How to Use Quantum Indistinguishability Obfuscation

Andrea Coladangelo
Paul G. Allen School of Computer Science and
Engineering, University of Washington
Seattle, WA, USA
coladan@cs.washington.edu

Sam Gunn
Department of EECS, UC Berkeley
Berkeley, CA, USA
gunn@berkeley.edu

ABSTRACT

Quantum copy protection, introduced by Aaronson [1], enables giving out a quantum program-description that cannot be meaningfully duplicated. Despite over a decade of study, copy protection is only known to be possible for a very limited class of programs.

As our first contribution, we show how to achieve "best-possible" copy protection for all programs. We do this by introducing *quantum state indistinguishability obfuscation* (qsiO), a notion of obfuscation for *quantum* descriptions of classical programs. We show that applying qsiO to a program immediately achieves best-possible copy protection.

Our second contribution is to show that, assuming injective one-way functions exist, qsiO is concrete copy protection for a large family of puncturable programs — significantly expanding the class of copy-protectable programs. A key tool in our proof is a new variant of unclonable encryption (UE) that we call *coupled unclonable encryption* (cUE). While constructing UE in the standard model remains an important open problem, we are able to build cUE from one-way functions. If we additionally assume the existence of UE, then we can further expand the class of puncturable programs for which qsiO is copy protection.

Finally, we construct qsiO relative to an efficient quantum oracle.

CCS CONCEPTS

• **Theory of computation → Cryptographic primitives**; *Quantum complexity theory*.

KEYWORDS

quantum cryptography, indistinguishability obfuscation

ACM Reference Format:
Andrea Coladangelo and Sam Gunn. 2024. How to Use Quantum Indistinguishability Obfuscation. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing (STOC '24), June 24–28, 2024, Vancouver, BC, Canada.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3618260.3649779

1 INTRODUCTION

A copy-protected program is one that can be evaluated by a user on arbitrary inputs, but not duplicated into a second, functionally equivalent program. Since copy protection is impossible to achieve with classical information alone, Aaronson [1] proposed leveraging quantum information as a way to achieve provable copy protection.

Despite significant research, constructions of copy protection remain elusive. Even *defining* copy protection is often quite subtle, with the right definition depending on the class of programs being copy protected. On the positive side, we know that copy protection can be achieved in either black-box models or for special classes of programs like pseudorandom functions and point functions [2, 6, 8, 16, 17]. On the negative side, it is immediate that learnable programs cannot be copy protected [1], and it is also known that there exist unlearnable programs that cannot be copy protected [8]. Outside of these extremes, the landscape of copy protection remains poorly understood. For instance, our current understanding does not address copy protection for complex non-cryptographic software, e.g. video games. In general, the input/output behavior of a video game has almost no formal guarantees, so it seems difficult to achieve provable copy protection. This leads us to ask,

When are non-cryptographic programs copy protectable? (1)

A useful answer to this question should include conditions that can be heuristically verified in order to determine whether a given program is plausibly copy protectable.

Of course, even if a program *can* be copy protected, it is not in general clear *how* to copy-protect it. We would additionally like to know,

Is there a principled strategy for
copy-protecting programs in general? (2)

In this work we introduce *quantum state indistinguishability obfuscation* (qsiO), which allows us to make progress on both of these questions. To address Question (2), we show that qsiO is *optimal* copy protection for every class of programs. Therefore, assuming qsiO exists, Question (1) reduces to determining which programs are actually copy protected by qsiO. We provide a partial answer to this question by showing that, roughly, copying a qsiO obfuscation is at least as hard as "filling in" the program on an input that has been redacted from the program description.

Quantum state indistinguishability obfuscation (qsiO*).* An obfuscator is an algorithm that takes as input a circuit C and outputs an "unintelligible" program C' with the same functionality as C [9].

The most immediate generalization of this to the quantum setting is an obfuscator that takes as input a (classical description of) a quantum circuit Q and outputs a (classical description of) a functionally equivalent quantum circuit Q'.

However, in this work we will be interested in encoding functionalities (classical or quantum) in *quantum states.* In more detail, if Q is a quantum circuit and ρ is a quantum state, then we say that (Q, ρ) is

STOC '24, June 24–28, 2024, Vancouver, BC, Canada
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0383-6/24/06
https://doi.org/10.1145/3618260.3649779

a *quantum implementation* of a function f if $\Pr[Q(\rho, x) = f(x)] = 1$ for all x in the domain of f.

Several prior works have studied the question of whether obfuscators that are allowed to output quantum implementations are more powerful than obfuscators that can only output classical information, i.e. whether they can obfuscate a larger class of functionalities [4, 5, 8, 10, 11, 14]. However, all of these works consider obfuscators with classical input (and only the output is possibly a quantum state).

In contrast, a quantum state indistinguishability obfuscator Obf takes as input a quantum implementation of some function f, and outputs another quantum implementation of f. We say that Obf is a *quantum state indistinguishability obfuscator* if, for any pair of quantum implementations (Q_1, ρ_1) and (Q_2, ρ_2) of the same function f,

$$\mathsf{Obf}(Q_1, \rho_1) \approx \mathsf{Obf}(Q_2, \rho_2)$$

(where "≈" denotes computational indistinguishability). Note that we only consider obfuscation for (Q, ρ) that implement some function f. In general, one could consider obfuscation for arbitrary quantum functionalities, but this is outside of the scope of our work.

1.1 Our Results

Best-possible copy protection. The connection between qsiO and copy protection becomes clear through the observation that qsiO is *best-possible* copy protection in the following (informal) sense: if a program f can be copy protected, then obfuscating it using qsiO will copy-protect it. This follows from the fact that the qsiO obfuscation of a program is indistinguishable from the qsiO obfuscation of any copy-protected version of the program. Therefore, assuming qsiO exists, Question (1) reduces to determining which qsiO obfuscations result in copy protection.

This result also directly addresses Question (2) by providing a universal heuristic to achieve copy protection. Furthermore, when using qsiO one does not need to worry about the subtleties that arise when defining copy protection for a particular class of programs; we are guaranteed that qsiO will achieve the best possible *kind* of copy protection as well.

A construction of qsiO *relative to a quantum oracle.* In order to support the plausibility of qsiO, we describe a proof-of-principle construction relative to an efficient quantum oracle. It is unclear how this quantum oracle can be heuristically instantiated — however, it is often the case that such oracle constructions are the precursors to simpler instantiable constructions, or standard model constructions.

Copy protection for puncturable programs. The fact that qsiO is best-possible copy protection suggests that we should try to prove that it *is* copy protection for certain classes of functions. We find that exploring conditions under which qsiO is copy protection sheds new light on Question (1) as well.

Assuming injective one-way functions, we show that qsiO copy-protects:

(A) Any puncturable program with "indistinguishability" at the punctured point.

(B) Any puncturable program with "non-reproducibility" at the punctured point, under the additional assumption that unclonable encryption exists.

The idea of puncturing, along with techniques for how to use it, comes from [22] where it is used extensively to build applications of classical *iO*. For convenience, we refer to puncturing with indistinguishability and non-reproducibility at the punctured point as *decision* and *search* puncturing, respectively. A puncturing procedure for a class of programs $\mathcal{F}$ is an efficient algorithm Puncture that takes as input a description of a program $f \in \mathcal{F}$ and a point $x \in \mathrm{Domain}(f)$, and outputs the description of a new program f_x. This program should satisfy $f_x(z) = f(z)$ for all $z \in \mathrm{Domain}(f) \setminus \{x\}$ as well as an additional security property:

- For *decision puncturing*, we require $(f_x, f(x)) \approx (f_x, f(x'))$ for a random x'. In [22] it was shown that one-way functions imply the existence of decision puncturable pseudorandom functions.
- For *search puncturing*, we require that no efficient adversary can compute from f_x any output y such that $\mathsf{Ver}(f, x, y) = 1$, for some efficient (public or private) verification procedure Ver. For example, if f is a signing function with a hard-coded secret key or a message authentication code, $\mathsf{Ver}(f, x, y)$ would use the verification key to check that y is a valid signature or authentication tag for x. In [12] it was shown how to build search puncturable signing functions from indistinguishability obfuscation and one-way functions.

These results highlight some generic properties of programs that imply copy protectability, making progress on Question (1): if a program can be described on *all but one* input (i.e. it can be punctured), then in order to copy a qsiO obfuscation of the original program one must spend a comparable amount of work to that required to fill in the program's value at the missing point.

Techniques for the use of qsiO*.* One of the main contributions of this work is a technical toolkit for the use of qsiO. The reader familiar with classical indistinguishability obfuscation (iO) will recall that it is often used in conjunction with puncturing to obtain interesting applications. For qsiO, we identify *unclonable encryption* [15, 19] as the key primitive that, alongside puncturing, unlocks applications to copy protection. Informally, unclonable encryption is a secret-key encryption scheme where ciphertexts are "unclonable".

As a key technical tool in our proof of (A), we introduce a new variant of unclonable encryption which we call *coupled unclonable encryption.* Whereas constructing (full-fledged) unclonable encryption in the standard model remains an important open problem, we are able to build our variant from one-way functions,[1] and we show that it suffices for (A). Given the notorious difficulty of building unclonable encryption in the standard model, we believe that our variant is of independent interest.

To further showcase our techniques, we show that assuming injective one-way functions and unclonable encryption, qsiO achieves

[1] If one is satisfied with encrypting messages of a *fixed* polynomial length, then cUE exists unconditionally. This is a simple corollary of our result. However, in our applications of cUE, the messages are potentially much longer than the secret keys, and we therefore require a pseudorandom generator.

a strong notion of copy protection for point functions which is beyond the reach of existing techniques.

1.2 Comparison to Previous Work

Two works are particularly related to ours: [2], which also studies copy protection for general programs; and [16], which considers provable copy protection for specific functionalities that are similar to some of the ones we consider here.

[2] takes a very different approach than ours to copy protection for general programs. By moving to a black-box model, they are able to build copy protection for *all* unlearnable programs. However, it is known that there exist unlearnable programs that cannot be copy protected [8], so the black-box construction of [2] does not address Question (1) about *which* programs could be copy protectable. In contrast, qsiO could plausibly exist in the standard model for *all programs.* Furthermore, we are able to identify specific properties that differentiate programs for which qsiO is copy protection.

While the black-box construction of [2] does naturally suggest a heuristic copy protection scheme for arbitrary programs (by replacing black-box obfuscation with iO), there is no "best-possible" guarantee comparable to qsiO. There may exist programs that can be copy protected, and yet this heuristic construction nonetheless fails to copy-protect them. In order to address Question (1), [2] give a non-black-box construction of copy *detection* for any watermarkable program, assuming public-key quantum money. They interpret this construction as evidence that copy *protection* might exist for watermarkable programs as well.

[16] does not directly consider the problem of copy protection for general functionalities. Instead, one of the main results (under an information-theoretic conjecture that was later proven to be true in [18]) is that punturable pseudorandom functions can be copy protected using iO, assuming sub-exponentially-secure LWE. Compared to our provable copy protection results, the advantage of [16] is that iO is much more well-studied than qsiO.[2] However, their result is limited to puncturable pseudorandom functions (and does not seem to extend further), while our results are applicable to a much broader class of puncturable functionalities. Additionally, our results do not rely on "structured" assumptions like LWE.

1.3 Technical Overview

Definitions. Throughout this technical overview, we will fix a universal quantum evaluation circuit Eval. Instead of considering implementations as circuit-state pairs (C, ρ), we will assume that the description of C is included in ρ. Therefore we will view qsiO schemes as acting only on the quantum part, ρ.

As in the introduction, we say that ρ *implements* a function f if, for all x, $\Pr[\mathsf{Eval}(\rho, x) = f(x)] = 1$ (or is negligibly close to 1). An obfuscator Obf is a qsiO scheme if it satisfies:

- (Correctness) if ρ implements f, then $\mathsf{Obf}(\rho)$ implements f, and
- (Security) if ρ, ρ' both implement f, then $\mathsf{Obf}(\rho) \approx \mathsf{Obf}(\rho')$.

We will write $\mathsf{qsiO}(\rho)$ to refer to a qsiO obfuscation of ρ.

Best-possible copy protection. With the definition of qsiO in hand, it is not difficult to prove that $\mathsf{qsiO}(f)$ is best-possible copy protection for any functionality f. Here is a sketch of the argument; for a more complete treatment see the full version.

Let $\mathcal{F}$ be any class of programs for which some copy protection scheme CP exists. That is, CP is an efficient quantum algorithm such that for $f \in \mathcal{F}$, $\mathsf{CP}(f)$ outputs a quantum state ρ such that $\mathsf{Eval}(\rho, x) = f(x)$ for all $x \in \mathsf{Domain}(f)$, and there is some guarantee of "unclonability" on ρ. It turns out that the fact that qsiO is best-possible copy protection is not sensitive to the the precise definition of "unclonability" — whatever definition of unclonability is satisfied by CP, qsiO achieves the same guarantee. The key observation is that any adversary who wins the unclonability game for $\mathsf{qsiO}(f)$ must necessarily win the unclonability game for $\mathsf{qsiO}(\mathsf{CP}(f))$ as well, or else it would break the qsiO security guarantee! Since we can efficiently apply qsiO to $\mathsf{CP}(f)$ to prepare $\mathsf{qsiO}(\mathsf{CP}(f)) \approx \mathsf{qsiO}(f)$, it follows that $\mathsf{qsiO}(f)$ is at least as secure as $\mathsf{CP}(f)$.

Construction of qsiO *relative to a quantum oracle.* Our construction of qsiO relative to a quantum oracle is simple, although the security proof is fairly involved. On input a quantum implementation ρ of some function f, qsiO samples a uniformly random Clifford unitary C and outputs the state $\tilde{\rho} = C\rho C^\dagger$, alongside an oracle implementing the unitary $G_C = C^\dagger \mathsf{Eval} C$, where Eval is a universal circuit. In other words, qsiO applies a Clifford one-time pad to the input state ρ; the oracle G_C undoes the one-time pad, evaluates the function f, and then re-applies the one-time pad.

The "Clifford twirl" is sufficient to argue security against adversaries that make a *single* query, but a more careful argument is required to handle general adversaries. This argument makes use of the "admissible oracle lemma" from [20].

Unclonable encryption. As is often the case with classical iO [22], we find that qsiO does not by itself yield the applications we are most interested in. Instead, we combine qsiO with one-way functions and variants of unclonable encryption to build copy protection. We describe some background and a new result on unclonable encryption before discussing copy protection.

Unclonable encryption (UE), formally introduced by Broadbent and Lord [15],[3] can be viewed as an unclonable version of secret key encryption. A UE scheme consists of a generation algorithm that samples a classical secret key sk, an encryption algorithm Enc that outputs a quantum state, and a decryption algorithm Dec that outputs a message. The security guarantee says that, without the secret key, an adversary given $\mathsf{Enc}(\mathsf{sk}; m)$ cannot prepare two states which can later be used to decrypt the message m (when provided the secret key sk). We require UE schemes to have semantic security — that is, the two states cannot both be used to learn non-negligible information about the message. Formally, a UE scheme $(\mathsf{Enc}, \mathsf{Dec})$ is secure if no efficient adversary can win the following security game with probability noticeably greater than $1/2$:

$\mathtt{UE\text{-}Expt}(\lambda)$:

[2] Despite significant research though, a construction of post-quantum iO from well-founded assumptions is still not known.

[3] The notion was informally put forward by Gottesman in [19], who left constructing it as an open question. Broadbent and Lord [15] formalized the notion, and achieved the first provably secure construction. We remark that Broadbent and Lord refer to what we call unclonable encryption as unclonable encryption with "unclonable indistinguishability."

(1) The adversary sends the challenger a message m.
(2) The challenger samples a challenge bit $c \leftarrow \{0,1\}$ and a secret key $\mathsf{sk} \leftarrow \{0,1\}^\lambda$.
  (a) If $c = 0$, the challenger samples a random message r of the same length as m and sends $\mathsf{Enc}(\mathsf{sk}; r)$ to the adversary.
  (b) If $c = 1$, the challenger sends $\mathsf{Enc}(\mathsf{sk}; m)$ to the adversary.
(3) The adversary splits into two non-communicating parties A and B.
(4) The challenger sends each of A and B the secret key sk.
(5) A outputs a bit a' and B outputs a bit b'. The adversary wins if $a' = b' = c$.

The first provably secure construction of UE was proposed in [15], and it satisfied a "search-based" notion of security in the quantum random oracle model (QROM). Subsequent work [6, 7] achieved the "decision" version of UE that we consider here, still in the QROM. We conjecture that UE for single-bit messages can be built (for general messages) in the standard model, assuming one-way functions.

One of the key insights of Broadbent and Lord [15] is to link the "search-based" notion of UE to the following "monogamy of entanglement" result from [23], which says that no (unbounded) adversary can win the following security game with probability noticeably greater than 0:

`Search-Expt`(λ):

(1) The challenger samples $x, \theta \leftarrow \{0,1\}^\lambda$ and sends $|x^\theta\rangle$ to the adversary. Here, $|x^\theta\rangle$ is shorthand for $H^\theta |x\rangle$, where H^θ denotes Hadamard gates applied to the qubits where the corresponding bit in θ is 1.
(2) The adversary splits into two non-communicating parties A and B.
(3) The challenger sends each of A and B the basis θ.
(4) A and B output strings x_A, x_B. The adversary wins if $x_A = x_B = x$.

The reason that this result does not immediately yield UE (by using x as a one-time pad for the message) is that the adversaries are required to guess *all of* the message in `Search-Expt`, whereas the adversaries in `UE-Expt` are merely required to learn *anything at all about* the message. For instance, if the adversary simply passes the first half of the qubits of $|x^\theta\rangle$ to A and the second half to B, then both A and B can learn half of x. It is natural to attempt to evade this issue by using a randomness extractor. For a single-bit message m, we could use the following as a candidate unclonable encryption:

$$|x^\theta\rangle,\ m \oplus u \cdot x \tag{2}$$

where $x, \theta, u \leftarrow \{0,1\}^\lambda$, and the dot product $u \cdot x$ is taken over $\mathbb{F}_2$. The secret key is $\mathsf{sk} = (\theta, u)$, and the decryption algorithm simply reads x, computes $u \cdot x$, and removes the one-time pad on m.

Intuitively, it would seem that an adversary needs to learn all of x in order to guess $u \cdot x$. This is typically proven using the quantum Goldreich-Levin reduction [3, 13]. Given a single quantum query to a predictor that successfully guesses $u \cdot x$ with probability $1/2 + \varepsilon$ (over a random choice of u), the quantum Goldreich-Levin reduction produces a guess for the entire string x with probability $\mathsf{poly}(\lambda)(\varepsilon)$. Since an adversary that wins `UE-Expt` must have both parts A and B guess $u \cdot x$ correctly, we can run the quantum Goldreich-Levin reduction to show that each of A and B has at least a $\mathsf{poly}(\lambda)(\varepsilon)$ probability of guessing x. However, there is no guarantee that they guess x correctly *simultaneously*, so this reduction might never win `Search-Expt`!

We do not know how to prove that the candidate UE scheme of Equation (2) is secure. Instead, we relax the requirement of UE so that a similar reduction works. This results in a variant of UE that we call *coupled unclonable encryption* (cUE). In cUE, a ciphertext encrypts two messages under two independent secret keys. Each secret key alone works to decrypt the corresponding message. In the security game, A receives one secret key, and B receives the other. Our cUE encryption scheme for single-bit messages m_A, m_B is:

$$|x^\theta\rangle,\ m_A \oplus u \cdot x,\ m_B \oplus v \cdot x \tag{3}$$

where $x, \theta, u, v \leftarrow \{0,1\}^\lambda$. The secret keys are $\mathsf{sk}_A = (\theta, u)$ and $\mathsf{sk}_B = (\theta, v)$. Now that u and v are independent, it is possible to prove that the above reduction works. Indeed, as we were working on this manuscript, similar "simultaneous" Goldreich-Levin theorems were proven in [7, 21]. However, both of these works leave open the question of running a similar reduction for *many-bit* messages. Specifically, in [21], the authors ask whether one can use many inner products to encrypt many bits, noting that their techniques do not extend to this setting. We answer this question in the affirmative, by carrying out a version of a "hybrid argument" on quantum operators.

This result is crucial for our copy protection applications, which require cUE for many-bit messages. Formally, the security guarantee of cUE states that an adversary cannot win the following game with probability noticeably greater than 1/2:

`cUE-Expt`(λ):

(1) The adversary sends the challenger two messages m_A, m_B.
(2) The challenger samples two challenge bits $a, b \leftarrow \{0,1\}$, two secret keys $\mathsf{sk}_A, \mathsf{sk}_B \leftarrow \{0,1\}^\lambda$, and two random messages r_A, r_B of the same lengths as m_A, m_B, respectively.
(3) Let $m_A^0 = m_A, m_B^0 = m_B$, and $m_A^1 = r_A, m_B^1 = r_B$. The challenger sends $\mathsf{Enc}(\mathsf{sk}_A, \mathsf{sk}_B; m_A^a, m_B^b)$ to the adversary.
(4) The adversary splits into two non-communicating parties A and B.
(5) The challenger sends sk_A to A and sk_B to B.
(6) A outputs a bit a' and B outputs a bit b'. The adversary wins if $a' = a$ and $b' = b$.

For general (many-bit) messages m_A, m_B, our cUE encryptions are essentially[4]

$$|x^\theta\rangle,\ m_A \oplus \mathsf{PRG}(Ux),\ m_B \oplus \mathsf{PRG}(Vx). \tag{4}$$

where U, V are wide $\mathbb{F}_2$ matrices of appropriate dimensions, Ux, Vx denote matrix-vector products, and PRG is any pseudorandom generator with appropriate stretch. Since the lengths of Ux and Vx are fixed as a function of λ, but the adversary can choose m_A, m_B of whatever length it wishes, we need to use pseudorandom generators to potentially stretch Ux and Vx to the proper lengths.

We divide the proof of security for Equation (4) into two steps (which we carry out in the full version of the paper). First, we show that one of Ux and Vx is completely unpredictable to the

[4] This construction does not technically satisfy the syntax of `cUE-Expt`, because the secret keys (θ, U) and (θ, V) are not independent. This minor issue is resolved in the full version.

corresponding pirate; we call this property *unclonable randomness*. This is the core of the cUE proof and perhaps the most technical part of this work, requiring a new and delicate argument that resolves the aforementioned open question of [21]. Second, we invoke the security of the PRG to see that the cUE scheme is secure. Thus, assuming only the existence of one-way functions, there exists a cUE scheme that encrypts messages of arbitrary polynomial length.

We then show that cUE suffices to show that qsiO copy-protects puncturable programs with indistinguishability at the punctured point.

Remark 1. *In [6], the authors discuss "issues with using extractors." The proposal for UE in Equation (2) falls within the category of extractor-based schemes that they are referring to, so the issues with natural proof techniques discussed there apply. However, the security of the UE scheme described above is not ruled out by their impossibility result (Theorem 1.3). Furthermore, our constructions of single-bit and general cUE in Equations (3) and (4) are also extractor-based schemes in a similar sense, and we are nonetheless able to prove them secure. Therefore, we hope that our insights for constructing cUE may eventually be useful for constructing UE, as they may evade some of the barriers discussed in [6].*

Finally, we show that one can generically add a functionality that we call *key testing* to any UE or cUE scheme, using qsiO and injective one-way functions. Key testing means that there is an algorithm Test which determines whether a given string z is a valid key for a given encryption σ. Key testing turns out to be crucial for our proofs of copy protection from qsiO. The main idea to upgrade a UE or cUE scheme to one with key testing is to append to the ciphertext a qsiO obfuscation of the program δ_{sk} (which is zero everywhere except at sk). Intuitively, this allows one to check the validity of a secret key, while at the same time preserving unclonability thanks to the properties of qsiO.

Copy protection for PRFs. Armed with cUE, we can apply qsiO to achieve copy protection for certain classes of functions. For the purposes of the technical overview, we will only describe how qsiO copy-protects pseudo-random functions (PRFs). This description highlights some of the main ideas behind our proof technique for the more general results in the full version. The basic idea of the proof technique is to use the qsiO guarantee to replace the PRF with a punctured version, where the values of the PRF at the challenge points are hard coded under a cUE encryption.

We explain this more precisely. Suppose that $\mathcal{F}_\lambda$ is a family of puncturable PRFs with domain $\{0,1\}^\lambda$ and range $\{0,1\}^{n(\lambda)}$. It was shown in [22] that puncturable PRFs can be built from any one-way function. We will prove that qsiO is a secure copy protection scheme for $\mathcal{F}_\lambda$ via a sequence of hybrids, beginning with the PRF copy protection security game:

$\mathtt{CP\text{-}Expt\text{-}PRF}(\lambda)$:

(1) The challenger samples $f \leftarrow \mathcal{F}_\lambda$, $a, b \leftarrow \{0,1\}$, $x_A, x_B \leftarrow \{0,1\}^\lambda$, and $y_A^0, y_B^0 \leftarrow \{0,1\}^{n(\lambda)}$. Let $y_A^1 = f(x_A)$ and $y_B^1 = f(x_B)$.
(2) The challenger sends the adversary qsiO(f).
(3) The adversary splits into two non-communicating parties A and B.
(4) The challenger sends x_A, y_A^a to A and x_B, y_B^b to B.
(5) A outputs a bit a' and B outputs a bit b'. The adversary wins if $a' = a$ and $b' = b$.

In other words, in this security game, the parties A and B are trying to decide whether they received a pair (x, y) where $y = f(x)$ or where y is uniformly random.

Let f_{x_A,x_B} be f punctured at x_A, x_B, let Enc be a cUE scheme with key testing, and let

$$\sigma = \mathsf{Enc}(x_A, x_B; f(x_A), f(x_B)).$$

Our first hybrid uses the qsiO guarantee to replace qsiO(f) with qsiO$(P[f_{x_A,x_B}, \sigma])$, where $P[f_{x_A,x_B}, \sigma]$ is a program (formally a *quantum implementation* of a program) that does the following on input z:

(1) Use key testing to check whether z is a valid key for σ. If not, terminate and output $f_{x_A,x_B}(z)$.
(2) Otherwise, use z to decrypt σ and output the result.

Since $P[f_{x_A,x_B}, \sigma](z) = f(z)$ for all z, we have qsiO$(P[f_{x_A,x_B}, \sigma]) \approx$ qsiO(f). Therefore, the adversary's success probability in the game $\mathtt{CP\text{-}Expt\text{-}PRF}(\lambda)$ does not change if the challenger instead sends qsiO$(P[f_{x_A,x_B}, \sigma])$ instead of qsiO(f) in step 2. Call this modified experiment $\mathtt{Hybrid}_1(\lambda)$.

Now, the pseudorandomness of f at the punctured points implies that

$$(f_{x_A,x_B}, f(x_A), f(x_B), \mathsf{Enc}(x_A, x_B; f(x_A), f(x_B))) \approx (f_{x_A,x_B}, \tilde{y}_A^1, \tilde{y}_B^1, \mathsf{Enc}(x_A, x_B; \tilde{y}_A^1, \tilde{y}_B^1))$$

where $\tilde{y}_A^1, \tilde{y}_B^1$ are random strings from the range of f. Therefore, the adversary's success probability is again preserved if we replace $f(x_A), f(x_B)$ with $\tilde{y}_A^1, \tilde{y}_B^1$ in $\mathtt{Hybrid}_1(\lambda)$. We also rename y_A^0, y_B^0 (introduced in step 1 of the original experiment) to $\tilde{y}_A^0, \tilde{y}_B^0$ for convenience of notation. Then, $\mathtt{Hybrid}_2(\lambda)$ is the following.

$\mathtt{Hybrid}_2(\lambda)$:

(1) The challenger samples $f \leftarrow \mathcal{F}_\lambda$, $a, b \leftarrow \{0,1\}$, $x_A, x_B \leftarrow \{0,1\}^\lambda$, and $\tilde{y}_A^0, \tilde{y}_B^0, \tilde{y}_A^1, \tilde{y}_B^1 \leftarrow \{0,1\}^{n(\lambda)}$.
(2) The challenger prepares $\tilde{\sigma} = \mathsf{Enc}(x_A, x_B; \tilde{y}_A^1, \tilde{y}_B^1)$ and sends the adversary qsiO$(P[f_{x_A,x_B}, \tilde{\sigma}])$.
(3) The adversary splits into two non-communicating parties A and B.
(4) The challenger sends $x_A, \tilde{y}_A^a$ to A and $x_B, \tilde{y}_B^b$ to B.
(5) A outputs a bit a' and B outputs a bit b'. The adversary wins if $a' = a$ and $b' = b$.

Our last hybrid, $\mathtt{Hybrid}_3(\lambda)$, will be the same as $\mathtt{Hybrid}_2(\lambda)$ except that the challenger sends the adversary qsiO$(P[f, \tilde{\sigma}])$ instead of qsiO$(P[f_{x_A,x_B}, \tilde{\sigma}])$ in step 2. The adversary's success probability is negligibly close between $\mathtt{Hybrid}_2(\lambda)$ and $\mathtt{Hybrid}_3(\lambda)$ because $P[f, \tilde{\sigma}]$ and $P[f_{x_A,x_B}, \tilde{\sigma}]$ are functionally equivalent, and so qsiO$(P[f_{x_A,x_B}, \tilde{\sigma}]) \approx$ qsiO$(P[f, \tilde{\sigma}])$.

Finally, notice that $\mathtt{Hybrid}_3(\lambda)$ is now quite close to the cUE experiment $\mathtt{cUE\text{-}Expt}(\lambda)$! It's not difficult to see that there is a direct reduction from $\mathtt{cUE\text{-}Expt}(\lambda)$ to $\mathtt{Hybrid}_3(\lambda)$, because qsiO$(P[f, \tilde{\sigma}])$ can be generated from $\tilde{\sigma}$ by sampling $f \leftarrow \mathcal{F}_\lambda$.

ACKNOWLEDGEMENTS

We thank James Bartusek, Bhaskar Roberts, Mark Zhandry, Henry Yuen, and Fermi Ma for helpful conversations. We would also like

to thank Prabhanjan Ananth for alerting us to the fact that we were missing citations of existing simultaneous Goldreich-Levin theorems in a previous version.

REFERENCES

[1] Scott Aaronson. 2009. Quantum Copy-Protection and Quantum Money. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity, CCC 2009, Paris, France, 15-18 July 2009*. IEEE Computer Society, 229–242. https://doi.org/10.1109/CCC.2009.42

[2] Scott Aaronson, Jiahui Liu, Qipeng Liu, Mark Zhandry, and Ruizhe Zhang. 2021. New Approaches for Quantum Copy-Protection. In *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 12825)*, Tal Malkin and Chris Peikert (Eds.). Springer, 526–555. https://doi.org/10.1007/978-3-030-84242-0_19

[3] Mark Adcock and Richard Cleve. 2002. A Quantum Goldreich-Levin Theorem with Cryptographic Applications. In *STACS 2002*, Helmut Alt and Afonso Ferreira (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 323–334.

[4] Gorjan Alagic, Zvika Brakerski, Yfke Dulek, and Christian Schaffner. 2021. Impossibility of quantum virtual black-box obfuscation of classical circuits. In *Advances in Cryptology–CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part I 41*. Springer, 497–525.

[5] Gorjan Alagic and Bill Fefferman. 2016. On Quantum Obfuscation. *CoRR* abs/1602.01771 (2016). arXiv:1602.01771 http://arxiv.org/abs/1602.01771

[6] Prabhanjan Ananth, Fatih Kaleoglu, Xingjian Li, Qipeng Liu, and Mark Zhandry. 2022. On The Feasibility Of Unclonable Encryption, And More. In *Advances in Cryptology – CRYPTO 2022: 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15–18, 2022, Proceedings, Part II* (Santa Barbara, CA, USA). Springer-Verlag, Berlin, Heidelberg, 212–241. https://doi.org/10.1007/978-3-031-15979-4_8

[7] Prabhanjan Ananth, Fatih Kaleoglu, and Qipeng Liu. 2023. Cloning Games: A General Framework for Unclonable Primitives. In *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part V (Lecture Notes in Computer Science, Vol. 14085)*, Helena Handschuh and Anna Lysyanskaya (Eds.). Springer, 66–98. https://doi.org/10.1007/978-3-031-38554-4_3

[8] Prabhanjan Ananth and Rolando L. La Placa. 2021. Secure Software Leasing. In *Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 12697)*, Anne Canteaut and François-Xavier Standaert (Eds.). Springer, 501–530. https://doi.org/10.1007/978-3-030-77886-6_17

[9] Boaz Barak, Oded Goldreich, Rusell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. 2001. On the (im) possibility of obfuscating programs. In *Annual international cryptology conference*. Springer, 1–18.

[10] James Bartusek, Fuyuki Kitagawa, Ryo Nishimaki, and Takashi Yamakawa. 2023. Obfuscation of Pseudo-Deterministic Quantum Circuits. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, Barna Saha and Rocco A. Servedio (Eds.). ACM, 1567–1578. https://doi.org/10.1145/3564246.3585179

[11] James Bartusek and Giulio Malavolta. 2022. Indistinguishability Obfuscation of Null Quantum Circuits and Applications. In *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 - February 3, 2022, Berkeley, CA, USA (LIPIcs, Vol. 215)*, Mark Braverman (Ed.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 15:1–15:13. https://doi.org/10.4230/LIPIcs.ITCS.2022.15

[12] Mihir Bellare, Igors Stepanovs, and Brent Waters. 2016. New Negative Results on Differing-Inputs Obfuscation. In *Proceedings, Part II, of the 35th Annual International Conference on Advances in Cryptology — EUROCRYPT 2016 - Volume 9666*. Springer-Verlag, Berlin, Heidelberg, 792–821.

[13] Ethan Bernstein and Umesh Vazirani. 1997. Quantum Complexity Theory. *SIAM J. Comput.* 26, 5 (1997), 1411–1473. https://doi.org/10.1137/S0097539796300921 arXiv:https://doi.org/10.1137/S0097539796300921

[14] Anne Broadbent and Raza Ali Kazmi. 2021. Constructions for Quantum Indistinguishability Obfuscation. In *Progress in Cryptology - LATINCRYPT 2021 - 7th International Conference on Cryptology and Information Security in Latin America, Bogotá, Colombia, October 6-8, 2021, Proceedings (Lecture Notes in Computer Science, Vol. 12912)*, Patrick Longa and Carla Ràfols (Eds.). Springer, 24–43. https://doi.org/10.1007/978-3-030-88238-9_2

[15] Anne Broadbent and Sébastien Lord. 2020. Uncloneable Quantum Encryption via Oracles. In *15th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC 2020, June 9-12, 2020, Riga, Latvia (LIPIcs, Vol. 158)*, Steven T. Flammia (Ed.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 4:1–4:22. https://doi.org/10.4230/LIPICS.TQC.2020.4

[16] Andrea Coladangelo, Jiahui Liu, Qipeng Liu, and Mark Zhandry. 2021. Hidden Cosets and Applications to Unclonable Cryptography. In *Advances in Cryptology – CRYPTO 2021*, Tal Malkin and Chris Peikert (Eds.). Springer International Publishing, Cham, 556–584.

[17] Andrea Coladangelo, Christian Majenz, and Alexander Poremba. 2020. Quantum copy-protection of compute-and-compare programs in the quantum random oracle model. *IACR Cryptol. ePrint Arch.* (2020), 1194. https://eprint.iacr.org/2020/1194

[18] Eric Culf and Thomas Vidick. 2022. A monogamy-of-entanglement game for subspace coset states. *Quantum* 6 (2022), 791.

[19] Daniel Gottesman. 2003. Uncloneable encryption. *Quantum Inf. Comput.* 3, 6 (2003), 581–602. https://doi.org/10.26421/QIC3.6-2

[20] Sam Gunn, Nathan Ju, Fermi Ma, and Mark Zhandry. 2023. Commitments to Quantum States. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, Barna Saha and Rocco A. Servedio (Eds.). ACM, 1579–1588. https://doi.org/10.1145/3564246.3585198

[21] Srijita Kundu and Ernest Y.-Z. Tan. 2023. Device-independent uncloneable encryption. arXiv:2210.01058 [quant-ph]

[22] Amit Sahai and Brent Waters. 2021. How to Use Indistinguishability Obfuscation: Deniable Encryption, and More. *SIAM J. Comput.* 50, 3 (2021), 857–908. https://doi.org/10.1137/15M1030108

[23] Marco Tomamichel, Serge Fehr, Jędrzej Kaniewski, and Stephanie Wehner. 2013. A monogamy-of-entanglement game with applications to device-independent quantum cryptography. *New Journal of Physics* 15, 10 (oct 2013), 103002. https://doi.org/10.1088/1367-2630/15/10/103002

Received 2023-11-12; accepted 2024-02-11