Copy-Protection from Unclonable Puncturable Obfuscation, Revisited *

Prabhanjan Ananth[1], Amit Behera[2], Zikuan Huang[3], Fuyuki Kitagawa[4,5], Takashi Yamakawa[4,5]

[1]University of California, Santa Barbara
prabhanjan@cs.ucsb.edu
[2]NTT Research
amitbehera1767@gmail.com
[3]Tsinghua University
hzk21@mails.tsinghua.edu.cn
[4]NTT Social Informatics Laboratories
{fuyuki.kitagawa,takashi.yamakawa}@ntt.com
[5]NTT Research Center for Theoretical Quantum Information

October 8, 2025

Abstract

Quantum copy-protection is a functionality-preserving compiler that transforms a classical program into an unclonable quantum program. This primitive has emerged as a foundational topic in quantum cryptography, with significant recent developments. However, characterizing the functionalities that can be copy-protected is still an active and ongoing research direction.

Assuming the existence of indistinguishability obfuscation and learning with errors, we show the existence of copy-protection for a variety of classes of functionalities, including puncturable cryptographic functionalities and subclasses of evasive functionalities. This strictly improves upon prior works, which were either based on the existence of heuristic assumptions [Ananth and Behera CRYPTO'24] or were based on the classical oracle model [Coladangelo and Gunn STOC'24]. Moreover, our constructions satisfy a new and much stronger security definition compared to the ones studied in the prior works. To design copy-protection, we follow the blueprint of constructing copy-protection from unclonable puncturable obfuscation (UPO) [Ananth and Behera CRYPTO'24] and present a new construction of UPO by leveraging the recently introduced techniques from [Kitagawa and Yamakawa TCC'25].

*This article is a merger of the following two works, and subsumes both the works: [ABH25] (https://eprint.iacr.org/2025/1207.pdf) and [KY25a] (https://eprint.iacr.org/2025/1264.pdf).

Contents

1 Introduction

Quantum copy-protection is one of the prominent notions in quantum cryptography. Informally speaking, copy-protection, introduced by Aaronson [Aar09a], is a functionality preserving compiler that turns an efficiently implementable function f into a quantum state ρ_f such that given a single copy of ρ_f, it is computationally infeasible to produce two or more copies that approximately compute f. This primitive exemplifies the power of quantum mechanics in cryptography, belonging to an emerging and rapidly expanding class of *unclonable* primitives. These primitives exploit the fundamental no-cloning principle of quantum mechanics to achieve cryptographic tasks that are provably impossible using classical computational resources. This area has seen remarkable momentum in recent years with many feasibility results [CMP20a, CLLZ21, ALL+21a, AKL+22, LLQZ22, AKL23, CHV23, AB24, CG24b], negative results [AP21, AK22], study of weaker variants [ALL+21b, AP21, BGK+24] and stronger security notions [ÇG24a]. Yet despite this remarkable progress, some important questions remain.

Copy-protectable functionalities. Ever since it was first conceived in 2009, the central research question in the topic of copy-protection has been to understand the classes of functionalities that can be copy-protected.

Ideally, we would want to identify which properties the functionalities need to satisfy in order for them to be copy-protected. It was observed in [Aar09a] that functionalities that are efficiently learnable cannot be copy-protected in any meaningful way. On the other hand, it was believed that unlearnable functionalities can be copy-protected. However, a recent work [AP21] shows that even unlearnability is not a sufficient criteria for copy-protectable functionalities. Following [AP21], the works of [CMP20b, CLLZ21, LLQZ22, AKL+22, AKL23, CHV23] showed that some specific cryptographic functionalities (PRF, decryption and signing functionalities) and point functions can be copy-protected. While these works made significant advances, they failed to shed insight on the type of properties the functionalities needed to satisfy in order for them to copy-protectable[1].

Couple of recent works [AB24, CG24b] took a different route and made a major progress towards characterizing the class of functionalities that can be copy-protected. They identified a property called *puncturability* and argued that as long as a class of functionalities satisfied *puncturability* then they can be copy-protected. However, these works came with a major caveat: their constructions were either based on heuristic assumptions[2] or in the classical oracle model. Moreover, there was also a restriction on the output length of functionalities and only long output[3] puncturable functionalities can be copy-protected. Additionally, [AB24] also showed feasibility of copy-protecting a natural subclass of evasive functions albeit under heuristic assumptions: These works left open the following important question:

Which classes of functionalities can be copy-protected under well-studied assumptions?

Revisiting security definitions. The existing security definitions of copy-protection is quite restrictive and does not capture many attack scenarios. Before we elaborate, let us first recall the typical security definition of copy-protection that has been studied in the literature. The security experiment starts with the (adversarial) efficient Alice receiving a copy-protection of f. It then creates a bipartite state, and shares with two non-communicating efficient adversarial parties Bob and Charlie. Then, Bob and Charlie respectively receive *uniformly random*[4] inputs $x_{\mathcal{B}}$ and $x_{\mathcal{C}}$ and to break security, they need to simultaneously produce $f(x_{\mathcal{B}})$ and $f(x_{\mathcal{C}})$. For security to hold, it is crucial that Bob and Charlie do not receive any other auxiliary information about the function. As an example, all bets are off if Bob learns the output of the function on related inputs (say, $x_B + 1$) or even if $x_{\mathcal{B}}$ is sampled from a different distribution other than the uniform distribution. In some settings, it is crucial that the security needs to be strengthened to allow for Bob and Charlie to also receive some leakage about f. A case study to consider is copy-protecting decryption functionalities. Ideally, we would like to achieve chosen ciphertext attack (CCA) security: even if Bob and Charlie receive access to

[1] [ALL+21a, LLQZ22] did conjecture that any class of functionalities that can be watermarked can also be copy-protected. However, as far as we know, there is no proof of this conjecture.

[2] In particular, they proposed a new conjecture called the *simultaneous inner product conjecture* and based security of copy-protection under this conjecture along with indistinguishability obfuscation. Till date, this conjecture has neither been proven nor disproven.

[3] By long output, here we mean that the output length is $\omega(\log(\lambda))$.

[4] We note that this is not the case for all functionalities. For instance, for point functions, the distribution over $(x_{\mathcal{B}}, x_{\mathcal{C}})$ is not uniform. However, for many other functionalities, such as pseudorandom functions [CLLZ21], signing functionalities [LLQZ22], uniform distribution is the one commonly considered in the literature.

a decryption oracle, they should not be able to decrypt the challenge ciphertext. Unfortunately, the existing security definition of copy-protection does not capture CCA attacks.

This prompts the following question:

> *Can we achieve a stronger definition of copy-protection wherein the security holds even if Bob and Charlie receive non-uniform inputs as well as some leakage about the function?*

1.1 Our Results in a Nutshell

We make progress on the aforementioned questions. We identify interesting classes of functionalities that can be copy-protected under well studied assumptions, strictly improving upon existing results. Our feasibility also satisfy stronger security definitions wherein the security guarantees hold even if the adversarial Bob and Charlie receive non-uniform challenge inputs and additional information about the functionality being copy-protected. Stating our results formally would involve recalling and introducing many definitions which in turn would overwhelm the reader. Instead, we will present the main consequences of our results. The formal result statements (which are more general) would be stated in Section 9.

Puncturable functionalities. A class of (efficiently implementable) functionalities $\mathcal{F}$ is puncturable[5] if it is equipped with an efficient puncturing algorithm $\mathsf{Puncture}$ such that given a function $f \in \mathcal{F}$ and an input x, it produces a punctured function f_x such that even given f_x, it is computationally infeasible to produce $f(x)$ except with probability negligibly close to 2^{-m}, where m is the output length of f. Several puncturable cryptographic functionalities have been studied starting with the seminal works on puncturing pseudorandom functions (see for example [BW13]).

We show the following:

Theorem 1.1 (Informal). *Assuming indistinguishability obfuscation and one-way functions, there exist copy-protection for:*

- *m-bit output puncturable functionalities, where $m = \omega(\log(\lambda))$ and,*
- *1-bit output puncturable functionalities*

Prior works on copy-protecting m-bit output puncturable functionalities, for $m = \omega(\log(\lambda))$, either relied upon heuristic assumptions [AB24] or were based in the classical oracle model [CG24b]. In contrast, our result is based on well-studied assumptions. Moreover, copy-protecting 1-bit output puncturable functionalities in the plain model was not even known prior to our work.

Another (stronger) variant of puncturing security we can consider is *pseudorandomness puncturing security* wherein it is required that even given f_x, $f(x)$ is computationally indistinguishable from uniform. For such functionalities, we can consider a different security definition for copy-protection. Instead of Bob and Charlie receiving $x_{\mathcal{B}}$ and $x_{\mathcal{C}}$ respectively, they instead respectively receive $(x_{\mathcal{B}}, y_{\mathcal{B}})$ and $(x_{\mathcal{C}}, y_{\mathcal{C}})$ respectively. Here, $y_{\mathcal{B}}$ (resp., $y_{\mathcal{C}}$) is either $C(x_{\mathcal{B}})$ (resp., $C(x_{\mathcal{C}})$) or sampled uniformly at random. This variant was first studied by [CLLZ21] in the context of copy-protecting pseudorandom functions. On the other hand, we study this definition for the more general class of pseudorandom puncturable functionalities. We call a copy-protection scheme satisfying this variant to be *pseudorandom* copy-protection.

We show the following:

Theorem 1.2 (Informal). *Assuming indistinguishability obfuscation and one-way functions, there exist pseudorandom copy-protection for pseudorandom functionalities (for all output lengths).*

The prior work on pseudorandom copy-protection[6] [CG24b] relied upon the classical oracle model [BBV24].

[5] In the technical sections, we refer to this as unpredictability-style puncturable security (see Definition 7.1).

[6] [CG24b] referred to it as decision copy-protection.

Multi-point functions. A class of functionalities $\mathcal{F}$ is a class of k-point functions if every function $f : \{0,1\}^n \to \{0,1\}$ in $\mathcal{F}$ satisfies the property that there are exactly k inputs $x_1, \ldots, x_k \in \{0,1\}^n$ such that $f(x_i) = 1$ and for every $x \notin \{x_1, \ldots, x_k\}$, $f(x) = 0$. When $k = 1$, this precisely corresponds to the case of point functions. Copy-protecting point functions has been extensively studied [Aar09b, CMP20a, AKL+22, AKL23, CHV23] and the work of [CHV23] presents a construction of copy-protection for point functions from indistinguishability obfuscation and learning with errors. However, their work does not address the more general problem of k-point functions, for arbitrary k.

We show the following:

Theorem 1.3 (Informal). *Assuming indistinguishability obfuscation and one-way functions, there exist copy-protection for k-point functions, where $k > 1$ and k is polynomial.*

Stronger Security Definition. In addition to the above new results, we also study stronger security definitions for copy-protection. Specifically, we strengthen the security definitions in two ways:

- *Entropic inputs*: in the existing definitions of copy-protection, Bob and Charlie are given uniformly random (independently generated or identical) inputs. However, the existing definitions do not readily generalize to the setting when the inputs are not generated uniformly at random and instead have high entropy. In our definitions (see Section 3.1), Bob and Charlie respectively receive the inputs $x_{\mathcal{B}}$ and $x_{\mathcal{C}}$ such that the inputs are sampled from a high entropic distribution.
- *Oracle access*: We additionally give Bob $\mathcal{B}$ and Charlie $\mathcal{C}$ access to an oracle that computes f, where f is the program being copy-protected. To avoid trivial attacks, the oracle that Bob (resp., Charlie) has access to outputs $\perp$ on the input $x_{\mathcal{B}}$ (resp., $x_{\mathcal{C}}$). Moreover, even Alice $\mathcal{A}$ receives oracle access to C. We note that the oracle access to all the adversaries in our security definitions(see Section 3.1) are with respect to quantum superposition queries (see Section 2 on Page 14 for the formal definition of superposition queries).

To the best of our knowledge, this is the first work that studies the above strengthenings of the security definition of copy-protection. Both the results discussed earlier (Theorem 1.1, Theorem 1.2 and Theorem 1.3) satisfy (**under the additional assumption of learning with errors**) the stronger security definition of copy-protection that incorporates both the above bullets[7].

Blueprint: Copy-protection from UPO. Our main approach is to design copy-protection from a recently introduced primitive called unclonable puncturable obfuscation (UPO) [AB24], which combines the concepts of program obfuscation and unclonability. In particular, they showed that UPO for m-bit output puncturable functionalities (for $m \in \omega(\log(\lambda))$, is also a copy-protection scheme for the same class of functionalities. However, their construction of UPO relied upon heuristic assumptions. The main innovation of our work is to show that (a variant of) UPO can be achieved based on well-studied assumptions, which leads us to the first bullet of Theorem 1.1. Specifically, we construct UPO from indistinguishability obfuscation and one-way functions (and from indistinguishability obfuscation and learning with errors in the high min-entropic setting). Additionally, we present new constructions of copy-protection from UPO, showing that not just m-bit output puncturable functionalities (for $m \in \omega(\log(\lambda))$) but also broader classes of functionalities with varying output sizes can be copy-protected based on the variant of UPO that we construct, which in turn lets us prove Theorem 1.2 and Theorem 1.3.

1.2 Technical Overview

Recap: Unclonable Puncturable Obfuscation (UPO). We first recall the definition of unclonable puncturable obfuscation. UPO consists of two efficient algorithms $(\mathit{Obf}, \mathit{QEval})$. The algorithm Obf takes as input a classical circuit $C : \{0,1\}^{\ell_{\mathsf{inp}}} \to \{0,1\}^{\ell_{\mathsf{out}}}$ and outputs a quantum state $\widetilde{C}$. The algorithm QEval takes as input a quantum state $\widetilde{C}$, an input x and outputs $C(x)$.

[7] In the context of pseudorandom copy-protection, our construction satisfies an even stronger notion that implies both the "identical coin" and "independent coin" versions, with the above-mentioned strengthening. We refer to this stronger notion as the "+" version of the strengthened pseudorandom copy-protection (see Definition 3.4), due to it's similarity with the "+"-notion studied in the context of Single Decryptor Encryption (SDE) introduced in [KY25b].

Couple of security notions have already been considered for UPO [AB24], and we introduce an additional security notion in this work. However, all the (new and old) definitions follow the same template: the adversarial Alice ($\mathcal{A}$) either receives obfuscation of C (when challenge bit $b = 0$) or obfuscation of C punctured at a random point(s) (when $b = 1$). Here, C is adversarially chosen but the random points at which C is punctured is hidden from $\mathcal{A}$. Then, $\mathcal{A}$ computes a bipartite state and shares it with two non-communicating adversaries Bob $\mathcal{B}$ and Charlie ($\mathcal{C}$) who then receive the random points. At this point, $\mathcal{B}$ and $\mathcal{C}$ need to simultaneously output one bit each such that they both together are correlated with b. The different variants depend on (a) how the random points are sampled and, (b) how the output of $\mathcal{B}$ and $\mathcal{C}$ are correlated with b.

In more detail, the experiment is a game between the challenger and the adversary $\mathcal{A}_{\mathsf{cp}} = (\mathcal{A}, \mathcal{B}, \mathcal{C})$.

1. On input 1^λ, $\mathcal{A}$ sends a circuit $C : \{0,1\}^{\ell_{\mathsf{inp}}} \to \{0,1\}^{\ell_{\mathsf{out}}}$ (to be obfuscated) together with two circuits $\mu_{\mathcal{B}} : \{0,1\}^{\ell_{\mathsf{inp}}} \to \{0,1\}^{\ell_{\mathsf{out}}}$ and $\mu_{\mathcal{C}} : \{0,1\}^{\ell_{\mathsf{inp}}} \to \{0,1\}^{\ell_{\mathsf{out}}}$. The role of $\mu_{\mathcal{B}}$ and $\mu_{\mathcal{C}}$ will be clear below.

2. The challenger does the following: it samples $(x_{\mathcal{B}}, x_{\mathcal{C}})$ from some input distribution. It then computes: $Y_{\mathcal{B},0} \leftarrow C(x_{\mathcal{B}})$, $Y_{\mathcal{C},0} \leftarrow C(x_{\mathcal{C}})$, $Y_{\mathcal{B},1} \leftarrow \mu_{\mathcal{B}}(x_{\mathcal{B}})$ and, $Y_{\mathcal{C},1} \leftarrow \mu_{\mathcal{C}}(x_{\mathcal{C}})$. It also samples two bits $(\mathsf{coin}_{\mathcal{B}}, \mathsf{coin}_{\mathcal{C}})$ from some distribution. The challenger then generates obfuscation of $C^*[x_{\mathcal{B}}, x_{\mathcal{C}}, Y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}}, Y_{\mathcal{C},\mathsf{coin}_{\mathcal{C}}}]$, where C^* is a circuit that on input $x_{\mathcal{B}}$, it outputs $Y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}}$ and on input $x_{\mathcal{C}}$, it outputs $Y_{\mathcal{C},\mathsf{coin}_{\mathcal{C}}}$ and all other inputs, it behaves exactly like C.

3. $\mathcal{A}$ creates a bipartite state q over registers $\mathsf{R}_{\mathcal{B}}$ and $\mathsf{R}_{\mathcal{C}}$. Then, $\mathcal{A}$ sends register $\mathsf{R}_{\mathcal{B}}$ to $\mathcal{B}$ and register $\mathsf{R}_{\mathcal{C}}$ to $\mathcal{C}$. The challenger sends $x_{\mathcal{B}}$ and $x_{\mathcal{C}}$ to $\mathcal{B}$ and $\mathcal{C}$, respectively.

4. $\mathcal{B}$ and $\mathcal{C}$ respectively output $\mathsf{coin}'_{\mathcal{B}}$ and $\mathsf{coin}'_{\mathcal{C}}$.

Depending on how $(x_{\mathcal{B}}, x_{\mathcal{C}})$ is sampled and how the winning condition is defined, we can consider three different security definitions:

1. **Identical-secure UPO** [AB24]: $x_{\mathcal{B}} = x_{\mathcal{C}}$ and moreover, $x_{\mathcal{B}}$ is sampled uniformly at random from $\{0,1\}^{\ell_{\mathsf{inp}}}$. $\mathsf{coin}_{\mathcal{B}} = \mathsf{coin}_{\mathcal{C}}$ and moreover, $\mathsf{coin}_{\mathcal{B}}$ is sampled uniformly at random. $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ wins the above experiment if $\mathsf{coin}'_{\mathcal{B}} = \mathsf{coin}'_{\mathcal{C}} = \mathsf{coin}_{\mathcal{B}}$.

2. **Correlated-secure UPO** [AB24]: $x_{\mathcal{B}}$ and $x_{\mathcal{C}}$ are independently sampled from the uniform distribution. On the other hand, $(\mathsf{coin}_{\mathcal{B}}, \mathsf{coin}_{\mathcal{C}})$ is sampled as above. The winning condition is also defined the same as above.

3. **UPO$^+$**: this is a new definition that we introduce in this work. $(x_{\mathcal{B}}, x_{\mathcal{C}})$ is sampled as in correlated-secure UPO. On the other hand, $\mathsf{coin}_{\mathcal{B}}$ and $\mathsf{coin}_{\mathcal{C}}$ are sampled uniformly at random. The winning condition is defined as follows: $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ win if $(\mathsf{coin}'_{\mathcal{B}} \oplus \mathsf{coin}'_{\mathcal{C}}) = (\mathsf{coin}_{\mathcal{B}} \oplus \mathsf{coin}_{\mathcal{C}})$. We also consider a strengthening where $(x_{\mathcal{B}}, x_{\mathcal{C}})$ are sampled from a high-entropic distribution.

Ref. [AB24] constructed identical-secure and correlated-secure UPO from heuristic assumptions. The same work also showed construction of copy-protection for long-output puncturable functionalities from correlated-secure UPO and for subclass of evasive functionalities from identical-secure UPO.

Our Contributions. We show the following:

- UPO$^+$implies correlated-secure UPO (Theorem 3.17).
- We show how to construct UPO$^+$based on indistinguishability obfuscation (Section 5) and depending on whether the inputs $(x_{\mathcal{B}}, x_{\mathcal{C}})$ are sampled from a high-entropic distribution or uniform we either respectively require learning with errors or one-way functions.

Both of them combined provides the backbone to prove Theorems 1.1 and 1.2 and Section 8.1.1. We now discuss the main ideas behind the construction of UPO$^+$.

1.2.1 1.2.1 Construction of UPO$^+$

We first start by recalling important properties about coset states and the monogamy properties associated with them.

Coset states and strong monogamy property with auxiliary input. Our construction relies on coset states and the *computational strong monogamy of entanglement property*. We introduce the relevant notations following [CLLZ21, AKL+22]. For a subspace $A \subseteq \mathbb{F}_2^n$, let

- $A^\perp$ denote its dual,
- $\mathsf{Can}_A(s)$ be the lexicographically least element of $A + s$ (computable in $\mathrm{poly}(n)$ time),
- $\mathsf{CS}(A) := \{\mathsf{Can}_A(s) : s \in \mathbb{F}_2^n\}$.

For canonical representatives $(s,t) \in \mathsf{CS}(A) \times \mathsf{CS}(A^\perp)$, define the *coset state*

$$|A_{s,t}\rangle \;=\; \tfrac{1}{\sqrt{|A|}} \sum_{a \in A} (-1)^{\langle a,t\rangle} |a+s\rangle\,.$$

Applying $H^{\otimes n}$ maps $|A_{s,t}\rangle$ to $\left|A^\perp_{t,s}\right\rangle$.

We often write $i\mathcal{O}(A+s)$ and $i\mathcal{O}(A^\perp+t)$ to denote obfuscated programs by $i\mathcal{O}$ that verify membership in $A+s$ and $A^\perp+t$, respectively.

Previous work [KY25b] has shown that the following *computational strong indistniguishability monogamy property* holds based on its search variant [CLLZ21, CV22]. Consider the following game played by an adversary that consists of a tuple of algorithms $(\mathcal{A},\mathcal{B},\mathcal{C})$.

1. $\mathcal{A}$ is given a coset state $|A_{s,t}\rangle$, where $A \subseteq \mathbb{F}_2^n$ is a uniformly random subspace of dimension $n/2$, and (s,t) are uniformly drawn from $\mathsf{CS}(A) \times \mathsf{CS}(A^\perp)$. Additionally, $\mathcal{A}$ receives two obfuscated membership checking programs $i\mathcal{O}(A+s)$ and $i\mathcal{O}(A^\perp+t)$. $\mathcal{A}$ outputs a bipartite state q over the registers $\mathsf{R}_\mathcal{B}$ and $\mathsf{R}_\mathcal{C}$.
2. $\mathcal{B}$ (resp. $\mathcal{C}$) is given the register $\mathsf{R}_\mathcal{B}$ (resp. $\mathsf{R}_\mathcal{C}$), uniformly random string $r_\mathcal{B} \leftarrow \mathbb{F}_2^n$ (resp. $r_\mathcal{C} \leftarrow \mathbb{F}_2^n$), and the description of A, and outputs a bit $b_\mathcal{B}$ (resp. $b_\mathcal{C}$).
3. The adversary wins if $b_\mathcal{B} \oplus b_\mathcal{C} = \langle r_\mathcal{B}, s\rangle \oplus \langle r_\mathcal{C}, t\rangle$, where $\langle\cdot,\cdot\rangle$ denotes the inner product.

The property guarantees that assuming that $i\mathcal{O}$ is a secure indistinguishability obfuscator and the existence of OWFs, any QPT adversary can win the game with probability at most $1/2 + \mathsf{negl}(\lambda)$.

In this work, we introduce a variant, which we call the *computational strong indistinguishability monogamy property with simulatable auxiliary input*. In this variant, the adversary $(\mathcal{A},\mathcal{B},\mathcal{C})$ may additionally receive auxiliary inputs that depend on A, $r_\mathcal{B}$, and $r_\mathcal{C}$. The property guarantees that any QPT adversary can win the game with probability at most $1/2 + \mathsf{negl}(\lambda)$, provided that $\mathcal{A}$'s auxiliary input is statistically simulatable without using A, $r_\mathcal{B}$, or $r_\mathcal{C}$. The motivation for this variant will become evident in the security proof sketch of our UPO scheme below.

We prove that this variant holds under the assumption that $i\mathcal{O}$ is a secure iO and there exist OWFs. The proof roughly proceeds as follows. We first consider the information-theoretic version of the property, where obfuscated circuits are not provided and the adversary is unbounded. In this setting, we can simulate $\mathcal{A}$'s auxiliary input using the simulator without relying on $(A, r_\mathcal{B}, r_\mathcal{C})$, and subsequently sample the auxiliary inputs of $\mathcal{B}$ and $\mathcal{C}$ from the corresponding conditional distribution given $(A, r_\mathcal{B}, r_\mathcal{C})$ in the second stage. This "reverse sampling" may be computationally inefficient, but efficiency is irrelevant in the information-theoretic setting. Consequently, the additional auxiliary inputs do not give the adversary any advantage, and the problem reduces to the known version without auxiliary inputs. Finally, we lift this argument to the computational setting by applying the technique of [CLLZ21].

Construction. Our construction of UPO is simple and closely resembles the copy-protection scheme in the classical oracle model by Aaronson et al. [ALL+21b].

Let F be a puncturable PRF (PPRF).

- $\mathcal{O}bf(1^\lambda, C)$ Sample a PPRF key K and a coset state $|A_{s,t}\rangle$ where $(s,t) \leftarrow \mathsf{CS}(A) \times \mathsf{CS}(A^\perp)$. Generate obfuscated circuits

$$C_{\mathsf{mc}} \leftarrow i\mathcal{O}(A+s), \qquad C^\perp_{\mathsf{mc}} \leftarrow i\mathcal{O}(A^\perp+t).$$

$P_1[C_{\mathsf{mc}}, C, K](a, x)$

Hardwired constant: A description of a boolean circuits C_{mc} and C, and a string K.

Input: (a, x).

1. If $C_{\mathsf{mc}}(a) = 0$, output $\bot$. Otherwise, go to the next step.
2. Output $\mathsf{F}_K(x) \oplus C(x)$.

Figure 1: The description of the circuit P_1.

$P_2[C^{\bot}_{\mathsf{mc}}, K](a, x)$

Hardwired constant: A description of a boolean circuits $C^{\bot}_{\mathsf{mc}}$ and a string K.

Input: (a, x).

1. If $C^{\bot}_{\mathsf{mc}}(a) = 0$, output $\bot$. Otherwise, go to the next step.
2. Output $\mathsf{F}_K(x)$.

Figure 2: The description of the circuit P_2.

Then generate obfuscated circuits

$$\widetilde{P}_1 \leftarrow i\mathcal{O}\big(P_1[C_{\mathsf{mc}}, C, K]\big), \qquad \widetilde{P}_2 \leftarrow i\mathcal{O}\big(P_2[C^{\bot}_{\mathsf{mc}}, K]\big)$$

where P_1 and P_2 are circuits described in Figure 1 and Figure 2, respectively. Output $\widetilde{C} := (|A_{s,t}\rangle, \widetilde{P}_1, \widetilde{P}_2)$.

- $\mathcal{QE}val\left(\widetilde{C} = (|A_{s,t}\rangle, \widetilde{P}_1, \widetilde{P}_2), x\right)$**:** Starting with $|A_{s,t}\rangle$, coherently evaluate $\widetilde{P}_1$ to obtain $\mathsf{F}_K(x) \oplus C(x)$, uncompute to recover $|A_{s,t}\rangle$, apply $H^{\otimes n}$ to get $\left|A^{\bot}_{t,s}\right\rangle$, coherently evaluate $\widetilde{P}_2$ to obtain $\mathsf{F}_K(x)$, take the XOR of them to get $C(x)$ and output it.

Correctness is immediate. We prove that it satisfies $\mathcal{D}$-UPO$^+$security for any product distribution $\mathcal{D}$ that has sufficiently high min-entropy assuming the security of iO and LWE assumption.[8] Below we outline the ideas for the security proof.

Security proof sketch. We had informally defined UPO$^+$earlier. We will more explicitly define the security experiment below and in particular, we will also incorporate the fact that $\mathcal{B}$'s and $\mathcal{C}$'s inputs are sampled from a high min-entropic distribution. For a product distribution $\mathcal{D} = \mathcal{D}_{\mathcal{B}} \times \mathcal{D}_{\mathcal{C}}$ and a QPT adversary $\mathcal{A}_{\mathsf{cp}} = (\mathcal{A}, \mathcal{B}, \mathcal{C})$, the security experiment of $\mathcal{D}$-UPO$^+$security works as follows.

1. On input 1^λ, $\mathcal{A}$ sends a circuit $C : \{0,1\}^{\ell_{\mathsf{inp}}} \to \{0,1\}^{\ell_{\mathsf{out}}}$ together with two circuits $\mu_{\mathcal{B}} : \{0,1\}^{\ell_{\mathsf{inp}}} \to \{0,1\}^{\ell_{\mathsf{out}}}$ and $\mu_{\mathcal{C}} : \{0,1\}^{\ell_{\mathsf{inp}}} \to \{0,1\}^{\ell_{\mathsf{out}}}$.
2. The challenger does the following.
   - Choose $\mathsf{coin}_{\mathcal{B}} \leftarrow \{0,1\}$, generate $x_{\mathcal{B}} \leftarrow \mathcal{D}_{\mathcal{B}}(1^\lambda)$, $Y_{\mathcal{B},0} \leftarrow C(x_{\mathcal{B}})$, and $Y_{\mathcal{B},1} \leftarrow \mu_{\mathcal{B}}(x_{\mathcal{B}})$.
   - Choose $\mathsf{coin}_{\mathcal{C}} \leftarrow \{0,1\}$, generate $x_{\mathcal{C}} \leftarrow \mathcal{D}_{\mathcal{C}}(1^\lambda)$, $Y_{\mathcal{C},0} \leftarrow C(x_{\mathcal{C}})$, and $Y_{\mathcal{C},1} \leftarrow \mu_{\mathcal{C}}(x_{\mathcal{C}})$.

   The challenger generates $\widetilde{C} := (|A_{s,t}\rangle, \widetilde{P}_1, \widetilde{P}_2) \leftarrow \mathcal{O}bf(1^\lambda, C^*[x_{\mathcal{B}}, x_{\mathcal{C}}, Y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}}, Y_{\mathcal{C},\mathsf{coin}_{\mathcal{C}}}])$ and sends $\widetilde{C}$ to $\mathcal{A}$. Here, $C^*[x_{\mathcal{B}}, x_{\mathcal{C}}, Y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}}, Y_{\mathcal{C},\mathsf{coin}_{\mathcal{C}}}]$ is a circuit that on input $x_{\mathcal{B}}$, outputs $Y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}}$ and on input $x_{\mathcal{C}}$, outputs $Y_{\mathcal{C},\mathsf{coin}_{\mathcal{C}}}$ and on any other input, it behaves exactly like C.
3. $\mathcal{A}$ creates a bipartite state q over registers $\mathsf{R}_{\mathcal{B}}$ and $\mathsf{R}_{\mathcal{C}}$. Then, $\mathcal{A}$ sends register $\mathsf{R}_{\mathcal{B}}$ to $\mathcal{B}$ and register $\mathsf{R}_{\mathcal{C}}$ to $\mathcal{C}$.

[8] While no LWE-based primitive appears in the construction, we rely on lossy functions in the security proof.

$P'_1[\mathsf{C_{mc}}, C, x_{\mathcal{B}}, x_{\mathcal{C}}, K\{x_{\mathcal{B}}, x_{\mathcal{C}}\}, w_{\mathcal{B}}, w_{\mathcal{C}}](a, x)$

Hardwired constant: Descriptions of boolean circuits $\mathsf{C_{mc}}$ andC and string $(x_{\mathcal{B}}, x_{\mathcal{C}}, K\{x_{\mathcal{B}}, x_{\mathcal{C}}\}, w_{\mathcal{B}}, w_{\mathcal{C}})$, where $K\{x_{\mathcal{B}}, x_{\mathcal{C}}\}$ describes a key for PPRF punctured at $\{x_{\mathcal{B}}, x_{\mathcal{C}}\}$.

Input: (a, x).

1. If $\mathsf{C_{mc}}(a) = 0$, output $\bot$. Otherwise, go to the next step.
2. if $x = x_{\mathcal{B}}$, output the hardwired value $w_{\mathcal{B}}$. Otherwise, go to the next step.
3. if $x = x_{\mathcal{C}}$, output the hardwired value $w_{\mathcal{C}}$. Otherwise, go to the next step.
4. Output $\mathsf{F}_{K\{x_{\mathcal{B}}, x_{\mathcal{C}}\}}(x) \oplus C(x)$.

Figure 3: The description of the circuit P'_1.

$P'_2[\mathsf{C}^{\bot}_{\mathsf{mc}}, x_{\mathcal{B}}, x_{\mathcal{C}}, K\{x_{\mathcal{B}}, x_{\mathcal{C}}\}, w_{\mathcal{B}}, w_{\mathcal{C}}](a, x)$

Hardwired constant: A description of a boolean circuit $\mathsf{C}^{\bot}_{\mathsf{mc}}$ and string $(x_{\mathcal{B}}, x_{\mathcal{C}}, K\{x_{\mathcal{B}}, x_{\mathcal{C}}\}, w_{\mathcal{B}}, w_{\mathcal{C}})$, where $K\{x_{\mathcal{B}}, x_{\mathcal{C}}\}$ describes a key for PPRF punctured at $\{x_{\mathcal{B}}, x_{\mathcal{C}}\}$.

Input: (a, x).

1. If $\mathsf{C}^{\bot}_{\mathsf{mc}}(a) = 0$, output $\bot$. Otherwise, go to the next step.
2. if $x = x_{\mathcal{B}}$, output the hardwired value $w_{\mathcal{B}}$. Otherwise, go to the next step.
3. if $x = x_{\mathcal{C}}$, output the hardwired value $w_{\mathcal{C}}$. Otherwise, go to the next step.
4. Output $\mathsf{F}_{K\{x_{\mathcal{B}}, x_{\mathcal{C}}\}}(x)$.

Figure 4: The description of the circuit P'_2.

4. The challenger sends $x_{\mathcal{B}}$ and $x_{\mathcal{C}}$ to $\mathcal{B}$ and $\mathcal{C}$, respectively.
5. $\mathcal{B}$ and $\mathcal{C}$ respectively output $\mathsf{coin}'_{\mathcal{B}}$ and $\mathsf{coin}'_{\mathcal{C}}$. The challenger outputs 1 if $\mathsf{coin}'_{\mathcal{B}} \oplus \mathsf{coin}'_{\mathcal{C}} = \mathsf{coin}_{\mathcal{B}} \oplus \mathsf{coin}_{\mathcal{C}}$ otherwise outputs 0.

Our goal is to show that the probability that the above experiment returns 1 is negligibly close to $1/2$. To prove this, we consider the following sequence of hybrids.

First, by the standard puncturing technique, we can replace the obfuscations

$$\widetilde{P}_1 \leftarrow i\mathcal{O}\big(P_1[\mathsf{C_{mc}}, C^*[x_{\mathcal{B}}, x_{\mathcal{C}}, Y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}}, Y_{\mathcal{C},\mathsf{coin}_{\mathcal{C}}}], K]\big), \qquad \widetilde{P}_2 \leftarrow i\mathcal{O}\big(P_2[\mathsf{C}^{\bot}_{\mathsf{mc}}, K]\big)$$

with

$$iO(P'_1[\mathsf{C_{mc}}, C, x_{\mathcal{B}}, x_{\mathcal{C}}, K\{x_{\mathcal{B}}, x_{\mathcal{C}}\}, y_{\mathcal{B}} \oplus Y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}}, y_{\mathcal{C}} \oplus Y_{\mathcal{C},\mathsf{coin}_{\mathcal{C}}}]),$$
$$i\mathcal{O}(P'_2[\mathsf{C}^{\bot}_{\mathsf{mc}}, x_{\mathcal{B}}, x_{\mathcal{C}}, K\{x_{\mathcal{B}}, x_{\mathcal{C}}\}, y_{\mathcal{B}}, y_{\mathcal{C}}]),$$

where $K\{x_{\mathcal{B}}, x_{\mathcal{C}}\}$ is a punctured PPRF key punctured on $\{x_{\mathcal{B}}, x_{\mathcal{C}}\}$, $y_{\mathcal{B}}, y_{\mathcal{C}}$ are uniformly random, and P'_1 and P'_2 are the circuits described in Figures 3 and 4, respectively.

Next, since $y_{\mathcal{C}}$ is uniformly random, the distribution of $y_{\mathcal{C}} \oplus Y_{\mathcal{C},\mathsf{coin}_{\mathcal{C}}}$ is also uniformly random. Therefore, by relabeling $y_{\mathcal{C}} \oplus Y_{\mathcal{C},\mathsf{coin}_{\mathcal{C}}}$ as $y_{\mathcal{C}}$, the obfuscated circuits can be rewritten as follows:

$$iO(P'_1[\mathsf{C_{mc}}, C, x_{\mathcal{B}}, x_{\mathcal{C}}, K\{x_{\mathcal{B}}, x_{\mathcal{C}}\}, y_{\mathcal{B}} \oplus Y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}}, y_{\mathcal{C}}]),$$
$$i\mathcal{O}(P'_2[\mathsf{C}^{\bot}_{\mathsf{mc}}, x_{\mathcal{B}}, x_{\mathcal{C}}, K\{x_{\mathcal{B}}, x_{\mathcal{C}}\}, y_{\mathcal{B}}, y_{\mathcal{C}} \oplus Y_{\mathcal{C},\mathsf{coin}_{\mathcal{C}}}]).$$

Here, we introduce a trick similar to one used in [KY25b]: we replace the challenge bits $\mathsf{coin}_{\mathcal{B}}$ and $\mathsf{coin}_{\mathcal{C}}$ with $\mathsf{coin}_{\mathcal{B}} \oplus \langle r_{\mathcal{B}}, s\rangle$ and $\mathsf{coin}_{\mathcal{C}} \oplus \langle r_{\mathcal{C}}, t\rangle$ for uniformly random $r_{\mathcal{B}}$ and $r_{\mathcal{C}}$, respectively. Note that this does not change the distribution of the challenge bits, since they are uniformly random. With this modification, the obfuscated circuits become

$$iO(P'_1[\mathsf{C_{mc}}, C, x_{\mathcal{B}}, x_{\mathcal{C}}, K\{x_{\mathcal{B}}, x_{\mathcal{C}}\}, y_{\mathcal{B}} \oplus Y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}} \oplus \langle r_{\mathcal{B}}, s\rangle}, y_{\mathcal{C}}]),$$

$$i\mathcal{O}(P_2'[C_{\mathsf{mc}}^{\perp}, x_{\mathcal{B}}, x_C, K\{x_{\mathcal{B}}, x_C\}, y_{\mathcal{B}}, y_C \oplus Y_{C,\mathsf{coin}_C \oplus \langle r_{\mathcal{B}}, s\rangle}]).$$

The winning condition is now

$$\mathsf{coin}'_{\mathcal{B}} \oplus \mathsf{coin}'_C \;=\; \mathsf{coin}_{\mathcal{B}} \oplus \mathsf{coin}_C \oplus \langle r_{\mathcal{B}}, s\rangle \oplus \langle r_C, t\rangle.$$

What this means is that if we can somehow simulate the hybrid knowing $\mathsf{coin}_{\mathcal{B}}, \mathsf{coin}_C$, but without knowing s or t, then we may obtain a reduction to the computational strong indistinguishability monogamy property. Indeed, whenever the above condition holds, we have

$$(\mathsf{coin}'_{\mathcal{B}} \oplus \mathsf{coin}_{\mathcal{B}}) \oplus (\mathsf{coin}'_C \oplus \mathsf{coin}_C) \;=\; \langle r_{\mathcal{B}}, s\rangle \oplus \langle r_C, t\rangle.$$

However, the issue is that simulating the hybrid for the adversary still requires knowledge of s and t, or at least $\langle r_{\mathcal{B}}, s\rangle$ and $\langle r_C, t\rangle$, since these bits determine which of $Y_{\mathcal{B},0}$ or $Y_{\mathcal{B},1}$ (resp. $Y_{C,0}$ or $Y_{C,1}$) is embedded into the obfuscated circuits.

[KY25b] resolved a similar issue in the context of SDE based on the following observation: the first (resp. second) obfuscated circuit outputs a non-$\perp$ value only when the first input a belongs to $A + s$ (resp. $A^{\perp} + t$). But given such an element, if one knows the description of the subspace A, one can efficiently recover s (resp. t). Therefore, instead of choosing between $Y_{\mathcal{B},0}$ and $Y_{\mathcal{B},1}$ (resp. $Y_{C,0}$ and $Y_{C,1}$) depending on $\langle r_{\mathcal{B}}, s\rangle$ (resp. $\langle r_C, t\rangle$), we can let the obfuscated circuits choose the appropriate one by deriving s (resp. t) whenever needed. This idea eliminates the need for explicitly knowing s or t when generating the obfuscated circuits.

While this approach is sufficient in the context of SDE as in [KY25b], we still face a problem: the above idea requires embedding the description of A and $r_{\mathcal{B}}$ or r_C into the obfuscated circuits. However, in the security experiment of UPO, the obfuscated circuits must be generated in the first stage, before the state is split. On the other hand, in the experiment for the computational strong indistinguishability monogamy property, the description of A and $r_{\mathcal{B}}, r_C$ are only given to the adversary in the second stage. This creates a discrepancy. Even relying on the simulatable auxiliary-input variant of the monogamy property, almost no information on $(A, r_{\mathcal{B}}, r_C)$ is available in the first stage. Thus, we need an additional trick to simulate the obfuscated circuits without knowing $(A, r_{\mathcal{B}}, r_C)$.

To implement this idea, we want to ensure that whenever the first obfuscated circuit takes $x_{\mathcal{B}}$ as input (resp. the second circuit takes x_C), it can internally derive $(A, r_{\mathcal{B}})$ (resp. (A, r_C)), which is then used to compute $\langle r_{\mathcal{B}}, s\rangle$ (resp. $\langle r_C, t\rangle$) as above. At the same time, the circuits should not leak any information about A or $(r_{\mathcal{B}}, r_C)$, so that the reduction to the computational strong indistinguishability monogamy property with simulatable auxiliary inputs remains valid. To reconcile these conflicting requirements, we rely on lossy functions.

Roughly, we take a lossy function F in injective mode and a universal hash function h, and embed $(F(x_{\mathcal{B}}),\ h(x_{\mathcal{B}}) \oplus A\|r_{\mathcal{B}})$ into the first obfuscated circuit and $(F(x_C),\ h(x_C) \oplus A\|r_C)$ into the second. The circuits are then modified to recognize $x_{\mathcal{B}}$ and x_C using $F(x_{\mathcal{B}})$ and $F(x_C)$, and to recover $A\|r_{\mathcal{B}}$ and $A\|r_C$ by XORing with $h(x_{\mathcal{B}})$ and $h(x_C)$, respectively. This addresses the first requirement. For the second requirement, we instead choose F from its lossy mode, where the image size is much smaller than the input space. Then, when $x_{\mathcal{B}}$ and x_C have sufficiently large min-entropy compared to the length of $A\|r_{\mathcal{B}}$ and $A\|r_C$, they retain high min-entropy even given $F(x_{\mathcal{B}})$ and $F(x_C)$. By the leftover hash lemma, $h(x_{\mathcal{B}})$ and $h(x_C)$ are statistically close to uniform, so almost no information about $(A, r_{\mathcal{B}}, r_C)$ is embedded in the circuits. As the two modes of the lossy function are computationally indistinguishable, both requirements are virtually satisfied. More precisely, we begin by considering the injective mode, which allows us to modify the obfuscated circuit using the security of iO. We then rely on the indistinguishability of the two modes to switch to the lossy mode.

Here, it is essential to rely on the simulatable auxiliary input variant of the computational strong indistinguishability monogamy property. The reason is as follows. The values $(F(x_{\mathcal{B}}),\ h(x_{\mathcal{B}}) \oplus A\|r_{\mathcal{B}},\ F(x_C),\ h(x_C) \oplus A\|r_C)$ are embedded in the obfuscated circuits. While these values are statistically simulatable without using $(A, r_{\mathcal{B}}, r_C)$ in lossy mode as argued above, the reduction must still provide the UPO adversary with consistent $x_{\mathcal{B}}$ and x_C, sampled from the corresponding conditional distribution in the second stage. The auxiliary input variant of the computational strong indistinguishability monogamy property precisely captures this scenario: we can view $(h(x_{\mathcal{B}}) \oplus A\|r_{\mathcal{B}},\ h(x_C) \oplus A\|r_C)$ as $\mathcal{A}$'s auxiliary input, $x_{\mathcal{B}}$ as $\mathcal{B}$'s auxiliary input, and x_C as $\mathcal{C}$'s auxiliary input. Therefore, the security of the UPO scheme reduces to this property.

Alternative security proof without LWE. Our security proof outlined above relies on LWE since our construction makes use of lossy functions. We also give an alternative proof that solely relies on the existence of iO and OWF, thus

avoiding the LWE assumption, in the special case where the distributions of $x_{\mathcal{B}}$ and $x_{\mathcal{C}}$ are uniformly random. This proof relies on a primitive called key-robust non-committing encryption, which we introduce in this work and construct from any keyed injective OWFs, which exist assuming iO and OWFs.

1.2.2 Copy-Protection from UPO: Two Approaches

We discuss two approaches to construct copy-protection from UPO. Before we discuss the approaches in more detail, we will first recall the security definitions of both UPO and copy-protection:

- In the security definition of copy-protection, $\mathcal{A}$ receives as input copy-protection of a circuit C sampled from some distribution. In the splitting phase, $\mathcal{A}$ then computes a bipartite state which it then shares with $\mathcal{B}$ and $\mathcal{C}$. After the splitting phase, $\mathcal{B}$ receives $x_{\mathcal{B}}$ and $\mathcal{C}$ receives $x_{\mathcal{C}}$, where $(x_{\mathcal{B}}, x_{\mathcal{C}})$ is sampled from some distribution. In the pseudorandom copy-protection definition, $\mathcal{B}$ and $\mathcal{C}$ additionally respectively receive $y_{\mathcal{B}}$ and $y_{\mathcal{C}}$, which is either the true function value or sampled uniformly at random. Moreover, for both (standard) copy-protection and pseudorandom copy-protection, we can consider the stronger (called the *oracular*) definition wherein: (a) $(x_{\mathcal{B}}, x_{\mathcal{C}})$ are sampled from a high-entropic distribution and, (b) $\mathcal{B}$ (resp., $\mathcal{C}$) has oracle access to C punctured at $x_{\mathcal{B}}$ (resp., $x_{\mathcal{C}}$). Moreover, $\mathcal{A}$ also receives oracle access to C.
- In the security definition of UPO, $\mathcal{A}$ receives as input obfuscation of either C (if challenge bit $b = 0$) or C punctured at $(x_{\mathcal{B}}, x_{\mathcal{C}})$ (if challenge bit $b = 1$), where $(x_{\mathcal{B}}, x_{\mathcal{C}})$ is chosen from some distribution. Here, C is chosen by $\mathcal{A}$. After the splitting phase, $\mathcal{B}$ receive as input $x_{\mathcal{B}}$ and $\mathcal{C}$ receive as input $x_{\mathcal{C}}$.

The construction of copy-protection from UPO in both the approaches is the same: to copy-protect a circuit C, obfuscate C using the UPO scheme. The evaluation algorithm of the copy-protection scheme is the same as the evaluation algorithm of the UPO scheme.

Towards reducing copy-protection to UPO, the main step is to move from copy-protecting the circuit C to an intermediate hybrid wherein $\mathcal{A}$ receives copy-protection of C (with probability 0.5) and copy-protection of C punctured at the inputs $(x_{\mathcal{B}}, x_{\mathcal{C}})$ (with probability 0.5). Once we move to this intermediate hybrid we can then invoke the security of UPO to complete the proof.

Approach 1. Pseudorandom Puncturable Functionalities (Definition 7.1). First, we start by considering copy-protecting pseudorandom puncturable functionalities. We briefly discussed the definition of pseudorandom puncturable functionalities in Section 1.1.

At a high level, the proof proceeds as follows: (for now, let us not consider the oracular definition mentioned above)

- In the first step, we consider the copy-protection experiment played between $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ and the challenger.
- In the second step, instead of copy-protecting C, instead copy-protect $\widetilde{C}$ defined as follows: on input $x_{\mathcal{B}}$, it outputs $y_{\mathcal{B}} = C(x_{\mathcal{B}})$ and on input $x_{\mathcal{C}}$, it outputs $y_{\mathcal{C}} = C(x_{\mathcal{C}})$ and on all other inputs, it behaves exactly like C. By iO security[9], this is indistinguishable from the previous step.
- In the third step,
  - With probability 0.5, $\mathcal{A}$ gets copy-protection of $\widetilde{C}$ (as defined in the above bullet) and,
  - With probability 0.5, it gets copy-protection of $\widetilde{C}'$, where $\widetilde{C}'$ is the same as $\widetilde{C}$ except that $y_{\mathcal{B}}$ and $y_{\mathcal{C}}$ are sampled uniformly at random.

  The indistinguishability of the second and third step follows from the pseudorandom puncturing security.
- Finally, in the fourth step,
  - With probability 0.5, $\mathcal{A}$ gets copy-protection of C (as against $\widetilde{C}$ in the above step)
  - With probability 0.5, it gets copy-protection of $\widetilde{C}'$, where $\widetilde{C}'$ is as defined above.

[9] Due to the composition theorem of [AB24], we can assume without loss of generality, an UPO scheme satisfies iO security.

Since the functionality of $\widetilde{C}$ is exactly the same as the functionality of C, the indistinguishability of the third and fourth step follows from iO security.

Once we reach the fourth step, we can then invoke the UPO security to complete the proof since the fourth step corresponds to the security experiment of UPO.

While this proof template can be suitably expanded to get a full fledged proof, this is inadequate for the stronger oracular definition. The reason being that since $\mathcal{A}$, $\mathcal{B}$ and, $\mathcal{C}$ now can compute C on additional inputs owing to the fact that they have access to an oracle that computes C; note that the oracles that $\mathcal{B}$ and $\mathcal{C}$ receive access to disallow queries respectively on $x_{\mathcal{B}}$ and $x_{\mathcal{C}}$. Fortunately, we leverage standard techniques [BBBV97b] combined with the observation that in the UPO security experiment, $\mathcal{B}$ and $\mathcal{C}$ know the circuit description C in the clear, to show that even the stronger oracular definition is satisfied.

Approach 2. Progmaskable Functionalities (Definition 8.4). We consider another class of functionalities, called *progmaskable* functionalities. We consider a simplified version of the definition below. At a high level, a function class is progmaskable (with respect to appropriate distributions over the circuit class and input distributions) if the following two distributions are computationally indistinguishable:

- (C, x), where C is sampled from a distribution over the circuit class and x is sampled from a distribution over the input class.
- $(\widetilde{C}, x)$, where x is sampled from some input distribution (potentially different from the one considered in the first bullet) and $\widetilde{C}$ is sampled as follows: first sample C as in the first bullet, then sample y from some distribution that depends on (x, C) and then $\widetilde{C}$ is defined to be the same as C except on input x, it outputs y. In other words, $\widetilde{C}$ is obtained by *programming* the output of C on x to be y.

  The distribution over y is important: we consider the half-correct distribution where $y = C(x)$ with probability 0.5 and y is sampled uniformly at random subject to the condition that $y \neq C(x)$, i.e., y is correct with probability 0.5.

The indistinguishability of the above two distributions implies that given the programmed $\widetilde{C}$, it is both computationally hard to find the true output of C on x, and also that (C, x) can be *masked* as $(\widetilde{C}, x)$.

In the more general definition (Definition 8.4), we consider multiple inputs $x_1, \ldots, x_\ell$ instead of just one input. Progmaskable functionalities capture a large class of functionalities, such as pseudorandom puncturable functionalities (Theorem 8.18) and k-point functions (Theorem 8.9)[10].

The proof of security for copy-protecting progmaskable functionalities proceeds as follows:

- In the first step, we consider the copy-protection experiment played between $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ and the challenger. That is, $\mathcal{A}$ receives copy-protection of C, $\mathcal{B}$ receives $x_{\mathcal{B}}$ and $\mathcal{C}$ receives $x_{\mathcal{C}}$.
- In the second step, $\mathcal{A}$ receives copy-protection of $\widetilde{C}$, where $\widetilde{C}$ is a circuit that behaves the same as C except that on input $x_{\mathcal{B}}$, it outputs $y_{\mathcal{B}}$ and on input $x_{\mathcal{C}}$, it outputs $y_{\mathcal{C}}$. Here, $y_{\mathcal{B}}$ and $y_{\mathcal{C}}$ are sampled from the half-correct distribution. That is,
  - With probability 0.5, $y_{\mathcal{B}} = C(x_{\mathcal{B}})$ and $y_{\mathcal{C}} = C(x_{\mathcal{C}})$,
  - With probability 0.5, $y_{\mathcal{B}}$ and $y_{\mathcal{C}}$ are sampled from the uniform distribution subject to the condition that $y_{\mathcal{B}} \neq C(x_{\mathcal{B}})$ and $y_{\mathcal{C}} \neq C(x_{\mathcal{C}})$.

  Moreover, $\mathcal{B}$ receives $x_{\mathcal{B}}$ and $\mathcal{C}$ receives $x_{\mathcal{C}}$.

  The indistinguishability of the first and the second step follows from the progmaskable security.
- In the third and final step, we do the following:
  - With probability 0.5, $\mathcal{A}$ receives copy-protection of C,

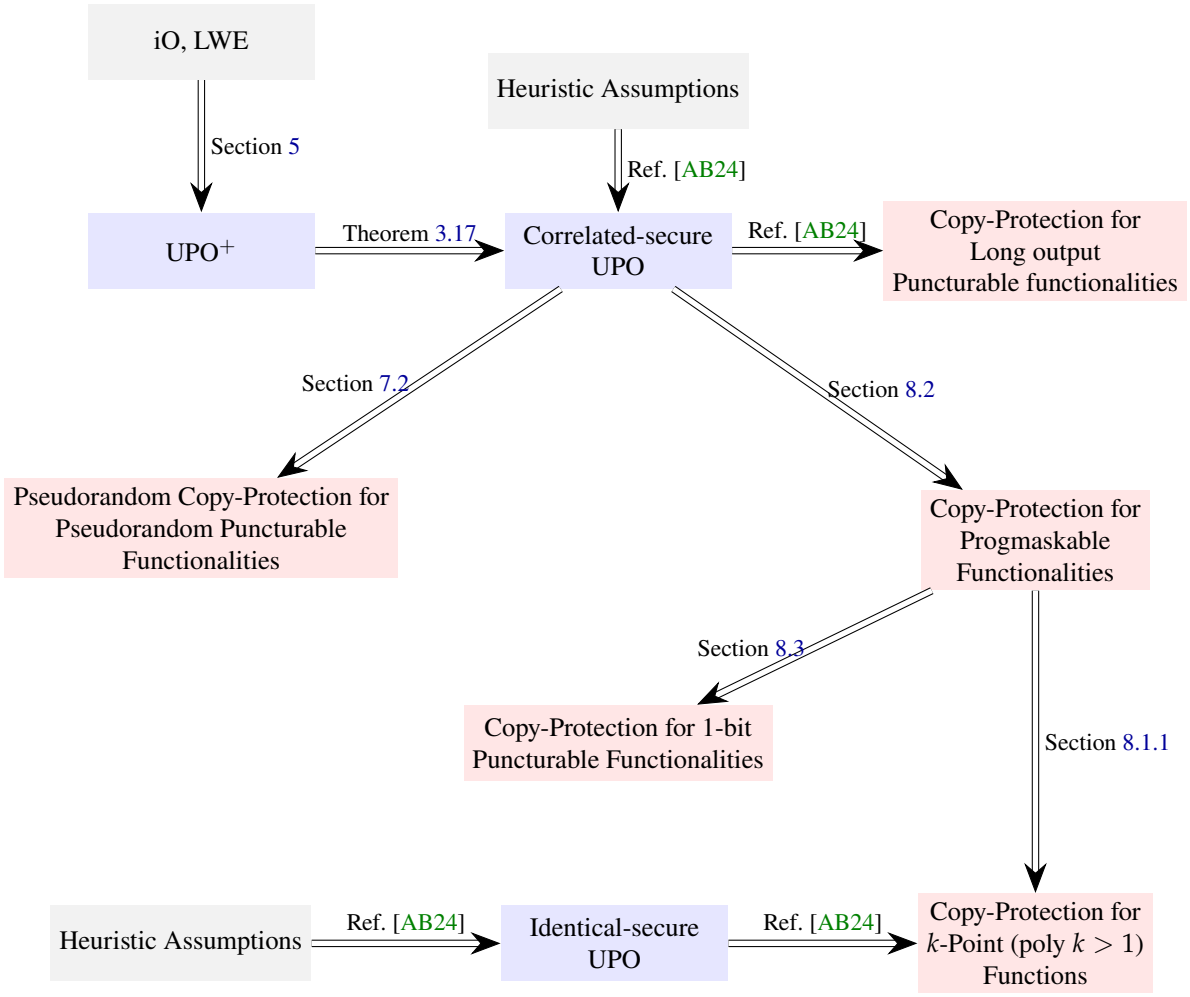
[10] In fact, they capture an even larger class of functionalities called preimage-sampleable evasive functionalities introduced in [AB24], see Theorem 8.6 for the formal statement.

– With probability 0.5, $\mathcal{A}$ receives copy-protection of $\widetilde{C}$ where $\widetilde{C}$ is the same as in the second step except that $y_{\mathcal{B}}$ and y_C are *always* sampled from the uniform distribution subject to the condition that $y_{\mathcal{B}} \neq C(x_{\mathcal{B}})$ and $y_C \neq C(x_C)$. In particular, $y_{\mathcal{B}}$ and y_C are not sampled from the uniform distribution.

We can now immediately reduce the third step to the security of UPO, which completes the proof.

As before this proof template does not work for the oracular definition and we need to invoke quantum query lower bounds [BBBV97b] in order to show that the stronger definition is satisfied.

As mentioned before, progmaskable functionalities capture pseudorandom puncturable functionalities (Theorem 8.18). As a result, (unpredictable-style) copy-protection of progmaskable functionalities implies (unpredictable-style) copy-protection of 1-bit output (unpredictable-style) puncturable functionalities (leading us to the second bullet of Theorem 1.1). [11] The same approach yields non-trivial bound of $1/2$-security but not full security for copy-protection of m-bit output pseudorandom puncturable functionalities, for $m > 1$, because we only get copy-protection of progmaskable functionalities with bound $1/2$ on adversarial success probability, which amounts to full security in 1-bit output setting but may not provide full security for $m > 1$. We leave the question of strengthening the copy-protection of progmaskable m-bit functionalities to get a bound of $1/2^m$ on adversarial success probability as an open question.



[11] We also use the fact that unpredictable-style and pseudorandom-style puncturing security are equivalent in the 1-bit output setting (see Lemma 7.3).

1.2.3 Concurrent and Independent Work

A couple of works [ÇG25a, ÇG25b] concurrently also address the problem of copy-protecting pseudorandom puncturable functionalities under high min-entropic distributions. However, their approach is different, and in particular, they do not go through the route of unclonable puncturable obfuscation.

2 Preliminaries

Notations and conventions. In this paper, standard math or sans serif font stands for classical algorithms (e.g., C or Gen) and classical variables (e.g., x or pk). Calligraphic font stands for quantum algorithms (e.g., $\mathcal{Gen}$) and calligraphic font and/or the bracket notation for (mixed) quantum states (e.g., q or $|\psi\rangle$).

Let $[\ell]$ denote the set of integers $\{1, \cdots, \ell\}$, λ denote a security parameter, and $y := z$ denote that y is set to be, defined, or substituted by z. For any finite set X, we use $|X|$ to denote the size of X, and for any $\ell \in \mathbb{N}$ such that $\ell \leq |X|$, we use $\binom{X}{\ell}$ to denote the set of all ℓ-element subsets of X. For a finite set X and a distribution D, $x \leftarrow X$ or $x \xleftarrow{\$} X$ denotes selecting an element from X uniformly at random, $x \leftarrow D$ denotes sampling an element x according to D. Moreover, $\mathsf{Uniform}_{\mathbf{X}}$ denotes the uniformly random distribution on the set X. Let $y \leftarrow \mathsf{A}(x)$ and $y \leftarrow \mathcal{A}(\chi)$ denote assigning to y the output of a probabilistic or deterministic algorithm A and a quantum algorithm $\mathcal{A}$ on an input x and χ, respectively. When we explicitly show that A uses randomness r, we write $y \leftarrow \mathsf{A}(x;r)$. PPT and QPT algorithms stand for probabilistic polynomial-time algorithms and polynomial-time quantum algorithms, respectively. Let negl denote a negligible function, $\mathsf{non\text{-}negl}$ denote a non-negligible function, and poly denote a positive polynomial. For a string $x \in \{0,1\}^n$, $x[i]$ is its i-th bit. For strings $x, y \in \{0,1\}^n$, $\langle x, y\rangle$ denotes $\bigoplus_{i\in[n]} x[i] \cdot y[i]$. We often use the similar notation for vectors $x, y \in \mathbb{F}_2^n$ by identifying elements of $\mathbb{F}_2$ and $\{0,1\}$ in the natural manner. For reals x, y and $\delta > 0$, we write $x \approx_\delta y$ to mean $|x - y| \leq \delta$.

We use gray sans serif font (e.g., X) to stand for a quantum register. For a quantum state q over registers R_1 and R_2, we write $q[\mathsf{R}_1]$ to denote the portion of the state on register R_1. Similarly, for a pure quantum state $|\psi\rangle$ over registers R_1 and R_2, we write $|\psi\rangle_{\mathsf{R}_1}$ to denote the portion of the state on register R_1. For quantum states q and q', $\|q - q'\|_{tr}$ denotes their trace distance. For any classical circuit $C : \{0,1\}^n \to \{0,1\}^m$, and a subset of inputs $S \subset \{0,1\}^n$, we denote $C \setminus S$ to be the circuit that on every input $x \in S$, outputs $\perp$, and on every input $x \in \{0,1\}^n \setminus S$, outputs $C(x)$. Finally, for any classical circuit $C : \{0,1\}^n \to \{0,1\}^m$, and a quantum algorithm $\mathcal{A}$, we say that a QPT $\mathcal{A}$ has oracle access to C, if $\mathcal{A}$ has two special quantum query registers, a n-qubit input register I and a m-qubit output register O, such that during it's execution, $\mathcal{A}$ can make polynomially many queries to C in superposition, where each superposition query corresponds to the following: $\mathcal{A}$ sends the registers I and O to the oracle, on which the oracle applies the unitary version of C, $\mathcal{U}_C := \sum_{x\in\{0,1\}^n} |x\rangle\langle x|_{\mathsf{I}} \otimes \sum_{y\in\{0,1\}^m} |y \oplus C(x)\rangle\langle y|_{\mathsf{O}}$, and sends the registers back to $\mathcal{A}$.

2.1 Quantum Query Lower Bound

Theorem 2.1 ([BBBV97a]). *Let $\mathcal{A}$ be an adversary with oracle access to $H : \{0,1\}^m \to \{0,1\}^n$ that makes at most T queries. Define $|\phi_i\rangle$ as the global state after $\mathcal{A}$ makes i queries, and $W_y(|\phi_i\rangle)$ as the sum of squared amplitudes in $|\phi_i\rangle$ of terms in which $\mathcal{A}$ queries H on input y. Let $\epsilon > 0$ and let $F \subseteq [0, T-1] \times \{0,1\}^m$ be a set of time-string pairs such that $\sum_{(i,y)\in F} W_y(|\phi_i\rangle) \leq \epsilon^2/T$.*

Let H' be an oracle obtained by reprogramming H on inputs $(i, y) \in F$ to arbitrary outputs. Define $|\phi'_i\rangle$ as above for H'. Then, $\mathsf{TD}(|\phi_T\rangle\langle\phi_T|, |\phi'_T\rangle\langle\phi'_T|) \leq \epsilon/2$.

2.2 Min-Entropy, Universal Hash Functions, and Leftover Hash Lemma

The min-entropy of a classical random variable X is defined as

$$H_\infty(X) := -\log\Big(\max_x \Pr[X = x]\Big).$$

We often say that a distribution D over classical strings has min-entropy p if a classical random variable X according to the distribution D has min-entropy p.

The average min-entropy of a classical random variable X conditioned on another classical random variable Z is defined as

$$\widetilde{H}_\infty(X \mid Z) := -\log\Big(\mathbb{E}_{z\leftarrow Z} \max_x \Pr[X = x \mid Z = z]\Big).$$

The chain rule states that if Z is a random variable over ℓ-bit strings, then

$$\widetilde{H}_\infty(X \mid Z) \geq H_\infty(X) - \ell.$$

A family of functions $\mathcal{H} : \mathcal{X} \to \mathcal{Y}$ is said to be universal, if it holds that for all $x, x' \in \mathcal{X}$ such that $x \neq x'$, we have

Lemma 2.2 (Leftover Hash Lemma). *Let $\mathcal{H} : \mathcal{X} \to \mathcal{Y}$ be a universal hash function family. For (possibly correlated) classical random variables X and Z such that $\widetilde{H}_\infty(X \mid Z) \geq \log|\mathcal{Y}| + 2\log\left(\frac{1}{\epsilon}\right)$, the statistical distance between*

$$\{h, h(X), Z)\} \quad \textit{and} \quad \{h, Y, Z)\}$$

is at most ϵ, where $h \leftarrow \mathcal{H}$ and Y is uniformly random in $\mathcal{Y}$.

2.3 Classical Cryptographic Primitives

Indistinguishability obfuscation

Definition 2.3 (Indistinguishability Obfuscator [BGI+12]). *A PPT algorithm $i\mathcal{O}$ is a secure indistinguishabilitu obfuscation (iO) for a classical circuit class $\{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ if it satisfies the following two conditions.*

Functionality preserving: *For any security parameter $\lambda \in \mathbb{N}$, circuit $C \in \mathcal{C}_\lambda$, and input x, we have that*

$$\Pr\big[C'(x) = C(x) \mid C' \leftarrow i\mathcal{O}(C)\big] = 1 \ .$$

Indistinguishability: *For any QPT* Sampler *that outputs two circuits $C_0, C_1 \in \mathcal{C}_\lambda$ along with a quantum auxiliary information aux and QPT distinguisher $\mathcal{A}$, the following holds:*

If $\Pr\big[\forall x\ C_0(x) = C_1(x) \wedge |C_0| = |C_1| \mid (C_0, C_1, aux) \leftarrow \mathsf{Sampler}(1^\lambda)\big] > 1 - \mathsf{negl}(\lambda)$, *then we have*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{io}}_{i\mathcal{O},\mathcal{D}}(\lambda) := \Big|&\Pr\Big[\mathcal{A}(i\mathcal{O}(C_0), aux) = 1 \mid (C_0, C_1, aux) \leftarrow \mathsf{Sampler}(1^\lambda)\Big] \\ &- \Pr\Big[\mathcal{A}(i\mathcal{O}(C_1), \mathsf{aux}) = 1 \mid (C_0, C_1, aux) \leftarrow \mathsf{Sampler}(1^\lambda)\Big]\Big| \leq \mathsf{negl}(\lambda).\end{aligned}$$

There are several candidates of secure iO for polynomial-size classical circuits against quantum adversaries [BGMZ18, CHVW19, AP20, BDGM20, WW21, GP21, DQV+21, BDGM22, BDJ+24, HJL25, CLW25].

Pseudorandom functions.

Definition 2.4 (Puncturable PRF). *A puncturable PRF (PPRF) is a tuple of algorithms* $\mathsf{PPRF} = (\mathsf{PRF.Gen}, \mathsf{F}, \mathsf{Puncture})$ *where $\{\mathsf{F}_K : \{0,1\}^{\ell_{\mathsf{inp}}} \to \{0,1\}^{\ell_{\mathsf{out}}} \mid K \in \{0,1\}^\lambda\}$ is a PRF family and satisfies the following two conditions.*

Punctured correctness: *For any polynomial-sized set $S \subseteq \{0,1\}^{\ell_{\mathsf{inp}}}$ and any $x \in \{0,1\}^{\ell_{\mathsf{inp}}} \setminus S$, it holds that*

$$\Pr\Big[\mathsf{F}_K(x) = \mathsf{F}_{K\{S\}}(x) \mid K \leftarrow \mathsf{PRF.Gen}(1^\lambda), K\{S\} \leftarrow \mathsf{Puncture}(K, S)\Big] = 1.$$

Pseudorandom at punctured point: *For any polynomial-sized set $S \subseteq \{0,1\}^{\ell_{\mathsf{inp}}}$ and any QPT distinguisher $\mathcal{A}$, it holds that*

$$|\Pr\Big[\mathcal{A}(\mathsf{F}_{K\{S\}}, \{\mathsf{F}_K(x_i)\}_{x_i\in S}) \to 1\Big] - \Pr\Big[\mathcal{A}(\mathsf{F}_{K\{S\}}, (\mathcal{U}_{\ell_{\mathsf{out}}})^{|S|}) \to 1\Big]| \leq \mathsf{negl}(\lambda),$$

where $K \leftarrow \mathsf{PRF.Gen}(1^\lambda)$, $K\{S\} \leftarrow \mathsf{Puncture}(K, S)$ and $\mathcal{U}_{\ell_{\mathsf{out}}}$ denotes the uniform distribution over $\{0,1\}^{\ell_{\mathsf{out}}}$.

If $S = \{x_1^*, x_2^*, \ldots, x_n^*\}$ for some $n \in \mathbb{N}$, we simply denote the punctured key by $K\{x_1^*, x_2^*, \ldots, x_n^*\}$ instead of $K\{\{x_1^*, x_2^*, \ldots, x_n^*\}\}$.

Goldwasser-Goldreich-Micali tree-based construction of PRFs (GGM PRF) [GGM86] from OWF yield puncturable PRFs where the size of the punctured key grows polynomially with the size of the set S being punctured [BW13, BGI14, KPTZ13].

Keyed injective one-way functions.

Definition 2.5 (Keyed Injective $\mathcal{D}$-One-Way Functions). *Let $\mathcal{D}$ be a distribution over $\{0,1\}^{\ell_{\mathsf{inp}}}$. A keyed injective $\mathcal{D}$-one-way function IOWF is given by a PPT algorithm Gen that takes 1^λ as input, and outputs a description of a classical-polynomial-time-computable function $\mathsf{F} : \{0,1\}^{\ell_{\mathsf{inp}}} \to \{0,1\}^{\ell_{\mathsf{out}}}$. It satisfies the following properties.*

Injectivity: *With overwhelming probability over the choice of $\mathsf{F} \leftarrow \mathsf{Gen}(1^\lambda)$, F is injective.*

$\mathcal{D}$-One-Wayness: *For all QPT algorithms $\mathcal{A}$,*

$$\Pr\left[\mathsf{F}(x') = \mathsf{F}(x) \mid \mathsf{F} \leftarrow \mathsf{Gen}(1^\lambda), x \leftarrow \mathcal{D}, x' \leftarrow \mathcal{A}(1^\lambda, \mathsf{F}, \mathsf{F}(x))\right] = \mathsf{negl}(\lambda).$$

For the uniform distribution $\mathcal{U}$, keyed injective $\mathcal{U}$-one-way functions exist assuming the existence of iO and OWFs [BPW16]. Moreover, assuming the LWE assumption, for any distribution $\mathcal{D}$ that has min-entropy at least λ^c for some constant $c > 0$, there exist keyed injective $\mathcal{D}$-one-way functions. This directly follows from the construction of lossy functions from LWE [PW11] (Theorem 2.7).

Lossy functions.

Definition 2.6 (Lossy Functions). *A collection of $(\ell_{\mathsf{inp}}, \ell_{\mathsf{loss}})$-lossy functions LF is given by a pair of two PPT algorithms $(\mathsf{Gen}_{\mathsf{inj}}, \mathsf{Gen}_{\mathsf{loss}})$.*

- *The injective function generation algorithm $\mathsf{Gen}_{\mathsf{inj}}$ takes the security parameter 1^λ as an input, and outputs a description of a classical-polynomial-time-computable injective function F_{inj} over the domain $\{0,1\}^{\ell_{\mathsf{inp}}}$.*
- *The lossy function generation algorithm $\mathsf{Gen}_{\mathsf{loss}}$ takes the security parameter 1^λ as an input, and outputs a description of a classical-polynomial-time-computable function F_{loss} over the domain $\{0,1\}^{\ell_{\mathsf{inp}}}$ whose image size is at most $2^{\ell_{\mathsf{inp}} - \ell_{\mathsf{loss}}}$.*

LF *satisfies the following mode indistinguishability.*

Mode Indistinguishability: *We have $F_{\mathsf{inj}} \stackrel{\mathsf{c}}{\approx} F_{\mathsf{loss}}$, where $F_{\mathsf{inj}} \leftarrow \mathsf{Gen}_{\mathsf{inj}}(1^\lambda)$ and $F_{\mathsf{loss}} \leftarrow \mathsf{Gen}_{\mathsf{loss}}(1^\lambda)$.*

It is known that lossy functions can be constructed assuming the hardness of LWE [PW11]. The following formulation is adapted from [CLLZ21].

Theorem 2.7 ([PW11]). *Assuming the polynomial hardness of LWE, for any constant $c > 0$, there exists a collection of $(\ell_{\mathsf{inp}}, \ell_{\mathsf{loss}})$-lossy functions such that $\ell_{\mathsf{inp}} - \ell_{\mathsf{loss}} \leq \ell_{\mathsf{inp}}^c$.*

2.4 Quantum Goldreich-Levin

Ananth et al. [AKY25] extended the quantum Goldreich-Levin lemma [AC02, CLLZ21] to the non-local setting, where two non-communicating algorithms with potentially entangled quantum inputs are involved.[12].

[12] A slightly weaker version is also proven in [AKL23]

Lemma 2.8 (Simultaneous Quantum Goldreich-Levin [AKY25]). *There exists a QPT oracle algorithm $\mathcal{Ext}$ that satisfies the following. Let $n \in \mathbb{N}$, $x_{\mathcal{B}}, x_{\mathcal{C}} \in \{0,1\}^n$, $\epsilon \in [0, 1/2]$, and q be a quantum state over registers $\mathsf{R}_{\mathcal{B}}$ and $\mathsf{R}_{\mathcal{C}}$. Let $\mathcal{B}$ (resp. $\mathcal{C}$) be a quantum algorithm that takes $\mathsf{R}_{\mathcal{B}}$ (resp. $\mathsf{R}_{\mathcal{C}}$) and an n-bit string $r_{\mathcal{B}}$ (resp. $r_{\mathcal{C}}$) as input, and outputs a bit. If we have*

$$\Pr\left[b_{\mathcal{B}} \oplus b_{\mathcal{C}} = \langle r_{\mathcal{B}}, x_{\mathcal{B}}\rangle \oplus \langle r_{\mathcal{C}}, x_{\mathcal{C}}\rangle : \begin{array}{l} r_{\mathcal{B}}, r_{\mathcal{C}} \leftarrow \{0,1\}^n \\ b_{\mathcal{B}} \leftarrow \mathcal{B}(q[\mathsf{R}_{\mathcal{B}}], r_{\mathcal{B}}) \\ b_{\mathcal{C}} \leftarrow \mathcal{C}(q[\mathsf{R}_{\mathcal{C}}], r_{\mathcal{C}}) \end{array}\right] \geq \frac{1}{2} + \epsilon,$$

then, we have

$$\Pr\left[\mathcal{Ext}(\mathcal{B}, q[\mathsf{R}_{\mathcal{B}}]) \to x_{\mathcal{B}} \wedge \mathcal{Ext}(\mathcal{C}, q[\mathsf{R}_{\mathcal{C}}]) \to x_{\mathcal{C}}\right] \geq 4\epsilon^2.$$

where $\mathcal{B}$ and $\mathcal{C}$ in the input of $\mathcal{Ext}$ mean their descriptions.

Lemma 2.8 was originally proven in [AKY25, Lemma 5] for the special case of $x_{\mathcal{B}} = x_{\mathcal{C}}$. However, their proof does not rely on this assumption, and the same argument holds for the more general case where $x_{\mathcal{B}} \neq x_{\mathcal{C}}$, as observed in [KY25b].

2.5 Useful Lemma

The following lemma is implicit in [AKY25]

Lemma 2.9. *Let E_0, E_1, E_2 be events and $0 < \delta < 1$. If $\Pr[E_0] = 1/2$, $\Pr[E_1 \mid E_0] \approx_\delta \Pr[E_1 \mid \neg E_0]$, and $\Pr[E_2 \mid E_0] \approx_\delta \Pr[E_2 \mid \neg E_0]$, then it holds that*

$$\Pr[E_1 \wedge E_2] + \Pr[\neg E_1 \wedge \neg E_2] \approx_\delta \Pr[E_1 \wedge E_2 \mid E_0] + \Pr[\neg E_1 \wedge \neg E_2 \mid \neg E_0].$$

Proof. By the assumptions, we have

$$\Pr[E_1 \mid E_0] + \Pr[\neg E_1 \mid \neg E_0] \approx_\delta 1$$

and

$$\Pr[E_2 \mid E_0] + \Pr[\neg E_2 \mid \neg E_0] \approx_\delta 1.$$

Then we have

$$\begin{aligned}
&\Pr[E_1 \wedge E_2 \mid E_0] + \Pr[\neg E_1 \wedge \neg E_2 \mid \neg E_0] \\
&= (\Pr[E_1 \mid E_0] - \Pr[E_1 \wedge \neg E_2 \mid E_0]) + (\Pr[\neg E_1 \mid \neg E_0] - \Pr[\neg E_1 \wedge E_2 \mid \neg E_0]) \\
&\approx_\delta 1 - \Pr[E_1 \wedge \neg E_2 \mid E_0] - \Pr[\neg E_1 \wedge E_2 \mid \neg E_0])) \\
&= 1 - (\Pr[\neg E_2 \mid E_0] - \Pr[\neg E_1 \wedge \neg E_2 \mid E_0]) - (\Pr[E_2 \mid \neg E_0] - \Pr[E_1 \wedge E_2 \mid \neg E_0]) \\
&\approx_\delta \Pr[\neg E_1 \wedge \neg E_2 \mid E_0]) + \Pr[E_1 \wedge E_2 \mid \neg E_0].
\end{aligned}$$

Thus, we have

$$\begin{aligned}
&\Pr[E_1 \wedge E_2] + \Pr[\neg E_1 \wedge \neg E_2] \\
&= \frac{1}{2}(\Pr[E_1 \wedge E_2 \mid E_0] + \Pr[E_1 \wedge E_2 \mid \neg E_0]) + \frac{1}{2}(\Pr[\neg E_1 \wedge \neg E_2 \mid E_0] + \Pr[\neg E_1 \wedge \neg E_2 \mid \neg E_0]) \\
&= \frac{1}{2}(\Pr[E_1 \wedge E_2 \mid E_0]) + \Pr[\neg E_1 \wedge \neg E_2 \mid \neg E_0]) + \frac{1}{2}(\Pr[\neg E_1 \wedge \neg E_2 \mid E_0] + \Pr[E_1 \wedge E_2 \mid \neg E_0]) \\
&\approx_\delta \frac{1}{2}(\Pr[\neg E_1 \wedge \neg E_2 \mid E_0] + \Pr[E_1 \wedge E_2 \mid \neg E_0]) + \frac{1}{2}(\Pr[\neg E_1 \wedge \neg E_2 \mid E_0] + \Pr[E_1 \wedge E_2 \mid \neg E_0]) \\
&= \Pr[\neg E_1 \wedge \neg E_2 \mid E_0]) + \Pr[E_1 \wedge E_2 \mid \neg E_0]
\end{aligned}$$

□

3 Definitions

3.1 Definition of Copy-Protection

First, we review the existing definition, which we refer to as unpredictability-style copy-protection.

Definition 3.1 (Oracular Unpredictability-Style Copy-Protection). *Let* $\mathsf{Circ} = \{C_k\}_{k\in\mathcal{K}_\lambda}$ *be a family of keyed circuits with* n*-bit inputs and* m*-bit outputs. A copy-protection scheme for* Circ *is a pair of two algorithms* $\mathsf{CP} = (\mathit{CopyProtect}, \mathcal{E}\mathit{val})$.

$\mathit{CopyProtect}(1^\lambda, C_k) \to \widetilde{C}$**:** *The copy-protection algorithm is a QPT algorithm that takes a security parameter* 1^λ *and a circuit* $C_k \in \mathsf{Circ}$*, and outputs a quantum state* $\widetilde{C}$.

$\mathcal{E}\mathit{val}(\widetilde{C}, x) \to y$**:** *The evaluation algorithm is a QPT algorithm that takes a quantum state* $\widetilde{C}$ *and an input* $x \in \{0,1\}^n$*, and outputs a value* y.

Evaluation correctness: *For every* $k \in \mathcal{K}_\lambda$ *and* $x \in \{0,1\}^n$*, we have*

$$\Pr\left[\mathcal{E}\mathit{val}(\widetilde{C}, x) = C_k(x) \;\middle|\; \widetilde{C} \leftarrow \mathit{CopyProtect}(1^\lambda, C_k)\right] = 1 - \mathsf{negl}(\lambda).$$

Oracular Unpredictability-Style Copy-Protection Anti-Piracy: *For a distribution* $\mathcal{D}$ *over* $\mathcal{K}_\lambda \times \{0,1\}^n \times \{0,1\}^n$*, consider the following game* $\mathsf{Exp}^{\mathsf{unp\text{-}cp}}_{\mathsf{CP},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda)$ *between the challenger and an adversary* $\mathcal{A}_{\mathsf{cp}} = (\mathcal{A}, \mathcal{B}, \mathcal{C})$ *below.*

1. *The challenger chooses* $(k, x_1^*, x_2^*) \leftarrow \mathcal{D}$*, generates* $\widetilde{C} \leftarrow \mathit{CopyProtect}(1^\lambda, C_k)$*, and sends* $\widetilde{C}$*, as well as oracle access to* C_k *to* $\mathcal{A}$ *(see Section 2 (on Page 14) for the formal definiton of oracle access).*
2. $\mathcal{A}^{C_k}$ *creates a bipartite state* q *over registers* $\mathsf{R}_\mathcal{B}$ *and* $\mathsf{R}_\mathcal{C}$*. Then,* $\mathcal{A}$ *sends register* $\mathsf{R}_\mathcal{B}$ *to* $\mathcal{B}$ *and register* $\mathsf{R}_\mathcal{C}$ *to* $\mathcal{C}$.
3. *The challenger sends* x_1^* *and oracle access to* $C_k \setminus \{x_1^*\}$ *to* $\mathcal{B}$ *and sends* x_2^* *and oracle access to* $C_k \setminus \{x_2^*\}$ *to* $\mathcal{C}$*, where for any circuit* C_k *and set* $S \subset \boldsymbol{X}$*,* $C_k \setminus S$ *denotes the circuit that for any input* $x \in S$ *outputs* $\bot$*, and for every* $x \in \boldsymbol{X} \setminus S$ *outputs* $C_k(x)$.
4. $\mathcal{B}^{C_k\setminus\{x_1^*\}}$ *and* $\mathcal{C}^{C_k\setminus\{x_2^*\}}$ *respectively output* $y_\mathcal{B}$ *and* $y_\mathcal{C}$*. The challenger outputs* 1 *if* $y_\mathcal{B} = C_k(x_1^*) \wedge y_\mathcal{C} = C_k(x_2^*)$ *otherwise outputs* 0.

We say that CP *satisifies* **oracular-$\mathcal{D}$-unpredictability-style-CP anti-piracy**, *or simply,* **oracular-$\mathcal{D}$-CP anti-piracy**, *for* Circ *if for any QPT adversary* $\mathcal{A}_{\mathsf{cp}}$*, there exists a trivial adversary* $\mathcal{A}^{triv}_{\mathsf{cp}} \in \mathsf{Triv}\left(\mathsf{Exp}^{\mathsf{unp\text{-}cp}}_{\mathsf{CP},\mathcal{D}}\right)$ *and a negligible function* $\mathsf{negl}(\cdot)$ *it holds that*

$$\Pr\left[\mathsf{Exp}^{\mathsf{unp\text{-}cp}}_{\mathsf{CP},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda) = 1\right] \le \Pr\left[\mathsf{Exp}^{\mathsf{unp\text{-}cp}}_{\mathsf{CP},\mathcal{D},\mathcal{A}^{triv}_{\mathsf{cp}}}(\lambda) = 1\right] + \mathsf{negl}(\lambda),$$

where $\mathsf{Triv}\left(\mathsf{Exp}^{\mathsf{unp\text{-}cp}}_{\mathsf{CP},\mathcal{D}}\right) := \mathsf{Triv}^1\left(\mathsf{Exp}^{\mathsf{unp\text{-}cp}}_{\mathsf{CP},\mathcal{D}}\right) \cup \mathsf{Triv}^2\left(\mathsf{Exp}^{\mathsf{unp\text{-}cp}}_{\mathsf{CP},\mathcal{D}}\right)$ *where for every* $i \in [2]$,

$$\mathsf{Triv}^i\left(\mathsf{Exp}^{\mathsf{unp\text{-}cp}}_{\mathsf{CP},\mathcal{D}}\right) := \{(\mathcal{A}^i_{triv}, \mathcal{B}, \mathcal{C} \mid \mathcal{B}, \mathcal{C} : \text{ QPT adversaries}\},$$

where $\mathcal{A}^1_{triv}$ *(respectively,* $\mathcal{A}^2_{triv}$*) is the adversary that on receiveing a state* q *sends the state to* $\mathcal{B}$ *(respectively,* $\mathcal{C}$*), and sends* $|\bot\rangle\langle\bot|$ *to* $\mathcal{C}$ *(respectively,* $\mathcal{B}$*).*

Similarly, we say that CP *satisfies* $\mathcal{D}$**-unpredictability-style-CP anti-piracy**, *if the same holds as above, but in the absence of any oracle access to the adversaries.*

Moreover, for any noticeable function $\alpha(\cdot)$*, we say that a copy-protection scheme* CP *satisifies* $\alpha(\lambda)$*-oracular-*$\mathcal{D}$*-unpredictability-style-CP anti-piracy for* Circ*, if for any QPT adversary* $\mathcal{A}_{\mathsf{cp}}$*, it holds that*

$$\Pr\left[\mathsf{Exp}^{\mathsf{unp\text{-}cp}}_{\mathsf{CP},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda) = 1\right] \le \alpha(\lambda) + \mathsf{negl}(\lambda),$$

for some negligible function $\mathsf{negl}(\cdot)$.

Remark 3.2. In this work, unless specified otherwise, for most circuit classes with $n(\lambda)$-bit input and $m(\lambda)$-bit output, especially the ones satisfying some form of puncturable security (see Definition 7.1), we will show $1/2^{m(\lambda)}$-oracular-$\mathcal{D}$-unpredictability-style-CP anti-piracy for some appropriate distribution $\mathcal{D}$. This is the best possible result since there always exists a trivial adversary, $(\mathcal{A}^1_{triv}, \mathcal{B}^1_{triv}, \mathcal{C}^1_{triv})$ that wins with probability $1/2^{m(\lambda)}$, where $\mathcal{A}^1_{triv}$ is as defined in Definition 3.1, and $\mathcal{B}^1_{triv}$ just honestly evaluates the circuit on the respective challenge point by running $\mathcal{E}val$ on the state received, and $\mathcal{C}^1_{triv}$ outputs a uniformly random guess.

Next, we define pseudorandomness-style copy-protection, which is a generalization of the indistinguishability-based copy-protection definition defined in [CLLZ21] for pseudorandom functions.

Definition 3.3 (Oracular Pseudorandomness-Style Copy-Protection Anti-piracy). *Let* $\mathsf{Circ} = \{C_k\}_{k\in\mathcal{K}_\lambda}$ *be a family of keyed circuits with n-bit inputs and m-bit outputs, and let* $\mathsf{CP} = (\mathcal{C}opy\mathcal{P}rotect, \mathcal{E}val)$ *be a copy-protection scheme for* Circ *with the same syntax and evaluation correctness as in Definition 3.1.*

For a distribution $\mathcal{D}$ *over* $\mathcal{K}_\lambda \times \{0,1\}^n \times \{0,1\}^n$*, consider the following correlated pseudrandomness-style copy-protection anti-piracy game* $\mathsf{Exp}^{\mathsf{correlated\text{-}pr\text{-}cp}}_{\mathsf{CP},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda)$ *between the challenger and an adversary* $\mathcal{A}_{\mathsf{cp}} = (\mathcal{A}, \mathcal{B}, \mathcal{C})$ *below.*

1. *The challenger chooses* $(k, x^*_1, x^*_2) \leftarrow \mathcal{D}(1^\lambda)$*, generates* $\widetilde{C} \leftarrow \mathcal{C}opy\mathcal{P}rotect(1^\lambda, C_k)$*, and sends* $\widetilde{C}$ *as well as oracle access to* C_k *to* $\mathcal{A}$*.*
2. $\mathcal{A}^{C_k}$ *creates a bipartite state* q *over registers* $\mathsf{R}_\mathcal{B}$ *and* $\mathsf{R}_\mathcal{C}$*. Then,* $\mathcal{A}$ *sends register* $\mathsf{R}_\mathcal{B}$ *to* $\mathcal{B}$ *and register* $\mathsf{R}_\mathcal{C}$ *to* $\mathcal{C}$*.*
3. *The challenger chooses* $\mathsf{coin} \xleftarrow{\$} \{0,1\}$*, and for every* $i \in [2]$ *sets* $y^0_i := C_k(x^*_i)$*, and chooses* $y^1_i \xleftarrow{\$} \{0,1\}^m$*. The challenger sends* $(x^*_1, y^{\mathsf{coin}}_1)$ *and oracle access to* $C \setminus \{x^*_1\}$ *to* $\mathcal{B}$ *and sends* $(x^*_2, y^{\mathsf{coin}}_2)$ *and oracle access to* $C \setminus \{x^*_2\}$ *to* $\mathcal{C}$*.*
4. $\mathcal{B}^{C\setminus\{x^*_1\}}$ *and* $\mathcal{C}^{C\setminus\{x^*_2\}}$ *respectively output* $\mathsf{coin}'_\mathcal{B}$ *and* $\mathsf{coin}'_\mathcal{C}$*. The challenger outputs* 1 *if* $\mathsf{coin}'_\mathcal{B} = \mathsf{coin}'_\mathcal{C} = \mathsf{coin}$ *otherwise outputs* 0*.*

We say that CP *satisfies* **oracular-$\mathcal{D}$-pseudorandomness-style-CP anti-piracy** *for* Circ*, if for any QPT adversary* $\mathcal{A}_{\mathsf{cp}}$*, it holds that*

$$\Pr\left[\mathsf{Exp}^{\mathsf{correlated\text{-}pr\text{-}cp}}_{\mathsf{CP},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda),$$

for some negligible function $\mathsf{negl}(\cdot)$*.*

Next, we introduce a strengthening of pseudorandomness-style copy-protection by adopting the "+"-style formulation introduced by [KY25b] in the context of SDE.

Definition 3.4 (Oracular Pseudorandomness-Style Copy-Protection plus Anti-piracy). *Let* $\mathsf{Circ} = \{C_k\}_{k\in\mathcal{K}_\lambda}$ *be a family of keyed circuits with n-bit inputs and m-bit outputs, and let* $\mathsf{CP} = (\mathcal{C}opy\mathcal{P}rotect, \mathcal{E}val)$ *be a copy-protection scheme for* Circ *with the same syntax and evaluation correctness as in Definition 3.1.*

For a distribution $\mathcal{D}$ *over* $\mathcal{K}_\lambda \times \{0,1\}^n \times \{0,1\}^n$*, consider the following independent pseudrandomness-style copy-protection-plus anti-piracy game* $\mathsf{Exp}^{\mathsf{independent\text{-}pr\text{-}cp+}}_{\mathsf{CP},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda)$ *between the challenger and an adversary* $\mathcal{A}_{\mathsf{cp}} = (\mathcal{A}, \mathcal{B}, \mathcal{C})$ *below.*

1. *The challenger chooses* $(k, x^*_1, x^*_2) \leftarrow \mathcal{D}$*, generates* $\widetilde{C} \leftarrow \mathcal{C}opy\mathcal{P}rotect(1^\lambda, C_k)$*, and sends* $\widetilde{C}$ *as well as oracle access to* C_k *to* $\mathcal{A}$*.*
2. $\mathcal{A}^{C_k}$ *creates a bipartite state* q *over registers* $\mathsf{R}_\mathcal{B}$ *and* $\mathsf{R}_\mathcal{C}$*. Then,* $\mathcal{A}$ *sends register* $\mathsf{R}_\mathcal{B}$ *to* $\mathcal{B}$ *and register* $\mathsf{R}_\mathcal{C}$ *to* $\mathcal{C}$*.*
3. *The challenger chooses* $\mathsf{coin}_\mathcal{B}, \mathsf{coin}_\mathcal{C} \xleftarrow{\$} \{0,1\}$*, and for every* $i \in [2]$ *sets* $y^0_i := C_k(x^*_i)$*, and chooses* $y^1_i \xleftarrow{\$} \{0,1\}^m$*. The challenger sends* $(x^*_1, y^{\mathsf{coin}_\mathcal{B}}_1)$ *and oracle access to* $C \setminus \{x^*_1\}$ *to* $\mathcal{B}$ *and sends* $(x^*_2, y^{\mathsf{coin}_\mathcal{C}}_2)$ *and oracle access to* $C \setminus \{x^*_2\}$ *to* $\mathcal{C}$*.*

4. $\mathcal{B}^{C\setminus\{x_1^*\}}$ *and* $\mathcal{C}^{C\setminus\{x_2^*\}}$ *respectively output* $\mathsf{coin}'_{\mathcal{B}}$ *and* $\mathsf{coin}'_{\mathcal{C}}$. *The challenger outputs* 1 *if* $\mathsf{coin}'_{\mathcal{B}} \oplus \mathsf{coin}'_{\mathcal{C}} = \mathsf{coin}_{\mathcal{B}} \oplus \mathsf{coin}_{\mathcal{C}}$, *otherwise outputs* 0.

We say that CP *satisfies* **oracular-$\mathcal{D}$-pseudorandomness-style-CP+ anti-piracy** *for* Circ, *if for any QPT adversary* $\mathcal{A}_{\mathsf{cp}}$, *it holds that*

$$\Pr\left[\mathsf{Exp}^{\mathsf{independent\text{-}pr\text{-}cp}+}_{\mathsf{CP},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda)=1\right] \leq \frac{1}{2}+\mathsf{negl}(\lambda),$$

for some negligible function $\mathsf{negl}(\cdot)$.

Relationships. We explore the relationships between the different notions of copy-protection below.

Lemma 3.5. *If a copy-protection scheme* CP *satisfies oracular-$\mathcal{D}$-pseudorandomness-style-CP+ anti-piracy (Definition 3.4) for a circuit class* Circ, *then it also satisfies oracular-$\mathcal{D}$-pseudorandomness-style-CP anti-piracy (Definition 3.3), i.e., without the "+" anti-piracy.*

Proof of Lemma 3.5. The proof is analogous to that of the implication from CPA$^+$anti-piracy to $\mathcal{D}_{\mathsf{iden\text{-}bit},\mathsf{ind\text{-}msg}}$-CPA anti-piracy for SDE given in [KY25b, Theorem 6.20].

Let $\mathcal{A}_{\mathsf{cp}} = (\mathcal{A},\mathcal{B},\mathcal{C})$ be any adversary for oracular-$\mathcal{D}$-pseudorandomness-style-CP anti-piracy (Definition 3.4) against CP for the circuit class Circ.

It is enough to prove that

$$\Pr\left[\mathsf{Exp}^{\mathsf{correlated\text{-}pr\text{-}cp}}_{\mathsf{CP},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda)=1\right] \leq \Pr\left[\mathsf{Exp}^{\mathsf{independent\text{-}pr\text{-}cp}+}_{\mathsf{CP},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda)=1\right].$$

In an execution of $\mathsf{Exp}^{\mathsf{independent\text{-}pr\text{-}cp}+}_{\mathsf{CP},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda)$, let $\mathsf{coin}_{\mathcal{B}}$ and $\mathsf{coin}_{\mathcal{C}}$ be the challenge bits chosen by the challenger for $\mathcal{B}$ and $\mathcal{C}$ and $\mathsf{coin}'_{\mathcal{B}}$ and $\mathsf{coin}'_{\mathcal{C}}$ be the outputs of $\mathcal{B}$ and $\mathcal{C}$, respectively. Let $E_{\mathcal{B}}$ be the event that $\mathsf{coin}'_{\mathcal{B}} = \mathsf{coin}_{\mathcal{B}}$, $E_{\mathcal{C}}$ be the event that $\mathsf{coin}'_{\mathcal{C}} = \mathsf{coin}_{\mathcal{C}}$, and $E_{\mathsf{iden\text{-}bit}}$ be the event that $\mathsf{coin}_{\mathcal{B}} = \mathsf{coin}_{\mathcal{C}}$. Clearly, we have

$$\Pr\left[\mathsf{Exp}^{\mathsf{independent\text{-}pr\text{-}cp}+}_{\mathsf{CP},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda)=1\right] = \Pr[E_{\mathcal{B}}\wedge E_{\mathcal{C}}]+\Pr[\neg E_{\mathcal{B}}\wedge\neg E_{\mathcal{C}}]$$

and

$$\Pr\left[\mathsf{Exp}^{\mathsf{correlated\text{-}pr\text{-}cp}}_{\mathsf{CP},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda)=1\right] = \Pr[E_{\mathcal{B}}\wedge E_{\mathcal{C}} \mid E_{\mathsf{iden\text{-}bit}}].$$

Next, note that by definition of the security game $\mathsf{Exp}^{\mathsf{independent\text{-}pr\text{-}cp}+}_{\mathsf{CP},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda)$,

$$\Pr[E_{\mathcal{B}} \mid E_{\mathsf{iden\text{-}bit}}] = \Pr[E_{\mathcal{B}} \mid \neg E_{\mathsf{iden\text{-}bit}}]$$

and

$$\Pr[E_{\mathcal{C}} \mid E_{\mathsf{iden\text{-}bit}}] = \Pr[E_{\mathcal{C}} \mid \neg E_{\mathsf{iden\text{-}bit}}].$$

Also, clearly we have $\Pr[E_{\mathsf{iden\text{-}bit}}] = 1/2$. Thus, by Lemma 2.9, we have

$$\begin{aligned}&\Pr[E_{\mathcal{B}}\wedge E_{\mathcal{C}}]+\Pr[\neg E_{\mathcal{B}}\wedge\neg E_{\mathcal{C}}]\\ =\ &\Pr[E_{\mathcal{B}}\wedge E_{\mathcal{C}} \mid E_{\mathsf{iden\text{-}bit}}]+\Pr[\neg E_{\mathcal{B}}\wedge\neg E_{\mathcal{C}} \mid \neg E_{\mathsf{iden\text{-}bit}}] \geq \Pr[E_{\mathcal{B}}\wedge E_{\mathcal{C}} \mid E_{\mathsf{iden\text{-}bit}}].\end{aligned}$$

Thus, we have

$$\Pr\left[\mathsf{Exp}^{\mathsf{correlated\text{-}pr\text{-}cp}}_{\mathsf{CP},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda)=1\right] \leq \Pr\left[\mathsf{Exp}^{\mathsf{independent\text{-}pr\text{-}cp}+}_{\mathsf{CP},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda)=1\right],$$

which completes the proof of Lemma 3.5.

□

Remark 3.6. In Definition 3.3, we considered a correlated version where the challenge points for $\mathcal{B}$ and $\mathcal{C}$ could be different, but the challenge coin is the same for both. We can even consider an independent version where the coins for $\mathcal{B}$ and $\mathcal{C}$ are also sampled independently, but note that this version immediately follows from the plus security. Since our constructions of copy-protection with pseudorandomness-style copy-protection anti-piracy directly satisfy the plus security, we also obtain the independent version of Definition 3.3 for our constructions.

It is easy to see that pseudorandomness-style anti-piracy security implies unpredictability-style anti-piracy security if $m = \omega(\log \lambda)$.

Lemma 3.7 (Pseudorandomness Implies Unpredictability $\omega(\log\lambda)$-Bit Circuits). *Let* Circ *be a circuit class with output length $m = \omega(\log\lambda)$, then for any copy-protection scheme* CP *for* Circ*, oracular-$\mathcal{D}$-pseudorandomness-style-CP anti-piracy implies oracular-$\mathcal{D}$-unpredictability-style-CP anti-piracy.*

Proof of Lemma 3.7. Let $\mathcal{A}_{\mathsf{cp}} = (\mathcal{A}, \mathcal{B}, \mathcal{C})$ be any adversary for oracular-$\mathcal{D}$-unpredictability-style-CP anti-piracy (Definition 3.3) against CP for the circuit class Circ. We construct a reduction adversary $\mathcal{R} := (\mathcal{A}, \mathcal{R}_{\mathcal{B}}, \mathcal{R}_{\mathcal{C}})$ in the correlated psedurandomness-style anti-piracy game, where $\mathcal{R}_{\mathcal{B}}$ (respectively, $\mathcal{R}_{\mathcal{C}}$) on receiveing a state σ_B (respectively, σ_C) from $\mathcal{R}_{\mathcal{A}}$, and challenge (x_1, y_1) (respectively, (x_2, y_2)) and oracle access to $C_k \setminus \{x_1\}$ (respectively, $C_k \setminus \{x_2\}$) runs $y_1' \leftarrow \mathcal{B}(\sigma_B, x_1)$ (respectively, $y_2' \leftarrow \mathcal{C}(\sigma_C, x_2)$), and outputs 0 if $y_1' = y_1$ (respectively, $y_2' = y_2$) and 1 otherwise. Let coin be the challenge bit in the correlated pseudorandomness-style anti-piracy game, $\mathsf{Exp}^{\mathsf{correlated\text{-}pr\text{-}cp}}_{\mathsf{CP},\mathcal{D},\mathcal{R}}(\lambda)$. Note that by description of $(\mathcal{R}_{\mathcal{B}}, \mathcal{R}_{\mathcal{C}})$, in the $\mathsf{coin} = 1$ event of $\mathsf{Exp}^{\mathsf{correlated\text{-}pr\text{-}cp}}_{\mathsf{CP},\mathcal{D},\mathcal{R}}(\lambda)$, for each $i \in [2]$, the probability that $y_i' = y_i$ is $1/2^m$. Hence,

$$\Pr[1 \leftarrow \mathcal{R}_{\mathcal{B}} \wedge 1 \leftarrow \mathcal{R}_{\mathcal{C}} \mid \mathsf{coin} = 1] = \Pr\left[y_i' \neq y_i \forall i \in [2] \mid \mathsf{coin} = 1\right] \geq 1 - \frac{2}{2^m} = 1 - \frac{1}{2^{m-1}}.$$

Next, clearly, the view of the adversaries $\mathcal{B}$ and $\mathcal{C}$ in the simulated game is the same as in the unpredictability-style anti-piracy game $\mathsf{Exp}^{\mathsf{unp\text{-}cp}}_{\mathsf{CP},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda)$, and hence, conditioned on $\mathsf{coin} = 0$, i.e., $y_i = C_k(x_i)$ for both $i \in [2]$, we get that the event $y_i' = y_i$ for every $i \in [2]$ exactly corresponds to the winning event in the unpredictability-style anti-piracy game $\mathsf{Exp}^{\mathsf{unp\text{-}cp}}_{\mathsf{CP},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda)$. Hence,

$$\Pr[0 \leftarrow \mathcal{R}_{\mathcal{B}} \wedge 0 \leftarrow \mathcal{R}_{\mathcal{C}} \mid \mathsf{coin} = 0] = \Pr\left[y_i' = y_i \forall i \in [2] \mid \mathsf{coin} = 0\right] = \Pr\left[\mathsf{Exp}^{\mathsf{unp\text{-}cp}}_{\mathsf{CP},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda) = 1\right].$$

Combining the last two equations, we conclude that

$$\Pr\left[\mathsf{Exp}^{\mathsf{correlated\text{-}pr\text{-}cp}}_{\mathsf{CP},\mathcal{D},\mathcal{R}}(\lambda) = 1\right] \geq \frac{\Pr\left[\mathsf{Exp}^{\mathsf{unp\text{-}cp}}_{\mathsf{CP},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda) = 1\right] + 1 - \frac{1}{2^{m-1}}}{2}. \tag{1}$$

Since we assume that CP satisfies oracular-$\mathcal{D}$-pseudorandomness-style-CP anti-piracy, there exists a negligible function $\mathsf{negl}(\cdot)$ such that

$$\Pr\left[\mathsf{Exp}^{\mathsf{correlated\text{-}pr\text{-}cp}}_{\mathsf{CP},\mathcal{D},\mathcal{R}}(\lambda) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

Combining last equation with (1), we conclude that,

$$1/2 + \mathsf{negl}(\lambda) \geq \Pr\left[\mathsf{Exp}^{\mathsf{correlated\text{-}pr\text{-}cp}}_{\mathsf{CP},\mathcal{D},\mathcal{R}}(\lambda) = 1\right] \geq \frac{\Pr\left[\mathsf{Exp}^{\mathsf{unp\text{-}cp}}_{\mathsf{CP},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda) = 1\right] + 1 - \frac{1}{2^{m-1}}}{2}.$$

Hence, we conclude that,

$$\Pr\left[\mathsf{Exp}^{\mathsf{unp\text{-}cp}}_{\mathsf{CP},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda) = 1\right] \leq 1/2^{m-1} + 2\mathsf{negl}(\lambda),$$

which is negligible in λ since $m(\lambda) \in \omega(\log(\lambda))$. □

Remark 3.8. If the output length m of the circuit class Circ satisfies $m \leq O(\log(\lambda))$, then for any copy-protection scheme CP for Circ, oracular-$\mathcal{D}$-pseudorandomness-style-CP anti-piracy implies $\alpha(\lambda)$-oracular-$\mathcal{D}$-unpredictability-style-CP anti-piracy for some non-negligible but non-trivial bound $\alpha(\lambda)$.

3.2 Definitions of Unclonable Puncturable Obfuscation

Next, we recall the definition of the main tool used in this work to achieve our main results.

Definition 3.9 (Unclonable Puncturable Obfuscation (UPO) [AB24]). *A UPO scheme* UPO *for the circuit class* $\mathsf{Circ} = \{C : \{0,1\}^{\ell_{\mathsf{cin}}} \to \{0,1\}^{\ell_{\mathsf{cout}}}\}$ *is a pair of two algorithms* $(\mathcal{O}bf, \mathcal{E}val)$.

$\mathcal{O}bf(1^\lambda, C) \to \widetilde{C}$**:** *The obfuscation algorithm is a QPT algorithm that takes a security parameter* 1^λ *and a circuit* $C \in \mathsf{Circ}$*, and outputs a quantum state* $\widetilde{C}$.

$\mathcal{E}val(\widetilde{C}, x) \to y$**:** *The evaluation algorithm is a QPT algorithm that takes a quantum state* $\widetilde{C}$ *and an input* $x \in \{0,1\}^{\ell_{\mathsf{cin}}}$*, and outputs a value* y.

Evaluation correctness: *For every* $C \in \mathsf{Circ}$ *and* $x \in \{0,1\}^{\ell_{\mathsf{cin}}}$*, we have*

$$\Pr\left[\mathcal{E}val(\widetilde{C}, x) = C(x) \;\middle|\; \widetilde{C} \leftarrow \mathcal{O}bf(1^\lambda, C)\right] = 1 - \mathsf{negl}(\lambda).$$

We say that it satisfies perfect correctness if the above probability is 1.

Evaluation correctness implies the reusability of the quantum obfuscated state ρ_C thanks to the gentle measurement lemma.

To define the security notions of UPO, we first introduce notation for punctured circuits.

Definition 3.10 (Punctured Circuit). *Let* $C : \{0,1\}^{\ell_{\mathsf{cin}}} \to \{0,1\}^{\ell_{\mathsf{cout}}}$ *be a circuit. Let* $x_{\mathcal{B}}, x_C \in \{0,1\}^{\ell_{\mathsf{cin}}}$ *and* $y_{\mathcal{B}}, y_C \in \{0,1\}^{\ell_{\mathsf{out}}}$*. We define the punctured circuit* $C^*[x_{\mathcal{B}}, x_C, y_{\mathcal{B}}, y_C]$ *as follows:*

- *If* $x_{\mathcal{B}} \neq x_C$*,*

$$C^*[x_{\mathcal{B}}, x_C, y_{\mathcal{B}}, y_C](x) = \begin{cases} C(x), & x \in \{0,1\}^{\ell_{\mathsf{cin}}} \setminus \{x_{\mathcal{B}}, x_C\} \\ y_{\mathcal{B}}, & x = x_{\mathcal{B}} \\ y_C, & x = x_C \end{cases}.$$

- *If* $x_{\mathcal{B}} = x_C$*,*

$$C^*[x_{\mathcal{B}}, x_C, y_{\mathcal{B}}, y_C](x) = \begin{cases} C(x), & x \in \{0,1\}^{\ell_{\mathsf{cin}}} \setminus \{x_{\mathcal{B}}\} \\ y_{\mathcal{B}}, & x = x_{\mathcal{B}} \end{cases}.$$

Note that there exists a canonical algorithm $\mathsf{Program}_{canonical}$ *to perform the above-described puncturing for any circuit class* Circ*, i.e., given a circuit key/index from* Circ*,* $\mathsf{Program}_{canonical}$ *outputs the circuit that has* C *hardcoded using which it outputs according to* $C^*[x_{\mathcal{B}}, x_C, y_{\mathcal{B}}, y_C]$.

First, we recall two security notions introduced in [AB24]: UPO security and generalized UPO security.[13]

Definition 3.11 ($\mathcal{D}$-Generalized UPO Security [AB24]). *Let* $\mathsf{UPO} = (\mathcal{O}bf, \mathcal{E}val)$ *be a UPO scheme for the circuit class* $\mathsf{Circ} = \{C : \{0,1\}^{\ell_{\mathsf{cin}}} \to \{0,1\}^{\ell_{\mathsf{cout}}}\}$*. Let* $\mathcal{D}$ *be a distribution over* $\{0,1\}^{\ell_{\mathsf{cin}}} \times \{0,1\}^{\ell_{\mathsf{cin}}}$*. We consider the* $\mathcal{D}$*-generalized UPO anti-piracy game* $\mathsf{Exp}^{\mathsf{gen\text{-}upo}}_{\mathsf{UPO},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda)$ *between the challenger and an adversary* $\mathcal{A}_{\mathsf{cp}} = (\mathcal{A}, \mathcal{B}, \mathcal{C})$ *below.*

1. *On input* 1^λ*,* $\mathcal{A}$ *sends* $C \in \mathsf{Circ}$ *together with two circuits* $\mu_{\mathcal{B}} : \{0,1\}^{\ell_{\mathsf{cin}}} \to \{0,1\}^{\ell_{\mathsf{cout}}}$ *and* $\mu_C : \{0,1\}^{\ell_{\mathsf{cin}}} \to \{0,1\}^{\ell_{\mathsf{cout}}}$ *to the challenger.*
2. *The challenger generates* $\mathsf{coin} \leftarrow \{0,1\}$*,* $(x_{\mathcal{B}}, x_C) \leftarrow \mathcal{D}$*, and generates* $\widetilde{C}$ *as follows:*
   - *If* $\mathsf{coin} = 0$*,* $\widetilde{C} \leftarrow \mathcal{O}bf(1^\lambda, C)$.

[13] In [AB24], UPO security is introduced first, followed by its generalization, generalized UPO security. In contrast, for convenience of presentation, we adopt the reverse order: we first define generalized UPO security and then present UPO security as a special case.

- *If* $\mathsf{coin} = 1$, $\widetilde{C} \leftarrow \mathcal{O}bf(1^\lambda, C^*[x_\mathcal{B}, x_\mathcal{C}, \mu_\mathcal{B}(x_\mathcal{B}), \mu_\mathcal{C}(x_\mathcal{C})])$.

The challenger sends $\widetilde{C}$ *to* $\mathcal{A}$.

3. $\mathcal{A}$ *creates a bipartite state* q *over registers* $\mathsf{R}_\mathcal{B}$ *and* $\mathsf{R}_\mathcal{C}$. *Then,* $\mathcal{A}$ *sends register* $\mathsf{R}_\mathcal{B}$ *to* $\mathcal{B}$ *and register* $\mathsf{R}_\mathcal{C}$ *to* $\mathcal{C}$.
4. *The challenger sends* $x_\mathcal{B}$ *and* $x_\mathcal{C}$ *to* $\mathcal{B}$ *and* $\mathcal{C}$, *respectively.*
5. $\mathcal{B}$ *and* $\mathcal{C}$ *respectively output* $\mathsf{coin}'_\mathcal{B}$ *and* $\mathsf{coin}'_\mathcal{C}$. *The challenger outputs* 1 *if* $\mathsf{coin}'_\mathcal{B} = \mathsf{coin}'_\mathcal{C} = \mathsf{coin}$ *otherwise outputs* 0.

We say that UPO *is original* $\mathcal{D}$*-generalized UPO secure if for any QPT adversary* $\mathcal{A}_\mathsf{cp}$, *it holds that*

$$\Pr\left[\mathsf{Exp}^{\mathsf{gen\text{-}upo}}_{\mathsf{UPO},\mathcal{D},\mathcal{A}_\mathsf{cp}}(\lambda) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

Definition 3.12 ($\mathcal{D}$-UPO Security [AB24]). *$\mathcal{D}$-UPO Security is defined similarly to $\mathcal{D}$-generalized UPO Security Definition 3.11 except that $\mu_\mathcal{B}$ and $\mu_\mathcal{C}$ are limited to be circuits that output $\bot$ on all inputs.*

Example 3.13 (Examples of distribution $\mathcal{D}$). In this work, we mainly focus on the following two types of distributions: product distributions and diagonal distributions.

- We say that a distribution $\mathcal{D}$ over $\{0,1\}^{\ell_\mathsf{cin}} \times \{0,1\}^{\ell_\mathsf{cin}}$ is a product distribution if it can be written as $\mathcal{D} = \mathcal{D}_\mathcal{B} \times \mathcal{D}_\mathcal{C}$ where $\mathcal{D}_\mathcal{B}$ and $\mathcal{D}_\mathcal{C}$ are distributions over $\{0,1\}^{\ell_\mathsf{cin}}$. As a special case of product distributions, we write $\mathcal{U}$ to mean the uniform distribution over $\{0,1\}^{\ell_\mathsf{cin}} \times \{0,1\}^{\ell_\mathsf{cin}}$.
- We say that a distribution $\mathcal{D}$ over $\{0,1\}^{\ell_\mathsf{cin}} \times \{0,1\}^{\ell_\mathsf{cin}}$ is a diagonal distribution if its support belongs to $\{(x,x) : x \in \{0,1\}^{\ell_\mathsf{cin}}\}$. As a special case of diagonal distributions, we write $\mathcal{ID}_\mathcal{U}$ to mean the uniform diagonal distribution that generates (x,x) for uniformly random $x \leftarrow \{0,1\}^{\ell_\mathsf{cin}}$.

In this paper, we introduce a new security notion of UPO which we call generalized UPO$^+$security. . This extends the generalized UPO security Definition 3.11 by adopting the "+"-style formulation introduced by [KY25b] in the context of SDE.

Definition 3.14 ($\mathcal{D}$-Generalized UPO$^+$Security). *Let* $\mathsf{UPO} = (\mathcal{O}bf, \mathcal{E}val)$ *be a UPO scheme for the circuit class* $\mathsf{Circ} = \{C : \{0,1\}^{\ell_\mathsf{cin}} \to \{0,1\}^{\ell_\mathsf{cout}}\}$. *Let* $\mathcal{D}$ *be a distribution over* $\{0,1\}^{\ell_\mathsf{cin}} \times \{0,1\}^{\ell_\mathsf{cin}}$. *We consider the* $\mathcal{D}$*-UPO*$^+$ *anti-piracy game* $\mathsf{Exp}^{\mathsf{gen\text{-}upoplus}}_{\mathsf{UPO},\mathcal{D},\mathcal{A}_\mathsf{cp}}(\lambda)$ *between the challenger and an adversary* $\mathcal{A}_\mathsf{cp} = (\mathcal{A}, \mathcal{B}, \mathcal{C})$ *below.*

1. *On input* 1^λ, $\mathcal{A}$ *sends* $C \in \mathsf{Circ}$ *together with two circuits* $\mu_\mathcal{B} : \{0,1\}^{\ell_\mathsf{cin}} \to \{0,1\}^{\ell_\mathsf{cout}}$ *and* $\mu_\mathcal{C} : \{0,1\}^{\ell_\mathsf{cin}} \to \{0,1\}^{\ell_\mathsf{cout}}$ *to the challenger.*
2. *The challenger generates* $\mathsf{coin}_\mathcal{B} \leftarrow \{0,1\}$, $\mathsf{coin}_\mathcal{C} \leftarrow \{0,1\}$, *and* $(x_\mathcal{B}, x_\mathcal{C}) \leftarrow \mathcal{D}$, *and sets as follows:*

$$y_{\mathcal{B},0} := C(x_\mathcal{B}), \quad y_{\mathcal{B},1} := \mu_\mathcal{B}(x_\mathcal{B}), \quad y_{\mathcal{C},0} := C(x_\mathcal{C}), \quad y_{\mathcal{C},1} := \mu_\mathcal{C}(x_\mathcal{C}).$$

The challenger generates $\rho_C \leftarrow \mathcal{O}bf(1^\lambda, C^*[x_\mathcal{B}, x_\mathcal{C}, y_{\mathcal{B},\mathsf{coin}_\mathcal{B}}, y_{\mathcal{C},\mathsf{coin}_\mathcal{C}}])$ *and sends* ρ_C *to* $\mathcal{A}$.

3. $\mathcal{A}$ *creates a bipartite state* q *over registers* $\mathsf{R}_\mathcal{B}$ *and* $\mathsf{R}_\mathcal{C}$. *Then,* $\mathcal{A}$ *sends register* $\mathsf{R}_\mathcal{B}$ *to* $\mathcal{B}$ *and register* $\mathsf{R}_\mathcal{C}$ *to* $\mathcal{C}$.
4. *The challenger sends* $x_\mathcal{B}$ *and* $x_\mathcal{C}$ *to* $\mathcal{B}$ *and* $\mathcal{C}$, *respectively.*
5. $\mathcal{B}$ *and* $\mathcal{C}$ *respectively output* $\mathsf{coin}'_\mathcal{B}$ *and* $\mathsf{coin}'_\mathcal{C}$. *The challenger outputs* 1 *if* $\mathsf{coin}'_\mathcal{B} \oplus \mathsf{coin}'_\mathcal{C} = \mathsf{coin}_\mathcal{B} \oplus \mathsf{coin}_\mathcal{C}$ *otherwise outputs* 0.

We say that UPO *is* $\mathcal{D}$*-UPO*$^+$ *secure if for any QPT adversary* $\mathcal{A}_\mathsf{cp}$, *it holds that*

$$\Pr\left[\mathsf{Exp}^{\mathsf{gen\text{-}upoplus}}_{\mathsf{UPO},\mathcal{D},\mathcal{A}_\mathsf{cp}}(\lambda) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

In some applications, we assume that UPO also satisfies security as indistinguishability obfuscation (iO). For clarity, we define what it means for a UPO to satisfy iO security. Note that iO security can be generically added by first applying iO, followed by UPO.

Definition 3.15 (iO Security). *We say that a UPO scheme* $\mathsf{UPO} = (\mathcal{O}bf, \mathcal{E}val)$ *satisfies iO security if for any PPT* $\mathsf{Sampler}$ *and QPT adversary* $\mathcal{A}$*, the following holds:*

If $\Pr\left[\forall x\, C_0(x) = C_1(x) \wedge |C_0| = |C_1| \mid (C_0, C_1, \mathsf{aux}) \leftarrow \mathsf{Sampler}(1^\lambda)\right] = 1 - \mathsf{negl}(\lambda)$*, then we have*

$$\mathsf{Adv}^{\mathsf{io}}_{i\mathcal{O},\mathcal{D}}(\lambda) := \Big|\Pr\left[\mathcal{A}(\mathcal{O}bf(1^\lambda, C_0), \mathsf{aux}) = 1 \mid (C_0, C_1, \mathsf{aux}) \leftarrow \mathsf{Sampler}(1^\lambda)\right] - \Pr\left[\mathcal{A}(\mathcal{O}bf(1^\lambda, C_1), \mathsf{aux}) = 1 \mid (C_0, C_1, \mathsf{aux}) \leftarrow \mathsf{Sampler}(1^\lambda)\right]\Big| \leq \mathsf{negl}(\lambda).$$

The following lemma is easy to prove along the lines of the composition theorem proven in [AB24, Theorem 12].

Lemma 3.16. *Let* $\mathsf{Circ} = \{C : \{0,1\}^{\ell_{\mathsf{cin}}} \to \{0,1\}^{\ell_{\mathsf{cout}}}\}$ *be a circuit class and* $\mathcal{D}$ *be a distribution over* $\{0,1\}^{\ell_{\mathsf{cin}}} \times \{0,1\}^{\ell_{\mathsf{cin}}}$*. For any* $X \in \{\mathcal{D}$*-UPO,*$\mathcal{D}$*-generalized UPO,* $\mathcal{D}$*-generalized UPO*$^+\}$*, if there exist a secure iO scheme and a UPO scheme for the circuit class* Circ *that satisfies* X *security, there exists a UPO scheme for the circuit class* Circ *that satisfies both iO security and* X *security.*

The above lemma allows us to assume, without loss of generality, that UPO (under any flavor of security) satisfies iO security , provided that iO exists.

We show that for any product distribution $\mathcal{D} = \mathcal{D}_{\mathcal{B}} \times \mathcal{D}_{\mathcal{C}}$, $\mathcal{D}$-generalized UPO$^+$security implies $\mathcal{D}$-generalized UPO security, assuming that the UPO satisfies iO security and there exist keyed injective $\mathcal{D}_{\mathcal{B}}$- and $\mathcal{D}_{\mathcal{C}}$-one-way functions.

Theorem 3.17. *Let* $\mathsf{UPO} = (\mathcal{O}bf, \mathcal{E}val)$ *be a UPO scheme for the circuit class* $\mathsf{Circ} = \{C : \{0,1\}^{\ell_{\mathsf{cin}}} \to \{0,1\}^{\ell_{\mathsf{cout}}}\}$*. Let* $\mathcal{D} = \mathcal{D}_{\mathcal{B}} \times \mathcal{D}_{\mathcal{C}}$ *be a product of two distributions* $\mathcal{D}_{\mathcal{B}}$ *and* $\mathcal{D}_{\mathcal{C}}$ *over* $\{0,1\}^{\ell_{\mathsf{cin}}}$*. If* UPO *satisfies* $\mathcal{D}$*-generalized UPO*$^+$ *security and* $i\mathcal{O}$*-security, and there exist keyed injective* $\mathcal{D}_{\mathcal{B}}$*- and* $\mathcal{D}_{\mathcal{C}}$ *one-way functions, then* UPO *also satisfies* $\mathcal{D}$*-generalized UPO security.*

Proof of Theorem 3.17. We note that the proof is similar to the implication from CPA$^+$anti-piracy to $\mathcal{D}_{\mathsf{iden\text{-}bit,ind\text{-}msg}}$-CPA anti-piracy for SDE given in [KY25b, Theorem 6.20].

Let $\mathcal{A}_{\mathsf{cp}} = (\mathcal{A}, \mathcal{B}, \mathcal{C})$ be any adversary for $\mathcal{D}$-generalized UPO security of UPO. Consider the following hybrid experiments.

Hyb_0**:** This is $\mathsf{Exp}^{\mathsf{gen\text{-}upo}}_{\mathsf{UPO},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda)$.

We have

$$\Pr[\mathsf{Hyb}_0 = 1] = \Pr\left[\mathsf{Exp}^{\mathsf{gen\text{-}upo}}_{\mathsf{UPO},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda) = 1\right].$$

Hyb_1**:** Same as Hyb_0 except that if $\mathsf{coin} = 0$, $\widetilde{C}$ is generated as $\widetilde{C} \leftarrow \mathcal{O}bf(1^\lambda, C^*[x_{\mathcal{B}}, x_{\mathcal{C}}, C(x_{\mathcal{B}}), C(x_{\mathcal{C}})])$.

From the iO security of UPO, we have

$$|\Pr[\mathsf{Hyb}_0 = 1] - \Pr[\mathsf{Hyb}_1 = 1]| = \mathsf{negl}(\lambda).$$

We below prove that

$$\Pr[\mathsf{Hyb}_1 = 1] \leq \Pr\left[\mathsf{Exp}^{\mathsf{gen\text{-}upoplus}}_{\mathsf{UPO},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda) = 1\right] + \mathsf{negl}(\lambda).$$

In an execution of $\mathsf{Exp}^{\mathsf{gen\text{-}upoplus}}_{\mathsf{UPO},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda)$, let $\mathsf{coin}_{\mathcal{B}}$ and $\mathsf{coin}_{\mathcal{C}}$ be the challenge bits chosen by the challenger for $\mathcal{B}$ and $\mathcal{C}$ and $\mathsf{coin}'_{\mathcal{B}}$ and $\mathsf{coin}'_{\mathcal{C}}$ be the outputs of $\mathcal{B}$ and $\mathcal{C}$, respectively. Let $E_{\mathcal{B}}$ be the event that $\mathsf{coin}'_{\mathcal{B}} = \mathsf{coin}_{\mathcal{B}}$, $E_{\mathcal{C}}$ be the event that $\mathsf{coin}'_{\mathcal{C}} = \mathsf{coin}_{\mathcal{C}}$, and $E_{\mathsf{iden\text{-}bit}}$ be the event that $\mathsf{coin}_{\mathcal{B}} = \mathsf{coin}_{\mathcal{C}}$. Clearly, we have

$$\Pr\left[\mathsf{Exp}^{\mathsf{gen\text{-}upoplus}}_{\mathsf{UPO},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda) = 1\right] = \Pr[E_{\mathcal{B}} \wedge E_{\mathcal{C}}] + \Pr[\neg E_{\mathcal{B}} \wedge \neg E_{\mathcal{C}}]$$

and

$$\Pr[\mathsf{Hyb}_1 = 1] = \Pr[E_{\mathcal{B}} \wedge E_{\mathcal{C}} \mid E_{\mathsf{iden\text{-}bit}}].$$

Here, we prove the following claim:

Claim 3.18. It holds that

$$\Pr[E_{\mathcal{B}} \mid E_{\mathsf{iden\text{-}bit}}] \approx_{\mathsf{negl}(\lambda)} \Pr[E_{\mathcal{B}} \mid \neg E_{\mathsf{iden\text{-}bit}}].$$

and

$$\Pr[E_{\mathcal{C}} \mid E_{\mathsf{iden\text{-}bit}}] \approx_{\mathsf{negl}(\lambda)} \Pr[E_{\mathcal{C}} \mid \neg E_{\mathsf{iden\text{-}bit}}].$$

We first finish the proof of Theorem 3.17 assuming Claim 3.18. It is clear that we have $\Pr[E_{\mathsf{iden\text{-}bit}}] = 1/2$. Thus, by Lemma 2.9, we have

$$\begin{aligned}&\Pr[E_{\mathcal{B}} \wedge E_{\mathcal{C}}] + \Pr[\neg E_{\mathcal{B}} \wedge \neg E_{\mathcal{C}}]\\ &\approx_{\mathsf{negl}(\lambda)} \Pr[E_{\mathcal{B}} \wedge E_{\mathcal{C}} \mid E_{\mathsf{iden\text{-}bit}}] + \Pr[\neg E_{\mathcal{B}} \wedge \neg E_{\mathcal{C}} \mid \neg E_{\mathsf{iden\text{-}bit}}] \geq \Pr[E_{\mathcal{B}} \wedge E_{\mathcal{C}} \mid E_{\mathsf{iden\text{-}bit}}].\end{aligned}$$

This implies

$$\Pr\left[\mathsf{Exp}^{\mathsf{gen\text{-}upoplus}}_{\mathsf{UPO},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda) = 1\right] + \mathsf{negl}(\lambda) \geq \Pr[\mathsf{Hyb}_1 = 1].$$

Thus, we have

$$\Pr\left[\mathsf{Exp}^{\mathsf{gen\text{-}upo}}_{\mathsf{UPO},\mathcal{A}_{\mathsf{cp}}}(\lambda) = 1\right] \leq \Pr\left[\mathsf{Exp}^{\mathsf{gen\text{-}upoplus}}_{\mathsf{UPO},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda) = 1\right] + \mathsf{negl}(\lambda).$$

Since we assume that UPO satisfies $\mathcal{D}$-generalized UPO$^+$security, $\Pr\left[\mathsf{Exp}^{\mathsf{gen\text{-}upoplus}}_{\mathsf{UPO},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda) = 1\right] = \mathsf{negl}(\lambda)$. Thus, $\Pr\left[\mathsf{Exp}^{\mathsf{gen\text{-}upo}}_{\mathsf{UPO},\mathcal{A}_{\mathsf{cp}}}(\lambda) = 1\right] = \mathsf{negl}(\lambda)$. This completes the proof of Theorem 3.17.

We are left to prove Claim 3.18.

Proof of Claim 3.18. We only show the proof for the first equality since the other can be proven similarly.

Let Gen be a generator for keyed injective $\mathcal{D}_{\mathcal{C}}$-one-way function. We consider the following sequence of hybrids:

Hyb_0**:** This hybrid works similarly to $\mathsf{Exp}^{\mathsf{gen\text{-}upoplus}}_{\mathsf{UPO},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda)$ conditioned on $E_{\mathsf{iden\text{-}bit}}$, i.e., $\mathsf{coin}_{\mathcal{B}} = \mathsf{coin}_{\mathcal{C}}$, and outputs 1 if $\mathsf{coin}_{\mathcal{B}} = \mathsf{coin}'_{\mathcal{B}}$.

By definition, we have

$$\Pr[\mathsf{Hyb}_0 = 1] = \Pr[E_{\mathcal{B}} \mid E_{\mathsf{iden\text{-}bit}}].$$

Hyb_1**:** This is identical to Hyb_0 except that the challenger samples $\mathsf{F} \leftarrow \mathsf{Gen}(1^\lambda)$, computes $z_{\mathcal{C}} := \mathsf{F}(x_{\mathcal{C}})$, and generates $\widetilde{\mathcal{C}}$ as follows:

$$\widetilde{\mathcal{C}} \leftarrow \begin{cases} \mathcal{O}bf(1^\lambda, C'_0[x_{\mathcal{B}}, y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}}]) & \mathsf{coin}_{\mathcal{C}} = 0\\ \mathcal{O}bf(1^\lambda, C'_1[x_{\mathcal{B}}, z_{\mathcal{C}}, y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}}, y_{\mathcal{C},1}]) & \mathsf{coin}_{\mathcal{C}} = 1\end{cases}$$

where

$$C'_0[x_{\mathcal{B}}, y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}}](x) = \begin{cases} C(x), & x \neq x_{\mathcal{B}}\\ y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}}, & x = x_{\mathcal{B}}\end{cases}$$

and[14]

$$C'_1[x_{\mathcal{B}}, z_{\mathcal{C}}, y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}}, y_{\mathcal{C},1}](x) = \begin{cases} C(x), & x \neq x_{\mathcal{B}} \wedge \mathsf{F}(x) \neq z_{\mathcal{C}}\\ y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}}, & x = x_{\mathcal{B}}\\ y_{\mathcal{C},1}, & \mathsf{F}(x) = z_{\mathcal{C}}\end{cases}.$$

Since $y_{\mathcal{C},0} = C(x_{\mathcal{C}})$, when $\mathsf{coin}_{\mathcal{C}} = 0$, $C'_0[x_{\mathcal{B}}, y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}}]$ is functionally equivalent to $C^*[x_{\mathcal{B}}, x_{\mathcal{C}}, y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}}, y_{\mathcal{C},0}]$. Moreover, by the injectivity of F, $C'_1[x_{\mathcal{B}}, z_{\mathcal{C}}, y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}}, y_{\mathcal{C},1}]$ is functionally equivalent to $C^*[x_{\mathcal{B}}, x_{\mathcal{C}}, y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}}, y_{\mathcal{C},1}]$ with overwhelming probability. Thus, by the iO security of UPO, we have

$$|\Pr[\mathsf{Hyb}_0 = 1] - \Pr[\mathsf{Hyb}_1 = 1]| \leq \mathsf{negl}(\lambda).$$

[14]For completeness, we define $C'_1[x_{\mathcal{B}}, z_{\mathcal{C}}, y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}}, y_{\mathcal{C},1}](x)$ to be $y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}}$ when x satisfies both $x = x_{\mathcal{B}}$ and $\mathsf{F}(x) = z_{\mathcal{C}}$, though this case can be defined arbitrarily since we have $\mathsf{F}(x_{\mathcal{B}}) = z_{\mathcal{C}}$ only with a negligible probability.

Hyb_2**:** This is identical to Hyb_1 except that $\widetilde{C}$ is generated as

$$\widetilde{C} \leftarrow \mathcal{O}bf(1^\lambda, C'_0[x_{\mathcal{B}}, y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}}])$$

regardless of the value of coin_C.

When $\mathsf{coin}_C = 0$, nothing is changed. When $\mathsf{coin}_C = 1$, by the injectivity of F, $C'_0[x_{\mathcal{B}}, y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}}]$ and $C'_1[x_{\mathcal{B}}, z_C, y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}}, y_{C,1}]$ differ only on a single point x_C with overwhelming probability. Moreover, given descriptions of these two circuits, it is computationally infeasible to find x_C by the $\mathcal{D}_C$-one-wayness of F. Since the iO security implies differing-input obfuscation security in the single-differing input setting [BCP14], we have

$$|\Pr[\mathsf{Hyb}_1 = 1] - \Pr[\mathsf{Hyb}_2 = 1]| \leq \mathsf{negl}(\lambda).$$

Hyb_3**:** This is identical to Hyb_2 except that we choose $\mathsf{coin}_{\mathcal{B}}$ and coin_C conditioned on $\mathsf{coin}_{\mathcal{B}} \neq \mathsf{coin}_C$ instead of on $\mathsf{coin}_{\mathcal{B}} = \mathsf{coin}_C$.

Note that the marginal distribution of $\mathsf{coin}_{\mathcal{B}}$ is uniform in both Hyb_2 and Hyb_3 and no information of coin_C is given to $\mathcal{B}$. Thus, the above modification does not change the view of $\mathcal{B}$, and thus we have

$$\Pr[\mathsf{Hyb}_2 = 1] = \Pr[\mathsf{Hyb}_3 = 1].$$

Hyb_4**:** This hybrid works similarly to $\mathsf{Exp}^{\mathsf{gen\text{-}upoplus}}_{\mathsf{UPO},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda)$ conditioned on $\neg E_{\mathsf{iden\text{-}bit}}$, i.e., $\mathsf{coin}_{\mathcal{B}} \neq \mathsf{coin}_C$, and outputs 1 if $\mathsf{coin}_{\mathcal{B}} = \mathsf{coin}'_{\mathcal{B}}$.

By repeating similar arguments to those from Hyb_0 to Hyb_2 in the reversed order, we have

$$|\Pr[\mathsf{Hyb}_3 = 1] - \Pr[\mathsf{Hyb}_4 = 1]| \leq \mathsf{negl}(\lambda).$$

Also, we clearly have

$$\Pr[\mathsf{Hyb}_4 = 1] = \Pr[E_{\mathcal{B}} \mid \neg E_{\mathsf{iden\text{-}bit}}].$$

Combining the above, we conclude

$$\Pr[E_{\mathcal{B}} \mid E_{\mathsf{iden\text{-}bit}}] \approx_{\mathsf{negl}(\lambda)} \Pr[E_{\mathcal{B}} \mid \neg E_{\mathsf{iden\text{-}bit}}].$$

□

□

4 Strong Monogamy Property of Coset States with Auxiliary Inputs

In this section, we introduce a new variant of the strong monogamy property of coset states that considers certain auxiliary inputs, and prove it.

Notations. For a subspace $A \subseteq \mathbb{F}_2^n$, we use the following notations introduced by [CLLZ21, AKL+22].

- $A^\perp := \{b \in \mathbb{F}_2^n : \forall a \in A,\ \langle a, b\rangle = 0\}$.
- Can_A is a function such that for any $s \in \mathbb{F}_2^n$, $\mathsf{Can}_A(s)$ is the lexicographically smallest vector contained in $A + s$, which we call the canonical representative of coset $A + s$. For any $s' \in A + s$, we have $\mathsf{Can}_A(s') = \mathsf{Can}_A(s)$. Can_A is polynomial-time computable given the description of A.
- $\mathsf{CS}(A) = \{\mathsf{Can}_A(s) : s \in \mathbb{F}_2^n\}$.

For a subspace $A \subseteq \mathbb{F}_2^n$ and $(s,t) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$, we define the *coset state* $|A_{s,t}\rangle$ as

$$|A_{s,t}\rangle := \frac{1}{\sqrt{|A|}} \sum_{a \in A} (-1)^{\langle a,t \rangle} |a+s\rangle .$$

By applying $H^{\otimes n}$ to $|A_{s,t}\rangle$, we obtain $\left|A^{\perp}_{t,s}\right\rangle$, where H is the Hadamard gate (thus $H^{\otimes n}$ is the Quantum Fourier Transformation over $\mathbb{F}_2^n$.)

When the context is clear, we often abuse notation by writing A to refer to its description, represented as a tuple of basis vectors. We often write $i\mathcal{O}(A+s)$ to denote an obfuscation by $i\mathcal{O}$ of the program that verifies membership in $A+s$.

We remark the following easy lemma, which is obvious from the definitions.

Lemma 4.1. *For a subspace $A \subseteq \mathbb{F}_2^n$, $s \in \mathsf{CS}(A)$, and $s' \in \mathbb{F}_2^n$, the following two conditions are equivalent:*

1. $s = s'$
2. $s' \in A + s$ *and* $\mathsf{Can}_A(s') = s'$.

Strong monogamy property. An information-theoretic property known as the strong monogamy property of coset states (Theorem 4.4) was conjectured by Coladangelo et al.[CLLZ21] and later proven by Culf and Vidick[CV22]. Coladangelo et al. [CLLZ21] also extended this property to a computational version using iO. In this paper, we introduce a further variant of the property that considers an indistinguishability game instead of a search game, and, more importantly, allows the adversary to receive certain auxiliary inputs.[15] The formal statement is given below:

Theorem 4.2 (Computational Strong Indistinguishability Monogamy Property of Coset States with Simulatable Auxiliary Inputs). *Let $i\mathcal{O}$ be iO. Let $n \in \mathbb{N}$ be a polynomial in λ. Let $\mathcal{Aux} = (\mathcal{AuxSetup}, \mathcal{AuxGen}_{\mathcal{B}}, \mathcal{AuxGen}_{\mathcal{C}})$ be a tuple of QPT algorithms with the following syntax.*

$\mathcal{AuxSetup}(1^\lambda)$**:** *This algorithm takes a security parameter 1^λ, and outputs a classical public parameter* pp.

$\mathcal{AuxGen}_{\mathcal{X}}(\mathsf{pp}, A, r_{\mathcal{X}})$ where $\mathcal{X} \in \{\mathcal{B}, \mathcal{C}\}$**:** *This algorithm takes a public parameter* pp*, a description of a subspace $A \subseteq \mathbb{F}_2^n$, and $r_{\mathcal{X}} \in \{0,1\}^n$, and outputs a pair of classical strings $(z_{1,\mathcal{X}}, z_{2,\mathcal{X}})$.*

We assume that, the first coordinate of the output of $\mathcal{AuxGen}_{\mathcal{X}}(\mathsf{pp}, A, r_{\mathcal{X}})$ can be statistically simulated from pp *alone, without using $(A, r_{\mathcal{X}})$. That is, for $\mathcal{X} \in \{\mathcal{B}, \mathcal{C}\}$, there exists a (not necessarily polynomial-time) quantum algorithm $\mathcal{Sim}_{\mathcal{X}}$ such that for any subspace $A \subseteq \mathbb{F}_2^n$ and any $r_{\mathcal{X}} \in \{0,1\}^n$, the following distributions are statistically indistinguishable:*

$$\left\{ (\mathsf{pp}, z_{1,\mathcal{X}}) : \begin{array}{l} \mathsf{pp} \leftarrow \mathcal{AuxSetup}(1^\lambda) \\ (z_{1,\mathcal{X}}, z_{2,\mathcal{X}}) \leftarrow \mathcal{AuxGen}_{\mathcal{X}}(\mathsf{pp}, A, r_{\mathcal{X}}) \end{array} \right\} \approx \left\{ (\mathsf{pp}, z_{1,\mathcal{X}}) : \begin{array}{l} \mathsf{pp} \leftarrow \mathcal{AuxSetup}(1^\lambda) \\ z_{1,\mathcal{X}} \leftarrow \mathcal{Sim}_{\mathcal{X}}(\mathsf{pp}) \end{array} \right\}.$$

Consider the following experiment $\mathsf{Exp}^{\mathsf{comp\text{-}strong\text{-}ind\text{-}moe\text{-}aux}}_{i\mathcal{O},\mathcal{Aux},\mathcal{A}_{\mathsf{moe}}}(\lambda)$ between a challenger and an adversary $\mathcal{A}_{\mathsf{moe}} = (\mathcal{A}, \mathcal{B}, \mathcal{C})$.

1. *The challenger picks a uniformly random subspace $A \subseteq \mathbb{F}_2^n$ of dimension $n/2$, and two uniformly random canonical representatives $(s,t) \in \mathsf{CS}(A) \times \mathsf{CS}(A^\perp)$. The challenger picks $r_{\mathcal{B}}, r_{\mathcal{C}} \leftarrow \{0,1\}^n$, and generates $\mathsf{pp} \leftarrow \mathcal{AuxSetup}(1^\lambda)$, $(z_{1,\mathcal{B}}, z_{2,\mathcal{B}}) \leftarrow \mathcal{AuxGen}_{\mathcal{B}}(\mathsf{pp}, A, r_{\mathcal{B}})$, and $(z_{1,\mathcal{C}}, z_{2,\mathcal{C}}) \leftarrow \mathcal{AuxGen}_{\mathcal{C}}(\mathsf{pp}, A, r_{\mathcal{C}})$. It sends $|A_{s,t}\rangle$, $i\mathcal{O}(A+s)$, $i\mathcal{O}(A^\perp + t)$, pp, $z_{1,\mathcal{B}}$, and $z_{1,\mathcal{C}}$ to $\mathcal{A}$.*
2. *$\mathcal{A}$ creates a bipartite state q over registers $\mathsf{R}_{\mathcal{B}}$ and $\mathsf{R}_{\mathcal{C}}$. Then, $\mathcal{A}$ sends register $\mathsf{R}_{\mathcal{B}}$ to $\mathcal{B}$ and register $\mathsf{R}_{\mathcal{C}}$ to $\mathcal{C}$.*
3. *The challenger sends $(A, r_{\mathcal{B}}, z_{2,\mathcal{B}})$ to $\mathcal{B}$ and $(A, r_{\mathcal{C}}, z_{2,\mathcal{C}})$ to $\mathcal{C}$.*
4. *$\mathcal{B}$ and $\mathcal{C}$ respectively output $b_{\mathcal{B}}$ and $b_{\mathcal{C}}$. The challenger outputs 1 if $b_{\mathcal{B}} \oplus b_{\mathcal{C}} = \langle r_{\mathcal{B}}, s \rangle \oplus \langle r_{\mathcal{C}}, t \rangle$ and outputs 0 otherwise.*

[15] An indistinguishability version without auxiliary inputs was already introduced in [KY25b].

Assuming $i\mathcal{O}$ is a secure iO and there exist OWFs, for any QPT adversary $\mathcal{A}_{\mathsf{moe}} = (\mathcal{A}, \mathcal{B}, \mathcal{C})$, it holds that

$$\Pr\left[\mathsf{Exp}^{\mathsf{comp\text{-}strong\text{-}ind\text{-}moe\text{-}aux}}_{i\mathcal{O},\mathcal{A}ux,\mathcal{A}_{\mathsf{moe}}}(\lambda) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

Remark 4.3. We do not need to explicitly provide $(A, r_{\mathcal{B}})$ to $\mathcal{B}$ and $(A, r_{\mathcal{C}})$ to $\mathcal{C}$, as these can be included in $z_{2,\mathcal{B}}$ and $z_{2,\mathcal{C}}$ without loss of generality. However, we present them explicitly here for clarity.

We prove Theorem 4.2 in Section 4.1.

4.1 Proof of Theorem 4.2

In this subsection, we prove Theorem 4.2. Our proof builds on the strong monogamy property of coset states [CLLZ21, CV22], which we gradually extend to eventually establish the desired result.

We begin by recalling the strong monogamy property of coset states.

Theorem 4.4 (Strong Monogamy Property of Coset States [CLLZ21, CV22]). *Let $n \in \mathbb{N}$ be a polynomial in λ. Consider the following experiment $\mathsf{Exp}^{\mathsf{strong\text{-}moe}}_{\mathcal{A}_{\mathsf{moe}}}(\lambda)$ between a challenger and an adversary $\mathcal{A}_{\mathsf{moe}} = (\mathcal{A}, \mathcal{B}, \mathcal{C})$.*

1. *The challenger picks a uniformly random subspace $A \subseteq \mathbb{F}_2^n$ of dimension $n/2$, and vectors $(s,t) \leftarrow \mathbb{F}_2^n \times \mathbb{F}_2^n$. It sends $|A_{s,t}\rangle$ to $\mathcal{A}$.*
2. *$\mathcal{A}$ creates a bipartite state q over registers $\mathsf{R}_{\mathcal{B}}$ and $\mathsf{R}_{\mathcal{C}}$. Then, $\mathcal{A}$ sends register $\mathsf{R}_{\mathcal{B}}$ to $\mathcal{B}$ and register $\mathsf{R}_{\mathcal{C}}$ to $\mathcal{C}$.*
3. *The challenger sends A to both $\mathcal{B}$ and $\mathcal{C}$.*
4. *$\mathcal{B}$ and $\mathcal{C}$ respectively output $s_{\mathcal{B}}$ and $t_{\mathcal{C}}$. The challenger outputs 1 if $s_{\mathcal{B}} \in (A+s)$ and $t_{\mathcal{C}} \in (A^{\perp}+t)$, and outputs 0 otherwise*

For any (not necessarily polynomial-time) quantum adversary $\mathcal{A}_{\mathsf{moe}} = (\mathcal{A}, \mathcal{B}, \mathcal{C})$, it holds that

$$\Pr\left[\mathsf{Exp}^{\mathsf{strong\text{-}moe}}_{\mathcal{A}_{\mathsf{moe}}}(\lambda) = 1\right] \leq \mathsf{negl}(\lambda).$$

Remark 4.5. In Theorem 4.4, the pair (s,t) is sampled uniformly from the entire space $\mathbb{F}_2^n \times \mathbb{F}_2^n$, rather than from $\mathsf{CS}(A) \times \mathsf{CS}(A^{\perp})$ as in Theorem 4.2. Although both versions are equivalent at this point, this choice is motivated by the fact that the proof of the extended version of the property, presented in Corollary 4.6, appears to rely on (s,t) being uniformly distributed over $\mathbb{F}_2^n \times \mathbb{F}_2^n$. The discrepancy in the distribution of (s,t) is resolved later in the proof, specifically during the game hops from Hyb_0 to Hyb_2 in the proof of Theorem 4.2, given at the end of this subsection.

[CLLZ21] showed that Theorem 4.4 implies the following corollary, which we refer to as the *extended strong monogamy property* of coset states. In this variant, the adversary $\mathcal{A}$ additionally receives the descriptions of a uniformly random subspace $B \subseteq \mathbb{F}_2^n$ of dimension $3n/4$ that contains A, and a uniformly random subspace $C \subseteq A$ of dimension $n/4$, along with the vectors $s+v$ and $t+w$, where $v \leftarrow B$ and $w \leftarrow C^{\perp}$. The differences from Theorem 4.4 are highlighted in bold red text.

Corollary 4.6 (Extended Strong Monogamy Property of Coset States [CLLZ21][16]). *Let $n \in \mathbb{N}$ be a polynomial in λ. Consider the following experiment $\mathsf{Exp}^{\mathsf{ex\text{-}strong\text{-}moe}}_{\mathcal{A}_{\mathsf{moe}}}(\lambda)$ between a challenger and an adversary $\mathcal{A}_{\mathsf{moe}} = (\mathcal{A}, \mathcal{B}, \mathcal{C})$.*

1. *The challenger picks a uniformly random subspace $A \subseteq \mathbb{F}_2^n$ of dimension $n/2$,* **a uniformly random subspace B of dimension $3n/4$ that contains A, and a uniformly random subspace $C \subseteq A$ of dimension $n/4$.** *The challenger picks vectors $(s,t) \leftarrow \mathbb{F}_2^n \times \mathbb{F}_2^n$* **and $(v,w) \leftarrow B \times C^{\perp}$.**

   It sends $|A_{s,t}\rangle$, **the descriptions of B and C, and vectors $s+v$ and $t+w$** *to $\mathcal{A}$.*

[16] The reduction from Corollary 4.6 to Theorem 4.4 is provided in Lemma C.16 of the arXiv version of [CLLZ21]: https://arxiv.org/pdf/2107.05692.

2. *$\mathcal{A}$ creates a bipartite state q over registers $\mathsf{R}_{\mathcal{B}}$ and $\mathsf{R}_{\mathcal{C}}$. Then, $\mathcal{A}$ sends register $\mathsf{R}_{\mathcal{B}}$ to $\mathcal{B}$ and register $\mathsf{R}_{\mathcal{C}}$ to $\mathcal{C}$.*

3. *The challenger sends A to both $\mathcal{B}$ and $\mathcal{C}$.*

4. *$\mathcal{B}$ and $\mathcal{C}$ respectively output $s_{\mathcal{B}}$ and $t_{\mathcal{C}}$. The challenger outputs 1 if $s_{\mathcal{B}} \in (A+s)$ and $t_{\mathcal{C}} \in (A^{\perp}+t)$, and outputs 0 otherwise*

For any (not necessarily polynomial-time) quantum adversary $\mathcal{A}_{\mathsf{moe}} = (\mathcal{A}, \mathcal{B}, \mathcal{C})$, it holds that

$$\Pr\left[\mathsf{Exp}^{\mathsf{ex\text{-}strong\text{-}moe}}_{\mathcal{A}_{\mathsf{moe}}}(\lambda) = 1\right] \leq \mathsf{negl}(\lambda).$$

In Corollary 4.6, note that the statement remains equivalent even if the goals of $\mathcal{B}$ and $\mathcal{C}$ are modified to output $\mathsf{Can}_A(s)$ and $\mathsf{Can}_{A^{\perp}}(t)$, respectively.[17] With this observation, the following corollary follows immediately by applying the simultaneous quantum Goldreich-Levin lemma (Lemma 2.8) to Corollary 4.6. The differences from Corollary 4.6 are highlighted in bold red text.

Corollary 4.7 (Extended Strong Indistinguishability Monogamy Property of Coset States). *Let $n \in \mathbb{N}$ be a polynomial in λ. Consider the following experiment $\mathsf{Exp}^{\mathsf{ex\text{-}strong\text{-}ind\text{-}moe}}_{\mathcal{A}_{\mathsf{moe}}}(\lambda)$ between a challenger and an adversary $\mathcal{A}_{\mathsf{moe}} = (\mathcal{A}, \mathcal{B}, \mathcal{C})$.*

1. *The challenger picks a uniformly random subspace $A \subseteq \mathbb{F}_2^n$ of dimension $n/2$, a uniformly random subspace B of dimension $3n/4$ that contains A, and a uniformly random subspace $C \subseteq A$ of dimension $n/4$. The challenger picks vectors $(s,t) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$ and $(v,w) \leftarrow B \times C^{\perp}$.*

   It sends $|A_{s,t}\rangle$, the descriptions of B and C, and vectors $s+v$ and $t+w$ to $\mathcal{A}$.

2. *$\mathcal{A}$ creates a bipartite state q over registers $\mathsf{R}_{\mathcal{B}}$ and $\mathsf{R}_{\mathcal{C}}$. Then, $\mathcal{A}$ sends register $\mathsf{R}_{\mathcal{B}}$ to $\mathcal{B}$ and register $\mathsf{R}_{\mathcal{C}}$ to $\mathcal{C}$.*

3. **The challenger chooses** $r_{\mathcal{B}}, r_{\mathcal{C}} \leftarrow \{0,1\}^n$**.** *The challenger sends $(A, r_{\mathcal{B}})$ to $\mathcal{B}$, and $(A, r_{\mathcal{C}})$ to $\mathcal{C}$.*

4. *$\mathcal{B}$ and $\mathcal{C}$ respectively output $b_{\mathcal{B}}$* **and** *$b_{\mathcal{C}}$. The challenger outputs 1* **if** *$b_{\mathcal{B}} \oplus b_{\mathcal{C}} = \langle r_{\mathcal{B}}, \mathsf{Can}_A(s)\rangle \oplus \langle r_{\mathcal{C}}, \mathsf{Can}_{A^{\perp}}(t)\rangle$ and outputs 0 otherwise.*

For any (not necessarily polynomial-time) quantum adversary $\mathcal{A}_{\mathsf{moe}} = (\mathcal{A}, \mathcal{B}, \mathcal{C})$, it holds that

$$\Pr\left[\mathsf{Exp}^{\mathsf{ex\text{-}strong\text{-}ind\text{-}moe}}_{\mathcal{A}_{\mathsf{moe}}}(\lambda) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

Next, we consider a variant of the above property in which $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ receive auxiliary inputs as in Theorem 4.2, but do not receive the obfuscated programs. The differences from Corollary 4.7 are highlighted in bold red text.

Corollary 4.8 (Extended Strong Indistinguishability Monogamy Property of Coset States with Simulatable Auxiliary Inputs). *Let $n \in \mathbb{N}$ be a polynomial in λ.* **Let** *$\mathcal{Aux} = (\mathcal{AuxSetup}, \mathcal{AuxGen}_{\mathcal{B}}, \mathcal{AuxGen}_{\mathcal{C}})$* **and** *$(\mathcal{Sim}_{\mathcal{B}}, \mathcal{Sim}_{\mathcal{C}})$* **be (not necessarily polynomial-time) quantum algorithms satisfying the conditions in Theorem 4.2.** *Consider the following experiment $\mathsf{Exp}^{\mathsf{ex\text{-}strong\text{-}ind\text{-}moe\text{-}aux}}_{\mathcal{Aux},\mathcal{A}_{\mathsf{moe}}}(\lambda)$ between a challenger and an adversary $\mathcal{A}_{\mathsf{moe}} = (\mathcal{A}, \mathcal{B}, \mathcal{C})$.*

1. *The challenger picks a uniformly random subspace $A \subseteq \mathbb{F}_2^n$ of dimension $n/2$, a uniformly random subspace B of dimension $3n/4$ that contains A, and a uniformly random subspace $C \subseteq A$ of dimension $n/4$. The challenger picks vectors $(s,t) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$ and $(v,w) \leftarrow B \times C^{\perp}$.* **The challenger picks** *$r_{\mathcal{B}}, r_{\mathcal{C}} \leftarrow \{0,1\}^n$***, and generates** *$\mathsf{pp} \leftarrow \mathcal{AuxSetup}(1^{\lambda})$***,** *$(z_{1,\mathcal{B}}, z_{2,\mathcal{B}}) \leftarrow \mathcal{AuxGen}_{\mathcal{B}}(\mathsf{pp}, A, r_{\mathcal{B}})$***, and** *$(z_{1,\mathcal{C}}, z_{2,\mathcal{C}}) \leftarrow \mathcal{AuxGen}_{\mathcal{C}}(\mathsf{pp}, A, r_{\mathcal{C}})$***.**

   It sends $|A_{s,t}\rangle$, the descriptions of B and C, vectors $s+v$ and $t+w$, pp**,** *$z_{1,\mathcal{B}}$***, and** *$z_{1,\mathcal{C}}$ to $\mathcal{A}$.*

2. *$\mathcal{A}$ creates a bipartite state q over registers $\mathsf{R}_{\mathcal{B}}$ and $\mathsf{R}_{\mathcal{C}}$. Then, $\mathcal{A}$ sends register $\mathsf{R}_{\mathcal{B}}$ to $\mathcal{B}$ and register $\mathsf{R}_{\mathcal{C}}$ to $\mathcal{C}$.*

3. *The challenger sends $(A, r_{\mathcal{B}}, z_{2,\mathcal{B}})$ to $\mathcal{B}$, and $(A, r_{\mathcal{C}}, z_{2,\mathcal{C}})$ to $\mathcal{C}$.*

[17] This is because, given the description of A, one can efficiently compute $\mathsf{Can}_A(s_{\mathcal{B}}) = \mathsf{Can}_A(s)$ and $\mathsf{Can}_{A^{\perp}}(t_{\mathcal{C}}) = \mathsf{Can}_{A^{\perp}}(t)$ from $s_{\mathcal{B}} \in (A+s)$ and $t_{\mathcal{C}} \in (A^{\perp}+t)$, respectively.

4. *$\mathcal{B}$ and $\mathcal{C}$ respectively output $b_{\mathcal{B}}$ and $b_{\mathcal{C}}$. The challenger outputs 1 if $b_{\mathcal{B}} \oplus b_{\mathcal{C}} = \langle r_{\mathcal{B}}, \mathsf{Can}_A(s)\rangle \oplus \langle r_{\mathcal{C}}, \mathsf{Can}_{A^\perp}(t)\rangle$ and outputs 0 otherwise.*

For any (not necessarily polynomial-time) quantum adversary $\mathcal{A}_{\mathsf{moe}} = (\mathcal{A}, \mathcal{B}, \mathcal{C})$, it holds that

$$\Pr\left[\mathsf{Exp}^{\mathsf{ex\text{-}strong\text{-}ind\text{-}moe\text{-}aux}}_{\mathcal{AuX},\mathcal{A}_{\mathsf{moe}}}(\lambda) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

Proof of Corollary 4.8. Let $\mathcal{A}_{\mathsf{moe}} = (\mathcal{A}, \mathcal{B}, \mathcal{C})$ be a (not necessarily polynomial-time) quantum adversary as in Corollary 4.8. Then we construct a (not necessarily polynomial-time) quantum adversary $\mathcal{A}'_{\mathsf{moe}} = (\mathcal{A}', \mathcal{B}', \mathcal{C}')$ as in Corollary 4.7 as follows:

$\mathcal{A}'$: Upon receiving $(|A_{s,t}\rangle, B, C, s+v, t+w)$, it generates $\mathsf{pp} \leftarrow \mathcal{AuxSetup}(1^\lambda)$, $z_{1,\mathcal{B}} \leftarrow \mathcal{Sim}_{\mathcal{B}}(\mathsf{pp})$, and $z_{1,\mathcal{C}} \leftarrow \mathcal{Sim}_{\mathcal{C}}(\mathsf{pp})$. Then it runs $\mathcal{A}$ on input $(|A_{s,t}\rangle, B, C, s+v, t+w, z_{1,\mathcal{B}}, z_{1,\mathcal{C}})$ to obtain a bipartite state q over registers $\mathsf{R}_{\mathcal{B}}$ and $\mathsf{R}_{\mathcal{C}}$. It sends $z_{1,\mathcal{B}}$ and $q[\mathsf{R}_{\mathcal{B}}]$ to $\mathcal{B}'$, and $z_{1,\mathcal{C}}$ and $q[\mathsf{R}_{\mathcal{C}}]$ to $\mathcal{C}'$.

$\mathcal{B}'$: Upon receiving $(z_{1,\mathcal{B}}, q[\mathsf{R}_{\mathcal{B}}])$ from $\mathcal{A}'$ and $(A, r_{\mathcal{B}})$ from the challenger, it samples $(z_1, z_2) \leftarrow \mathcal{AuxGen}_{\mathcal{B}}(\mathsf{pp}, A, r_{\mathcal{B}})$ conditioned on $z_1 = z_{1,\mathcal{B}}$ (using unbounded computational power), and sets $z_{2,\mathcal{B}} = z_2$.[18] Finally, it runs $\mathcal{B}$ on input $(q[\mathsf{R}_{\mathcal{B}}], A, r_{\mathcal{B}}, z_{2,\mathcal{B}})$ to obtain $b_{\mathcal{B}}$, and outputs $b_{\mathcal{B}}$.

$\mathcal{C}'$: Upon receiving $(z_{1,\mathcal{C}}, q[\mathsf{R}_{\mathcal{C}}])$ from $\mathcal{A}'$ and $(A, r_{\mathcal{C}})$ from the challenger, it samples $(z_1, z_2) \leftarrow \mathcal{AuxGen}_{\mathcal{C}}(\mathsf{pp}, A, r_{\mathcal{C}})$ conditioned on $z_1 = z_{1,\mathcal{C}}$ (using unbounded computational power), and sets $z_{2,\mathcal{C}} = z_2$.[19] Finally, it runs $\mathcal{C}$ on input $(q[\mathsf{R}_{\mathcal{C}}], A, r_{\mathcal{C}}, z_{2,\mathcal{C}})$ to obtain $b_{\mathcal{C}}$, and outputs $b_{\mathcal{C}}$.

By the simulatability condition, for any A and r, with overwhelming probability over the choice of $\mathsf{pp} \leftarrow \mathcal{AuxSetup}(1^\lambda)$, $\mathcal{Sim}_{\mathcal{X}}(\mathsf{pp})$ statistically simulates the first coordinate of $\mathcal{AuxGen}_{\mathcal{X}}(\mathsf{pp}, A, r_{\mathcal{X}})$ for $\mathcal{X} \in \{\mathcal{B}, \mathcal{C}\}$. Therefore, the joint distribution of $(z_{1,\mathcal{B}}, z_{2,\mathcal{B}}, z_{1,\mathcal{C}}, z_{2,\mathcal{C}})$ generated by $\mathcal{A}'_{\mathsf{moe}}$ is statistically close to that in $\mathsf{Exp}^{\mathsf{ex\text{-}strong\text{-}ind\text{-}moe\text{-}aux}}_{\mathcal{AuxGen},\mathcal{A}_{\mathsf{moe}}}(\lambda)$. Apart from this, $\mathcal{A}'_{\mathsf{moe}}$ perfectly simulates the execution of $\mathsf{Exp}^{\mathsf{ex\text{-}strong\text{-}ind\text{-}moe\text{-}aux}}_{\mathcal{AuxGen},\mathcal{A}_{\mathsf{moe}}}(\lambda)$ for $\mathcal{A}_{\mathsf{moe}}$. Therefore, the difference between $\Pr\left[\mathsf{Exp}^{\mathsf{ex\text{-}strong\text{-}ind\text{-}moe\text{-}aux}}_{\mathcal{AuX},\mathcal{A}_{\mathsf{moe}}}(\lambda) = 1\right]$ and $\Pr\left[\mathsf{Exp}^{\mathsf{ex\text{-}strong\text{-}ind\text{-}moe}}_{\mathcal{A}'_{\mathsf{moe}}}(\lambda) = 1\right]$ is negligible. Hence, Corollary 4.8 follows directly from Corollary 4.7.

□

We reduce Theorem 4.2 to Corollary 4.8 using a technique similar to that of [CLLZ21], which upgrades the strong monogamy property of coset states (Theorem 4.4) to its computational analogue. This technique relies on a result by Zhandry [Zha19], which shows that, assuming the existence of OWFs, any iO also serves as a subspace-hiding obfuscator.

Below, we state the definition of a subspace-hiding obfuscator.

Definition 4.9 ([Zha19]). *A subspace hiding obfuscator (shO) for a field $\mathbb{F}$ and dimensions d_0, d_1 is a PPT algorithm shO satisfying the following:*

Syntax. shO *takes as input the description of a linear subspace $S \subseteq \mathbb{F}^n$ of dimension $d \in \{d_0, d_1\}$ and outputs a classical circuit $\hat{S}$.*

Correctness. *For any linear subspace $S \subseteq \mathbb{F}^n$ of dimension $d \in \{d_0, d_1\}$, it holds that*

$$\Pr\left[\forall x \in \mathbb{F}^n,\ S(x) = \hat{S}(x) : \hat{S} \leftarrow \mathsf{shO}(S)\right] \geq 1 - \mathsf{negl}(\lambda)$$

where $S(x)$ is the function that decides membership in S.

Security. *Consider the following game between an adversary and a challenger, indexed by a bit b.*

- *The adversary submits to the challenger a subspace S_0 of dimension d_0*

[18] If $z_{1,\mathcal{B}}$ is not in the support of the first coordinate of the output of $\mathcal{AuxGen}(\mathsf{pp}, A, r_{\mathcal{B}})$, the algorithm simply aborts.
[19] If $z_{1,\mathcal{C}}$ is not in the support of the first coordinate of the output of $\mathcal{AuxGen}(\mathsf{pp}, A, r_{\mathcal{C}})$, the algorithm simply aborts.

- *The challenger chooses a random subspace $S_1 \subseteq \mathbb{F}^n$ of dimension d_1 such that $S_0 \subseteq S_1$. It then runs $\hat{S} \leftarrow \mathsf{shO}(S_b)$, and gives $\hat{S}$ to the adversary*
- *The adversary makes a guess b' for b.*

For all QPT adversaries, it holds that $|\Pr[b' = b] - 1/2| \leq \mathsf{negl}(\lambda)$ in the above game.

Lemma 4.10 ([Zha19]). *If OWFs exist, then any iO, appropriately padded, is also a subspace hiding obfuscator for field $\mathbb{F}$ and dimensions d_0, d_1, as long as $|\mathbb{F}|^{n-d_1}$ is exponential.*

Remark 4.11. The original statement in [Zha19] assumes the existence of injective OWFs, but it is easy to see that keyed injective OWFs (Definition 2.5) suffice, which are known to exist if iO and OWFs exist [BPW16].

Then we prove Theorem 4.2.

Proof of Theorem 4.2. Let shO be a subspace hiding obfuscator for field $\mathbb{F}$ and dimensions $d_0 = n/2$ and $d_1 = 3n/4$. Such a subspace hiding obfuscator exists assuming the existence of iO and OWFs by Lemma 4.10.

For a QPT adversary $\mathcal{A}_{\mathsf{moe}} = (\mathcal{A}, \mathcal{B}, \mathcal{C})$, we consider the following sequence of hybrids.

Hyb_0**:** This is the original security experiment $\mathsf{Exp}^{\mathsf{comp\text{-}strong\text{-}ind\text{-}moe\text{-}aux}}_{i\mathcal{O},\mathcal{A}ux_{\mathcal{C}},\mathcal{A}_{\mathsf{moe}}}(\lambda)$ described in Theorem 4.2.

Hyb_1**:** This is identical to Hyb_0 except that the hybrid outputs 1 if $b_{\mathcal{B}} \oplus b_{\mathcal{C}} = \langle r_{\mathcal{B}}, \mathsf{Can}_A(s)\rangle \oplus \langle r_{\mathcal{C}}, \mathsf{Can}_{A^\perp}(t)\rangle$ and outputs 0 otherwise.

Since $s \in \mathsf{CS}(A)$ and $t \in \mathsf{CS}(A^\perp)$, we have $\mathsf{Can}_A(s) = s$ and $\mathsf{Can}_{A'}(t) = t$. Thus, we clearly have

$$\Pr[\mathsf{Hyb}_1 = 1] = \Pr[\mathsf{Hyb}_0 = 1].$$

Hyb_2**:** This is identical to Hyb_1 except that s and t are sampled uniformly from $\mathbb{F}_2^n$ instead of $\mathsf{CS}(A)$ and $\mathsf{CS}(A^\perp)$.

Note that the only information of (s,t) given to the adversary is $|A_{s,t}\rangle$, $i\mathcal{O}(A+s)$, and $i\mathcal{O}(A^\perp+t)$. For any $(s,t) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$, $|A_{s,t}\rangle = \left|A_{\mathsf{Can}_A(s),\mathsf{Can}_{A^\perp}(t)}\right\rangle$, and $i\mathcal{O}(A+s)$ and $i\mathcal{O}(A^\perp+t)$ are computationally indistinguishable from $i\mathcal{O}(A+\mathsf{Can}_A(s))$, and $i\mathcal{O}(A^\perp+\mathsf{Can}_{A^\perp}(t))$, respectively, by the security of iO. Moreover, we have $\mathsf{Can}_A(s) = \mathsf{Can}_A(\mathsf{Can}_A(s))$ and $\mathsf{Can}_{A^\perp}(t) = \mathsf{Can}_{A^\perp}(\mathsf{Can}_{A^\perp}(t))$. This means that the probability to output 1 only negligibly changes even if we modify Hyb_2 by replacing s and t with $\mathsf{Can}_A(s)$ and $\mathsf{Can}_{A^\perp}(t)$, respectively. For uniformly random $(s,t) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$, $(\mathsf{Can}_A(s), \mathsf{Can}_{A^\perp}(t))$ is uniformly random over $\mathsf{CS}(A) \times \mathsf{CS}(A^\perp)$. Thus the game obtained by replacing s and t in Hyb_2 with $\mathsf{Can}_A(s)$ and $\mathsf{Can}_{A^\perp}(t)$ is identical to Hyb_1. Thus, we have

$$|\Pr[\mathsf{Hyb}_2 = 1] - \Pr[\mathsf{Hyb}_1 = 1]| \leq \mathsf{negl}(\lambda).$$

Hyb_3**:** This is identical to Hyb_2 except that the first obfuscated program $i\mathcal{O}(A+s)$ given to $\mathcal{A}$ is replaced with $i\mathcal{O}(\mathsf{shO}(A)(\cdot - s))$ where $\mathsf{shO}(A)(\cdot - s)$ is a program that takes $x \in \mathbb{F}^n$ as input and outputs $\mathsf{shO}(A)(x-s)$.

By the correctness of shO, $\mathsf{shO}(A)(\cdot - s)$ is functionally equivalent to the membership testing program of $A+s$ with an overwhelming probability. Thus, by the security of $i\mathcal{O}$, we have

$$|\Pr[\mathsf{Hyb}_3 = 1] - \Pr[\mathsf{Hyb}_2 = 1]| \leq \mathsf{negl}(\lambda).$$

Hyb_4**:** This is identical to Hyb_3 except that the first obfuscated program $i\mathcal{O}(\mathsf{shO}(A)(\cdot - s))$ given to $\mathcal{A}$ is replaced with $i\mathcal{O}(\mathsf{shO}(B)(\cdot - s))$ where $B \subseteq \mathbb{F}_2^n$ is a uniformly random subspace of dimension $3n/4$ that contains A.

By the security of shO, we have

$$|\Pr[\mathsf{Hyb}_4 = 1] - \Pr[\mathsf{Hyb}_3 = 1]| \leq \mathsf{negl}(\lambda).$$

Hyb_5**:** This is identical to Hyb_4 except that the first obfuscated program $i\mathcal{O}(\mathsf{shO}(B)(\cdot - s))$ given to $\mathcal{A}$ is replaced with $i\mathcal{O}(\mathsf{shO}(B)(\cdot - (s+v)))$ for a uniformly random element $v \in B$.

By the correctness of shO, $\mathsf{shO}(B)(\cdot - s)$ and $\mathsf{shO}(B)(\cdot - (s+v))$ are functionally equivalent with an overwhelming probability. Thus, by the security of $i\mathcal{O}$, we have

$$|\Pr[\mathsf{Hyb}_5 = 1] - \Pr[\mathsf{Hyb}_4 = 1]| \leq \mathsf{negl}(\lambda).$$

Hyb_6**:** This is identical to Hyb_5 except that the second obfuscated program $i\mathcal{O}(A^\perp + t)$ given to $\mathcal{A}$ is replaced with $i\mathcal{O}(\mathsf{shO}(C^\perp)(\cdot - (t+w)))$ where C is a uniformly random subspace of A of dimension $n/4$ and $w \in C^\perp$ is a uniformly random element of $C^\perp$.

By a similar argument to the hops from Hyb_2 to Hyb_5, we have

$$|\Pr[\mathsf{Hyb}_6 = 1] - \Pr[\mathsf{Hyb}_5 = 1]| \leq \mathsf{negl}(\lambda).$$

One can see that Hyb_6 is identical to $\mathsf{Exp}^{\mathsf{ex\text{-}strong\text{-}ind\text{-}moe\text{-}aux}}_{\mathcal{A}_{\mathsf{aux}},\mathcal{A}_{\mathsf{moe}}}(\lambda)$ except that $\mathcal{A}$ is given $i\mathcal{O}(\mathsf{shO}(B)(\cdot - (s+v)))$ and $i\mathcal{O}(\mathsf{shO}(C^\perp)(\cdot - (t+w)))$ instead of $(B, C, s+v, t+w)$. Since $i\mathcal{O}(\mathsf{shO}(B)(\cdot - (s+v)))$ and $i\mathcal{O}(\mathsf{shO}(C^\perp)(\cdot - (t+w)))$ can be efficiently generated from $(B, C, s+v, t+w)$, a straightforward reduction to Corollary 4.8 gives us the following:

$$\Pr[\mathsf{Hyb}_6 = 1] \leq 1/2 + \mathsf{negl}(\lambda).$$

Combining the above, we can conclude that for any QPT adversary $\mathcal{A}_{\mathsf{moe}}$,

$$\Pr[\mathsf{Hyb}_0 = 1] \leq 1/2 + \mathsf{negl}(\lambda),$$

completing the proof of Theorem 4.2. □

5 Construction of UPO

We prove the following theorem.

Theorem 5.1. *Assuming the existence of polynomially secure iO and the LWE assumption, for any constant $c > 0$ and any polynomials ℓ_{inp} and ℓ_{out}, there exists a UPO for the circuit class $\mathsf{Circ} = \{C : \{0,1\}^{\ell_{\mathsf{inp}}} \to \{0,1\}^{\ell_{\mathsf{out}}}\}$ that satisfies $\mathcal{D}$-generalized UPO$^+$ security and $\mathcal{D}$-generalized UPO security for any QPT-samplable product distributions $\mathcal{D} = \mathcal{D}_B \times \mathcal{D}_C$, where $\mathcal{D}_B$ and $\mathcal{D}_C$ have min-entropy at least λ^c.*

Moreover, if $\mathcal{D}_B$ and $\mathcal{D}_C$ are uniformly random, the existence of OWFs (in place of the LWE assumption) together with iO suffices.

5.1 Construction

Let n, ℓ_{inp}, and ℓ_{out} be polynomials in the security parameter λ.

We use the following building blocks.

- iO for polynomial size classical circuits $i\mathcal{O}$.
- PPRF $\mathsf{PPRF} = (\mathsf{F}, \mathsf{Puncture})$ with the domain $\{0,1\}^{\ell_{\mathsf{inp}}}$ and the range $\{0,1\}^{\ell_{\mathsf{out}}}$.

We remark that additional cryptographic primitives are used in the security proof, as described in the relevant subsections.

We below construct a UPO scheme $\mathsf{UPO} = \mathsf{UPO}.(\mathcal{Obf}, \mathcal{Eval})$ for the circuit class $\mathsf{Circ} = \{C : \{0,1\}^{\ell_{\mathsf{inp}}} \to \{0,1\}^{\ell_{\mathsf{out}}}\}$.

$\mathsf{UPO}.\mathcal{Obf}(1^\lambda, C)$**:**

$P_1[\mathsf{C}_{\mathsf{mc}}, C, K](a, x)$

Hardwired constant: A description of a boolean circuits C_{mc} and C, and a string K.

Input: (a, x).

1. If $\mathsf{C}_{\mathsf{mc}}(a) = 0$, output $\bot$. Otherwise, go to the next step.
2. Output $\mathsf{F}_K(x) \oplus C(x)$.

Figure 5: The description of the circuit P_1.

$P_2[\mathsf{C}^{\bot}_{\mathsf{mc}}, K](a, x)$

Hardwired constant: A description of a boolean circuits $\mathsf{C}^{\bot}_{\mathsf{mc}}$ and a string K.

Input: (a, x).

1. If $\mathsf{C}^{\bot}_{\mathsf{mc}}(a) = 0$, output $\bot$. Otherwise, go to the next step.
2. Output $\mathsf{F}_K(x)$.

Figure 6: The description of the circuit P_2.

- Generate a uniformly random subspace $A \subseteq \mathbb{F}_2^n$ of dimension $\frac{n}{2}$, and two uniformly random elements $(s, t) \in \mathsf{CS}(A) \times \mathsf{CS}(A^{\bot})$, where n is specified later.
- Define circuits $\mathsf{C}_{\mathsf{mc}} := i\mathcal{O}(A + s)$ and $\mathsf{C}^{\bot}_{\mathsf{mc}} := i\mathcal{O}(A^{\bot} + t)$.
- Generate a key $K \leftarrow \{0,1\}^{\lambda}$ for PPRF.
- Generate two obfuscated circuits $\widetilde{P}_1 \leftarrow i\mathcal{O}(P_1[\mathsf{C}_{\mathsf{mc}}, C, K])$ and $\widetilde{P}_2 \leftarrow i\mathcal{O}(P_2[\mathsf{C}^{\bot}_{\mathsf{mc}}, K])$, where P_1 and P_2 are the circuits described in Figure 5 and Figure 6.
- Output $\widetilde{C} := (|A_{s,t}\rangle, \widetilde{P}_1, \widetilde{P}_2)$.

$\mathsf{UPO}.\mathcal{E}val(\widetilde{C}, x)$**:**

- Parse $\widetilde{C} = (\rho, \widetilde{P}_1, \widetilde{P}_2)$, where ρ is a quantum state, and $\widetilde{P}_1$ and $\widetilde{P}_2$ are interpreted as descriptions of circuits.
- Let E_1 and E_2 be unitaries that respectively work as follows:

$$|z_1\rangle |z_2\rangle \mapsto |z_1\rangle \left|z_2 \oplus \widetilde{P}_1(z_1, x)\right\rangle \quad \text{and} \quad |z_1\rangle |z_2\rangle \mapsto |z_1\rangle \left|z_2 \oplus \widetilde{P}_2(z_1, x)\right\rangle$$

  for any $z_1 \in \{0,1\}^n$ and $z_2 \in \{0,1\}^{\ell_{\mathsf{out}}}$.
- Apply E_1 to $\rho \otimes \left|0^{\ell_{\mathsf{out}}}\right\rangle$ and measure the second register to obtain an outcome y_1. Then, trace out the second register and obtain the state ρ'.
- Apply E_2 to $H^{\otimes n}\rho' \otimes \left|0^{\ell_{\mathsf{out}}}\right\rangle$ and measure the second register to obtain an outcome y_2.
- Output $Y := y_1 \oplus y_2$.

Perfect correctness. We prove

$$\Pr\left[\mathsf{UPO}.\mathcal{E}val(\widetilde{C}, x) = C(x) \;\middle|\; \widetilde{C} \leftarrow \mathsf{UPO}.\mathcal{O}bf(1^{\lambda}, C)\right] = 1$$

for any $C \in \mathsf{Circ}$ and $x \in \{0,1\}^{\ell_{\mathsf{inp}}}$. Fix $C \in \mathsf{Circ}$ and $x \in \{0,1\}^{\ell_{\mathsf{inp}}}$. Suppose we generate $\widetilde{C} := (|A_{s,t}\rangle, \widetilde{P}, \widetilde{P}_2)$ as prescribed in the description of $\mathsf{UPO}.\mathcal{O}bf$. Letting y_1 and y_2 be the values respectively computed in the third and forth item of $\mathsf{UPO}.\mathcal{E}val(\widetilde{C}, x)$, we prove the followings.

- We prove that $y_1 = \mathsf{F}_K(x) \oplus C(x)$ holds with probability 1. For any $a \in A + s$, we have $\widetilde{P}_1(a, x) = \mathsf{F}_K(x) \oplus C(x)$ from the correctness of $i\mathcal{O}$. Thus, given that $|A_{s,t}\rangle = \frac{1}{\sqrt{|A|}} \sum_{a \in A} (-1)^{\langle a,t \rangle} |a + s\rangle$, if we apply the unitary E_1 to $|A_{s,t}\rangle \otimes \left|0^{\ell_{\mathsf{out}}}\right\rangle$, we obtain the state $|A_{s,t}\rangle \otimes |\mathsf{F}_K(x) \oplus C(x)\rangle$. This means $y = \mathsf{F}_K(x) \oplus C(x)$ holds with probability 1, and the state obtained by tracing out the second register after measuring y is $|A_{s,t}\rangle$.
- We then prove $y_2 = \mathsf{F}_K(x)$. For any $a \in A^\perp + t$, we have $\widetilde{P}_2(a, x) = \mathsf{F}_K(x)$ from the correctness of $i\mathcal{O}$. Thus, given that $H^{\otimes n} |A_{s,t}\rangle = \frac{1}{\sqrt{|A^\perp|}} \sum_{a \in A^\perp} (-1)^{\langle a,s \rangle} |a + t\rangle$, if we apply the unitary E_2 to $H^{\otimes n} |A_{s,t}\rangle \otimes \left|0^{\ell_{\mathsf{out}}}\right\rangle$, we obtain the state $H^{\otimes n} |A_{s,t}\rangle \otimes |\mathsf{F}_K(x)\rangle$. This means $y_2 = \mathsf{F}_K(x)$ holds.

The above shows the perfect correctness of UPO.

5.2 Proof of Security

In the security proof, we additionally use the following primitives, where ℓ_{loss} is a polynomial in λ and $\ell_A := n^2/2$ is the number of bits needed to represent a subspace $A \subseteq \mathbb{F}_2^n$ of dimension $n/2$.

- Universal hash function family $\mathcal{H} : \{0,1\}^{\ell_{\mathsf{inp}}} \to \{0,1\}^{\ell_A + n}$.
- Collection of $(\ell_{\mathsf{inp}}, \ell_{\mathsf{loss}})$-lossy functions $\mathsf{LF} = (\mathsf{Gen}_{\mathsf{inj}}, \mathsf{Gen}_{\mathsf{loss}})$.

We prove the following theorem.

Theorem 5.2. *Let $\mathcal{D}$ be a distribution over $\{0,1\}^{\ell_{\mathsf{inp}}} \times \{0,1\}^{\ell_{\mathsf{aux}}}$ with conditional min-entropy at least $\ell_A + n + (\ell_{\mathsf{inp}} - \ell_{\mathsf{loss}}) + \omega(\log \lambda)$. Then, assuming the existence of $(\ell_{\mathsf{inp}}, \ell_{\mathsf{loss}})$-lossy functions LF, UPO satisfies $\mathcal{D}$-generalized UPO$^+$ security.*

We may choose ℓ_A and n to be arbitrarily small polynomials. Assuming the LWE assumption, by Theorem 2.7, we may also make $(\ell_{\mathsf{inp}} - \ell_{\mathsf{loss}})$ an arbitrarily small polynomial while allowing ℓ_{inp} and ℓ_{out} to be any polynomials. Moreover, $\mathcal{D}$-generalized UPO$^+$security implies $\mathcal{D}$-generalized UPO security without any additional assumption by Theorem 3.17 since our construction clearly satisfies iO security and there exist keyed injective $\mathcal{D}$-OWFs assuming the LWE assumption. Therefore, the former part of Theorem 5.1 follows.

Proof of Theorem 5.2. Let $\mathcal{A}_{\mathsf{cp}} = (\mathcal{A}, \mathcal{B}, \mathcal{C})$ be a QPT adversary against $\mathcal{D}$-generalized UPO$^+$security of UPO. We consider the following sequence of hybrids.

Hyb_0**:** This is the original security experiment $\mathsf{Exp}^{\mathsf{gen\text{-}upoplus}}_{\mathsf{UPO},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda)$ in Definition 3.14. More specifically, it works as follows.

1. On input 1^λ, $\mathcal{A}$ sends $C \in \mathsf{Circ}$ together with two circuits $\mu_{\mathcal{B}} : \{0,1\}^{\ell_{\mathsf{inp}}} \to \{0,1\}^{\ell_{\mathsf{out}}}$ and $\mu_{\mathcal{C}} : \{0,1\}^{\ell_{\mathsf{inp}}} \to \{0,1\}^{\ell_{\mathsf{out}}}$.
2. The challenger does the following.
   - Choose $\mathsf{coin}_{\mathcal{B}} \leftarrow \{0,1\}$, generate $x_{\mathcal{B}} \leftarrow \mathcal{D}_{\mathcal{B}}(1^\lambda)$, $Y_{\mathcal{B},0} \leftarrow C(x_{\mathcal{B}})$, and $Y_{\mathcal{B},1} \leftarrow \mu_{\mathcal{B}}(x_{\mathcal{B}})$.
   - Choose $\mathsf{coin}_{\mathcal{C}} \leftarrow \{0,1\}$, generate $x_{\mathcal{C}} \leftarrow \mathcal{D}_{\mathcal{C}}(1^\lambda)$, $Y_{\mathcal{C},0} \leftarrow C(x_{\mathcal{C}})$, and $Y_{\mathcal{C},1} \leftarrow \mu_{\mathcal{C}}(x_{\mathcal{C}})$.

   The challenger generates $\widetilde{C} := (|A_{s,t}\rangle, \widetilde{P}_1, \widetilde{P}_2) \leftarrow \mathsf{UPO}.\mathcal{O}bf(1^\lambda, C^*[x_{\mathcal{B}}, x_{\mathcal{C}}, Y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}}, Y_{\mathcal{C},\mathsf{coin}_{\mathcal{C}}}])$ and sends $\widetilde{C}$ to $\mathcal{A}$.
3. $\mathcal{A}$ creates a bipartite state q over registers $\mathsf{R}_{\mathcal{B}}$ and $\mathsf{R}_{\mathcal{C}}$. Then, $\mathcal{A}$ sends register $\mathsf{R}_{\mathcal{B}}$ to $\mathcal{B}$ and register $\mathsf{R}_{\mathcal{C}}$ to $\mathcal{C}$.
4. The challenger sends $x_{\mathcal{B}}$ and $x_{\mathcal{C}}$ to $\mathcal{B}$ and $\mathcal{C}$, respectively.
5. $\mathcal{B}$ and $\mathcal{C}$ respectively output $\mathsf{coin}'_{\mathcal{B}}$ and $\mathsf{coin}'_{\mathcal{C}}$. The challenger outputs 1 if $\mathsf{coin}'_{\mathcal{B}} \oplus \mathsf{coin}'_{\mathcal{C}} = \mathsf{coin}_{\mathcal{B}} \oplus \mathsf{coin}_{\mathcal{C}}$ otherwise outputs 0.

$P_1'[\mathsf{C}_{\mathsf{mc}}, C, x_{\mathcal{B}}, x_C, K\{x_{\mathcal{B}}, x_C\}, w_{\mathcal{B}}, w_C](a, x)$

Hardwired constant: Descriptions of boolean circuits C_{mc} andC and string $(x_{\mathcal{B}}, x_C, K\{x_{\mathcal{B}}, x_C\}, w_{\mathcal{B}}, w_C)$, where $K\{x_{\mathcal{B}}, x_C\}$ describes a key for PPRF punctured at $\{x_{\mathcal{B}}, x_C\}$.

Input: (a, x).

1. If $\mathsf{C}_{\mathsf{mc}}(a) = 0$, output $\bot$. Otherwise, go to the next step.
2. if $x = x_{\mathcal{B}}$, output the hardwired value $w_{\mathcal{B}}$. Otherwise, go to the next step.
3. if $x = x_C$, output the hardwired value w_C. Otherwise, go to the next step.
4. Output $\mathsf{F}_{K\{x_{\mathcal{B}}, x_C\}}(x) \oplus C(x)$.

Figure 7: The description of the circuit P_1'.

$P_2'[\mathsf{C}^{\bot}_{\mathsf{mc}}, x_{\mathcal{B}}, x_C, K\{x_{\mathcal{B}}, x_C\}, w_{\mathcal{B}}, w_C](a, x)$

Hardwired constant: A description of a boolean circuit $\mathsf{C}^{\bot}_{\mathsf{mc}}$ and string $(x_{\mathcal{B}}, x_C, K\{x_{\mathcal{B}}, x_C\}, w_{\mathcal{B}}, w_C)$, where $K\{x_{\mathcal{B}}, x_C\}$ describes a key for PPRF punctured at $\{x_{\mathcal{B}}, x_C\}$.

Input: (a, x).

1. If $\mathsf{C}^{\bot}_{\mathsf{mc}}(a) = 0$, output $\bot$. Otherwise, go to the next step.
2. if $x = x_{\mathcal{B}}$, output the hardwired value $w_{\mathcal{B}}$. Otherwise, go to the next step.
3. if $x = x_C$, output the hardwired value w_C. Otherwise, go to the next step.
4. Output $\mathsf{F}_{K\{x_{\mathcal{B}}, x_C\}}(x)$.

Figure 8: The description of the circuit P_2'.

By the definition, we clearly have

$$\Pr[\mathsf{Hyb}_0 = 1] = \Pr\left[\mathsf{Expt}^{\mathsf{upoplus}}_{\mathsf{UPO},\mathcal{D},\mathcal{A}_{\mathsf{cp}}}(\lambda) = 1\right].$$

We below use the circuits P_1' and P_2' respectively described in Figure 7 and Figure 8.

Hyb_1**:** This is identical to Hyb_0 except for the following changes, where $K\{x_{\mathcal{B}}, x_C\} \leftarrow \mathsf{Puncture}(K, \{x_{\mathcal{B}}, x_C\})$, $y_{\mathcal{B}} \leftarrow \mathsf{F}_K(x_{\mathcal{B}})$, and $y_C \leftarrow \mathsf{F}_K(x_C)$.

- $\widetilde{P}_1$ is generated as $\widetilde{P}_1 \leftarrow i\mathcal{O}(P_1'[\mathsf{C}_{\mathsf{mc}}, C, x_{\mathcal{B}}, x_C, K\{x_{\mathcal{B}}, x_C\}, y_{\mathcal{B}} \oplus Y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}}, y_C \oplus Y_{C,\mathsf{coin}_C}])$.
- $\widetilde{P}_2$ is generated as $\widetilde{P}_2 \leftarrow i\mathcal{O}(P_2'[\mathsf{C}^{\bot}_{\mathsf{mc}}, x_{\mathcal{B}}, x_C, K\{x_{\mathcal{B}}, x_C\}, y_{\mathcal{B}}, y_C])$.

From the security of $i\mathcal{O}$, we have

$$|\Pr[\mathsf{Hyb}_0 = 1] - \Pr[\mathsf{Hyb}_1 = 1]| = \mathsf{negl}(\lambda).$$

Hyb_2**:** This is identical to Hyb_1 except that $y_{\mathcal{B}}$ and y_C are generated as uniformly random strings instead of $\mathsf{F}_K(x_{\mathcal{B}})$ and $\mathsf{F}_K(x_C)$, respectively. From the security of PPRF, we have

$$|\Pr[\mathsf{Hyb}_1 = 1] - \Pr[\mathsf{Hyb}_2 = 1]| = \mathsf{negl}(\lambda).$$

Hereafter, we assume that $x_{\mathcal{B}} \neq x_C$, which holds with overwhelming probability.

Hyb_3**:** This is identical to Hyb_2 except that we replace y_C with $y_C \oplus Y_{C,\mathsf{coin}_C}$. More specifically, the following changes are applied.

- $\widetilde{P}_1$ is generated as $\widetilde{P}_1 \leftarrow i\mathcal{O}(P'_1[\mathsf{C}_{\mathsf{mc}}, C, x_{\mathcal{B}}, x_C, K\{x_{\mathcal{B}}, x_C\}, y_{\mathcal{B}} \oplus Y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}}, y_C])$.
- $\widetilde{P}_2$ is generated as $\widetilde{P}_2 \leftarrow i\mathcal{O}(P'_2[\mathsf{C}^{\perp}_{\mathsf{mc}}, x_{\mathcal{B}}, x_C, K\{x_{\mathcal{B}}, x_C\}, y_{\mathcal{B}}, y_C \oplus Y_{C,\mathsf{coin}_C}])$.

Since y_C is uniformly at random, so is $y_C \oplus Y_{C,\mathsf{coin}_C}$. Thus, we have

$$\Pr[\mathsf{Hyb}_2 = 1] = \Pr[\mathsf{Hyb}_3 = 1].$$

Hyb_4**:** This is identical to Hyb_3 except that $\mathsf{coin}_{\mathcal{B}}$ and coin_C are replaced with $\mathsf{coin}_{\mathcal{B}} \oplus \langle r_{\mathcal{B}}, s\rangle$ and $\mathsf{coin}_C \oplus \langle r_C, t\rangle$, respectively, where $r_{\mathcal{B}}, r_C \leftarrow \{0,1\}^n$. More specifically, the following three changes are applied.

- $\widetilde{P}_1$ is generated as $\widetilde{P}_1 \leftarrow i\mathcal{O}(P'_1[\mathsf{C}_{\mathsf{mc}}, C, x_{\mathcal{B}}, x_C, K\{x_{\mathcal{B}}, x_C\}, y_{\mathcal{B}} \oplus Y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}\oplus\langle r_{\mathcal{B}},s\rangle}, y_C])$.
- $\widetilde{P}_2$ is generated as $\widetilde{P}_2 \leftarrow i\mathcal{O}(P'_2[\mathsf{C}^{\perp}_{\mathsf{mc}}, x_{\mathcal{B}}, x_C, K\{x_{\mathcal{B}}, x_C\}, y_{\mathcal{B}}, y_C \oplus Y_{C,\mathsf{coin}_C\oplus\langle r_C,t\rangle}])$.
- The challenger outputs 1 if $\mathsf{coin}'_{\mathcal{B}} \oplus \mathsf{coin}'_C = \mathsf{coin}_{\mathcal{B}} \oplus \mathsf{coin}_C \oplus \langle r_{\mathcal{B}}, s\rangle \oplus \langle r_C, t\rangle$ and otherwise outputs 0.

Since $(\mathsf{coin}_{\mathcal{B}}, \mathsf{coin}_C)$ is distributed uniformly at random, so is $(\mathsf{coin}_{\mathcal{B}} \oplus \langle r_{\mathcal{B}}, s\rangle, \mathsf{coin}_C \oplus \langle r_C, t\rangle)$. Thus, we have

$$\Pr[\mathsf{Hyb}_3 = 1] = \Pr[\mathsf{Hyb}_4 = 1].$$

We below use the circuits Q_1 and Q_2 respectively described in Figure 9 and Figure 10.

Hyb_5**:** This is identical to Hyb_4 except for the following changes, where $h \leftarrow \mathcal{H}$, $F_{\mathsf{lf}} \leftarrow \mathsf{Gen}_{\mathsf{inj}}(1^\lambda)$, $\eta_{\mathcal{B}} = F_{\mathsf{lf}}(x_{\mathcal{B}})$, and $\eta_C = F_{\mathsf{lf}}(x_C)$.

- $\widetilde{P}_1$ is generated as $\widetilde{P}_1 \leftarrow i\mathcal{O}(Q_1[\mathsf{C}_{\mathsf{mc}}, C, \mu_{\mathcal{B}}, h, F_{\mathsf{lf}}, \mathsf{str}_{\mathcal{B}}, \eta_{\mathcal{B}}, \eta_C, K, y_{\mathcal{B}}, y_C, \mathsf{coin}_{\mathcal{B}}])$, where $\mathsf{str}_{\mathcal{B}} = (A \| r_{\mathcal{B}}) \oplus h(x_{\mathcal{B}})$.
- $\widetilde{P}_2$ is generated as $\widetilde{P}_2 \leftarrow i\mathcal{O}(Q_2[\mathsf{C}^{\perp}_{\mathsf{mc}}, C, \mu_C, h, F_{\mathsf{lf}}, \mathsf{str}_C, \eta_{\mathcal{B}}, \eta_C, K, y_{\mathcal{B}}, y_C, \mathsf{coin}_C])$, where $\mathsf{str}_C = (A^{\perp} \| r_C) \oplus h(x_C)$.

We prove that

$$P'_1[\mathsf{C}_{\mathsf{mc}}, C, x_{\mathcal{B}}, x_C, K\{x_{\mathcal{B}}, x_C\}, y_{\mathcal{B}} \oplus Y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}\oplus\langle r_{\mathcal{B}},s\rangle}, y_C]$$

and

$$Q_1[\mathsf{C}_{\mathsf{mc}}, C, \mu_{\mathcal{B}}, h, F_{\mathsf{lf}}, \mathsf{str}_{\mathcal{B}}, \eta_{\mathcal{B}}, \eta_C, K, y_{\mathcal{B}}, y_C, \mathsf{coin}_{\mathcal{B}}]$$

are functionally equivalent. For an input (a, x), we consider the following cases.

- The case where $C_{\mathsf{mc}}(a) = 0$ (i.e., $a \notin A + s$). In this case, both circuits output $\perp$ according to the first items of Figures 7 and 9, respectively.
- The case where $C_{\mathsf{mc}}(a) = 1$ (i.e., $a \in A + s$), and $x = x_{\mathcal{B}}$. In this case, $P'_1[\mathsf{C}_{\mathsf{mc}}, C, x_{\mathcal{B}}, x_C, K\{x_{\mathcal{B}}, x_C\}, y_{\mathcal{B}} \oplus Y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}\oplus\langle r_{\mathcal{B}},s\rangle}, y_C]$ clearly outputs the hardwired value $y_{\mathcal{B}} \oplus Y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}\oplus\langle r_{\mathcal{B}},s\rangle}$ according to the second item of Figure 7. We show that $Q_1[\mathsf{C}_{\mathsf{mc}}, C, \mu_{\mathcal{B}}, h, F_{\mathsf{lf}}, \mathsf{str}_{\mathcal{B}}, \eta_{\mathcal{B}}, \eta_C, K, y_{\mathcal{B}}, y_C, \mathsf{coin}_{\mathcal{B}}]$ also outputs $y_{\mathcal{B}} \oplus Y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}\oplus\langle r_{\mathcal{B}},s\rangle}$ in this case. $x = x_{\mathcal{B}}$ implies $F_{\mathsf{lf}}(x) = \eta_{\mathcal{B}}$, and thus the computation goes to the second item of Figure 9. Since $x = x_{\mathcal{B}}$, we have $\mathsf{str}_{\mathcal{B}} \oplus h(x) = A \| r_{\mathcal{B}}$. Since $a \in A + s$, we also have $\mathsf{Can}_A(a) = s$. Thus, we can conclude that it outputs $y_{\mathcal{B}} \oplus Y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}\oplus\langle r_{\mathcal{B}},s\rangle}$, where we recall that $Y_{\mathcal{B},0} = C(x_{\mathcal{B}})$ and $Y_{\mathcal{B},1} = \mu_{\mathcal{B}}(x_{\mathcal{B}})$.
- The case where $C_{\mathsf{mc}}(a) = 1$ (i.e., $a \in A + s$), and $x = x_C$. Since F_{lf} is injective and we assume $x_{\mathcal{B}} \neq x_C$, the latter condition implies $F_{\mathsf{lf}}(x) \neq \eta_{\mathcal{B}}$ and $F_{\mathsf{lf}}(x) = \eta_C$. Then, we can see that both circuits output y_C in this case according to the third items of Figures 7 and 9, respectively.
- The case where $C_{\mathsf{mc}}(x) = 1$ (i.e., $a \in A + s$), and $x \notin \{x_{\mathcal{B}}, x_C\}$. Since F_{lf} is injective, the latter condition implies $F_{\mathsf{lf}}(x) \notin \{\eta_{\mathcal{B}}, \eta_C\}$. Then, we can see that both circuits output $\mathsf{F}_K(x) \oplus C(x)$ according to the fourth items of Figures 7 and 9, respectively.

$Q_1[\mathsf{C}_{\mathsf{mc}}, C, \mu_{\mathcal{B}}, h, F_{\mathsf{lf}}, \mathsf{str}_{\mathcal{B}}, \eta_{\mathcal{B}}, \eta_{\mathcal{C}}, K, y_{\mathcal{B}}, y_{\mathcal{C}}, \mathsf{coin}_{\mathcal{B}}](a, x)$

Hardwired constant: Descriptions of boolean circuits $(\mathsf{C}_{\mathsf{mc}}, C, \mu_{\mathcal{B}})$, descriptions of functions (h, F_{lf}), strings $(\mathsf{str}_{\mathcal{B}}, \eta_{\mathcal{B}}, \eta_{\mathcal{C}}, K, y_{\mathcal{B}}, y_{\mathcal{C}})$, and a bit $\mathsf{coin}_{\mathcal{B}}$.

Input: (a, x).

1. If $\mathsf{C}_{\mathsf{mc}}(a) = 0$, output $\bot$. Otherwise, go to the next step.
2. If $F_{\mathsf{lf}}(x) = \eta_{\mathcal{B}}$, do the following.
   - Compute $S \| r \leftarrow \mathsf{str}_{\mathcal{B}} \oplus h(x)$.
   - Output $y_{\mathcal{B}} \oplus C(x)$ if $\mathsf{coin}_{\mathcal{B}} \oplus \langle r, \mathsf{Can}_S(a) \rangle = 0$ and $y_{\mathcal{B}} \oplus \mu_{\mathcal{B}}(x)$ if $\mathsf{coin}_{\mathcal{B}} \oplus \langle r, \mathsf{Can}_S(a) \rangle = 1$.

   Otherwise, go to the next step.
3. If $F_{\mathsf{lf}}(x) = \eta_{\mathcal{C}}$, output the hardwired value $y_{\mathcal{C}}$. Otherwise, go to the next step.
4. Output $\mathsf{F}_K(x) \oplus C(x)$.

Figure 9: The description of the circuit Q_1.

$Q_2[\mathsf{C}^{\bot}_{\mathsf{mc}}, C, \mu_{\mathcal{C}}, h, F_{\mathsf{lf}}, \mathsf{str}_{\mathcal{C}}, \eta_{\mathcal{B}}, \eta_{\mathcal{C}}, K, y_{\mathcal{B}}, y_{\mathcal{C}}, \mathsf{coin}_{\mathcal{C}}](a, x)$

Hardwired constant: Descriptions of boolean circuits $(\mathsf{C}^{\bot}_{\mathsf{mc}}, C, \mu_{\mathcal{C}})$, descriptions of functions (h, F_{lf}), strings $(\mathsf{str}_{\mathcal{C}}, \eta_{\mathcal{B}}, \eta_{\mathcal{C}}, K, y_{\mathcal{B}}, y_{\mathcal{C}})$, and a bit $\mathsf{coin}_{\mathcal{C}}$.

Input: (a, x).

1. If $\mathsf{C}^{\bot}_{\mathsf{mc}}(a) = 0$, output $\bot$. Otherwise, go to the next step.
2. If $F_{\mathsf{lf}}(x) = \eta_{\mathcal{B}}$, output the hardwired value $y_{\mathcal{B}}$. Otherwise, go to the next step.
3. If $F_{\mathsf{lf}}(x) = \eta_{\mathcal{C}}$, do the following.
   - Compute $S \| r \leftarrow \mathsf{str}_{\mathcal{B}} \oplus h(x)$.
   - Output $y_{\mathcal{C}} \oplus C(x)$ if $\mathsf{coin}_{\mathcal{C}} \oplus \langle r, \mathsf{Can}_S(a) \rangle = 0$ and $y_{\mathcal{C}} \oplus \mu_{\mathcal{C}}(x)$ if $\mathsf{coin}_{\mathcal{C}} \oplus \langle r, \mathsf{Can}_S(a) \rangle = 1$.

   Otherwise, go to the next step.
4. Output $\mathsf{F}_K(x)$.

Figure 10: The description of the circuit Q_2.

This shows that the two circuits are functionally equivalent.

We can similarly prove that $P'_2[\mathsf{C}^{\bot}_{\mathsf{mc}}, x_{\mathcal{B}}, x_{\mathcal{C}}, K\{x_{\mathcal{B}}, x_{\mathcal{C}}\}, y_{\mathcal{B}}, y_{\mathcal{C}} \oplus Y_{\mathcal{C}, \mathsf{coin}_{\mathcal{C}} \oplus \langle r_{\mathcal{C}}, t \rangle}]$ and $Q_2[\mathsf{C}^{\bot}_{\mathsf{mc}}, C, \mu_{\mathcal{C}}, h, F_{\mathsf{lf}}, \mathsf{str}_{\mathcal{C}}, \eta_{\mathcal{B}}, \eta_{\mathcal{C}}, K, y_{\mathcal{B}}, y_{\mathcal{C}}, \mathsf{coin}_{\mathcal{C}}]$ are functionally equivalent. Then, from the above discussion and the security of $i\mathcal{O}$, we have

$$|\Pr[\mathsf{Hyb}_4 = 1] - \Pr[\mathsf{Hyb}_5 = 1]| = \mathsf{negl}(\lambda).$$

Hyb_6**:** This is identical to Hyb_5 except F_{lf} is generated in lossy mode, that is, $F_{\mathsf{lf}} \leftarrow \mathsf{Gen}_{\mathsf{loss}}(1^\lambda)$. From the mode indistinguishability of LF, we have

$$|\Pr[\mathsf{Hyb}_5 = 1] - \Pr[\mathsf{Hyb}_6 = 1]| = \mathsf{negl}(\lambda).$$

We below show that

$$\Pr[\mathsf{Hyb}_6 = 1] \leq \frac{1}{2} + \mathsf{negl}(\lambda)$$

by a reduction to the computational strong indistinguishability monogamy property of coset states with simulatable auxiliary input (Theorem 4.2) with respect to the following auxiliary input generator $\mathcal{Aux} = (\mathcal{AuxSetup}, \mathcal{AuxGen}_{\mathcal{B}}, \mathcal{AuxGen}_{\mathcal{C}})$ and the corresponding simulator $\mathcal{Sim} = (\mathcal{Sim}_{\mathcal{B}}, \mathcal{Sim}_{\mathcal{C}})$.

$\mathcal{AuxSetup}(1^\lambda)$**:**

- Generate $h \leftarrow \mathcal{H}$ and $F_{\mathsf{lf}} \leftarrow \mathsf{Gen}_{\mathsf{loss}}(1^\lambda)$.
- Output $\mathsf{pp} = (h, F_{\mathsf{lf}})$.

$\mathcal{AuxGen}_{\mathcal{X}}(\mathsf{pp}, A, r_{\mathcal{X}})$ where $\mathcal{X} \in \{\mathcal{B}, \mathcal{C}\}$**:**

- Parse $\mathsf{pp} = (h, F_{\mathsf{lf}})$.
- Sample $x_{\mathcal{X}} \leftarrow \mathcal{D}_{\mathcal{X}}(1^\lambda)$ and compute $\eta_{\mathcal{X}} := F_{\mathsf{lf}}(x_{\mathcal{X}})$.
- Set
$$\mathsf{str}_{\mathcal{X}} := \begin{cases} (A \| r_{\mathcal{B}}) \oplus h(x_{\mathcal{B}}) & \text{if } \mathcal{X} = \mathcal{B} \\ (A^\perp \| r_{\mathcal{C}}) \oplus h(x_{\mathcal{C}}) & \text{if } \mathcal{X} = \mathcal{C} \end{cases}$$
- Output $z_{1,\mathcal{X}} := (\eta_{\mathcal{X}}, \mathsf{str}_{\mathcal{X}})$ and $z_{2,\mathcal{X}} := x_{\mathcal{X}}$.

$\mathcal{S}im_{\mathcal{X}}(1^\lambda)$ where $\mathcal{X} \in \{\mathcal{B}, \mathcal{C}\}$**:**

- Generate $\eta_{\mathcal{X}}$ in the same way as in $\mathcal{AuxGen}_{\mathcal{X}}$.
- Sample $\mathsf{str}_{\mathcal{X}} \leftarrow \{0,1\}^{\ell_A + n}$.
- Output $z_{1,\mathcal{X}} := (\eta_{\mathcal{X}}, \mathsf{str}_{\mathcal{X}})$.

First, we show that they satisfy the simulatability condition of Theorem 4.2.

Lemma 5.3. *For $\mathcal{X} \in \{\mathcal{B}, \mathcal{C}\}$, any subspace $A \subseteq \mathbb{F}_2^n$, and any $r_{\mathcal{X}} \in \{0,1\}^n$,*

$$\left\{ (\mathsf{pp}, z_{1,\mathcal{X}}) : \begin{array}{l} \mathsf{pp} \leftarrow \mathcal{AuxSetup}(1^\lambda) \\ (z_{1,\mathcal{X}}, z_{2,\mathcal{X}}) \leftarrow \mathcal{AuxGen}_{\mathcal{X}}(\mathsf{pp}, A, r_{\mathcal{X}}) \end{array} \right\} \approx \left\{ (\mathsf{pp}, z_{1,\mathcal{X}}) : \begin{array}{l} \mathsf{pp} \leftarrow \mathcal{AuxSetup}(1^\lambda) \\ z_{1,\mathcal{X}} \leftarrow \mathcal{S}im_{\mathcal{X}}(\mathsf{pp}) \end{array} \right\}.$$

Proof of Lemma 5.3. In the execution of $\mathcal{AuxGen}_{\mathcal{X}}(\mathsf{pp}, A, r_{\mathcal{X}})$, note that $\eta_{\mathcal{X}} = F_{\mathsf{lf}}(x_{\mathcal{X}})$ is the only variables that depend on $x_{\mathcal{X}}$. By the assumption, $x_{\mathcal{X}}$ has min-entropy at least $\ell_A + n + (\ell_{\mathsf{inp}} - \ell_{\mathsf{loss}}) + \omega(\log\lambda)$. Since the image size of F_{lf} is at most $2^{\ell_{\mathsf{inp}} - \ell_{\mathsf{loss}}}$, by the chain rule, $x_{\mathcal{X}}$ has conditional min-entropy at least $\ell_A + n + \omega(\log\lambda)$ given $\eta_{\mathcal{X}}$. Thus, by the leftover hash lemma (Lemma 2.2), $(h, F_{\mathsf{lf}}, \eta_{\mathcal{X}}, h(x_{\mathcal{X}})) \approx (h, F_{\mathsf{lf}}, \eta_{\mathcal{X}}, u'_{\mathcal{X}})$ where $u'_{\mathcal{X}} \leftarrow \{0,1\}^{\ell_k}$. This immediately implies Lemma 5.3. □

We now construct the following adversary $\mathcal{A}_{\mathsf{moe}} = (\widehat{\mathcal{A}}, \widehat{\mathcal{B}}, \widehat{\mathcal{C}})$ for the experiment in Theorem 4.2 using $\mathcal{A}_{\mathsf{cp}} = (\mathcal{A}, \mathcal{B}, \mathcal{C})$.

1. $\widehat{\mathcal{A}}$ is given $|A_{s,t}\rangle$, $\mathsf{C}_{\mathsf{mc}} = i\mathcal{O}(A + s)$, $\mathsf{C}^\perp_{\mathsf{mc}} = i\mathcal{O}(A^\perp + t)$, $\mathsf{pp} = (h, F_{\mathsf{lf}})$, $z_{1,\mathcal{B}} = (\eta_{\mathcal{B}}, \mathsf{str}_{\mathcal{B}})$, and $z_{1,\mathcal{C}} = (\eta_{\mathcal{C}}, \mathsf{str}_{\mathcal{C}})$. $\widehat{\mathcal{A}}$ invokes $\mathcal{A}$ with 1^λ and obtains the circuits $(C, \mu_{\mathcal{B}}, \mu_{\mathcal{C}})$. $\widehat{\mathcal{A}}$ generates $\rho_C := (|A_{s,t}\rangle, \widetilde{P}, \widetilde{P}^\perp)$ as in Hyb_6. $\widehat{\mathcal{A}}$ can generate it with the given inputs, circuits $(C, \mu_{\mathcal{B}}, \mu_{\mathcal{C}})$ declared by $\mathcal{A}$, and $(K, \mathsf{coin}_{\mathcal{B}}, y_{\mathcal{B}}, \mathsf{coin}_{\mathcal{C}}, y_{\mathcal{C}})$ sampled by $\widehat{\mathcal{A}}$ itself. $\widehat{\mathcal{A}}$ then sends ρ_C to $\mathcal{A}$.
2. When $\mathcal{A}$ outputs a bipartite state q over registers $\mathsf{R}_{\mathcal{B}}$ and $\mathsf{R}_{\mathcal{C}}$, $\widehat{\mathcal{A}}$ sends $(\mathsf{coin}_{\mathcal{B}}, q[\mathsf{R}_{\mathcal{B}}])$ to $\mathcal{B}$ and $(\mathsf{coin}_{\mathcal{C}}, q[\mathsf{R}_{\mathcal{C}}])$ to $\mathcal{C}$.
3. $\widehat{\mathcal{B}}$ and $\widehat{\mathcal{C}}$ are respectively given $(A, r_{\mathcal{B}}, z_{2,\mathcal{B}} = x_{\mathcal{B}})$ and $(A, r_{\mathcal{C}}, z_{2,\mathcal{C}} = x_{\mathcal{C}})$ and behave as follows.
   - $\widehat{\mathcal{B}}$ invokes $\mathcal{B}$ with $q[\mathsf{R}_{\mathcal{B}}]$ and $x_{\mathcal{B}}$, and obtains $\mathsf{coin}'_{\mathcal{B}}$. $\widehat{\mathcal{B}}$ outputs $\mathsf{coin}'_{\mathcal{B}} \oplus \mathsf{coin}_{\mathcal{B}}$.
   - $\widehat{\mathcal{C}}$ invokes $\mathcal{C}$ with $q[\mathsf{R}_{\mathcal{C}}]$ and $x_{\mathcal{C}}$, and obtains $\mathsf{coin}'_{\mathcal{C}}$. $\widehat{\mathcal{C}}$ outputs $\mathsf{coin}'_{\mathcal{C}} \oplus \mathsf{coin}_{\mathcal{C}}$.

$\mathcal{A}_{\mathsf{moe}}$ perfectly simulates Hyb_6 for $\mathcal{A}_{\mathsf{cp}}$. If $\mathcal{A}_{\mathsf{cp}}$ wins the simulated experiment (i.e., Hyb_6 outputs 1), we have $\mathsf{coin}'_{\mathcal{B}} \oplus \mathsf{coin}'_{\mathcal{C}} = \mathsf{coin}_{\mathcal{B}} \oplus \mathsf{coin}_{\mathcal{C}} \oplus \langle r_{\mathcal{B}}, s\rangle \oplus \langle r_{\mathcal{C}}, t\rangle$, that is, $(\mathsf{coin}'_{\mathcal{B}} \oplus \mathsf{coin}_{\mathcal{B}}) \oplus (\mathsf{coin}'_{\mathcal{C}} \oplus \mathsf{coin}_{\mathcal{C}}) = \langle r_{\mathcal{B}}, s\rangle \oplus \langle r_{\mathcal{C}}, t\rangle$. Thus, by Theorem 4.2 and Lemma 5.3, we have

$$\Pr[\mathsf{Hyb}_6 = 1] = \Pr\left[\mathsf{Exp}^{\mathsf{comp\text{-}strong\text{-}ind\text{-}moe\text{-}aux}}_{i\mathcal{O}, \mathcal{AUX}, \mathcal{A}_{\mathsf{moe}}}(\lambda) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

Combining the above, we have

$$\Pr\left[\mathsf{Expt}^{\mathsf{upoplus}}_{\mathsf{UPO}, \mathcal{D}, \mathcal{A}_{\mathsf{cp}}}(\lambda) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

This completes the proof of Theorem 5.2. □

6 Alternative Proof of Security for Uniform Inputs

Here, we give an alternative security proof for the special case of the uniform input distribution. An advantage of this alternative proof is that it does not rely on the additional assumption of lossy functions.

In the security proof, instead of universal hash functions and lossy functions, we rely on a new primitive which we call key-robust non-committing encryption.

Definition 6.1 (Key-Robust Non-committing Encryption). *A key-robust non-committing encryption scheme* NCE *with the message space* $\mathcal{M}$ *is a tuple of five PPT algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake}, \mathsf{Open})$.

$\mathsf{Gen}(1^\lambda)$**:** *The key generation algorithm takes as input a security parameter* 1^λ *and outputs a key* k.

$\mathsf{Enc}(k, m) \to \mathsf{ct}$ *The encryption algorithm takes as input a key* k *and a message* $m \in \mathcal{M}$*, and outputs a ciphertext* ct.

$\mathsf{Dec}(k, \mathsf{ct}) \to m'$ *The decryption algorithm is a deterministic algorithm that takes as input a key* k *and a ciphertext* ct*, and outputs* $m' \in \mathcal{M} \cup \{\bot\}$.

$\mathsf{Fake}(1^\lambda) \to (\mathsf{ct}^*, \mathsf{st})$ *The fake algorithm takes as input a security parameter* 1^λ*, and outputs a fake ciphertext* ct^* *and a state information* st.

$\mathsf{Open}(\mathsf{st}, m) \to k^*$ *The open algorithm takes as input a state information* st *and a message* $m \in \mathcal{M}$*, and outputs a key* k^*.

We require it to satisfy the following three properties.

Correctness: *With overwhelming probability over the choice of* $k \leftarrow \mathsf{Gen}(1^\lambda)$*, for any* $m \in \mathcal{M}$*,*

$$\Pr[\mathsf{Dec}(k, \mathsf{Enc}(k, m)) = m] = 1.$$

Key-Robustness: *With overwhelming probability over the choice of* $k \leftarrow \mathsf{Gen}(1^\lambda)$*, for any* $m \in \mathcal{M}$ *and any* $k' \neq k$*,*

$$\Pr\big[\mathsf{Dec}(k', \mathsf{Enc}(k, m)) = \bot\big] = 1.$$

Non-committing: *For any message* $m \in \mathcal{M}$*, we have*

$$(\mathsf{ct}, k) \overset{\mathsf{c}}{\approx} (\mathsf{ct}^*, k^*),$$

where $k \leftarrow \mathsf{Gen}(1^\lambda)$, $\mathsf{ct} \leftarrow \mathsf{Enc}(k, m)$, $(\mathsf{ct}^*, \mathsf{st}) \leftarrow \mathsf{Fake}(1^\lambda)$*, and* $k^* \leftarrow \mathsf{Open}(\mathsf{st}, \mathsf{ct}^*)$.

We say that a key k *is good if both* $\Pr[\mathsf{Dec}(k, \mathsf{Enc}(k, m)) = m] = 1$ *and* $\Pr[\mathsf{Dec}(k', \mathsf{Enc}(k, m)) = \bot] = 1$ *hold for all* m *and* $k' \neq k$*. By the union bound, a key generated by* $\mathsf{Gen}(1^\lambda)$ *is good with an overwhelming probability.*

Theorem 6.2. *Assuming the existence of keyed injective OWFs, for any polynomial* ℓ_m*, there exists a key-robust non-committing encryption scheme with the message space* $\{0,1\}^{\ell_m}$*, where the key length is linear in* ℓ_m.

See Appendix A for the proof.

Note that keyed injective one-way functions exist assuming the existence of iO and one-way functions [BPW16], thus a key-robust non-committing encryption scheme exists assuming he existence of iO and one-way functions.

We rely on the following primitive, where ℓ_k is a polynomial in λ and $\ell_A := n^2/2$ is the number of bits needed to represent a subspace $A \subseteq \mathbb{F}_2^n$ of dimension $n/2$.

- A key-robust non-committing encryption scheme $\mathsf{NCE} = (\mathsf{NCE.Enc}, \mathsf{NCE.Dec}, \mathsf{NCE.Fake}, \mathsf{NCE.Open})$ with the message space $\{0,1\}^{\ell_A + n}$ and the key space $\{0,1\}^{\ell_k}$.

We prove the following theorem.

Theorem 6.3. *Let* $\mathcal{U}$ *be the uniformly random distribution over* $\{0,1\}^{\ell_{\mathsf{inp}}}$ *where* $\ell_{\mathsf{inp}} \geq \ell_k$*. Then,* UPO *satisfies* $\mathcal{U}$*-generalized UPO*$^+$ *security.*

For any polynomial ℓ_{inp}, we may choose sufficiently small polynomial ℓ_k such that $\ell_{\mathsf{inp}} \geq \ell_k$. Also, we may choose ℓ_{out} to be any polynomial. Moreover, $\mathcal{U}$-generalized UPO$^+$security implies $\mathcal{U}$-generalized UPO security without any additional assumption by Theorem 3.17 since our construction clearly satisfies iO security and there exist keyed injective $\mathcal{U}$-OWFs assuming the existence of iO and OWFs. Therefore, the latter part of Theorem 5.1 follows.

Proof of Theorem 6.3. For simplicity, we give the proof assuming $\ell_{\mathsf{inp}} = \ell_k$ while the extension to the general case is straightforward.

Let $\mathcal{A}_{\mathsf{cp}} = (\mathcal{A}, \mathcal{B}, C)$ be a QPT adversary against $\mathcal{D}$-generalized UPO$^+$security of UPO. We consider the following sequence of hybrids.

Hyb_0-Hyb_4**:** They are identical to those in the security proof in Section 5.2 except that $x_{\mathcal{B}}$ and x_C are chosen uniformly randomly instead of from the distributions $\mathcal{D}_{\mathcal{B}}$ and $\mathcal{D}_C$, respectively.[20]

We below use the circuits Q'_1 and Q'_2 respectively described in Figure 11 and Figure 12.

Hyb_5**:** This is identical to Hyb_4 except for the following changes,where $k_{\mathcal{B}} \leftarrow \mathsf{NCE.Gen}(1^\lambda)$, $k_C \leftarrow \mathsf{NCE.Gen}(1^\lambda)$, $\mathsf{ct}_{\mathcal{B}} \leftarrow \mathsf{NCE.Enc}(k_{\mathcal{B}}, A\|r_{\mathcal{B}})$, and $\mathsf{ct}_C \leftarrow \mathsf{NCE.Enc}(k_C, A\|r_C)$.

- $\widetilde{P}_1$ is generated as $\widetilde{P}_1 \leftarrow i\mathcal{O}(Q'_1[\mathsf{C_{mc}}, C, \mu_{\mathcal{B}}, \mathsf{str}_{\mathcal{B}}, \mathsf{str}_C, \mathsf{ct}_{\mathcal{B}}, \mathsf{ct}_C, K, y_{\mathcal{B}}, y_C, \mathsf{coin}_{\mathcal{B}}])$, where $\mathsf{str}_{\mathcal{B}} = k_{\mathcal{B}} \oplus x_{\mathcal{B}}$.
- $\widetilde{P}_2$ is generated as $\widetilde{P}_2 \leftarrow i\mathcal{O}(Q'_2[C^{\perp}_{\mathsf{mc}}, C, \mu_C, \mathsf{str}_{\mathcal{B}}, \mathsf{str}_C, \mathsf{ct}_{\mathcal{B}}, \mathsf{ct}_C, K, y_{\mathcal{B}}, y_C, \mathsf{coin}_C])$, where $\mathsf{str}_C = k_C \oplus x_C$.

Below, we assume that both $k_{\mathcal{B}}$ and k_C are good as per Definition 6.1, i.e., the perfect correctness and key-robustness hold under those keys, since this holds with overwhelming probability.

We prove that

$$P'_1[\mathsf{C_{mc}}, C, x_{\mathcal{B}}, x_C, K\{x_{\mathcal{B}}, x_C\}, y_{\mathcal{B}} \oplus Y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}\oplus\langle r_{\mathcal{B}},s\rangle}, y_C]$$

and

$$Q'_1[\mathsf{C_{mc}}, C, \mu_{\mathcal{B}}, \mathsf{str}_{\mathcal{B}}, \mathsf{ct}_{\mathcal{B}}, \mathsf{ct}_C, K, y_{\mathcal{B}}, y_C, \mathsf{coin}_{\mathcal{B}}]$$

are functionally equivalent. For an input (a, x), we consider the following cases.

- The case where $C_{\mathsf{mc}}(a) = 0$ (i.e., $a \notin A + s$). In this case, both circuits output $\perp$ according to the first items of Figures 7 and 11, respectively.
- The case where $C_{\mathsf{mc}}(a) = 1$ (i.e., $a \in A + s$), and $x = x_{\mathcal{B}}$. In this case, $P'_1[\mathsf{C_{mc}}, C, x_{\mathcal{B}}, x_C, K\{x_{\mathcal{B}}, x_C\}, y_{\mathcal{B}} \oplus Y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}\oplus\langle r_{\mathcal{B}},s\rangle}, y_C]$ clearly outputs the hardwired value $y_{\mathcal{B}} \oplus Y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}\oplus\langle r_{\mathcal{B}},s\rangle}$ according to the second item of Figure 7. We show that $Q'_1[\mathsf{C_{mc}}, C, \mu_{\mathcal{B}}, \mathsf{str}_{\mathcal{B}}, \mathsf{ct}_{\mathcal{B}}, \mathsf{ct}_C, K, y_{\mathcal{B}}, y_C, \mathsf{coin}_{\mathcal{B}}]$ also outputs $y_{\mathcal{B}} \oplus Y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}\oplus\langle r_{\mathcal{B}},s\rangle}$ in this case. $x = x_{\mathcal{B}}$ implies $k'_{\mathcal{B}} = \mathsf{str}_{\mathcal{B}} \oplus x = k_{\mathcal{B}}$, and thus $\mathsf{NCE.Dec}(k'_{\mathcal{B}}, \mathsf{ct}_{\mathcal{B}}) = A\|r_{\mathcal{B}} \neq \perp$ by the correctness of NCE. Therefore, the computation goes to the second item of Figure 11. Since $a \in A + s$, we also have $\mathsf{Can}_A(a) = s$. Thus, we can conclude that it outputs $y_{\mathcal{B}} \oplus Y_{\mathcal{B},\mathsf{coin}_{\mathcal{B}}\oplus\langle r_{\mathcal{B}},s\rangle}$, where we recall that $Y_{\mathcal{B},0} = C(x_{\mathcal{B}})$ and $Y_{\mathcal{B},1} = \mu_{\mathcal{B}}(x_{\mathcal{B}})$.
- The case where $C_{\mathsf{mc}}(a) = 1$ (i.e., $a \in A + s$), and $x = x_C$. Since we assume $x_{\mathcal{B}} \neq x_C$, we have $k'_{\mathcal{B}} = \mathsf{str}_{\mathcal{B}} \oplus x \neq k_{\mathcal{B}}$ and $k'_C = \mathsf{str}_C \oplus x = k_C$. Thus, by the key-robustness and correctness of NCE, $\mathsf{NCE.Dec}(k'_{\mathcal{B}}, \mathsf{ct}_{\mathcal{B}}) = \perp$ and $\mathsf{NCE.Dec}(k'_C, \mathsf{ct}_C) = A\|r_C \neq \perp$. Then, we can see that both circuits output y_C in this case according to the third items of Figures 7 and 11, respectively.
- The case where $C_{\mathsf{mc}}(x) = 1$ (i.e., $a \in A + s$), and $x \notin \{x_{\mathcal{B}}, x_C\}$. Then we have $k'_{\mathcal{B}} = \mathsf{str}_{\mathcal{B}} \oplus x \neq k_{\mathcal{B}}$ and $k'_C = \mathsf{str}_C \oplus x \neq k_C$. By the key-robustness of NCE, $\mathsf{NCE.Dec}(k'_{\mathcal{B}}, \mathsf{ct}_{\mathcal{B}}) = \perp$ and $\mathsf{NCE.Dec}(k'_C, \mathsf{ct}_C) = \perp$. Then, we can see that both circuits output $\mathsf{F}_K(x) \oplus C(x)$ according to the fourth items of Figures 7 and 11, respectively.

[20] Note that the universal hash functions and lossy functions are not used at or before Hyb_4.

$Q_1'[\mathsf{C_{mc}}, C, \mu_{\mathcal{B}}, \mathsf{str}_{\mathcal{B}}, \mathsf{str}_{\mathcal{C}}, \mathsf{ct}_{\mathcal{B}}, \mathsf{ct}_{\mathcal{C}}, K, y_{\mathcal{B}}, y_{\mathcal{C}}, \mathsf{coin}_{\mathcal{B}}](a, x)$

Hardwired constant: Descriptions of boolean circuits $(\mathsf{C_{mc}}, C, \mu_{\mathcal{B}})$, strings $(\mathsf{str}_{\mathcal{B}}, \mathsf{str}_{\mathcal{C}}, \mathsf{ct}_{\mathcal{B}}, \mathsf{ct}_{\mathcal{C}}, K, y_{\mathcal{B}}, y_{\mathcal{C}})$, and a bit $\mathsf{coin}_{\mathcal{B}}$.
Input: (a, x).

1. If $\mathsf{C_{mc}}(a) = 0$, output $\bot$. Otherwise, go to the next step.
2. If $\mathsf{NCE.Dec}(k'_{\mathcal{B}}, \mathsf{ct}_{\mathcal{B}}) \neq \bot$ where $k'_{\mathcal{B}} \leftarrow \mathsf{str}_{\mathcal{B}} \oplus x$, do the following.
   - Compute $S \| r \leftarrow \mathsf{NCE.Dec}(k'_{\mathcal{B}}, \mathsf{ct}_{\mathcal{B}})$.
   - Output $y_{\mathcal{B}} \oplus C(x)$ if $\mathsf{coin}_{\mathcal{B}} \oplus \langle r, \mathsf{Can}_S(a) \rangle = 0$ and $y_{\mathcal{B}} \oplus \mu_{\mathcal{B}}(x)$ if $\mathsf{coin}_{\mathcal{B}} \oplus \langle r, \mathsf{Can}_S(a) \rangle = 1$.

   Otherwise, go to the next step.
3. If $\mathsf{NCE.Dec}(k'_{\mathcal{C}}, \mathsf{ct}_{\mathcal{C}}) \neq \bot$ where $k'_{\mathcal{C}} \leftarrow \mathsf{str}_{\mathcal{C}} \oplus x$, output the hardwired value $y_{\mathcal{C}}$. Otherwise, go to the next step.
4. Output $\mathsf{F}_K(x) \oplus C(x)$.

Figure 11: The description of the circuit Q_1'.

$Q_2'[\mathsf{C}^{\bot}_{\mathsf{mc}}, C, \mu_{\mathcal{C}}, \mathsf{str}_{\mathcal{B}}, \mathsf{str}_{\mathcal{C}}, \mathsf{ct}_{\mathcal{B}}, \mathsf{ct}_{\mathcal{C}}, K, y_{\mathcal{B}}, y_{\mathcal{C}}, \mathsf{coin}_{\mathcal{C}}](a, x)$

Hardwired constant: Descriptions of boolean circuits $(\mathsf{C}^{\bot}_{\mathsf{mc}}, C, \mu_{\mathcal{C}})$, strings $(\mathsf{str}_{\mathcal{B}}, \mathsf{str}_{\mathcal{C}}, \mathsf{ct}_{\mathcal{B}}, \mathsf{ct}_{\mathcal{C}}, K, y_{\mathcal{B}}, y_{\mathcal{C}})$, and a bit $\mathsf{coin}_{\mathcal{C}}$.
Input: (a, x).

1. If $\mathsf{C}^{\bot}_{\mathsf{mc}}(a) = 0$, output $\bot$. Otherwise, go to the next step.
2. If $\mathsf{NCE.Dec}(k'_{\mathcal{B}}, \mathsf{ct}_{\mathcal{B}}) \neq \bot$ where $k'_{\mathcal{B}} \leftarrow \mathsf{str}_{\mathcal{B}} \oplus x$, output the hardwired value $y_{\mathcal{B}}$. Otherwise, go to the next step.
3. If $\mathsf{NCE.Dec}(k'_{\mathcal{C}}, \mathsf{ct}_{\mathcal{C}}) \neq \bot$ where $k'_{\mathcal{C}} \leftarrow \mathsf{str}_{\mathcal{C}} \oplus x$, do the following.
   - Compute $S \| r \leftarrow \mathsf{NCE.Dec}(k'_{\mathcal{C}}, \mathsf{ct}_{\mathcal{C}})$.
   - Output $y_{\mathcal{C}} \oplus C(x)$ if $\mathsf{coin}_{\mathcal{C}} \oplus \langle r, \mathsf{Can}_S(a) \rangle = 0$ and $y_{\mathcal{C}} \oplus \mu_{\mathcal{C}}(x)$ if $\mathsf{coin}_{\mathcal{C}} \oplus \langle r, \mathsf{Can}_S(a) \rangle = 1$.

   Otherwise, go to the next step.
4. Output $\mathsf{F}_K(x)$.

Figure 12: The description of the circuit Q_2'.

This shows that the two circuits are functionally equivalent.

We can similarly prove that $P_2'[\mathsf{C}^{\bot}_{\mathsf{mc}}, x_{\mathcal{B}}, x_{\mathcal{C}}, K\{x_{\mathcal{B}}, x_{\mathcal{C}}\}, y_{\mathcal{B}}, y_{\mathcal{C}} \oplus Y_{\mathcal{C}, \mathsf{coin}_{\mathcal{C}} \oplus \langle r_{\mathcal{C}}, t \rangle}]$ and $Q_2'[\mathsf{C}^{\bot}_{\mathsf{mc}}, C, \mu_{\mathcal{C}}, \mathsf{str}_{\mathcal{B}}, \mathsf{str}_{\mathcal{C}}, \mathsf{ct}_{\mathcal{B}}, \mathsf{ct}_{\mathcal{C}}, K, y_{\mathcal{B}}, y_{\mathcal{C}}, \mathsf{coin}_{\mathcal{C}}]$ are functionally equivalent. Then, from the above discussion and the security of $i\mathcal{O}$, we have

$$|\Pr[\mathsf{Hyb}_4 = 1] - \Pr[\mathsf{Hyb}_5 = 1]| = \mathsf{negl}(\lambda).$$

Hyb_6**:** This is identical to Hyb_5 except for the following changes.

- $(\mathsf{ct}_{\mathcal{B}}, k_{\mathcal{B}})$ are generated as $(\mathsf{ct}_{\mathcal{B}}, \mathsf{st}_{\mathcal{B}}) \leftarrow \mathsf{NCE.Fake}(1^\lambda)$ and $k_{\mathcal{B}} \leftarrow \mathsf{NCE.Open}(\mathsf{st}_{\mathcal{B}}, A \| r_{\mathcal{B}})$.
- $(\mathsf{ct}_{\mathcal{C}}, k_{\mathcal{C}})$ are generated as $(\mathsf{ct}_{\mathcal{C}}, \mathsf{st}_{\mathcal{C}}) \leftarrow \mathsf{NCE.Fake}(1^\lambda)$ and $k_{\mathcal{C}} \leftarrow \mathsf{NCE.Open}(\mathsf{st}_{\mathcal{C}}, A^{\bot} \| r_{\mathcal{C}})$.

From the non-committing property of NCE, we have

$$|\Pr[\mathsf{Hyb}_5 = 1] - \Pr[\mathsf{Hyb}_6 = 1]| = \mathsf{negl}(\lambda).$$

Hyb_7**:** This is identical to Hyb_6 except that $\mathsf{str}_{\mathcal{B}}$ and $\mathsf{str}_{\mathcal{C}}$ are chosen as uniformly random strings and then we set $x_{\mathcal{B}} := \mathsf{str}_{\mathcal{B}} \oplus k_{\mathcal{B}}$ and $x_{\mathcal{C}} := \mathsf{str}_{\mathcal{C}} \oplus k_{\mathcal{C}}$.

Since the challenge inputs $x_{\mathcal{B}}, x_{\mathcal{C}}$ are uniformly random, the strings $\mathsf{str}_{\mathcal{B}} = k_{\mathcal{B}} \oplus x_{\mathcal{B}}, \mathsf{str}_{\mathcal{C}} = k_{\mathcal{C}} \oplus x_{\mathcal{C}}$ are also uniformly random. Thus, this hybrid is identical to the previous one from the view of the adversary, and thus,

$$\Pr[\mathsf{Hyb}_6 = 1] = \Pr[\mathsf{Hyb}_7 = 1]$$

We below show that

$$\Pr[\mathsf{Hyb}_7 = 1] \leq \frac{1}{2} + \mathsf{negl}(\lambda)$$

by a reduction to the computational strong indistinguishability monogamy property of coset states with no auxiliary information, i.e., the special case of Theorem 4.2 where $\mathcal{Aux} = (\mathcal{Aux}\mathit{Setup}, \mathcal{Aux}\mathit{Gen})$ outputs nothing, and thus the simulatability condition is trivially satisfied.[21]

We now construct the following adversary $\mathcal{A}_{\mathsf{moe}} = (\widehat{\mathcal{A}}, \widehat{\mathcal{B}}, \widehat{\mathcal{C}})$ for the experiment in Theorem 4.2 with no auxiliary input using $\mathcal{A}_{\mathsf{cp}} = (\mathcal{A}, \mathcal{B}, \mathcal{C})$.

1. $\widehat{\mathcal{A}}$ is given $|A_{s,t}\rangle$, and $\mathsf{C}_{\mathsf{mc}} = i\mathcal{O}(A + s)$, $\mathsf{C}^{\perp}_{\mathsf{mc}} = i\mathcal{O}(A^{\perp} + t)$. $\widehat{\mathcal{A}}$ invokes $\mathcal{A}$ with 1^{λ} and obtains the circuits $(C, \mu_{\mathcal{B}}, \mu_{\mathcal{C}})$. $\widehat{\mathcal{A}}$ generates $\rho_C := (|A_{s,t}\rangle, \widetilde{P}, \widetilde{P^{\perp}})$ as in Hyb_7. $\widehat{\mathcal{A}}$ can generate it with the given inputs, circuits $(C, \mu_{\mathcal{B}}, \mu_{\mathcal{C}})$ declared by $\mathcal{A}$, and $(K, \mathsf{coin}_{\mathcal{B}}, y_{\mathcal{B}}, \mathsf{str}_{\mathcal{B}}, \mathsf{ct}_{\mathcal{B}}, \mathsf{st}_{\mathcal{B}}, \mathsf{coin}_{\mathcal{C}}, y_{\mathcal{C}}, \mathsf{str}_{\mathcal{C}}, \mathsf{ct}_{\mathcal{C}}, \mathsf{st}_{\mathcal{C}})$ sampled by $\widehat{\mathcal{A}}$ itself. $\widehat{\mathcal{A}}$ then sends ρ_C to $\mathcal{A}$.
2. When $\mathcal{A}$ outputs a bipartite state q over registers $\mathsf{R}_{\mathcal{B}}$ and $\mathsf{R}_{\mathcal{C}}$, $\widehat{\mathcal{A}}$ sends $(\mathsf{coin}_{\mathcal{B}}, q[\mathsf{R}_{\mathcal{B}}])$ to $\mathcal{B}$ and $(\mathsf{coin}_{\mathcal{C}}, q[\mathsf{R}_{\mathcal{C}}])$ to $\mathcal{C}$.
3. $\widehat{\mathcal{B}}$ and $\widehat{\mathcal{C}}$ are respectively given $(A, r_{\mathcal{B}})$ and $(A, r_{\mathcal{C}})$ and behave as follows.
   - $\widehat{\mathcal{B}}$ computes $k_{\mathcal{B}} \leftarrow \mathsf{NCE.Open}(\mathsf{st}_{\mathcal{B}}, A \| r_{\mathcal{B}})$ and $x_{\mathcal{B}} := \mathsf{str}_{\mathcal{B}} \oplus k_{\mathcal{B}}$, invokes $\mathcal{B}$ with $q[\mathsf{R}_{\mathcal{B}}]$ and $x_{\mathcal{B}}$, and obtains $\mathsf{coin}'_{\mathcal{B}}$. $\widehat{\mathcal{B}}$ outputs $\mathsf{coin}'_{\mathcal{B}} \oplus \mathsf{coin}_{\mathcal{B}}$.
   - $\widehat{\mathcal{C}}$ computes $k_{\mathcal{C}} \leftarrow \mathsf{NCE.Open}(\mathsf{st}_{\mathcal{C}}, A^{\perp} \| r_{\mathcal{C}})$ and $x_{\mathcal{C}} := \mathsf{str}_{\mathcal{C}} \oplus k_{\mathcal{C}}$, invokes $\mathcal{C}$ with $q[\mathsf{R}_{\mathcal{C}}]$ and $x_{\mathcal{C}}$, and obtains $\mathsf{coin}'_{\mathcal{C}}$. $\widehat{\mathcal{C}}$ outputs $\mathsf{coin}'_{\mathcal{C}} \oplus \mathsf{coin}_{\mathcal{C}}$.

$\mathcal{A}_{\mathsf{moe}}$ perfectly simulates Hyb_7 for $\mathcal{A}_{\mathsf{cp}}$. If $\mathcal{A}_{\mathsf{cp}}$ wins the simulated experiment (i.e., Hyb_7 outputs 1), we have $\mathsf{coin}'_{\mathcal{B}} \oplus \mathsf{coin}'_{\mathcal{C}} = \mathsf{coin}_{\mathcal{B}} \oplus \mathsf{coin}_{\mathcal{C}} \oplus \langle r_{\mathcal{B}}, s\rangle \oplus \langle r_{\mathcal{C}}, t\rangle$, that is, $(\mathsf{coin}'_{\mathcal{B}} \oplus \mathsf{coin}_{\mathcal{B}}) \oplus (\mathsf{coin}'_{\mathcal{C}} \oplus \mathsf{coin}_{\mathcal{C}}) = \langle r_{\mathcal{B}}, s\rangle \oplus \langle r_{\mathcal{C}}, t\rangle$. Thus, by Theorem 4.2, we have

$$\Pr[\mathsf{Hyb}_7 = 1] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

Combining the above, we have

$$\Pr\left[\mathsf{Expt}^{\mathsf{upoplus}}_{\mathsf{UPO},\mathcal{U},\mathcal{A}_{\mathsf{cp}}}(\lambda) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

This completes the proof of Theorem 6.3. □

7 Oracular Pseudorandomness-Style Copy-Protection

In this section, we show oracular pseudorandomness-style copy-protection for all puncturable secure circuit classes.

7.1 Puncturable Secure Circuits

We first generalized the existing definitions of (average-case) puncturable secure circuits [AB24, CG24b], in terms of the number of points of puncture, and define puncturing security as follows:

Definition 7.1. *For any $\ell := \ell(\lambda)$ which is a polynomial function of λ, let $\mathsf{Circ} = \{\mathsf{C} : \boldsymbol{X} \to \boldsymbol{Y}\}$ be a circuit class, where $\boldsymbol{X} = \{0,1\}^n, \boldsymbol{Y} = \{0,1\}^m$ for polynomials $m := m(\lambda), n := n(\lambda)$, equipped with an efficient deterministic algorithm $\mathsf{Puncture}$ that satisfies puncturing correctness, i.e., $\mathsf{Puncture}(\mathsf{C}, x)$ outputs a circuit that outputs $\mathsf{C}(x)$ if and only if $x \neq x'$. We say that $(\mathsf{Circ}, \mathsf{Puncture})$ satisfies ℓ-point m-bit $(\mathcal{D}_{\boldsymbol{X}}, \mathcal{D}_{\mathsf{Circ}})$-unpredictability-style puncturing security (generalized from [AB24]) if for every QPT adversary $\mathcal{A}$, for every $\ell' \leq \ell$, the probability that $\mathcal{A}$ succeeds in the following security game is at most $1 - \left(1 - \frac{1}{2^m}\right)^{\ell'} + \mathsf{negl}$:*

[21] For the special case of Theorem 4.2 with no auxiliary information, the proof given in Section 4.1 is redundant. Simply applying the simultaneous quantum Goldreich-Levin lemma (Lemma 2.8) to the computational strong monogamy property of coset states [CLLZ21] suffices, as shown in [KY25b].

- $x_1 \leftarrow \mathcal{D}_{\boldsymbol{X}}, \ldots, x_{\ell'} \leftarrow \mathcal{D}_{\boldsymbol{X}}$,
- $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$,
- $\widehat{C} \leftarrow \mathsf{Puncture}(C, (x_1, \ldots, x_{\ell'}))$,
- $y'_1, \ldots, y'_{\ell'} \leftarrow \mathcal{A}(\widehat{C}, (x_1, \ldots, x_{\ell'}))$ *and,*
- $\mathcal{A}$ *wins if there exists* $i \in [\ell']$ *such that* $y'_i = C(x_i)$. [22]

Moreover we say that the circuit class satisfies ℓ-point m-bit $(\mathcal{D}_{\boldsymbol{X}}, \mathcal{D}_{\mathsf{Circ}})$-pseudorandomness-style puncturing security (generalized from the decision puncturing notion in [CG24b]) if for every $\ell' \leq \ell$,

$$\{\widehat{C}, (x_1, \ldots, x_{\ell'}), (C(x_1), \ldots, C(x_{\ell'}))\}_{\widehat{C} \leftarrow \mathsf{Puncture}(C, (x_1, \ldots, x_{\ell'})), x_1 \leftarrow \mathcal{D}_{\boldsymbol{X}}, \ldots, x_{\ell'} \leftarrow \mathcal{D}_{\boldsymbol{X}}, C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)}$$

$$\approx_c \{\widehat{C}, (x_1, \ldots, x_{\ell'}), (Y_1, \ldots, Y_{\ell'})\}_{\widehat{C} \leftarrow \mathsf{Puncture}(C, (x_1, \ldots, x_{\ell'})), x_1 \leftarrow \mathcal{D}_{\boldsymbol{X}}, \ldots, x_{\ell'} \leftarrow \mathcal{D}_{\boldsymbol{X}}, C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda), Y_1, \ldots, Y_{\ell'} \xleftarrow{\$} \boldsymbol{Y}}.$$

Next, we review the relationship between unpredictability-style and pseudorandomness-style puncturing security in various parameter regimes.

Remark 7.2. Clearly, if a circuit class satisfies ℓ-point m-bit $(\mathcal{D}_{\mathbf{X}}, \mathcal{D}_{\mathsf{Circ}})$-pseudorandomness-style puncturing security then it also satisfies 1-point m-bit $(\mathcal{D}_{\mathbf{X}}, \mathcal{D}_{\mathsf{Circ}})$-pseudorandomness-style puncturing security.[23]

Lemma 7.3. *For any $\ell := \ell(\lambda)$, $m := m(\lambda)$ which are polynomials in the security parameter λ, if a circuit class $(\mathsf{Circ}, \mathsf{Puncture})$ satisfies ℓ-point m-bit $(\mathcal{D}_{\boldsymbol{X}}, \mathcal{D}_{\mathsf{Circ}})$-pseudorandomness-style puncturing security then it also satisfies ℓ-point m-bit $(\mathcal{D}_{\boldsymbol{X}}, \mathcal{D}_{\mathsf{Circ}})$-unpredictability-style puncturing security. Moreover, if $m = \ell = 1$, the converse is also true, i.e., 1-point 1-bit $(\mathcal{D}_{\boldsymbol{X}}, \mathcal{D}_{\mathsf{Circ}})$-pseudorandomness-style puncturing security is equivalent to 1-point 1-bit $(\mathcal{D}_{\boldsymbol{X}}, \mathcal{D}_{\mathsf{Circ}})$-unpredictability-style puncturing security.*

Proof of Lemma 7.3. Let $(\mathsf{Circ}, \mathsf{Puncture})$ be a circuit class satisfying ℓ-point m-bit $(\mathcal{D}_{\mathbf{X}}, \mathcal{D}_{\mathsf{Circ}})$-pseudorandomness-style puncturing security. Let $\mathcal{A}$ be an adversary that wins the $(\mathcal{D}_{\mathbf{X}}, \mathcal{D}_{\mathsf{Circ}})$-unpredictability-style puncture security game with respect to some $\ell' \leq \ell$ with probability $p \geq \left(1 - (1 - 1/2^m)^{\ell'}\right)$. Hence the advantage of $\mathcal{A}$, $Adv(\mathcal{A}) = p - \left(1 - (1 - 1/2^m)^{\ell'}\right)$. Consider an adversary $\mathcal{B}$ in the $(\mathcal{D}_{\mathbf{X}}, \mathcal{D}_{\mathsf{Circ}})$-pseudorandomness-style puncture security game: $\mathcal{B}$ on input $(\widehat{C}, (x_1, \ldots, x_{\ell'}), (y_1, \ldots, y_{\ell'}))$, runs $y'_1, \ldots, y'_{\ell'} \leftarrow \mathcal{A}(\widehat{C}, (x_1, \ldots, x_{\ell'}))$, and outputs 0 if there exists $i \in [\ell']$ such that $y'_i = y_i$, else outputs 1. Clearly, for any $\ell' \leq \ell$,

$$\Pr_{\substack{y_i \leftarrow C(x_i) \forall i \in [\ell'], \\ \widehat{C} \leftarrow \mathsf{Puncture}(C, (x_1, \ldots, x_{\ell'})), \\ x_i \leftarrow \mathcal{D}_{\mathbf{X}} \forall i \in [\ell'], C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)}} [\mathcal{B}(\widehat{C}, (x_1, \ldots, x_{\ell'}), (y_1, \ldots, y_{\ell'})) = 1]$$

$$= 1 - \Pr_{\substack{(y'_1, \ldots, y'_{\ell'}) \leftarrow \mathcal{A}(\widehat{C}, (x_1, \ldots, x_{\ell'})), \\ y_i \leftarrow C(x_i) \forall i \in [\ell'], \\ \widehat{C} \leftarrow \mathsf{Puncture}(C, (x_1, \ldots, x_{\ell'})), \\ x_i \leftarrow \mathcal{D}_{\mathbf{X}} \forall i \in [\ell'], C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)}} [\exists i, \quad y'_i = y_i] = 1 - p.$$

[22] The definition in [AB24] was only defined for search function classes, i.e., with $m \in \omega(\log(\lambda))$, but we generalize their definition for function classes with arbitrary output sizes

[23] By a proof similar to that of Theorem 8.17 that we are going to see next, we believe that the implication in the converse direction is also true for the circuit class under a post-quantum iO $i\mathcal{O}$, i.e., if $i\mathcal{O}(\mathsf{Circ})$ satisfies 1-point m-bit $(\mathcal{D}_{\mathbf{X}}, \mathcal{D}_{\mathsf{Circ}})$-pseudorandomness-style puncturing security, then $i\mathcal{O}(\mathsf{Circ})$ should satisfy the ℓ-point m-bit $(\mathcal{D}_{\mathbf{X}}, \widetilde{\mathcal{D}}_{\mathsf{Circ}})$-pseudorandomness-style puncturing security where $\widetilde{\mathcal{D}}_{\mathsf{Circ}}$ is the same distribution as in Theorem 8.17. Hence, ℓ-point pseudorandomness-style puncturing security for any iO obfuscated circuit class should be morally equivalent to 1-point security, for any polynomial ℓ.

Next,

$$\Pr_{\substack{y_i \xleftarrow{\$} \mathbf{Y} \forall i \in [\ell'], \\ \widehat{C} \leftarrow \mathsf{Puncture}(C,(x_1,\ldots,x_{\ell'})), \\ x_i \leftarrow \mathcal{D}_{\mathbf{X}} \forall i \in [\ell'], C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)}} [\mathcal{B}(\widehat{C},(x_1,\ldots,x_{\ell'}),(y_1,\ldots,y_{\ell'}))=1]$$

$$= \Pr_{\substack{(y'_1,\ldots,y'_{\ell'}) \leftarrow \mathcal{A}(\widehat{C},(x_1,\ldots,x_{\ell'})), \\ y_i \xleftarrow{\$} \mathbf{Y} \forall i \in [\ell'], \\ \widehat{C} \leftarrow \mathsf{Puncture}(C,(x_1,\ldots,x_{\ell'})), \\ x_i \leftarrow \mathcal{D}_{\mathbf{X}} \forall i \in [\ell'], C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)}} [y'_i \neq y_i, \forall i \in [\ell']] = \left(1 - \frac{1}{2^m}\right)^{\ell'}.$$

Hence, the distinguishing advantage of $\mathcal{B}$ is given by

$$Adv(\mathcal{B}) = \left| \Pr_{\substack{y_i \leftarrow C(x_i) \forall i \in [\ell'], \\ \widehat{C} \leftarrow \mathsf{Puncture}(C,(x_1,\ldots,x_{\ell'})), \\ x_i \leftarrow \mathcal{D}_{\mathbf{X}} \forall i \in [\ell'], C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)}} [\mathcal{B}(\widehat{C},(x_1,\ldots,x_{\ell'}),(y_1,\ldots,y_{\ell'}))=1 \right.$$

$$\left. - \Pr_{\substack{y_i \xleftarrow{\$} \mathbf{Y} \forall i \in [\ell'], \\ \widehat{C} \leftarrow \mathsf{Puncture}(C,(x_1,\ldots,x_{\ell'})), \\ x_i \leftarrow \mathcal{D}_{\mathbf{X}} \forall i \in [\ell'], C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)}} [\mathcal{B}(\widehat{C},(x_1,\ldots,x_{\ell'}),(y_1,\ldots,y_{\ell'}))=1] \right|$$

$$= \left| 1 - p - \left(1 - \frac{1}{2^m}\right)^{\ell'} \right| = p - \left(1 - \left(1 - \frac{1}{2^m}\right)^{\ell'}\right) = Adv(\mathcal{A}).$$

Since $(\mathsf{Circ}, \mathsf{Puncture})$ satisfies ℓ-point m-bit $(\mathcal{D}_{\mathbf{X}}, \mathcal{D}_{\mathsf{Circ}})$-pseudorandomness-style puncturing security, $Adv(\mathcal{A}) = Adv(\mathcal{B})$ must be negligible, which completes the proof for the first part of the lemma.

Next for the "Moreover" part, let $(\mathsf{Circ}, \mathsf{Puncture})$ be a boolean output circuit class satisfying 1-point 1-bit $(\mathcal{D}_{\mathbf{X}}, \mathcal{D}_{\mathsf{Circ}})$-unpredictability-style puncturing security. It is enough to show that for every distinguisher $\mathcal{B}$,

$$Adv(\mathcal{B}) = \left| \Pr_{y \leftarrow C(x), \widehat{C} \leftarrow \mathsf{Puncture}(C,x), x \leftarrow \mathcal{D}_{\mathbf{X}}, C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)} [\mathcal{B}(\widehat{C},x,y)=1] \right.$$

$$\left. - \Pr_{y \xleftarrow{\$} \mathbf{Y}, \widehat{C} \leftarrow \mathsf{Puncture}(C,x), x \leftarrow \mathcal{D}_{\mathbf{X}}, C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)} [\mathcal{B}(\widehat{C},x,y)=1] \right|$$

is negligible.

For any bit $b^* \in \{0,1\}$ let $p_{b^*} := \Pr_{\widehat{C} \leftarrow \mathsf{Puncture}(C,x), C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)|_{C(x)=b^*}, x \leftarrow \mathcal{D}_{\mathbf{X}}}[B(\widehat{C},x,b^*)=1]$, and let $q_{b^*} := \Pr_{\widehat{C} \leftarrow \mathsf{Puncture}(C,x), C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)|_{C(x)=1-b^*}, x \leftarrow \mathcal{D}_{\mathbf{X}}}[B(\widehat{C},x,b^*)=1]$, where $\mathcal{D}_{\mathsf{Circ}}(1^\lambda)\,|_{C(x)=b^*}$ is the conditional distribution given by $\mathcal{D}_{\mathsf{Circ}}(1^\lambda)$ conditioned on the event that the sampled circuit C has output value b^* on x. Next for any bit $b^* \in \{0,1\}$ let $z_{b^*} := \Pr_{\widehat{C} \leftarrow \mathsf{Puncture}(C,x), C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda), x \leftarrow \mathcal{D}_{\mathbf{X}}}[C(x)=b^*]$

Next, note that for any $b^* \in \{0,1\}$, if $|p_{b^*} - q_{b^*}|$ is non-negligible, then WLOG, assume $p_{b^*} \geq q_{b^*}$, then we can violate 1-point $(\mathcal{D}_{\mathbf{X}}, \mathcal{D}_{\mathsf{Circ}})$-unpredictability-style puncturing security, by considering the following adversary $\mathcal{A}$, which on input $(\widehat{C},x)$ runs $d \leftarrow \mathcal{B}(\widehat{C},x,b^*)$ and outputs b^* if $d=1$ and $1-b^*$ if $d=0$. It is easy to see that the success

probability of $\mathcal{A}$ in the unpredictability puncture security game is precisely $p_{b^*} - q_{b^*}$. Hence ,we conclude by the 1-point $(\mathcal{D}_{\mathbf{X}}, \mathcal{D}_{\mathsf{Circ}})$-unpredictability-style puncturing security of $(\mathsf{Circ}, \mathsf{Puncture})$ that

$$|p_{b^*} - q_{b^*}| \leq \mathsf{negl}_{b^*}(\lambda), \tag{2}$$

for some negligible function $\mathsf{negl}_{b^*}(\cdot)$. Similarly, if $|z_{b^*} - 1/2|$ is non-negligible, then WLOG, assume $z_{b^*} \geq 1/2$, then we can violate 1-point $(\mathcal{D}_{\mathbf{X}}, \mathcal{D}_{\mathsf{Circ}})$-unpredictability-style puncturing security, by considering the following adversary $\widetilde{\mathcal{A}}$, which on input $(\widehat{C}, x)$ always outputs b^*. Clearly, the success probability of $\mathcal{A}$ in the unpredictability puncture security game is precisely $z_{b^*} - 1/2$. Hence ,we conclude by the 1-point $(\mathcal{D}_{\mathbf{X}}, \mathcal{D}_{\mathsf{Circ}})$-unpredictability-style puncturing security of $(\mathsf{Circ}, \mathsf{Puncture})$ that

$$|z_{b^*} - 1/2| \leq \widetilde{\mathsf{negl}}_{b^*}(\lambda), \tag{3}$$

for some negligible function $\widetilde{\mathsf{negl}}_{b^*}(\cdot)$. Finally, note that by definition,

$$\Pr_{y \leftarrow C(x), \widehat{C} \leftarrow \mathsf{Puncture}(C,x), x \leftarrow \mathcal{D}_{\mathbf{X}}, C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)} [\mathcal{B}(\widehat{C}, x, y) = 1] = \sum_{b^* \in \{0,1\}} z_{b^*} p_{b^*},$$

and

$$\begin{aligned}\Pr_{y \xleftarrow{\$} \mathbf{Y}, \widehat{C} \leftarrow \mathsf{Puncture}(C,x), x \leftarrow \mathcal{D}_{\mathbf{X}}, C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)} [\mathcal{B}(\widehat{C}, x, y) = 1] &= \sum_{b^* \in \{0,1\}} z_{b^*} \frac{(p_{b^*} + q_{1-b^*})}{2} \\ &= \sum_{b^* \in \{0,1\}} \frac{p_{b^*} z_{b^*} + q_{b^*} z_{1-b^*}}{2}.\end{aligned}$$

With this in mind, note that the advantage of $\mathcal{B}$ in the pseudorandomness-style puncture security game,

$$\begin{aligned}
&Adv(\mathcal{B}) \\
&= \left| \Pr_{\substack{y \leftarrow C(x), \widehat{C} \leftarrow \mathsf{Puncture}(C,x), \\ x \leftarrow \mathcal{D}_{\mathbf{X}}, C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)}} [\mathcal{B}(\widehat{C}, x, y) = 1] \right. \\
&\left. - \Pr_{\substack{y \xleftarrow{\$} \mathbf{Y}, \widehat{C} \leftarrow \mathsf{Puncture}(C,x), \\ x \leftarrow \mathcal{D}_{\mathbf{X}}, C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)}} [\mathcal{B}(\widehat{C}, x, y) = 1] \right| \\
&= \left| \sum_{b^* \in \{0,1\}} z_{b^*} p_{b^*} - \sum_{b^* \in \{0,1\}} \frac{p_{b^*} z_{b^*} + q_{b^*} z_{1-b^*}}{2} \right| \\
&\leq \left| \sum_{b^* \in \{0,1\}} \frac{p_{b^*}}{2} - \sum_{b^* \in \{0,1\}} \frac{\frac{p_{b^*} + q_{b^*}}{2}}{2} \right| + \sum_{b^* \in \{0,1\}} \left| \frac{p_{b^*} - q_{1-b^*}}{2} \widetilde{\mathsf{negl}}_{b^*}(\lambda) \right| \\
&\leq \left| \sum_{b^* \in \{0,1\}} \frac{p_{b^*}}{2} - \sum_{b^* \in \{0,1\}} \frac{\frac{p_{b^*} + q_{b^*}}{2}}{2} \right| + \epsilon(\lambda) && \text{By (3)} \\
&= \left| \sum_{b^* \in \{0,1\}} \frac{p_{b^*} - q_{b^*}}{4} \right| + \epsilon(\lambda) \\
&\leq 1/4 \sum_{b^* \in \{0,1\}} |p_{b^*} - q_{b^*}| + \epsilon(\lambda) \\
&\leq \frac{\sum_{b^* \in \{0,1\}} \mathsf{negl}_{b^*}}{4} + \epsilon(\lambda), && \text{By (2)}
\end{aligned}$$

where $\epsilon(\lambda) := \left(\max_{b^*} \widetilde{\mathsf{negl}}_{b^*}(\lambda)\right) \sum_{b^* \in \{0,1\}} \frac{|p_{b^*} - q_{1-b^*}|}{2}$ is a negligible function in λ and hence the last expression is negligible in the security parameter λ. □

7.2 Oracular Pseudorandomness-Style Copy-Protection for Pseudorandomness-Style Puncturable-Secure Circuits

Theorem 7.4. *Let $\mathcal{D}_{\boldsymbol{X}}$ be a high min-entropy distribution on the input space $\boldsymbol{X}$ and $(\mathsf{Circ}, \mathsf{Puncture})$ be a puncturable circuit class satisfying 2-point m-bit $(\mathcal{D}_{\boldsymbol{X}}, \mathcal{D}_{\mathsf{Circ}})$-pseudorandomness-style puncturing security (see Definition 7.1).*

Then any UPO scheme $\mathsf{UPO} = (\mathcal{O}bf, \mathcal{E}val)$ satisfying $(\mathcal{D}_{\boldsymbol{X}} \times \mathcal{D}_{\boldsymbol{X}})$-generalized UPO anti-piracy for Circ with respect to $\mathsf{Program}_{canonical}$, as well as iO security (see Definition 3.11 and Lemma 3.16), is also a $(\mathcal{D}_{\mathsf{Circ}} \times \mathcal{D}_{\boldsymbol{X}} \times \mathcal{D}_{\boldsymbol{X}})$-oracular-pseudorandomness-style copy-protection scheme for Circ with $\mathcal{C}opy\mathcal{P}rotect = \mathsf{UPO}.\mathcal{O}bf$ and $\mathcal{E}val = \mathsf{UPO}.\mathcal{E}val$ (see Definition 3.3).

Moreover, if the underlying UPO scheme $\mathsf{UPO} = (\mathcal{O}bf, \mathcal{E}val)$ satisfies $(\mathcal{D}_{\boldsymbol{X}} \times \mathcal{D}_{\boldsymbol{X}})$-generalized-UPO$^+$ antipiracy for Circ (see Definition 3.14) as well as iO security (Lemma 3.16), then the copy-protection scheme also satisfies $(\mathcal{D}_{\mathsf{Circ}} \times \mathcal{D}_{\boldsymbol{X}} \times \mathcal{D}_{\boldsymbol{X}})$-oracular-pseudorandomness-style CP+ anti-piracy (see Definition 3.4).

Proof of Theorem 7.4. The correctness is immediate from the correctness of UPO.

Let $\mathsf{Program}$ denote the algorithm that on input $(C, (x_1, \ldots, x_\ell), (y_1, \ldots, y_\ell))$ outputs runs $\widehat{C}_{x_1,\ldots,x_\ell} \leftarrow \mathsf{Puncture}(C, (x_1, \ldots, x_\ell))$, and then outputs the circuit $\widehat{C}$ that has $(y_1, \ldots, y_\ell)$ and $\widehat{C}_{x_1,\ldots,x_\ell}$ hardcoded in it such that on every input $x \in \mathbf{X} \setminus \{x_1, \ldots, x_\ell\}$, $\widehat{C}$ outputs $\widehat{C}_{x_1,\ldots,x_\ell}(x)$ and for every input $x = x_j$ for $j \in [\ell]$, outputs y_j.

We provide the $(\mathcal{D}_{\mathsf{Circ}} \times \mathcal{D}_{\mathbf{X}} \times \mathcal{D}_{\mathbf{X}})$-oracular-pseudorandomness-style copy-protection anti-piracy below. Consider the following hybrids. The changes are highlighted in red.

$\mathbf{H}_0$: this corresponds to the pseudorandomness-style copy-protection security experiment. Denote the copy-protection adversary to be $(\mathcal{A}, \mathcal{B}, \mathcal{C})$.

- Sample $x_1 \leftarrow \mathcal{D}_{\mathbf{X}}$ and $x_2 \leftarrow \mathcal{D}_{\mathbf{X}}$ independently, and $b \xleftarrow{\$} \{0,1\}$,
- $\mathcal{A}$ gets $\mathsf{UPO}.\mathcal{O}bf(1^\lambda, C)$ and oracle access to C, where $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$,
- After the splitting experiment, along with oracle access to $C \setminus \{x_1\}$ and $C \setminus \{x_2\}$ respectively, if $b = 0$, $\mathcal{B}$ and $\mathcal{C}$ receive (x_1, y_1^0) and (x_2, y_2^0) repectively, where $y_i^0 = C(x_i)$ for every $i \in [2]$, and if $b = 1$, they receive (x_1, y_1^1) and (x_2, y_2^1) repectively, where $y_i^1 \xleftarrow{\$} \mathbf{Y}$ for every $i \in [2]$.
- $(\mathcal{B}, \mathcal{C})$ output $(b_{\mathcal{B}}, b_{\mathcal{C}})$.

The output of the hybrid is $b_{\mathcal{B}}, b_{\mathcal{C}}, b$. Denote the probability that $b_{\mathcal{B}} = b_{\mathcal{C}} = b$ to be p_0.

$\mathbf{H}_1$: this hybrid is defined as follows:

- Sample $x_1 \leftarrow \mathcal{D}_{\mathbf{X}}$ and $x_2 \leftarrow \mathcal{D}_{\mathbf{X}}$ independently, and $b \xleftarrow{\$} \{0,1\}$,
- $\mathcal{A}$ gets oracle access to $\widehat{C}$, and the state $\mathsf{UPO}.\mathcal{O}bf(1^\lambda, \widehat{C})$, where if $b = 0$, $\widehat{C} = C$, $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$, else $\widehat{C} \leftarrow \mathsf{Program}(C, (x_1, x_2), (y_1, y_2))$, $y_1 \leftarrow C(x_1), y_2 \leftarrow C(x_2), C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$.
- After the splitting experiment, along with oracle access to $\widehat{C} \setminus \{x_1\}$ and $\widehat{C} \setminus \{x_2\}$ respectively, if $b = 0$, $\mathcal{B}$ and $\mathcal{C}$ receive (x_1, y_1^0) and (x_2, y_2^0) repectively, where $y_i^0 = C(x_i)$ for every $i \in [2]$, and if $b = 1$, they receive (x_1, y_1^1) and (x_2, y_2^1) repectively, where $y_i^1 \xleftarrow{\$} \mathbf{Y}$ for every $i \in [2]$.
- $(\mathcal{B}, \mathcal{C})$ output $(b_{\mathcal{B}}, b_{\mathcal{C}})$.

The output of the hybrid is $b_{\mathcal{B}}, b_{\mathcal{C}}, b$. Denote the probability that $b_{\mathcal{B}} = b_{\mathcal{C}} = b$ to be p_1.

The only difference from $\mathbf{H}_0$ to $\mathbf{H}_1$ is that we replaced C with $\widehat{C}$. Since the functionalities of C and $\widehat{C}$ are the same, by the indistinguishability obfuscation guarantees of UPO, computational indistinguishability between the outputs of $\mathbf{H}_0$ and $\mathbf{H}_1$ holds. Hence

$$|p_0 - p_1| \leq \epsilon_1(\lambda),$$

for some negligible function $\epsilon_1(\cdot)$.

$\mathbf{H}_2$: this hybrid is defined as follows:

- Sample $x_1 \leftarrow \mathcal{D}_{\mathbf{X}}$ and $x_2 \leftarrow \mathcal{D}_{\mathbf{X}}$ independently, and $b \xleftarrow{\$} \{0,1\}$,
- $\mathcal{A}$ gets oracle access to $\widehat{C}$, and the state $\mathsf{UPO}.\mathcal{O}\mathit{bf}(1^\lambda, \widehat{C})$, where if $b = 0$, $\widehat{C} = C$, $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$, else $\widehat{C} \leftarrow \mathsf{Program}(C, (x_1, x_2), (y_1, y_2))$, $y_1, y_2 \xleftarrow{\$} \mathbf{Y}$, $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$.
- After the splitting experiment, along with oracle access to $\widehat{C} \setminus \{x_1\}$ and $\widehat{C} \setminus \{x_2\}$ respectively, if $b = 0$, $\mathcal{B}$ and $\mathcal{C}$ receive (x_1, y_1^0) and (x_2, y_2^0) repectively, where $y_i^0 = C(x_i)$ for every $i \in [2]$, and if $b = 1$, they receive (x_1, y_1^1) and (x_2, y_2^1) repectively, where $y_i^1 \xleftarrow{\$} \mathbf{Y}$ for every $i \in [2]$.
- $(\mathcal{B}, \mathcal{C})$ output $(b_{\mathcal{B}}, b_{\mathcal{C}})$.

The output of the hybrid is $b_{\mathcal{B}}, b_{\mathcal{C}}, b$. Denote the probability that $b_{\mathcal{B}} = b_{\mathcal{C}} = b$ to be p_2.

The indistinguishability between the outputs of $\mathbf{H}_1$ and $\mathbf{H}_2$ holds by 2-point m-bit puncturable circuit class satisfying $(\mathcal{D}_{\mathbf{X}}, \mathcal{D}_{\mathsf{Circ}})$-pseudorandomness-style puncturing security of Circ with respect to Puncture. Hence

$$|p_1 - p_2| \leq \epsilon_2(\lambda),$$

for some negligible function $\epsilon_2(\cdot)$.

$\mathbf{H}_3$: this hybrid is defined as follows:

- Sample $x_1 \leftarrow \mathcal{D}_{\mathbf{X}}$ and $x_2 \leftarrow \mathcal{D}_{\mathbf{X}}$ independently, and $b \xleftarrow{\$} \{0,1\}$,
- $\mathcal{A}$ gets oracle access to $\widehat{C}$, and the state $\mathsf{UPO}.\mathcal{O}\mathit{bf}(1^\lambda, \widehat{C})$, where if $b = 0$, $\widehat{C} = C$, $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$, else $\widehat{C} \leftarrow \mathsf{Program}(C, (x_1, x_2), (y_1, y_2))$, $y_1, y_2 \xleftarrow{\$} \mathbf{Y}$, $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$.
- After the splitting experiment, along with oracle access to $\widehat{C} \setminus \{x_1\}$ and $\widehat{C} \setminus \{x_2\}$ respectively, if $b = 0$, $\mathcal{B}$ and $\mathcal{C}$ receive (x_1, y_1^0) and (x_2, y_2^0) repectively, where $y_i^0 = C(x_i)$ for every $i \in [2]$, and if $b = 1$, they receive (x_1, y_1^1) and (x_2, y_2^1) repectively, where $y_i^1 = C(x_i)$ for every $i \in [2]$.
- $(\mathcal{B}, \mathcal{C})$ output $(b_{\mathcal{B}}, b_{\mathcal{C}})$.

The output of the hybrid is $b_{\mathcal{B}}, b_{\mathcal{C}}, b$. Denote the probability that $b_{\mathcal{B}} = b_{\mathcal{C}} = b$ to be p_3.

The indistinguishability between the outputs of $\mathbf{H}_1$ and $\mathbf{H}_2$ holds by 2-point m-bit puncturable circuit class satisfying $(\mathcal{D}_{\mathbf{X}}, \mathcal{D}_{\mathsf{Circ}})$-pseudorandomness-style puncturing security of Circ with respect to Puncture. Hence

$$|p_2 - p_3| \leq \epsilon_3(\lambda),$$

for some negligible function $\epsilon_3(\cdot)$.

$\mathbf{H}_4$: this hybrid is defined as follows:

- Sample $x_1 \leftarrow \mathcal{D}_{\mathbf{X}}$ and $x_2 \leftarrow \mathcal{D}_{\mathbf{X}}$ independently, and $b \xleftarrow{\$} \{0,1\}$,

- $\mathcal{A}$ gets oracle access to $\widehat{C}$, and the state $\mathsf{UPO}.\mathcal{O}bf(1^\lambda, \widehat{C})$, where if $b = 0$, $\widehat{C} = C$, $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$, else $\widehat{C} \leftarrow \mathsf{Program}(C, (x_1, x_2), (y_1, y_2))$, where $y_1, y_2 \xleftarrow{\$} \mathbf{Y}$, $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$.
- After the splitting experiment, along with oracle access to $\widehat{C} \setminus \{x_1\}$ and $\widehat{C} \setminus \{x_2\}$ respectively, $\mathcal{B}$ and $\mathcal{C}$ receive $(x_1, C(x_1))$ and $(x_2, C(x_2))$ respectively.
- $(\mathcal{B}, \mathcal{C})$ output $(b_{\mathcal{B}}, b_{\mathcal{C}})$.

The output of the hybrid is $b_{\mathcal{B}}, b_{\mathcal{C}}, b$. Denote the probability that $b_{\mathcal{B}} = b_{\mathcal{C}} = b$ to be p_4.

The outputs of $\mathbf{H}_3$ and $\mathbf{H}_4$ have the same distribution since the distribution over y_0 and y_1 was the same in $\mathbf{H}_3$. Hence,

$$p_4 = p_3.$$

$\mathbf{H}_5$: this hybrid is defined as follows:

- Sample $x_1 \leftarrow \mathcal{D}_{\mathbf{X}}$ and $x_2 \leftarrow \mathcal{D}_{\mathbf{X}}$ independently, and $b \xleftarrow{\$} \{0,1\}$,
- $\mathcal{A}$ gets oracle access to $\widehat{C}$, and the state $\mathsf{UPO}.\mathcal{O}bf(1^\lambda, \widehat{C})$, where if $b = 0$, $\widehat{C} = C$, $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$, else $\widehat{C} \leftarrow \mathsf{Program}(C, (x_1, x_2), (y_1, y_2))$, where $y_1, y_2 \xleftarrow{\$} \mathbf{Y}$, $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$.
- After the splitting experiment, along with oracle access to $C \setminus \{x_1\}$ and $\widehat{C} \setminus \{x_2\}$ respectively, $\mathcal{B}$ and $\mathcal{C}$ receive $(x_1, C(x_1))$ and $(x_2, C(x_2))$ respectively.
- $(\mathcal{B}, \mathcal{C})$ output $(b_{\mathcal{B}}, b_{\mathcal{C}})$.

The output of the hybrid is $b_{\mathcal{B}}, b_{\mathcal{C}}, b$. Denote the probability that $b_{\mathcal{B}} = b_{\mathcal{C}} = b$ to be p_5.

Since $\widehat{C} \setminus \{x_1\}$ and $C \setminus \{x_1\}$ differ in functionality only at x_2, the challenge point for $\mathcal{C}$, the only difference between $\mathbf{H}_4$ and $\mathbf{H}_5$ is that we change the oracle access to $\mathcal{B}$ by changing the oracle output at x_2.

Let $q_{\mathcal{B}}(\lambda)$ be the number of oracle queries that $\mathcal{B}$ makes for some fixed polynomial $q_{\mathcal{B}}(\lambda)$[24]. In $\mathbf{H}_4$, for each $i \in [q_{\mathcal{B}}]$, let $W^i_{4,\mathcal{B}}$ denote the total weight of i^{th} query on x_2, where the i^{th} query can be written as $\sum_{x,y} \alpha_{x,y,i} |x\rangle_{\mathsf{I}} |y\rangle_{\mathsf{O}}$, for input and output registers I and O respectively. In other words, $W^i_{4,\mathcal{B}} = \sum_{y \in \mathbf{Y}} |\alpha_{x_2,y,i}|^2$. Let $W_{4,\mathcal{B}} = \sum_i W^i_{4,\mathcal{B}}$ be the combined weight of $\mathcal{B}$'s queries on x_2 as query points in $\mathbf{H}_4$. We will show the following claim.

Claim 7.5. $\mathbb{E}[W_{4,\mathcal{B}}]$ is negligible in λ.

Combining Claim 7.5 with the fact that $q_{\mathcal{B}}$ is a polynomial in λ, we conclude that $q_{\mathcal{B}} \cdot \mathbb{E}[W_{4,\mathcal{B}}]$ is also negligible in λ. Hence, by Theorem 2.1, where we treat $T = q_{\mathcal{B}}$, we conclude that changing the oracle output for $\mathcal{B}$ at x_2, in $\mathbf{H}_4$ results in a statistically indistinguishable output distribution, i.e., $\{b_{\mathcal{B}}, b_{\mathcal{C}}, b\}$. Since the only difference between $\mathbf{H}_4$ and $\mathbf{H}_5$ is that the oracle output for $\mathcal{B}$ at x_2 is different, we conclude that the output distribution in $\mathbf{H}_4$ is computationally indistinguishable[25] from that of $\mathbf{H}_5$. Hence,

$$|p_4 - p_5| = \epsilon_5(\lambda),$$

for some negligible function $\epsilon_5(\cdot)$.

Next, we prove Claim 7.5 to complete the proof of indistinguishability for the output distributions of $\mathbf{H}_4$ and $\mathbf{H}_5$.

Proof of Claim 7.5. Let $W_{0,\mathcal{B}}$ be the combined weight of $\mathcal{B}$'s queries on x_2 as query points in $\mathbf{H}_0$, i.e., $W_{0,\mathcal{B}} = \sum_{i \in [q_{\mathcal{B}}]} W^i_{0,\mathcal{B}}$ where for each $i \in [q_{\mathcal{B}}]$, $W^i_{\mathcal{B}}$ denotes the total weight of i^{th} query on x_2. In $\mathbf{H}_0$, x_2 can be sampled after $\mathcal{B}$'s queries as x_2 is sampled independently of $\mathcal{B}$'s state and challenge x_1. Therefore, since x_2 are sampled from $\mathcal{D}_{\mathbf{X}}$, there exists a negliglbe function $\mathsf{negl}(\cdot)$ defined as $\mathsf{negl}(\lambda) := 2^{-\mathsf{min\text{-}entropy}(\mathcal{D}_{\mathbf{X}})}$, such that for each $i \in [q_{\mathcal{B}}]$, $\mathbb{E}[W^i_{0,\mathcal{B}}] = \mathsf{negl}(\lambda)$, i.e., $\mathbb{E}[W_{0,\mathcal{B}}] = q_{\mathcal{B}} \cdot \mathsf{negl}(\lambda)$, which is negligible in the security parameter λ.

Next consider, the following distinguishing attack $(\mathcal{A}, \widehat{\mathcal{B}}, \widehat{\mathcal{C}})$:

[24] WLOG, we can assume that $\mathcal{B}$ makes a fixed polynomial number of queries in all the hybrids.

[25] The reason this indistinguihsability is computational rather than statistical is that Claim 7.5 holds only for computationally bounded $\mathcal{B}$.

1. $\widehat{\mathcal{B}}$ runs $\mathcal{B}$ on the challenge and the state received from $\mathcal{A}$ and performs the oracle queries from $\mathcal{B}$ and feeds the output back to $\mathcal{B}$ up until the j^{th} query, where $j \xleftarrow{\$} [q_{\mathcal{A}}]$ is sampled ahead of time. On the j^{th} query, $\widehat{\mathcal{B}}$ measures the query input register I. Let the measurement outcome be x'. $\widehat{\mathcal{B}}$ outputs x'.

2. $\widehat{\mathcal{C}}$ on receiving a challenge from the challenger, outputs the challenge itself.

For any $h \in \{0,4\}$, let the probability that with adversaries $(\mathcal{A}, \widehat{\mathcal{B}}, \widehat{\mathcal{C}})$ in $\mathbf{H}_h$, $\widehat{\mathcal{B}}$ and $\widehat{\mathcal{C}}$ outputs the same string, be p_h^{eq}.

Since the output distribution of $\mathbf{H}_4$ is computationally indistinguishable from that of $\mathbf{H}_0$, $|p_4^{eq} - p_0^{eq}|$ must be negligible in λ. However, note that by definition of $\widehat{\mathcal{B}}$ and $\widehat{\mathcal{C}}$, they output the same string if and only if x', the measurement outcome of a random query of $\mathcal{B}$, is the same as x_2, the challenge for $\mathcal{C}$. Hence, for any $h \in \{0,4\}$,

$$p_h^{eq} = \mathbb{E}_{i \xleftarrow{\$} [q_{\mathcal{B}}]} \mathbb{E}[W_{h,\mathcal{B}}^i] = \sum_{i \in [q_{\mathcal{B}}]} \frac{\mathbb{E}[W_{h,\mathcal{B}}^i]}{q_{\mathcal{B}}} = \frac{\mathbb{E}[W_{h,\mathcal{B}}]}{q_{\mathcal{B}}}.$$

Therefore, by the computational indistinguishability of the output distributions of $\mathbf{H}_0$ and $\mathbf{H}_4$, we conclude that $\frac{|\mathbb{E}[W_{4,\mathcal{B}}] - \mathbb{E}[W_{0,\mathcal{B}}]|}{q_{\mathcal{B}}} = |\frac{\mathbb{E}[W_{4,\mathcal{B}}]}{q_{\mathcal{B}}} - \frac{\mathbb{E}[W_{0,\mathcal{B}}}{q_{\mathcal{B}}}]| = |p_4^{eq} - p_0^{eq}|$ is negligible, which implies that $|\mathbb{E}[W_{4,\mathcal{B}}] - \mathbb{E}[W_{0,\mathcal{B}}]|$ must be negligible in λ, since $q_{\mathcal{B}}$ is polynomial in λ. Moreover since $\mathbb{E}[W_{0,\mathcal{B}}]$ is negligible as concluded above, $\mathbb{E}[W_{4,\mathcal{B}}]$ must be negligible in λ. □

$\mathbf{H}_6$: this hybrid is defined as follows:

- Sample $x_1 \leftarrow \mathcal{D}_{\mathbf{X}}$ and $x_2 \leftarrow \mathcal{D}_{\mathbf{X}}$ independently, and $b \xleftarrow{\$} \{0,1\}$,
- $\mathcal{A}$ gets oracle access to $\widehat{C}$, and the state $\mathsf{UPO}.\mathcal{O}\mathit{bf}(1^\lambda, \widehat{C})$, where if $b = 0$, $\widehat{C} = C$, $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$, else $\widehat{C} \leftarrow \mathsf{Program}(C, (x_1, x_2), (y_1, y_2))$, where $y_1, y_2 \xleftarrow{\$} \mathbf{Y}$, $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$.
- After the splitting experiment, along with oracle access to $C \setminus \{x_1\}$ and $C \setminus \{x_2\}$ respectively, $\mathcal{B}$ and $\mathcal{C}$ receive $(x_1, C(x_1))$ and $(x_2, C(x_2))$ respectively.
- $(\mathcal{B}, \mathcal{C})$ output $(b_{\mathcal{B}}, b_{\mathcal{C}})$.

The output of the hybrid is $b_{\mathcal{B}}, b_{\mathcal{C}}, b$. Denote the probability that $b_{\mathcal{B}} = b_{\mathcal{C}} = b$ to be p_6.

Since $\widehat{C} \setminus \{x_2\}$ and $C \setminus \{x_2\}$ differ in functionality only at x_1, the challenge point for $\mathcal{B}$, the only difference between $\mathbf{H}_5$ and $\mathbf{H}_6$ is that we change the oracle access to $\mathcal{C}$ by changing the oracle output at x_1.

By analogous arguments as in the proof of indistinguishability of $\mathbf{H}_4$ and $\mathbf{H}_5$ but with the roles of $\mathcal{B}$ and $\mathcal{C}$ switched, we can conclude that the outputs of $\mathbf{H}_5$ and $\mathbf{H}_6$ are computationally indistinguishable and hence,

$$|p_6 - p_5| = \epsilon_6(\lambda),$$

for some negligible function $\epsilon_6(\cdot)$.

$\mathbf{H}_7$: this hybrid is defined as follows:

- Sample $x_1 \leftarrow \mathcal{D}_{\mathbf{X}}$ and $x_2 \leftarrow \mathcal{D}_{\mathbf{X}}$ independently, and $b \xleftarrow{\$} \{0,1\}$,
- $\mathcal{A}$ gets oracle access to C, and the state $\mathsf{UPO}.\mathcal{O}\mathit{bf}(1^\lambda, \widehat{C})$, where if $b = 0$, $\widehat{C} = C$, $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$, else $\widehat{C} \leftarrow \mathsf{Program}(C, (x_1, x_2), (y_1, y_2))$, where $y_1, y_2 \xleftarrow{\$} \mathbf{Y}$, $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$.
- After the splitting experiment, along with oracle access to $C \setminus \{x_1\}$ and $C \setminus \{x_2\}$ respectively, $\mathcal{B}$ and $\mathcal{C}$ receive $(x_1, C(x_1))$ and $(x_2, C(x_2))$ respectively.
- $(\mathcal{B}, \mathcal{C})$ output $(b_{\mathcal{B}}, b_{\mathcal{C}})$.

The output of the hybrid is $b_{\mathcal{B}}, b_{\mathcal{C}}, b$. Denote the probability that $b_{\mathcal{B}} = b_{\mathcal{C}} = b$ to be p_7.

Since $\widehat{C}$ and C differ in functionality only at x_1 and x_2, the respective challenge points for $\mathcal{B}$ and $\mathcal{C}$, the only difference between $\mathbf{H}_7$ and $\mathbf{H}_6$ is that we change the oracle access to $\mathcal{A}$ by changing the oracle output at x_1 and x_2.

Let $q_{\mathcal{A}}(\lambda)$ be the number of oracle queries that $\mathcal{B}$ makes for some fixed polynomial $q_{\mathcal{A}}(\lambda)$. In $\mathbf{H}_6$, for each $i \in [q_{\mathcal{A}}]$, let $W^i_{6,\mathcal{A}}$ denote the total weight of i^{th} query on x_1 or x_2, where the i^{th} query can be written as $\sum_{x,y} \alpha_{x,y,i} |x\rangle_{\mathsf{I}} |y\rangle_{\mathsf{O}}$, for input and output registers I and O respectively. In other words, $W^i_{6,\mathcal{A}} = \sum_{x\in\{x_1,x_2\},y\in\mathbf{Y}} |\alpha_{x,y,i}|^2$. Let $W_{6,\mathcal{A}} = \sum_i W^i_{2,\mathcal{A}}$ be the combined weight of $\mathcal{A}$'s queries on x_1 or x_2 as query points in $\mathbf{H}_6$. We will show the following claim.

Claim 7.6. $\mathbb{E}[W_{6,\mathcal{A}}]$ is negligible in λ.

Combining Claim 7.6 with the fact that $q_{\mathcal{A}}$ is a polynomial in λ, we conclude that $q_{\mathcal{A}} \cdot \mathbb{E}[W_{6,\mathcal{A}}]$ is also negligible in λ. Hence, by Theorem 2.1, where we treat $T = q_{\mathcal{A}}$, we conclude that changing the oracle output for $\mathcal{A}$ at x_1 and x_2, in $\mathbf{H}_6$ results in a statistically indistinguishable output distribution, i.e., $\{b_{\mathcal{B}}, b_{\mathcal{C}}, b\}$. Since the only difference between $\mathbf{H}_6$ and $\mathbf{H}_7$ is that the oracle outputs for $\mathcal{A}$ at x_1 and x_2, are different, we conclude that the output distribution in $\mathbf{H}_6$ is computationally indistinguishable[26] from that of $\mathbf{H}_7$. Hence,

$$|p_7 - p_6| = \epsilon_7(\lambda),$$

for some negligible function $\epsilon_7(\cdot)$.

Next, we prove Claim 7.6 to complete the proof of indistinguishability for $\mathbf{H}_6$ and $\mathbf{H}_7$.

Proof of Claim 7.6. Let $W_{0,\mathcal{A}}$ be the combined weight of $\mathcal{A}$'s queries on x_1 or x_2, the challenge points for $\mathcal{B}$ and $\mathcal{C}$, as query points in $\mathbf{H}_0$, i.e., $W_{0,\mathcal{A}} = \sum_{i\in[q_{\mathcal{A}}]} W^i_{0,\mathcal{A}}$ where for each $i \in [q_{\mathcal{A}}]$, $W^i_{\mathcal{A}}$ denotes the total weight of i^{th} query on x_1 or x_2. In $\mathbf{H}_0$, x_1 and x_2 can be sampled after $\mathcal{A}$'s queries as x_1, x_2 are sampled independent of $\mathcal{A}$'s input state. Therefore, since x_1, x_2 are sampled from $\mathcal{D}_{\mathbf{X}}$, there exists a negliglbe function $\mathsf{negl}(\cdot)$ defined as $\mathsf{negl}(\lambda) := 2^{-\text{min-entropy}(\mathcal{D}_{\mathbf{X}})}$, such that for each $i \in [q_{\mathcal{A}}]$, $\mathbb{E}[W^i_{0,\mathcal{A}}] \leq 2 \cdot \mathsf{negl}(\lambda)$, i.e., $\mathbb{E}[W_{0,\mathcal{A}}] \leq 2q_{\mathcal{A}} \cdot \mathsf{negl}(\lambda)$, which is negligible in the security parameter λ.

Next consider, the following distinguishing attack $(\widehat{\mathcal{A}}, \widehat{\mathcal{B}}, \widehat{\mathcal{C}})$:

1. $\widehat{\mathcal{A}}$ runs $\mathcal{A}$ on the received program ρ from the challenger and performs the oracle queries from $\mathcal{A}$ and feeds the output back to $\mathcal{A}$ up until the j^{th} query, where $j \xleftarrow{\$} [q_{\mathcal{A}}]$ is sampled ahead of time. On the j^{th} query, $\widehat{\mathcal{A}}$ measures the query input register I. Let the measurement outcome be x'. $\mathcal{A}$ sends x' to both $\mathcal{B}$ and $\mathcal{C}$.

2. $\widehat{\mathcal{B}}$ on receiving string x' from $\mathcal{A}$ and a challenge string x_1 from the challenger checks if $x_1 = x'$, and if so output 0 else outputs 1.

3. $\widehat{\mathcal{C}}$ on receiving string x' from $\mathcal{A}$ and a challenge string x_2 from the challenger checks if $x_2 = x'$, and if so output 0 else outputs 1.

For any $h \in \{0, 6\}$, let the probability that with adversaries $(\mathcal{A}, \widehat{\mathcal{B}}, \widehat{\mathcal{C}})$ in $\mathbf{H}_h$, $\widehat{\mathcal{B}}$ and $\widehat{\mathcal{C}}$ both output the string 1, be p^{11}_h.

Since the output distribution of $\mathbf{H}_6$ is computationally indistinguishable from that of $\mathbf{H}_0$, $|p^{11}_6 - p^{11}_0|$ must be negligible in λ. However, note that by definition of $\widehat{\mathcal{A}}, \widehat{\mathcal{B}}, \widehat{\mathcal{C}}$, $\widehat{\mathcal{B}}$ and $\widehat{\mathcal{C}}$ both output 1 if and only if x', the measurement outcome of a random query of $\mathcal{A}$, satisfies $x' \neq x_1$ and $x' \neq x_2$, i.e., $x' \notin \{x_1, x_2\}$.

Hence, for any $h \in \{0, 6\}$,

$$p^{11}_h = 1 - \Pr\left[x' \in \{x_1, x_2\}\right] = 1 - \mathbb{E}_{i\xleftarrow{\$}[q_{\mathcal{A}}]} \mathbb{E}[W^i_{h,\mathcal{A}}] = 1 - \sum_{i\in[q_{\mathcal{A}}]} \frac{\mathbb{E}[W^i_{h,\mathcal{A}}]}{q_{\mathcal{A}}} = 1 - \frac{\mathbb{E}[W_{h,\mathcal{A}}]}{q_{\mathcal{A}}}.$$

Therefore, by the computational indistinguishability of $\mathbf{H}_0$ and $\mathbf{H}_6$, we conclude that $\frac{|\mathbb{E}[W_{6,\mathcal{A}}] - \mathbb{E}[W_{0,\mathcal{A}}]|}{q_{\mathcal{A}}} = |p^{11}_6 - p^{11}_0|$ is negligible, which implies that $|\mathbb{E}[W_{6,\mathcal{A}}] - \mathbb{E}[W_{0,\mathcal{A}}]|$ must be negligible in λ because $q_{\mathcal{A}}$ is polynomial in λ. Moreover since $\mathbb{E}[W_{0,\mathcal{A}}]$ is negligible as concluded above, $\mathbb{E}[W_{6,\mathcal{A}}]$ must be negligible in λ. □

[26] The reason this indistinguishability is computational rather than statistical is that Claim 7.6 holds only for computationally bounded $\mathcal{A}$.

$\mathbf{H}_8$: this hybrid is defined as follows:

- Sample $x_1 \leftarrow \mathcal{D}_{\mathbf{X}}$ and $x_2 \leftarrow \mathcal{D}_{\mathbf{X}}$ independently, and $b \xleftarrow{\$} \{0,1\}$,
- $\mathcal{A}$ gets oracle access to C, and the state $\mathsf{UPO}.\mathcal{O}bf(1^\lambda, \widehat{C})$, where if $b = 0$, $\widehat{C} = C$, $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$, else $\widehat{C} \leftarrow \mathsf{Program}_{canonical}(C, (x_1, x_2), (y_1, y_2))$, where $y_1, y_2 \xleftarrow{\$} \mathbf{Y}$, $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$.
- After the splitting experiment, along with oracle access to $C \setminus \{x_1\}$ and $C \setminus \{x_2\}$ respectively, $\mathcal{B}$ and $\mathcal{C}$ receive $(x_1, C(x_1))$ and $(x_2, C(x_2))$ respectively.
- $(\mathcal{B}, \mathcal{C})$ output $(b_{\mathcal{B}}, b_{\mathcal{C}})$.

The output of the hybrid is $b_{\mathcal{B}}, b_{\mathcal{C}}, b$. Denote the probability that $b_{\mathcal{B}} = b_{\mathcal{C}} = b$ to be p_8.

The only difference between $\mathbf{H}_7$ and $\mathbf{H}_8$ is that we replace $\mathsf{Program}$ with $\mathsf{Program}_{canonical}$, and since the output circuits of both $\mathsf{Program}$ and $\mathsf{Program}_{canonical}$ have the same functionality by the puncturing correctness (see Definition 3.10), by the iO security of UPO, the outputs of $\mathbf{H}_7$ and $\mathbf{H}_8$ must be computationally indistinguishable, and hence,

$$|p_7 - p_8| = \epsilon_8(\lambda),$$

for some negligible function $\epsilon_8(\cdot)$.

Reduction to UPO Let $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ be an adversary in $\mathbf{H}_8$. We will construct a reduction adversary $(\mathcal{R}_{\mathcal{A}}, \mathcal{R}_{\mathcal{B}}, \mathcal{R}_{\mathcal{C}})$ in the $\mathcal{D}_{\mathbf{X}} \times \mathcal{D}_{\mathbf{X}}$-generalized UPO anti-piracy game.

- $\mathcal{R}_{\mathcal{A}}$ samples a circuit $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$ and $y_1, y_2 \xleftarrow{\$}$ and sends $(C, 1_{y_1}, 1_{y_2})$ to the challenger, where the programmed circuits (at the points of puncture) 1_{y_1} and 1_{y_2} are the constant circuits always outputting y_1 and y_2, respectively.
- On receiving an obfuscated state ρ from the challenger, $\mathcal{R}_{\mathcal{A}}$ runs $\sigma_{\mathcal{B},\mathcal{C}} \leftarrow \mathcal{A}^C(\rho)$ using the knowledge of C, and sends the corresponding registers of $\sigma_{\mathcal{B},\mathcal{C}}$ to $\mathcal{B}$ and $\mathcal{C}$ along with the description of C to both $\mathcal{B}$ and $\mathcal{C}$.
- $\mathcal{R}_{\mathcal{B}}$ on receiveing $x_{\mathcal{B}}$ computes $y_{\mathcal{B}} \leftarrow C(x_{\mathcal{B}})$ and runs $b_{\mathcal{B}} \leftarrow \mathcal{B}^{C\setminus\{x_1\}}(\sigma_{\mathcal{B}}, (x_{\mathcal{B}}, y_{\mathcal{B}}))$ using the knowledge of C and x_1, and outputs $b_{\mathcal{B}}$.
- $\mathcal{R}_{\mathcal{C}}$ on input $x_{\mathcal{C}}$ does the symmetrical version of above using $x_{\mathcal{C}}$, C and $\sigma_{\mathcal{C}}$, and outputs $b_{\mathcal{C}}$.

Clearly, the view of $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ as simulated above by $(\mathcal{R}_{\mathcal{A}}, \mathcal{R}_{\mathcal{B}}, \mathcal{R}_{\mathcal{C}})$ is the same as their view in $\mathbf{H}_8$, and the event $b = b_{\mathcal{B}} = b_{\mathcal{C}}$ in the above UPO game exactly corresponds to the event $b = b_{\mathcal{B}} = b_{\mathcal{C}}$ in $\mathbf{H}_8$ which completes the proof of $(\mathcal{D}_{\mathsf{Circ}} \times \mathcal{D}_{\mathbf{X}} \times \mathcal{D}_{\mathbf{X}})$-oracular-pseudorandomness-style copy-protection anti-piracy.

For the "Moreover" part, we note that if we can start from the $(\mathcal{D}_{\mathsf{Circ}} \times \mathcal{D}_{\mathbf{X}} \times \mathcal{D}_{\mathbf{X}})$-oracular-pseudorandomness-style CP+ antipiracy game, then following the same hybrid arguments as above, we will get to the following "+"-version of $\mathbf{H}_8$. The differences between $\mathbf{H}_8$ and $\mathbf{H}_8^+$ are indicated in *blue*.

$\mathbf{H}_8^+$: this hybrid is defined as follows:

- Sample $x_1 \leftarrow \mathcal{D}_{\mathbf{X}}$ and $x_2 \leftarrow \mathcal{D}_{\mathbf{X}}$ independently, and $b_{\mathcal{B}}, b_{\mathcal{C}} \xleftarrow{\$} \{0,1\}$,
- $\mathcal{A}$ gets oracle access to C, and the state $\mathsf{UPO}.\mathcal{O}bf(1^\lambda, \widehat{C})$, where if $b_{\mathcal{B}} = b_{\mathcal{C}} = 0$, $\widehat{C} = C$, $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$, if $b_{\mathcal{B}} = b_{\mathcal{C}} = 1$, $\widehat{C} \leftarrow \mathsf{Program}_{canonical}(C, (x_1, x_2), (y_1, y_2))$, if $b_{\mathcal{B}} = 0, b_{\mathcal{C}} = 1$, $\widehat{C} \leftarrow \mathsf{Program}_{canonical}(C, (x_2), (y_2))$, and if $b_{\mathcal{B}} = 1, b_{\mathcal{C}} = 0$, $\widehat{C} \leftarrow \mathsf{Program}_{canonical}(C, (x_1), (y_1))$, where $y_1, y_2 \xleftarrow{\$} \mathbf{Y}$, $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$.
- After the splitting experiment, along with oracle access to $C \setminus \{x_1\}$ and $C \setminus \{x_2\}$ respectively, $\mathcal{B}$ and $\mathcal{C}$ receive $(x_1, C(x_1))$ and $(x_2, C(x_2))$ respectively.
- $(\mathcal{B}, \mathcal{C})$ output $(b'_{\mathcal{B}}, b'_{\mathcal{C}})$.

The output of the hybrid is $b_{\mathcal{B}}, b_C, b'_{\mathcal{B}}, b'_C$. Denote the probability that $b'_{\mathcal{B}} \oplus b'_C = b_{\mathcal{B}} \oplus b_C$ be p_8^+.

By the same hybrid arguments as between $\mathbf{H}_0$ and $\mathbf{H}_8$, we get that the success probability in the original $(\mathcal{D}_{\mathsf{Circ}} \times \mathcal{D}_{\mathbf{X}} \times \mathcal{D}_{\mathbf{X}})$-oracular-pseudorandomness-style CP+ antipiracy game is negligibly close to p_8^+. Moreover, it is easy to check that the success probability of the same reduction adversary $(\mathcal{R}_{\mathcal{A}}, \mathcal{R}_{\mathcal{B}}, \mathcal{R}_C)$ as described above, but in the $\mathcal{D}_{\mathbf{X}} \times \mathcal{D}_{\mathbf{X}}$-generalized-UPO$^+$antipiracy game, is also p_8^+. Hence, by the $\mathcal{D}_{\mathbf{X}} \times \mathcal{D}_{\mathbf{X}}$-generalized-UPO$^+$antipiracy of UPO, the success probability in the original copy-protection+ antipiracy game is at most $1/2 + \mathsf{negl}(\lambda)$ for some negligible function $\mathsf{negl}(\cdot)$. □

Combining Theorem 7.4 with Theorems 3.17 and 5.1, we get the following corollary.

Corollary 7.7 (Copy-protection with pseudorandomness-style CP+ anti-piracy from concrete assumptions). *Let $\boldsymbol{X} = \{0,1\}^n$ where $n = n(\lambda)$ is a polynomial in λ, and let $\mathcal{D}_{\boldsymbol{X}}$ be a high min-entropy distribution on the input space $\boldsymbol{X}$ and $(\mathsf{Circ}, \mathsf{Puncture})$ be a puncturable circuit class satisfying 2-point m-bit $(\mathcal{D}_{\boldsymbol{X}}, \mathcal{D}_{\mathsf{Circ}})$-pseudorandomness-style puncturing security (Definition 7.1). Then, assuming the existence of polynomially secure iO and the LWE assumption, there exists a copy-protection scheme for Circ that satisfies $(\mathcal{D}_{\mathsf{Circ}} \times \mathcal{D}_{\boldsymbol{X}} \times \mathcal{D}_{\boldsymbol{X}})$-oracular-pseudorandomness-style CP+ anti-piracy (Definition 3.4). Moreover, if $\mathcal{D}_{\boldsymbol{X}} = \mathsf{Uniform}_{\boldsymbol{X}}$, then we can replace the LWE assumption with the existence of OWFs.*

8 Oracular Unpredictability-Style Copy-Protection

We begin with recalling the copy-protection construction [AB24] from UPO for puncturable circuit classes that satisfy 2-point m-bit unpredictability-style puncturing security[27], for $m \in \omega(\log(\lambda))$.

Theorem 8.1 (Adapted from [AB24, Theorem 56]). *Let $\mathcal{D}_{\boldsymbol{X}}$ be a high min-entropy distribution on the input space $\boldsymbol{X}$ and $(\mathsf{Circ}, \mathsf{Puncture})$ be a puncturable circuit class satisfying 2-point m-bit $(\mathcal{D}_{\boldsymbol{X}}, \mathcal{D}_{\mathsf{Circ}})$-unpredictability-style puncturing security, where $m \in \omega(\log \lambda)$ (see Definition 7.1). Then any UPO scheme $\mathsf{UPO} = (\mathcal{O}bf, \mathcal{E}val)$ that satisfies $(\mathcal{D}_{\boldsymbol{X}} \times \mathcal{D}_{\boldsymbol{X}})$ UPO anti-piracy for Circ with respect to $\mathsf{Puncture}$ (see Definition 3.12), is also a $(\mathcal{D}_{\mathsf{Circ}} \times \mathcal{D}_{\boldsymbol{X}} \times \mathcal{D}_{\boldsymbol{X}})$-unpredictability-style copy-protection scheme for Circ (see Definition 3.1) with $\mathcal{C}opy\mathcal{P}rotect = \mathsf{UPO}.\mathcal{O}bf$ and $\mathcal{E}val = \mathsf{UPO}.\mathcal{E}val$.*[28]

Combining with Theorems 3.17 and 5.1, we get the following immediate corollary.

Corollary 8.2. *Let $\boldsymbol{X} = \{0,1\}^n$ where $n = n(\lambda)$ is a polynomial in λ, and let $\mathcal{D}_{\boldsymbol{X}}$ be a high min-entropy distribution on the input space $\boldsymbol{X}$ and $(\mathsf{Circ}, \mathsf{Puncture})$ be a puncturable circuit class satisfying 2-point m-bit $(\mathcal{D}_{\boldsymbol{X}}, \mathcal{D}_{\mathsf{Circ}})$-unpredictability-style puncturing security (see Definition 7.1). Then, assuming the existence of polynomially secure iO and the LWE assumption, there exists a copy-protection scheme for Circ satisfying $(\mathcal{D}_{\mathsf{Circ}} \times \mathcal{D}_{\boldsymbol{X}} \times \mathcal{D}_{\boldsymbol{X}})$-unpredictability-style CP anti-piracy (see Definition 3.1). Moreover, if $\mathcal{D}_{\boldsymbol{X}} = \mathsf{Uniform}_{\boldsymbol{X}}$, then we can replace the LWE assumption with the existence of OWFs.*

Remark 8.3 (Upgrade to oracular CP anti-piracy). The anti-piracy notions for copy-protection considered in [AB24] are non-oracular, which is reflected in the copy-protection anti-piracy notions yielded in Theorem 8.1 and Corollary 8.2. However, by the same hybrid arguments as in the proof of Theorem 7.4 ($\mathbf{H}_4$ to $\mathbf{H}_7$), the copy-protection anti-piracy guarantees provided in Theorem 8.1 and Corollary 8.2 can be upgraded to satisfy oracular unpredictability-style CP anti-piracy.

8.1 Progmaskable circuit classes

In [AB24], the authors introduced the notion of Preimage-sampleability for evasive functions that captures a wide variety of evasive circuit classes, including point functions and k-point functions. In this section, we will study

[27] In [AB24], the authors only considered a single definition of puncturing security, that coincides with 2-point m-bit unpredictability-style puncturing security that we defined in Definition 7.1

[28] In [AB24], the authors only considered the uniformly random challenge distribution, i.e., $\mathcal{D}_{\mathbf{X}} = \mathsf{Uniform}_{\mathbf{X}}$, but it can be easily checked that their proof also generalizes to any high min-entropy challenge distribution $\mathcal{D}_{\mathbf{X}}$, as long as the puncturing security and UPO anti-piracy also holds with respect to $\mathcal{D}_{\mathbf{X}}$.

the notion of *Progmaskability*, which can be intuitively seen as a generalization of Preimage-sampleability beyond evasive circuit classes and beyond boolean output functions, such that it captures other classes of circuit classes, such as puncturable secure circuit classes (see Theorem 8.18). Moreover, we simplify the need for an auxiliary circuit class in Preimage-sampleability by demanding the circuit class itself to be the auxiliary circuit class. Hence, formally, *Progmaskability* is a generalization of a subclass of Preimage-sampleability where the auxiliary circuit class and the distribution on it are respectively the same as the circuit class and the distribution on it. We note that interesting instantiations of Preimage-sampleable evasive circuit classes, such as k-point functions, satisfy this subclass of Preimage-sampleability. Finally, note that Preimage-sampleability is defined with respect to a single input point. We also generalize this aspect to consider multiple input points, and define ℓ-point *Progmaskability* for any $\ell \in \mathrm{poly}(\lambda)$. These generalizations from Preimage-sampleability to *Progmaskability* are formally proved in Theorem 8.6.
For a circuit $C : \mathbf{X} \to \mathbf{Y}$, for every $y \in \mathbf{Y}$, we define $C^{-1}(y) = \{x \in \mathbf{X} \ : \ C(x) = y\}$.

Definition 8.4. *We say that a circuit class* Circ *is ℓ-points $(\mathcal{D}_{\boldsymbol{X}^\ell}, \widetilde{\mathcal{D}}_{\boldsymbol{X}}, \mathcal{D}_{\boldsymbol{Y}}, \mathcal{D}_{\mathsf{Circ}})$-Progmaskable if there exists an efficient algorithm* $\mathsf{Program}$ *and an efficiently sampleable distribution $\widetilde{\mathcal{D}}_{\boldsymbol{X}}$ such that the following two distributions are computationally indistinguishable:*

- $\mathcal{D}_0(1^\lambda)$*: outputs* $(C, x_1, \ldots, x_\ell)$*, where:*
  - $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$ [29]*,*
  - $x_1, \ldots, x_\ell \leftarrow \mathcal{D}_{\boldsymbol{X}^\ell}(C)$*.*
- $\mathcal{D}_1(1^\lambda)$*: outputs* $(\widehat{C}, x_1, \ldots, x_\ell)$*, where:*
  - $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$*,*
  - $x_1 \leftarrow \widetilde{\mathcal{D}}_{\boldsymbol{X}}, \ldots, x_\ell \leftarrow \widetilde{\mathcal{D}}_{\boldsymbol{X}}$*,*
  - $y_1, \ldots, y_\ell \leftarrow \mathcal{D}_{\boldsymbol{Y}}(C, x_1, \ldots, x_\ell)$ *and,*
  - $\widehat{C} \leftarrow \mathsf{Program}(C, (x_1, \ldots, x_\ell), (y_1, \ldots, y_\ell))$*, where:*

$$\widehat{C}(x') = \begin{cases} C(x') & \text{if } x' \notin \{x_1, \ldots, x_\ell\}, \\ y_i & \text{if } x' = x_i \forall i \in [\ell]. \end{cases}$$

When the context is clear, we will omit $\mathcal{D}_{\mathsf{Circ}}$. We emphasize that both the distributions $\mathcal{D}_{\mathbf{X}^\ell}$ and $\mathcal{D}_{\mathbf{Y}}$ do depend on the circuit C. Additionally, the distribution $\mathcal{D}_{\mathbf{Y}}$ also depends on the input x. For non-triviality we only consider $\mathcal{D}_{\mathbf{Y}}(C, x_1, x_2, \ldots, x_\ell)$ that is different from the trivial distribution that outputs $C(x_1), \ldots, C(x_\ell)$.

A special case of the above definition is when the sampling procedure $\mathcal{D}_{\mathbf{Y}}(C, x)$ is defined as follows:

Definition 8.5 (Special case (half-correct distribution).). *A distribution $\mathcal{D}_{\boldsymbol{Y}}(C, x)$ is called a half-correct distribution or simply $\mathcal{D}^{\mathsf{half\text{-}correct}}_{\boldsymbol{Y}}$ if $\mathcal{D}_{\boldsymbol{Y}}(C, x)$ is sampled as follows:*

- *With probability* 0.5*, output $y_i = C(x_i)$ for every $i \in [\ell]$ and,*
- *With probability* 0.5*, output $\{y_1, \ldots, y_\ell\}$, where $y_i \xleftarrow{\$} \boldsymbol{Y} \setminus \{C(x)\}$ for every $i \in [\ell]$.*

Next, we discuss some interesting instantiations of *Progmaskable* circuit classes.

Theorem 8.6. *Any* $\mathsf{Preimage\text{-}sampleable}$ *evasive circuit class* $(\mathsf{Circ}, \mathcal{D}_{\mathsf{Circ}})$ *with respect to itself (i.e., the* $\mathsf{Preimage\text{-}sampleability}$ *holds with respect to* $(\mathsf{Circ}, \mathcal{D}_{\mathsf{Circ}})$ *as the auxilliary circuit class and distribution) and puncturing algorithm* $\mathsf{Program}$ *is* 1*-point* 1*-bit $(\mathcal{D}^{Evasive}_{\boldsymbol{X}}, \mathsf{Uniform}_{\boldsymbol{X}}, \mathcal{D}^{\mathsf{half\text{-}correct}}_{\boldsymbol{X}}, \mathcal{D}_{\mathsf{Circ}})$-Progmaskable with respect to* $\widehat{\mathsf{Program}}$*, where* $\widehat{\mathsf{Program}}$ *on input $(C, (x_1, \ldots, x_\ell), (y_1, \ldots, y_\ell))$ checks if $y_i = C(x_i)$ for every $i \in [\ell]$ and if so outputs C, else outputs $\mathsf{Program}(C, (x_1, \ldots, x_\ell), (y_1, \ldots, y_\ell))$, and $\mathcal{D}^{Evasive}_{\boldsymbol{X}}$ is defined as:*

1. *With probability* $1/2$*, $\mathcal{D}^{Evasive}_{\boldsymbol{X}}(C)$ outputs $x \xleftarrow{\$} S(C)$, where $S(C)$ is the set of all preimages of* 1 *under C.*

[29] WLOG, we view $\mathcal{D}_{\mathsf{Circ}}$ as outputting a key or index that corresponds to the circuit C in the circuit class Circ.

2. *With probability* $1/2$, $\mathcal{D}_{\boldsymbol{X}}^{Evasive}(C)$ *outputs* $x \xleftarrow{\$} \boldsymbol{X}$.

Proof of Theorem 8.6. By definition of $\widehat{\mathsf{Program}}$ and $\mathcal{D}_{\mathbf{Y}}^{\mathsf{half\text{-}correct}}$ (see Definition 8.5), we can decompose the distribution $\mathcal{D}_1$ in Definition 8.4 with respect to $(\mathcal{D}_{\mathbf{X}}^{Evasive}, \mathsf{Uniform}_{\mathbf{X}}, \mathcal{D}_{\mathbf{X}}^{\mathsf{half\text{-}correct}}, \mathcal{D}_{\mathsf{Circ}})$ as:

1. With probability $1/2$, outputs C, x where $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$, and $x \xleftarrow{\$} \mathbf{X}$.
2. With probability $1/2$, outputs $\widehat{C}, x$ where $\widehat{C} \leftarrow \mathsf{Program}(C, x, 1-y)$, $y \leftarrow C(x)$, $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$, and $x \xleftarrow{\$} \mathbf{X}$.

We call the above sampler $\mathbf{H}_0$, and consider the following hybrid samplers.
$\mathbf{H}_1$:

1. With probability $1/2$, outputs C, x where $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$, and $x \xleftarrow{\$} \mathbf{X}$.
2. With probability $1/2$, outputs $\widehat{C}, x$ where $\widehat{C} \leftarrow \mathsf{Program}(C, x, 1)$, $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$, and $x \xleftarrow{\$} \mathbf{X}$.

Since Circ is evasive, $C(x) = 0$ for $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$, and $x \xleftarrow{\$} \mathbf{X}$ with overwhelming probability. Hence, $\mathbf{H}_0$, i.e., $\mathcal{D}_1$ with respect to $(\mathcal{D}_{\mathbf{X}}^{Evasive}, \mathsf{Uniform}_{\mathbf{X}}, \mathcal{D}_{\mathbf{X}}^{\mathsf{half\text{-}correct}}, \mathcal{D}_{\mathsf{Circ}})$ is statitically indistinguihsable from $\mathbf{H}_1$, i.e., the outputs of $\mathbf{H}_0$ and $\mathbf{H}_1$ are statistically indistinguishable.
$\mathbf{H}_2$:

1. With probability $1/2$, outputs C, x where $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$, and $x \xleftarrow{\$} \mathbf{X}$.
2. With probability $1/2$, outputs (C, x) where $x \leftarrow S(C)$, $C \leftarrow \mathcal{D}_{\mathsf{Circ}}$, and $S(C)$ denotes the preimage set of 1 under C.

By the Preimage-sampleability property, the conditional distribution in Item 2 in $\mathbf{H}_1$, is computationally indistinguishable from Item 2 in $\mathbf{H}_2$. Hence, the outputs of $\mathbf{H}_1$ and $\mathbf{H}_2$ must be computationally indistinguishable.

Note that by definition of $\mathcal{D}_0$ in Definition 8.4 with respect to $(\mathcal{D}_{\mathbf{X}}^{Evasive}, \mathsf{Uniform}_{\mathbf{X}}, \mathcal{D}_{\mathbf{X}}^{\mathsf{half\text{-}correct}}, \mathcal{D}_{\mathsf{Circ}})$, $\mathbf{H}_2$ is the same as $\mathcal{D}_0$. Hence, we conclude that the output distribution of $\mathcal{D}_1$, (which is the same as that of $\mathbf{H}_0$) is computationally indistinguishable from that of $\mathcal{D}_0$ (which is the same as that of $\mathbf{H}_2$), which concludes the proof of *Progmaskability* for Circ. □

8.1.1 k-Point Functions for k greater than 1

Consider the following class of functions $\mathsf{Func}^k = \{f_S : \{0,1\}^n \to \{0,1\} \ : \ S \subseteq \{0,1\}^n, |S| = k\}$, where f_S is defined as follows:

$$f_S(x) = \begin{cases} 1 & \text{if } x \in S, \\ 0 & \text{otherwise.} \end{cases}$$

There is an efficiently computable circuit class Circ^k that implements Func^k as long as k is polynomial in n. That is, for every f_S, there is a unique circuit C_S whose implementation is specified as follows: on input x,

- If $x \in S$, output 1.
- Else, output 0.

Define the sampler for $\mathcal{D}_{\mathsf{Circ}^k}$ as follows: first sample a uniformly random set $S \subseteq \{0,1\}^n$ such that $|S| = k$. Output S.

Definition 8.7. *We define the distribution* $\mathcal{D}_{\boldsymbol{X}^\ell}^{Evasive}(S)$ *which samples as follows.*

- *With probability* $1/2$, *sample* $x_1, \ldots, x_\ell \xleftarrow{\$} \binom{S}{\ell}$, *see Section 2.*

- *Otherwise, sample* $x_i \xleftarrow{\$} (\{0,1\}^n \backslash S)$ *independently for each* $i \in [\ell]$.

We consider $\mathcal{D}_{\mathbf{Y}}^{\mathsf{half\text{-}correct}}$ to be the half-correct distribution (see Definition 8.5), and $\mathsf{Uniform}_{\mathbf{X}}$ to be the uniformly random distribution on $\mathbf{X}$.

Remark 8.8 (Viewing obfuscated keyed-circuit classes). For a circuit class Circ, we consider the obfuscated circuit class $i\mathcal{O}(\mathsf{Circ}) := \{i\mathcal{O}(1^\lambda, C) \ : \ C \in \mathsf{Circ}\}$, where we view $i\mathcal{O}(\mathsf{Circ})$ as indexed with $\{C, r\}$ where C is an index in Circ and r is the randomness required to run the $i\mathcal{O}$ compiler.

Theorem 8.9. *Let* $k \geq \ell$*, where* $k := k(\lambda)$*,* $\ell := \ell(\lambda)$ *are both polynomials in the security parameter* λ*, and let* $\boldsymbol{X} := \boldsymbol{X}(\lambda)$ *be a super-polynomial-sized domain (in the security parameter* λ*). Let* $\mathsf{Circ}^k := \{C_S\}_{S \in \binom{\boldsymbol{X}}{[k]}}$ *be a set of circuits implementing* Func^k*, the set of all* k*-point functions on* $\boldsymbol{X}$*, i.e., Boolean functions on* $\boldsymbol{X}$ *that have exactly* k*-preimages of* 1*, where for each* $S \subset \boldsymbol{X}$*,* $|S| = k$*,* C_S *denotes the* k*-point function circuit with* S *as the preimage of* 1*.*

Assuming post-quantum iO exists and post-quantum one-way functions exists, $i\mathcal{O}(\mathsf{Circ}^k)$ *satisfies* $(\widetilde{\mathcal{D}^{Evasive}}_{\boldsymbol{X}^\ell}, \mathsf{Uniform}_{\boldsymbol{X}}, \mathcal{D}_{\boldsymbol{Y}}^{\mathsf{half\text{-}correct}}, \widetilde{\mathcal{D}}$ *Progmaskability (see Definition 8.5) with respect to an efficient algorithm* $\mathsf{Program}$ *defined as follows: on input* $(S, (x_1, x_2, \ldots, x_\ell), (y_1, y_2, \ldots, y_\ell))$*, it outputs* $\widehat{C} \leftarrow i\mathcal{O}(1^\lambda, C')$ *where* $C'(x')$ *is defined as follows:*

- *If* $x' = x_i$ *for some* i*, output* y_i*,*
- *Else, output* $C_S(x')$*,*

and where the distribution $\widetilde{\mathcal{D}}_{\mathsf{Circ}^k}$ *over* k*-point functions is defined as:*

- *Sample a uniform random subset* S *of* $\boldsymbol{X}$ *of size* k*.*
- *Sample randomness* r *for* $i\mathcal{O}$ *and output* (S, r) *which corresponds to the circuit* $i\mathcal{O}(1^\lambda, C_S; r)$*.*

and $\widetilde{\mathcal{D}^{Evasive}}_{\boldsymbol{X}^\ell}(S, r) := \mathcal{D}^{Evasive}_{\boldsymbol{X}^\ell}(S)$ *where* $\mathcal{D}^{Evasive}_{\boldsymbol{X}^\ell}$ *is as defined in Definition 8.7,*

Proof of Theorem 8.9. We first prove the following claim.

Claim 8.10. Let $D := D(\lambda)$ be a super-polynomial-sized set in λ. Assuming the existence of a keyed injective $\mathcal{U}$-one-way function IOWF (see Definition 2.5) with domain D, which can be constructed from post-quantum iO and one-way functions [BPW16], the following two distributions are indistinguishable.

- The membership circuit $i\mathcal{O}(C_S)$ where S is a uniform random subset of D of size $k+1$.
- The membership circuit $i\mathcal{O}(C_S)$ where S is a uniform random subset of D of size k.

Proof. We describe the hybrids below.
$\mathbf{H}_0$: $i\mathcal{O}(C_S)$ where S is a uniform random subset of D of size $k+1$. This is the first distribution.
$\mathbf{H}_1$: $i\mathcal{O}(\widehat{C})$ where $\widehat{C}$ is sampled as follows:

1. Sample a uniform subset S of D of size k.
2. Sample a uniform random point $x^* \in D \backslash S$.
3. $\widehat{C} \leftarrow \mathsf{Program}(S, x^*, 1)$.

Note that the distribution of $\widehat{C}$ is exactly the distribution of C_S in $\mathbf{H}_0$. By $i\mathcal{O}$ security, $\mathbf{H}_0$ and $\mathbf{H}_1$ are computationally indistinguishable.
$\mathbf{H}_2$: $i\mathcal{O}(\widehat{C})$ where $\widehat{C}$ is sampled as follows:

1. Sample a uniform subset S of D of size k.
2. Sample a uniform random point $x^* \xleftarrow{\$} D$.

3. $\widehat{C} \leftarrow \mathsf{Program}(S, x^*, 1)$.

$\mathbf{H}_1$ and $\mathbf{H}_2$ are negligibly close because $\frac{k}{|D|}$ is negligible.
$\mathbf{H}_3$: $i\mathcal{O}(\widehat{C})$ where $\widehat{C}$ is sampled as follows:

1. Sample a uniform subset S of D of size k.
2. Sample a uniform random point $x^* \xleftarrow{\$} D$.
3. Sample $\mathsf{F} \leftarrow \mathsf{IOWF.Gen}(1^\lambda)$.
4. $\widehat{C}$ be the following circuit:

$$\widehat{C}(x) = \begin{cases} 1 & \mathsf{F}(\mathsf{x}) = \mathsf{y}^*, \\ C_S(x) & \text{otherwise,} \end{cases}$$

   where the hardcoded value $y^* = \mathsf{F}(x^*)$.

Note that except with negligible probability over the generation of F, the functionalities of $\widehat{C}(x)$ in $\mathbf{H}_2$ and $\mathbf{H}_3$ are the same since except with negligible probability over the generation of F, F is injective. Therefore, by $i\mathcal{O}$ security, $\mathbf{H}_2$ and $\mathbf{H}_3$ are computationally indistinguishable.
$\mathbf{H}_4$: $i\mathcal{O}(\widehat{C})$ where $\widehat{C}$ is sampled as follows:

1. Sample a uniform subset S of D of size k.
2. Sample a uniform random point $x^* \xleftarrow{\$} D$.
3. $\widehat{C}$ is the circuit C_S.

Note that the circuits under $i\mathcal{O}$ in $\mathbf{H}_3$ and $\mathbf{H}_4$ differ only at the point x^*. Since every indistinguishability obfuscator is also a single-point differing input obfuscator ($di\mathcal{O}$) (see Ref. [BCP14]), by the single-point $di\mathcal{O}$ security of $i\mathcal{O}$, if $\mathbf{H}_3$ and $\mathbf{H}_4$ are computationally indistinguishable, then there exists an algorithm that, given the circuits $\widehat{C}$ from $\mathbf{H}_3$ and $\mathbf{H}_4$ can compute x^*. However, note that $\widehat{C}$ in both hybrids can be computed given just y^* and F, and hence we can construct an inverter for the keyed one-way function IOWF. Hence, if $\mathbf{H}_3$ and $\mathbf{H}_4$ are computationally indistinguishable, then we can violate the one-wayness of IOWF. Therefore, $\mathbf{H}_3$ and $\mathbf{H}_4$ must be computationally indistinguishable. □

Now we prove Theorem 8.9 using Claim 8.10.We describe the hybrids below.
$\mathbf{H}_0$: This is the hybrid that describes $\mathcal{D}_1$ in the Progmaskability definition.

- Sample $S \subseteq \mathbf{X}$ uniformly at random so that $|S| = k$.
- Sample $x_1^* \leftarrow \mathsf{Uniform}_{\mathbf{X}}, \ldots, x_\ell^* \leftarrow \mathsf{Uniform}_{\mathbf{X}}$.
- Sample $b \xleftarrow{\$} \{0,1\}$.
- Output $(i\mathcal{O}(\mathsf{Program}(C_S, (x_1^*, x_2^*, \ldots x_\ell^*), (C_S(x_1^*) \oplus b, C_S(x_2^*) \oplus b, \ldots, C_S(x_\ell^*) \oplus b)), (x_1^*, x_2^*, \ldots x_\ell^*))$.

$\mathbf{H}_1$: Similar to $\mathbf{H}_0$ but we sample $x_1^*, \ldots, x_\ell^*$ uniformly at random conditioned on them being distinct.

- Sample $\{x_1^*, x_2^*, \ldots, x_\ell^*\} \xleftarrow{\$} \binom{\mathbf{X}}{\ell}$.
- Sample $b \xleftarrow{\$} \{0,1\}$.
- Sample $S \subseteq \mathbf{X} \backslash \{x_1^*, x_2^*, \ldots x_\ell^*\}$ uniformly at random so that $|S| = k$.
- Output $(i\mathcal{O}(\mathsf{Program}(C_S, (x_1^*, x_2^*, \ldots x_\ell^*), (b, b, \ldots, b)), (x_1^*, x_2^*, \ldots x_\ell^*))$.

Note that $\mathbf{X} = \{0,1\}^n$, and since $k/2^n$ and $\ell/2^n$ are negligible, with $1 - \mathsf{negl}$ probability in $\mathbf{H}_0$, we have all x_i^* are all distinct and none of them are in S. Thus $\mathbf{H}_0$ and $\mathbf{H}_1$ are statistically indistinguishable.
$\mathbf{H}_2$: In this hybird we modify the size of S.

- Sample $\{x_1^*, x_2^*, \ldots, x_\ell^*\} \xleftarrow{\$} \binom{\mathbf{X}}{\ell}$.
- Sample $b \xleftarrow{\$} \{0,1\}$.
- If $b = 0$, sample $S' \subseteq \mathbf{X} \backslash \{x_1^*, x_2^*, \ldots x_\ell^*\}$ uniformly at random so that $|S'| = k$. Otherwise, sample $S' \subseteq \mathbf{X} \backslash \{x_1^*, x_2^*, \ldots x_\ell^*\}$ uniformly at random so that $|S'| = k - \ell$.
- Output $(i\mathcal{O}(\mathsf{Program}(C_{S'}, (x_1^*, x_2^*, \ldots x_\ell^*), (b, b, \ldots, b)), (x_1^*, x_2^*, \ldots x_\ell^*))$.

Using Claim 8.10 multiple times with the domain $D = X \backslash \{x_1^*, x_2^*, \ldots x_\ell^*\}$, we see that $\mathbf{H}_1$ is computationally indistinguishable from $\mathbf{H}_2$.
$\mathbf{H}_3$: This is the hybrid that describes $\mathcal{D}_0$ in the Progmaskability definition.

- Sample $S \subseteq \mathbf{X}$ uniformly at random so that $|S| = k$.
- Sample $(x_1^*, x_2^*, \ldots x_\ell^*) \leftarrow \mathcal{D}_{\mathbf{X}^\ell}^{Evasive}(S)$.
- Output $(i\mathcal{O}(C_S), (x_1^*, x_2^*, \ldots x_\ell^*))$.

Note that in $\mathbf{H}_3$, the number of point with value 1 in the final outputted circuit $i\mathcal{O}(\mathsf{Program}(C_{S'}, (x_1^*, x_2^*, \ldots x_\ell^*), (b, b, \ldots, b)))$ is exactly k. These points are S' if $b = 0$ and are the union of S' and $\{x_1^*, \ldots, x_\ell^*\}$ if $b = 1$. Note that the union of S' and $\{x_1^*, \ldots, x_\ell^*\}$ is a uniform subset of X of size k. Thus, by $i\mathcal{O}$ security, this circuit is indistinguishable from $i\mathcal{O}(C_S)$ for a uniform random subset $S \subseteq \mathbf{X}$ of size k. □

8.2 Oracular Unpredictability-Style Copy-Protection for Progmaskable Circuits

Let $\mathbf{X} = \{0,1\}^\lambda$ be the input space.

Theorem 8.11. *Let* $(\mathsf{Circ}, \mathsf{Program})$ *be a* 2*-point* $(\mathcal{D}_{\mathbf{X}^2}, \widetilde{\mathcal{D}}_{\mathbf{X}}, \mathcal{D}_{\mathbf{Y}}, \mathcal{D}_{\mathsf{Circ}})$ *Progmaskable circuit class, such that* $\widetilde{\mathcal{D}}_{\mathbf{X}}$ *has min-entropy at least* λ^ϵ *for some* $\epsilon \geq 0$*, and* $\mathcal{D}_{\mathbf{Y}}$ *is the half-correct distribution (see Definition 8.5). Then, any UPO scheme* $\mathsf{UPO} = (\mathcal{O}bf, \mathcal{E}val)$ *satisfying* $(\widetilde{\mathcal{D}}_{\mathbf{X}} \times \widetilde{\mathcal{D}}_{\mathbf{X}})$*-generalized UPO security for* Circ *with respect to* $\mathsf{Program}_{canonical}$*, as well as iO security, is also a copy-protection scheme* $(\mathcal{C}opyProtect, \mathcal{E}val)$ *for* Circ *with* $\mathcal{C}opyProtect = \mathsf{UPO}.\mathcal{O}bf$ *and* $\mathcal{E}val = \mathsf{UPO}.\mathcal{E}val$*, satisfying* $1/2$*-oracular-*$\mathcal{D}_{\mathbf{X}^2,\mathsf{Circ}}$*-unpredictable-style-CP anti-piracy, where* $\mathcal{D}_{\mathbf{X}^2,\mathsf{Circ}}$ *is defined as:* $C, x_1, x_2 \leftarrow \mathcal{D}_{\mathbf{X}^2,\mathsf{Circ}}(1^\lambda)$*, where* $x_1, x_2 \leftarrow \mathcal{D}_{\mathbf{X}^2}(C)$ *and* $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$.

Proof of Theorem 8.11. The correctness of $(\mathcal{C}opyProtect, \mathcal{E}val)$ is immediate from the correctness of UPO. Next, we prove security. We assume that Circ satisfies $(\mathcal{D}_{\mathbf{X}}, \widetilde{\mathcal{D}}_{\mathbf{X}}, \mathcal{D}_{\mathbf{Y}}, \mathcal{D}_{\mathsf{Circ}})$-Progmaskability property, where $\mathcal{D}_{\mathbf{Y}}$ is a half-correct distribution (see Definition 8.5).

We provide the proof of security below. Consider the following hybrids. The changes are highlighted in red.

$\mathbf{H}_0$: this corresponds to the unpredictability-style copy-protection security experiment. Denote the copy-protection adversary to be $(\mathcal{A}, \mathcal{B}, \mathcal{C})$.

- $\mathcal{A}$ gets $\mathsf{UPO}.\mathcal{O}bf(1^\lambda, C)$ and oracle access to C, where $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$.
- Sample $x_1, x_2 \leftarrow \mathcal{D}_{\mathbf{X}^2}(C)$.
- After the splitting experiment, $\mathcal{B}$ receives x_1 and oracle access to $C \setminus \{x_1\}$, and $\mathcal{C}$ receives x_2 and oracle access to $C \setminus \{x_2\}$[30].

[30] Here oracle access to $C \setminus S$ for some set of points S refers to the oracle access to a function that outputs $C(z)$ on all points $z \notin S$ and outputs $\perp$ if $z \in S$.

- $(\mathcal{B}, \mathcal{C})$ output $(y_{\mathcal{B}}, y_{\mathcal{C}})$.

The output of the hybrid is $C, x_1, x_2, y_{\mathcal{B}}, y_{\mathcal{C}}$. Denote the probability that $(y_{\mathcal{B}}, y_{\mathcal{C}}) = (C(x_1), C(x_2))$ to be p_0.

$\mathbf{H}_1$: this hybrid is defined as follows:

- $\mathcal{A}$ gets $\mathsf{UPO}.\mathcal{O}bf(1^\lambda, \widehat{C})$ and oracle access to $\widehat{C}$, where:
  - $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$,
  - $x_1 \leftarrow \widetilde{\mathcal{D}}_{\mathbf{X}}$, $x_2 \leftarrow \widetilde{\mathcal{D}}_{\mathbf{X}}$.
  - $y_1, y_2 \leftarrow \mathcal{D}_{\mathbf{Y}}(C, x_1, x_2)$ and,
  - $\widehat{C} \leftarrow \mathsf{Program}(C, (x_1, x_2), (y_1, y_2))$.
- After the splitting experiment, $\mathcal{B}$ receives x_1 and oracle access to $\widehat{C} \setminus \{x_1\}$, and $\mathcal{C}$ receives x_2 and oracle access to $\widehat{C} \setminus \{x_2\}$.
- $(\mathcal{B}, \mathcal{C})$ output $(y_{\mathcal{B}}, y_{\mathcal{C}})$.

The output of the hybrid is $\widehat{C}, x_1, x_2, y_{\mathcal{B}}, y_{\mathcal{C}}$. Denote the probability that $(y_{\mathcal{B}}, y_{\mathcal{C}}) = (\widehat{C}(x_1), \widehat{C}(x_2))$ to be p_1.

From the Progmaskability property, the outputs of $\mathbf{H}_0$ and $\mathbf{H}_1$ are computationally indistinguishable. Hence,

$$|p_1 - p_0| = \epsilon_1(\lambda), \tag{4}$$

for some negligible function $\epsilon_1(\cdot)$.

$\mathbf{H}_2$: this hybrid is defined as follows:

- $\mathcal{A}$ gets $\mathsf{UPO}.\mathcal{O}bf(1^\lambda, \widehat{C})$ and oracle access to $\widehat{C}$, where:
  - $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$,
  - $x_1 \leftarrow \widetilde{\mathcal{D}}_{\mathbf{X}}$, $x_2 \leftarrow \widetilde{\mathcal{D}}_{\mathbf{X}}$.
  - Sample a bit $b \xleftarrow{\$} \{0,1\}$ uniformly at random, and if $b = 0$ set $\widehat{C} = C$, else if $b = 1$, sample $y_1 \xleftarrow{\$} \mathbf{Y} \setminus \{C(x_1)\}, y_2 \xleftarrow{\$} \mathbf{Y} \setminus \{C(x_2)\}$ and set,
  - $\widehat{C} \leftarrow \mathsf{Program}(C, (x_1, x_2), (y_1, y_2))$.
- After the splitting experiment, $\mathcal{B}$ receives x_1 and oracle access to $\widehat{C} \setminus \{x_1\}$, and $\mathcal{C}$ receives x_2 and oracle access to $\widehat{C} \setminus \{x_2\}$.
- $(\mathcal{B}, \mathcal{C})$ output $(y_{\mathcal{B}}, y_{\mathcal{C}})$.

The output of the hybrid is $\widehat{C}, x_1, x_2, y_{\mathcal{B}}, y_{\mathcal{C}}$. Denote the probability that $(y_{\mathcal{B}}, y_{\mathcal{C}}) = (\widehat{C}(x_1), \widehat{C}(x_2))$ to be p_2.

Note that $\mathbf{H}_2$ is the union of two events $b = 0$ and $b = 1$, each with probability half, where the event $b = 1$ in $\mathbf{H}_2$ is identically distributed as the event $y_{\mathcal{B}} \neq C(x_1) \wedge y_{\mathcal{C}} \neq C(x_2)$ in $\mathbf{H}_1$, which happens with probability $\frac{1}{2}$. On the other hand, the event $b = 0$ in $\mathbf{H}_2$ is computationally indistinguishable from the event $y_{\mathcal{B}} = C(x_1) \wedge y_{\mathcal{C}} = C(x_2)$ in $\mathbf{H}_1$ which happens with probability $\frac{1}{2}$, by the io-security of the underlying UPO, $\mathsf{UPO} = (\mathcal{O}bf, \mathcal{E}val)$. Hence, by the io-security of UPO, $\mathbf{H}_2$ and $\mathbf{H}_1$ are computationally indistinguishable. Therefore,

$$|p_2 - p_1| = \epsilon_2(\lambda), \tag{5}$$

for some negligible function $\epsilon_2(\cdot)$.
$\mathbf{H}_3$: this hybrid is defined as follows:

- $\mathcal{A}$ gets $\mathsf{UPO}.\mathcal{O}bf(1^\lambda, \widehat{C})$ and oracle access to $\widehat{C}$, where:
  - $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$,
  - $x_1 \leftarrow \widetilde{\mathcal{D}}_{\mathbf{X}}$, $x_2 \leftarrow \widetilde{\mathcal{D}}_{\mathbf{X}}$.
  - Sample a bit $b \xleftarrow{\$} \{0,1\}$ uniformly at random, and if $b = 0$ set $\widehat{C} = C$, else if $b = 1$, sample $y_1 \xleftarrow{\$} \mathbf{Y} \setminus \{C(x_1)\}, y_2 \xleftarrow{\$} \mathbf{Y} \setminus \{C(x_2)\}$ and set,
  - $\widehat{C} \leftarrow \mathsf{Program}(C, (x_1, x_2), (y_1, y_2))$.
- After the splitting experiment, $\mathcal{B}$ receives x_1 and oracle access to $C \setminus \{x_1\}$, and $\mathcal{C}$ receives x_2 and oracle access to $\widehat{C} \setminus \{x_2\}$.
- $(\mathcal{B}, \mathcal{C})$ output $(y_{\mathcal{B}}, y_{\mathcal{C}})$.

The output of the hybrid is $\widehat{C}, x_1, x_2, y_{\mathcal{B}}, y_{\mathcal{C}}$. Denote the probability that $(y_{\mathcal{B}}, y_{\mathcal{C}}) = (\widehat{C}(x_1), \widehat{C}(x_2))$ to be p_3.

Since $\widehat{C} \setminus \{x_1\}$ and $C \setminus \{x_1\}$ differ in functionality only at x_2, the challenge point for $\mathcal{C}$, the only difference between $\mathbf{H}_2$ and $\mathbf{H}_3$ is that we change the oracle access to $\mathcal{B}$ by changing the oracle output at x_2.

Let $q_{\mathcal{B}}(\lambda)$ be the number of oracle queries that $\mathcal{B}$ makes for some fixed polynomial $q_{\mathcal{B}}(\lambda)$[31]. In $\mathbf{H}_2$, for each $i \in [q_{\mathcal{B}}]$, let $W^i_{2,\mathcal{B}}$ denote the total weight of i^{th} query on x_2, where the i^{th} query can be written as $\sum_{x,y} \alpha_{x,y,i} |x\rangle_{\mathsf{I}} |y\rangle_{\mathsf{O}}$, for input and output registers I and O respectively. In other words, $W^i_{2,\mathcal{B}} = \sum_{y \in \mathbf{Y}} |\alpha_{x_2,y,i}|^2$. Let $W_{2,\mathcal{B}} = \sum_i W^i_{2,\mathcal{B}}$ be the combined weight of $\mathcal{B}$'s queries on x_2 as query points in $\mathbf{H}_2$. We will show the following claim.

Claim 8.12. $\mathbb{E}[W_{2,\mathcal{B}}]$ is negligible in λ.

Combining Claim 8.12 with the fact that $q_{\mathcal{B}}$ is a polynomial in λ, we conclude that $q_{\mathcal{B}} \cdot \mathbb{E}[W_{4,\mathcal{B}}]$ is also negligible in λ. Hence, by Theorem 2.1, where we treat $T = q_{\mathcal{B}}$, we conclude that changing the oracle output for $\mathcal{B}$ at x_2 in $\mathbf{H}_2$ results in a statistically indistinguishable output distribution, i.e., $\{\widehat{C}, x_1, x_2, y_{\mathcal{B}}, y_{\mathcal{C}}\}$. Since the only difference between $\mathbf{H}_2$ and $\mathbf{H}_3$ is that the oracle output for $\mathcal{B}$ at x_2 is different, we conclude that the output distribution in $\mathbf{H}_2$ is computationally indistinguishable[32] from that of $\mathbf{H}_3$. Hence,

$$|p_3 - p_2| = \epsilon_3(\lambda), \tag{6}$$

for some negligible function $\epsilon_3(\cdot)$.

Next, we prove Claim 8.12 to complete the proof of indistinguishability for $\mathbf{H}_2$ and $\mathbf{H}_3$.

Proof of Claim 8.12. Let $W_{0,\mathcal{B}}$ be the combined weight of $\mathcal{B}$'s queries on x_2 as query points in $\mathbf{H}_0$, i.e., $W_{0,\mathcal{B}} = \sum_{i \in [q_{\mathcal{B}}]} W^i_{0,\mathcal{B}}$ where for each $i \in [q_{\mathcal{B}}]$, $W^i_{\mathcal{B}}$ denotes the total weight of i^{th} query on x_2. In $\mathbf{H}_0$, x_2 can be sampled after $\mathcal{B}$'s queries as x_2 is sampled independently of $\mathcal{B}$'s state and challenge x_1. Therefore, since x_2 is sampled from $\widetilde{\mathcal{D}}_{\mathbf{X}}$, there exists a negliglbe function $\mathsf{negl}(\cdot)$ defined as $\mathsf{negl}(\lambda) := 2^{-\mathsf{min\text{-}entropy}(\widetilde{\mathcal{D}}_{\mathbf{X}}(1^\lambda))}$, such that for each $i \in [q_{\mathcal{B}}]$, $\mathbb{E}[W^i_{0,\mathcal{B}}] = \mathsf{negl}(\lambda)$, i.e., $\mathbb{E}[W_{0,\mathcal{B}}] = q_{\mathcal{B}} \cdot \mathsf{negl}(\lambda)$, which is negligible in the security parameter λ.

Next consider, the following distinguishing attack $(\mathcal{A}, \widehat{\mathcal{B}}, \widehat{\mathcal{C}})$:

1. $\widehat{\mathcal{B}}$ runs $\mathcal{B}$ on the challenge and the state received from $\mathcal{A}$ and performs the oracle queries from $\mathcal{B}$ and feeds the output back to $\mathcal{B}$ up until the j^{th} query, where $j \xleftarrow{\$} [q_{\mathcal{B}}]$ is sampled ahead of time. On the j^{th} query, $\widehat{\mathcal{B}}$ measures the query input register I. Let the measurement outcome be x'. $\widehat{\mathcal{B}}$ outputs x'.
2. $\widehat{\mathcal{C}}$ on receiving a challenge from the challenger, outputs the challenge itself.

[31] WLOG, we can assume that $\mathcal{B}$ makes a fixed polynomial number of queries in all the hybrids.

[32] The reason this indistinguihsability is computational rather than statistical is that Claim 8.12 holds only for computationally bounded $\mathcal{B}$.

For any $h \in \{0,2\}$, let the probability that with adversaries $(\mathcal{A}, \widehat{\mathcal{B}}, \widehat{\mathcal{C}})$ in $\mathbf{H}_h$, $\widehat{\mathcal{B}}$ and $\widehat{\mathcal{C}}$ outputs the same string, be p_h^{eq}.

Since the output distribution of $\mathbf{H}_2$ is computationally indistinguishable from that of $\mathbf{H}_0$, $|p_2^{eq} - p_0^{eq}|$ must be negligible in λ. However, note that by definition of $\widehat{\mathcal{B}}$ and $\widehat{\mathcal{C}}$, they output the same string if and only if x', the measurement outcome of a random query of $\mathcal{B}$, is the same as x_2, the challenge for $\mathcal{C}$. Hence, for any $h \in \{0,2\}$, $p_h^{eq} = \mathbb{E}_{i \xleftarrow{\$} [q_{\mathcal{B}}]} \mathbb{E}[W_{h,\mathcal{B}}^i] = \frac{\sum_{i \in [q_{\mathcal{B}}]} \mathbb{E}[W_{h,\mathcal{B}}^i]}{q_{\mathcal{B}}} = \frac{\mathbb{E}[W_{h,\mathcal{B}}]}{q_{\mathcal{B}}}$. Therefore, by the computational indistinguishability of the output distributions of $\mathbf{H}_0$ and $\mathbf{H}_2$, we conclude that $|\frac{\mathbb{E}[W_{2,\mathcal{B}}]}{q_{\mathcal{B}}} - \frac{\mathbb{E}[W_{0,\mathcal{B}}]}{q_{\mathcal{B}}}|$ is negligible in λ, which implies that $|\mathbb{E}[W_{2,\mathcal{B}}] - \mathbb{E}[W_{0,\mathcal{B}}]|$ must be negligible in λ, because $q_{\mathcal{B}}$ is polynomial in λ. Moreover since $\mathbb{E}[W_{0,\mathcal{B}}]$ is negligible as concluded above, $\mathbb{E}[W_{2,\mathcal{B}}]$ must be negligible in λ. □

$\mathbf{H}_4$: this hybrid is defined as follows:

- $\mathcal{A}$ gets $\mathsf{UPO}.\mathcal{O}bf(1^\lambda, \widehat{C})$ and oracle access to $\widehat{C}$, where:
  - $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$,
  - $x_1 \leftarrow \widetilde{\mathcal{D}}_{\mathbf{X}}, x_2 \leftarrow \widetilde{\mathcal{D}}_{\mathbf{X}}$.
  - Sample a bit $b \xleftarrow{\$} \{0,1\}$ uniformly at random, and if $b = 0$ set $\widehat{C} = C$, else if $b = 1$, sample $y_1 \xleftarrow{\$} \mathbf{Y} \setminus \{C(x_1)\}, y_2 \xleftarrow{\$} \mathbf{Y} \setminus \{C(x_2)\}$ and set,
  - $\widehat{C} \leftarrow \mathsf{Program}(C, (x_1, x_2), (y_1, y_2))$.
- After the splitting experiment, $\mathcal{B}$ receives x_1 and oracle access to $C \setminus \{x_1\}$, and $\mathcal{C}$ receives x_2 and oracle access to $C \setminus \{x_2\}$.
- $(\mathcal{B}, \mathcal{C})$ output $(y_{\mathcal{B}}, y_{\mathcal{C}})$.

The output of the hybrid is $\widehat{C}, x_1, x_2, y_{\mathcal{B}}, y_{\mathcal{C}}$. Denote the probability that $(y_{\mathcal{B}}, y_{\mathcal{C}}) = (\widehat{C}(x_1), \widehat{C}(x_2))$ to be p_4.

Since $\widehat{C} \setminus \{x_2\}$ and $C \setminus \{x_2\}$ differ in functionality only at x_1, the challenge point for $\mathcal{B}$, the only difference between $\mathbf{H}_3$ and $\mathbf{H}_4$ is that we change the oracle access to $\mathcal{C}$ by changing the oracle output at x_1.

By analogous arguments as in the proof of indistinguishability of $\mathbf{H}_2$ and $\mathbf{H}_3$ but with the roles of $\mathcal{B}$ and $\mathcal{C}$ switched, we can conclude that the outputs of $\mathbf{H}_3$ and $\mathbf{H}_4$ are computationally indistinguishable and hence,

$$|p_4 - p_3| = \epsilon_4(\lambda), \tag{7}$$

for some negligible function $\epsilon_4(\cdot)$.

$\mathbf{H}_5$: this hybrid is defined as follows:

- $\mathcal{A}$ gets $\mathsf{UPO}.\mathcal{O}bf(1^\lambda, C)$ and oracle access to C, where:
  - $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$,
  - $x_1 \leftarrow \widetilde{\mathcal{D}}_{\mathbf{X}}, x_2 \leftarrow \widetilde{\mathcal{D}}_{\mathbf{X}}$.
  - Sample a bit $b \xleftarrow{\$} \{0,1\}$ uniformly at random, and if $b = 0$ set $\widehat{C} = C$, else if $b = 1$, sample $y_1 \xleftarrow{\$} \mathbf{Y} \setminus \{C(x_1)\}, y_2 \xleftarrow{\$} \mathbf{Y} \setminus \{C(x_2)\}$ and set,
  - $\widehat{C} \leftarrow \mathsf{Program}(C, (x_1, x_2), (y_1, y_2))$.
- After the splitting experiment, $\mathcal{B}$ receives x_1 and oracle access to $C \setminus \{x_1\}$, and $\mathcal{C}$ receives x_2 and oracle access to $C \setminus \{x_2\}$.
- $(\mathcal{B}, \mathcal{C})$ output $(y_{\mathcal{B}}, y_{\mathcal{C}})$.

The output of the hybrid is $\widehat{C}, x_1, x_2, y_{\mathcal{B}}, y_{\mathcal{C}}$. Denote the probability that $(y_{\mathcal{B}}, y_{\mathcal{C}}) = (\widehat{C}(x_1), \widehat{C}(x_2))$ to be p_5.

Since $\widehat{C}$ and C differ in functionality only at x_1 and x_2, the respective challenge points for $\mathcal{B}$ and $\mathcal{C}$, the only difference between $\mathbf{H}_4$ and $\mathbf{H}_5$ is that we change the oracle access to $\mathcal{A}$ by changing the oracle output at x_1 and x_2.

Let $q_{\mathcal{A}}(\lambda)$ be the number of oracle queries that $\mathcal{B}$ makes for some fixed polynomial $q_{\mathcal{A}}(\lambda)$. In $\mathbf{H}_4$, for each $i \in [q_{\mathcal{A}}]$, let $W^i_{4,\mathcal{A}}$ denote the total weight of i^{th} query on x_1 or x_2, where the i^{th} query can be written as $\sum_{x,y} \alpha_{x,y,i} |x\rangle_{\mathsf{I}} |y\rangle_{\mathsf{O}}$, for input and output registers I and O respectively. In other words, $W^i_{4,\mathcal{A}} = \sum_{x \in \{x_1,x_2\}, y \in \mathbf{Y}} |\alpha_{x,y,i}|^2$. Let $W_{4,\mathcal{A}} = \sum_i W^i_{2,\mathcal{A}}$ be the combined weight of $\mathcal{A}$'s queries on x_1 or x_2 as query points in $\mathbf{H}_4$. We will show the following claim.

Claim 8.13. $\mathbb{E}[W_{4,\mathcal{A}}]$ is negligible in λ.

Combining Claim 8.13 with the fact that $q_{\mathcal{A}}$ is a polynomial in λ, we conclude that $q_{\mathcal{A}} \cdot \mathbb{E}[W_{4,\mathcal{A}}]$ is also negligible in λ. Hence, by Theorem 2.1, where we treat $T = q_{\mathcal{A}}$, we conclude that changing the oracle output for $\mathcal{A}$ at x_1 and x_2, in $\mathbf{H}_4$ results in a statistically indistinguishable output distribution, i.e., $\{\widehat{C}, x_1, x_2, y_{\mathcal{B}}, y_{\mathcal{C}}\}$. Since the only difference between $\mathbf{H}_4$ and $\mathbf{H}_5$ is that the oracle outputs for $\mathcal{A}$ at x_1 and x_2, are different, we conclude that the output distribution in $\mathbf{H}_4$ is computationally indistinguishable[33] from that of $\mathbf{H}_5$. Hence,

$$|p_5 - p_4| = \epsilon_5(\lambda), \tag{8}$$

for some negligible function $\epsilon_5(\cdot)$.

Next, we prove Claim 8.13 to complete the proof of indistinguishability for $\mathbf{H}_4$ and $\mathbf{H}_5$.

Proof of Claim 8.13. Let $W_{0,\mathcal{A}}$ be the combined weight of $\mathcal{A}$'s queries on x_1 or x_2, the challenge points for $\mathcal{B}$ and $\mathcal{C}$, as query points in $\mathbf{H}_0$, i.e., $W_{0,\mathcal{A}} = \sum_{i \in [q_{\mathcal{A}}]} W^i_{0,\mathcal{A}}$ where for each $i \in [q_{\mathcal{A}}]$, $W^i_{\mathcal{A}}$ denotes the total weight of i^{th} query on x_1 or x_2. In $\mathbf{H}_0$, x_1 and x_2 can be sampled after $\mathcal{A}$'s queries as x_1, x_2 are sampled independent of $\mathcal{A}$'s input state. Therefore, since x_1, x_2 are sampled from $\widetilde{\mathcal{D}}_{\mathbf{X}}$, there exists a negliglbe function $\mathsf{negl}(\cdot)$ defined as $\mathsf{negl}(\lambda) := 2^{-\mathsf{min\text{-}entropy}(\widetilde{\mathcal{D}}_{\mathbf{X}})}$, such that for each $i \in [q_{\mathcal{A}}]$, $\mathbb{E}[W^i_{0,\mathcal{A}}] \leq 2 \cdot \mathsf{negl}(\lambda)$, i.e., $\mathbb{E}[W_{0,\mathcal{A}}] \leq 2q_{\mathcal{A}} \cdot \mathsf{negl}(\lambda)$, which is negligible in the security parameter λ.

Next consider, the following distinguishing attack $(\widehat{\mathcal{A}}, \widehat{\mathcal{B}}, \widehat{\mathcal{C}})$:

1. $\widehat{\mathcal{A}}$ runs $\mathcal{A}$ on the received program ρ from the challenger and performs the oracle queries from $\mathcal{A}$ and feeds the output back to $\mathcal{A}$ up until the j^{th} query, where $j \xleftarrow{\$} [q_{\mathcal{A}}]$ is sampled ahead of time. On the j^{th} query, $\widehat{\mathcal{A}}$ measures the query input register I. Let the measurement outcome be x'. $\mathcal{A}$ sends x' to both $\mathcal{B}$ and $\mathcal{C}$.
2. $\widehat{\mathcal{B}}$ on receiving string x' from $\mathcal{A}$ and a challenge string x_1 from the challenger checks if $x_1 = x'$, and if so output 0 else outputs 1.
3. $\widehat{\mathcal{C}}$ on receiving string x' from $\mathcal{A}$ and a challenge string x_2 from the challenger checks if $x_2 = x'$, and if so output 0 else outputs 1.

For any $h \in \{0, 4\}$, let the probability that with adversaries $(\mathcal{A}, \widehat{\mathcal{B}}, \widehat{\mathcal{C}})$ in $\mathbf{H}_h$, $\widehat{\mathcal{B}}$ and $\widehat{\mathcal{C}}$ both output the string 1, be p^{11}_h.

Since the output distribution of $\mathbf{H}_4$ is computationally indistinguishable from that of $\mathbf{H}_0$, $|p^{11}_4 - p^{11}_0|$ must be negligible in λ. However, note that by definition of $\widehat{\mathcal{A}}, \widehat{\mathcal{B}}, \widehat{\mathcal{C}}$, $\widehat{\mathcal{B}}$ and $\widehat{\mathcal{C}}$ both output 1 if and only if x', the measurement outcome of a random query of $\mathcal{A}$, satisfies $x' \neq x_1$ and $x' \neq x_2$, i.e., $x' \notin \{x_1, x_2\}$.

Hence, for any $h \in \{0, 4\}$,

$$p^{11}_h = \Pr[x' \notin \{x_1, x_2\}] = 1 - \Pr[x' \in \{x_1, x_2\}] = 1 - \mathbb{E}_{i \xleftarrow{\$} [q_{\mathcal{A}}]} \mathbb{E}[W^i_{h,\mathcal{A}}] = 1 - \sum_{i \in [q_{\mathcal{A}}]} \frac{\mathbb{E}[W^i_{h,\mathcal{A}}]}{q_{\mathcal{A}}} = 1 - \frac{\mathbb{E}[W_{h,\mathcal{A}}]}{q_{\mathcal{A}}}.$$

Therefore, by the computational indistinguishability of $\mathbf{H}_0$ and $\mathbf{H}_4$, we conclude that $\frac{|\mathbb{E}[W_{4,\mathcal{A}}] - \mathbb{E}[W_{0,\mathcal{A}}]|}{q_{\mathcal{A}}} = |p^{11}_4 - p^{11}_0|$ is negligible in λ, which implies that $|\mathbb{E}[W_{4,\mathcal{A}}] - \mathbb{E}[W_{0,\mathcal{A}}]|$ must be negligible in λ, since $q_{\mathcal{A}}$ is a polynomial in λ. Moreover since $\mathbb{E}[W_{0,\mathcal{A}}]$ is negligible as concluded above, $\mathbb{E}[W_{4,\mathcal{A}}]$ must be negligible in λ. □

[33] The reason this indistinguihsability is computational rather than statistical is that Claim 8.13 holds only for computationally bounded $\mathcal{A}$.

$\mathbf{H}_6$: In this hybrid, we change the algorithm Program with the cannonical algorithm $\mathsf{Program}_{canonical}$, i.e., $\mathbf{H}_6$ is defined as follows:

- $\mathcal{A}$ gets $\mathsf{UPO}.\mathcal{O}bf(1^\lambda, C)$ and oracle access to C, where:
  - $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$,
  - $x_1 \leftarrow \widetilde{\mathcal{D}}_{\mathbf{X}}$, $x_2 \leftarrow \widetilde{\mathcal{D}}_{\mathbf{X}}$.
  - Sample a bit $b \xleftarrow{\$} \{0,1\}$ uniformly at random, and if $b = 0$ set $\widehat{C} = C$, else if $b = 1$, sample $y_1 \xleftarrow{\$} \mathbf{Y} \setminus \{C(x_1)\}, y_2 \xleftarrow{\$} \mathbf{Y} \setminus \{C(x_2)\}$ and set,
  - $\widehat{C} \leftarrow \mathsf{Program}_{canonical}(C, (x_1, x_2), (y_1, y_2))$.
- After the splitting experiment, $\mathcal{B}$ receives x_1 and oracle access to $C \setminus \{x_1\}$, and $\mathcal{C}$ receives x_2 and oracle access to $C \setminus \{x_2\}$.
- $(\mathcal{B}, \mathcal{C})$ output $(y_{\mathcal{B}}, y_{\mathcal{C}})$.

The output of the hybrid is $\widehat{C}, x_1, x_2, y_{\mathcal{B}}, y_{\mathcal{C}}$. Denote the probability that $(y_{\mathcal{B}}, y_{\mathcal{C}}) = (\widehat{C}(x_1), \widehat{C}(x_2))$ to be p_6.
The only difference between $\mathbf{H}_5$ and $\mathbf{H}_6$ is that we replace Program with $\mathsf{Program}_{canonical}$, and since the output circuits of both Program and $\mathsf{Program}_{canonical}$ have the same functionality by the puncturing correctness (see Definition 3.10), by the iO security of UPO, the outputs of $\mathbf{H}_5$ and $\mathbf{H}_6$ must be computationally indistinguishable, and hence,

$$|p_6 - p_5| = \epsilon_6(\lambda), \tag{9}$$

for some negligible function $\epsilon_6(\cdot)$.
Observe that from the $(\widetilde{\mathcal{D}}_{\mathbf{X}} \times \widetilde{\mathcal{D}}_{\mathbf{X}})$-generalized UPO security of UPO, p_6 is at most $\frac{1}{2} + \mathsf{negl}(\lambda)$. In particular, in the reduction to the generalized UPO, the reduction adversary $(\mathcal{R}_{\mathcal{A}}, \mathcal{R}_{\mathcal{B}}, \mathcal{R}_{\mathcal{C}})$ against the $(\widetilde{\mathcal{D}}_{\mathbf{X}} \times \widetilde{\mathcal{D}}_{\mathbf{X}})$-generalized UPO security of UPO uses the adversary $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ in $\mathbf{H}_6$ in the following way.

1. $\mathcal{R}_{\mathcal{A}}$ samples $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$ and sends C along with the circuits $\mu_{R_{\mathcal{B}}}()$ and $\mu_{R_{\mathcal{C}}}()$ for $R_{\mathcal{B}}, R_{\mathcal{C}} \xleftarrow{\$} [2^k - 1]$ where for any $r \in [2^k - 1]$, $\mu_r(x)$ outputs the binary representation of $(\widetilde{(C(x) + r) \mod 2^k})^{th}$ element of of $\{0,1\}^k$ and for any vector $z \in \{0,1\}^k$, $\widetilde{z}$ denotes the representation of z in $\mathbb{Z}_{2^k}$.
2. $\mathcal{R}_{\mathcal{A}}$ upon receiveing the obfuscated program, runs $\mathcal{A}$ on the program while simulating oracle access to C to get a state $(\sigma_{\mathcal{B},\mathcal{C}})$ and finally outputs $\sigma_{\mathcal{B},\mathcal{C}}$ along with the description of C.
3. $\mathcal{R}_{\mathcal{B}}$ (respectively, $\mathcal{R}_{\mathcal{C}}$) on receiving x_1 (respectively, x_2) as the challenge from the challenger, and state $\sigma_{\mathcal{B}}$ (respectively, $\sigma_{\mathcal{C}}$) from $\mathcal{R}_{\mathcal{A}}$, runs $\mathcal{B}$ on $\sigma_{\mathcal{B}}$ (respectively, $\mathcal{C}$ on $\sigma_{\mathcal{C}}$) by simulating oracle access to $C \setminus \{x_1\}$ (respectively, $C \setminus \{x_2\}$) using C and x_1 (respectively, x_2).
4. $\mathcal{R}_{\mathcal{B}}$ (respectively, $\mathcal{R}_{\mathcal{C}}$) checks if the output of $\mathcal{B}$ (respectively, $\mathcal{C}$) is $C(x_1)$ (respectively $C(x_2)$) and if so, outputs 0, else, outputs 1.

Since $\widehat{C} = C$ in the $b = 0$ case in $\mathbf{H}_6$, the event $(y_{\mathcal{B}}, y_{\mathcal{C}}) = (\widehat{C}(x_1), \widehat{C}(x_2))$ in the $b = 0$ case in $\mathbf{H}_6$ corresponds to the event where $\mathcal{R}_{\mathcal{B}}$ and $\mathcal{R}_{\mathcal{C}}$ both output 0 in the unpunctured case of the above generalized UPO-security game for UPO with adversaries $(\mathcal{R}_{\mathcal{A}}, \mathcal{R}_{\mathcal{B}}, \mathcal{R}_{\mathcal{C}})$, i.e.,

$$\Pr[\mathcal{R}_{\mathcal{B}} \text{ and } \mathcal{R}_{\mathcal{C}} \text{ output } 0 \mid \text{Unpunctured}] = \Pr\Big[(y_{\mathcal{B}}, y_{\mathcal{C}}) = (\widehat{C}(x_1), \widehat{C}(x_2)) \mid b = 0 \text{ in } \mathbf{H}_6\Big]. \tag{10}$$

Next since in the $b = 1$ case in $\mathbf{H}_6$, $C(x_i) \neq \widehat{C}(x_i)$ for both $i = 1$ and $i = 2$, the event $(y_{\mathcal{B}}, y_{\mathcal{C}}) = (\widehat{C}(x_1), \widehat{C}(x_2))$ in the $b = 1$ case in $\mathbf{H}_6$ implies the event $y_{\mathcal{B}} \neq C(x_1) \wedge y_{\mathcal{C}} \neq C(x_2)$ in the $b = 1$ case in $\mathbf{H}_6$, which in turn corresponds

to the event where $\mathcal{R}_{\mathcal{B}}$ and $\mathcal{R}_{\mathcal{C}}$ both output 1 in the punctured case of the the above generalized UPO-security game for UPO. Hence we conclude that,

$$\begin{aligned}\Pr[\mathcal{R}_{\mathcal{B}} \text{ and } \mathcal{R}_{\mathcal{C}} \text{ output } 1 \mid \text{Punctured}] &= \Pr[y_{\mathcal{B}} \neq C(x_1) \wedge y_{\mathcal{C}} \neq C(x_2) \mid b = 1 \text{ in } \mathbf{H}_6] \\ &\geq \Pr\Big[(y_{\mathcal{B}}, y_{\mathcal{C}}) = (\widehat{C}(x_1), \widehat{C}(x_2)) \mid b = 1 \text{ in } \mathbf{H}_6\Big]. \end{aligned} \tag{11}$$

Therefore, we conclude by the $(\widetilde{\mathcal{D}}_{\mathbf{X}} \times \widetilde{\mathcal{D}}_{\mathbf{X}})$-generalized UPO-security of UPO, there exists a negligible function $\mathsf{negl}(\cdot)$, such that

$$\begin{aligned}&1/2 + \mathsf{negl}(\lambda) \\ &\geq \frac{\Pr[\mathcal{R}_{\mathcal{B}} \text{ and } \mathcal{R}_{\mathcal{C}} \text{ output } 0 \mid \text{Unpunctured}] + \Pr[\mathcal{R}_{\mathcal{B}} \text{ and } \mathcal{R}_{\mathcal{C}} \text{ output } 1 \mid \text{Punctured}]}{2} \\ &\geq \frac{\Pr\Big[(y_{\mathcal{B}}, y_{\mathcal{C}}) = (\widehat{C}(x_1), \widehat{C}(x_2)) \mid b = 0 \text{ in } \mathbf{H}_6\Big] + \Pr\Big[(y_{\mathcal{B}}, y_{\mathcal{C}}) = (\widehat{C}(x_1), \widehat{C}(x_2)) \mid b = 1 \text{ in } \mathbf{H}_6\Big]}{2} \quad \text{By (10) and (11).} \\ &= p_6.\end{aligned}$$

Therefore, by the hybrid arguments (Eqs. (4), (5), (6), (7), (8), and (9)), $p_0 \leq \frac{1}{2} + \widetilde{\mathsf{negl}}(\lambda)$ for some negligible function $\widetilde{\mathsf{negl}}(\cdot)$.

□

Combining Theorem 8.9 and Theorem 8.11, we can prove that k-point function families can be copy-protected.

Corollary 8.14. *Let $k = k(\lambda)$ be a polynomial in the security parameter such that $k > 1$, and let $\mathsf{Circ}^k := \{C_S\}_{S \in \binom{\mathbf{X}}{k}}$ be a set of circuits implementing Func^k, the set of all k-point functions on $\mathbf{X}$, i.e., Boolean functions on $\mathbf{X}$ that have exactly k-preimages of 1, where for each $S \subset \mathbf{X}$, $|S| = k$, C_S denotes the circuit corresponding to the k-point function with S as the preimage of 1.*

Then any UPO scheme $\mathsf{UPO} = (\mathcal{O}bf, \mathcal{E}val)$ satisfying $(\mathsf{Uniform}_{\mathbf{X}} \times \mathsf{Uniform}_{\mathbf{X}})$-generalized UPO security for Circ^k with respect to $\mathsf{Program}_{canonical}$, as well as iO security, is also a copy-protection scheme $(\mathcal{C}opyProtect, \mathcal{E}val)$ for Circ^k with $\mathcal{C}opyProtect = \mathsf{UPO}.\mathcal{O}bf$ and $\mathcal{E}val = \mathsf{UPO}.\mathcal{E}val$, satisfying oracular-$\mathcal{D}^{Evasive}_{\mathbf{X}^2,\mathsf{Circ}^k}$-unpredictable-style anti-piracy, where $\mathcal{D}^{Evasive}_{\mathbf{X}^2,\mathsf{Circ}^k}$ is defined as: $S, x_1, x_2 \leftarrow \mathcal{D}^{Evasive}_{\mathbf{X}^2,\mathsf{Circ}}(1^\lambda)$, where $S \xleftarrow{\$} \binom{\mathbf{X}}{k}$ and $x_1, x_2 \leftarrow \mathcal{D}^{Evasive}_{\mathbf{X}^2}(S)$, where $\mathcal{D}^{Evasive}_{\mathbf{X}^2}$ is as defined in Definition 8.7.

Proof of Corollary 8.14. The correctness of $(\mathcal{C}opyProtect, \mathcal{E}val)$, where recall $\mathcal{C}opyProtect = \mathsf{UPO}.\mathcal{O}bf$ and $\mathcal{E}val = \mathsf{UPO}.\mathcal{E}val$, immediately follows from the correctness of UPO.

Next, we prove oracular-$\mathcal{D}^{Evasive}_{\mathbf{X}^2,\mathsf{Circ}^k}$-unpredictable-style CP anti-piracy of $(\mathcal{C}opyProtect, \mathcal{E}val)$ for Circ^k.

Let $i\mathcal{O}$ be a post-quantum iO scheme, and consider the circuit class $i\mathcal{O}(\mathsf{Circ}^k)$, where each key is viewed as (S, r) where $S \in \binom{\mathbf{X}}{k}$ is an index for of a cicruit in Circ^k and r is a string of size that of the randomness needed to run $i\mathcal{O}$ compiler. Let $\widetilde{\mathcal{D}}_{\mathsf{Circ}^k}$ be a distribution on $i\mathcal{O}(\mathsf{Circ}^k)$ and Program be a programming algorithm for $i\mathcal{O}(\mathsf{Circ}^k)$, defined as follows:

- $\widetilde{\mathcal{D}}_{\mathsf{Circ}^k}$ outputs S, r where $S \xleftarrow{\$} \binom{\mathbf{X}}{k}$, and $r \xleftarrow{\$} \{0,1\}^w$ where w is the length of the randomness required to run the $i\mathcal{O}$ compiler.
- $\widetilde{\mathsf{Program}}$ is defined as follows: on input $((C, r), (x_1, \ldots, x_\ell), (\mu_1, \ldots, \mu_\ell))$, where $\mu_1, \ldots, \mu_\ell$ are programming circuits for the punctured points, $\widetilde{\mathsf{Program}}$ computes $y_i \leftarrow \mu_i(x_i)$ for every $i \in [\ell]$, and outputs $\widehat{C} \leftarrow \mathsf{Program}(C, (x_1, \ldots, x_\ell), (y_1, \ldots, y_\ell))$.
  - If $x' = x_i$ for any $i \in [\ell]$, output y_i,
  - Else, output $C(x')$.

By Theorem 8.9 since $k \geq 2$, we conclude that $i\mathcal{O}(\mathsf{Circ}^k)$ with respect to $\mathsf{Program}$ is $(\widetilde{\mathcal{D}^{Evasive}}_{\mathbf{X}^2}, \mathsf{Uniform}_{\mathbf{X}}, \mathcal{D}_{\mathbf{Y}}^{\mathsf{half\text{-}correct}}, \widetilde{\mathcal{D}}_{\mathsf{Circ}^k})$-Progmaskable circuit class where $\mathcal{D}_{\mathbf{Y}}^{\mathsf{half\text{-}correct}}$ is the half-correct distribution (see Definition 8.5), and $\widetilde{\mathcal{D}^{Evasive}}_{\mathbf{X}^2}(S,r) := \mathcal{D}^{Evasive}_{\mathbf{X}^2}(S)$ where $\mathcal{D}^{Evasive}_{\mathbf{X}^2}$ is as defined in Definition 8.7. Therefore, by Theorem 8.11, we conclude that $(\mathit{CopyProtect}, \mathit{Eval})$ satisfies $1/2$-oracular -$\widetilde{\mathcal{D}^{Evasive}}_{\mathbf{X}^2,\mathsf{Circ}^k}$-unpredictable-style-CP antipiracy for $i\mathcal{O}(\mathsf{Circ})$, where $\widetilde{\mathcal{D}^{Evasive}}_{\mathbf{X}^2,\mathsf{Circ}^k}$ first samples $S, r \leftarrow \widetilde{\mathcal{D}^{Evasive}}_{\mathsf{Circ}^k}(1^\lambda)$ and then samples $x_1, x_2 \leftarrow \widetilde{\mathcal{D}^{Evasive}}_{\mathbf{X}^2}(S,r)$, i.e., $x_1, x_2 \leftarrow \mathcal{D}^{Evasive}_{\mathbf{X}^2}(S)$.

However, by the iO-security of UPO, we can move to a hybrid where in the oracular-copy-protection anti-piracy game, the challenger after sampling $(S,r) \leftarrow \widetilde{\mathcal{D}^{Evasive}}_{\mathsf{Circ}^k}(1^\lambda)$, instead of running $\mathit{CopyProtect}$ on $i\mathcal{O}(C_S; r)$, runs $\mathit{CopyProtect}$ on C_S after sufficient padding to match the circuit size of $i\mathcal{O}(C_S; r)$; and the winning condition remains the same. Clearly, by the iO security of UPO, the output of the copy-protection adversary, even in the presence of the correct output values at the challenge points, must be computationally indistinguishable from that of the real game, and hence the winning probability of the adversary in this hybrid is at most $\frac{1}{2}$ up to negligible factors. However, note that the hybrid game is the same as the oracular-$\mathcal{D}^{Evasive}_{\mathbf{X}^2,\mathsf{Circ}^k}$-unpredictability-style-CP anti-piracy game of $(\mathit{CopyProtect}, \mathit{Eval})$ for Circ^k. Hence, $(\mathit{CopyProtect}, \mathit{Eval})$ satisfies oracular-$\mathcal{D}^{Evasive}_{\mathbf{X}^2,\mathsf{Circ}^k}$-unpredictability-style-CP anti-piracy for Circ.

□

Combining Corollary 8.14 with the "Moreover" part of Theorem 5.1, we immediately get Corollary 8.15.

Corollary 8.15. *Let $\boldsymbol{X} = \{0,1\}^n$ where $n = n(\lambda)$ is a polynomial in λ, and let $k = k(\lambda)$ be a polynomial in the security parameter such that $k > 1$. Let $\mathsf{Circ}^k := \{C_S\}_{S \in \binom{\boldsymbol{X}}{k}}$ be a set of circuits implementing Func^k, the set of all k-point functions on $\boldsymbol{X}$ where for each $S \subset \boldsymbol{X}, |S| = k$, C_S denotes the circuit corresponding to the k-point function with S as the preimage of 1.*

Then, assuming the existence of polynomially secure iO and the OWFs, there exists a copy-protection scheme for Circ^k that satisfies oracular-$\mathcal{D}^{Evasive}_{\boldsymbol{X}^2,\mathsf{Circ}^k}$-unpredictable-style-CP anti-piracy (see Definition 3.1), where $\mathcal{D}^{Evasive}_{\boldsymbol{X}^2,\mathsf{Circ}^k}$ is defined as: $S, x_1, x_2 \leftarrow \mathcal{D}^{Evasive}_{\boldsymbol{X}^2,\mathsf{Circ}}(1^\lambda)$, where $S \xleftarrow{\$} \binom{\boldsymbol{X}}{k}$ and $x_1, x_2 \leftarrow \mathcal{D}^{Evasive}_{\boldsymbol{X}^2}(S)$, where $\mathcal{D}^{Evasive}_{\boldsymbol{X}^2}$ is as defined in Definition 8.7.

8.3 Relation between Progmaskable and Puncturable Circuits

First, we note the following property of 1-point pseudorandomness style puncturable secure circuit class.

Lemma 8.16. *For any circuit class Circ that satisfies 1-point m-bit $(\mathcal{D}_{\boldsymbol{X}}, \mathcal{D}_{\mathsf{Circ}})$-pseudorandomness style puncturing security (with respect to $\mathsf{Puncture}$), the following holds:*

$$\{\widehat{C}, x, C(x)\}_{\widehat{C} \leftarrow \mathsf{Puncture}(C,x), x \leftarrow \mathcal{D}_{\boldsymbol{X}}, C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)} \approx_c \{\widehat{C}, x, Y\}_{\mathsf{Puncture}(C,x), x \leftarrow \mathcal{D}_{\boldsymbol{X}}, Y \xleftarrow{\$} \boldsymbol{Y} \setminus \{C(x)\}}. \tag{12}$$

Proof of Lemma 8.16. Let the ensembles in the LHS and the RHS of (12) be D_0 and D_1, respectively. Moreover, let D_2 be the ensemble

$$\{\widehat{C}, x, Y\}_{\mathsf{Puncture}(C,x), x \leftarrow \mathcal{D}_{\mathbf{X}}, Y \xleftarrow{\$} \mathbf{Y}}.$$

Hence, for any given distinguisher $\mathcal{A}$, we get

$$
\begin{aligned}
&\left(1-\frac{1}{2^m}\right)\left|\Pr_{\{\widehat{C},x,y\}\leftarrow D_0}[\mathcal{A}(\widehat{C},x,y)=1]-\Pr_{\{\widehat{C},x,y\}\leftarrow D_1}[\mathcal{A}(\widehat{C},x,y)=1]\right|\\
&=\left|\left(1-\frac{1}{2^m}\right)\Pr_{\{\widehat{C},x,y\}\leftarrow D_0}[\mathcal{A}(\widehat{C},x,y)=1]-\left(1-\frac{1}{2^m}\right)\Pr_{\{\widehat{C},x,y\}\leftarrow D_1}[\mathcal{A}(\widehat{C},x,y)=1]\right|\\
&\leq\left|\left(1-\frac{1}{2^m}\right)\Pr_{\{\widehat{C},x,y\}\leftarrow D_0}[\mathcal{A}(\widehat{C},x,y)=1]+\frac{1}{2^m}\Pr_{\{\widehat{C},x,y\}\leftarrow D_0}[\mathcal{A}(\widehat{C},x,y)=1]\right.\\
&\left.-\left(\left(1-\frac{1}{2^m}\right)\Pr_{\{\widehat{C},x,y\}\leftarrow D_1}[\mathcal{A}(\widehat{C},x,y)=1]+\frac{1}{2^m}\Pr_{\{\widehat{C},x,y\}\leftarrow D_0}[\mathcal{A}(\widehat{C},x,y)=1]\right)\right|\\
&=\left|\Pr_{\{\widehat{C},x,y\}\leftarrow D_0}[\mathcal{A}(\widehat{C},x,y)=1]\right.\\
&\left.-\left(\left(1-\frac{1}{2^m}\right)\Pr_{\{\widehat{C},x,y\}\leftarrow D_1}[\mathcal{A}(\widehat{C},x,y)=1]+\frac{1}{2^m}\Pr_{\{\widehat{C},x,y\}\leftarrow D_0}[\mathcal{A}(\widehat{C},x,y)=1]\right)\right|\\
&=\left|\Pr_{\{\widehat{C},x,y\}\leftarrow D_0}[\mathcal{A}(\widehat{C},x,y)=1]-\Pr_{\{\widehat{C},x,y\}\leftarrow D_2}[\mathcal{A}(\widehat{C},x,y)=1]\right|\\
&\leq \mathsf{negl}(\lambda),
\end{aligned}
$$

for some negligible function $\mathsf{negl}(\cdot)$, where the last equality follows from the above-mentioned formulation of D_2, and the last inequality follows from 1-point m-bit $(\mathcal{D}_{\mathbf{X}},\mathcal{D}_{\mathsf{Circ}})$-pseudorandomness style puncturing security of Circ. Therefore, we conclude that,

$$
|\Pr_{\{\widehat{C},x,y\}\leftarrow D_0}[\mathcal{A}(\widehat{C},x,y)=1]-\Pr_{\{\widehat{C},x,y\}\leftarrow D_1}[\mathcal{A}(\widehat{C},x,y)=1]|\leq\frac{\mathsf{negl}(\lambda)}{1-\frac{1}{2^m}}\leq 2\cdot\mathsf{negl}(\lambda),
$$

which is also negligible in the security parameter λ. Hence, we conclude that $D_1\approx_c D_0$. □

Using Lemma 8.16 we can prove the following theorem for 1-point pseudorandomness-style puncturable circuits.

Theorem 8.17 (Different string pseudorandomness from 1-point pseudorandomness-style puncturable circuits). *Let $i\mathcal{O}$ be a post-quantum iO scheme, and let $\ell:=\ell(\lambda)$ be a polynomial in the security parameter, λ. Then, for any circuit class* Circ *that satisfies* 1*-point m-bit $(\mathcal{D}_{\boldsymbol{X}},\mathcal{D}_{\mathsf{Circ}})$-pseudorandomness style puncturing security (with respect to* Puncture*), and up to sufficient padding of the punctured circuit before running the iO compiler, the following holds:*

$$
\begin{aligned}
&\{\widetilde{\widehat{C}},(x_1,\ldots,x_\ell),(C(x_1),\ldots,C(x_\ell))\}_{\substack{\widetilde{\widehat{C}}\leftarrow i\mathcal{O}(\widehat{C}),\widehat{C}\leftarrow\mathsf{Puncture}(C,(x_1,\ldots,x_\ell)),\\ x_1\leftarrow\mathcal{D}_{\boldsymbol{X}},\ldots x_\ell\leftarrow\mathcal{D}_{\boldsymbol{X}},C\leftarrow\mathcal{D}_{\mathsf{Circ}}(1^\lambda)}}\\
&\approx_c\{\widetilde{\widehat{C}},(x_1,\ldots,x_\ell),(Y_1,\ldots,Y_\ell))\}_{\substack{\widetilde{\widehat{C}}\leftarrow i\mathcal{O}(\widehat{C}),\widehat{C}\leftarrow\mathsf{Puncture}(C,(x_1,\ldots,x_\ell)),\\ Y_1\xleftarrow{\$}\boldsymbol{Y}\setminus\{C(x_1)\},\ldots Y_\ell\xleftarrow{\$}\boldsymbol{Y}\setminus\{C(x_\ell)\},\\ x_1\leftarrow\mathcal{D}_{\boldsymbol{X}},\ldots x_\ell\leftarrow\mathcal{D}_{\boldsymbol{X}},C\leftarrow\mathcal{D}_{\mathsf{Circ}}(1^\lambda)}}.
\end{aligned}
\tag{13}
$$

Proof of Theorem 8.17. We consider $\ell+1$-hybrids, $\mathbf{H}_0,\mathbf{H}_1,\ldots,\mathbf{H}_\ell$, where for every $i\in\{0,1,2,\ldots,\ell\}$, $\mathbf{H}_i$ outputs the ensemble:

$$
\{\widetilde{\widehat{C}},(x_1,\ldots,x_\ell),(C(x_1),\ldots,C(x_{\ell-i}),Y_{\ell-i+1},\ldots,Y_\ell)\},
$$

where

$$
\widetilde{\widehat{C}}\leftarrow i\mathcal{O}(\widehat{C}),\widehat{C}\leftarrow\mathsf{Puncture}(C,(x_1,\ldots,x_\ell)),\quad Y_j\xleftarrow{\$}\mathbf{Y}\setminus\{C(x_j)\},\forall\ell-i<j\leq\ell,
$$

$$x_i \leftarrow \mathcal{D}_{\mathbf{X}}, \forall i \in [\ell], \quad C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda).$$

upto sufficient padding of the punctured circuit before running the iO compiler on it. Since the outputs of $\mathbf{H}_0$ and $\mathbf{H}_\ell$ are the same as LHS and RHS, respectively, of (13), it is enough to show that for every $i \in \{0, 1, \ldots, \ell-1\}$, the outputs of $\mathbf{H}_i$ and $\mathbf{H}_{i+1}$ are computationally indistinguishable.

To do that, fix an $i \in \{0, 1, \ldots, \ell-1\}$. We consider the following hybrids:

$\mathbf{H}_i^0$: This is the same as $\mathbf{H}_i$.

$\mathbf{H}_i^1$: In this hybrid, we compute the obfuscated circuit differently, i.e., the hybrid outputs

$$\{\widetilde{\widehat{C}}, (x_1, \ldots, x_\ell), (Y_1, \ldots, Y_\ell)\},$$

where as before,

$$Y_j \leftarrow C(x_j), \forall 1 \le j \le \ell - i, \quad Y_j \xleftarrow{\$} \mathbf{Y} \setminus \{C(x_j)\}, \forall \ell - i + 1 \le j \le \ell,$$

$$x_i \leftarrow \mathcal{D}_{\mathbf{X}}, \forall i \in [\ell], \quad C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda), \text{ but } \widetilde{\widehat{C}} \leftarrow i\mathcal{O}(\widehat{C}),$$

where $\widehat{C}$ is the circuit that has $\widehat{C}_{x_{\ell-i}}$ hardcoded in it where $\widehat{C}_{x_{\ell-i}} \leftarrow \mathsf{Puncture}(C, x_{\ell-i})$, such that on every $x \in \{x_1, \ldots, x_\ell\}$, $\widehat{C}$ outputs $\perp$ and on every $x \in \mathbf{X} \setminus \{x_1, \ldots, x_\ell\}$ $\widehat{C}$ outputs $\widehat{C}_{x_{\ell-i}}(x)$.

Note that across hybrids, $\mathbf{H}_i^0$ and $\mathbf{H}_i^1$, we did not change the functionality of $\widehat{C}$, hence by the iO security of $i\mathcal{O}$, the outputs of $\mathbf{H}_i^0$ and $\mathbf{H}_i^1$ must be computationally indistinguishable.

$\mathbf{H}_i^2$: In this hybrid, we change the distribtution on $Y_{\ell-i}$, i.e., the hybrid outputs

$$\{\widetilde{\widehat{C}}, (x_1, \ldots, x_\ell), (Y_1, \ldots, Y_\ell)\},$$

where

$$Y_j \leftarrow C(x_j), \forall 1 \le j \le \ell - i - 1, \quad Y_j \xleftarrow{\$} \mathbf{Y} \setminus \{C(x_j)\}, \forall \ell - i \le j \le \ell,$$

$$x_i \leftarrow \mathcal{D}_{\mathbf{X}}, \forall i \in [\ell], \quad C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda), \text{ and } \widetilde{\widehat{C}} \leftarrow i\mathcal{O}(\widehat{C}),$$

where $\widehat{C}$ is the circuit that has $\widehat{C}_{x_{\ell-i}}$ hardcoded in it where $\widehat{C}_{x_{\ell-i}} \leftarrow \mathsf{Puncture}(C, x_{\ell-i})$, such that on every $x \in \{x_1, \ldots, x_\ell\}$, $\widehat{C}$ outputs $\perp$ and on every $x \in \mathbf{X} \setminus \{x_1, \ldots, x_\ell\}$ $\widehat{C}$ outputs $\widehat{C}_{x_{\ell-i}}(x)$.

Since Circ satisfies 1-point m-bit $(\mathcal{D}_{\mathbf{X}}, \mathcal{D}_{\mathsf{Circ}})$-pseudorandomness style puncturing security (with respect to $\mathsf{Puncture}$), by Lemma 8.16, the outputs of $\mathbf{H}_i^1$ and $\mathbf{H}_i^2$ must be computationally indistinguishable. This is because, given any distinguisher $\mathcal{A}$ that distinguishes between the outputs of $\mathbf{H}_i^1$ and $\mathbf{H}_i^2$, we can construct the following distinguisher $\mathcal{B}$ that distinguishes between the LHS and RHS distributions in Lemma 8.16 (respectively, denoted as D_0 and D_1), with the same distinguishing advantage as $\mathcal{A}$, as follows. Given a sample $\widehat{C}_{x'}, x', Y'$, $\mathcal{B}$ samples $x_j \leftarrow \mathcal{D}_{\mathbf{X}}$ for every $j \in [\ell] \setminus \{\ell - i\}$ in an IID fashion, and set $x_{\ell-i} := x'$. Then $\mathcal{B}$ generates $\widetilde{\widehat{C}} \leftarrow i\mathcal{O}(\widehat{C})$ where $\widehat{C}$ is the circuit that has $\widehat{C}_{x'}$ hardcoded in it, such that on every $x \in \{x_1, \ldots, x_\ell\}$, $\widehat{C}$ outputs $\perp$ and on every $x \in \mathbf{X} \setminus \{x_1, \ldots, x_\ell\}$ $\widehat{C}$ outputs $\widehat{C}_{x'}(x)$. Next, for every $1 \le j < \ell - i$, $\mathcal{B}$ samples $Y_j \leftarrow C(x_j)$, and for every $\ell - i < j \le \ell$, samples $Y_j \xleftarrow{\$} \mathbf{Y} \setminus \{C(x_j)$, and sets $Y_{\ell-i} := Y'$. Finally $\mathcal{B}$ runs $\mathcal{A}$ on $\widetilde{\widehat{C}}, (x_1, \ldots, x_\ell), (Y_1, \ldots, Y_\ell)$, and outputs the outcome. Clearly, if $\widehat{C}_{x'}, x', Y' \leftarrow D_0$, then the input of $\mathcal{A}$ has the same distribution as $\mathbf{H}_i^1$ and if $\widehat{C}_{x'}, x', Y' \leftarrow D_1$, then the input of $\mathcal{A}$ has the same distribution as $\mathbf{H}_i^2$. Hence, $\mathcal{B}$ has the same advantage in distinguishing D_0 and D_1, as $\mathcal{A}$ has in distinguishing the outputs of $\mathbf{H}_i^1$ and $\mathbf{H}_i^2$.

$\mathbf{H}_i^3$: In this hybrid, we compute the obfuscated circuit differently, i.e., the hybrid outputs

$$\{\widetilde{\widehat{C}}, (x_1, \ldots, x_\ell), (Y_1, \ldots, Y_\ell)\},$$

where

$$Y_j \leftarrow C(x_j\}, \forall 1 \le j \le \ell - i - 1, \quad Y_j \xleftarrow{\$} \mathbf{Y} \setminus \{C(x_j)\}, \forall \ell - i \le j \le \ell, \quad x_i \leftarrow \mathcal{D}_{\mathbf{X}}, \forall i \in [\ell],$$

and $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$, but $\widetilde{\widehat{C}} \leftarrow i\mathcal{O}(\widehat{C})$ where $\widehat{C} \leftarrow \mathsf{Puncture}(C, (x_1, \ldots, x_\ell))$. Note that across hybrids, $\mathbf{H}_i^2$ and $\mathbf{H}_i^3$, we did not change the functionality of $\widehat{C}$, hence by the iO security of $i\mathcal{O}$, the outputs of $\mathbf{H}_i^2$ and $\mathbf{H}_i^3$ must be computationally indistinguishable.

However, note that $\mathbf{H}_i^3$ is the same as $\mathbf{H}_{i+1}$. Therefore, by hybrid, arguments we conclude that the outputs of $\mathbf{H}_i$ and $\mathbf{H}_{i+1}$ are computationally indistinguishable. Since $i \in \{0, 1, \ldots, \ell - 1\}$ was arbitrary we conclude that the same holds for every $i \in \{0, 1, \ldots, \ell - 1\}$, which completes the proof. □

Next, we show that 1-point pseudorandomness-style puncturing secure circuit classes are Progmaskable.

Theorem 8.18. *Let $i\mathcal{O}$ be a post-quantum iO scheme, and let $\ell := \ell(\lambda)$ be a polynomial in the security parameter λ. Then, for any circuit class* Circ *that satisfies* 1*-point m-bit $(\mathcal{D}_{\boldsymbol{X}}, \mathcal{D}_{\mathsf{Circ}})$-pseudorandomness-style puncturing security (with respect to* $\mathsf{Puncture}$*) (see Definition 7.1), then $i\mathcal{O}(\mathsf{Circ})$ satisfies ℓ-points $((\mathcal{D}_{\boldsymbol{X}})^\ell, \mathcal{D}_{\boldsymbol{X}}, \mathcal{D}_{\boldsymbol{Y}}^{\mathsf{half\text{-}correct}}, \widetilde{\mathcal{D}}_{\mathsf{Circ}})$-Progmaskability security with respect to an efficient algorithm* $\mathsf{Program}$*, where:*

- *$\mathcal{D}_{\boldsymbol{Y}}^{\mathsf{half\text{-}correct}}$: half-correct distribution (see Definition 8.5),*
- *$\widetilde{\mathcal{D}}_{\mathsf{Circ}}$ outputs C, r where $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$, and $r \xleftarrow{\$} \{0,1\}^w$ where w is the length of the randomness required to run the $i\mathcal{O}$ compiler.*
- $\mathsf{Program}$ *is defined as follows: on input $((C, r), (x_1, \ldots, x_\ell), (y_1, \ldots, y_\ell))$, it outputs $\widehat{C}$. Here, $\widehat{C} = i\mathcal{O}(1^\lambda, C'; r)$ and $C'(x')$ is defined as follows:*
  - *If $x' = x_i$ for some $i \in [\ell]$, output y_i,*
  - *Else, output $C_{x_1,\ldots,x_\ell}(x')$, where $C_{x_1,\ldots,x_\ell} \leftarrow \mathsf{Puncture}(C, (x_1, \ldots, x_\ell))$.*

Proof of Theorem 8.18. We describe the hybrids below.

$\mathbf{H}_0$: this corresponds to $\mathcal{D}_0$ in Definition 8.4. That is, this hybrid outputs $(\widehat{C}, x_1, \ldots, x_\ell) \leftarrow \mathcal{D}_0$, where $x_i \leftarrow \mathcal{D}_{\mathbf{X}}$ for every $i \in [\ell]$ and $\widehat{C} = i\mathcal{O}(C; r)$, $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$ and $r \xleftarrow{\$} \{0,1\}^w$ where w is the length of the randomness required to run the $i\mathcal{O}$ compiler, i.e., $\widehat{C} \leftarrow i\mathcal{O}(C)$.

$\mathbf{H}_1$: outputs $(\widehat{C}, x_1, \ldots, x_\ell)$, where as before, $x_i \leftarrow \mathcal{D}_{\mathbf{X}}$ for every $i \in [\ell]$, but $\widehat{C} \leftarrow i\mathcal{O}(1^\lambda, C'; r)$ where C' is defined as follows: Sample $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$ and $r \xleftarrow{\$} \{0,1\}^w$ where w is the length of the randomness required to run the $i\mathcal{O}$ compiler and then $C'(x')$ is defined as follows:

- If $x' = x_i$ for some $i \in [\ell]$, output $C(x_i)$,
- Else, output $C_{x_1,\ldots,x_\ell}(x')$, where $C_{x_1,\ldots,x_\ell} \leftarrow \mathsf{Puncture}(C, (x_1, \ldots, x_\ell))$.

By iO security of $i\mathcal{O}$, $\mathbf{H}_0$ and $\mathbf{H}_1$ are computationally indistinguishable.

$\mathbf{H}_2$: Let X be a uniform boolean random variable, i.e., $X = 0$ with probability $\frac{1}{2}$, and $X = 1$ with probability $\frac{1}{2}$. Sample X and if $X = 0$, output $(\widehat{C}, x_1, \ldots, x_\ell) \leftarrow \mathbf{H}_1$. Else, output $(\widehat{C}, x_1, \ldots, x_\ell)$ generated as follows: generate $\widehat{C} \leftarrow i\mathcal{O}(1^\lambda, C')$, where $r \xleftarrow{\$} \{0,1\}^w$ where w is the length of the randomness required to run the $i\mathcal{O}$ compiler and C' is defined as follows: Sample $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$ and $y_1 \xleftarrow{\$} \mathbf{Y} \setminus \{C(x_1)\}, \ldots, y_\ell \xleftarrow{\$} \mathbf{Y} \setminus \{C(x_\ell)\}$, and then $C'(x')$ is defined as follows:

- If $x' = x_i$ for some $i \in [\ell]$, output y_i,
- Else, output $C_{x_1,\ldots,x_\ell}(x')$, where $C_{x_1,\ldots,x_\ell} \leftarrow \mathsf{Puncture}(C, (x_1, \ldots, x_\ell))$.

$\mathbf{H}_2$ is a union of two events $X = 0$ and $X = 1$. Conditioned on $X = 0$, $\mathbf{H}_2$ is the same as $\mathbf{H}_1$, and conditioned on $X = 1$, it is the same as $\mathbf{H}_1$ except that we changed the distribution of $y_1, y_2, \ldots, y_\ell$ to $y_1 \xleftarrow{\$} \mathbf{Y} \setminus \{C(x_1)\}, \ldots, y_\ell \xleftarrow{\$} \mathbf{Y} \setminus \{C(x_\ell)\}$. However, the change is indistinguishable by the $(\mathcal{D}_{\mathbf{X}}, \mathcal{D}_{\mathsf{Circ}})$-puncturing security of Circ. Hence, by puncturing security, the outputs of $\mathbf{H}_1$ and $\mathbf{H}_2$ are computationally indistinguishable by Theorem 8.17 since Circ satisifes 1-point m-bit $(\mathcal{D}_{\mathbf{X}}, \mathcal{D}_{\mathsf{Circ}})$-pseudorandomness style puncturing security with respect to Puncture and $i\mathcal{O}$ is a post-quantum iO.

Note that the only difference between the $X = 0$ and $X = 1$ cases in $\mathbf{H}_2$ is that we changed the programmed value of C' at $x_1, \ldots, x_\ell$. Therefore, $\mathbf{H}_2$ can be rewritten as follows.
$\mathbf{H}_3$: Output $(\widehat{C}, x_1, \ldots, x_\ell)$ generated as follows: generate $\widehat{C} \leftarrow i\mathcal{O}(1^\lambda, C')$, where C' is defined as follows: Sample $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$ and $y_1 \xleftarrow{\$} \mathbf{Y} \setminus \{C(x_1)\}, \ldots, y_\ell \xleftarrow{\$} \mathbf{Y} \setminus \{C(x_\ell)\}$. Let X be a uniform boolean random variable, i.e., $X = 0$ with probability $\frac{1}{2}$, and $X = 1$ with probability $\frac{1}{2}$. Sample X and if $X = 0$, for any input x', $C'(x')$ is defined as follows:

- If $x' = x_i$ for some $i \in [\ell]$, output $C(x_i)$,
- Else, output $C_{x_1,\ldots,x_\ell}(x')$, where $C_{x_1,\ldots,x_\ell} \leftarrow \mathsf{Puncture}(C, (x_1, \ldots, x_\ell))$.

and if $X = 1$, $C'(x')$ is defined as follows:

- If $x' = x_i$ for some $i \in [\ell]$, output y_i,
- Else, output $C_{x_1,\ldots,x_\ell}(x')$, where $C_{x_1,\ldots,x_\ell} \leftarrow \mathsf{Puncture}(C, (x_1, \ldots, x_\ell))$.

Clearly, the outputs of $\mathbf{H}_2$ and $\mathbf{H}_3$ are identically distributed.
$\mathbf{H}_4$: this corresponds to $\mathcal{D}_1$ in Definition 8.4 where $\widetilde{\mathcal{D}}_{\mathbf{X}} = \mathcal{D}_{\mathbf{X}}$. In other words, $\mathbf{H}_4$ outputs $(\widehat{C}, x_1, \ldots, x_\ell) \leftarrow \mathcal{D}_1$.

The outputs of hybrids $\mathbf{H}_3$ and $\mathbf{H}_4$ are identically distributed because the distribution of the programmed values at the points of puncture $\{x_i\}_{i\in[\ell]}$ in $\mathbf{H}_3$ is the same as the half-correct distribution (see Definition 8.5).

□

Combining Theorem 8.18 with Theorem 8.11, we get the following corollary.

Corollary 8.19 (Unpredictable copy-protection for pseudorandomness-style puncturable secure circuit class from UPO). *Let* $(\mathsf{Circ}, \mathsf{Puncture})$ *be a circuit class satisfying* 1*-point* 1*-bit* $(\mathcal{D}_{\boldsymbol{X}}, \mathcal{D}_{\mathsf{Circ}})$*-pseudorandomness style puncturing security. Then assuming post quantum iO exists, any UPO scheme* $\mathsf{UPO} = (\mathit{Obf}, \mathit{Eval})$ *satisfying* $(\mathcal{D}_{\boldsymbol{X}} \times \mathcal{D}_{\boldsymbol{X}})$*-generalized UPO security for* Circ *with respect to* $\mathsf{Program}_{\mathit{canonical}}$ *as well as iO security is also a copy-protection scheme* $(\mathit{CopyProtect}, \mathit{Eval})$ *for* Circ *with* $\mathit{CopyProtect}(\cdot) = \mathsf{UPO}.\mathit{Obf}?$ *and* $\mathit{Eval} = \mathsf{UPO}.\mathit{Eval}$*, satisfying oracular-*$(\mathcal{D}_{\boldsymbol{X}} \times \mathcal{D}_{\boldsymbol{X}} \times \mathcal{D}_{\mathsf{Circ}})$*-unpredictable-style-CP anti-piracy.*

Proof of Corollary 8.19. The correctness of $(\mathit{CopyProtect}, \mathit{Eval})$, where recall $\mathit{CopyProtect} = \mathsf{UPO}.\mathit{Obf}$ and $\mathit{Eval} = \mathsf{UPO}.\mathit{Eval}$, immediately follows from the correctness of UPO.

Next, we prove oracular-$(\mathcal{D}_{\mathbf{X}} \times \mathcal{D}_{\mathbf{X}} \times \mathcal{D}_{\mathsf{Circ}})$-unpredictable-style CP anti-piracy of $(\mathit{CopyProtect}, \mathit{Eval})$ for Circ.

Let $i\mathcal{O}$ be a post-quantum iO scheme, and consider the circuit class $i\mathcal{O}(\mathsf{Circ})$, where each key is viewed as (C, r) for some circuit key/description C of a cicruit in Circ and r is a string of size that of the randomness needed to run $i\mathcal{O}$ compiler. Let $\widetilde{\mathcal{D}}_{\mathsf{Circ}}$ be a distribution on Circ and Program be a programming algorithm for $i\mathcal{O}(\mathsf{Circ})$, defined as follows:

- $\widetilde{\mathcal{D}}_{\mathsf{Circ}}$ outputs C, r where $C \leftarrow \mathcal{D}_{\mathsf{Circ}}(1^\lambda)$, and $r \xleftarrow{\$} \{0,1\}^w$ where w is the length of the randomness required to run the $i\mathcal{O}$ compiler.
- $\widetilde{\mathsf{Program}}$ is defined as follows: on input $((C, r), (x_1, \ldots, x_\ell), (\mu_1, \ldots, \mu_\ell))$, where $\mu_1, \ldots, \mu_\ell$ are programming circuits for the punctured points, $\widetilde{\mathsf{Program}}$ computes $y_i \leftarrow \mu_i(x_i)$ for every $i \in [\ell]$, and outputs $\widehat{C} \leftarrow \mathsf{Program}(C, (x_1, \ldots, x_\ell), (y_1, \ldots, y_\ell))$.

- If $x' = x_i$ for some $i \in [\ell]$, output y_i,
- Else, output $C(x')$.

By Theorem 8.18, we conclude that $i\mathcal{O}(\mathsf{Circ})$ with respect to $\mathsf{Program}$ is a 2-points 1-bit $((\mathcal{D}_{\mathbf{X}} \times \mathcal{D}_{\mathbf{X}}, \mathcal{D}_{\mathbf{X}}, \mathcal{D}_{\mathbf{Y}}^{\mathsf{half\text{-}correct}}, \widetilde{\mathcal{D}}_{\mathsf{Circ}})$-Progmaskable circuit class where $\mathcal{D}_{\mathbf{Y}}^{\mathsf{half\text{-}correct}}$ is the half-correct distribution (see Definition 8.5). Therefore, by Theorem 8.11, we conclude that $(\mathit{CopyProtect}, \mathit{Eval})$ satisfies $1/2$-oracular -$(\mathcal{D}_{\mathbf{X}} \times \mathcal{D}_{\mathbf{X}}, \widetilde{\mathcal{D}}_{\mathsf{Circ}})$-unpredictable-style-CP antipiracy for $i\mathcal{O}(\mathsf{Circ})$. However, by the iO-security of UPO, we can move to a hybrid where in the oracular-copy-protection anti-piracy game, the challenger after sampling $(C, r) \leftarrow \widetilde{\mathcal{D}}_{\mathsf{Circ}}$, instead of running $\mathit{CopyProtect}$ on $i\mathcal{O}(C; r)$, runs $\mathit{CopyProtect}$ on C after sufficient padding to match the circuit size of $i\mathcal{O}(C; r)$; the winning condition remains the same. Clearly, by the iO security of UPO, the output of the copy-protection adversary, even in the presence of the correct output values at the challenge points, must be computationally indistinguishable from that of the real game, and hence the winning probability of the adversary in this hybrid is at most $\frac{1}{2}$ up to negligible factors. However, note that the hybrid game is the same as the oracular-$(\mathcal{D}_{\mathbf{X}} \times \mathcal{D}_{\mathbf{X}} \times \mathcal{D}_{\mathsf{Circ}})$-unpredictability-style-CP anti-piracy game of $(\mathit{CopyProtect}, \mathit{Eval})$ for Circ. Hence, $(\mathit{CopyProtect}, \mathit{Eval})$ satisfies oracular-$(\mathcal{D}_{\mathbf{X}} \times \mathcal{D}_{\mathbf{X}} \times \mathcal{D}_{\mathsf{Circ}})$-unpredictability-style-CP anti-piracy for Circ.

□

By combining Corollary 8.19 with Theorems 3.17 and 5.1, we immediately get the following corollary.

Corollary 8.20 (Unpredictable copy-protection for pseudorandomness style puncturable secure circuit class). *Let $(\mathsf{Circ}, \mathsf{Puncture})$ be a circuit class satisfying 1-point 1-bit $(\mathcal{D}_{\boldsymbol{X}}, \mathcal{D}_{\mathsf{Circ}})$-pseudorandomness style puncturing security (see Definition 7.1). Then, assuming the existence of polynomially secure iO and the LWE assumption, then there exists a copy-protection scheme for Circ that satisfies $(\mathcal{D}_{\boldsymbol{X}} \times \mathcal{D}_{\boldsymbol{X}} \times \mathcal{D}_{\mathsf{Circ}})$-oracular-unpredictable-style-CP anti-piracy (see Definition 3.1). Moreover, if $\mathcal{D}_{\boldsymbol{X}} = \mathsf{Uniform}_{\boldsymbol{X}}$, then we can replace the LWE assumption with the existence of OWFs.*

9 Summary of Results

To summarize, we get the following feasibility results for various kinds of copy-protection. Let $\mathbf{X} = \{0,1\}^{n(\lambda)}$ or in short $\mathbf{X} = \{0,1\}^n$ be the input space.

1. **Corollary 7.7:** Assuming the existence of polynomially secure iO and the LWE assumption, for any puncturable circuit class $(\mathsf{Circ}, \mathsf{Puncture})$ satisfying 2-point m-bit $(\mathcal{D}_{\mathbf{X}}, \mathcal{D}_{\mathsf{Circ}})$-pseudorandomness-style puncturing security (Definition 7.1), there exists a copy-protection scheme that satisfies $(\mathcal{D}_{\mathsf{Circ}} \times \mathcal{D}_{\mathbf{X}} \times \mathcal{D}_{\mathbf{X}})$-oracular-pseudorandomness-style CP+ anti-piracy (Definition 3.4). Moreover, if $\mathcal{D}_{\mathbf{X}} = \mathsf{Uniform}_{\mathbf{X}}$, then we can replace the LWE assumption with the existence of OWFs.

2. **Corollary 8.2:** Assuming the existence of polynomially secure iO and the LWE assumption, for any puncturable circuit class $(\mathsf{Circ}, \mathsf{Puncture})$ satisfying 2-point m-bit $(\mathcal{D}_{\mathbf{X}}, \mathcal{D}_{\mathsf{Circ}})$-unpredictability-style puncturing security (see Definition 7.1), there exists a copy-protection scheme that satisfies $(\mathcal{D}_{\mathsf{Circ}} \times \mathcal{D}_{\mathbf{X}} \times \mathcal{D}_{\mathbf{X}})$-unpredictability-style CP anti-piracy (see Definition 3.1). Moreover, if $\mathcal{D}_{\mathbf{X}} = \mathsf{Uniform}_{\mathbf{X}}$, then we can replace the LWE assumption with the existence of OWFs.

3. **Corollary 8.15:** For $k \geq 2$, assuming the existence of polynomially secure iO and the OWFs, there exists a copy-protection scheme for Circ^k, a circuit implementation of set of all k-point functionalities, that satisfies oracular-$\mathcal{D}^{\mathit{Evasive}}_{\mathbf{X}^2, \mathsf{Circ}^k}$-unpredictable-style anti-piracy (see Definition 3.1), where $\mathcal{D}^{\mathit{Evasive}}_{\mathbf{X}^2, \mathsf{Circ}^k}$ is defined as: $S, x_1, x_2 \leftarrow \mathcal{D}^{\mathit{Evasive}}_{\mathbf{X}^2, \mathsf{Circ}}(1^\lambda)$, where $S \xleftarrow{\$} \binom{\mathbf{X}}{k}$ and $x_1, x_2 \leftarrow \mathcal{D}^{\mathit{Evasive}}_{\mathbf{X}^2}(S)$, where each $S \in \binom{\mathbf{X}}{k}$ is an index for a circuit in Circ^k, and $\mathcal{D}^{\mathit{Evasive}}_{\mathbf{X}^2}$ is as defined in Definition 8.7.

4. **Corollary 8.20:** Assuming the existence of polynomially secure iO and the LWE assumption, then for any circuit class $(\mathsf{Circ}, \mathsf{Puncture})$ satisfying 1-point 1-bit $(\mathcal{D}_{\mathbf{X}}, \mathcal{D}_{\mathsf{Circ}})$-pseudorandomness style puncturing security (see Definition 7.1), there exists a copy-protection scheme that satisfies $(\mathcal{D}_{\mathbf{X}} \times \mathcal{D}_{\mathbf{X}} \times \mathcal{D}_{\mathsf{Circ}})$-oracular-unpredictable-style-CP anti-piracy (see Definition 3.1). Moreover, if $\mathcal{D}_{\mathbf{X}} = \mathsf{Uniform}_{\mathbf{X}}$, then we can replace the LWE assumption with the existence of OWFs.

Acknowledgments

Prabhanjan Ananth is supported by the National Science Foundation under the grants FET-2329938, CAREER-2341004 and, FET-2530160. Amit Behera was partially funded by the Israel Science Foundation (grant No. 2527/24) and the European Union (ERC-2022-COG, ACQUA, 101087742). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

References

[Aar09a] 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. (Cited on page 3.)

[Aar09b] Scott Aaronson. Quantum copy-protection and quantum money. In 2009 24th Annual IEEE Conference on Computational Complexity, pages 229–242. IEEE, 2009. (Cited on page 5.)

[AB24] Prabhanjan Ananth and Amit Behera. A modular approach to unclonable cryptography. In Leonid Reyzin and Douglas Stebila, editors, CRYPTO 2024, Part VII, volume 14926 of LNCS, pages 3–37. Springer, Cham, August 2024. (Cited on page 3, 4, 5, 6, 11, 12, 13, 22, 23, 24, 42, 43, 52.)

[ABH25] Prabhanjan Ananth, Amit Behera, and Zikuan Huang. Copy-protection from UPO, revisited. Cryptology ePrint Archive, Paper 2025/1207, 2025. (Cited on page 1.)

[AC02] Mark Adcock and Richard Cleve. A quantum goldreich-levin theorem with cryptographic applications. In Helmut Alt and Afonso Ferreira, editors, STACS 2002, 19th Annual Symposium on Theoretical Aspects of Computer Science, Antibes - Juan les Pins, France, March 14-16, 2002, Proceedings, volume 2285 of Lecture Notes in Computer Science, pages 323–334. Springer, 2002. (Cited on page 16.)

[AK22] Prabhanjan Ananth and Fatih Kaleoglu. A note on copy-protection from random oracles. arXiv preprint arXiv:2208.12884, 2022. (Cited on page 3.)

[AKL+22] Prabhanjan Ananth, Fatih Kaleoglu, Xingjian Li, Qipeng Liu, and Mark Zhandry. On the feasibility of unclonable encryption, and more. In Yevgeniy Dodis and Thomas Shrimpton, editors, CRYPTO 2022, Part II, volume 13508 of LNCS, pages 212–241. Springer, Cham, August 2022. (Cited on page 3, 5, 7, 26.)

[AKL23] Prabhanjan Ananth, Fatih Kaleoglu, and Qipeng Liu. Cloning games: A general framework for unclonable primitives. In Helena Handschuh and Anna Lysyanskaya, editors, CRYPTO 2023, Part V, volume 14085 of LNCS, pages 66–98. Springer, Cham, August 2023. (Cited on page 3, 5, 16.)

[AKY25] Prabhanjan Ananth, Fatih Kaleoglu, and Henry Yuen. Simultaneous haar indistinguishability with applications to unclonable cryptography. In ITCS 2025, 2025. (Cited on page 16, 17.)

[ALL+21a] Scott Aaronson, Jiahui Liu, Qipeng Liu, Mark Zhandry, and Ruizhe Zhang. New approaches for quantum copy-protection. In Annual International Cryptology Conference, pages 526–555. Springer, 2021. (Cited on page 3.)

[ALL+21b] Scott Aaronson, Jiahui Liu, Qipeng Liu, Mark Zhandry, and Ruizhe Zhang. New approaches for quantum copy-protection. In Tal Malkin and Chris Peikert, editors, CRYPTO 2021, Part I, volume 12825 of LNCS, pages 526–555, Virtual Event, August 2021. Springer, Cham. (Cited on page 3, 7.)

[AP20] Shweta Agrawal and Alice Pellet-Mary. Indistinguishability obfuscation without maps: Attacks and fixes for noisy linear FE. In Anne Canteaut and Yuval Ishai, editors, EUROCRYPT 2020, Part I, volume 12105 of LNCS, pages 110–140. Springer, Cham, May 2020. (Cited on page 15.)

[AP21] Prabhanjan Ananth and Rolando L La Placa. Secure software leasing. In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pages 501–530. Springer, 2021. (Cited on page 3.)

[BBBV97a] Charles H Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. SIAM journal on Computing, 26(5):1510–1523, 1997. (Cited on page 14.)

[BBBV97b] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh V. Vazirani. Strengths and weaknesses of quantum computing. SIAM J. Comput., 26(5):1510–1523, 1997. (Cited on page 12, 13.)

[BBV24] James Bartusek, Zvika Brakerski, and Vinod Vaikuntanathan. Quantum state obfuscation from classical oracles. In Proceedings of the 56th Annual ACM Symposium on Theory of Computing, pages 1009–1017, 2024. (Cited on page 4.)

[BCP14] Elette Boyle, Kai-Min Chung, and Rafael Pass. On extractability obfuscation. In Yehuda Lindell, editor, TCC 2014, volume 8349 of LNCS, pages 52–73. Springer, Berlin, Heidelberg, February 2014. (Cited on page 26, 56.)

[BDGM20] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Candidate iO from homomorphic encryption schemes. In Anne Canteaut and Yuval Ishai, editors, EUROCRYPT 2020, Part I, volume 12105 of LNCS, pages 79–109. Springer, Cham, May 2020. (Cited on page 15.)

[BDGM22] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Factoring and pairings are not necessary for IO: Circular-secure LWE suffices. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, ICALP 2022, volume 229 of LIPIcs, pages 28:1–28:20. Schloss Dagstuhl, July 2022. (Cited on page 15.)

[BDJ+24] Pedro Branco, Nico Döttling, Abhishek Jain, Giulio Malavolta, Surya Mathialagan, Spencer Peters, and Vinod Vaikuntanathan. Pseudorandom obfuscation and applications. Cryptology ePrint Archive, Paper 2024/1742, 2024. (Cited on page 15.)

[BGI+12] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. Journal of the ACM, 59(2):6:1–6:48, 2012. (Cited on page 15.)

[BGI14] Elette Boyle, Shafi Goldwasser, and Ioana Ivan. Functional signatures and pseudorandom functions. In Hugo Krawczyk, editor, PKC 2014, volume 8383 of LNCS, pages 501–519. Springer, Berlin, Heidelberg, March 2014. (Cited on page 16.)

[BGK+24] James Bartusek, Vipul Goyal, Dakshita Khurana, Giulio Malavolta, Justin Raizes, and Bhaskar Roberts. Software with certified deletion. In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pages 85–111. Springer, 2024. (Cited on page 3.)

[BGMZ18] James Bartusek, Jiaxin Guan, Fermi Ma, and Mark Zhandry. Return of GGH15: Provable security against zeroizing attacks. In Amos Beimel and Stefan Dziembowski, editors, TCC 2018, Part II, volume 11240 of LNCS, pages 544–574. Springer, Cham, November 2018. (Cited on page 15.)

[BPW16] Nir Bitansky, Omer Paneth, and Daniel Wichs. Perfect structure on the edge of chaos - trapdoor permutations from indistinguishability obfuscation. In Eyal Kushilevitz and Tal Malkin, editors, TCC 2016-A, Part I, volume 9562 of LNCS, pages 474–502. Springer, Berlin, Heidelberg, January 2016. (Cited on page 16, 31, 39, 55.)

[BW13] Dan Boneh and Brent Waters. Constrained pseudorandom functions and their applications. In Kazue Sako and Palash Sarkar, editors, ASIACRYPT 2013, Part II, volume 8270 of LNCS, pages 280–300. Springer, Berlin, Heidelberg, December 2013. (Cited on page 4, 16.)

[ÇG24a] Alper Çakan and Vipul Goyal. Unclonable cryptography with unbounded collusions and impossibility of hyperefficient shadow tomography. In TCC, 2024. (Cited on page 3.)

[CG24b] Andrea Coladangelo and Sam Gunn. How to use quantum indistinguishability obfuscation. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, 56th ACM STOC, pages 1003–1008. ACM Press, June 2024. (Cited on page 3, 4, 42, 43.)

[ÇG25a] Alper Çakan and Vipul Goyal. How to copy-protect all puncturable functionalities without conjectures: A unified solution to quantum protection. Cryptology ePrint Archive, Paper 2025/1197, 2025. (Cited on page 14.)

[ÇG25b] Alper Çakan and Vipul Goyal. How to copy-protect malleable-puncturable cryptographic functionalities under arbitrary challenge distributions. Cryptology ePrint Archive, Paper 2025/1357, 2025. (Cited on page 14.)

[CHV23] Céline Chevalier, Paul Hermouet, and Quoc-Huy Vu. Semi-quantum copy-protection and more. Cryptology ePrint Archive, Report 2023/244, 2023. (Cited on page 3, 5.)

[CHVW19] Yilei Chen, Minki Hhan, Vinod Vaikuntanathan, and Hoeteck Wee. Matrix PRFs: Constructions, attacks, and applications to obfuscation. In Dennis Hofheinz and Alon Rosen, editors, TCC 2019, Part I, volume 11891 of LNCS, pages 55–80. Springer, Cham, December 2019. (Cited on page 15.)

[CLLZ21] Andrea Coladangelo, Jiahui Liu, Qipeng Liu, and Mark Zhandry. Hidden cosets and applications to unclonable cryptography. In Tal Malkin and Chris Peikert, editors, CRYPTO 2021, Part I, volume 12825 of LNCS, pages 556–584, Virtual Event, August 2021. Springer, Cham. (Cited on page 3, 4, 7, 16, 19, 26, 27, 28, 30, 42.)

[CLW25] Valerio Cini, Russell W. F. Lai, and Ivy K. Y. Woo. Lattice-based obfuscation from NTRU and equivocal LWE. Cryptology ePrint Archive, Paper 2025/1129, 2025. (Cited on page 15.)

[CMP20a] Andrea Coladangelo, Christian Majenz, and Alexander Poremba. Quantum copy-protection of compute-and-compare programs in the quantum random oracle model. arXiv (CoRR), abs/2009.13865, 2020. (Cited on page 3, 5.)

[CMP20b] Andrea Coladangelo, Christian Majenz, and Alexander Poremba. Quantum copy-protection of compute-and-compare programs in the quantum random oracle model. arXiv preprint arXiv:2009.13865, 2020. (Cited on page 3.)

[CV22] Eric Culf and Thomas Vidick. A monogamy-of-entanglement game for subspace coset states. Quantum, 6:791, sep 2022. (Cited on page 7, 27, 28.)

[DQV+21] Lalita Devadas, Willy Quach, Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. Succinct LWE sampling, random polynomials, and obfuscation. In Kobbi Nissim and Brent Waters, editors, TCC 2021, Part II, volume 13043 of LNCS, pages 256–287. Springer, Cham, November 2021. (Cited on page 15.)

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. Journal of the ACM, 33(4):792–807, 1986. (Cited on page 16.)

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In 21st ACM STOC, pages 25–32. ACM Press, May 1989. (Cited on page 74.)

[GP21] Romain Gay and Rafael Pass. Indistinguishability obfuscation from circular security. In Samir Khuller and Virginia Vassilevska Williams, editors, 53rd ACM STOC, pages 736–749. ACM Press, June 2021. (Cited on page 15.)

[HJL25] Yao-Ching Hsieh, Aayush Jain, and Huijia Lin. Lattice-based post-quantum iO from circular security with random opening assumption (part II: zeroizing attacks against private-coin evasive LWE assumptions). Cryptology ePrint Archive, Paper 2025/390, 2025. (Cited on page 15.)

[KPTZ13] Aggelos Kiayias, Stavros Papadopoulos, Nikos Triandopoulos, and Thomas Zacharias. Delegatable pseudorandom functions and applications. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, ACM CCS 2013, pages 669–684. ACM Press, November 2013. (Cited on page 16.)

[KY25a] Fuyuki Kitagawa and Takashi Yamakawa. Copy protecting cryptographic functionalities over entropic inputs. Cryptology ePrint Archive, Paper 2025/1264, 2025. (Cited on page 1.)

[KY25b] Fuyuki Kitagawa and Takashi Yamakawa. Foundations of single-decryptor encryption. Cryptology ePrint Archive, Paper 2025/1219, 2025. (Cited on page 5, 7, 9, 10, 17, 19, 20, 23, 24, 27, 42.)

[LLQZ22] Jiahui Liu, Qipeng Liu, Luowen Qian, and Mark Zhandry. Collusion resistant copy-protection for watermarkable functionalities. In Eike Kiltz and Vinod Vaikuntanathan, editors, TCC 2022, Part I, volume 13747 of LNCS, pages 294–323. Springer, Cham, November 2022. (Cited on page 3.)

[PW11] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. SIAM Journal on Computing, 40(6):1803–1844, 2011. (Cited on page 16.)

[WW21] Hoeteck Wee and Daniel Wichs. Candidate obfuscation via oblivious LWE sampling. In Anne Canteaut and François-Xavier Standaert, editors, EUROCRYPT 2021, Part III, volume 12698 of LNCS, pages 127–156. Springer, Cham, October 2021. (Cited on page 15.)

[Zha19] Mark Zhandry. Quantum lightning never strikes the same state twice. In Yuval Ishai and Vincent Rijmen, editors, EUROCRYPT 2019, Part III, volume 11478 of LNCS, pages 408–438. Springer, Cham, May 2019. (Cited on page 30, 31.)

A Proof of Theorem 6.2

Here, we prove Theorem 6.2.

Proof of Theorem 6.2. It suffices to construct a scheme for single-bit messages since we can expand the message space by a simple bit-by-bit encryption (under independent keys for each bit). Let FGen be a generation algorithm for a keyed injective one-way function. Then we construct a key-robust non-committing encryption scheme with the message space $\{0,1\}$ and key space $\{0,1\}^{\ell_{\mathsf{inp}}}$ as follows.

$\mathsf{Gen}(1^\lambda)$: On input 1^λ, generate $\mathsf{F} \leftarrow \mathsf{FGen}(1^\lambda)$ and $x \leftarrow \{0,1\}^{\ell_{\mathsf{inp}}}$, and output $k := (\mathsf{F}, x)$.

$\mathsf{Enc}(k, m)$: On input (F, x) and $m \in \{0,1\}$, sample $x' \leftarrow \{0,1\}^{\ell_{\mathsf{inp}}}$ and $r_0, r_1 \leftarrow \{0,1\}^{\ell_{\mathsf{inp}}}$, sets

$$\mathsf{ct}_m := (\mathsf{F}(x), r_m, \langle r_m, x\rangle)$$

and

$$\mathsf{ct}_{m\oplus 1} := (\mathsf{F}(x'), r_{m\oplus 1}, \langle r_{m\oplus 1}, x'\rangle \oplus 1),$$

and output $\mathsf{ct} := (\mathsf{F}, \mathsf{ct}_0, \mathsf{ct}_1)$.

$\mathsf{Dec}(k, \mathsf{ct})$: On input k and $\mathsf{ct} = (\mathsf{F}, \mathsf{ct}_0, \mathsf{ct}_1)$, if the first component of k disagrees with F, output $\bot$. Otherwise, parse $k = (\mathsf{F}, x)$ and $\mathsf{ct}_b = (y_b, r_b, \beta_b)$ for $b \in \{0,1\}$. If there is unique $b \in \{0,1\}$ such that $y_b = \mathsf{F}(x)$ and $\beta_b = \langle r_b, x\rangle$, then output b, and output $\bot$ otherwise.

$\mathsf{Fake}(1^\lambda)$: On input 1^λ, generate $\mathsf{F} \leftarrow \mathsf{FGen}(1^\lambda)$, sample $x_0, x_1 \leftarrow \{0,1\}^{\ell_{\mathsf{inp}}}$ and $r_0, r_1 \leftarrow \{0,1\}^{\ell_{\mathsf{inp}}}$, sets

$$\mathsf{ct}^*_b := (\mathsf{F}(x_b), r_b, \langle r_b, x_b\rangle)$$

for $b \in \{0,1\}$, and output $\mathsf{ct}^* := (\mathsf{F}, \mathsf{ct}^*_0, \mathsf{ct}^*_1)$ and $\mathsf{st} := (\mathsf{F}, x_0, x_1)$.

$\mathsf{Open}(\mathsf{st}, m)$: On input st and $m \in \{0,1\}$, parse $\mathsf{st} = (\mathsf{F}, x_0, x_1)$ and output $k^* := (\mathsf{F}, x_m)$.

Correctness and key-robustness are clear from the injectivity of FGen. Below, we show the non-committing property. Fix $m \in \{0,1\}$ and consider the following hybrids of distributions.

D_0**:** This corresponds to the distribution in the LHS of the non-committing property. That is, it samples as follows: Generate $\mathsf{F} \leftarrow \mathsf{FGen}(1^\lambda)$, sample $x, x' \leftarrow \{0,1\}^{\ell_{\mathsf{inp}}}$, and $r_0, r_1 \leftarrow \{0,1\}^{\ell_{\mathsf{inp}}}$, set $\mathsf{ct}_m := (\mathsf{F}(x), r_m, \langle r_m, x\rangle)$ and $\mathsf{ct}_{m\oplus 1} := (\mathsf{F}(x'), r_{m\oplus 1}, \langle r_{m\oplus 1}, x'\rangle \oplus 1)$, and output $(\mathsf{ct} = (\mathsf{F}, \mathsf{ct}_0, \mathsf{ct}_1), k = (\mathsf{F}, x))$.

D_1**:** This is identical to D_0 except that $\mathsf{ct}_{m\oplus 1}$ is generated as $\mathsf{ct}_{m\oplus 1} := (\mathsf{F}(x'), r_{m\oplus 1}, \langle r_{m\oplus 1}, x'\rangle)$, By the one-wayness of FGen and the Goldreich-Levin theorem [GL89], $D_1 \overset{\mathsf{c}}{\approx} D_2$.

D_2**:** This corresponds to the distribution in the LHS of the non-committing property. That is, it samples as follows: Generate $\mathsf{F} \leftarrow \mathsf{FGen}(1^\lambda)$, sample $x_0, x_1 \leftarrow \{0,1\}^{\ell_{\mathsf{inp}}}$, and $r_0, r_1 \leftarrow \{0,1\}^{\ell_{\mathsf{inp}}}$, set $\mathsf{ct}_b := (\mathsf{F}(x_b), r_b, \langle r_b, x_b\rangle)$ for $b \in \{0,1\}$, and output $(\mathsf{ct} = (\mathsf{ct}_0, \mathsf{ct}_1), k = (\mathsf{F}, x_m))$.

It is easy to verify that D_1 and D_2 are identical: $D_1 \equiv D_2$.

Combining above, $D_0 \overset{\mathsf{c}}{\approx} D_2$, and thus the non-committing property is proven. □