How to Use Quantum Indistinguishability Obfuscation

Andrea Coladangelo[*] Sam Gunn[†]

December 8, 2023

Abstract

Quantum copy protection, introduced by Aaronson [Aar09], 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.

[*] Paul G. Allen School of Computer Science and Engineering, University of Washington. Email: `coladan@cs.washington.edu`.

[†] UC Berkeley. Email: `gunn@berkeley.edu`. Supported by a Google PhD Fellowship.

Contents

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 [Aar09] 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 [CMP20, AP21, ALL+21, CLLZ21, AKL+22]. On the negative side, it is immediate that learnable programs cannot be copy protected [Aar09], and it is also known that there exist unlearnable programs that cannot be copy protected [AP21]. 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 [BGI+01].

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 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 [AF16, BK21, AP21, ABDS21, BM22, BKNY23]. 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 "$\approx$" 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 [SW21] 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 $\mathsf{Puncture}$ that takes as input a description of a program $f \in \mathcal{F}$ and a point $x \in \text{Domain}(f)$, and outputs the description of a new program f_x. This program should satisfy $f_x(z) = f(z)$ for all $z \in \text{Domain}(f) \setminus \{x\}$ 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 [SW21] 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 [BSW16] 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* as the key primitive that, alongside puncturing, unlocks applications to copy protection.

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 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: [ALL+21], which also studies copy protection for general programs; and [CLLZ21], which considers provable copy protection for specific functionalities that are similar to some of the ones we consider here.

[ALL+21] 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 [AP21], so the black-box construction of [ALL+21] 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 [ALL+21] 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), [ALL+21] 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.

[CLLZ21] 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 [CV22]) 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 [CLLZ21] 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:

[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.

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

- (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 Theorem 1.

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 $x \in \text{Domain}(f)$, and there is some guarantee of "unclonability" on ρ. It turns out that Theorem 1 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 [GJMZ23].

Unclonable encryption. As is often the case with classical iO [SW21], 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) 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:

$\texttt{UE-Expt}(\lambda)$:

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 $a' = b' = c$.

Currently, the only provably secure constructions of UE are in the quantum random oracle model [AKL$^+$22, AKL23]. We conjecture that UE for single-bit messages can be built unconditionally in the standard model, and for general messages assuming one-way functions. In fact, a result of [TFKW13] already proved that a simple scheme based on "monogamy of entanglement" yields an unconditional "search" version of UE. Their result 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 $\left|x^\theta\right\rangle$ to the adversary. Here, $\left|x^\theta\right\rangle$ is shorthand for $H^\theta \left|x\right\rangle$, 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 $\left|x^\theta\right\rangle$ 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:

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

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 [BV97, AC02]. 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 $\mathsf{poly}(\varepsilon)$. 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 $\mathsf{poly}(\varepsilon)$ 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:

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

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 [KT23, AKL23]. However, both of these works leave open the question of running a similar reduction for *many-bit* messages. Specifically, in [KT23], 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 in Section 3.1, 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 $1/2$:

$\texttt{cUE-Expt}(\lambda)$:

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[3]

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

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. First, in Section 3.1 we show that one of Ux and Vx is completely unpredictable to the 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 [KT23]. In Section 3.2, 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.

In Section 4.2, we show that cUE suffices to show that qsiO copy-protects puncturable programs with indistinguishability at the punctured point.

Remark 1. *In [AKL$^+$22], the authors discuss "issues with using extractors." The proposal for UE in Equation (2) falls within the category of extractor-based schemes that they are referring to, so the issues with natural proof techniques discussed there apply. However, the security of the UE scheme described above is not ruled out by their impossibility result (Theorem 1.3). Furthermore, our constructions of single-bit and general cUE in Equations (3) and (4) are also extractor-based schemes in a similar sense, and we are nonetheless able to prove them secure. Therefore, we hope that our insights for constructing cUE may eventually be useful for constructing UE, as they may evade some of the barriers discussed in [AKL$^+$22].*

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.

[3] This construction does not technically satisfy the syntax of $\texttt{cUE-Expt}$, because the secret keys (θ, U) and (θ, V) are not independent. This minor issue is resolved in Section 3.2.

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 of Section 4. 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 [SW21] 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:

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

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 $\mathsf{qsiO}(f)$.
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 $\mathsf{qsiO}(f)$ with $\mathsf{qsiO}(P[f_{x_A,x_B}, \sigma])$, 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, $\mathsf{qsiO}(P[f_{x_A,x_B}, \sigma]) \approx \mathsf{qsiO}(f)$. Therefore, the adversary's success probability in $\texttt{CP-Expt-PRF}(\lambda)$ does not change if the challenger instead sends $\mathsf{qsiO}(P[f_{x_A,x_B}, \sigma])$ instead of $\mathsf{qsiO}(f)$ in step 2. Call this modified experiment $\texttt{Hybrid}_1(\lambda)$.

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 $\texttt{Hybrid}_1(\lambda)$. 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, $\texttt{Hybrid}_2(\lambda)$ is the following.

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

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

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

1.4 Preliminaries

We introduce some notation that we will use throughout the paper.

We denote a *quantum polynomial-time algorithm* with the acronym QPT. Formally, this is a polynomial-time uniform family of quantum circuits, where each circuit in the family is specified by a sequence of unitary operations and measurements. A quantum algorithm may in general receive (mixed) quantum states as inputs and produce (mixed) quantum states as outputs.

For a distribution D, the notation $x \leftarrow D$ denotes sampling an element from D; for a set S, $x \leftarrow S$ denotes sampling an element uniformly at random from S. For distributions $\mathcal{D}, \mathcal{D}'$, we write $\mathcal{D} \approx \mathcal{D}'$ and $\mathcal{D} \equiv \mathcal{D}'$ to indicate computational and statistical indistinguishability, respectively.

We denote by $\mathcal{C}_d$ the Clifford group for dimension d, i.e., the set of d-dimensional unitary operators that conjugate d-dimensional generalized Pauli matrices to d-dimensional generalized Pauli matrices. If the dimension is clear from the context, we simply write $\mathcal{C}$.

We denote by δ_S the indicator function for a set S, with $\delta_S(x) = 1$ if $x \in S$ and $\delta_S(x) = 0$ otherwise. For a point s, it is understood that $\delta_s := \delta_{\{s\}}$.

For a string $x \in \{0,1\}^n$, we use $|x| = n$ to denote the length of the string. By default, all operations on bitstrings are assumed to be performed over $\mathbb{F}_2$.

1.5 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.

2 Quantum State Indistinguishability Obfuscation (qsiO)

In this section, we define quantum state indistinguishability obfuscation (qsiO). We show that qsiO achieves "best-possible" copy protection, and we describe a construction of qsiO relative to a quantum oracle.

2.1 Definitions

We start by defining a "quantum implementation" of a classical function.

Definition 1 (Quantum implementation of a classical function)**.** *Let $l_{in}, l_{out} \in \mathbb{N}$, $f : \{0,1\}^{l_{in}} \to \{0,1\}^{l_{out}}$, and $\epsilon \in [0,1]$. A* $(1-\epsilon)$-quantum implementation *of f is a pair (ρ, C), where ρ is a state and C is a quantum circuit, such that*

$$\forall x \in \{0,1\}^{l_{in}},\ \Pr[C(\rho, x) = f(x)] \geq 1 - \epsilon\,.$$

For a quantum implementation (ρ, C), we refer to its *size* as the maximum between the number of qubits of ρ and the number of gates of the circuit C. We now define qsiO.

Definition 2 (Quantum state indistinguishability obfuscator (qsiO))**.** *Let $\{Q_\lambda\}_{\lambda\in\mathbb{N}}$ be a family of $(1 - \mathsf{negl}(\lambda))$-quantum implementations (of possibly different functions), where negl is some negligible function. A* quantum state indistinguishability obfuscator *for $\{Q_\lambda\}_{\lambda\in\mathbb{N}}$ is a QPT algorithm qsiO that takes as input a security parameter 1^λ, a quantum implementation $(\rho, C) \in Q_\lambda$, and outputs a pair (ρ', C'). Additionally, qsiO should satisfy the following.*

- *(Correctness) There exists a negligible function negl' such that, for any $\lambda \in \mathbb{N}$, if $(\rho, C) \in Q_\lambda$ is a $(1 - \mathsf{negl}(\lambda))$-quantum implementation of some function f, then $\mathsf{qsiO}(\rho, C)$ is a $(1 - \mathsf{negl}'(\lambda))$-quantum implementation of f.*
- *(Security) For any QPT distinguisher D, there exists a negligible function negl'' such that the following holds. For all λ and all pairs of $(1 - \mathsf{negl}(\lambda))$-quantum implementations $(\rho_0, C_0), (\rho_1, C_1) \in Q_\lambda$ of the same function f,*

$$\Big| \Pr\big[D(1^\lambda, \mathsf{qsiO}(1^\lambda, (\rho_0, C_0))) = 1\big] - \Pr\big[D(1^\lambda, \mathsf{qsiO}(1^\lambda, (\rho_1, C_1))) = 1\big] \Big| \leq \mathsf{negl}''(\lambda)\,.$$

In this paper, we will make use of qsiO for all polynomial-size quantum implementations. That is, we will assume the existence of qsiO for $\{Q_\lambda\}_{\lambda\in\mathbb{N}}$, where Q_λ is the set of $(1 - \mathsf{negl}(\lambda))$-quantum implementations of size at most λ, for some negligible function negl.

For ease of notation, we will often omit writing 1^λ as an input to qsiO. We will sometimes apply qsiO to a circuit C *without* auxiliary quantum input, or to a classical circuit C. In this case, we simply write $\mathsf{qsiO}(C)$. If the circuit C is classical we sometimes identify it with the function f that it is computing, and simply write $\mathsf{qsiO}(f)$.

2.2 Best-possible copy protection

It is not hard to see that qsiO, as defined in the previous section, achieves *best-possible* copy protection. In this section, we state a definition of copy protection that is quite general, and encompasses all the variants that we will later consider in Section 4.

Definition 3 (Copy protection, correctness). *Let $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ be a family of classical circuits. A QPT algorithm CP is a copy protection scheme for $\mathcal{F}$ if the following holds, for some negligible function negl:*

- *CP takes as input a security parameter 1^λ and a circuit $f \in \mathcal{F}_\lambda$, and outputs a $(1 - \mathsf{negl}(\lambda))$-quantum implementation (ρ, C) of f with probability $1 - \mathsf{negl}(\lambda)$.*

The definition of security below is stated in terms of a circuit Ver that the challenger runs on each half of a state received from the adversary (the "pirate"). Some readers may be more familiar with a security game where the challenger samples a pair of inputs to the copy-protected function f, and expects two parties Alice and Bob to return the value of f at those inputs. The security game in Figure 1 subsumes such a security game (by taking $\mathsf{Ver}(f, A)$ to be the circuit that first samples an input x to f, and then runs "Alice's circuit" A^x on the input state). We elect to keep the definition general here, so as not to limit the applicability of our "best-possible copy protection" result (Theorem 1). Later, when we discuss copy protection of concrete functionalities in Section 4, we opt for a more explicit description of the security game.

Definition 4 (Copy protection, security)**.** *Let $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ be a family of classical circuits. Let CP be a copy protection scheme for $\mathcal{F}$.*

Let $\mathsf{Ver} = \{\mathsf{Ver}_\lambda\}_{\lambda\in\mathbb{N}}$ be a uniform family of polynomial-size quantum circuits, where Ver_λ takes as input a function $f \in \mathcal{F}_\lambda$, a family of $\mathsf{poly}(\lambda)$-size quantum circuits $\{Q^x\}_{x\in\mathrm{Domain}(f)}$, and a quantum state, and outputs a single bit. Let $\delta : \mathbb{N} \to [0, 1]$.

We say that CP is (Ver, δ)-secure if, for all QPT algorithms Adv, there exists a negligible function negl such that, for all λ,

$$\Pr\left[\texttt{CP-Expt}_{\mathsf{CP},\mathsf{Adv},\mathsf{Ver}}(\lambda) = 1\right] \leq \delta(\lambda) + \mathsf{negl}(\lambda)\,,$$

where $\texttt{CP-Expt}_{\mathsf{CP},\mathsf{Adv},\mathsf{Ver}}$ is defined in Figure 1.[4] *For a function $f \in F_\lambda$ and a family of circuits Q, we use the notation $\mathsf{Ver}(f, Q) := \mathsf{Ver}(f, Q, \cdot)$ (so $\mathsf{Ver}(f, Q)$ denotes a quantum circuit that takes as input a state and outputs a single bit).*

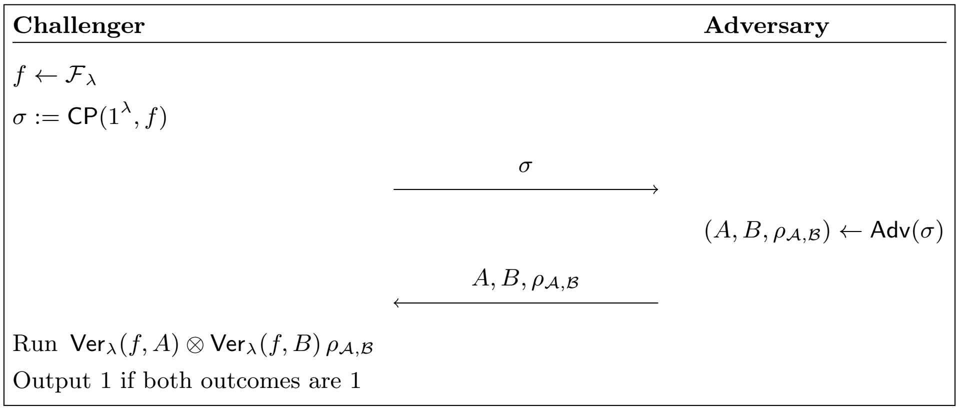

Challenger
Adversary
$f \leftarrow \mathcal{F}_\lambda$
$\sigma := \mathsf{CP}(1^\lambda, f)$
σ
$(A, B, \rho_{\mathcal{A},\mathcal{B}}) \leftarrow \mathsf{Adv}(\sigma)$
$A, B, \rho_{\mathcal{A},\mathcal{B}}$
Run $\mathsf{Ver}_\lambda(f, A) \otimes \mathsf{Ver}_\lambda(f, B)\, \rho_{\mathcal{A},\mathcal{B}}$
Output 1 if both outcomes are 1

Figure 1: $\texttt{CP-Expt}_{\mathsf{CP},\mathsf{Adv},\mathsf{Ver}}(\lambda)$. The challenger samples $f \leftarrow \mathcal{F}_\lambda$, creates the quantum implementation $\sigma = \mathsf{CP}(f)$, and sends it to the adversary. The adversary maps this to a state $\rho_{\mathcal{A}\mathcal{B}}$ on the two registers $\mathcal{A}$, $\mathcal{B}$, and sends $\rho_{\mathcal{A}\mathcal{B}}$ back to the challenger, along with (descriptions of) families of quantum circuits A and B on $\mathcal{A}$ and $\mathcal{B}$ respectively. The challenger runs $\mathsf{Ver}_\lambda(f, A) \otimes \mathsf{Ver}_\lambda(f, B)$ on $\rho_{\mathcal{A},\mathcal{B}}$, and outputs 1 if both outcomes are 1.

Theorem 1 ("Best-possible" copy protection)**.** *Let $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ be a family of classical circuits. Suppose there exists a copy protection scheme for $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ that is (Ver, δ)-secure (for some Ver, and δ as in Definition 4). Let qsiO be a secure quantum state indistinguishability obfuscator for $\mathcal{F}$. Then, qsiO is a (Ver, δ)-secure copy protection scheme for $\mathcal{F}$.*

[4] Note that in this security game the function f is sampled uniformly from $\mathcal{F}_\lambda$. This restriction is essentially without loss of generality, since one can pad the description of the circuits f with additional bits that do not affect the circuit itself, but serve the purpose of changing the probability mass on a particular circuit.

Proof. Let CP be the (Ver, δ)-secure copy protection scheme for $\mathcal{F}$ that exists by hypothesis. Let Adv be any efficient adversary for $\texttt{CP-Expt}$. Since the challenger in $\texttt{CP-Expt}$ is also efficient, we have by the security guarantee of qsiO, that there exists a negligible function negl such that, for all λ,

$$\Pr\left[\texttt{CP-Expt}_{\mathsf{qsiO},\mathsf{Adv},\mathsf{Ver}}(\lambda) = 1\right] \leq \Pr\left[\texttt{CP-Expt}_{\mathsf{qsiO}\circ\mathsf{CP},\mathsf{Adv},\mathsf{Ver}}(\lambda) = 1\right] + \mathsf{negl}(\lambda). \tag{5}$$

Consider a reduction $\mathsf{Red}[\mathsf{Adv}]$ for $\texttt{CP-Expt}$ that simply applies qsiO before forwarding σ to Adv, and then forwards the response from Adv to the challenger. Formally, Red is defined by the following behavior in $\texttt{CP-Expt}$.

$\texttt{CP-Expt}_{\mathsf{CP},\mathsf{Red}[\mathsf{Adv}],\mathsf{Ver}}(\lambda)$:

1. The challenger samples $f \leftarrow \mathcal{F}_\lambda$, computes $\sigma = \mathsf{CP}(f)$, and sends σ to Red.
2. Red computes $\sigma' = \mathsf{qsiO}(\sigma) = (\mathsf{qsiO} \circ \mathsf{CP})(f)$ and sends σ' to Adv.
3. Adv sends $(A, B, \rho_{\mathcal{A}\mathcal{B}})$ to Red, which forwards this to the challenger.
4. The challenger then runs $\mathsf{Ver}_\lambda(f, A) \otimes \mathsf{Ver}_\lambda(f, B)\, \rho_{\mathcal{A},\mathcal{B}}$ and outputs 1 if both outcomes are 1.

By construction,

$$\Pr\left[\texttt{CP-Expt}_{\mathsf{qsiO}\circ\mathsf{CP},\mathsf{Adv},\mathsf{Ver}}(\lambda) = 1\right] = \Pr\Big[\texttt{CP-Expt}_{\mathsf{CP},\mathsf{Red}[\mathsf{Adv}],\mathsf{Ver}}(\lambda) = 1\Big]. \tag{6}$$

Finally, the assumption that CP is a (Ver, δ)-secure copy protection scheme for $\mathcal{F}$ implies that

$$\Pr\Big[\texttt{CP-Expt}_{\mathsf{CP},\mathsf{Red}[\mathsf{Adv}],\mathsf{Ver}}(\lambda) = 1\Big] = \mathsf{negl}(\lambda). \tag{7}$$

Combining Equations (5), (6), and (7) gives the result. □

Remark 2. *In this work we only consider* qsiO *for quantum implementations of deterministic functions. It would be interesting to explore an extended definition that allows for quantum implementations of* randomized *functions. It is plausible that a proper formalization would yield best-possible one time programs in a similar way to Theorem 1.*

2.3 Constructing qsiO

In this section, we give a construction Obf of qsiO relative to a quantum oracle. Before describing it formally, we give an informal description:

- Obf takes as input a quantum implementation (ρ, Eval) of some function f, where Eval is assumed to be a universal evaluation circuit without loss of generality.
- Obf samples a uniformly random Clifford unitary C and outputs the state $\tilde{\rho} = C\rho C^\dagger$, alongside an oracle implementing the unitary $G_C = C\mathsf{Eval}C^\dagger$ (where Eval here refers to the unitary part of the evaluation circuit).

In other words, qsiO applies a Clifford one-time pad to "hide" 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. This allows a user to evaluate f, while intuitively keeping the state ρ hidden at all times.

The main tool in our proof is the "Clifford twirl" [ABOEM17], which would already suffice if the adversary were only allowed to make a *single* query to G_C. However, the adversary can make any polynomial number of queries, so a more careful argument is required. Our argument additionally makes use of a recently-introduced tool called the "admissible oracle lemma" [GJMZ23], which allows us to reduce the security of the many-query game to the security of the one-query game.

Construction 1. Obf *takes as input a quantum implementation* (ρ, Eval) *of some function* f*, where* ρ *is a state on a register* $\mathcal{A}$*. We assume without loss of generality that the circuit* Eval *consists of a unitary on* $\mathcal{A}$ *as well as an input register* $\mathcal{X}$ *and an output register* $\mathcal{Y}$*, followed by a measurement of register* $\mathcal{Y}$*. For ease of notation, we will identify the algorithm* Eval *(which includes a measurement) with its unitary part when it is clear from the context. We assume without loss of generality that the unitary* Eval *uncomputes all of its intermediate steps, leaving the result on* $\mathcal{Y}$*.*

$\mathsf{Obf}(\rho, U)$ *proceeds as follows:*

- *Sample* $C \leftarrow \mathcal{C}$*, where* $\mathcal{C}$ *is the Clifford group. Let* $\tilde{\rho} = C\rho C^\dagger$*.*
- *Let* G_C *be the unitary acting on registers* $\mathcal{A}, \mathcal{X}, \mathcal{Y}$ *defined as* $G_C = (C_\mathcal{A} \otimes I_{\mathcal{X}\mathcal{Y}})\,\mathsf{Eval}(C^\dagger_\mathcal{A} \otimes I_{\mathcal{X}\mathcal{Y}})$*.*
- *Let* $\tilde{U}^{G_C}$ *be the quantum circuit, with oracle access to* G_C*, that behaves as follows: On input* ρ*, run* G_C *on input* $\rho_\mathcal{A} \otimes |x\rangle\langle x|_\mathcal{X} \otimes |0\rangle\langle 0|_\mathcal{Y}$*; measure register* $\mathcal{Y}$ *and output the outcome.*
- *Output* $(\tilde{\rho}, \tilde{U}^{G_C})$ *(since this is an oracle construction, what we mean is that the algorithm* Obf *outputs the description of the oracle algorithm* $\tilde{U}$*, and the oracle* G_C *is publicly available).*

We show that Construction 1 is qsiO in a model where the adversary has access to the oracle G_C.

Theorem 2. *Construction 1 is* qsiO*.*

Proof. We prove security via three hybrids. The first hybrid corresponds to the original qsiO security game. The second corresponds to a "purified" version of the qsiO game, which is easily seen to be equivalent to the original. The third hybrid is identical to the second, except that the adversary has access to a different oracle: This new oracle does not evaluate the function unless the register containing C is in uniform superposition.

Finally, we show that the distinguishing advantage in the third hybrid is *zero* by invoking the "admissible oracle lemma" of [GJMZ23].

Hybrid 1: The original qsiO security game.

Hybrid 2: A "purified" version of the qsiO game. Let $(|\psi_1\rangle, U_1)$ and $(|\psi_2\rangle, U_2)$ be two quantum implementations of the same classical functionality $f : \{0,1\}^{l_{in}} \to \{0,1\}^{l_{out}}$, where $|\psi_1\rangle$ and $|\psi_2\rangle$ are states on some register $\mathcal{B}_1$, and U_1, U_2 are unitaries on $\mathcal{B}_1$ and some other register $\mathcal{R}$. For the rest of the proof we assume, for simplicity and without loss of generality, that the unitaries U_1 and U_2 are equal to a fixed universal unitary $\mathsf{Eval}_{\mathcal{B}_1,\mathcal{R}}$.

Let $\mathcal{A}, \mathcal{B}_1, \mathcal{B}_2, \mathcal{R}$ be registers, and let Π' be the subspace spanned by all the states of the form

$$|C\rangle_\mathcal{A} \otimes C(\,|\psi\rangle_{\mathcal{B}_1} \otimes\, |0^\lambda\rangle_{\mathcal{B}_2}) \otimes |x, y\rangle_\mathcal{R}\ , \tag{8}$$

where C is any Clifford unitary on register $\mathcal{B} := \mathcal{B}_1\,\mathcal{B}_2$, $|\psi\rangle$ is any state on $\mathcal{B}_1$, and $(x, y) \in \{0,1\}^{l_{in}} \times \{0,1\}^{l_{out}}$.

The unitary G' acts as identity on the orthogonal complement of Π', and as follows on Π':

$$G'\Pi' = \sum_{C\in\mathcal{C}} |C\rangle\langle C|_\mathcal{A} \otimes (C_\mathcal{B} \otimes I_\mathcal{R})(\mathsf{Eval}_{\mathcal{B}_1,\mathcal{R}} \otimes I_{\mathcal{B}_2})(C^\dagger_\mathcal{B} \otimes I_\mathcal{R}).$$

In the rest of the section, when it is clear from the context, we omit writing tensor products with identities, e.g. we write

$$G'\Pi' = \sum_{C\in\mathcal{C}} |C\rangle\langle C|_\mathcal{A} \otimes C_\mathcal{B}(\mathsf{Eval}_{\mathcal{B}_1,\mathcal{R}})C^\dagger_\mathcal{B}.$$

The game is as follows:

1. The challenger samples $b \leftarrow \{0,1\}$. Then, it creates the state $|\mathcal{C}|^{-1/2} \sum_{C\in\mathcal{C}} |C\rangle_\mathcal{A} \otimes C(\,|\psi_b\rangle \otimes\, |0^\lambda\rangle)_\mathcal{B}$.

2. The adversary receives register $\mathcal{B}$ from the challenger, as well as query access to the oracle G (where the adversary controls $\mathcal{R}$). The adversary returns a guess $b' \in \{0,1\}$.

The adversary wins if $b' = b$.

Hybrid 3: Identical to Hybrid 2, except the adversary has access to a different oracle G defined as follows.

- Let Π be the subspace spanned by all the states of the form
$$|\mathcal{C}|^{-1/2} \sum_{C \in \mathcal{C}} |C\rangle \otimes C(\,|\psi\rangle \otimes \,|0^\lambda\rangle) \otimes |x,y\rangle \ , \tag{9}$$
where $|\psi\rangle$ is any state on $\mathcal{B}_1$ and $(x,y) \in \{0,1\}^{l_{in}} \times \{0,1\}^{l_{out}}$. The unitary G acts as identity on the orthogonal complement of Π, and as follows on Π:
$$G\Pi = \sum_{C \in \mathcal{C}} |C\rangle\langle C| \otimes C(\mathsf{Eval})C^\dagger$$

We first show that the adversary's advantage in Hybrid 1 and Hybrid 2 is identical.

Lemma 3. *For any adversary A,*
$$\Pr[A \text{ wins in Hybrid 1}] = \Pr[A \text{ wins in Hybrid 2}]\,.$$

Proof. This is immediate since Hybrid 2 is just a purification of Hybrid 1. □

Lemma 4. *For any adversary A for Hybrids 2 and 3, there exists a negligible function* negl *such that, for all λ,*
$$\big|\Pr[A \text{ wins in Hybrid 2}] - \Pr[A \text{ wins in Hybrid 3}]\big| \leq \mathsf{negl}(\lambda)\,.$$

Proof. Let $W_\mathcal{C} = \sum_{C \in \mathcal{C}} |C\rangle\langle C| \otimes C$. Define
$$\mathsf{Eval}_1 := \mathsf{Eval} \cdot \big(I_{\mathcal{A}\,\mathcal{B}_1\,\mathcal{R}} \otimes \big|0^\lambda\big\rangle\big\langle 0^\lambda\big|_{\mathcal{B}_2}\,\big) + I_{\mathcal{A}\,\mathcal{B}_1\,\mathcal{R}} \otimes (I - \big|0^\lambda\big\rangle\big\langle 0^\lambda\big|_{\mathcal{B}_2})\,.$$
Then, notice that we can write G' (from Hybrid 2) as
$$G' = W_\mathcal{C}\,\mathsf{Eval}_1(W_\mathcal{C})^\dagger\,. \tag{10}$$

Let $|\tau\rangle = |\,\mathcal{C}\,|^{-1/2} \sum_{C \in \mathcal{C}} |C\rangle$, and define
$$\mathsf{Eval}_2 := \mathsf{Eval} \cdot \big(\,|\tau\rangle\langle\tau|_\mathcal{A} \otimes I_{\mathcal{B}_1\,\mathcal{R}} \otimes \big|0^\lambda\big\rangle\big\langle 0^\lambda\big|_{\mathcal{B}_2}\,\big) + (I - |\tau\rangle\langle\tau|_\mathcal{A}) \otimes I_{\mathcal{B}_1\,\mathcal{R}} \otimes \big|0^\lambda\big\rangle\big\langle 0^\lambda\big|_{\mathcal{B}_2} + I_{\mathcal{A}\,\mathcal{B}_1\,\mathcal{R}} \otimes (I - \big|0^\lambda\big\rangle\big\langle 0^\lambda\big|_{\mathcal{B}_2})\,.$$
We can write G (from Hybrid 3) as
$$G = W_\mathcal{C}\,\mathsf{Eval}_2(W_\mathcal{C})^\dagger\,. \tag{11}$$

Let A be an adversary for Hybrid 2 and 3. Recall that Hybrids 2 and 3 are identical except that the oracle is G' in Hybrid 2 and G in Hybrid 3. Recall that the challenger initializes registers $\mathcal{A}, \mathcal{B}$ in the state
$$|\Psi_0\rangle := |\mathcal{C}|^{-1/2} \sum_{C \in \mathcal{C}} |C\rangle_\mathcal{A} \otimes C(\,|\psi_b\rangle \otimes \,|0^\lambda\rangle)_\mathcal{B}\,. \tag{12}$$
for some $b \in \{0,1\}$ Then A then receives register $\mathcal{B}$. Let $\mathcal{R}$ denote the register where Eval writes its output, and let $\mathcal{Z}$ denote an additional work register used by A (which includes an output register). Let q be the number of queries to the oracle (G' or G) made by A. Without loss of generality, for some unitary U on $\mathcal{B}\,\mathcal{R}\,\mathcal{Z}$, we have that A applies the sequence of unitaries $(G'U)^q$ in Hybrid 2, and $(GU)^q$ in Hybrid 3.

We will prove Lemma 5, which implies that A's success probability in Hybrids 2 and 3 is negligibly close, as long as $q = \mathsf{poly}(\lambda)$. This will complete the proof of Lemma 4. □

Lemma 5. *Let $|\Psi_0\rangle_{\mathcal{A}\mathcal{B}}$ be as defined in (12) (note that this state depends on λ). Let $|\phi\rangle_{\mathcal{R}\mathcal{Z}}$ be any state, and U any unitary (both of which implicitly depend on λ), and let $q \in \mathbb{N}$. Then, for all λ,*

$$\| (G'U)^q |\Psi_0\rangle \otimes |\phi\rangle - (GU)^q |\Psi_0\rangle \otimes |\phi\rangle \| = q(q+1)2^{-\lambda} . \tag{13}$$

Proof. For convenience, we use the following notation throughout this proof: for $\epsilon > 0$ and states $|u\rangle$ and $|v\rangle$, we write $|u\rangle \equiv_{\epsilon} |v\rangle$ as a shorthand for $\| |u\rangle - |v\rangle \| \leq \epsilon$. Let m be the number of qubits in register $\mathcal{B}_1$. We prove Lemma 5 by induction on the number of queries. Precisely, we show that, for all $i \in \{0, \dots, q\}$:

(i) There exist unnormalized states $|\phi_{x,z}\rangle_{\mathcal{R}\mathcal{Z}}$ for $x, z \in \{0,1\}^{m+\lambda}$ such that, for all λ,

$$(G'U)^i |\Psi_0\rangle \equiv_{i \cdot 2^{-\lambda}} |\mathcal{C}|^{-1/2} \sum_{C \in \mathcal{C}} \sum_{x,z \in \{0,1\}^{m+\lambda}} |C\rangle_{\mathcal{A}} \otimes X^x Z^z C(|\psi_b\rangle \otimes |0^\lambda\rangle)_{\mathcal{B}} \otimes |\phi_{x,z}\rangle .$$

(ii) For all λ,

$$(G'U)^i |\Psi_0\rangle \otimes |\phi\rangle \equiv_{i(i+1) \cdot 2^{-\lambda}} (GU)^i |\Psi_0\rangle \otimes |\phi\rangle .$$

Clearly, both (i) and (ii) hold for $i = 0$. Now, suppose (i) and (ii) hold for some i. We show that they both hold also for $i+1$. By the inductive hypothesis, we have

$$(G'U)^i |\Psi_0\rangle \otimes |\phi\rangle \equiv_{i \cdot 2^{-\lambda}} |\mathcal{C}|^{-1/2} \sum_{C \in \mathcal{C}} \sum_{x,z \in \{0,1\}^{m+\lambda}} |C\rangle_{\mathcal{A}} \otimes X^x Z^z C(|\psi_b\rangle \otimes |0^\lambda\rangle)_{\mathcal{B}} \otimes |\phi_{x,z}\rangle ,$$

for some unnormalized states $|\phi_{x,y}\rangle_{\mathcal{R}\mathcal{Z}}$ for $x, z \in \{0,1\}^{m+\lambda}$. Then, we have

$$(W_{\mathcal{C}})^\dagger U (G'U)^i |\Psi_0\rangle \otimes |\phi\rangle \equiv_{i \cdot 2^{-\lambda}} (W_{\mathcal{C}})^\dagger U |\mathcal{C}|^{-1/2} \sum_{C \in \mathcal{C}} \sum_{x,z \in \{0,1\}^{m+\lambda}} |C\rangle_{\mathcal{A}} \otimes X^x Z^z C(|\psi_b\rangle \otimes |0^\lambda\rangle)_{\mathcal{B}} \otimes |\phi_{x,z}\rangle . \tag{14}$$

By expanding the $\mathcal{B}$ register in the Pauli basis, we can write $U = \sum_{x,y \in \{0,1\}^{m+\lambda}} X^x Z^z \otimes U_{xz}$ for some operators U_{xz}. Then, plugging this into (14) we get

$$\begin{aligned}
(W_{\mathcal{C}})^\dagger U (G'U)^i |\Psi_0\rangle \otimes |\phi\rangle &\equiv_{i \cdot 2^{-\lambda}} (W_{\mathcal{C}})^\dagger |\mathcal{C}|^{-1/2} \sum_{x',z'} \sum_{C \in \mathcal{C}} \sum_{x,z} |C\rangle_{\mathcal{A}} \otimes X^{x'} Z^{z'} X^x Z^z C(|\psi_b\rangle \otimes |0^\lambda\rangle)_{\mathcal{B}} \otimes U_{x',z'} |\phi_{x,z}\rangle \\
&= (W_{\mathcal{C}})^\dagger |\mathcal{C}|^{-1/2} \sum_{C \in \mathcal{C}} \sum_{x,z,x',z'} |C\rangle_{\mathcal{A}} \otimes (-1)^{z' \cdot x} X^{x+x'} Z^{z+z'} C(|\psi_b\rangle \otimes |0^\lambda\rangle)_{\mathcal{B}} \otimes U_{x',z'} |\phi_{x,z}\rangle \\
&= |\mathcal{C}|^{-1/2} \sum_{C \in \mathcal{C}} \sum_{x,z,x',z'} |C\rangle_{\mathcal{A}} \otimes (-1)^{z' \cdot x} C^\dagger X^{x+x'} Z^{z+z'} C(|\psi_b\rangle \otimes |0^\lambda\rangle)_{\mathcal{B}} \otimes U_{x',z'} |\phi_{x,z}\rangle \\
&= |\mathcal{C}|^{-1/2} \sum_{C \in \mathcal{C}} |C\rangle_{\mathcal{A}} \otimes (|\psi_b\rangle \otimes |0^\lambda\rangle)_{\mathcal{B}} \otimes |\tilde{\phi}_{0,0}\rangle \\
&\quad + |\mathcal{C}|^{-1/2} \sum_{C \in \mathcal{C}} |C\rangle_{\mathcal{A}} \otimes \sum_{(x,z) \neq (0^{m+\lambda}, 0^{m+\lambda})} C^\dagger X^x Z^z C(|\psi_b\rangle \otimes |0^\lambda\rangle)_{\mathcal{B}} \otimes |\tilde{\phi}_{x,z}\rangle ,
\end{aligned} \tag{15}$$

for some unnormalized states $|\tilde{\phi}_{x,z}\rangle$. We will show that the second summand in the last expression has exponentially small weight on states such that the state on register $\mathcal{B}_2$ is $|0\rangle^\lambda$. Precisely, we will show that

$$\left\| I_{\mathcal{A}\mathcal{B}_1\mathcal{R}} \otimes |0^\lambda\rangle\langle 0^\lambda|_{\mathcal{B}_2} \; |\mathcal{C}|^{-1/2} \sum_{C \in \mathcal{C}} |C\rangle_{\mathcal{A}} \otimes \sum_{(x,z) \neq (0^{m+\lambda}, 0^{m+\lambda})} C^\dagger X^x Z^z C(|\psi_b\rangle \otimes |0^\lambda\rangle)_{\mathcal{B}} \otimes |\tilde{\phi}_{x,z}\rangle \right\| \leq 2^{-\lambda} . \tag{16}$$

We will prove (16) at the end. Now, recall that a Clifford operator is uniquely specified by the fact that it maps Pauli operators to Pauli operators when acting by conjugation. For $C \in \mathcal{C}$ and $x, z \in \{0,1\}^{m+\lambda}$, let $\pi_C^X(x,z) \in \{0,1\}^\lambda$ be defined such that

$$(X^{\tilde{x}_1} \otimes X^{\pi_C^X(x,z)}) Z^{\tilde{z}} = C^\dagger X^x Z^z C .$$

for some $\tilde{x}_1 \in \{0,1\}^m, \tilde{z} \in \{0,1\}^{m+\lambda}$. In other words, $\pi_C^X(x,z)$ corresponds to the *last* λ *bits* of the Pauli X string obtained by conjugating $X^x Z^z$ by C.

Assuming (16) is true, we have

$$\left\| |\mathcal{C}|^{-1/2} \sum_{C\in\mathcal{C}} |C\rangle_{\mathcal{A}} \otimes \sum_{\substack{(x,z)\neq(0^{m+\lambda},0^{m+\lambda}):\\ \pi_C^X(x,z)=0^\lambda}} C^\dagger X^x Z^z C(\,|\psi_b\rangle \otimes |0^\lambda\rangle)_{\mathcal{B}} \otimes |\tilde{\phi}_{x,z}\rangle \right\| \leq 2^{-\lambda}\,. \tag{17}$$

Then, we have

$$\begin{aligned}
&\mathsf{Eval}_1(W_{\mathcal{C}})^\dagger U (G'U)^i\,|\Psi_0\rangle \otimes |\phi\rangle \\
&\equiv_{i\cdot 2^{-\lambda}+2^{-\lambda}} \mathsf{Eval}_1\Bigg(|\mathcal{C}|^{-1/2} \sum_{C\in\mathcal{C}} |C\rangle_{\mathcal{A}} \otimes (\,|\psi_b\rangle \otimes |0^\lambda\rangle)_{\mathcal{B}} \otimes |\tilde{\phi}_{0,0}\rangle \\
&\qquad\qquad + |\mathcal{C}|^{-1/2} \sum_{C\in\mathcal{C}} |C\rangle_{\mathcal{A}} \otimes \sum_{\substack{(x,z)\neq(0^{m+\lambda},0^{m+\lambda}):\\ \pi_C^X(x,z)\neq 0^\lambda}} C^\dagger X^x Z^z C(\,|\psi_b\rangle \otimes |0^\lambda\rangle)_{\mathcal{B}} \otimes |\tilde{\phi}_{x,z}\rangle \Bigg) \qquad (18)\\
&= \mathsf{Eval}\ |\mathcal{C}|^{-1/2} \sum_{C\in\mathcal{C}} |C\rangle_{\mathcal{A}} \otimes (\,|\psi_b\rangle \otimes |0^\lambda\rangle)_{\mathcal{B}} \otimes |\tilde{\phi}_{0,0}\rangle \\
&\quad + |\mathcal{C}|^{-1/2} \sum_{C\in\mathcal{C}} |C\rangle_{\mathcal{A}} \otimes \sum_{\substack{(x,z)\neq(0^{m+\lambda},0^{m+\lambda}):\\ \pi_C^X(x,z)\neq 0^\lambda}} C^\dagger X^x Z^z C(\,|\psi_b\rangle \otimes |0^\lambda\rangle)_{\mathcal{B}} \otimes |\tilde{\phi}_{x,z}\rangle\,, \qquad (19)
\end{aligned}$$

where (18) follows from (15), (17), unitarity of Eval_1, and a triangle inequality; and (19) follows from the definition of Eval_1.

Notice that, by definition of Eval,

$$\mathsf{Eval}(\,|\psi_b\rangle \otimes |0^\lambda\rangle)_{\mathcal{B}} \otimes |\tilde{\phi}_{0,0}\rangle = (\,|\psi_b\rangle \otimes |0^\lambda\rangle)_{\mathcal{B}} \otimes |\tilde{\tilde{\phi}}_{0,0}\rangle\,,$$

for some other state $|\tilde{\tilde{\phi}}_{0,0}\rangle$. So, plugging this into (19), gives

$$\begin{aligned}
\mathsf{Eval}_1(W_{\mathcal{C}})^\dagger U (G'U)^i\,|\Psi_0\rangle \otimes |\phi\rangle &\equiv_{(i+1)\cdot 2^{-\lambda}} |\mathcal{C}|^{-1/2} \sum_{C\in\mathcal{C}} |C\rangle_{\mathcal{A}} \otimes (\,|\psi_b\rangle \otimes |0^\lambda\rangle)_{\mathcal{B}} \otimes |\tilde{\tilde{\phi}}_{0,0}\rangle \\
&+ |\mathcal{C}|^{-1/2} \sum_{C\in\mathcal{C}} |C\rangle_{\mathcal{A}} \otimes \sum_{\substack{(x,z)\neq(0^{m+\lambda},0^{m+\lambda}):\\ \pi_C^X(x,z)\neq 0^\lambda}} C^\dagger X^x Z^z C(\,|\psi_b\rangle \otimes |0^\lambda\rangle)_{\mathcal{B}} \otimes |\tilde{\phi}_{x,z}\rangle\,.
\end{aligned}$$

Applying $W_{\mathcal{C}}$ to both sides, we get, by the unitarity of $W_{\mathcal{C}}$ and using its definition,

$$\begin{aligned}
W_{\mathcal{C}}\,\mathsf{Eval}_1(W_{\mathcal{C}})^\dagger U (G'U)^i\,|\Psi_0\rangle \otimes |\phi\rangle &\equiv_{(i+1)\cdot 2^{-\lambda}} |\mathcal{C}|^{-1/2} \sum_{C\in\mathcal{C}} |C\rangle_{\mathcal{A}} \otimes C(\,|\psi_b\rangle \otimes |0^\lambda\rangle)_{\mathcal{B}} \otimes |\tilde{\tilde{\phi}}_{0,0}\rangle \\
&+ |\mathcal{C}|^{-1/2} \sum_{C\in\mathcal{C}} |C\rangle_{\mathcal{A}} \otimes \sum_{\substack{(x,z)\neq(0^{m+\lambda},0^{m+\lambda}):\\ \pi_C^X(x,z)\neq 0^\lambda}} X^x Z^z C(\,|\psi_b\rangle \otimes |0^\lambda\rangle)_{\mathcal{B}} \otimes |\tilde{\phi}_{x,z}\rangle\,.
\end{aligned}$$

Plugging $G' = W_{\mathcal{C}}\,\mathsf{Eval}_1(W_{\mathcal{C}}^\dagger)$ into the above gives

$$\begin{aligned}
(G'U)^{i+1}\,|\Psi_0\rangle \otimes |\phi\rangle &\equiv_{(i+1)\cdot 2^{-\lambda}} |\mathcal{C}|^{-1/2} \sum_{C\in\mathcal{C}} |C\rangle_{\mathcal{A}} \otimes C(\,|\psi_b\rangle \otimes |0^\lambda\rangle)_{\mathcal{B}} \otimes |\tilde{\tilde{\phi}}_{0,0}\rangle \\
&+ |\mathcal{C}|^{-1/2} \sum_{C\in\mathcal{C}} |C\rangle_{\mathcal{A}} \otimes \sum_{\substack{(x,z)\neq(0^{m+\lambda},0^{m+\lambda}):\\ \pi_C^X(x,z)\neq 0^\lambda}} X^x Z^z C(\,|\psi_b\rangle \otimes |0^\lambda\rangle)_{\mathcal{B}} \otimes |\tilde{\phi}_{x,z}\rangle\,, \qquad (20)
\end{aligned}$$

which establishes item (i) of the inductive step. Next, we establish item (ii). From (19), we know that

$$\begin{aligned}
&\mathsf{Eval}_1(W_{\mathcal{C}})^{\dagger}U(G'U)^{i}\left|\Psi_0\right\rangle \otimes \left|\phi\right\rangle \\
&\equiv_{(i+1)\cdot 2^{-\lambda}} \mathsf{Eval}\ |\mathcal{C}|^{-1/2}\sum_{C\in\mathcal{C}} \left|C\right\rangle_{\mathcal{A}} \otimes \left(\,\left|\psi_b\right\rangle \otimes \left|0^{\lambda}\right\rangle\right)_{\mathcal{B}} \otimes \left|\tilde{\phi}_{0,0}\right\rangle \\
&\quad + |\mathcal{C}|^{-1/2}\sum_{C\in\mathcal{C}} \left|C\right\rangle_{\mathcal{A}} \otimes \sum_{\substack{(x,z)\neq(0^{m+\lambda},0^{m+\lambda}):\\ \pi_C^X(x,z)\neq 0^{\lambda}}} C^{\dagger}X^{x}Z^{z}C(\,\left|\psi_b\right\rangle \otimes \left|0^{\lambda}\right\rangle)_{\mathcal{B}} \otimes \left|\tilde{\phi}_{x,z}\right\rangle . \qquad (21)
\end{aligned}$$

Then, we have

$$\begin{aligned}
\text{RHS of (21)} &= \mathsf{Eval}\left(\,\left|\tau\right\rangle\!\left\langle\tau\right|_{\mathcal{A}} \otimes I_{\mathcal{B}_1\,\mathcal{R}} \otimes \left|0^{\lambda}\right\rangle\!\left\langle 0^{\lambda}\right|_{\mathcal{B}_2}\right) |\mathcal{C}|^{-1/2}\sum_{C\in\mathcal{C}} \left|C\right\rangle_{\mathcal{A}} \otimes \left(\,\left|\psi_b\right\rangle \otimes \left|0^{\lambda}\right\rangle\right)_{\mathcal{B}} \otimes \left|\tilde{\phi}_{0,0}\right\rangle \\
&\quad + |\mathcal{C}|^{-1/2}\sum_{C\in\mathcal{C}} \left|C\right\rangle_{\mathcal{A}} \otimes \sum_{\substack{(x,z)\neq(0^{m+\lambda},0^{m+\lambda}):\\ \pi_C^X(x,z)\neq 0^{\lambda}}} C^{\dagger}X^{x}Z^{z}C(\,\left|\psi_b\right\rangle \otimes \left|0^{\lambda}\right\rangle)_{\mathcal{B}} \otimes \left|\tilde{\phi}_{x,z}\right\rangle \qquad (22) \\
&= \mathsf{Eval}_2\Bigg(|\mathcal{C}|^{-1/2}\sum_{C\in\mathcal{C}} \left|C\right\rangle_{\mathcal{A}} \otimes \left(\,\left|\psi_b\right\rangle \otimes \left|0^{\lambda}\right\rangle\right)_{\mathcal{B}} \otimes \left|\tilde{\phi}_{0,0}\right\rangle \\
&\qquad + |\mathcal{C}|^{-1/2}\sum_{C\in\mathcal{C}} \left|C\right\rangle_{\mathcal{A}} \otimes \sum_{\substack{(x,z)\neq(0^{m+\lambda},0^{m+\lambda}):\\ \pi_C^X(x,z)\neq 0^{\lambda}}} C^{\dagger}X^{x}Z^{z}C(\,\left|\psi_b\right\rangle \otimes \left|0^{\lambda}\right\rangle)_{\mathcal{B}} \otimes \left|\tilde{\phi}_{x,z}\right\rangle\Bigg) \\
&\qquad (23) \\
&\equiv_{(i+1)\cdot 2^{-\lambda}} \mathsf{Eval}_2(W_{\mathcal{C}})^{\dagger}U(G'U)^{i}\left|\Psi_0\right\rangle \otimes \left|\phi\right\rangle\ , \qquad (24)
\end{aligned}$$

where (22) follows from the definition of $\left|\tau\right\rangle$; (23) from the definition of Eval_2; and (24) follows from the identical reasons as (18).

Overall, this gives, by a triangle inequality,

$$\mathsf{Eval}_1(W_{\mathcal{C}})^{\dagger}U(G'U)^{i}\left|\Psi_0\right\rangle \otimes \left|\phi\right\rangle \equiv_{2(i+1)\cdot 2^{-\lambda}} \mathsf{Eval}_2(W_{\mathcal{C}})^{\dagger}U(G'U)^{i}\left|\Psi_0\right\rangle \otimes \left|\phi\right\rangle\ . \qquad (25)$$

Hence, we have

$$\begin{aligned}
(G'U)^{i+1}\left|\Psi_0\right\rangle \otimes \left|\phi\right\rangle &= G'U(G'U)^{i}\left|\Psi_0\right\rangle \otimes \left|\phi\right\rangle \\
&= W_{\mathcal{C}}\,\mathsf{Eval}_1\,W_{\mathcal{C}}^{\dagger}U(G'U)^{i}\left|\Psi_0\right\rangle \otimes \left|\phi\right\rangle \qquad (26) \\
&\equiv_{2(i+1)\cdot 2^{-\lambda}} W_{\mathcal{C}}\,\mathsf{Eval}_2\,W_{\mathcal{C}}^{\dagger}U(G'U)^{i}\left|\Psi_0\right\rangle \otimes \left|\phi\right\rangle \qquad (27) \\
&= GU(G'U)^{i}\left|\Psi_0\right\rangle \otimes \left|\phi\right\rangle \qquad (28) \\
&\equiv_{i\cdot(i+1)\cdot 2^{-\lambda}} GU(GU)^{i}\left|\Psi_0\right\rangle \otimes \left|\phi\right\rangle \qquad (29) \\
&= (GU)^{i+1}\left|\Psi_0\right\rangle \otimes \left|\phi\right\rangle\ ,
\end{aligned}$$

where (26) is due to (10); (27) is due to (25); (28) is due to (11); and (29) is by the inductive hypothesis. Overall, by a triangle inequality, we have

$$(G'U)^{i+1}\left|\Psi_0\right\rangle \otimes \left|\phi\right\rangle \equiv_{2(i+1)\cdot 2^{-\lambda}+i\cdot(i+1)\cdot 2^{-\lambda}} (GU)^{i+1}\left|\Psi_0\right\rangle \otimes \left|\phi\right\rangle\ ,$$

which simplifies to

$$(G'U)^{i+1}\left|\Psi_0\right\rangle \otimes \left|\phi\right\rangle \equiv_{(i+1)(i+2)\cdot 2^{\lambda}} (GU)^{i+1}\left|\Psi_0\right\rangle \otimes \left|\phi\right\rangle\ ,$$

This establishes exactly item (ii) of the inductive hypothesis, and hence Lemma 5 (assuming Equation (16)).

To conclude the proof of Lemma 5, we are left with proving Equation (16). We will make use of the "Clifford twirl."

Lemma 6 (Clifford twirl [ABOEM17])**.** *Let $n \in \mathbb{N}$. Let $|\Psi\rangle$ be any state on n qubits. Let $x, z, x', z' \in \{0,1\}^n$ such that $(x,z) \neq (x',z')$. Let $\mathcal{C}$ be the Clifford group on n qubits. Then,*

$$\sum_{C\in\mathcal{C}} C^\dagger X^{x'} Z^{z'} C \,|\Psi\rangle\langle\Psi|\, C^\dagger X^x Z^z C = 0\,.$$

We have

$$\begin{align}
&\Big\| I_{\mathcal{A}\mathcal{B}_1\mathcal{R}} \otimes |0^\lambda\rangle\langle 0^\lambda|_{\mathcal{B}_2} \ |\mathcal{C}|^{-1/2} \sum_{C\in\mathcal{C}} |C\rangle_{\mathcal{A}} \otimes \sum_{(x,z)\neq(0^{m+\lambda},0^{m+\lambda})} C^\dagger X^x Z^z C(\,|\psi_b\rangle \otimes\, |0^\lambda\rangle)_{\mathcal{B}} \otimes |\tilde{\phi}_{x,z}\rangle \Big\|^2 \notag\\
&= \frac{1}{|\,\mathcal{C}\,|} \sum_{C\in\mathcal{C}} \big\| (I \otimes |0^\lambda\rangle\langle 0^\lambda|) \sum_{(x,z)\neq(0^{m+\lambda},0^{m+\lambda})} C^\dagger X^x Z^z C(\,|\psi_b\rangle \otimes\, |0^\lambda\rangle)_{\mathcal{B}} \otimes |\tilde{\phi}_{x,z}\rangle \big\|^2 \notag\\
&= \frac{1}{|\,\mathcal{C}\,|} \sum_{C\in\mathcal{C}} \sum_{(x,z)\neq(0^{m+\lambda},0^{m+\lambda})} \big\| (I \otimes |0^\lambda\rangle\langle 0^\lambda|) C^\dagger X^x Z^z C(\,|\psi_b\rangle \otimes\, |0^\lambda\rangle)_{\mathcal{B}} \otimes |\tilde{\phi}_{x,z}\rangle \big\|^2 \notag\\
&+ \frac{1}{|\,\mathcal{C}\,|} \sum_{C\in\mathcal{C}} \sum_{(x,z)\neq(x',z')\neq(0^{m+\lambda},0^{m+\lambda})} \langle\psi|\, \langle 0^\lambda|\, C^\dagger X^x Z^z C (I \otimes |0^\lambda\rangle\langle 0^\lambda|) C^\dagger X^{x'} Z^{z'} C\,|\psi\rangle\, |0^\lambda\rangle \cdot \langle \tilde{\phi}_{x,z} | \tilde{\phi}_{x',z'} \rangle \notag\\
&= \frac{1}{|\,\mathcal{C}\,|} \sum_{C\in\mathcal{C}} \sum_{(x,z)\neq(0^{m+\lambda},0^{m+\lambda})} \big\| |\tilde{\phi}_{x,z}\rangle \big\|^2 \notag\\
&+ \frac{1}{|\,\mathcal{C}\,|} \sum_{C\in\mathcal{C}} \sum_{(x,z)\neq(x',z')\neq(0^{m+\lambda},0^{m+\lambda})} \mathrm{Tr}\Big[(I \otimes |0^\lambda\rangle\langle 0^\lambda|) C^\dagger X^{x'} Z^{z'} C (|\psi\rangle\langle\psi| \otimes |0^\lambda\rangle\langle 0^\lambda|) C^\dagger X^x Z^z C \Big] \notag\\
&= \sum_{(x,z)\neq(0^{m+\lambda},0^{m+\lambda})} \big\| |\tilde{\phi}_{x,z}\rangle \big\|^2 \frac{1}{|\,\mathcal{C}\,|} \sum_{\substack{C\in\mathcal{C}: \\ \pi_C^X(x,z)\neq 0^\lambda}} 1 \notag\\
&+ \sum_{(x,z)\neq(x',z')\neq(0^{m+\lambda},0^{m+\lambda})} \mathrm{Tr}\Big[(I \otimes |0^\lambda\rangle\langle 0^\lambda|) \frac{1}{|\,\mathcal{C}\,|} \sum_{C\in\mathcal{C}} C^\dagger X^{x'} Z^{z'} C (|\psi\rangle\langle\psi| \otimes |0^\lambda\rangle\langle 0^\lambda|) C^\dagger X^x Z^z C \Big] \notag\\
&= \sum_{(x,z)\neq(0^{m+\lambda},0^{m+\lambda})} \big\| |\tilde{\phi}_{x,z}\rangle \big\|^2 \frac{1}{|\,\mathcal{C}\,|} \sum_{\substack{C\in\mathcal{C}: \\ \pi_C^X(x,z)\neq 0^\lambda}} 1 \tag{30}\\
&= \sum_{(x,z)\neq(0^{m+\lambda},0^{m+\lambda})} \big\| |\tilde{\phi}_{x,z}\rangle \big\|^2 \cdot 2^{-\lambda}\,, \tag{31}
\end{align}$$

where (30) follows from the Clifford twirl (Lemma 6), and (31) follows from the fact that for any $(x,z) \neq (0^{m+\lambda}, 0^{m+\lambda})$, the fraction of $C \in \mathcal{C}$ such that $\pi_C^X(x,z) = 0^\lambda$ is exactly $2^{-\lambda}$. We claim that

$$\sum_{(x,z)} \big\| |\tilde{\phi}_{x,z}\rangle \big\|^2 = 1\,.$$

Assuming this is the case, we have

$$\Big\| I_{\mathcal{A}\mathcal{B}_1\mathcal{R}} \otimes |0^\lambda\rangle\langle 0^\lambda|_{\mathcal{B}_2} \ |\mathcal{C}|^{-1/2} \sum_{C\in\mathcal{C}} |C\rangle_{\mathcal{A}} \otimes \sum_{(x,z)\neq(0^{m+\lambda},0^{m+\lambda})} C^\dagger X^x Z^z C(\,|\psi_b\rangle \otimes |0^\lambda\rangle)_{\mathcal{B}} \otimes |\tilde{\phi}_{x,z}\rangle \Big\|^2 \leq 2^{-\lambda}\,, \tag{32}$$

as desired. We now prove the claim. The calculation is similar to the we just performed. We have

$$\begin{aligned}
1 &= \left\| |\mathcal{C}|^{-1/2} \sum_{C \in \mathcal{C}} |C\rangle_{\mathcal{A}} \otimes \sum_{(x,z)} C^\dagger X^x Z^z C(\,|\psi_b\rangle \otimes |0^\lambda\rangle)_{\mathcal{B}} \otimes |\tilde{\phi}_{x,z}\rangle \right\|^2 \\
&= \frac{1}{|\,\mathcal{C}\,|} \sum_{C \in \mathcal{C}} \sum_{x,z} \left\| |\tilde{\phi}_{x,z}\rangle \right\|^2 \\
&+ \frac{1}{|\,\mathcal{C}\,|} \sum_{C \in \mathcal{C}} \sum_{(x,z) \neq (x',z')} \langle\psi| \left\langle 0^\lambda \right| C^\dagger X^x Z^z X^{x'} Z^{z'} C^\dagger \,|\psi\rangle \left|0^\lambda\right\rangle \cdot \langle \tilde{\phi}_{x,z} | \tilde{\phi}_{x',z'} \rangle \\
&= \sum_{x,z} \left\| |\tilde{\phi}_{x,z}\rangle \right\|^2 \\
&+ \frac{1}{|\,\mathcal{C}\,|} \sum_{C \in \mathcal{C}} \sum_{(x,z) \neq (x',z')} \mathrm{Tr}\Big[\frac{1}{|\,\mathcal{C}\,|} \sum_{C \in \mathcal{C}} C^\dagger X^{x'} Z^{z'} C(|\psi\rangle\!\langle\psi| \otimes \left|0^\lambda\right\rangle\!\left\langle 0^\lambda\right|) C^\dagger X^x Z^z C \Big] \\
&= \sum_{x,z} \left\| |\tilde{\phi}_{x,z}\rangle \right\|^2 ,
\end{aligned}$$

where the last line follows again by the Clifford twirl. This concludes the proof of Lemma 5. □

Lemma 7. *For any adversary A for Hybrid 3,*

$$\Pr[A \text{ wins in Hybrid 3}] = \frac{1}{2}.$$

The proof of Lemma 7 is a simple application of the "admissible oracle lemma" from [GJMZ23]. In fact it is a very special case of that lemma, where the adversary has unbounded computation and the indistinguishability is perfect. We state this simple case of the admissible oracle lemma before presenting our proof of Lemma 7.

Definition 5 ((W, Π)-distinguishing game, [GJMZ23])**.** *Let $(\mathcal{A}, \mathcal{B})$ be two quantum registers. Let W be a binary observable and Π be a projector on $(\mathcal{A}, \mathcal{B})$ such that Π commutes with W. Consider the following distinguishing game:*

1. *The adversary sends a quantum state on registers $(\mathcal{A}, \mathcal{B})$ to the challenger.*
2. *The challenger chooses a random bit $b \leftarrow \{0,1\}$. Next, it measures measures $\{\Pi, I - \Pi\}$; if the measurement rejects, abort and output a random bit $b' \leftarrow \{0,1\}$. Otherwise, the challenger applies W^b to $(\mathcal{A}, \mathcal{B})$, and returns $\mathcal{B}$ to the adversary.*
3. *The adversary outputs a guess b'.*

We define the distinguishing advantage of the adversary to be $|\Pr[b' = b] - 1/2|$.

Lemma 8 (Admissible oracle lemma — special case [GJMZ23])**.** *Suppose that every adversary achieves zero advantage in the (W, Π)-distinguishing game. Let G be an admissible unitary, i.e.,*

- *G commutes with both W and Π, and*
- *G acts identically on $I - \Pi$, i.e., $G(I - \Pi) = I - \Pi$.*

Then every adversary achieves zero advantage in the (W, Π)-distinguishing game, even when given oracle access to G.

Proof of Lemma 7. We apply Lemma 8 with the following choices of Π, W, G:

- Π is the projection on $\mathcal{A}, \mathcal{B}$ to all states of the form $|\mathcal{C}|^{-1/2} \sum_{C \in \mathcal{C}} |C\rangle \otimes C(\,|\psi\rangle \otimes |0^\lambda\rangle)$, where $\mathcal{C}$ is the Clifford group and $|\psi\rangle$ is any state.

- For $b \in \{0,1\}$, $|\tilde{\psi_b}\rangle = |\mathcal{C}|^{-1/2} \sum_{C \in \mathcal{C}} |C\rangle \otimes C(\,|\psi_b\rangle \otimes |0^\lambda\rangle)$ are the states to be distinguished.
- $W = |\tilde{\psi_0}\rangle\langle\tilde{\psi_1}| + |\tilde{\psi_1}\rangle\langle\tilde{\psi_0}| + (I - |\tilde{\psi_0}\rangle\langle\tilde{\psi_0}| - |\tilde{\psi_1}\rangle\langle\tilde{\psi_1}|)$ is the operator that swaps $|\tilde{\psi_0}\rangle$ and $|\tilde{\psi_1}\rangle$ and otherwise acts as the identity on the subspace orthogonal to the span of $|\tilde{\psi_0}\rangle$ and $|\tilde{\psi_1}\rangle$.
- G is the unitary that acts as Eval on the range of Π, and acts as the identity on the range of $I - \Pi$.

It is immediate from the fact that the Clifford group is a unitary 1-design that the (W, Π)-distinguishing game has perfect security. It is also easy to see that G is an admissible oracle for (W, Π). Therefore, Lemma 8 implies that no adversary can obtain any advantage for distinguishing between the $\mathcal{B}$ registers of $|\tilde{\psi_0}\rangle$ and $|\tilde{\psi_1}\rangle$, even given oracle access to G. □

This completes the proof of Theorem 2. □

3 Unclonable Encryption

In this section we introduce a variant of unclonable encryption (UE) that we call *coupled unclonable encryption* (cUE). Coupled unclonable encryption is a weaker primitive than UE, in the sense that any secure UE scheme can be used to build a secure cUE scheme. It closely resembles UE, the main difference being that in cUE there are *two* encryption keys that decrypt *two* messages. The main result of this section is that, unlike UE — which we do not know how to construct from standard assumptions — we can build cUE from one-way functions. The main technical ideas behind our construction are presented in Section 3.1, and the cUE construction and proof of security are given in Section 3.2.

Beyond being interesting in its own right, we will show in Section 4 that, in conjunction with qsiO, cUE is already sufficient to build copy protection for certain interesting classes of functions. In order to obtain these applications, we will need an additional feature of UE or cUE that we call *key testing*. This feature is described in Section 3.3, where we also show that key testing can be generically added to any UE or cUE scheme using qsiO and injective one-way functions.

For an outline of the ideas and techniques used in this section, see the technical overview (Section 1.3).

3.1 Unclonable randomness

We find it is easier to reason about a slightly weaker primitive than cUE, which we call "unclonable randomness." Essentially, unclonable randomness is cUE but for random messages that the adversary does not choose: it allows one to encrypt *random* strings r, s under a secret key. The security guarantee says that it is not possible to split the encryption into two states which can both be used (together with the secret key) to learn *any* information about r and s.

Since we are able to build unclonable randomness unconditionally, and since it is just a building block for our cUE construction, we only formally define it for our particular construction (rather than as an abstract primitive). The security game for our unclonable randomness construction is given in Figure 2, and security is proven in Theorem 9. This result can be viewed as a decision version of the main result of [TFKW13].

Theorem 9. *For any computationally unbounded adversary* Adv*, and any* $n, \lambda \in \mathbb{N}$*,*

$$\Pr[\texttt{Rand-Expt}_{\mathsf{Adv}}(n,\lambda) = 1] \leq \frac{1}{2} + \mathsf{poly}(n) \cdot 2^{-\Omega(\lambda)}\,,$$

where $\texttt{Rand-Expt}_{\mathsf{Adv}}(n,\lambda)$ *is described in Figure 2.*

In particular, when $n = \mathsf{poly}(\lambda)$, the advantage is negligible in λ.

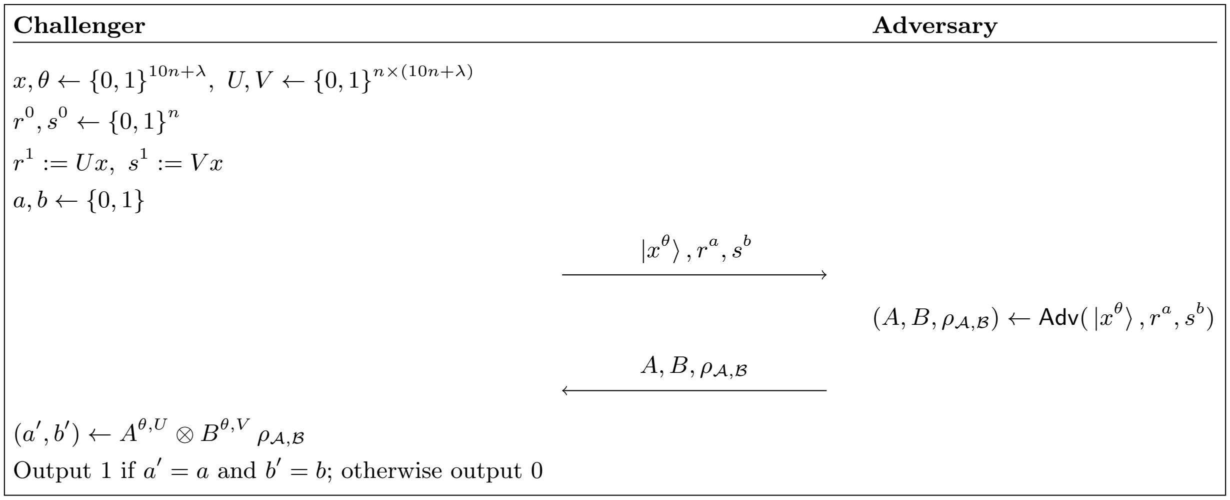

Challenger
Adversary
$x, \theta \leftarrow \{0,1\}^{10n+\lambda},\ U, V \leftarrow \{0,1\}^{n\times(10n+\lambda)}$
$r^0, s^0 \leftarrow \{0,1\}^n$
$r^1 := Ux,\ s^1 := Vx$
$a, b \leftarrow \{0,1\}$
$|x^\theta\rangle, r^a, s^b$
$(A, B, \rho_{\mathcal{A},\mathcal{B}}) \leftarrow \mathsf{Adv}(|x^\theta\rangle, r^a, s^b)$
$A, B, \rho_{\mathcal{A},\mathcal{B}}$
$(a', b') \leftarrow A^{\theta,U} \otimes B^{\theta,V}\ \rho_{\mathcal{A},\mathcal{B}}$
Output 1 if $a' = a$ and $b' = b$; otherwise output 0

Figure 2: $\texttt{Rand-Expt}_{\mathsf{Adv}}(n, \lambda)$. The challenger first generates random strings $x, \theta \leftarrow \{0,1\}^{10n+\lambda}, r^0, s^0 \leftarrow \{0,1\}^n$ and random matrices $U, V \leftarrow \{0,1\}^{n\times(10n+\lambda)}$. It then computes r^1 and s^1 as Ux and Vx respectively. The challenger samples random bits a and b, and sends the state $|x^\theta\rangle$ along with r^a and s^b to the adversary. The adversary then computes a quantum state $\rho_{\mathcal{A},\mathcal{B}}$ and circuit descriptions A and B, and sends $(A, B, \rho_{\mathcal{A},\mathcal{B}})$ back to the challenger. The challenger measures $A^{\theta,U}$ and $B^{\theta,V}$ on $\rho_{\mathcal{A},\mathcal{B}}$, obtaining outcomes a' and b'. The adversary wins if $a' = a$ and $b' = b$.

We prove Theorem 9 by reduction from a search version of the same game, defined in Figure 3. That this search version is secure follows straightforwardly from the results of [TFKW13], and is proven in Corollary 11.

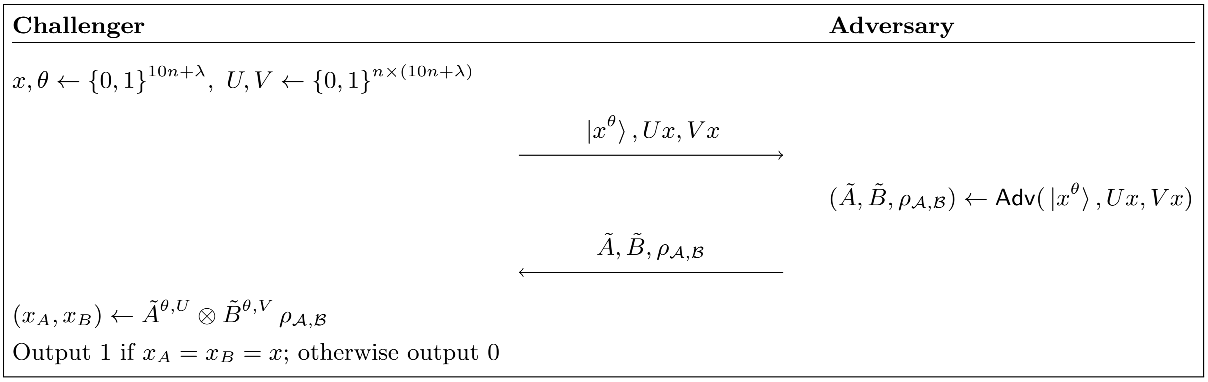

Challenger
Adversary
$x, \theta \leftarrow \{0,1\}^{10n+\lambda},\ U, V \leftarrow \{0,1\}^{n\times(10n+\lambda)}$
$|x^\theta\rangle, Ux, Vx$
$(\tilde{A}, \tilde{B}, \rho_{\mathcal{A},\mathcal{B}}) \leftarrow \mathsf{Adv}(|x^\theta\rangle, Ux, Vx)$
$\tilde{A}, \tilde{B}, \rho_{\mathcal{A},\mathcal{B}}$
$(x_A, x_B) \leftarrow \tilde{A}^{\theta,U} \otimes \tilde{B}^{\theta,V}\ \rho_{\mathcal{A},\mathcal{B}}$
Output 1 if $x_A = x_B = x$; otherwise output 0

Figure 3: $\texttt{Search-Expt}_{\mathsf{Adv}}(n, \lambda)$. The challenger generates random strings $x, \theta \leftarrow \{0,1\}^{10n+\lambda}$ and matrices $U, V \leftarrow \{0,1\}^{n\times(10n+\lambda)}$ and sends $|x^\theta\rangle, Ux, Vx$ to the adversary. The adversary responds with quantum circuits $\tilde{A}, \tilde{B}$ acting on a state $\rho_{\mathcal{A},\mathcal{B}}$. The challenger measures $\tilde{A}^{\theta,U}$ and $\tilde{B}^{\theta,V}$ on $\rho_{\mathcal{A},\mathcal{B}}$, obtaining outputs x_A and x_B. The adversary wins if $x_A = x_B = x$.

The $n = 0$ case is exactly the monogamy-of-entanglement game considered in [TFKW13]. Observe that when $n = 0$, U and V are empty and the first message is simply $|x^\theta\rangle$. When $n > 0$, the adversary is given some extra information about x.

Theorem 10 (Theorem 3 in [TFKW13]). *For $\lambda \in \mathbb{N}$ and any computationally unbounded adversary* Adv*,*

$$\Pr[\texttt{Search-Expt}_{\mathsf{Adv}}(0,\lambda) = 1] \leq \left(\frac{1}{2} + \frac{1}{2\sqrt{2}}\right)^{\lambda}.$$

In the following corollary of Theorem 10, we show that the general $\texttt{Search-Expt}$ reduces to the $n = 0$ case of $\texttt{Search-Expt}$.

Corollary 11. *For $n, \lambda \in \mathbb{N}$ and any computationally unbounded adversary* Adv*,*

$$\Pr[\texttt{Search-Expt}_{\mathsf{Adv}}(n,\lambda) = 1] \leq \left(\frac{1}{2} + \frac{1}{2\sqrt{2}}\right)^{\lambda}.$$

Proof. Suppose that Adv obtains advantage ε in $\texttt{Search-Expt}_{\mathsf{Adv}}(n,\lambda)$. We design a reduction $\mathsf{Red}[\mathsf{Adv}, n, \lambda]$ that uses Adv to play $\texttt{Search-Expt}(0, 10n+\lambda)$ by simply *guessing* the values Ux and Vx. Formally, Red is defined by the following behavior in $\texttt{Search-Expt}(0, 10n+\lambda)$.

$\texttt{Search-Expt}_{\mathsf{Red}[\mathsf{Adv},n,\lambda]}(0, 10n+\lambda)$:

1. The challenger samples $x, \theta \leftarrow \{0,1\}^{10n+\lambda}$ and sends $|x^{\theta}\rangle$ to the reduction.
2. The reduction samples $r, s \leftarrow \{0,1\}^n$ and sends $(|x^{\theta}\rangle, r, s)$ to Adv.
3. Adv outputs a state $\rho_{\mathcal{A},\mathcal{B}}$ and descriptions of $2^{10n+\lambda}$-outcome measurement families $\tilde{A}, \tilde{B}$.
4. The reduction samples $U, V \leftarrow \{0,1\}^{n\times(10n+\lambda)}$ and returns $(\tilde{A}^{\cdot,U}, \tilde{B}^{\cdot,V}, \rho_{\mathcal{A},\mathcal{B}})$ to the challenger.
5. The challenger obtains $x_A \leftarrow \tilde{A}^{\theta,U}(\rho_{\mathcal{A}})$ and $x_B \leftarrow \tilde{B}^{\theta,V}(\rho_{\mathcal{B}})$. The adversary wins if $x_A = x_B = x$.

The probability that Red samples U, V, r, s such that $Ux = r$ and $Vx = s$ is 2^{-2n}, and conditioned on this event the view of the adversary in $\texttt{Search-Expt}(n,\lambda)$ is exactly reproduced. Therefore, Red has advantage $\varepsilon/2^{2n}$ in $\texttt{Search-Expt}(0, 10n+\lambda)$. By Theorem 10, we have

$$\begin{aligned}
\varepsilon &\leq 2^{2n} \cdot \left(\frac{1}{2} + \frac{1}{2\sqrt{2}}\right)^{10n+\lambda} \\
&= \left[4 \cdot \left(\frac{1}{2} + \frac{1}{2\sqrt{2}}\right)^{10}\right]^{n} \cdot \left(\frac{1}{2} + \frac{1}{2\sqrt{2}}\right)^{\lambda} \\
&\leq \left(\frac{1}{2} + \frac{1}{2\sqrt{2}}\right)^{\lambda}.
\end{aligned}$$

□

In order to reduce the security of $\texttt{Rand-Expt}$ to that of $\texttt{Search-Expt}$, and prove Theorem 9, we require two lemmas.

Lemma 12. *Let $\{|\psi_z\rangle\}_{z\in\mathcal{Z}}$ be a family of states and $\{P_z, Q_z\}_{z\in\mathcal{Z}}$ be a family of operators. Suppose that $0 \leq P_z, Q_z \leq 1$ for all $z \in \mathcal{Z}$ and*

$$\mathop{\mathbb{E}}_{z\leftarrow\mathcal{Z}} \langle\psi_z| \left(\frac{1+P_z}{2}\right) \otimes \left(\frac{1+Q_z}{2}\right) |\psi_z\rangle \geq \frac{1}{2} + \varepsilon.$$

Then

$$\mathop{\mathbb{E}}_{z\leftarrow\mathcal{Z}} \langle\psi_z| P_z \otimes Q_z |\psi_z\rangle \geq \varepsilon^3.$$

Proof. Let $\{|\phi_z^i\rangle\}_i$ and $\{|\tau_z^j\rangle\}_j$ be eigenbases for P_z and Q_z, respectively. Then we can write

$$|\psi_z\rangle = \sum_{i,j} \alpha_z^{i,j} |\phi_z^i\rangle \otimes |\tau_z^j\rangle ,$$

and letting α_z denote the distribution over (i,j) with probabilities $|\alpha_z^{i,j}|^2$ we have

$$\mathop{\mathbb{E}}_{\substack{z\leftarrow\mathcal{Z}\\(i,j)\leftarrow\alpha_z}} \left(\frac{1}{2}+\frac{1}{2}\langle\phi_z^i| P_z |\phi_z^i\rangle\right)\left(\frac{1}{2}+\frac{1}{2}\langle\tau_z^j| Q_z |\tau_z^j\rangle\right) \geq \frac{1}{2}+\varepsilon.$$

By an averaging argument, it follows that

$$\Pr_{\substack{z\leftarrow\mathcal{Z}\\(i,j)\leftarrow\alpha_z}} \left[\langle\phi_z^i| P_z |\phi_z^i\rangle \geq \varepsilon \text{ and } \langle\tau_z^j| Q_z |\tau_z^j\rangle \geq \varepsilon\right] \geq \varepsilon.$$

Therefore

$$\begin{aligned}\mathop{\mathbb{E}}_{z\leftarrow\mathcal{Z}} \langle\psi_z| P_z\otimes Q_z |\psi_z\rangle &= \mathop{\mathbb{E}}_{\substack{z\leftarrow\mathcal{Z}\\(i,j)\leftarrow\alpha_z}} \langle\phi_z^i| P_z |\phi_z^i\rangle \langle\tau_z^j| Q_z |\tau_z^j\rangle \\ &\geq \varepsilon^3.\end{aligned}$$

□

A central component in our proof of Theorem 9 is the quantum Goldreich-Levin reduction of [BV97, AC02]. We recall that algorithm here. Let $\{A^u\}_{u\in\{0,1\}^n}$ be a collection of binary-outcome measurements and let $|\psi\rangle$ be a state.

$\mathsf{GL}(\{A^u\}_{u\in\{0,1\}^n})$:

1. Prepare the state
$$2^{-n/2}\sum_{u\in\{0,1\}^n} |u\rangle\otimes|\psi\rangle .$$
2. Apply $\sum_{u\in\{0,1\}^n}|u\rangle\langle u|\otimes A^u_{ph}$, where A^u_{ph} is the phase oracle for A^u — i.e., A^u_{ph} applies a phase of (-1) to the subspace where $A^u=1$ and acts as identity on the subspace where A^u.
3. Measure the $|u\rangle$ register in the Hadamard basis, and output the result.

Lemma 13 (Simultaneous quantum Goldreich-Levin computation)**.** *Let $\{A^u\}_{u\in\{0,1\}^n}$ and $\{B^v\}_{v\in\{0,1\}^n}$ be collections of binary-outcome measurements that act on disjoint registers $\mathcal{A}$ and $\mathcal{B}$. Let $|\psi\rangle$ be a state on $\mathcal{A},\mathcal{B}$. Then the probability that $\mathsf{GL}(\{A^u\}_{u\in\{0,1\}^n})\otimes\mathsf{GL}(\{B^v\}_{v\in\{0,1\}^n})|\psi\rangle$ returns (x,x) is*

$$\Pr\left[(x,x)\leftarrow\mathsf{GL}(\{A^u\}_{u\in\{0,1\}^n})\otimes\mathsf{GL}(\{B^v\}_{v\in\{0,1\}^n})|\psi\rangle\right] = \left\|(2\mathop{\mathbb{E}}_u \Pi_A^{x,u}-I)\otimes(2\mathop{\mathbb{E}}_v \Pi_B^{x,v}-I)|\psi\rangle\right\|^2,$$

where $\Pi_A^{x,u}$ and $\Pi_B^{x,u}$ are the projections onto the subspaces where A and B output $u\cdot x$ and $v\cdot x$, respectively.

Proof. The first step of $\mathsf{GL}(\{A^u\}_{u\in\{0,1\}^n})\otimes\mathsf{GL}(\{B^v\}_{v\in\{0,1\}^n})$ is to prepare the state

$$2^{-n}\sum_{u,v\in\{0,1\}^n}|u,v\rangle\otimes|\psi\rangle .$$

We then apply our controlled phase oracles to get the state

$$\begin{aligned}&2^{-n}\sum_{u,v\in\{0,1\}^n}|u,v\rangle\otimes(A^u\otimes B^v)|\psi\rangle\\ &=2^{-n}\sum_{u,v\in\{0,1\}^n}|u,v\rangle\otimes(-1)^{(u+v)\cdot x}\cdot(2\cdot\Pi_A^{x,u}-1)\otimes(2\cdot\Pi_B^{x,v}-1)|\psi\rangle ,\end{aligned}$$

where we have used the fact that

$$A^u_{ph} \left|\psi\right\rangle = (-1)^{u\cdot x} \cdot \Pi^{x,u}_A \left|\psi\right\rangle + (-1)^{1-u\cdot x} \cdot (1 - \Pi^{x,u}_A) \left|\psi\right\rangle = (-1)^{u\cdot x} \cdot (2 \cdot \Pi^{x,u}_A - 1) \left|\psi\right\rangle$$

(and similarly for B^v_{ph}).

Next we apply Hadamard gates to the $\left|u, v\right\rangle$ part, project onto $\left|x, x\right\rangle\!\left\langle x, x\right|$, and take the norm squared to find the probability that both $\mathsf{GL}(\{A^u\}_{u\in\{0,1\}^n})$ and $\mathsf{GL}(\{B^v\}_{v\in\{0,1\}^n})$ output x. That quantity is

$$\left\| 2^{-2n} \sum_{u,v\in\{0,1\}^n} \left|x, x\right\rangle \otimes (2 \cdot \Pi^{x,u}_A - 1) \otimes (2 \cdot \Pi^{x,v}_B - 1) \left|\psi\right\rangle \right\|^2$$

$$= \left\| \mathop{\mathbb{E}}_{u,v\in\{0,1\}^n} (2 \cdot \Pi^{x,u}_A - 1) \otimes (2 \cdot \Pi^{x,v}_B - 1) \left|\psi\right\rangle \right\|^2 . \qquad \square$$

Proof of Theorem 9. As anticipated, we reduce the security of `Rand-Expt` to that of `Search-Expt` (Figure 3), which we established in Corollary 11. The first step of the proof is to rewrite `Search-Expt` in an equivalent form.

Observe that the challenger in `Search-Expt` could sample the vectors corresponding to Ux, Vx *before* actually deciding on the matrices U, V, and the security game would be identical. That is, the challenger will sample random vectors $r, s \leftarrow \{0, 1\}^n$ and send $\left|x^\theta\right\rangle, r, s$ to the adversary in the first step. Later, in order to run $\tilde{A}, \tilde{B}$, it will just sample random matrices U, V conditioned on $Ux = r$ and $Vx = s$.

Before we give a formal description of this equivalent game, we define distributions that will be useful throughout the proof. For $i \in [n]$, $x \in \{0, 1\}^{10n+\lambda}$, and $r \in \{0, 1\}^n$, let $\mathcal{D}_i(x, r)$ be the distribution over matrices $U \in \{0, 1\}^{n\times(10n+\lambda)}$ where row $j \in [n]$ is sampled as

$$U_j = \begin{cases} u_j \leftarrow \{u \in \{0,1\}^{10n+\lambda} \mid u \cdot x = r_j\}, & j \leq i \\ u_j \leftarrow \{0,1\}^{10n+\lambda}, & j > i. \end{cases}$$

That is, $U \leftarrow \mathcal{D}_i(x, r)$ is a random matrix conditioned on the first i values of Ux being equal to the first i values of r. Now our equivalent formulation of `Search-Expt` is as follows. Without loss of generality we assume that the state $\rho_{\mathcal{A},\mathcal{B}}$ is a pure state $\left|\psi_{x,\theta,r,s}\right\rangle$.

$\texttt{Search-Expt}_{\mathsf{Adv}}(n, \lambda)$:

1. The challenger samples $x, \theta \leftarrow \{0, 1\}^{10n+\lambda}$ and $r, s \leftarrow \{0, 1\}^n$, then sends $\left|x^\theta\right\rangle, r, s$ to Adv.
2. Adv outputs a state $\left|\psi_{x,\theta,r,s}\right\rangle$ and descriptions of $2^{10n+\lambda}$-outcome measurement families $\tilde{A}, \tilde{B}$.
3. The challenger samples $U \leftarrow D_n(x, r), V \leftarrow D_n(x, s)$ and measures $\tilde{A}^{\theta,U}$ and $\tilde{B}^{\theta,V}$ on $\left|\psi_{x,\theta,r,s}\right\rangle$, obtaining outcomes x_A and x_B. The adversary wins if $x_A = x_B = x$.

We similarly rewrite `Rand-Expt` in an equivalent form where the challenger decides on the matrices at the end of the game.

$\texttt{Rand-Expt}_{\mathsf{Adv}}(n, \lambda)$:

1. The challenger samples $x, \theta \leftarrow \{0, 1\}^{10n+\lambda}$ and $r, s \leftarrow \{0, 1\}^n$, then sends $\left|x^\theta\right\rangle, r, s$ to Adv.
2. Adv outputs a state $\left|\psi_{x,\theta,r,s}\right\rangle$ and descriptions of binary-outcome measurement families A, B.
3. The challenger samples $a, b \leftarrow \{0, 1\}$, then samples $U \leftarrow D_{a\cdot n}(x, r), V \leftarrow D_{b\cdot n}(x, s)$ and measures $A^{\theta,U}$ and $B^{\theta,V}$ on $\left|\psi_{x,\theta,r,s}\right\rangle$, obtaining outcomes a' and b'. The adversary wins if $a' = a$ and $b' = b$.

Note that the first message is the same in both `Search-Expt` and `Rand-Expt`, and the challenge bits a, b in `Rand-Expt` are sampled independently from the first message. Given an adversary Adv for $\texttt{Rand-Expt}_{\mathsf{Adv}}(n,\lambda)$, we define an adversary $\mathsf{Red}[\mathsf{Adv}]$ for `Search-Expt`. The latter uses the [BV97, AC02] reduction where the adversaries guess random bits of r, s.

$\texttt{Search-Expt}_{\mathsf{Red}[\mathsf{Adv}]}(n,\lambda)$:

1. The challenger samples $x, \theta \leftarrow \{0,1\}^{10n+\lambda}$ and $r, s \leftarrow \{0,1\}^n$, then sends $|x^\theta\rangle, r, s$ to Red, which forwards everything to Adv.
2. Adv outputs a state $|\psi_{x,\theta,r,s}\rangle$ and descriptions of binary-outcome measurement families A, B.
3. Red samples random $i, j \leftarrow [n]$ and $\tilde{U}, \tilde{V} \leftarrow \{0,1\}^{n\times(10n+\lambda)}$. Let $\tilde{A}^{\theta,U} := \mathsf{GL}(\{A^{\theta,[U_{<i}||u||\tilde{U}_{>i}]}\}_{u\in\{0,1\}^n})$ and $\tilde{B}^{\theta,V} := \mathsf{GL}(\{B^{\theta,[V_{<i}||v||\tilde{V}_{>i}]}\}_{v\in\{0,1\}^n})$. Red sends $\tilde{A}, \tilde{B}, |\psi_{x,\theta,r,s}\rangle$ to the challenger.
4. The challenger samples $U \leftarrow D_n(x,r), V \leftarrow D_n(x,s)$ and measures $\tilde{A}^{\theta,U}$ and $\tilde{B}^{\theta,V}$ on $|\psi_{x,\theta,r,s}\rangle$, obtaining outcomes x_A and x_B. The reduction wins if $x_A = x_B = x$.

Suppose that

$$\Pr[\texttt{Rand-Expt}_{\mathsf{Adv}}(n,\lambda) = 1] \geq \frac{1}{2} + \varepsilon.$$

In `Rand-Expt`, we denote the projections onto outcomes a and b by A_a and B_b, respectively. Then,

$$\begin{aligned}
&\frac{1}{2} + \varepsilon \\
&\leq \mathop{\mathbb{E}}_{\substack{x,\theta\leftarrow\{0,1\}^{10n+\lambda}\\ r,s\leftarrow\{0,1\}^n}} \langle\psi_{x,\theta,r,s}| \left(\frac{\mathbb{E}_{U\leftarrow D_0(x,r)}[A_0^{\theta,U}] + \mathbb{E}_{U\leftarrow D_n(x,r)}[A_1^{\theta,U}]}{2}\right) \\
&\qquad\qquad \otimes \left(\frac{\mathbb{E}_{V\leftarrow D_0(x,s)}[B_0^{\theta,V}] + \mathbb{E}_{V\leftarrow D_n(x,s)}[B_1^{\theta,V}]}{2}\right) |\psi_{x,\theta,r,s}\rangle \\
&= \mathop{\mathbb{E}}_{\substack{x,\theta\leftarrow\{0,1\}^{10n+\lambda}\\ r,s\leftarrow\{0,1\}^n}} \langle\psi_{x,\theta,r,s}| \left(\frac{1 + \mathbb{E}_{U\leftarrow D_0(x,r)}[A_0^{\theta,U}] - \mathbb{E}_{U\leftarrow D_n(x,r)}[A_0^{\theta,U}]}{2}\right) \\
&\qquad\qquad \otimes \left(\frac{1 + \mathbb{E}_{V\leftarrow D_0(x,s)}[B_0^{\theta,V}] - \mathbb{E}_{V\leftarrow D_n(x,s)}[B_0^{\theta,V}]}{2}\right) |\psi_{x,\theta,r,s}\rangle \\
&\leq \mathop{\mathbb{E}}_{\substack{x,\theta\leftarrow\{0,1\}^{10n+\lambda}\\ r,s\leftarrow\{0,1\}^n}} \langle\psi_{x,\theta,r,s}| \left(\frac{1 + \left|\mathbb{E}_{U\leftarrow D_0(x,r)}[A_0^{\theta,U}] - \mathbb{E}_{U\leftarrow D_n(x,r)}[A_0^{\theta,U}]\right|}{2}\right) \\
&\qquad\qquad \otimes \left(\frac{1 + \left|\mathbb{E}_{V\leftarrow D_0(x,s)}[B_0^{\theta,V}] - \mathbb{E}_{V\leftarrow D_n(x,s)}[B_0^{\theta,V}]\right|}{2}\right) |\psi_{x,\theta,r,s}\rangle \,.
\end{aligned}$$

By Lemma 12, we have

$$\begin{aligned}
&\varepsilon^3 \leq \\
&\mathop{\mathbb{E}}_{\substack{x,\theta\leftarrow\{0,1\}^{10n+\lambda}\\ r,s\leftarrow\{0,1\}^n}} \langle\psi_{x,\theta,r,s}| \left|\mathop{\mathbb{E}}_{U\leftarrow D_0(x,r)}[A_0^{\theta,U}] - \mathop{\mathbb{E}}_{U\leftarrow D_n(x,r)}[A_0^{\theta,U}]\right| \otimes \left|\mathop{\mathbb{E}}_{V\leftarrow D_0(x,s)}[B_0^{\theta,V}] - \mathop{\mathbb{E}}_{V\leftarrow D_n(x,s)}[B_0^{\theta,V}]\right| |\psi_{x,\theta,r,s}\rangle \,.
\end{aligned}$$

Next we use a sort of “hybrid argument” to relate the *operators* in the two games:

$$\begin{aligned}
&\left|\mathop{\mathbb{E}}_{U\leftarrow D_0(x,r)}[A_0^{\theta,U}] - \mathop{\mathbb{E}}_{U\leftarrow D_n(x,r)}[A_0^{\theta,U}]\right| \\
&= \left|\sum_{i=1}^{n}\left(\mathop{\mathbb{E}}_{U\leftarrow D_i(x,r)}[A_0^{\theta,U}] - \mathop{\mathbb{E}}_{U\leftarrow D_{i-1}(x,r)}[A_0^{\theta,U}]\right)\right| \\
&= \left|\sum_{i=1}^{n}\mathop{\mathbb{E}}_{\substack{U\leftarrow D_n(x,r)\\ \tilde{U}\leftarrow D_0(x,r)}}\left(\mathop{\mathbb{E}}_{u:u\cdot x=r_i}[A_0^{\theta,[U_{<i}||u||\tilde{U}_{>i}]}] - \mathop{\mathbb{E}}_{u}[A_0^{\theta,[U_{<i}||u||\tilde{U}_{>i}]}]\right)\right| \\
&= \frac{1}{2}\left|\sum_{i=1}^{n}\mathop{\mathbb{E}}_{\substack{U\leftarrow D_n(x,r)\\ \tilde{U}\leftarrow D_0(x,r)}}\left(\mathop{\mathbb{E}}_{u:u\cdot x=r_i}[A_0^{\theta,[U_{<i}||u||\tilde{U}_{>i}]}] - \mathop{\mathbb{E}}_{u:u\cdot x\neq r_i}[A_0^{\theta,[U_{<i}||u||\tilde{U}_{>i}]}]\right)\right| \\
&\leq \frac{1}{2}\sum_{i=1}^{n}\mathop{\mathbb{E}}_{\substack{U\leftarrow D_n(x,r)\\ \tilde{U}\leftarrow D_0(x,r)}}\left|\mathop{\mathbb{E}}_{u:u\cdot x=r_i}[A_0^{\theta,[U_{<i}||u||\tilde{U}_{>i}]}] - \mathop{\mathbb{E}}_{u:u\cdot x\neq r_i}[A_0^{\theta,[U_{<i}||u||\tilde{U}_{>i}]}]\right| \\
&= \frac{n}{2}\mathop{\mathbb{E}}_{\substack{i\leftarrow[n]\\ U\leftarrow D_n(x,r)\\ \tilde{U}\leftarrow D_0(x,r)}}\left|2\mathop{\mathbb{E}}_{u}A_{u\cdot x+r_i}^{\theta,[U_{<i}||u||\tilde{U}_{>i}]} - 1\right|.
\end{aligned}$$

The above holds identically for the B part. We are now ready to bound the probability that our reduction wins `Search-Expt`. We begin by applying Lemma 13:

$$\begin{aligned}
&\Pr\left[\texttt{Search-Expt}_{\mathsf{Red[Adv]}}(n,\lambda)=1\right] \\
&= \mathop{\mathbb{E}}_{\substack{x,\theta\leftarrow\{0,1\}^{10n+\lambda}\\ r,s\leftarrow\{0,1\}^n}}\ \mathop{\mathbb{E}}_{\substack{i,j\leftarrow[n]\\ U\leftarrow D_n(x,r),\tilde{U}\leftarrow D_0(x,r)\\ V\leftarrow D_n(x,s),\tilde{V}\leftarrow D_0(x,s)}}\left\|(2\mathop{\mathbb{E}}_{u}A_{u\cdot x}^{\theta,[U_{<i}||u||\tilde{U}_{>i}]}-1)\otimes(2\mathop{\mathbb{E}}_{v}B_{v\cdot x}^{\theta,[V_{<j}||v||\tilde{V}_{>j}]}-1)\left|\psi_{x,\theta,r,s}\right\rangle\right\|^2 \\
&= \mathop{\mathbb{E}}_{\substack{x,\theta\leftarrow\{0,1\}^{10n+\lambda}\\ r,s\leftarrow\{0,1\}^n}}\ \mathop{\mathbb{E}}_{\substack{i,j\leftarrow[n]\\ U\leftarrow D_n(x,r),\tilde{U}\leftarrow D_0(x,r)\\ V\leftarrow D_n(x,s),\tilde{V}\leftarrow D_0(x,s)}}\left\langle\psi_{x,\theta,r,s}\right|\left|2\mathop{\mathbb{E}}_{u}A_{u\cdot x}^{\theta,[U_{<i}||u||\tilde{U}_{>i}]}-1\right|^2\otimes\left|2\mathop{\mathbb{E}}_{v}B_{v\cdot x}^{\theta,[V_{<j}||v||\tilde{V}_{>j}]}-1\right|^2\left|\psi_{x,\theta,r,s}\right\rangle \\
&= \mathop{\mathbb{E}}_{\substack{x,\theta\leftarrow\{0,1\}^{10n+\lambda}\\ r,s\leftarrow\{0,1\}^n}}\ \mathop{\mathbb{E}}_{\substack{i,j\leftarrow[n]\\ U\leftarrow D_n(x,r),\tilde{U}\leftarrow D_0(x,r)\\ V\leftarrow D_n(x,s),\tilde{V}\leftarrow D_0(x,s)}}\left\langle\psi_{x,\theta,r,s}\right|\left|2\mathop{\mathbb{E}}_{u}A_{u\cdot x+r_i}^{\theta,[U_{<i}||u||\tilde{U}_{>i}]}-1\right|^2\otimes\left|2\mathop{\mathbb{E}}_{v}B_{v\cdot x+s_j}^{\theta,[V_{<j}||v||\tilde{V}_{>j}]}-1\right|^2\left|\psi_{x,\theta,r,s}\right\rangle
\end{aligned}$$

where the last line is because $\left|2\,\mathbb{E}_u\,A_{u\cdot x}^{\theta,[U_{<i}||u||\tilde{U}_{>i}]}-1\right| = \left|2\,\mathbb{E}_u\,A_{u\cdot x+1}^{\theta,[U_{<i}||u||\tilde{U}_{>i}]}-1\right|$, and similarly for the B

term. Finally, we have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\substack{x,\theta\leftarrow\{0,1\}^{10n+\lambda}\\ r,s\leftarrow\{0,1\}^n}} \mathop{\mathbb{E}}_{\substack{i,j\leftarrow[n]\\ U\leftarrow D_n(x,r),\tilde{U}\leftarrow D_0(x,r)\\ V\leftarrow D_n(x,s),\tilde{V}\leftarrow D_0(x,s)}} \langle\psi_{x,\theta,r,s}| \left|2\mathop{\mathbb{E}}_u A^{\theta,[U_{<i}||u||\tilde{U}_{>i}]}_{u\cdot x+r_i} - 1\right|^2 \otimes \left|2\mathop{\mathbb{E}}_v B^{\theta,[V_{<j}||v||\tilde{V}_{>j}]}_{v\cdot x+s_j} - 1\right|^2 |\psi_{x,\theta,r,s}\rangle \\
&= \mathop{\mathbb{E}}_{\substack{x,\theta\leftarrow\{0,1\}^{10n+\lambda}\\ r,s\leftarrow\{0,1\}^n}} \mathop{\mathbb{E}}_{\substack{i,j\leftarrow[n]\\ U\leftarrow D_n(x,r),\tilde{U}\leftarrow D_0(x,r)\\ V\leftarrow D_n(x,s),\tilde{V}\leftarrow D_0(x,s)}} \left(\langle\psi_{x,\theta,r,s}| \left|2\mathop{\mathbb{E}}_u A^{\theta,[U_{<i}||u||\tilde{U}_{>i}]}_{u\cdot x+r_i} - 1\right| \otimes \left|2\mathop{\mathbb{E}}_v B^{\theta,[V_{<j}||v||\tilde{V}_{>j}]}_{v\cdot x+s_j} - 1\right| |\psi_{x,\theta,r,s}\rangle\right)^2 \\
&\geq \mathop{\mathbb{E}}_{\substack{x,\theta\leftarrow\{0,1\}^{10n+\lambda}\\ r,s\leftarrow\{0,1\}^n}} \langle\psi_{x,\theta,r,s}| \left(\mathop{\mathbb{E}}_{\substack{i\leftarrow[n]\\ U\leftarrow D_n(x,r)\\ \tilde{U}\leftarrow D_0(x,r)}} \left|2\mathop{\mathbb{E}}_u A^{\theta,[U_{<i}||u||\tilde{U}_{>i}]}_{u\cdot x+r_i} - 1\right|\right) \otimes \left(\mathop{\mathbb{E}}_{\substack{j\leftarrow[n]\\ V\leftarrow D_n(x,s)\\ \tilde{V}\leftarrow D_0(x,s)}} \left|2\mathop{\mathbb{E}}_v B^{\theta,[V_{<j}||v||\tilde{V}_{>j}]}_{v\cdot x+s_j} - 1\right|\right) |\psi_{x,\theta,r,s}\rangle^2 \\
&\geq \frac{4}{n^2} \mathop{\mathbb{E}}_{\substack{x,\theta\leftarrow\{0,1\}^{10n+\lambda}\\ r,s\leftarrow\{0,1\}^n}} \langle\psi_{x,\theta,r,s}| \left|\mathop{\mathbb{E}}_{U\leftarrow D_0(x,r)}[A_0^{\theta,U}] - \mathop{\mathbb{E}}_{U\leftarrow D_n(x,r)}[A_0^{\theta,U}]\right| \otimes \left|\mathop{\mathbb{E}}_{V\leftarrow D_0(x,s)}[B_0^{\theta,V}] - \mathop{\mathbb{E}}_{V\leftarrow D_n(x,s)}[B_0^{\theta,V}]\right| |\psi_{x,\theta,r,s}\rangle^2 \\
&\geq \frac{4\varepsilon^6}{n^2}.
\end{aligned}$$

where the first inequality is by convexity. By Corollary 11, this quantity is at most $\left(\frac{1}{2}+\frac{1}{2\sqrt{2}}\right)^\lambda$. Therefore, $\varepsilon \leq n^{1/3}\cdot 2^{-\Omega(\lambda)}$. □

3.2 Coupled unclonable encryption

In this subsection, we introduce *coupled unclonable encryption* (cUE). It is similar to UE, except that it involves the simultaneous encryption of two messages m_A and m_B under two secret keys sk_A and sk_B. Informally, security for cUE says that when a pirate processes the ciphertext into two parts, one given to Alice and the other to Bob, then after receiving sk_A and sk_B it is not possible for both of Alice and Bob to simultaneously recover any information about their respective messages m_A and m_B. While cUE is weaker than UE, we are able to make use of it in Section 4 as a central primitive in our proofs that qsiO copy-protects puncturable programs.

Before introducing cUE in Definition 7, we recall the definition of UE in Definition 6.

Definition 6. *A pair*[5] *of efficient quantum algorithms* $(\mathsf{Enc}, \mathsf{Dec})$ *is an* unclonable encryption (UE) scheme *if it satisfies the following conditions, for all* $\lambda, n \in \mathbb{N}$*:*

- *(Correctness) For all* $\mathsf{sk} \in \{0,1\}^\lambda$ *and* $m \in \{0,1\}^n$*,*

$$\mathsf{Dec}(\mathsf{sk}; \mathsf{Enc}(\mathsf{sk}; m)) \to m$$

- *(Security) For all polynomial-time adversaries* Adv*,*

$$\Pr\left[\mathtt{UE\text{-}Expt}_{\mathsf{Enc},\mathsf{Adv}}(\lambda) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda)\,,$$

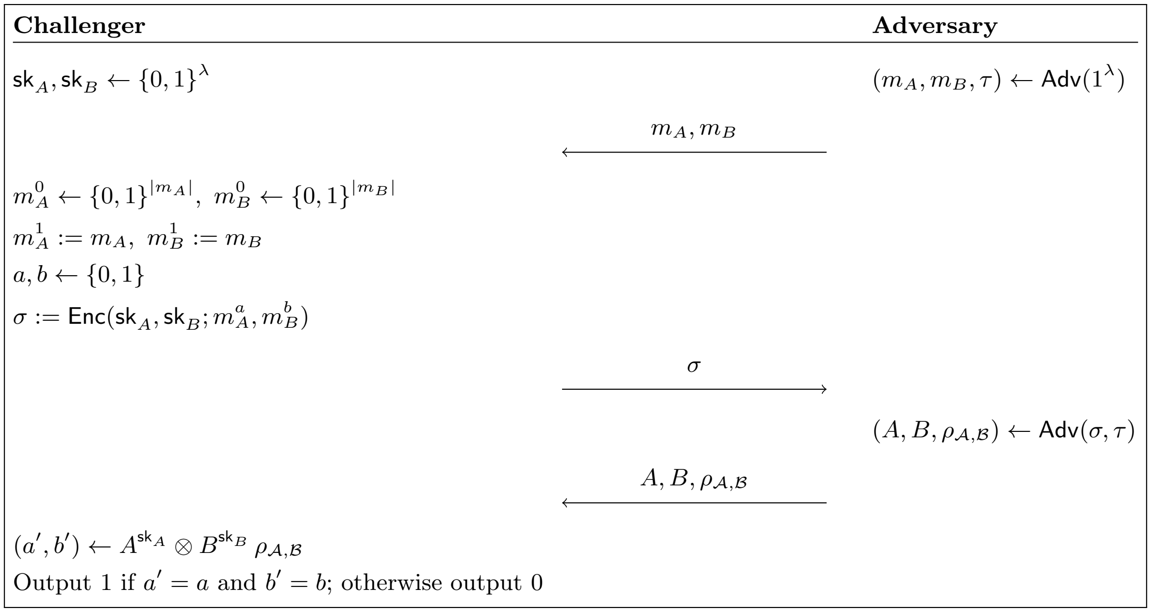
where $\mathtt{UE\text{-}Expt}_{\mathsf{Enc},\mathsf{Adv}}(\lambda)$ *is defined in Figure 4.*

Let $\{\mathsf{PRG}_\lambda\}_{\lambda\in\mathbb{N}}$ be a family of pseudo-random generators with 1 bit of stretch. For $n > \lambda$, define $\mathsf{PRG}_{\lambda,n} : \{0,1\}^\lambda \to \{0,1\}^n$ to be $\mathsf{PRG}_{n-1} \circ \cdots \circ \mathsf{PRG}_{\lambda+1} \circ \mathsf{PRG}_\lambda$, the $(n-\lambda)$-fold composition of PRG. For $n \leq \lambda$, define

[5] Note that an alternative definition might involve a third algorithm for generating keys. However, we require that the keys are sampled uniformly at random as this will make our qsiO applications simpler.

Challenger		**Adversary**
$\mathsf{sk} \leftarrow \{0,1\}^\lambda$		$m, \tau \leftarrow \mathsf{Adv}(1^\lambda)$
	$\xleftarrow{\quad m \quad}$	
$m^0 \leftarrow \{0,1\}^{\lvert m \rvert}$		
$m^1 := m$		
$c \leftarrow \{0,1\}$		
$\sigma := \mathsf{Enc}(\mathsf{sk}; m^c)$		
	$\xrightarrow{\quad \sigma \quad}$	
		$(A, B, \rho_{\mathcal{A},\mathcal{B}}) \leftarrow \mathsf{Adv}(\sigma, \tau)$
	$\xleftarrow{\quad A, B, \rho_{\mathcal{A},\mathcal{B}} \quad}$	
$(a', b') \leftarrow A^{\mathsf{sk}} \otimes B^{\mathsf{sk}}\, \rho_{\mathcal{A}\,\mathcal{B}}$		
Output 1 if $a' = b' = c$; otherwise output 0		

Figure 4: $\texttt{UE-Expt}_{\mathsf{Enc},\mathsf{Adv}}(\lambda)$. The challenger samples a secret encryption key sk, while the adversary decides on a message m and sends it to the challenger. The resulting internal state of the adversary is τ, which will be provided to the next part of the adversary. The challenger samples a fresh random message m^0, sets $m^1 := m$, and encrypts m^c for $c \leftarrow \{0,1\}$ using sk. The challenger sends the encryption σ to the adversary, who maps this to a state $\rho_{\mathcal{A}\,\mathcal{B}}$ on the two registers $\mathcal{A}, \mathcal{B}$ and returns $\rho_{\mathcal{A}\,\mathcal{B}}$ to the challenger, together with descriptions of (families of) quantum circuits A and B on $\mathcal{A}$ and $\mathcal{B}$, respectively, indexed by keys. The challenger runs A^{sk} and B^{sk} on $\rho_{\mathcal{A}\,\mathcal{B}}$, obtaining outcomes a' and b'. The adversary wins if $a' = b' = c$.



Challenger **Adversary**

$\mathsf{sk}_A, \mathsf{sk}_B \leftarrow \{0,1\}^\lambda$ $(m_A, m_B, \tau) \leftarrow \mathsf{Adv}(1^\lambda)$

$\xleftarrow{m_A, m_B}$

$m_A^0 \leftarrow \{0,1\}^{|m_A|},\ m_B^0 \leftarrow \{0,1\}^{|m_B|}$

$m_A^1 := m_A,\ m_B^1 := m_B$

$a, b \leftarrow \{0,1\}$

$\sigma := \mathsf{Enc}(\mathsf{sk}_A, \mathsf{sk}_B; m_A^a, m_B^b)$

$\xrightarrow{\sigma}$

$(A, B, \rho_{\mathcal{A},\mathcal{B}}) \leftarrow \mathsf{Adv}(\sigma, \tau)$

$\xleftarrow{A, B, \rho_{\mathcal{A},\mathcal{B}}}$

$(a', b') \leftarrow A^{\mathsf{sk}_A} \otimes B^{\mathsf{sk}_B}\ \rho_{\mathcal{A},\mathcal{B}}$

Output 1 if $a' = a$ and $b' = b$; otherwise output 0

Figure 5: $\texttt{cUE-Expt}_{\mathsf{Enc},\mathsf{Adv}}(\lambda)$. The challenger samples encryption keys $\mathsf{sk}_A, \mathsf{sk}_B \leftarrow \{0,1\}^\lambda$. The adversary outputs messages m_A, m_B; its internal state τ will be used later. The challenger generates messages m_A^0, m_B^0, sets $m_A^1 := m_A$, $m_B^1 := m_B$, and randomly decides bits a, b. It encrypts m_A^a, m_B^b with $\mathsf{sk}_A, \mathsf{sk}_B$ into σ and sends it to the adversary. The adversary, with state σ, τ, generates circuit descriptions A, B, and a state $\rho_{\mathcal{A},\mathcal{B}}$, and sends them to the challenger. The challenger applies A^{sk_A} to $\rho_{\mathcal{A}}$, giving a', and B^{sk_B} to $\rho_{\mathcal{B}}$, giving b'. The adversary wins if $a' = a$ and $b' = b$.

$\mathsf{PRG}_{\lambda,n} : \{0,1\}^\lambda \to \{0,1\}^n$ to be the restriction to the first n coordinates of the input. Then $\{\mathsf{PRG}_{\lambda,n}\}_{\lambda,n\in\mathbb{N}}$ is a family of pseudo-random generators of arbitrary stretch. The following is a natural candidate UE scheme:

$\mathsf{Enc}(\mathsf{sk}; m)$:

1. Parse sk as $(\theta, U) \in \{0,1\}^{11\lambda'} \times \{0,1\}^{\lambda' \times 11\lambda'}$, where λ' is the largest integer such that $11(\lambda')^2 + 11\lambda' \leq \lambda$.
2. Sample a random string $x \leftarrow \{0,1\}^{11\lambda'}$.
3. Output $(\,|x^\theta\rangle, m \oplus \mathsf{PRG}_{\lambda',|m|}(Ux))$, where
   - Ux denotes the matrix-vector product over $\mathbb{F}_2$, and
   - $|x^\theta\rangle$ is $H^\theta\,|x\rangle$, where H^θ denotes Hadamard gates applied to the qubits where the corresponding bit in θ is 1.

The decryption algorithm simply reads x (since sk includes θ) and computes

$$\Big(m \oplus \mathsf{PRG}_{\lambda',|m|}(Ux)\Big) \oplus \mathsf{PRG}_{\lambda',|m|}(Ux) = m.$$

Note that we require the pseudo-random generator because, in our definition, the secret key is of a fixed length λ whereas the length of the message is determined by the adversary. If we were satisfied with encrypting fixed-length messages (with a secret key that grows with the message length), then we could build cUE unconditionally.

Definition 7. *A pair of efficient quantum algorithms* $(\mathsf{Enc}, \mathsf{Dec})$ *is a* coupled unclonable encryption (cUE) scheme *if it satisfies the following conditions, for all* $\lambda, n_A, n_B \in \mathbb{N}$*:*

- *(Correctness) For all* $\mathsf{sk}_A, \mathsf{sk}_B \in \{0,1\}^\lambda$ *and* $m_A \in \{0,1\}^{n_A}, m_B \in \{0,1\}^{n_B}$,

$$\begin{aligned}&\mathsf{Dec}(0, \mathsf{sk}_A; \mathsf{Enc}(\mathsf{sk}_A, \mathsf{sk}_B; m_A, m_B)) \to m_A \text{ and}\\ &\mathsf{Dec}(1, \mathsf{sk}_B; \mathsf{Enc}(\mathsf{sk}_A, \mathsf{sk}_B; m_A, m_B)) \to m_B.\end{aligned}$$

- *(Security) For all polynomial-time adversaries* Adv,

$$\Pr\left[\texttt{cUE-Expt}_{\mathsf{Enc},\mathsf{Adv}}(\lambda) = 1\right] \le \frac{1}{2} + \mathsf{negl}(\lambda).$$

Remark 3. *The reader may be wondering whether the security guarantees of UE or cUE imply standard CPA security. For UE, it is straightforward to see that Definition 6 implies CPA security: An adversary breaking CPA encryption can be used in the UE game to recover a guess for the challenge bit c. Then the UE adversary can simply set A and B to be families of circuits that always output c. On the other hand, for cUE (Definition 7) the natural reduction implies that no adversary can simultaneously guess both challenges — leaving open the possibility that the adversary can guess one of the challenges. It is therefore not clear whether cUE security implies CPA security for each message separately.*

Theorem 9 about unclonable randomness gets us most of the way towards building cUE. The natural approach to construct cUE is to use the unclonable randomness as a one-time pad for the adversary's chosen message. However, there are two small technical issues. First, in cUE the keys sk_A and sk_B must be sampled independently, but the keys (θ, U) and (θ, V) in the unclonable randomness game cannot be sampled independently because they both contain θ. Second, the length of the message is determined by the adversary in the cUE game, whereas unclonable randomness has a fixed-length message as a function of λ. Therefore, our cUE scheme is slightly more complex than our unclonable randomness scheme, and additionally uses a pseudorandom generator.

We note that the matrix T in Construction 2 just serves to make the keys $\mathsf{sk}_A, \mathsf{sk}_B$ independent. If we were satisfied with the keys being partly identical (on θ) and partly independent (on U, V), then we would not need T.

Construction 2. *Let* $\{\mathsf{PRG}_{\lambda,n} : \{0,1\}^\lambda \to \{0,1\}^n\}_{\lambda,n\in\mathbb{N}}$ *be a family of pseudorandom generators. Define* Enc *and* Dec *as follows.*

- $\mathsf{Enc}(\mathsf{sk}_A, \mathsf{sk}_B; m_A, m_B)$*:*
  1. *Let* λ' *be the largest integer such that* $11(\lambda')^2 + 11\lambda' + 1 \le \lambda$*. We parse the secret keys as* $\mathsf{sk}_A = (\theta_A, U, -)$ *and* $\mathsf{sk}_B = (\theta_B, V, -)$*, where* $U, V \in \{0,1\}^{\lambda' \times (11\lambda')}$ *and* $\theta_A, \theta_B \in \{0,1\}^{11\lambda'+1}$*. If there are any leftover bits in* $\mathsf{sk}_A, \mathsf{sk}_B$ *we discard them.*
  2. *Sample* $x \leftarrow \{0,1\}^{11\lambda'}$ *and* $T \leftarrow \{0,1\}^{(11\lambda')\times(11\lambda'+1)}$ *conditioned on* $T\theta_A = T\theta_B$ *and* $\operatorname{rank}(T) = 11\lambda'$*. Define* $\theta := T\theta_A = T\theta_B$*.*
  3. *Output* $\left|x^\theta\right\rangle, T, m_A \oplus \mathsf{PRG}_{\lambda',|m_A|}(Ux), m_B \oplus \mathsf{PRG}_{\lambda',|m_B|}(Vx)$*.*
- $\mathsf{Dec}(p, \mathsf{sk}; c)$*: Parse* sk *as* $\mathsf{sk} = (\theta', W, -)$ *(again discarding any leftover bits). Parse* c *as* $c = (|\psi\rangle, T, c_0, c_1)$*. Compute* $\theta = T\theta'$*. Apply* H^θ *to* $|\psi\rangle$*, and then measure in the standard basis to obtain an outcome* x*. Output* $c_p \oplus \mathsf{PRG}_{\lambda',|c_p|}(Wx)$*.*

Theorem 14. *One-way functions imply the existence of cUE. In particular, Construction 2 is cUE.*

Proof. We reduce security of Construction 2 to security of unclonable randomness (recall that the latter is defined via $\texttt{Rand-Expt}$ from Figure 2). Let Adv be an efficient adversary for the security experiment $\texttt{cUE-Expt}$ for cUE, and let λ be a security parameter. Let λ' be the largest integer such that $11(\lambda')^2 + 11\lambda' + 1 \le \lambda$. We define an adversary $\mathsf{Red}[\mathsf{Adv}]$ for $\texttt{Rand-Expt}(\lambda', \lambda')$ as follows.

$\texttt{Rand-Expt}_{\mathsf{Red}[\mathsf{Adv}]}(\lambda', \lambda')$:

1. The challenger samples $a, b \leftarrow \{0,1\}$, $x, \theta \leftarrow \{0,1\}^{11\lambda'}$, $r^0, s^0 \leftarrow \{0,1\}^{\lambda'}$, and $U, V \leftarrow \{0,1\}^{\lambda' \times (11\lambda')}$. Let $r^1 := Ux$ and $s^1 := Vx$.
2. The challenger sends $|x^\theta\rangle, r^a, s^b$ to Red.
3. Red samples messages $(m_A, m_B) \leftarrow \mathsf{Adv}(1^\lambda)$ and a uniformly random rank-$(11\lambda')$ matrix T from $\{0,1\}^{(11\lambda') \times (11\lambda'+1)}$.
4. Red runs $(A, B, \rho_{\mathcal{A},\mathcal{B}}) \leftarrow \mathsf{Adv}(|x^\theta\rangle, T, m_A \oplus \mathsf{PRG}_{\lambda',|m_A|}(r^a), m_B \oplus \mathsf{PRG}_{\lambda',|m_B|}(s^b))$.
5. For $d \in \{0,1\}, \theta' \in \{0,1\}^{11\lambda'}, U' \in \{0,1\}^{\lambda' \times 11\lambda'}$, define the circuit $\tilde{A}_d^{\theta',U'}$ as follows:
   (a) Let $\theta_A, \theta_B \in \{0,1\}^{11\lambda'+1}$ be the two vectors such that $T\theta_A = T\theta_B = \theta'$. Let θ_A be whichever vector has the first differing bit between θ_A and θ_B equal to d; similarly let θ_B be the vector which has $1-d$ at the first differing location.
   (b) Let $\mathsf{sk}_A := (\theta_A, U', \mathtt{pad})$ where $\mathtt{pad}$ is a random string of length $\lambda - 11(\lambda')^2 - 11\lambda' - 1$.
   (c) Return the output of running A^{sk_A} on the input state.

   Define $\tilde{B}_d^{\theta',V'}$ similarly.
6. Red samples $d \leftarrow \{0,1\}$ and sends $\rho_{\mathcal{A},\mathcal{B}}$ and $\tilde{A}_d$, $\tilde{B}_d$ to the challenger.
7. The challenger measures $\tilde{A}_d^{\theta,U}$ and $\tilde{B}_d^{\theta,V}$ on $\rho_{\mathcal{A},\mathcal{B}}$, obtaining outcomes a' and b'. The reduction wins if $a' = a$ and $b' = b$.

The view of Adv and A, B in $\mathtt{Rand\text{-}Expt}_{\mathsf{Red}[\mathsf{Adv}]}(\lambda', \lambda')$ is computationally indistinguishable from that in $\mathtt{cUE\text{-}Expt}_{\mathsf{Enc},\mathsf{Adv}}(\lambda)$ by security of the PRG. Therefore,

$$\begin{aligned}\Pr\left[\mathtt{cUE\text{-}Expt}_{\mathsf{Enc},\mathsf{Adv}}(\lambda) = 1\right] &\leq \Pr\left[\mathtt{Rand\text{-}Expt}_{\mathsf{Red}[\mathsf{Adv}]}(\lambda', \lambda') = 1\right] + \mathsf{negl}(\lambda)\\ &\leq \frac{1}{2} + \mathsf{negl}(\lambda)\end{aligned}$$

where we have invoked Theorem 9 for the second inequality. □

A direct inspection of the proof of Theorem 14 gives the following.

Corollary 15. *cUE exists unconditionally, for messages of fixed length.*

Proof. The construction is identical to Construction 2, except that Ux and Vx are used directly as one-time pads, without first applying a PRG. Since the messages are of fixed length, one can sample U and V of the appropriate size. The security reduction is analogous to that for Theorem 14. □

3.3 Key testing

For our applications it will be important that our UE and cUE schemes have an additional property that we call *key testing*. This states that there should exist an efficient algorithm Test that determines whether a given secret key is "correct" for a given encryption.

Definition 8. *A triple of efficient quantum algorithms* $(\mathsf{Enc}, \mathsf{Dec}, \mathsf{Test})$ *is an* unclonable encryption scheme with key testing *if it satisfies the following conditions, for all* $\lambda, n \in \mathbb{N}$*:*

- *(Correctness) For all* $\mathsf{sk} \in \{0,1\}^\lambda$ *and* $m \in \{0,1\}^n$*,*

$$\mathsf{Dec}(\mathsf{sk}; \mathsf{Enc}(\mathsf{sk}; m)) \to m.$$

- *(Security) For all polynomial-time adversaries* Adv*,*

$$\Pr\left[\texttt{UE-Expt}_{\mathsf{Enc},\mathsf{Adv}}(\lambda) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

- *(Key testing) For all* $\mathsf{sk}, \mathsf{sk}' \in \{0,1\}^\lambda$ *and* $m \in \{0,1\}^n$*,*

$$\mathsf{Test}(\mathsf{sk}'; \mathsf{Enc}(\mathsf{sk}; m)) \to \delta_{\mathsf{sk}}(\mathsf{sk}')\,,$$

where δ_{sk} *is the indicator function that is* 1 *only at* sk*.*

Definition 9. *A triple of efficient quantum algorithms* $(\mathsf{Enc}, \mathsf{Dec}, \mathsf{Test})$ *is a* coupled unclonable encryption scheme with key testing *if it satisfies the following conditions, for all* $\lambda, n_A, n_B \in \mathbb{N}$*:*

- *(Correctness) For all* $\mathsf{sk}_A, \mathsf{sk}_B \in \{0,1\}^\lambda$ *and* $m_A \in \{0,1\}^{n_A}, m_B \in \{0,1\}^{n_B}$*,*

$$\begin{aligned}&\mathsf{Dec}(0, \mathsf{sk}_A; \mathsf{Enc}(\mathsf{sk}_A, \mathsf{sk}_B; m_A, m_B)) \to m_A \text{ and}\\ &\mathsf{Dec}(1, \mathsf{sk}_B; \mathsf{Enc}(\mathsf{sk}_A, \mathsf{sk}_B; m_A, m_B)) \to m_B.\end{aligned}$$

- *(Security) For all polynomial-time adversaries* Adv*,*

$$\Pr\left[\texttt{cUE-Expt}_{\mathsf{Enc},\mathsf{Adv}}(\lambda) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

- *(Key testing) For all* $\mathsf{sk}_A, \mathsf{sk}_B, \mathsf{sk}' \in \{0,1\}^\lambda$ *and* $m_A \in \{0,1\}^{n_A}, m_B \in \{0,1\}^{n_B}$*,*

$$\mathsf{Test}(\mathsf{sk}'; \mathsf{Enc}(\mathsf{sk}_A, \mathsf{sk}_B; m_A, m_B)) \to \begin{cases} 0, & \mathsf{sk}' = \mathsf{sk}_A \\ 1, & \mathsf{sk}' = \mathsf{sk}_B \\ \perp, & \mathsf{sk}' \notin \{\mathsf{sk}_A, \mathsf{sk}_B\} \end{cases}.$$

We can add key testing to any (coupled) unclonable encryption scheme using qsiO and injective one-way functions. The same construction and proof also work with classical indistinguishability obfuscation in place of qsiO, but we only state the result for qsiO because all of our applications use it.[6] 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 test the validity of a secret key, while at the same time preserving unclonability thanks to the properties of qsiO.

Theorem 16. *If injective one-way functions and* qsiO *exist, then any UE or cUE scheme can be compiled into one with key testing.*

Proof. For simplicity we only describe the compiler and proof for UE. The compiler for cUE is analogous.

Let $(\mathsf{Enc}, \mathsf{Dec})$ be a UE scheme. We build a UE scheme with key testing $(\mathsf{Enc}', \mathsf{Dec}', \mathsf{Test})$ as follows:

$$\begin{aligned}\mathsf{Enc}'(s; m) &= (A, \mathsf{Enc}(As; m), \mathsf{qsiO}(\delta_s))\\ \mathsf{Dec}'(s; (A, \sigma, \tau)) &= \mathsf{Dec}(As; \sigma)\\ \mathsf{Test}(s; (A, \sigma, \tau)) &= \mathsf{Eval}(\tau, s)\,,\end{aligned}$$

where Eval is a universal quantum evaluation circuit, $A \leftarrow \mathbb{F}_2^{\lambda \times 3\lambda}$ is a random matrix sampled by Enc', and the secret key $\mathsf{sk} = s$ is interpreted as a vector in $\mathbb{F}_2^{3\lambda}$.

Correctness and key testing are clear from the construction, so we turn to proving UE security.

[6] Note that qsiO does *not* trivially imply iO since qsiO outputs a quantum implementation.

For a circuit $f : \mathbb{F}_2^{3\lambda} \to \{0,1\}$, let $P[f] : \mathbb{F}_2^{3\lambda} \to \{0,1\}$ be defined by $P[f](x) = (1 - \delta_0(x)) \cdot f(x)$. That is, $P[f]$ is a circuit that outputs $f(x)$ for all x except 0, on which $P[f]$ always outputs 0.

By the qsiO guarantee, $\mathsf{qsiO}(\delta_s) \approx \mathsf{qsiO}(P[\delta_{\{0,s\}}])$. By Zhandry's subspace-hiding obfuscation result [Zha21], which assumes the existence of injective one-way functions, we have $\mathsf{qsiO}(P[\delta_{\{0,s\}}]) \approx \mathsf{qsiO}(P[\delta_T])$ for a random subspace $T \subseteq \mathbb{F}_2^{3\lambda}$ of dimension 2λ that contains s.

Conditioned on T, observe that s still has 2λ bits of min-entropy. Therefore the leftover hash lemma implies that (A, T, As) is $\mathsf{negl}(\lambda)$-close in statistical distance to (A, T, u) for $u \leftarrow \mathbb{F}_2^{\lambda}$, so

$$(A, \mathsf{Enc}(As; m), \mathsf{qsiO}(P[\delta_T])) \equiv (A, \mathsf{Enc}(u; m), \mathsf{qsiO}(P[\delta_T])).$$

Since A and $\mathsf{qsiO}(P[\delta_T])$ can be sampled independently from $\mathsf{Enc}(u; m)$, the UE security of $(\mathsf{Enc}', \mathsf{Dec}', \mathsf{Test})$ follows from the UE security of $(\mathsf{Enc}, \mathsf{Dec})$. □

4 Copy Protection

In Section 2, we showed that qsiO is "'best-possible" copy protection, and thus provides a principled heuristic for copy-protecting any functionality. In this section, our goal is to investigate which functionalities are *provably* copy protected by qsiO. We consider copy protection for three classes of functions, each with slightly different copy protection guarantees. All three security games begin with the challenger sending the adversary a quantum state that represents some copy-protected functionality; the adversary then applies some quantum channel to the received state, and creates a new state on two registers. The three security games differ from this point on:

1. In *decision* copy protection, each part of the adversary is given a uniformly random challenge input x, along with either (a) $f(x)$, or (b) $f(x')$ for a fresh random x'. The task is for both parts to correctly guess which case they are in.
2. In *search* copy protection, each part of the adversary is given a uniformly random input x, and asked to produce y satisfying some condition $\mathsf{Ver}(x, y)$.
3. In copy protection for point functions, each part of the adversary is given both the marked input and a uniformly random input. The task is for both parts to correctly guess which one is the marked input.

Whereas point functions are a particular class of functions, the notions of decision and search copy protection are applicable to many classes of functions. We show that the classes of "decision puncturable" and "search puncturable" programs can be decision copy protected and search copy protected, respectively. Roughly, a *decision puncturable program* does not reveal any information about the function value at the punctured point; a *search puncturable program* may reveal some information, but an efficient adversary cannot compute from it any output that passes some (public or private) verification procedure at the punctured point. We define these notions of puncturable programs precisely in Section 4.1.

Informally, our main results of this section are:

1. Assuming injective OWFs, qsiO decision-copy-protects any decision-puncturable program.
2. Assuming injective OWFs and UE, qsiO search-copy-protects any search-puncturable program.
3. Assuming injective OWFs and UE, qsiO copy-protects point functions.

Remark 4. *For clarity of presentation we assume throughout this section that all challenge input distributions in the copy protection security games are uniform. These results can be generalized to arbitrary distributions with high min-entropy using a randomness extractor.*

4.1 Puncturable programs

A puncturing procedure for a class of programs $\mathcal{F}$ is an efficient algorithm $\mathsf{Puncture}$ that takes as input a description of a program $f \in \mathcal{F}$ and polynomially-many points $x_1, \ldots, x_t \in \text{Domain}(f)$, and outputs the description of a new program $f_{x_1,\ldots,x_t}$. This program should satisfy $f_{x_1,\ldots,x_t}(z) = f(z)$ for all $z \in \text{Domain}(f) \setminus \{x_1, \ldots, x_t\}$ 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'. For instance, in [SW21] it was shown that one-way functions imply the existence of decision puncturable pseudo-random functions.
- For *search puncturing*, we require that no efficient adversary can compute, given f_x, an 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 [BSW16], puncturable signatures were constructed from injective one-way functions and (classical) indistinguishability obfuscation.

Definition 10 (Decision puncturable programs, [SW21])**.** *Let $\mathcal{F} = \{\mathcal{F}_\lambda : \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}\}_{\lambda\in\mathbb{N}}$ be a family of classical circuits. We say that $\mathcal{F}$ is* decision puncturable *if there exists an efficient algorithm* $\mathsf{Puncture}$ *such that, for each $\lambda \in \mathbb{N}$,*

- *For every $f \in \mathcal{F}_\lambda$ and all $\mathsf{poly}(\lambda)$-sized sets $S \subseteq \{0,1\}^{n(\lambda)}$, $\mathsf{Puncture}(f, S)$ outputs a $\mathsf{poly}(\lambda)$-sized circuit f_S such that for all $x \in \{0,1\}^{n(\lambda)} \setminus S$, $f_S(x) = f(x)$.*
- *For every QPT adversary $(\mathsf{Adv}_1, \mathsf{Adv}_2)$ such that $\mathsf{Adv}_1(1^\lambda)$ outputs a set $S \subseteq \{0,1\}^{n(\lambda)}$ and a state σ, if $f \leftarrow \mathcal{F}_\lambda$, $f_S \leftarrow \mathsf{Puncture}(f, S)$, and $\hat{S} \subseteq \{0,1\}^{n(\lambda)}$ is a uniformly random set of the same size as S,*

$$\left|\Pr[\mathsf{Adv}_2(\sigma, f_S, S, f(S)) = 1] - \Pr\left[\mathsf{Adv}_2(\sigma, f_S, S, f(\hat{S}) = 1\right]\right| = \mathsf{negl}(\lambda)\,,$$

where $f(S) = \{f(x) : x \in S\}$, and similarly for $f(\hat{S})$.

We only require search puncturable programs to be puncturable at a single point, because this definition suffices for our applications. This is also the definition given in [BSW16].

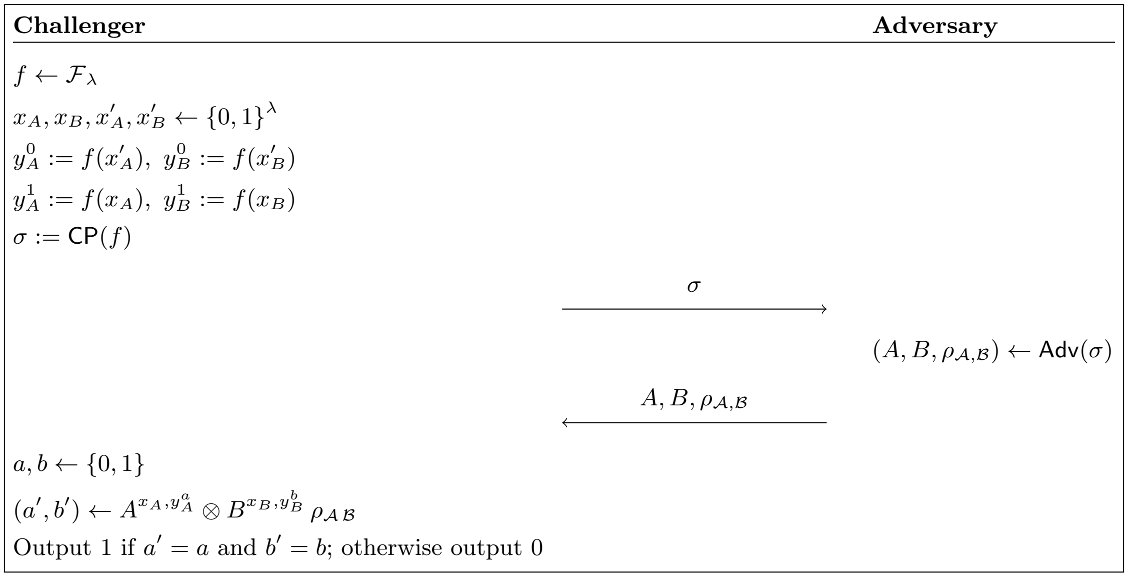
Definition 11 (Search puncturable programs)**.** *Let $\mathcal{F} = \{\mathcal{F}_\lambda : \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}\}_{\lambda\in\mathbb{N}}$ and $\mathsf{Ver} = \{\mathsf{Ver}_\lambda : \mathcal{F}_\lambda \times \{0,1\}^{n(\lambda)} \times \{0,1\}^{m(\lambda)}\}_{\lambda\in\mathbb{N}}$ be families of $\mathsf{poly}(\lambda)$-sized classical circuits. We say that $\mathcal{F}$ is* search puncturable *with respect to* Ver *if, for each $\lambda \in \mathbb{N}$, there exists an efficient algorithm* $\mathsf{Puncture}$ *such that*

- *For every $f \in \mathcal{F}_\lambda$ and all $x \in \{0,1\}^{n(\lambda)}$, $\mathsf{Puncture}(f, x)$ outputs a $\mathsf{poly}(\lambda)$-sized circuit f_x such that for all $x' \in \{0,1\}^{n(\lambda)} \setminus \{x\}$, $f_S(x') = f(x')$.*
- *For every QPT adversary $(\mathsf{Adv}_1, \mathsf{Adv}_2)$ such that $\mathsf{Adv}_1(1^\lambda)$ outputs a point $x \in \{0,1\}^{n(\lambda)}$ and a state σ, if $f \leftarrow \mathcal{F}_\lambda$ and $f_x \leftarrow \mathsf{Puncture}(f, x)$,*

$$\Pr[\mathsf{Ver}_\lambda(f, x, \mathsf{Adv}_2(\sigma, f_x, x)) = 1] = \mathsf{negl}(\lambda)\,.$$

4.2 Decision copy protection

All of the copy protection variants that we define in Section 4 have the same correctness definition (Definition 3). They only differ in their definition of security.



Challenger **Adversary**

$f \leftarrow \mathcal{F}_\lambda$

$x_A, x_B, x'_A, x'_B \leftarrow \{0,1\}^\lambda$

$y_A^0 := f(x'_A),\ y_B^0 := f(x'_B)$

$y_A^1 := f(x_A),\ y_B^1 := f(x_B)$

$\sigma := \mathsf{CP}(f)$

$\xrightarrow{\sigma}$

$(A, B, \rho_{\mathcal{A},\mathcal{B}}) \leftarrow \mathsf{Adv}(\sigma)$

$\xleftarrow{A, B, \rho_{\mathcal{A},\mathcal{B}}}$

$a, b \leftarrow \{0,1\}$

$(a', b') \leftarrow A^{x_A, y_A^a} \otimes B^{x_B, y_B^b}\ \rho_{\mathcal{A}\,\mathcal{B}}$

Output 1 if $a' = a$ and $b' = b$; otherwise output 0

Figure 6: $\texttt{CP-Expt-Decision}_{\mathsf{CP},\mathsf{Adv},\mathcal{F}}(\lambda)$. The challenger samples a function f from $\mathcal{F}_\lambda$ and $x_A, x_B, x'_A, x'_B \leftarrow \{0,1\}^\lambda$. It computes function values for pairs (x_A, x'_A), (x_B, x'_B) to produce challenge pairs, and sends the copy protection $\mathsf{CP}(f)$ to the adversary. The adversary returns a quantum state and quantum circuit descriptions A and B. The challenger measures these with chosen challenge pairs, deciding which function value to use based on random bits a, b. The adversary wins if the measured values match a, b.

Definition 12 (Decision copy protection security). *Let $\mathcal{F} = \{\mathcal{F}_\lambda : \{0,1\}^\lambda \to \{0,1\}^{m(\lambda)}\}_{\lambda\in\mathbb{N}}$ be a family of $\mathsf{poly}(\lambda)$-sized classical circuits. Let CP be a copy protection scheme for $\mathcal{F}$ (as in Definition 3). We say that CP is decision copy protection secure if, for all QPT algorithms Adv, there exists a negligible function $\mathsf{negl}(\lambda)$ such that, for all λ,*

$$\Pr\left[\texttt{CP-Expt-Decision}_{\mathsf{CP},\mathsf{Adv},\mathcal{F}}(\lambda) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda)\,,$$

where $\texttt{CP-Expt-Decision}_{\mathsf{CP},\mathsf{Adv},\mathcal{F}}(\lambda)$ is defined in Figure 6.

Theorem 17. *Let qsiO be a secure qsiO scheme. Let $\mathcal{F}$ be any family of decision-puncturable programs. Then, assuming injective one-way functions exist, qsiO is a copy protection scheme for $\mathcal{F}$ that is decision copy protection secure.*

Proof. The proof proceeds via a sequence of hybrids. Fix an adversary Adv for $\texttt{CP-Expt-Decision}_{\mathsf{qsiO},\mathsf{Adv},\mathcal{F}}(\lambda)$. We will show that the success probability of Adv is preserved across the hybrids, up to $\mathsf{negl}(\lambda)$. We will then argue that the final hybrid is secure by invoking the security of a cUE scheme built from an injective one-way function.

Hybrid 0: The original security game $\texttt{CP-Expt-Decision}_{\mathsf{qsiO},\mathsf{Adv},\mathcal{F}}(\lambda)$, for random a, b.

Hybrid 1: Let $(\mathsf{Enc}, \mathsf{Dec}, \mathsf{Test})$ be any cUE scheme with key-testing. By Theorem 16, such a scheme can be built from qsiO and any injective one-way function. Let $P[g, \sigma]$ be the following program, for a classical circuit g and a quantum state σ (which is supposed to be an output of Enc):

$P[g, \sigma](z)$:

1. Compute $\mathsf{Test}(z; \sigma) \to r$. If $r = \perp$, terminate and output $g(z)$; otherwise continue to step 2.

2. Compute $\mathsf{Dec}(r, z; \sigma)$ and output the result.

Hybrid 1 is same as Hybrid 0, except the challenger sends

$$\mathsf{qsiO}(P[f_{x_A,x_B}, \mathsf{Enc}(x_A, x_B; f(x_A), f(x_B))])$$

as the first message instead of $\mathsf{qsiO}(f)$, where $f_{x_A,x_B} \leftarrow \mathsf{Puncture}(f, \{x_A, x_B\})$. Crucially, notice that $P[f_{x_A,x_B}, \mathsf{Enc}(x_A, x_B; f(x_A), f(x_B))]$ is functionally equivalent to f. More precisely, it can be viewed as quantum implementation of f (as in Definition 1). Then, the fact that Adv wins Hybrid 1 with probability at most $\mathsf{negl}(\lambda)$ higher than in Hybrid 0 follows directly from the security guarantee of qsiO.

Note that if there was a bound on the size of the messages that Enc could encrypt in terms of the length of the secret keys, then we would only be able to encrypt the output of functions $f : \{0,1\}^\lambda \to \{0,1\}^{m(\lambda)}$ with sufficiently small output length $m(\lambda)$. Fortunately, in the definition of cUE we allow the adversary to choose the length of the message (in this case $m(\lambda)$).

Hybrid 2: The challenger samples $\tilde{x}_A, \tilde{x}_B \leftarrow \{0,1\}^\lambda$, sends

$$\mathsf{qsiO}(P[f_{x_A,x_B}, \mathsf{Enc}(x_A, x_B; f(\tilde{x}_A), f(\tilde{x}_B))])$$

as the first message, and uses $y_A^1 := f(\tilde{x}_A), y_B^1 := f(\tilde{x}_B)$ instead of $y_A^1 := f(x_A), y_B^1 := f(x_B)$. Suppose for a contradiction that an adversary Adv had non-negligibly higher success probability in Hybrid 2 than in Hybrid 1. Then, we can use Adv to break the decision puncturing security (Definition 10) of f as follows:

- Sample $x_A, x_B \leftarrow \{0,1\}^\lambda$.
- Receive $(f_{x_A,x_B}, y_A^1, y_B^1)$ where either $(y_A^1, y_B^1) = (f(x_A), f(x_B))$ or $(y_A^1, y_B^1) = (f(\tilde{x}_A), f(\tilde{x}_B))$ for uniformly random $\tilde{x}_A, \tilde{x}_B$.
- Simulate the rest of the Hybrid 1 experiment with Adv using $(f_{x_A,x_B}, y_A^1, y_B^1)$.

When $(y_A^1, y_B^1) = (f(x_A), f(x_B))$, the simulated experiment is distributed as in Hybrid 1. When $(y_A^1, y_B^1) = (f(\tilde{x}_A), f(\tilde{x}_B))$, it is distributed as in Hybrid 2.

Hybrid 3: Let $\tilde{P}[g, \sigma]$ be the following program, for a classical circuit g and a quantum state σ (which is supposed to be an output of Enc).

$\tilde{P}[g, \sigma](z)$:

1. Compute $\mathsf{Test}(z; \sigma) \to r$. If $r = \bot$, terminate and output $g(z)$; otherwise continue to step 2.
2. Compute $\mathsf{Dec}(r, z; \sigma) \to y$ and output $g(y)$.

Hybrid 3 is the same as Hybrid 2, except the challenger sends

$$\mathsf{qsiO}(\tilde{P}[f, \mathsf{Enc}(x_A, x_B; \tilde{x}_A, \tilde{x}_B)])$$

as the first message. That Hybrid 3 has the same success probability as Hybrid 2 follows from the security guarantee of qsiO.

We complete the proof by giving a reduction from an adversary Adv for Hybrid 3 to an adversary Red for the cUE game.

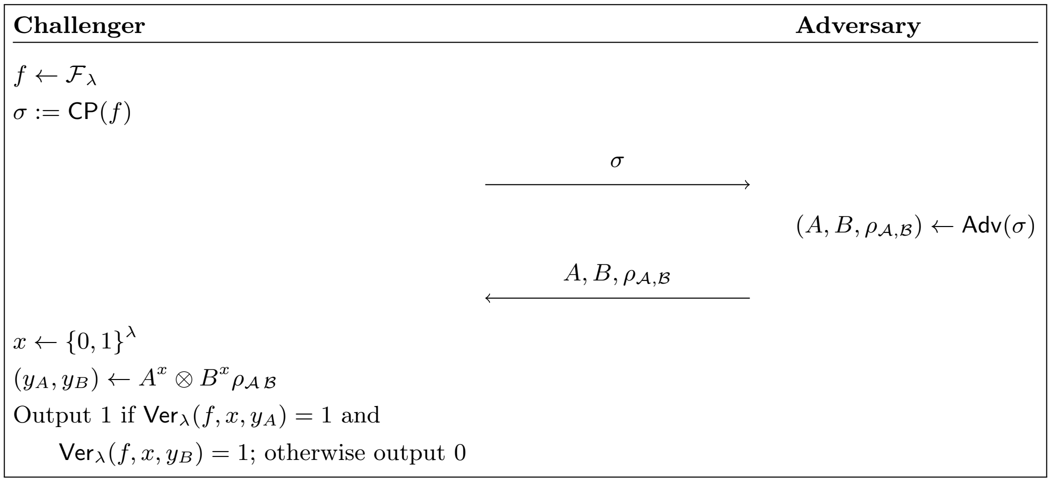
$\texttt{cUE-Expt}_{\mathsf{Enc},\mathsf{Red}[\mathsf{Adv}]}(\lambda)$:

1. Red samples $\tilde{x}_A, \tilde{x}_B \leftarrow \{0,1\}^\lambda$ and sends these to the challenger.
2. The challenger samples $x_A, x_B \leftarrow \{0,1\}^\lambda$ and $m_A^0, m_B^0 \leftarrow \{0,1\}^\lambda$, and lets $m_A^1 = \tilde{x}_A, m_B^1 = \tilde{x}_B$.
3. The challenger samples $a, b \leftarrow \{0,1\}$, computes $\sigma = \mathsf{Enc}(x_A, x_B; m_A^a, m_B^b)$ and sends σ to Red.

4. Red samples $f \leftarrow \mathcal{F}_\lambda$ and sends $\mathsf{qsiO}(\tilde{P}[f,\sigma])$ to Adv. Thus, Red obtains $(A, B, \rho_{\mathcal{A},\mathcal{B}}) \leftarrow \mathsf{Adv}(\mathsf{qsiO}(\tilde{P}[f,\sigma]))$.
5. Let $\tilde{A}^x = A^{x,f(\tilde{x}_A)}$ and $\tilde{B}^x = B^{x,f(\tilde{x}_B)}$.
6. Red sends $(\tilde{A}, \tilde{B}, \rho_{\mathcal{A},\mathcal{B}})$ to the challenger.
7. The challenger measures $\tilde{A}$ and $\tilde{B}$ on $\rho_{\mathcal{A},\mathcal{B}}$, obtaining a' and b'. The reduction wins if $a' = a$ and $b' = b$.

The view of Adv in $\mathtt{cUE\text{-}Expt}_{\mathsf{Enc},\mathsf{Red}[\mathsf{Adv}]}(\lambda)$ is exactly the same as in Hybrid 3. Hence, the cUE security of Enc implies that the success probability of Adv Hybrid 3 is at most $1/2 + \mathsf{negl}(\lambda)$. □

4.3 Search copy protection



Challenger **Adversary**

$f \leftarrow \mathcal{F}_\lambda$

$\sigma := \mathsf{CP}(f)$

$\xrightarrow{\sigma}$

$(A, B, \rho_{\mathcal{A},\mathcal{B}}) \leftarrow \mathsf{Adv}(\sigma)$

$\xleftarrow{A, B, \rho_{\mathcal{A},\mathcal{B}}}$

$x \leftarrow \{0,1\}^\lambda$

$(y_A, y_B) \leftarrow A^x \otimes B^x \rho_{\mathcal{A}\,\mathcal{B}}$

Output 1 if $\mathsf{Ver}_\lambda(f, x, y_A) = 1$ and

$\mathsf{Ver}_\lambda(f, x, y_B) = 1$; otherwise output 0

Figure 7: $\mathtt{CP\text{-}Expt\text{-}Search}_{\mathsf{CP},\mathsf{Adv},\mathcal{F},\mathsf{Ver}}(\lambda)$. The challenger samples a function $f \leftarrow \mathcal{F}_\lambda$ and sends $\mathsf{CP}(f)$ to the adversary. The adversary responds with two quantum circuits A, B and a quantum state $\rho_{\mathcal{A},\mathcal{B}}$. The challenger samples a random string x and measures A, B on $\rho_{\mathcal{A},\mathcal{B}}$ to obtain y_A, y_B. The adversary wins if both y_A and y_B pass the verification using function f, input x, and the measured output.

Definition 13. *Let $\mathcal{F} = \{\mathcal{F}_\lambda : \{0,1\}^\lambda \to \{0,1\}^{m(\lambda)}\}_{\lambda\in\mathbb{N}}$ and $\mathsf{Ver} = \{\mathsf{Ver}_\lambda : \mathcal{F}_\lambda \times \{0,1\}^\lambda \times \{0,1\}^{m(\lambda)}\}_{\lambda\in\mathbb{N}}$ be families of $\mathsf{poly}(\lambda)$-sized classical circuits. Let CP be a copy protection scheme for $\mathcal{F}$ (as in Definition 3). We say that CP is search copy protection secure with respect to Ver if, for all QPT algorithms Adv, there exists a negligible function $\mathsf{negl}(\lambda)$ such that, for all λ,*

$$\Pr\big[\mathtt{CP\text{-}Expt\text{-}Search}_{\mathsf{CP},\mathsf{Adv},\mathcal{F},\mathsf{Ver}}(\lambda) = 1\big] \le \mathsf{negl}(\lambda)\,,$$

where $\mathtt{CP\text{-}Expt\text{-}Search}_{\mathsf{CP},\mathsf{Adv},\mathcal{F},\mathsf{Ver}}(\lambda)$ is defined in Figure 7.

Theorem 18. *Let qsiO be a secure qsiO scheme. Let $\mathcal{F}$ be any family of search-puncturable programs with respect to Ver. Then, assuming injective one-way functions and unclonable encryption exist, qsiO is a copy protection scheme for $\mathcal{F}$ with respect to Ver that is search copy protection secure.*

Proof. Let $(\mathsf{Enc}, \mathsf{Dec}, \mathsf{Test})$ be any UE scheme with key-testing. By Theorem 16, such a scheme can be built from qsiO, UE, and any injective one-way function. Let $P[g,\sigma]$ be the following program, for a classical circuit g and a quantum state σ (which is supposed to be an output of Enc):

$P[g,\sigma](z)$:

1. Compute $\mathsf{Test}(z;\sigma)$. If it rejects, terminate and output $g(z)$; otherwise continue to step 2.
2. Compute $\mathsf{Dec}(z;\sigma)$. If the first λ bits of the decryption are not 0^λ, terminate and output $\perp$; otherwise interpret the remaining bits as the description of a circuit g' and output $g'(z)$.

We reduce from the UE game to $\texttt{CP-Expt-Search}$ as follows:

$\texttt{UE-Expt}_{\mathsf{Enc},\mathsf{Red}[\mathsf{Adv}]}(\lambda)$:

1. Red samples $f \leftarrow \mathcal{F}_\lambda$ and sends $0^\lambda||f$ to the challenger. Let $|f|$ be the number of bits in the description of f.
2. The challenger samples $x \leftarrow \{0,1\}^\lambda$, $m^0 \leftarrow \{0,1\}^{\lambda+|f|}$, and lets $m^1 = 0^\lambda||f$.
3. The challenger samples $c \leftarrow \{0,1\}$, computes $\sigma = \mathsf{Enc}(x;m^c)$ and sends σ to Red.
4. Red sends $\mathsf{qsiO}(P[f,\sigma])$ to Adv.
5. $(A, B, \rho_{\mathcal{A},\mathcal{B}}) \leftarrow \mathsf{Adv}(\mathsf{qsiO}(\tilde{P}[f,\sigma]))$
6. Red samples a random bit r and defines $\tilde{A}, \tilde{B}$ as follows:

   $\tilde{A}^x$: Run $A^x \rightarrow y$. If $\mathsf{Ver}(f,x,y) = 1$, output $a' = 1$; otherwise output $a' = r$.

   $\tilde{B}^x$: Run $B^x \rightarrow y$. If $\mathsf{Ver}(f,x,y) = 1$, output $b' = 1$; otherwise output $b' = r$.
7. Red sends $(\tilde{A}, \tilde{B}, \rho_{\mathcal{A},\mathcal{B}})$ to the challenger.
8. The challenger measures $\tilde{A}$ and $\tilde{B}$ on $\rho_{\mathcal{A},\mathcal{B}}$, obtaining a' and b'. The reduction wins if $a' = b' = c$.

By security of Enc, we have that

$$\Pr\Big[\texttt{UE-Expt}_{\mathsf{Enc},\mathsf{Red}[\mathsf{Adv}]}(\lambda) = 1\Big] \leq \frac{1}{2} + \mathsf{negl}(\lambda). \tag{33}$$

Now suppose that Adv wins $\texttt{CP-Expt-Search}$ with probability ε. We consider the $c = 1$ and $c = 0$ cases separately. For $\tilde{c} \in \{0,1\}$, let $\texttt{UE-Expt}_{\mathsf{Enc},\mathsf{Adv},\tilde{c}}$ denote the $c = \tilde{c}$ version of $\texttt{UE-Expt}_{\mathsf{Enc},\mathsf{Adv}}$.

The $c = 1$ case. By the qsiO guarantee,

$$\mathsf{qsiO}(f) \approx \mathsf{qsiO}(P[f, \mathsf{Enc}(x; 0^\lambda||f)]),$$

so both A and B output y's such that $\mathsf{Ver}(f,x,y) = 1$ with probability at least $\varepsilon - \mathsf{negl}(\lambda)$ in $\texttt{UE-Expt}_{\mathsf{Enc},\mathsf{Red}[\mathsf{Adv}]}(\lambda)$. By construction of $\tilde{A}, \tilde{B}$, it follows that

$$\Pr\Big[\texttt{UE-Expt}_{\mathsf{Enc},\mathsf{Red}[\mathsf{Adv}],1}(\lambda) = 1\Big] \geq \varepsilon + \frac{1-\varepsilon}{2} - \mathsf{negl}(\lambda) = \frac{1+\varepsilon}{2} - \mathsf{negl}(\lambda). \tag{34}$$

The $c = 0$ case. When $c = 0$, Red receives $\mathsf{qsiO}(P[f, \mathsf{Enc}(x; m^0)])$ for $m^0 \leftarrow \{0,1\}^{\lambda+|f|}$. Since m^0 begins with something other than 0^λ with probability $1 - \mathsf{negl}(\lambda)$, the qsiO guarantee implies that

$$\mathsf{qsiO}(P[f, \mathsf{Enc}(x; m^0)]) \approx \mathsf{qsiO}(P[f_x, \mathsf{Enc}(x; m^0)]).$$

By the search puncturing security of f_x (Definition 11), $\mathsf{Adv}(\mathsf{qsiO}(P[f_x, \mathsf{Enc}(x; m^0)]))$ cannot produce a y satisfying $\mathsf{Ver}(f,x,y) = 1$ with probability greater than $\mathsf{negl}(\lambda)$. Therefore, $\tilde{A}^x, \tilde{B}^x$ both output r with probability $1 - \mathsf{negl}(\lambda)$, and

$$\Pr\Big[\texttt{UE-Expt}_{\mathsf{Enc},\mathsf{Red}[\mathsf{Adv}],0}(\lambda) = 1\Big] \geq \frac{1}{2} - \mathsf{negl}(\lambda). \tag{35}$$

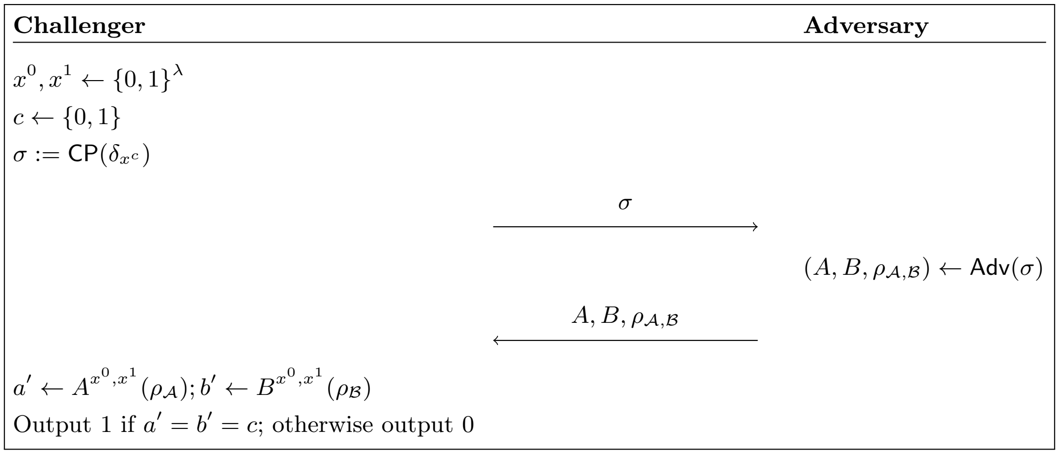
Putting together Inequalities 33, 34, and 35, we find that

$$\frac{1}{2} + \frac{\varepsilon}{4} \leq \frac{1}{2} + \mathsf{negl}(\lambda)$$

and therefore $\varepsilon \leq \mathsf{negl}(\lambda)$. □

4.4 Copy protection for point functions

We show that qsiO and unclonable encryption with key-testing yield copy protection for point functions with perfect correctness. The precise security notion that is achieved here is possibly stronger than those considered in previous works, e.g. [CMP20, AKL$^+$22]. In particular, in the security game that we consider, each of Alice and Bob receive the *same* challenge, which consists of *both* the marked input and a uniformly random input (in a random order), and they have to guess which one is the marked input.



Challenger
Adversary
$x^0, x^1 \leftarrow \{0,1\}^\lambda$
$c \leftarrow \{0,1\}$
$\sigma := \mathsf{CP}(\delta_{x^c})$
σ
$(A, B, \rho_{\mathcal{A},\mathcal{B}}) \leftarrow \mathsf{Adv}(\sigma)$
$A, B, \rho_{\mathcal{A},\mathcal{B}}$
$a' \leftarrow A^{x^0,x^1}(\rho_{\mathcal{A}}); b' \leftarrow B^{x^0,x^1}(\rho_{\mathcal{B}})$
Output 1 if $a' = b' = c$; otherwise output 0

Figure 8: $\texttt{CP-Expt-PtFunc}_{\mathsf{CP},\mathsf{Adv}}(\lambda)$. The challenger samples two random strings $x^0, x^1 \leftarrow \{0,1\}^\lambda$, selects a random bit c, and sends $\mathsf{CP}(\delta_{x^c})$ to the adversary. The adversary generates a state $\rho_{\mathcal{A},\mathcal{B}}$ and quantum circuit descriptions A, B and sends them back. The challenger applies A^{x^0,x^1} to $\rho_{\mathcal{A}}$, giving a', and B^{x^0,x^1} to $\rho_{\mathcal{B}}$, giving b'. The adversary wins if $a' = b' = c$.

Definition 14. *An efficient algorithm* CP *is a copy protection scheme for point functions if, for all efficient adversaries* Adv*,*

$$\Pr\left[\texttt{CP-Expt-PtFunc}_{\mathsf{CP},\mathsf{Adv}}(\lambda) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

Theorem 19. *Assuming injective one-way functions and unclonable encryption,* qsiO *copy-protects point functions.*

A new difficulty arises here that was not present when copy-protecting puncturable programs: For point functions, the copy protection security game is about distinguishing between *inputs*. A naive application of the techniques from Theorems 17 and 18 would therefore involve using UE to encrypt the secret keys. Instead, we will use UE to encrypt *the first bit* at which the two challenges differ.

Proof. Let T be a rank-λ matrix in $\mathbb{F}_2^{\lambda \times (\lambda+1)}$ and σ be a quantum state. For input $z \in \mathbb{F}_2^{\lambda+1}$, we define a program $P_{T,\sigma}$ as follows.

$P_{T,\sigma}(z)$:

1. Compute $\mathsf{Test}(Tz;\sigma)$. If it rejects, terminate and output 0.
2. Compute $x^0 \neq x^1$ such that $Tx^0 = Tx^1 = Tz$. For $c \in \{0,1\}$, let

$$x(c) = \begin{cases} x^0, & (x^0)_i = c \text{ where } i = \min\{j \in [\lambda+1] \mid (x^0)_j \neq (x^1)_j\} \\ x^1, & \text{otherwise.} \end{cases} \tag{36}$$

3. Compute $\mathsf{Dec}(Tz;\sigma) \to c$. If $x(c) = z$, output 1; otherwise output 0.

Observe that $P_{T,\mathsf{Enc}(\mathsf{sk};c)}(z) = \delta_{x(c)}(z)$, so $\mathsf{qsiO}(P_{T,\mathsf{Enc}(\mathsf{sk};c)}) \approx \mathsf{qsiO}(\delta_{x(c)})$.

We now describe a reduction Red that plays $\texttt{UE-Expt}$ using an adversary for the point function copy-protection game. We use a slight variant of $\texttt{UE-Expt}$ where the challenger encrypts a random bit, which is equivalent to the game presented in Figure 4 in the case of single bit messages.

$\texttt{UE-Expt}_{\mathsf{Enc},\mathsf{Red}[\mathsf{Adv}]}(\lambda)$:

1. The challenger samples $\mathsf{sk} \leftarrow \{0,1\}^\lambda$.
2. The challenger samples $c \leftarrow \{0,1\}$, computes $\sigma = \mathsf{Enc}(\mathsf{sk};c)$ and sends σ to Red.
3. Red samples a random rank-λ matrix $T \leftarrow \mathbb{F}_2^{\lambda\times(\lambda+1)}$ and sends $\mathsf{qsiO}(P_{T,\sigma})$ to Adv.
4. $(A, B, \rho_{\mathcal{A},\mathcal{B}}) \leftarrow \mathsf{Adv}(\mathsf{qsiO}(P_{T,\sigma}))$
5. Let $\tilde{A}^{\mathsf{sk}}$ do the following on input $\rho_{\mathcal{A}}$:
   (a) Compute $x^0 \neq x^1$ such that $Tx^0 = Tx^1 = \mathsf{sk}$.
   (b) Run $A^{x^0,x^1}(\rho_{\mathcal{A}}) \to a'$.
   (c) Let $x(\cdot)$ be defined as in (36). Output $\tilde{a} \in \{0,1\}$ such that $x(\tilde{a}) = x^{a'}$.

   Define $\tilde{B}^{\mathsf{sk}}$ similarly but with $B, \rho_{\mathcal{B}}$ instead.
6. Red sends $(\tilde{A}, \tilde{B}, \rho_{\mathcal{A},\mathcal{B}})$ to the challenger.
7. The challenger measures $\tilde{A}$ and $\tilde{B}$ on $\rho_{\mathcal{A},\mathcal{B}}$, obtaining a' and b'. The reduction wins if $a' = b' = c$.

By the security of qsiO,

$$\begin{aligned}&\Pr\left[\texttt{CP-Expt-PtFunc}_{\mathsf{CP},\mathsf{Adv}}(\lambda+1) = 1\right] \\ &\le \Pr\left[\texttt{UE-Expt}_{\mathsf{Enc},\mathsf{Red}[\mathsf{Adv}]}(\lambda) = 1\right] + \mathsf{negl}(\lambda)\end{aligned}$$

and by the security of Enc,

$$\begin{aligned}&\Pr\left[\texttt{UE-Expt}_{\mathsf{Enc},\mathsf{Red}[\mathsf{Adv}]}(\lambda) = 1\right] \\ &\le \frac{1}{2} + \mathsf{negl}(\lambda).\end{aligned}$$

□

References

[Aar09] Scott Aaronson. Quantum copy-protection and quantum money. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity, CCC 2009, Paris, France, 15-18 July 2009*, pages 229–242. IEEE Computer Society, 2009.

[ABDS21] Gorjan Alagic, Zvika Brakerski, Yfke Dulek, and Christian Schaffner. Impossibility of quantum virtual black-box obfuscation of classical circuits. In *Advances in Cryptology–CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part I 41*, pages 497–525. Springer, 2021.

[ABOEM17] Dorit Aharonov, Michael Ben-Or, Elad Eban, and Urmila Mahadev. Interactive proofs for quantum computations. *arXiv preprint arXiv:1704.04487*, 2017.

[AC02] Mark Adcock and Richard Cleve. A quantum goldreich-levin theorem with cryptographic applications. In Helmut Alt and Afonso Ferreira, editors, *STACS 2002*, pages 323–334, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[ADSS17] Gorjan Alagic, Yfke Dulek, Christian Schaffner, and Florian Speelman. Quantum fully homomorphic encryption with verification. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 438–467. Springer, 2017.

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

[AKL$^+$22] Prabhanjan Ananth, Fatih Kaleoglu, Xingjian Li, Qipeng Liu, and Mark Zhandry. 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*, page 212–241, Berlin, Heidelberg, 2022. Springer-Verlag.

[AKL23] Prabhanjan Ananth, Fatih Kaleoglu, and Qipeng Liu. Cloning games: A general framework for unclonable primitives. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part V*, volume 14085 of *Lecture Notes in Computer Science*, pages 66–98. Springer, 2023.

[ALL$^+$21] Scott Aaronson, Jiahui Liu, Qipeng Liu, Mark Zhandry, and Ruizhe Zhang. New approaches for quantum copy-protection. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I*, volume 12825 of *Lecture Notes in Computer Science*, pages 526–555. Springer, 2021.

[AP21] Prabhanjan Ananth and Rolando L. La Placa. Secure software leasing. In Anne Canteaut and François-Xavier Standaert, editors, *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*, volume 12697 of *Lecture Notes in Computer Science*, pages 501–530. Springer, 2021.

[BGI$^+$01] Boaz Barak, Oded Goldreich, Rusell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. On the (im) possibility of obfuscating programs. In *Annual international cryptology conference*, pages 1–18. Springer, 2001.

[BJL$^+$21] Anne Broadbent, Stacey Jeffery, Sébastien Lord, Supartha Podder, and Aarthi Sundaram. Secure software leasing without assumptions. In *Theory of Cryptography Conference*, pages 90–120. Springer, 2021.

[BK21] Anne Broadbent and Raza Ali Kazmi. Constructions for quantum indistinguishability obfuscation. In Patrick Longa and Carla Ràfols, editors, *Progress in Cryptology - LATINCRYPT 2021 - 7th International Conference on Cryptology and Information Security in Latin America, Bogotá, Colombia, October 6-8, 2021, Proceedings*, volume 12912 of *Lecture Notes in Computer Science*, pages 24–43. Springer, 2021.

[BKNY23] James Bartusek, Fuyuki Kitagawa, Ryo Nishimaki, and Takashi Yamakawa. Obfuscation of pseudo-deterministic quantum circuits. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 1567–1578. ACM, 2023.

[BM22] James Bartusek and Giulio Malavolta. Indistinguishability obfuscation of null quantum circuits and applications. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 - February 3, 2022, Berkeley, CA, USA*, volume 215 of *LIPIcs*, pages 15:1–15:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[BSW16] Mihir Bellare, Igors Stepanovs, and Brent Waters. 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*, page 792–821, Berlin, Heidelberg, 2016. Springer-Verlag.

[BV97] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997.

[CLLZ21] Andrea Coladangelo, Jiahui Liu, Qipeng Liu, and Mark Zhandry. Hidden cosets and applications to unclonable cryptography. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021*, pages 556–584, Cham, 2021. Springer International Publishing.

[CMP20] Andrea Coladangelo, Christian Majenz, and Alexander Poremba. Quantum copy-protection of compute-and-compare programs in the quantum random oracle model. *IACR Cryptol. ePrint Arch.*, page 1194, 2020.

[CV22] Eric Culf and Thomas Vidick. A monogamy-of-entanglement game for subspace coset states. *Quantum*, 6:791, 2022.

[GJMZ23] Sam Gunn, Nathan Ju, Fermi Ma, and Mark Zhandry. Commitments to quantum states. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 1579–1588. ACM, 2023.

[JLS18] Zhengfeng Ji, Yi-Kai Liu, and Fang Song. Pseudorandom quantum states. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part III*, volume 10993 of *Lecture Notes in Computer Science*, pages 126–152. Springer, 2018.

[KT23] Srijita Kundu and Ernest Y.-Z. Tan. Device-independent uncloneable encryption, 2023.

[SW21] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: Deniable encryption, and more. *SIAM J. Comput.*, 50(3):857–908, 2021.

[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, oct 2013.

[Zha21] Mark Zhandry. Quantum lightning never strikes the same state twice. or: Quantum money from cryptographic assumptions. *J. Cryptol.*, 34(1), jan 2021.