = \mathbb{1}$, and in the last step we use that the $\sigma^{a,b}$ have unit trace, for every $a, b \in \{0,1\}^m$. It now suffices to bound $|T|$, the number of error indices of weight less or equal to t. In total we have t indices to assign to $m+1$ possible choices (we add an additional degree of freedom to account for when there are no errors assigned). Since we have two independent indices $a, b \in \{0,1\}^m$, we get:

$$\Pr[v' = v] \leq 2^{-m}|T| \leq 2^{-m}(m+1)^{2t}.$$

This proves the claim. □

Let us now proceed with the security proof. We consider the following sequence of hybrids of $\mathsf{SSLGame}$. We will show that the optimal winning probability in each successive hybrid changes at most negligibly. We will then bound the optimal winning probability in the final hybrid.

H_0: This is the original game $\mathsf{SSLGame}$ in Section 6:

- The lessor runs $\mathsf{SSL.Lease}(1^\lambda, y \in \{0,1\}^\lambda)$ to sample $v \leftarrow \{0,1\}^m$ and $\theta \leftarrow G(y) \in \{0,1\}^m$, and sends $(|v^\theta\rangle, H(v))$ together with a circuit for $\mathsf{SSL.Eval}$ to the adversary $\mathcal{A}$.
- Having access to the random oracles G and H, the adversary $\mathcal{A}$ outputs a (possibly entangled) state σ on two registers Y and R, and sends the register Y to the lessor.
- For verification, the lessor runs $\mathsf{SSL.Verify}(y, \mathsf{Y})$: Measure the register Y in the H^θ basis according to $\theta = G(y)$. If the outcome is equal to v such that $H(v) = z$, the lessor outputs $\mathsf{ok} = 1$ and lets the game continue, otherwise, the lessor outputs $\mathsf{ok} = 0$ and $\mathcal{A}$ loses.
- Conditioning on $\mathsf{ok} = 1$, the lessor sends the adversary a sample $x \leftarrow \mathcal{D}_y$ to which $\mathcal{A}$ responds with a bit (we refer to this phase of the security game as the "input challenge phase"). Using the string y given as input, the lessor outputs 1, if the bit is equal to $P_y(x)$, and 0 otherwise.

H_1: The game is the same as before, except that in the input challenge phase the lessor samples $x \leftarrow D_y$, and sends $G(x)$ to $\mathcal{A}$, (instead of sending x directly).

H_2: The game is the same as before, except for the input challenge phase. The lessor samples $x \leftarrow D_y$. Then, if $x \neq y$, the lessor chooses $\theta' \leftarrow \{0,1\}^m$ and sends θ' to $\mathcal{A}$ (instead of $G(x)$).

H_3: The game is the same as before, except that the lessor samples $\theta \leftarrow \{0,1\}^m$ (instead of $\theta \leftarrow G(y)$). Then, in the input challenge phase, the lessor samples $x \leftarrow D_y$. If $x = y$, the lessor sends θ to $\mathcal{A}$.

H_4: The game is identical to the game before, except that we replace $H(v)$ with a uniformly random string $z \leftarrow \{0,1\}^\lambda$.

First, we show that the advantage of any adversary in H_4 is negligible. Again, in the rest of the section, we denote by $p(H_i)$ the optimal winning probability in hybrid H_i (see the proof in Section 4.1 for a clarification on what the expression $p(H_i)$ means formally).

Lemma 25. $p(H_4) \leq \frac{1}{2}$

Proof. First, the optimal probability of the adversary winning the game can only increase if we remove the verification portion of the game, and the lessor directly executes the input challenge phase.

Then, we consider the state received by the adversary in the two distinct cases of the input challenge phase.

- The lessor samples the marked point. In this case, the state received by the adversary is the following, which is completely independent of the oracle H:

$$\mathbb{E}_{\theta,v}\left(|v^\theta\rangle\langle v^\theta| \otimes |\theta\rangle\langle\theta|\right) \otimes \mathbb{E}_z |z\rangle\langle z| .$$

  Notice that the latter state is maximally mixed.

- The lessor samples a point other than the marked point. In this case, the adversary receives the following state, which is again independent of the oracle:

$$\mathbb{E}_{\theta,\theta',v}\left(|v^\theta\rangle\langle v^\theta|\otimes|\theta'\rangle\langle\theta'|\right)\otimes\mathbb{E}_z|z\rangle\langle z|\,.$$

  The latter state is again maximally mixed.

Thus, an adversary can win the game H_4 with probability at most $\frac{1}{2}$. □

We will now show that the optimal success probabilities in successive hybrids do not deviate by more than a negligible amount.

Lemma 26. $|p(H_1)-p(H_0)|=\mathsf{negl}(\lambda)$.

Proof. The proof is analogous to the proof of Lemma 10 in the security of our copy-protection scheme. The intuition is that, since G is a random oracle, the pre-image x does not help the adversary, and can be simulated. □

Lemma 27. $|p(H_2)-p(H_1)|=\mathsf{negl}(\lambda)$.

Proof. The proof is analogous to the proof of Lemma 11, where an adversary that wins with non-negligible difference in H_2 and H_1 yields a distinguisher for $G(X_\lambda)$ and $U_{m(\lambda)}$. □

Lemma 28. $|p(H_3)-p(H_2)|=\mathsf{negl}(\lambda)$.

Proof. The proof is analogous to the proof of Lemma 12, where an adversary that wins with probabilities that differ non-negligibly in H_3 and H_2 yields a distinguisher for $G(X_\lambda)$ and $U_{m(\lambda)}$. □

The crux of the security proof is showing that $p(H_3)$ and $p(H_4)$ are negligibly close.

Lemma 29. $|p(H_4)-p(H_3)|=\mathsf{negl}(\lambda)$.

The rest of the section is devoted to proving this lemma. At a high level, the proof has two parts:

- For any adversary making q queries to the oracle, we bound the difference between the winning probability in H_3 and in H_4 by $\mathsf{poly}(q)\cdot M$, where M is a quantity related to the probability that the adversary queries the oracle at the encoded string v.
- Then, we show that the quantity M is negligible.

Lemma 30. *Let $\mathcal{A}$ be an adversary for H_3 and H_4, making $\mathsf{poly}(\lambda)$ oracle queries (pre and post verification). Suppose that $\mathcal{A}$ passes the verification step with probability at least $\frac{1}{2}-\mathsf{negl}(\lambda)$ in H_3. Let $\mathcal{A}$ be specified by the unitary U (i.e. $\mathcal{A}$ alternates oracles calls with applications of U). Let $p_{v,\theta,z,H}\in[0,1]$, and let $\rho_{\mathsf{R}}^{v,\theta,z,H}$ be density matrices, for all v,θ,z,H. Let*

$$\sigma_{\mathsf{LR}}=\mathbb{E}_{v,\theta,z,H}\,p_{v,\theta,z,H}\left(|H\rangle\langle H|\otimes|v\rangle\langle v|\otimes|\theta\rangle\langle\theta|\otimes|z\rangle\langle z|\right)_{\mathsf{L}}\otimes\rho_{\mathsf{R}}^{v,\theta,z,H}$$

be the post-verification state of the lessor and $\mathcal{A}$ in H_4 conditioned on $\mathcal{A}$ passing the verification step. Let $\tau_\theta=\frac{1}{2}|\theta\rangle\langle\theta|+\frac{1}{2}\mathbb{E}_{\theta'}|\theta'\rangle\langle\theta'|$. Then,

$$|\Pr[\mathcal{A}\text{ wins in }H_3]-\Pr[\mathcal{A}\text{ wins in }H_4]|\leq\mathsf{poly}(\lambda)\cdot M+\mathsf{negl}(\lambda)\,,$$

where

$$\begin{aligned}M=&\frac{1}{2}\mathbb{E}_H\mathbb{E}_v\mathbb{E}_\theta\mathbb{E}_z\mathbb{E}_k p_{v,\theta,z,H}\mathrm{Tr}\Big[|v\rangle\langle v|\left(UO^{H_{v,z}}\right)^k\left(\rho_{\mathsf{R}}^{v,\theta,z,H}\otimes\tau_\theta\right)\left(UO^{H_{v,z}}\right)^k)^\dagger\Big]\\&+\frac{1}{2}\mathbb{E}_H\mathbb{E}_v\mathbb{E}_\theta\mathbb{E}_z\mathbb{E}_k p_{v,\theta,z,H}\mathrm{Tr}\Big[|v\rangle\langle v|\left(UO^{H}\right)^k\left(\rho_{\mathsf{R}}^{v,\theta,z,H}\otimes\tau_\theta\right)\left(UO^{H}\right)^k)^\dagger\Big].\end{aligned}$$

Proof. As we have done in several earlier proofs, we can recast H_3 as follows: $\mathcal{A}$ receives a uniformly random z, and gets access to a the reprogrammed oracle $H_{v,z}$. Let $|v^\theta\rangle$ denote the encoding of string v using basis θ. Let q_1 and q_2 denote the number of queries made by the adversary respectively before and after the verification phase.

First notice that the global states of the lessor and adversary right before the verification is executed are negligibly close in trace distance in H_3 and H_4.

$$\begin{aligned}
&\mathbb{E}_H\mathbb{E}_v\mathbb{E}_\theta\mathbb{E}_z \left|H\right\rangle\left\langle H\right| \otimes \left|v\right\rangle\left\langle v\right| \otimes \left|\theta\right\rangle\left\langle\theta\right| \otimes \left((UO^{H_{v,z}})^{q_1} \left|v^\theta\right\rangle\left\langle v^\theta\right| \otimes \left|z\right\rangle\left\langle z\right| \left((UO^{H_{v,z'}})^{q_1}\right)^\dagger\right) \\
\approx &\mathbb{E}_H\mathbb{E}_v\mathbb{E}_\theta\mathbb{E}_z \left|H\right\rangle\left\langle H\right| \otimes \left|v\right\rangle\left\langle v\right| \otimes \left|\theta\right\rangle\left\langle\theta\right| \otimes \left((UO^{H})^{q_1} \left|v^\theta\right\rangle\left\langle v^\theta\right| \otimes \left|z\right\rangle\left\langle z\right| \left((UO^{H})^{q_1}\right)^\dagger\right). \qquad (49)
\end{aligned}$$

Here we have stored the complete function H in an additional register, the quantum way of formulating indistinguishability of the joint distribution of H and the adversary's state.

Equation (49) follows from the one-way-to-hiding lemma (Lemma 7), and the fact that $\mathcal{A}$ only queries at v with negligible probability (otherwise $\mathcal{A}$ would straightforwardly imply an adversary that wins the monogamy game (more precisely the variant of Lemma 4).

It follows that:

- The probabilities of $\mathcal{A}$ passing the verification step in H_3 and in H_4 are negligibly close.
- The post-verification states, conditioned on passing verification must be negligibly close (this uses (49) together with the fact that, by hypothesis, $\mathcal{A}$ passes verification with probability at least $\frac{1}{2} - \mathsf{negl}(\lambda)$).

By definition, the joint state of lessor and adversary post-verification state in H_4 conditioned on $\mathcal{A}$ passing verification is

$$\sigma_{\mathsf{LR}} = \mathbb{E}_{v,\theta,z,H}\, p_{v,\theta,z,H} \left(\left|H\right\rangle\left\langle H\right| \otimes \left|v\right\rangle\left\langle v\right| \otimes \left|\theta\right\rangle\left\langle\theta\right| \otimes \left|z\right\rangle\left\langle z\right|\right)_{\mathsf{L}} \otimes \rho_{\mathsf{R}}^{v,\theta,z,H}.$$

Let the analogous state in H_3 be

$$\tilde{\sigma}_{\mathsf{LR}} = \mathbb{E}_{v,\theta,z,H}\, p_{v,\theta,z,H} \left(\left|H\right\rangle\left\langle H\right| \otimes \left|v\right\rangle\left\langle v\right| \otimes \left|\theta\right\rangle\left\langle\theta\right| \otimes \left|z\right\rangle\left\langle z\right|\right)_{\mathsf{L}} \otimes \tilde{\rho}_{\mathsf{R}}^{v,\theta,z,H}.$$

Then $\sigma_{\mathsf{L,R}} \approx \tilde{\sigma}_{\mathsf{L,R}}$. Now, denote by $\{\Pi^0, \Pi^1\}$ the projective measurement performed by $\mathcal{A}$ to guess the answer to the input challenge phase. Then,

$$\begin{aligned}
&\Pr[\mathcal{A} \text{ wins in } H_4 | \text{verification is passed}] \\
&= \mathbb{E}_{v,\theta,z,H}\, p_{v,\theta,z,H} \left[\frac{1}{2}\mathrm{Tr}\left[\Pi^1 (UO^H)^{q_2} \rho_{\mathsf{R}}^{v,\theta,z,H} \otimes \left|\theta\right\rangle\left\langle\theta\right| \left((UO^H)^{q_2}\right)^\dagger\right]\right. \\
&\left. + \frac{1}{2}\mathbb{E}_{\theta'}\mathrm{Tr}\left[\Pi^0 (UO^H)^{q_2} \rho_{\mathsf{R}}^{v,\theta,z,H} \otimes \left|\theta'\right\rangle\left\langle\theta'\right| \left((UO^H)^{q_2}\right)^\dagger\right]\right]. \qquad (50)
\end{aligned}$$

And, similarly,

$$\begin{aligned}
&\Pr[\mathcal{A} \text{ wins in } H_3 | \text{verification is passed}] \\
&= \mathbb{E}_{v,\theta,z,H}\, p_{v,\theta,z,H} \left[\frac{1}{2}\mathrm{Tr}\left[\Pi^1 (UO^{H_{v,z}})^{q_2} \tilde{\rho}_{\mathsf{R}}^{v,\theta,z,H} \otimes \left|\theta\right\rangle\left\langle\theta\right| \left((UO^{H_{v,z}})^{q_2}\right)^\dagger\right]\right. \\
&\left. + \frac{1}{2}\mathbb{E}_{\theta'}\mathrm{Tr}\left[\Pi^0 (UO^{H_{v,z}})^{q_2} \tilde{\rho}_{\mathsf{R}}^{v,\theta,z,H} \otimes \left|\theta'\right\rangle\left\langle\theta'\right| \left((UO^{H_{v,z}})^{q_2}\right)^\dagger\right]\right] \\
&\approx \mathbb{E}_{v,\theta,z,H}\, p_{v,\theta,z,H} \left[\frac{1}{2}\mathrm{Tr}\left[\Pi^1 (UO^{H_{v,z}})^{q_2} \rho_{\mathsf{R}}^{v,\theta,z,H} \otimes \left|\theta\right\rangle\left\langle\theta\right| \left((UO^{H_{v,z}})^{q_2}\right)^\dagger\right]\right. \\
&\left. + \frac{1}{2}\mathbb{E}_{\theta'}\mathrm{Tr}\left[\Pi^0 (UO^{H_{v,z}})^{q_2} \rho_{\mathsf{R}}^{v,\theta,z,H} \otimes \left|\theta'\right\rangle\left\langle\theta'\right| \left((UO^{H_{v,z}})^{q_2}\right)^\dagger\right]\right]. \qquad (51)
\end{aligned}$$

Using equations (50) and (51), and applying the O2H lemma twice (once to bound the distance between the first terms in expressions (50) and (51), and once to bound the distance between the second terms in (50) and (51)), we obtain:

$$
\begin{aligned}
&|\Pr[\mathcal{A} \text{ wins in } H_4 | \text{verification is passed}] - \Pr[\mathcal{A} \text{ wins in } H_3 | \text{verification is passed}]| \\
&\leq \mathsf{poly}(\lambda) \cdot \frac{1}{2} \mathbb{E}_{v,\theta,z,H}\, p_{v,\theta,z,H} \mathrm{Tr}\Big[|v\rangle\langle v| \left(UO^H\right)^k \left(\rho_{\mathsf{R}}^{v,\theta,z,H} \otimes |\theta\rangle\langle\theta|\right) \left(UO^H\right)^k)^\dagger \Big] \\
&+ \mathsf{poly}(\lambda) \cdot \frac{1}{2} \mathbb{E}_{v,\theta,z,H}\, p_{v,\theta,z,H} \mathrm{Tr}\Big[|v\rangle\langle v| \left(UO^{H_{v,z}}\right)^k \left(\rho_{\mathsf{R}}^{v,\theta,z,H} \otimes |\theta\rangle\langle\theta|\right) \left(UO^{H_{v,z}}\right)^k)^\dagger \Big] \\
&+ \mathsf{poly}(\lambda) \cdot \frac{1}{2} \mathbb{E}_{v,\theta,z,H,\theta'}\, p_{v,\theta,z,H} \mathrm{Tr}\Big[|v\rangle\langle v| \left(UO^H\right)^k \left(\rho_{\mathsf{R}}^{v,\theta,z,H} \otimes |\theta'\rangle\langle\theta'|\right) \left(UO^H\right)^k)^\dagger \Big] \\
&+ \mathsf{poly}(\lambda) \cdot \frac{1}{2} \mathbb{E}_{v,\theta,z,H,\theta'}\, p_{v,\theta,z,H} \mathrm{Tr}\Big[|v\rangle\langle v| \left(UO^{H_{v,z}}\right)^k \left(\rho_{\mathsf{R}}^{v,\theta,z,H} \otimes |\theta'\rangle\langle\theta'|\right) \left(UO^{H_{v,z}}\right)^k)^\dagger \Big] + \mathsf{negl}(\lambda) \\
&= \mathsf{poly}(\lambda) \cdot \frac{1}{2} \mathbb{E}_{v,\theta,z,H}\, p_{v,\theta,z,H} \mathrm{Tr}\Big[|v\rangle\langle v| \left(UO^H\right)^k \left(\rho_{\mathsf{R}}^{v,\theta,z,H} \otimes \tau_\theta\right) \left(UO^H\right)^k)^\dagger \Big] \\
&+ \mathsf{poly}(\lambda) \cdot \frac{1}{2} \mathbb{E}_{v,\theta,z,H}\, p_{v,\theta,z,H} \mathrm{Tr}\Big[|v\rangle\langle v| \left(UO^{H_{v,z}}\right)^k \left(\rho_{\mathsf{R}}^{v,\theta,z,H} \otimes \tau_\theta\right) \left(UO^{H_{v,z}}\right)^k)^\dagger \Big] + \mathsf{negl}(\lambda) \\
&= \mathsf{poly}(\lambda) \cdot M + \mathsf{negl}(\lambda)\,, \qquad (52)
\end{aligned}
$$

where to get two equalities we used the definition of τ_θ and M. This is the desired bound. □

In the rest of the section, we show that the quantity M from Lemma 30 is negligible. First of all, notice that M is negligible if and only if the second term in M is negligible, i.e. if and only if,

$$
\mathbb{E}_H \mathbb{E}_v \mathbb{E}_\theta \mathbb{E}_z \mathbb{E}_k p_{v,\theta} \mathrm{Tr}\Big[|v\rangle\langle v| \left(UO^H\right)^k \left(\rho_{\mathsf{R}}^{v,\theta,z,H} \otimes \tau_\theta\right) \left(UO^H\right)^k)^\dagger \Big] = \mathsf{negl}(\lambda)\,. \qquad (53)
$$

where we are using the same notation as in Lemma 30. Thus, what we wish to show is equivalent to showing that, for any adversary $\mathcal{A}$ in H_4 who passes verification with probability at least $\frac{1}{2} - \mathsf{negl}(\lambda)$, the probability of querying the oracle at the encoded string v at any point after a successful verification is negligible.

Thus, we will show that (53) is negligible. First, notice that an adversary $\mathcal{A}$ which passes verification in H_4 with probability at least $\frac{1}{2} - \mathsf{negl}(\lambda)$ and violates (53) immediately implies an adversary which succeeds at the following game $\widetilde{H}_0$ with non-negligible probability.

$\widetilde{H}_0$: This is identical to H_4 except we ask the adversary to return a guess v' for the encoded string v, instead of a bit. $\mathcal{A}$ wins if $v' = v$.

The reduction crucially uses the hypothesis that $\mathcal{A}$ passes verification with probability at least $\frac{1}{2} - \mathsf{negl}(\lambda)$. We will show through another sequence of hybrids (which we denote using tildes) that the optimal winning probability in $\widetilde{H}_0$ is negligible. This will complete the proof that the quantity in (53), and thus M is negligible, for any adversary $\mathcal{A}$ who passes verification with probability at least $\frac{1}{2} - \mathsf{negl}(\lambda)$. Since the optimal winning probability in H_3 and H_4 is at least $\frac{1}{2}$ (the honest strategy followed by random guessing achieves $\frac{1}{2}$), this concludes the proof of Lemma 29, and hence that the optimal winning probability in H_0 is at most $\frac{1}{2} + \mathsf{negl}(\lambda)$. The following are the hybrids.

$\widetilde{H}_1$: Instead of sampling $v \leftarrow \{0,1\}^m$ and $\theta \leftarrow \{0,1\}^m$ at the beginning of the game, the lessor now prepares an EPR pair on two registers X and Y, and sends the registers YZ of the state $|\phi^+\rangle_{\mathsf{XY}} \otimes |z\rangle_{\mathsf{Z}}$ to $\mathcal{A}$. Rather than running $\mathsf{SSL.Verify}$ for verification and measuring the register Y, the lessor now measures both registers X and Y in the H^θ basis for a random $\theta \leftarrow \{0,1\}^m$, and checks if the outcomes result in the same string, which we denote by v.

$\widetilde{H}_2$: This game is identical to the one before, except that we change the verification procedure as follows. Instead of measuring each of the registers X and Y in the H^θ basis, the lessor now measures a bipartite projector Π_θ^{eq} in order to check if the registers XY yield the same outcome if measured in the H^θ basis. We define the projector as follows:

$$
\Pi_\theta^{\mathsf{eq}} = \sum_{v \in \{0,1\}^m} H^\theta |v\rangle\langle v|_{\mathsf{X}} H^\theta \otimes H^\theta |v\rangle\langle v|_{\mathsf{Y}} H^\theta.
$$

Afterwards, the lessor measures register X in the H^θ basis to determine v.

We will denote these hybrids using a tilde to distinguish them from the original sequence of hybrids.

Lemma 31. $p(\widetilde{H}_1) = p(\widetilde{H}_0)$.

Proof. The argument is fairly standard. We consider the following two statements:

- sample $v \leftarrow \{0,1\}^m$, let $\theta \in \{0,1\}^m$, and output $\bigotimes_{i=1}^m |v_i^{\theta_i}\rangle_{\mathsf{Y}}$.
- create an m-qubit EPR pair $|\phi^+\rangle_{\mathsf{XY}}$, measure X in the H^θ basis, and output register Y.

It is evident that the equivalence of the two statements implies that $p(\widetilde{H}_1)$ and $p(\widetilde{H}_0)$ are identical. Note that we omit the register $|z\rangle$ in the proof, since it is independent of the EPR registers and thus does not affect the argument. Consider the following family of projectors given by

$$\{(H^\theta |v\rangle \langle v| H^\theta \otimes \mathbb{1}_{\mathsf{Y}})\}_{v\in\{0,1\}^m}.$$

We analyze the post-measurement state $|\psi_v\rangle / \sqrt{\langle\psi_v | \psi_v\rangle}$, for $|\psi_v\rangle = (H^\theta |v\rangle \langle v|_{\mathsf{X}} H^\theta \otimes \mathbb{1}_{\mathsf{Y}}) |\phi^+\rangle$:

$$\begin{aligned}
|\psi_v\rangle_{\mathsf{XY}} &= (H^\theta |v\rangle \langle v| H^\theta \otimes \mathbb{1}) |\phi^+\rangle_{\mathsf{XY}} \\
&= \Big((H^\theta \otimes \mathbb{1})(|v\rangle \langle v| \otimes \mathbb{1})(H^\theta \otimes \mathbb{1})\Big) |\phi^+\rangle_{\mathsf{XY}} \\
&= \Big((H^\theta \otimes \mathbb{1})(|v\rangle \langle v| \otimes \mathbb{1})(\mathbb{1} \otimes H^\theta)\Big) |\phi^+\rangle_{\mathsf{XY}} && \text{(Lemma 1)} \\
&= 2^{-m/2} \sum_{v'\in\{0,1\}^m} \Big((H^\theta \otimes \mathbb{1})(|v\rangle \langle v| \otimes \mathbb{1})(\mathbb{1} \otimes H^\theta)\Big) |v'\rangle_{\mathsf{X}} \otimes |v'\rangle_{\mathsf{Y}} \\
&= 2^{-m/2} \sum_{v'\in\{0,1\}^m} H^\theta |v\rangle_{\mathsf{X}} \langle v | v'\rangle \otimes H^\theta |v'\rangle_{\mathsf{Y}} \\
&= 2^{-m/2} H^\theta |v\rangle_{\mathsf{X}} \otimes H^\theta |v\rangle_{\mathsf{Y}}.
\end{aligned}$$

This proves the claim, since the Y register of $|\psi_v\rangle / \sqrt{\langle\psi_v | \psi_v\rangle}$ is identical to $\bigotimes_{i=1}^m |v_i^{\theta_i}\rangle$. □

Lemma 32. $p(\widetilde{H}_2) = p(\widetilde{H}_1)$

Proof. The lemma is immediate as the measurement in $\tilde{H}_2$ is a coarse-graining of the measurement in $\tilde{H}_1$, with the acceptance condition remaining the same. □

In the remaining part of the proof, we will show that $p(\widetilde{H}_2)$ is negligible. The following is an important technical lemma, which is inspired by Lemma 16 and Lemma 19 in [Unr15].

Lemma 33. $p(\widetilde{H}_2) = \mathsf{negl}(\lambda)$.

Proof. Let $\mathcal{A}$ be an adversary for $\widetilde{H}_2$. Denote by v' the final guess returned by the adversary, and by v the encoded string. Let ok be a random variable for whether the verification passes. Then, the winning probability of $\mathcal{A}$ in $\widetilde{H}_2$ is given by:

$$\Pr\left[v' = v \wedge \mathsf{ok} = 1\right].$$

We show that, for any $t \in [m]$,

$$\Pr\left[v' = v \wedge \mathsf{ok} = 1\right] \leq 2^{-m}(m+1)^{2t} + 2^{\frac{-t-1}{2}}. \tag{54}$$

Picking $t \approx \sqrt{m}$ then gives the desired result, as the RHS becomes negligible in λ.

Fix a basis choice $\theta \in \{0,1\}^m$. Let ρ_θ be the state on registers X, Y and R in $\widetilde{H}_2$ after the verification, where R is the leftover register held onto by $\mathcal{A}$ that also includes the challenge τ_θ (where τ_θ was defined in Lemma 30) sent by the lessor after verification.

In the analysis that follows, it is convenient to approximate ρ_θ by an ideal state that is diagonal in a basis for the image of $\Pi_t^{\mathsf{EPR}} \otimes \mathbb{1}_{\mathsf{R}}$, where Π_t^{EPR} is as defined in Lemma 24. Recall that Π_t^{EPR}

projects onto the subspace spanned by EPR pairs with up to t Pauli errors, i.e. onto the space spanned by the orthogonal basis states $\left\{ |\phi^+_{ab}\rangle : a,b \in \{0,1\}^m \right\}$, where

$$|\phi^+_{ab}\rangle = \frac{1}{\sqrt{2^m}} \sum_{v\in\{0,1\}^m} |v\rangle \otimes X^a Z^b |v\rangle . \tag{55}$$

We can use Lemma 3 to argue that there exists such an ideal state $\rho^{\mathsf{id}}_\theta$, and that the trace distance between the two states satisfies:

$$\mathsf{TD}(\rho_\theta, \rho^{\mathsf{id}}_\theta) \leq \sqrt{1 - \mathrm{Tr}\left[\left(\Pi^{\mathsf{EPR}}_t \otimes \mathbb{1}_{\mathsf{R}}\right)\rho_\theta\right]}.$$

We can represent the adversary's strategy in guessing v, after verification, by a projective measurement $\{\Pi_{v'}\}_{v'}$.

We are now ready to bound the probability $\Pr\left[v' = v \wedge \mathsf{ok} = 1\right]$. Let Θ be a random variable for the basis choice made by the lessor. Then, by marginalizing over Θ, we get:

$$\begin{aligned}
\Pr\left[v' = v \wedge \mathsf{ok} = 1\right] &= \sum_{\theta\in\{0,1\}^m} 2^{-m} \cdot \Pr\left[v' = v | \,\mathsf{ok} = 1 \wedge \Theta = \theta\right] \cdot \Pr[\mathsf{ok} = 1 | \Theta = \theta] \\
&\leq \sum_{\theta\in\{0,1\}^m} 2^{-m} \cdot \Pr\left[v' = v | \,\mathsf{ok} = 1 \wedge \Theta = \theta\right] \\
&= \mathbb{E}_\theta \Pr[v' = v \,|\, \mathsf{ok} = 1 \wedge \Theta = \theta].
\end{aligned} \tag{56}$$

Fix any θ. Using Lemma 22 and Lemma 24 we obtain:

$$\begin{aligned}
\Pr\left[v' = v \,|\, \mathsf{ok} = 1 \wedge \Theta = \theta\right] &\leq 2^{-m}(m+1)^{2t} + \mathsf{TD}(\rho_\theta, \rho^{\mathsf{id}}_\theta) \\
&\leq 2^{-m}(m+1)^{2t} + \sqrt{1 - \mathrm{Tr}\left[\left(\Pi^{\mathsf{EPR}}_t \otimes \mathbb{1}_{\mathsf{R}}\right)\rho_\theta\right]}.
\end{aligned} \tag{57}$$

Now, averaging over θ in the above inequality gives:

$$\begin{aligned}
\mathbb{E}_\theta \Pr[v' = v \,|\, \mathsf{ok} = 1 \wedge \Theta = \theta] &\leq 2^{-m}(m+1)^{2t} + \mathbb{E}_\theta \sqrt{1 - \mathrm{Tr}\left[\left(\Pi^{\mathsf{EPR}}_t \otimes \mathbb{1}_{\mathsf{R}}\right)\rho_\theta\right]} \\
&\leq 2^{-m}(m+1)^{2t} + \sqrt{\mathbb{E}_\theta \mathrm{Tr}\left[\left(\left(\mathbb{1} - \Pi^{\mathsf{EPR}}_t\right) \otimes \mathbb{1}_{\mathsf{R}}\right)\rho_\theta\right]}.
\end{aligned} \tag{58}$$

where the last inequality follows from Jensen's inequality. We will proceed to bound the above term $\mathbb{E}_\theta \mathrm{Tr}\left[\left(\left(\mathbb{1} - \Pi^{\mathsf{EPR}}_t\right) \otimes \mathbb{1}_{\mathsf{R}}\right)\rho_\theta\right]$ by 2^{-t-1}. Let us first show that for any $a, b \in \{0,1\}^m$:

$$p_{ab} \stackrel{\text{def}}{=} \sum_{\theta\in\{0,1\}^m} 2^{-m} \mathrm{Tr}\left[\left(\mathbb{1} - \Pi^{\mathsf{EPR}}_t\right)\Pi^{\mathsf{eq}}_\theta |\phi^+_{ab}\rangle\langle\phi^+_{ab}|_{\mathsf{XY}}\right] \leq 2^{-t-1}. \tag{59}$$

This follows from considering the following two cases:

- $w(a), w(b) \leq t$: Using Lemma 23 we find that one of the following is true. Depending on θ, either $\Pi^{\mathsf{eq}}_\theta |\phi^+_{ab}\rangle = 0$ or $\Pi^{\mathsf{eq}}_\theta |\phi^+_{ab}\rangle = |\phi^+_{ab}\rangle$. We also get that $\left(\mathbb{1} - \Pi^{\mathsf{EPR}}_t\right)|\phi^+_{ab}\rangle = 0$, since $\Pi^{\mathsf{EPR}}_t |\phi^+_{ab}\rangle = |\phi^+_{ab}\rangle$, and thus it follows that $p_{ab} = 0$.
- $\max\left(w(a), w(b)\right) \geq t+1$: Here, Lemma 23 implies that there are at most $2^m/2^{t+1}$ many values of θ for which it holds that $\Pi^{\mathsf{eq}}_\theta |\phi^+_{ab}\rangle \neq 0$, and thus $p_{ab} \leq 2^{-m} \cdot 2^m/2^{t+1} = 2^{-t-1}$.

Observe now that Π^{EPR}_t and $|\phi^+_{ab}\rangle\langle\phi^+_{ab}|$ are diagonal in the Bell basis, hence they commute. Lemma 23 implies that the same is also true for the projector Π^{eq}_θ. For any fixed $\theta \in \{0,1\}^m$, we express ρ_θ as a generic density operator on registers X, Y and R such that, for a finite index set I^θ, coefficients q_{ij} and an orthogonal basis $\{|\Psi^{i,\theta}\rangle : i \in I^\theta\}$ the registers X and Y:

$$\rho_\theta = \sum_{i,j\in I^\theta} q_{ij} \, |\Psi^{i,\theta}\rangle\langle\Psi^{j,\theta}|_{\mathsf{XY}} \otimes \sigma^{i,j,\theta}_{\mathsf{R}}, \tag{60}$$

where $\sigma^{i,j,\theta}$ are matrices for indices $i, j \in I^\theta$. Since we assumed that ρ_θ is the state conditioned on the verification being successful for some θ, we have the property that

$$\left(\Pi^{\mathsf{eq}}_\theta \otimes \mathbb{1}_{\mathsf{R}}\right)\rho_\theta\left(\Pi^{\mathsf{eq}}_\theta \otimes \mathbb{1}_{\mathsf{R}}\right) = \rho_\theta, \qquad \forall\theta \in \{0,1\}^m. \tag{61}$$

In other words, ρ_θ on is invariant under the action of the projector $\Pi^{\mathsf{eq}}_\theta \otimes \mathbb{1}_{\mathsf{R}}$. Then,

$$
\begin{aligned}
&\mathbb{E}_\theta \mathrm{Tr}\Big[\big(\mathbb{1} - \Pi^{\mathsf{EPR}}_t\big) \otimes \mathbb{1}_{\mathsf{R}}\, \rho_\theta\Big] \\
&= \sum_{\theta\in\{0,1\}^m} 2^{-m} \mathrm{Tr}\big[\big(\mathbb{1} - \Pi^{\mathsf{EPR}}_t\big) \otimes \mathbb{1}_{\mathsf{R}}\, \rho_\theta\big] \\
&= \sum_{\theta\in\{0,1\}^m} 2^{-m} \mathrm{Tr}\big[\big(\mathbb{1} - \Pi^{\mathsf{EPR}}_t\big) \otimes \mathbb{1}_{\mathsf{R}}\, \big(\Pi^{\mathsf{eq}}_\theta \otimes \mathbb{1}_{\mathsf{R}}\big)\rho_\theta\big(\Pi^{\mathsf{eq}}_\theta \otimes \mathbb{1}_{\mathsf{R}}\big)\big] \qquad \text{(Eq. (61))} \\
&= \sum_{\theta\in\{0,1\}^m} 2^{-m} \mathrm{Tr}\Big[\Big(\big(\mathbb{1} - \Pi^{\mathsf{EPR}}_t\big)\Pi^{\mathsf{eq}}_\theta \otimes \mathbb{1}_{\mathsf{R}}\Big)\, \rho_\theta\Big] \\
&= \sum_{\theta\in\{0,1\}^m} 2^{-m} \mathrm{Tr}\Big[\Big(\sum_{a,b\in\{0,1\}^m} |\phi^+_{ab}\rangle\langle\phi^+_{ab}|\Big)\big(\mathbb{1} - \Pi^{\mathsf{EPR}}_t\big)\Pi^{\mathsf{eq}}_\theta \otimes \mathbb{1}_{\mathsf{R}}\, \rho_\theta\Big] \\
&= \sum_{\theta\in\{0,1\}^m} \sum_{a,b\in\{0,1\}^m} 2^{-m}\, \mathrm{Tr}\Big[\, |\phi^+_{ab}\rangle\langle\phi^+_{ab}|\, \big(\mathbb{1} - \Pi^{\mathsf{EPR}}_t\big)\Pi^{\mathsf{eq}}_\theta \otimes \mathbb{1}_{\mathsf{R}}\, \rho_\theta\Big] \\
&= \sum_{\theta\in\{0,1\}^m} \sum_{a,b\in\{0,1\}^m} 2^{-m}\, \mathrm{Tr}\Big[\big(\mathbb{1} - \Pi^{\mathsf{EPR}}_t\big)\Pi^{\mathsf{eq}}_\theta \otimes \mathbb{1}_{\mathsf{R}}\, \big(\, |\phi^+_{ab}\rangle\langle\phi^+_{ab}| \otimes \mathbb{1}_{\mathsf{R}}\big)\rho_\theta\big(\, |\phi^+_{ab}\rangle\langle\phi^+_{ab}| \otimes \mathbb{1}_{\mathsf{R}}\big)\Big]
\end{aligned}
$$

In the third to last line, we inserted the complete set $\sum_{a,b} |\phi^+_{ab}\rangle\langle\phi^+_{ab}| = \mathbb{1}$. Then, using the definition of ρ in Eq.(60), we can continue to expand the expression above as follows:

$$
\begin{aligned}
&\sum_{\theta\in\{0,1\}^m} \sum_{a,b\in\{0,1\}^m} 2^{-m}\, \mathrm{Tr}\Big[\big(\mathbb{1} - \Pi^{\mathsf{EPR}}_t\big)\Pi^{\mathsf{eq}}_\theta \otimes \mathbb{1}_{\mathsf{R}}\, \big(\, |\phi^+_{ab}\rangle\langle\phi^+_{ab}| \otimes \mathbb{1}_{\mathsf{R}}\big)\rho_\theta\big(\, |\phi^+_{ab}\rangle\langle\phi^+_{ab}| \otimes \mathbb{1}_{\mathsf{R}}\big)\Big] \\
&= \sum_{\theta\in\{0,1\}^m} \sum_{a,b\in\{0,1\}^m} 2^{-m} \sum_{i,j\in I^\theta} q_{ij}\, \mathrm{Tr}\Big[\big(\mathbb{1} - \Pi^{\mathsf{EPR}}_t\big)\Pi^{\mathsf{eq}}_\theta\ |\phi^+_{ab}\rangle\langle\phi^+_{ab}|\, \big(\, |\Psi^{i,\theta}\rangle\langle\Psi^{j,\theta}|_{\mathsf{XY}}\big)\, |\phi^+_{ab}\rangle\langle\phi^+_{ab}| \otimes \sigma^{i,j,\theta}_{\mathsf{R}}\Big] \\
&= \sum_{a,b\in\{0,1\}^m} \sum_{i,j\in I^\theta} p_{ab}\, q_{ij}\, \langle\phi^+_{ab}|\, \big(\, |\Psi^{i,\theta}\rangle\langle\Psi^{j,\theta}|_{\mathsf{XY}}\big)\, |\phi^+_{ab}\rangle\ \mathrm{Tr}\big[\sigma^{i,j,\theta}_{\mathsf{R}}\big] \qquad \text{(by def.)} \\
&\le 2^{-t-1} \sum_{i,j\in I^\theta} q_{ij} \sum_{a,b\in\{0,1\}^m} \langle\phi^+_{ab}|\, \big(\, |\Psi^{i,\theta}\rangle\langle\Psi^{j,\theta}|_{\mathsf{XY}}\big)\, |\phi^+_{ab}\rangle\ \mathrm{Tr}\big[\sigma^{i,j,\theta}_{\mathsf{R}}\big] \qquad \text{(Eq. (59))} \\
&= 2^{-t-1} \sum_{i,j\in I^\theta} q_{ij} \mathrm{Tr}\big[\, |\Psi^{i,\theta}\rangle\langle\Psi^{j,\theta}|_{\mathsf{XY}}\big]\, \mathrm{Tr}\big[\sigma^{i,j,\theta}_{\mathsf{R}}\big] \quad = \quad 2^{-t-1}\, \mathrm{Tr}\big[\rho_\theta\big] \quad = \quad 2^{-t-1}.
\end{aligned}
$$

In the last line, we used that $\big\{\, |\phi^+_{ab}\rangle : a,b \in \{0,1\}^m\big\}$ is an orthogonal basis for XY. Thus, we get

$$\mathbb{E}_\theta \mathrm{Tr}\Big[\big(\mathbb{1} - \Pi^{\mathsf{EPR}}_t\big) \otimes \mathbb{1}_{\mathsf{R}}\, \rho_\theta\Big] \ \le\ 2^{-t-1}.$$

Plugging this bound in (58) and then into (56) gives

$$\Pr\left[v' = v \wedge \mathsf{ok} = 1\right] \le 2^{-m}(m+1)^{2t} + 2^{\frac{-t-1}{2}}\,. \tag{62}$$

Choosing $t \approx \sqrt{m}$ makes the RHS negligible. □

Corollary 5. $p(\widetilde{H}_0) = \mathsf{negl}(\lambda)$.

As we argued earlier, this concludes the proof of Lemma (29), and thus of Theorem 5.

6.3 Extension to compute-and-compare programs

In this section, we show that an SSL scheme for point functions, which is secure with respect to the appropriate program and challenge ensembles, implies an SSL scheme for compute-and-compare programs with the same level of security, with respect to apppropriate program and challenge ensembles. The idea is the same as in Section 5: to lease the compute-and-compare program $\mathsf{CC}[f, y]$, we first lease the point function P_y, and then hand out the function f in the clear.

Let (SSL-PF.Gen, SSL-PF.Lease, SSL-PF.Eval, SSL-PF.Verify) be any SSL scheme for point functions. The compute-and-compare program scheme is defined as follows:

Construction 4 (SSL scheme for compute-and-compare programs)**.** *The* SSL *scheme for compute-and-compare programs* (SSL-CC.Gen, SSL-CC.Lease, SSL-CC.Eval, SSL-CC.Verify) *is defined by:*

- SSL-CC.Gen(1^λ)*: Takes as input the security parameter* λ*. Then,*
  - *Let* $\mathsf{sk} \leftarrow$ SSL-PF.Gen(1^λ)*. Output* sk*.*
- SSL-CC.Lease$(1^\lambda, \mathsf{sk}, (f, y))$*: Takes as input a security parameter* λ*, a secret key* sk*, and a compute-and-compare program* $\mathsf{CC}[f, y]$*, specified succinctly by* f *and* y*. Then,*
  - *Let* $\rho =$ SSL-PF.Lease$(1^\lambda, \mathsf{sk}, y))$*. Output* (f, ρ)*.*
- SSL-CC.Eval$(1^\lambda, (f, \rho); x)$*: Takes as input a security parameter* λ*, an alleged program copy* (f, ρ)*, and a string* $x \in \{0,1\}^n$ *(where* n *is the size of the inputs to* f*). Then,*
  - *Compute* $y' = f(x)$*.*
  - *Let* $b \leftarrow$ SSL-PF.Eval$(\rho; y')$*. Output* b*.*
- SSL-CC.Verify$(1^\lambda, \mathsf{sk}, (f, \rho); \sigma)$*:*
  - *Let* $b' \leftarrow$ SSL-PF.Verify$(1^\lambda, \mathsf{sk}, y; \sigma)$*. Output* b'*.*

Recall the definition of the class of distributions over compute-and-compare programs $\mathcal{D}_{\mathsf{CC\text{-}UNP}}$ in Section 5. We recall it here for convenience.

- $\mathcal{D}_{\mathsf{CC\text{-}UNP}}$. We refer to this class as the class of *unpredictable compute-and-compare programs*. This consists of ensembles $D = \{D_\lambda\}$ where D_λ is a distribution over compute-and-compare programs such that $\mathsf{CC}[f, y] \leftarrow D_\lambda$ satisfies $\mathbf{H}_{\min}(y|f) \geq \lambda^\epsilon$ for some $\epsilon > 0$, and where the input length of f is λ and the output length is bounded by some polynomial $t(\lambda)$.

We also define the following class of distributions over input challenges:

- $\mathcal{D}_{\mathsf{CC\text{-}Chall\text{-}SSL}}$. An ensemble $D = \{D_{f,y}\}$, where each $D_{f,y}$ is a distribution over the domain of f, belongs to the class $\mathcal{D}_{\mathsf{CC\text{-}Chall\text{-}SSL}}$ if there exists an efficiently sampleable family $\{X_\lambda\}$ of distributions over $\{0,1\}^\lambda$ with $\mathbf{H}_{\min}(X_\lambda) \geq \lambda^\epsilon$, for some $\epsilon > 0$, and an efficiently sampleable family $\{Z_{f,y}\}$, where $Z_{f,y}$ is a distribution over the set $f^{-1}(y)$, such that $D_{f,y}$ is the following distribution (where λ is the size of inputs to f):
  - with probability $1/2$, sample $z \leftarrow Z_{f,y}$ and output z.
  - with probability $1/2$, sample $x \leftarrow X_\lambda$, and output x.

  We say the ensemble D is *specified* by the families $\{X_\lambda\}$ and $\{Z_{f,y}\}$.

Similarly to Section 5, we also define two classes of distributions over pairs of programs and challenges for compute-and-compare programs.

- $\mathcal{D}_{\mathsf{CC\text{-}pairs\text{-}stat\text{-}SSL}}$. This consists of pairs of ensembles $\left(D = \{D_\lambda\}, D' = \{D'_{f,y}\}\right)$ where $D \in \mathcal{D}_{\mathsf{CC\text{-}UNP}}$ and $D' \in \mathcal{D}_{\mathsf{CC\text{-}Chall\text{-}SSL}}$ satisfying the following. Let D' be specified by the families $\{X_\lambda\}$ and $\{Z_{f,y}\}$, and denote by $\mathsf{MarkedInput}\,(D_\lambda, \{Z_{f,y}\})$ the distribution over $\{0,1\}^\lambda$ induced by D_λ and $\{Z_{f,y}\}$, i.e.:
  - Sample $(f, y) \leftarrow D_\lambda$, then output $z \leftarrow Z_{f,y}$.

  For any fixed f_* with domain $\{0,1\}^\lambda$ such that (f_*, y_*) is in the support of D_λ for some y_*, denote by $\mathsf{MarkedInput}(D_\lambda, \{Z_{f,y}\})|_{f_*}$, the distribution $\mathsf{MarkedInput}(D_\lambda, \{Z_{f,y}\})$ conditioned on D_λ sampling f_*. Then, we require that, for any sequence $\{f_*^{(\lambda)}\}$ (where, for all λ, $(f_*^{(\lambda)}, y_*)$ is in the support of D_λ for some y_*) the families $\{X_\lambda\}$ and $\{\mathsf{MarkedInput}(D_\lambda, \{Z_{f,y}\})|_{f_*^{(\lambda)}}\}$ are statistically indistinguishable.
- $\mathcal{D}_{\mathsf{CC\text{-}pairs\text{-}comp\text{-}SSL}}$. This is defined in the same way as $\mathcal{D}_{\mathsf{CC\text{-}pairs\text{-}stat\text{-}SSL}}$, except that we only require $\{X_\lambda\}$ and $\{\mathsf{MarkedInput}(D_\lambda, \{Z_{f,y}\})|_{f_*^{(\lambda)}}\}$ to be *computationally* indistinguishable.

Theorem 6. *Let* (SSL-PF.Gen, SSL-PF.Lease, SSL-PF.Eval, SSL-PF.Verify) *be an* SSL *scheme for point functions that is* δ*-secure with respect to all pairs* $(D, D') \in \mathcal{D}_{\mathsf{PF\text{-}pairs\text{-}stat\text{-}SSL}}$ *(*$\in \mathcal{D}_{\mathsf{PF\text{-}pairs\text{-}comp\text{-}SSL}}$*). Then, the scheme of Construction 4 is a* δ*-secure* SSL *scheme for compute-and-compare programs with respect to all pairs* $(D, D') \in \mathcal{D}_{\mathsf{CC\text{-}pairs\text{-}stat\text{-}SSL}}$ *(*$\in \mathcal{D}_{\mathsf{CC\text{-}pairs\text{-}comp\text{-}SSL}}$*). The same conclusion holds relative to any oracle, i.e. when all algorithms have access to the same oracle, with respect to query-bounded (computationally bounded) adversaries.*

The proof of Theorem 6 uses a similar reduction to the point function security game as in the copy-protection variant in Theorem 4. The main difference is that the reduction between the SSL games now involves a verification step. We add the proof for completeness.

Proof of Theorem 6. We prove the claim for $(\{D_\lambda\}, \{D_{f,y}\}) \in \mathcal{D}_{\mathsf{CC\text{-}pairs\text{-}stat\text{-}SSL}}$ only, since the case of $(\{D_\lambda\}, \{D_{f,y}\}) \in \mathcal{D}_{\mathsf{CC\text{-}pairs\text{-}comp\text{-}SSL}}$ is virtually identical. Let $t(\lambda)$ be the length of strings in the range of f's sampled from D_λ and let the ensemble $\{D_{f,y}\}$ be specified by $\{X_\lambda\}$ and $\{Z_{f,y}\}$ (using the notation introduced above for ensembles in $\mathcal{D}_{\mathsf{CC\text{-}Chall\text{-}SSL}}$).

Let $\mathcal{A}$ be an adversary for the compute-and-compare SSL scheme of Construction 4 with respect to ensembles $\{D_\lambda\}$ and $\{D_{f,y}\}$ who wins at the SSL security game with probability $p(\lambda) > 0$. It then follows that for each λ there exists $f_*^{(\lambda)}$ such that $(f_*^{(\lambda)}, y)$ is in the support of D_λ for some y, and such that the probability that $\mathcal{A}$ wins is at least $p(\lambda)$, conditioned on $f_*^{(\lambda)}$ being sampled. We will construct an adversary $\mathcal{A}'$ that wins with probability $p(\lambda) - \mathsf{negl}(\lambda)$ in the point function security game with respect to the distributions $\{D'_{t(\lambda)}\}$ and $\{D'_y\}$, defined as follows:

- $D'_{t(\lambda)}$: sample $x \leftarrow X_\lambda$ and output the point function $P_{f_*^{(\lambda)}(x)}$.
- D'_y: sample $x \leftarrow D_{f_*^{(\lambda)},y}$ and output $f_*^{(\lambda)}(x)$.

The adversary $\mathcal{A}'$ against the point function SSL game acts as follows:

- $\mathcal{A}'$ receives a state ρ from the lessor, and then forwards $(f_*^{(\lambda)}, \rho)$ to adversary $\mathcal{A}$.
- $\mathcal{A}$ returns a supposed program copy σ for the point function to $\mathcal{A}'$ who then sends it back to the lessor for verification.
- Conditioning on the verification being successful, the lessor replies with a challenge input $x \leftarrow D'_y$. $\mathcal{A}'$ then samples $x' \leftarrow Z_{f,x}$, and runs $\mathcal{A}$ with input challenge x'.
- Let b be the bit returned by $\mathcal{A}$. The adversary $\mathcal{A}'$ replies with the same b to the lessor.

It is straightforward to check that the game "simulated" by $\mathcal{A}'$ for $\mathcal{A}$ is statistically indistinguishable from a security game with respect to $\{D_\lambda\}$ and $\{D_{f,y}\}$, conditioned on $f_*^{(\lambda)}$. Thus, we deduce, by hypothesis, that $\mathcal{A}$ passes verification and returns the correct bit with probability at least $p(\lambda) - \mathsf{negl}(\lambda)$, and thus $\mathcal{A}'$ wins with probability at least $p(\lambda) - \mathsf{negl}(\lambda)$. Crucially, note that $\left(\{D'_{t(\lambda)}\}, \{D'_y\}\right) \in \mathcal{D}_{\mathsf{PF\text{-}pairs\text{-}stat\text{-}SSL}}$. It follows that if the SSL is δ-secure, then the compute-and-compare scheme must also be δ-secure.

The proof of the theorem statement relative to any oracle is analogous. □

References

[Aar05] Scott Aaronson. Limitations of quantum advice and one-way communication. *Theory of Computing*, 1(1):1–28, 2005.

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

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

[AHU19] Andris Ambainis, Mike Hamburg, and Dominique Unruh. Quantum security proofs using semi-classical oracles. In *Annual International Cryptology Conference*, pages 269–295. Springer, 2019.

[ALP20] Prabhanjan Ananth and Rolando L La Placa. Secure software leasing. *arXiv preprint arXiv:2005.05289*, 2020.

[ALZ20] Scott Aaronson, Jiahui Liu, and Ruizhe Zhang. Quantum copy-protection from hidden subspaces. *arXiv preprint arXiv:2004.09674*, 2020.

[AMRS20] Gorjan Alagic, Christian Majenz, Alexander Russell, and Fang Song. Quantum-access-secure message authentication via blind-unforgeability. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020*, pages 788–817, Cham, 2020. Springer International Publishing.

[BBBV97] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26(5):1510–1523, October 1997.

[BDF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 41–69. Springer, 2011.

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

[BHH+19] Nina Bindel, Mike Hamburg, Kathrin Hövelmanns, Andreas Hülsing, and Edoardo Persichetti. Tighter proofs of cca security in the quantum random oracle model. In Dennis Hofheinz and Alon Rosen, editors, *Theory of Cryptography*, pages 61–90, Cham, 2019. Springer International Publishing.

[BL19] Anne Broadbent and Sébastien Lord. Uncloneable quantum encryption via random oracles. *arXiv preprint arXiv:1903.00130*, 2019.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, CCS '93, page 62–73, New York, NY, USA, 1993. Association for Computing Machinery.

[CGH04] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, July 2004.

[CMSZ19] Jan Czajkowski, Christian Majenz, Christian Schaffner, and Sebastian Zur. Quantum lazy sampling and game-playing proofs for quantum indifferentiability. *arXiv preprint arXiv:1904.11477*, 2019.

[EPR35] A. Einstein, B. Podolsky, and N. Rosen. Can quantum-mechanical description of physical reality be considered complete? *Phys. Rev.*, 47:777–780, May 1935.

[ES20] Edward Eaton and Fang Song. A note on the instantiability of the quantum random oracle. In Jintai Ding and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography*, pages 503–523, Cham, 2020. Springer International Publishing.

[GKW17] Rishab Goyal, Venkata Koppula, and Brent Waters. Lockable obfuscation. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 612–621. IEEE, 2017.

[Got03] Daniel Gottesman. Uncloneable encryption. *Quantum Information & Computation*, 3(6):581–602, 2003.

[HI19] Akinori Hosoyamada and Tetsu Iwata. 4-round luby-rackoff construction is a qprp. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology – ASIACRYPT 2019*, pages 145–174, Cham, 2019. Springer International Publishing.

[KRS09] Robert König, Renato Renner, and Christian Schaffner. The operational meaning of min- and max-entropy. *IEEE Trans. Inf. Theor.*, 55(9):4337–4347, September 2009.

[KSS+20] Veronika Kuchta, Amin Sakzad, Damien Stehlé, Ron Steinfeld, and Shi-Feng Sun. Measure-rewind-measure: Tighter quantum random oracle model proofs for one-way to hiding and cca security. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020*, pages 703–728, Cham, 2020. Springer International Publishing.

[NC11] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, USA, 10th edition, 2011.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '05, page 84–93, New York, NY, USA, 2005. Association for Computing Machinery.

[SW13] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: Deniable encryption, and more. Cryptology ePrint Archive, Report 2013/454, 2013. https://eprint.iacr.org/2013/454.

[TFKW13] Marco Tomamichel, Serge Fehr, Jędrzej Kaniewski, and Stephanie Wehner. A monogamy-of-entanglement game with applications to device-independent quantum cryptography. *New Journal of Physics*, 15(10):103002, 2013.

[Unr12] Dominique Unruh. Quantum proofs of knowledge. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 135–152. Springer, 2012.

[Unr15] Dominique Unruh. Revocable quantum timed-release encryption. *J. ACM*, 62(6), December 2015.

[Wat06] John Watrous. Zero-knowledge against quantum attacks, 2006.

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

[Wil13] Mark M. Wilde. *Quantum Information Theory*. Cambridge University Press, USA, 1st edition, 2013.

[Win99] A. Winter. Coding theorem and strong converse for quantum channels. *IEEE Transactions on Information Theory*, 45(7):2481–2485, 1999.

[WZ17] D. Wichs and G. Zirdelis. Obfuscating compute-and-compare programs under lwe. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 600–611, 2017.

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

[Zha17] Mark Zhandry. Quantum lightning never strikes the same state twice. *CoRR*, abs/1711.02276, 2017.

[Zha19] Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 239–268, Cham, 2019. Springer International Publishing.

A Appendix

A.1 Proof of Lemma 7

Proof. For any $x \in \{0,1\}^\lambda$, define $V_x^H = \left(UO^H(I - |x\rangle\langle x|)\right)^q$ and let $W_x^H = UO^H - V_x^H$. Then,

$$
\begin{aligned}
&\frac{1}{2}\mathbb{E}_H\mathbb{E}_{x\leftarrow X}\|\Pi^0(UO^H)^q\left(|H(x)\rangle\otimes|\psi_x\rangle\right)\|^2+\frac{1}{2}\mathbb{E}_H\mathbb{E}_{z\leftarrow\{0,1\}^m}\|\Pi^1(UO^H)^q\left(|z\rangle\otimes|\psi_x\rangle\right)\|^2\\
=&\frac{1}{2}\mathbb{E}_H\mathbb{E}_{x\leftarrow X}\mathbb{E}_{z\leftarrow\{0,1\}^m}\|\Pi^0(UO^{H_{x,z}})^q\left(|z\rangle\otimes|\psi_x\rangle\right)\|^2+\frac{1}{2}\mathbb{E}_H\mathbb{E}_{z\leftarrow\{0,1\}^m}\|\Pi^1(UO^H)^q\left(|z\rangle\otimes|\psi_x\rangle\right)\|^2\\
=&\frac{1}{2}\mathbb{E}_H\mathbb{E}_{x\leftarrow X}\mathbb{E}_{z\leftarrow\{0,1\}^m}\|\Pi^0(V_x^{H_{x,z}}+W_x^{H_{x,z}})\left(|z\rangle\otimes|\psi_x\rangle\right)\|^2+\frac{1}{2}\mathbb{E}_H\mathbb{E}_{x\leftarrow X}\mathbb{E}_{z\leftarrow\{0,1\}^m}\|\Pi^1(V_x^H+W_x^H)\left(|z\rangle\otimes|\psi_x\rangle\right)\|^2\\
\le&\frac{1}{2}\mathbb{E}_H\mathbb{E}_{x\leftarrow X}\mathbb{E}_{z\leftarrow\{0,1\}^m}\|\Pi^0V_x^{H_{x,z}}\left(|z\rangle\otimes|\psi_x\rangle\right)\|^2+\frac{1}{2}\mathbb{E}_H\mathbb{E}_{x\leftarrow X}\mathbb{E}_{z\leftarrow\{0,1\}^m}\|\Pi^1V_x^H\left(|z\rangle\otimes|\psi_x\rangle\right)\|^2\\
+&\frac{1}{2}(3q+2)q\,\mathbb{E}_H\mathbb{E}_{x\leftarrow X}\mathbb{E}_{z\leftarrow\{0,1\}^m}\mathbb{E}_k\|\,|x\rangle\langle x|\left(UO^{H_{x,z}}\right)^k|z\rangle\otimes|\psi_x\rangle\,\|\\
+&\frac{1}{2}(3q+2)q\,\mathbb{E}_H\mathbb{E}_{x\leftarrow X}\mathbb{E}_{z\leftarrow\{0,1\}^m}\mathbb{E}_k\|\,|x\rangle\langle x|\left(UO^H\right)^k|z\rangle\otimes|\psi_x\rangle\,\|\\
=&\frac{1}{2}\mathbb{E}_H\mathbb{E}_{x\leftarrow X}\mathbb{E}_{z\leftarrow\{0,1\}^m}\|\Pi^0V_x^{H_{x,z}}\left(|z\rangle\otimes|\psi_x\rangle\right)\|^2+\frac{1}{2}\mathbb{E}_H\mathbb{E}_{x\leftarrow X}\mathbb{E}_{z\leftarrow\{0,1\}^m}\|\Pi^1V_x^H\left(|z\rangle\otimes|\psi_x\rangle\right)\|^2\\
+&(3q+2)q\,M
\end{aligned}
\tag{63}
$$

where the first equality uses Lemma 5, and the inequality uses Lemma 18 in [BL19].

In order to prove the desired inequality, it is sufficient to show that

$$\frac{1}{2}\mathbb{E}_H\mathbb{E}_{x\leftarrow X}\mathbb{E}_{z\leftarrow\{0,1\}^m}\|\Pi^0 V_x^{H_{x,z}}\left(|z\rangle\otimes|\psi_x\rangle\right)\|^2+\frac{1}{2}\mathbb{E}_H\mathbb{E}_{x\leftarrow X}\mathbb{E}_{z\leftarrow\{0,1\}^m}\|\Pi^1 V_x^H\left(|z\rangle\otimes|\psi_x\rangle\right)\|^2\leq\frac{1}{2}\,. \tag{64}$$

Notice that $V_x^{H_{x,z}}=V_x^H$, since V_x^H projects onto the subspace orthogonal to x before every query to H. This implies that the LHS simplifies as

$$\begin{aligned}
&\frac{1}{2}\mathbb{E}_H\mathbb{E}_{x\leftarrow X}\mathbb{E}_{z\leftarrow\{0,1\}^m}\|\Pi^0 V_x^{H_{x,z}}\left(|z\rangle\otimes|\psi_x\rangle\right)\|^2+\frac{1}{2}\mathbb{E}_H\mathbb{E}_{x\leftarrow X}\mathbb{E}_{z\leftarrow\{0,1\}^m}\|\Pi^1 V_x^H\left(|z\rangle\otimes|\psi_x\rangle\right)\|^2\\
=&\frac{1}{2}\mathbb{E}_H\mathbb{E}_{x\leftarrow X}\mathbb{E}_{z\leftarrow\{0,1\}^m}\|\Pi^0 V_x^H\left(|z\rangle\otimes|\psi_x\rangle\right)\|^2+\frac{1}{2}\mathbb{E}_H\mathbb{E}_{x\in X}\mathbb{E}_{z\leftarrow\{0,1\}^m}\|\Pi^1 V_x^H\left(|z\rangle\otimes|\psi_x\rangle\right)\|^2\\
=&\frac{1}{2}\mathbb{E}_H\mathbb{E}_{x\in X}\mathbb{E}_{z\leftarrow\{0,1\}^m}\|V_x^H\left(|z\rangle\otimes|\psi_x\rangle\right)\|^2\\
\leq&\frac{1}{2}\,,
\end{aligned} \tag{65}$$

where to get the third line, we used the fact that Π^0,Π^1 are a complete pair of orthogonal projectors, and to get the last line we exploited properties of the Euclidean norm.

Combining (63) and (65) gives the desired inequality.

With a little extra work, one can show that M is negligible if and only if

$$\frac{1}{2}\mathbb{E}_H\mathbb{E}_{x\leftarrow X}\mathbb{E}_{z\leftarrow\{0,1\}^m}\|\Pi^1 V_x^H\left(|z\rangle\otimes|\psi_x\rangle\right)\|^2$$

is negligible. We refer the reader to the proof of Theorem 3 in [AHU19] for the full details. □

A.2 Proof of Lemma 10

We continue the proof of Lemma 10, using the notation we introduced in the main text. In what follows, $\mathcal{P}$, $\mathcal{F}_1$, $\mathcal{F}_2$ always have access to a uniformly random oracle H, but we omit writing this. We have

$$\begin{aligned}
&\Pr[(\mathcal{P}',\mathcal{F}_1',\mathcal{F}_2')\text{ win }H_1]\\
=&\frac{1}{3}\Pr\Big[\mathcal{F}_1^{\hat{G}_{x_1',w_1}}(\mathsf{A},x_1')=0\ \wedge\ \mathcal{F}_2^{\hat{G}_{x_2',w_2}}(\mathsf{B},x_2')=0\\
&\qquad:\mathsf{AB}\leftarrow\rho,\ \ \rho\leftarrow\mathcal{P}^{\hat{G}}\left(|v^{G(y)}\rangle,H(v)\right),P_y\leftarrow D_y,v\leftarrow\{0,1\}^{m(\lambda)},\\
&\qquad\quad w_1,w_2\leftarrow\{0,1\}^{m(\lambda)},x_1',x_2'\leftarrow X_\lambda\,,\,G\leftarrow\mathrm{Bool}(n,m(\lambda))\,,\,\hat{G}\leftarrow\mathrm{Bool}(n,m(\lambda))\Big]\\
+&\frac{1}{3}\Pr\Big[\mathcal{F}_1^{\hat{G}_{x_1',w_1}}(\mathsf{A},x_1')=1\ \wedge\ \mathcal{F}_2^{\hat{G}_{x_2',w_2}}(\mathsf{B},x_2')=0\\
&\qquad:\mathsf{AB}\leftarrow\rho,\ \ \rho\leftarrow\mathcal{P}^{\hat{G}}\left(|v^{G(y)}\rangle,H(v)\right),P_y\leftarrow D_y,v\leftarrow\{0,1\}^{m(\lambda)},\\
&\qquad\quad w_1\leftarrow G(y),w_2\leftarrow\{0,1\}^{m(\lambda)},x_1',x_2'\leftarrow X_\lambda\,,\,G\leftarrow\mathrm{Bool}(n,m(\lambda))\,,\,\hat{G}\leftarrow\mathrm{Bool}(n,m(\lambda))\Big]\\
+&\frac{1}{3}\Pr\Big[\mathcal{F}_1^{\hat{G}_{x_1',w_1}}(\mathsf{A},x_1')=0\ \wedge\ \mathcal{F}_2^{\hat{G}_{x_2',w_2}}(\mathsf{B},x_2')=1\\
&\qquad:\mathsf{AB}\leftarrow\rho,\ \ \rho\leftarrow\mathcal{P}^{\hat{G}}\left(|v^{G(y)}\rangle,H(v)\right),P_y\leftarrow D_y,v\leftarrow\{0,1\}^{m(\lambda)},\\
&\qquad\quad w_1\leftarrow\{0,1\}^{m(\lambda)},w_2\leftarrow G(y),x_1',x_2'\leftarrow X_\lambda\,,\,G\leftarrow\mathrm{Bool}(n,m(\lambda))\,,\,\hat{G}\leftarrow\mathrm{Bool}(n,m(\lambda))\Big]\,.
\end{aligned} \tag{66}$$

For the next step, the key observation is that $\mathcal{P}$ only queries the oracle at x_1' and x_2' with negligible weight. Likewise, $\mathcal{F}_1$ only queries x_2' with negligible weight, and $\mathcal{F}_2$ only queries x_1' with negligible weight. The reason why this is true in this case is that, if it were true, P could be used to construct an adversary that guesses a string sampled from X_λ with non-negligible probability. But X_λ has polynomial min-entropy. Thus, by an application of the one-way-to-hiding lemma, one can replace

the current oracle accesses of $\mathcal{P}$, $\mathcal{F}_1$ and $\mathcal{F}_2$ with oracle access to the function $\hat{G}_{(x_1',w_1),(x_2',w_2)}$. Hence, we have, up to negligible quantities,

$$
\begin{aligned}
(66) = \frac{1}{3}\Pr\Big[\mathcal{F}_1^{\hat{G}_{(x_1',w_1),(x_2',w_2)}}(\mathsf{A},x_1')=0 \ \wedge \ \mathcal{F}_2^{\hat{G}_{(x_1',w_1),(x_2',w_2)}}(\mathsf{B},x_2')=0 \\
: \mathsf{AB}\leftarrow\rho,\ \ \rho\leftarrow\mathcal{P}^{\hat{G}_{(x_1',w_1),(x_2',w_2)}}\left(\left|v^{G(y)}\right\rangle, H(v)\right), P_y\leftarrow D_y, v\leftarrow\{0,1\}^{m(\lambda)}, \\
w_1,w_2\leftarrow\{0,1\}^{m(\lambda)}, x_1',x_2'\leftarrow X_\lambda\,,\ G\leftarrow\mathrm{Bool}(n,m(\lambda))\,,\ \hat{G}\leftarrow\mathrm{Bool}(n,m(\lambda))\Big] \\
+\frac{1}{3}\Pr\Big[\mathcal{F}_1^{\hat{G}_{(x_1',w_1),(x_2',w_2)}}(\mathsf{A},x_1')=1 \ \wedge \ \mathcal{F}_2^{\hat{G}_{(x_1',w_1),(x_2',w_2)}}(\mathsf{B},x_2')=0 \\
: \mathsf{AB}\leftarrow\rho,\ \ \rho\leftarrow\mathcal{P}^{\hat{G}_{(x_1',w_1),(x_2',w_2)}}\left(\left|v^{w_1}\right\rangle, H(v)\right), y\leftarrow\{0,1\}^{\lambda}, v\leftarrow\{0,1\}^{m(\lambda)}, \\
w_1\leftarrow G(y), w_2\leftarrow\{0,1\}^{m(\lambda)}, x_1',x_2'\leftarrow X_\lambda\,,\ G\leftarrow\mathrm{Bool}(n,m(\lambda))\,,\ \hat{G}\leftarrow\mathrm{Bool}(n,m(\lambda))\Big] \\
+\frac{1}{3}\Pr\Big[\mathcal{F}_1^{\hat{G}_{(x_1',w_1),(x_2',w_2)}}(\mathsf{A},x_1')=0 \ \wedge \ \mathcal{F}_2^{\hat{G}_{(x_1',w_1),(x_2',w_2)}}(\mathsf{B},x_2')=1 \\
: \mathsf{AB}\leftarrow\rho,\ \ \rho\leftarrow\mathcal{P}^{\hat{G}_{(x_1',w_1),(x_2',w_2)}}\left(\left|v^{w_2}\right\rangle, H(v)\right), P_y\leftarrow D_y, v\leftarrow\{0,1\}^{m(\lambda)}, \\
w_1\leftarrow\{0,1\}^{m(\lambda)}, w_2\leftarrow G(y), x_1',x_2'\leftarrow X_\lambda\,,\ G\leftarrow\mathrm{Bool}(n,m(\lambda))\,,\ \hat{G}\leftarrow\mathrm{Bool}(n,m(\lambda))\Big]
\end{aligned}
\tag{67}
$$

For the second and third terms, it is convenient to further reprogram the oracle $\hat{G}$ at y, assigning as output a fresh uniformly random value in $\{0,1\}^{m(\lambda)}$. We can do this since none of the algorithms queries at y with non-negligible weight. In this way, we are able to rewrite (67) more concisely, up to negligible terms, as:

$$
\begin{aligned}
\frac{1}{3}\Pr\Big[\mathcal{F}_1^{G}(\mathsf{A},x_1')=0 \ \wedge \ \mathcal{F}_2^{G}(\mathsf{B},x_2')=0 \\
: \mathsf{AB}\leftarrow\rho,\ \ \rho\leftarrow\mathcal{P}^{G}\left(\left|v^{G(y)}\right\rangle, H(v)\right), P_y\leftarrow D_y, v\leftarrow\{0,1\}^{m(\lambda)}, \\
x_1',x_2'\leftarrow X_\lambda\,,\ G\leftarrow\mathrm{Bool}(n,m(\lambda))\Big] \\
+\frac{1}{3}\Pr\Big[\mathcal{F}_1^{G}(\mathsf{A},x_1')=1 \ \wedge \ \mathcal{F}_2^{G}(\mathsf{B},x_2')=0 \\
: \mathsf{AB}\leftarrow\rho,\ \ \rho\leftarrow\mathcal{P}^{G}\left(\left|v^{G(x_1')}\right\rangle, H(v)\right), v\leftarrow\{0,1\}^{m(\lambda)}, \\
x_1',x_2'\leftarrow X_\lambda\,,\ G\leftarrow\mathrm{Bool}(n,m(\lambda))\Big] \\
+\frac{1}{3}\Pr\Big[\mathcal{F}_1^{G}(\mathsf{A},x_1')=1 \ \wedge \ \mathcal{F}_2^{G}(\mathsf{B},x_2')=0 \\
: \mathsf{AB}\leftarrow\rho,\ \ \rho\leftarrow\mathcal{P}^{G}\left(\left|v^{G(x_2')}\right\rangle, H(v)\right), v\leftarrow\{0,1\}^{m(\lambda)}, \\
x_1',x_2'\leftarrow X_\lambda\,,\ G\leftarrow\mathrm{Bool}(n,m(\lambda))\Big]
\end{aligned}
\tag{68}
$$

Finally, notice that

$$
\begin{aligned}
(68) = &\frac{1}{3}\Pr(\mathcal{P},\mathcal{F}_1,\mathcal{F}_2)\text{ win }H_0|\text{ 0-input, 0-input}] \\
&+\frac{1}{3}\Pr(\mathcal{P},\mathcal{F}_1,\mathcal{F}_2)\text{ win }H_0|\text{ 1-input, 0-input}] \\
&+\frac{1}{3}\Pr(\mathcal{P},\mathcal{F}_1,\mathcal{F}_2)\text{ win }H_0|\text{ 0-input, 1-input}] \\
&=p\,,
\end{aligned}
\tag{69}
$$

which yields the desired result.