A Modular Approach to Unclonable Cryptography

Prabhanjan Ananth[1(✉)] and Amit Behera[2]

[1] University of California Santa Barbara, Santa Barbara, USA
prabhanjan@cs.ucsb.edu
[2] Ben-Gurion University, Be'er Sheva, Israel
behera@post.bgu.ac.il

Abstract. We explore a new pathway to designing unclonable cryptographic primitives. We propose a new notion called unclonable puncturable obfuscation (UPO) and study its implications for unclonable cryptography. Using UPO, we present modular (and in some cases, arguably, simple) constructions of many primitives in unclonable cryptography, including, public-key quantum money, quantum copy-protection for many classes of functionalities, unclonable encryption, and single-decryption encryption.

Notably, we obtain the following new results assuming the existence of UPO:

- We show that any cryptographic functionality can be copy-protected as long as it satisfies a notion of security, which we term puncturable security. Prior feasibility results focused on copy-protecting specific cryptographic functionalities.
- We show that copy-protection exists for any class of evasive functions as long as the associated distribution satisfies a preimage-sampleability condition. Prior works demonstrated copy-protection for point functions, which follows as a special case of our result.

We put forward two constructions of UPO. The first construction satisfies two notions of security based on the existence of (post-quantum) sub-exponentially secure indistinguishability obfuscation, injective one-way functions, the quantum hardness of learning with errors, and the two versions of a new conjecture called the simultaneous inner product conjecture. The security of the second construction is based on the existence of unclonable-indistinguishable bit encryption, injective one-way functions, and quantum-state indistinguishability obfuscation.

1 Introduction

Unclonable cryptography leverages the no-cloning principle of quantum mechanics [WZ82,Die82] to build many novel cryptographic notions that are otherwise impossible to achieve classically. This has been an active area of interest since the 1980s [Wie83]. In the past few years, researchers have investigated a dizzying variety of unclonable primitives such as quantum money [AC12,Zha19,

L. Reyzin and D. Stebila (Eds.): CRYPTO 2024, LNCS 14926, pp. 3–37, 2024.
https://doi.org/10.1007/978-3-031-68394-7_1

Shm22,LMZ23] and its variants [RS19,BS20,RZ21], quantum one-time programs [BGS13], copy-protection [Aar09,CLLZ21], tokenized signatures [BS16, CLLZ21], unclonable encryption [Got02,BL20] and its variants [KN23], secure software leasing [AL21], single-decryptor encryption [GZ20,CLLZ21], and many more [BKL23,GMR23,JK23].

Establishing the feasibility of unclonable primitives has been quite challenging. The adversarial structure considered in the unclonability setting (i.e., spatially separated and entangled) is quite different from what we typically encounter in the traditional cryptographic setting. This makes it difficult to leverage traditional classical techniques, commonly used in cryptographic proofs, to argue the security of unclonable primitives. As a result, there are two major gaping holes in the area.

- Unsolved Foundational Questions: Despite the explosion of results in the past few years, many fundamental questions in this area remain to be solved. This includes designing public-key quantum money schemes [AC12, Zha19] on well-studied assumptions. Another problem that is open is precisely characterizing the class of functionalities for which quantum copy-protection [Aar09] is possible.
- Lack of Abstractions: Due to the lack of good abstractions, proofs in the area of unclonable cryptography tend to be complex and use sophisticated tools, making the literature less accessible to the broader research community. This makes not only verification of proofs difficult but also makes it harder to use the techniques to obtain new feasibility results.

Overarching Goal of Our Work. We advocate for a modular approach to designing unclonable cryptography. Our goal is to identify an important unclonable cryptographic primitive that would serve as a useful abstraction leading to the design of other unclonable primitives. Ideally, we would like to abstract away all the complex details in the instantiation of this primitive, and it should be relatively easy, even to classical cryptographers, to use this primitive to study unclonability in the context of other cryptographic primitives. We believe that the identification and instantiation of such a primitive will speed up the progress in the design of unclonable primitives.

Indeed, similar explorations in other contexts, such as classical cryptography, have been fruitful. For instance, the discovery of indistinguishability obfuscation [BGI+01,GGH+16] (iO) revolutionized cryptography and led to the resolution of many open problems (for instance: [SW14,GGHR14,BZ17,BPR15]). Hence, there is merit to exploring the possibility of such a primitive in unclonable cryptography, as well.

Thus, we ask the following question:

Is there an "iO-like" primitive for unclonable cryptography?

We seek the pursuit of identifying unclonable primitives that would have a similar impact on unclonable cryptography as iO did on classical cryptography.

1.1 Our Contributions in a Nutshell

In our search for an *"iO-like"* primitive for unclonable cryptography, we propose a new notion called *unclonable puncturable obfuscation* (UPO) and explore its impact on unclonable cryptography.

New Feasibility Results. Specifically, using UPO and other well-studied cryptographic tools, we demonstrate the following new results.

- We show that any class of functionalities can be copy-protected as long as they are puncturable (more details in Sect. 1.2).
- We show that a large class of evasive functionalities can be copy-protected.

The above two results not only subsume all the copy-protectable functionalities studied in prior works but also capture new functionalities.

Even for functionalities that have been studied before our work, we get qualitatively new results. For instance, our result shows that **any** puncturable digital signature can be copy-protected whereas the work of [LLQZ22] shows a weaker result that the digital signature of [SW14] can be copy-protected. We get similar conclusions for copy-protection for pseudorandom functions.

Implication to Unclonable Cryptography. Apart from quantum copy-protection, UPO implies many of the foundational unclonable primitives such as public-key quantum money, unclonable encryption, and single-decryptor encryption. *The resulting constructions from UPO are conceptually different compared to the prior works.* Since building unclonable primitives is a daunting task even when relying on exotic computational assumptions, it becomes crucial to venture into alternative approaches. Moreover, this endeavor could potentially yield fresh perspectives on unclonable cryptography.

Simpler Constructions. We believe that some of our constructions are simpler than the prior works, albeit the underlying assumptions are incomparable[1]. The construction of copy-protection for puncturable functionalities yields simpler constructions of copy-protection for pseudorandom functions, studied in [CLLZ21], and copy-protection for signatures, studied in [LLQZ22].

One potential criticism of our work is that our construction of UPO is based on a new conjecture. Specifically, we show that UPO can be based on the existence of post-quantum secure iO, learning with errors and a new conjecture.

However, it is essential to keep in mind the following facts:

- Assumptions: If our conjectures are true, then this would mean that we can construct UPO from indistinguishability obfuscation and other standard assumptions. On the other hand, we currently do not know whether the other direction is true, i.e., whether UPO implies post-quantum indistinguishability obfuscation. As a result, it is plausible that UPO could be a weaker

[1] We assume UPO whereas the previous works assume post-quantum iO and other well-studied assumptions.

assumption than post-quantum iO! One consequence of this is the construction of public-key quantum money from generic assumptions weaker than post-quantum iO.
If our conjectures are false, by itself, this does not refute the existence of UPO. *We would like to emphasize that there is no reason to believe these conjectures are necessary for the existence of UPO.* Instead, it merely suggests that we need a different approach to investigate the feasibility of UPO.

– Pushing the Feasibility Landscape: Time and time again, in cryptography, we have been forced to invent new assumptions. In numerous instances, these assumptions have unveiled a previously uncharted realm of cryptographic primitives, expanding our understanding beyond what we once deemed feasible. While not all of the computational assumptions have survived the test of time, in some cases[2], the insights gained from their cryptanalysis have helped us to come up with more secure instantiations in the future. In a similar vein, being aggressive with exploring new assumptions could push the boundaries of unclonable cryptography.

We also present another construction of UPO from quantum state iO and unclonable encryption. We discuss this more at the end of Sect. 1.2.

1.2 Our Contributions

Definition. We discuss our results in more detail. Roughly speaking, unclonable puncturable obfuscation (UPO) defined for a class of circuits $\mathfrak{C}$ in $\mathsf{P/Poly}$, consists of two QPT algorithms $(\mathsf{Obf}, \mathsf{Eval})$ defined as follows:

– Obfuscation algorithm: Obf takes as input a classical circuit $C \in \mathfrak{C}$ and outputs a quantum state ρ_C.
– Evaluation algorithm: Eval takes as input a quantum state ρ_C, an input x, and outputs a value y.

In terms of correctness, we require $y = C(x)$. To define security, as is typically the case for unclonable primitives, we consider non-local adversaries of the form $(\mathcal{A}, \mathcal{B}, \mathcal{C})$. The security experiment, parameterized by a distribution $\mathcal{D}_{\mathcal{X}}$, is defined as follows:

– $\mathcal{A}$ (Alice) receives as input a quantum state ρ^* that is generated as follows. $\mathcal{A}$ sends a circuit C to the challenger, who then samples a bit b uniformly at random and samples $\left(x^{\mathcal{B}}, x^{\mathcal{C}}\right)$ from $\mathcal{D}_{\mathcal{X}}$. If $b = 0$, it sets ρ^* to be the output of Obf on input C, or if $b = 1$, it sets ρ^* to be the output of Obf on G, where G is a punctured circuit that has the same functionality as C on all the points except $x^{\mathcal{B}}$ and $x^{\mathcal{C}}$. It is important to note that $\mathcal{A}$ only receives ρ^* and in particular, $x^{\mathcal{B}}$ and $x^{\mathcal{C}}$ are hidden from $\mathcal{A}$.
– $\mathcal{A}$ then creates a bipartite state and shares one part with $\mathcal{B}$ (Bob) and the other part with $\mathcal{C}$ (Charlie).

[2] Several candidates of post-quantum indistinguishability obfuscation had to be broken before a candidate based on well founded assumptions was proposed [JLS21].

– $\mathcal{B}$ and $\mathcal{C}$ cannot communicate with each other. In the challenge phase, $\mathcal{B}$ receives $x^{\mathcal{B}}$ and $\mathcal{C}$ receives $x^{\mathcal{C}}$. Then, they each output bits $b_{\mathcal{B}}$ and $b_{\mathcal{C}}$.

$(\mathcal{A}, \mathcal{B}, \mathcal{C})$ win if $b_{\mathcal{B}} = b_{\mathcal{C}} = b$. The scheme is secure if they can only win with probability at most 0.5 (ignoring negligible additive factors).

Keyed Circuits. Towards formalizing the notion of puncturing circuits in a way that will be useful for applications, we consider keyed circuit classes in the above definition. Every circuit in a keyed circuit class is of the form $C_k(\cdot)$ for some key k. Any circuit class can be implemented as a keyed circuit class using universal circuits and thus, by considering keyed circuits, we are not compromising on the generality of the above definition.

Challenge Distributions. We could consider different settings of $\mathcal{D}_{\mathcal{X}}$. In this work, we mainly focus on two settings. In the first setting (referred to as *independent* challenge distribution), sampling $(x^{\mathcal{B}}, x^{\mathcal{C}})$ from $\mathcal{D}_{\mathcal{X}}$ is the same as sampling $x^{\mathcal{B}}$ and $x^{\mathcal{C}}$ uniformly at random (from the input space of C). In the second setting (referred to as *identical* challenge distribution), sampling $(x^{\mathcal{B}}, x^{\mathcal{C}})$ from $\mathcal{D}_{\mathcal{X}}$ is the same as sampling x uniformly at random and setting $x = x^{\mathcal{B}} = x^{\mathcal{C}}$.

Generalized UPO. In the above security experiment, we did not quite specify the behavior of the punctured circuit on the points $x^{\mathcal{B}}$ and $x^{\mathcal{C}}$. There are two ways to formalize and this results in two different definitions; we consider both of them in Sect. 2. In the first (basic) version, the output of the punctured circuit G on the punctured points is set to be $\perp$. This version would be the regular UPO definition. In the second (generalized) version, we allow $\mathcal{A}$ to control the output of the punctured circuit on inputs $x^{\mathcal{B}}$ and $x^{\mathcal{C}}$. For instance, $\mathcal{A}$ can choose and send the circuits $\mu_{\mathcal{B}}$ and $\mu_{\mathcal{C}}$ to the challenger. On input $x^{\mathcal{B}}$ (resp., $x^{\mathcal{C}}$), the challenger programs the punctured circuit G to output $\mu_{\mathcal{B}}(x^{\mathcal{B}})$ (resp., $\mu_{\mathcal{C}}(x^{\mathcal{C}})$). We refer to this version as *generalized UPO*.

Applications. We demonstrate several applications of UPO to unclonable cryptography.

We summarise the applications[3] in Fig. 1. For a broader context of these results, we refer the reader to related works section in the full version.

Copy-Protection for Puncturable Cryptographic Schemes (see the relevant sections in the full-version). We consider cryptographic schemes satisfying a property called puncturable security. Informally speaking, puncturable security says the following: given a secret key sk, generated using the scheme, it is possible to puncture the key at a couple of points $x^{\mathcal{B}}$ and $x^{\mathcal{C}}$ such that it is computationally infeasible to use the punctured secret key on $x^{\mathcal{B}}$ and $x^{\mathcal{C}}$. We formally define this in the full version. We show the following:

[3] We refer the reader unfamiliar with copy-protection, single-decryptor encryption, or unclonable encryption to the introduction section of [AKL23] for an informal explanation of these primitives.

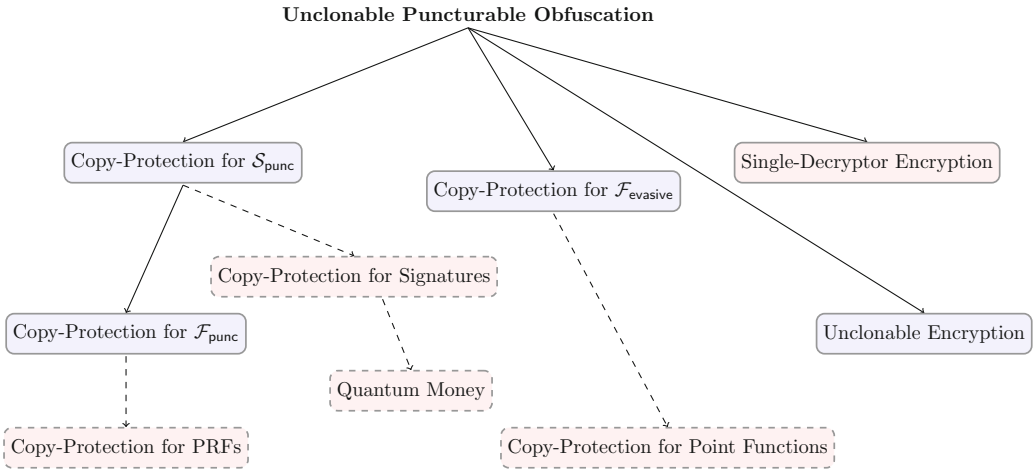

Fig. 1. Applications of Unclonable Puncturable Obfuscation. $\mathcal{S}_{\mathsf{punc}}$ denotes cryptographic schemes satisfying puncturable property. $\mathcal{F}_{\mathsf{punc}}$ denotes cryptographic functionalities satisfying functionalities satisfying puncturable property. $\mathcal{F}_{\mathsf{evasive}}$ denotes functionalities that are evasive with respect to a distribution $\mathcal{D}$ satisfying preimage-sampleability property. The dashed lines denote corollaries of our main results. The blue-filled boxes represent primitives whose feasibility was unknown prior to our work. The red-filled boxes represent primitives for which we get qualitatively different results or from incomparable assumptions when compared to previous works. (Color figure online)

Theorem 1. *Assuming UPO for* P/poly*, there exists copy-protection for any puncturable cryptographic scheme.*

Prior works [CLLZ21, LLQZ22] aimed at copy-protecting specific cryptographic functionalities whereas we, for the first time, characterize a broad class of cryptographic functionalities that can be copy-protected.

As a corollary, we obtain the following results assuming UPO.

– We show that **any** class of puncturable pseudorandom functions that can be punctured at two points [BW13, BGI14] can be copy-protected. The feasibility result of copy-protecting pseudorandom functions was first established in [CLLZ21]. A point to note here is that in [CLLZ21], given a class of puncturable pseudorandom functions, they transform this into a different class of pseudorandom functions[4] that is still puncturable and then copy-protect the resulting class. On the other hand, we show that *any* class of puncturable pseudorandom functions, which allows for the puncturing of two points, can be copy-protected. Hence, our result is qualitatively different than [CLLZ21].

[4] Specifically, they add a transformation to generically make the pseudorandom function extractable.

– We show that **any** digital signature scheme, where the signing key can be punctured at two points, can be copy-protected. Roughly speaking, a digital signature scheme is puncturable at two points if the signing key can be punctured on two messages $m^{\mathcal{B}}$ and $m^{\mathcal{C}}$ such that given the punctured signing key, it is computationally infeasible to produce a signature on one of the punctured messages. Our result rederives and generalizes a recent result by [LLQZ22] who showed how to copy-protect the digital signature scheme of [SW14].

In the technical sections, we first present a simpler result where we copy-protect puncturable functionalities (for more details, see the full-version) and we then extend this result to achieve copy-protection for puncturable cryptographic schemes (for more details, see the full-version).

Copy-Protection for Evasive Functions. We consider a class of evasive functions associated with a distribution $\mathcal{D}$ satisfying a property referred to as preimage-sampleability which is informally defined as follows: there exists a distribution $\mathcal{D}'$ such that sampling an evasive function from $\mathcal{D}$ along with an accepting point (i.e., the output of the function on this point is 1) is computationally indistinguishable from sampling a function from $\mathcal{D}'$ and then modifying this function by injecting a uniformly random point as the accepting point. We show the following.

Theorem 2. *Assuming generalized UPO for* P/poly*, there exists copy-protection for any class of functions that is evasive with respect to a distribution* $\mathcal{D}$ *satisfying preimage-sampleability property.*

Unlike Theorem 1, we assume generalized UPO in the above theorem.

As a special case, we obtain copy-protection for point functions. A recent work [CHV23] presented construction of copy-protection for point functions from post-quantum iO and other standard assumptions. Qualitatively, our results are different in the following ways:

– The challenge distribution considered in the security definition of [CHV23] is arguably not a natural one: with probability $\frac{1}{3}$, $\mathcal{B}$ and $\mathcal{C}$ get as input the actual point, with probability $\frac{1}{3}$, $\mathcal{B}$ gets the actual point while $\mathcal{C}$ gets a random value and finally, with probability $\frac{1}{3}$, $\mathcal{B}$ gets a random value while $\mathcal{C}$ gets the actual point. On the other hand, we consider identical challenge distribution; that is, $\mathcal{B}$ and $\mathcal{C}$ both receive the actual point with probability $\frac{1}{2}$ or they both receive a value picked uniformly at random.
– While the result of [CHV23] is restricted to point functions, we show how to copy-protect functions where the number of accepting points is a fixed polynomial.

We clarify that none of the above results on copy-protection contradicts the impossibility result by [AL21] who present a conditional result ruling out the possibility of copy-protecting contrived functionalities.

Unclonable Encryption. Finally, we show, for the first time, an approach to construct unclonable encryption in the plain model. We give a direct and simple construction of unclonable encryption for bits, see the full version for more details.

Theorem 3. *Assuming generalized UPO for* P/poly*, there exists a one-time unclonable bit-encryption scheme in the plain model.*

We also obtain a construction of unclonable encryption for arbitrary fixed length messages by first constructing public-key single-decryptor encryption (SDE) with an identical challenge distribution.

Theorem 4. *Assuming generalized UPO for* P/poly*, post-quantum indistinguishability obfuscation (iO), and post-quantum injective one-way functions, there exists a public-key single-decryptor encryption scheme with security against identical challenge distribution, see the full version for more details.*

[GZ20] showed that SDE with such a challenge distribution implies unclonable encryption. Prior work by [CLLZ21] demonstrated the construction of public-key single-decryptor encryption with security against independent challenge distribution, which is not known to imply unclonable encryption. We, thus, obtain the following corollary.

Corollary 1. *Assuming generalized UPO, post-quantum iO, and post-quantum injective one-way functions*[5]*, there exists a one-time unclonable encryption scheme in the plain model.*

Note that using the compiler of [AK21], we can generically transform a one-time unclonable encryption into a public-key unclonable encryption in the plain model under the same assumptions as above.

We note that this is the first construction of unclonable encryption in the plain model. All the previous works [BL20, AKL+22, AKL23] construct unclonable encryption in the quantum random oracle model. The disadvantage of our construction is that they leverage computational assumptions whereas the previous works [BL20, AKL+22, AKL23] are information-theoretically secure.

Apart from unclonable encryption, single-decryptor encryption also implies public-key quantum money, thereby giving the following corollary.

Corollary 2. *Assuming generalized UPO, post-quantum iO, and post-quantum one-way functions, there exists a public-key quantum money scheme.*

[5] Unlike Theorem 4, we do not need injective one-way functions here because the [GZ20] construction of unclonable encryption from single-decryptor encryption only requires selectively secure single-decryptor encryption with the above-mentioned challenge distribution, which we construct in our work using *any* post-quantum one-way functions along with the other assumptions; see the full version for more details.

The construction of quantum money from UPO offers a conceptually different approach to constructing public-key quantum money in comparison with other quantum money schemes such as [Zha19, LMZ23, Zha23].

As an aside, we also present a lifting theorem that lifts a selectively secure single-decryptor encryption into an adaptively secure construction, assuming the existence of post-quantum iO. Such a lifting theorem was not known prior to our work.

Construction. Finally we demonstrate a construction of generalized UPO for all classes of efficiently computable keyed circuits. We show that the same construction is secure with respect to both identical and independent challenge distributions. Specifically, we show the following:

Theorem 5 (Informal). *Suppose $\mathfrak{C}$ consists of polynomial-sized keyed circuits. Assuming the following:*

- *Post-quantum sub-exponentially secure indistinguishability obfuscation for* $\mathsf{P/poly}$*,*
- *Post-quantum sub-exponentially secure injective one-way functions,*
- *Compute-and-compare obfuscation secure against QPT adversaries and,*
- *Simultaneous inner product conjecture.*

there exists a generalized UPO *with respect to identical $\mathcal{D}_{\mathcal{X}}$ for $\mathfrak{C}$.*

On the Simultaneous Inner Product Conjecture: There are two different versions of the simultaneous inner product conjecture (Conjecture 1 and Conjecture 2) we rely upon to prove the security of our construction with respect to identical and independent challenge distributions. At a high level, the simultaneous inner product conjecture states that two (possibly entangled) QPT adversaries (i.e., non-local adversaries) should be unsuccessful in distinguishing $(\mathbf{r}, \langle \mathbf{r}, \mathbf{x} \rangle + m)$ versus $(\mathbf{r}, \langle \mathbf{r}, \mathbf{x} \rangle)$, where $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_Q^n, \mathbf{x} \xleftarrow{\$} \mathbb{Z}_Q^n, m \xleftarrow{\$} \mathbb{Z}_Q$ for every prime $Q \geq 1$. Moreover, the adversaries receive as input a bipartite state ρ that could depend on $\mathbf{x}$ with the guarantee that it should be infeasible to recover $\mathbf{x}$. As mentioned above, we consider two different versions of the conjecture. In the first version (*identical*), both the adversaries get the same sample $(\mathbf{r}, \langle \mathbf{r}, \mathbf{x} \rangle)$ or they both get $(\mathbf{r}, \langle \mathbf{r}, \mathbf{x} \rangle + m)$. In the second version (*independent*), the main difference is that $\mathbf{r}$ and $\mathbf{x}$ are sampled independently for both adversaries. Weaker versions of this conjecture have been investigated and proven to be unconditionally true [AKL23, KT22]. We refer the reader to Sect. 3 for a detailed discussion on the conjectures.

Composition: Another contribution of ours is a composition theorem (see the full version for more details), where we show how to securely compose unclonable puncturable obfuscation with a functionality-preserving compiler. In more detail, we show the following. Suppose UPO is a secure unclonable puncturable obfuscation scheme and let $\mathsf{Compiler}$ be a functionality-preserving circuit compiler. We define another scheme UPO' such that the obfuscation algorithm of

UPO', on input a circuit C, first runs the circuit compiler on C to obtain $\widetilde{C}$ and then it runs the obfuscation of UPO on $\widetilde{C}$ and outputs the result. The evaluation process can be similarly defined. We show that the resulting scheme UPO' is secure as long as UPO is secure. Our composition result allows us to compose UPO with other primitives such as different forms of program obfuscation without compromising on security. We use our composition theorem in some of the applications discussed earlier.

1.3 Technical Overview

We give an overview of the techniques behind our construction of UPO and the applications of UPO. We start with applications.

Applications

Copy-Protecting Puncturable Cryptographic Schemes. We begin by exploring methods to copy-protect puncturable pseudorandom functions. Subsequently, we generalize this approach to achieve copy-protection for a broader class of puncturable cryptographic schemes.

Case Study: Puncturable Pseudorandom Functions. Let $\mathcal{F} = \{f_k(\cdot) : \{0,1\}^n \rightarrow \{0,1\}^m \ : \ k \in \mathcal{K}_\lambda\}$ be a puncturable pseudorandom function (PRF) with λ being the security parameter and $\mathcal{K}_\lambda$ being the key space. To copy-protect $f_k(\cdot)$, we simply obfuscate $f_k(\cdot)$ using an unclonable puncturable obfuscation scheme UPO. To evaluate the copy-protected circuit on an input x, run the evaluation procedure of UPO.

To argue security, let us look at two experiments:

- The first experiment corresponds to the regular copy-protection security experiment. That is, $\mathcal{A}$ receives as input a copy-protected state ρ_{f_k}, which is copy-protection of f_k where k is sampled uniformly at random from the key space. It then creates a bipartite state which is split between $\mathcal{B}$ and $\mathcal{C}$, who are two non-communicating adversaries who can share some entanglement. Then, $\mathcal{B}$ and $\mathcal{C}$ independently receive as input x, which is picked uniformly at random. $(\mathcal{B},\mathcal{C})$ win if they simultaneously guess $f_k(x)$.
- The second experiment is similar to the first experiment except $\mathcal{A}$ receives as input copy-protection of f_k punctured at the point x, where x is the same input given to both $\mathcal{B}$ and $\mathcal{C}$.

Thanks to the puncturing security of $\mathcal{F}$, the probability that $(\mathcal{B},\mathcal{C})$ succeeds in the second experiment is negligible in λ. We would like to argue that $(\mathcal{B},\mathcal{C})$ succeed in the first experiment also with probability negligible in λ. Suppose not, we show that the security of UPO is violated.

Reduction to UPO: The reduction $\mathcal{R}_{\mathcal{A}}$ samples a uniformly random f_k and forwards it to the challenger of the UPO game. The challenger of the UPO game then generates either an obfuscation of f_k or the punctured circuit f_k punctured

at x. The obfuscated state is then sent to $\mathcal{R}_{\mathcal{A}}$, who in turn forwards this to $\mathcal{A}$ who prepares the bipartite state. The reduction $\mathcal{R}_{\mathcal{B}}$ (resp., $\mathcal{R}_{\mathcal{C}}$) then receives as input x which it duly forwards to $\mathcal{B}$ (resp., $\mathcal{C}$). Then, $\mathcal{B}$ and $\mathcal{C}$ each output $y_{\mathcal{B}}$ and $y_{\mathcal{C}}$. Then, $\mathcal{R}_{\mathcal{B}}$ outputs the **bit 0** if $f_k(x) = y_{\mathcal{B}}$, otherwise it outputs 1. Similarly, $\mathcal{R}_{\mathcal{C}}$ outputs **bit 0** if $f_k(x) = y_{\mathcal{C}}$, otherwise it outputs 1. The reason behind boldifying "bit 0" part will be discussed below.

Let us see how well $(\mathcal{R}_{\mathcal{A}}, \mathcal{R}_{\mathcal{B}}, \mathcal{R}_{\mathcal{C}})$ fares in the UPO game.

- *Case 1. Challenge bit is* $b = 0$. In this case, $\mathcal{R}_{\mathcal{A}}$ receives as input obfuscation of f_k with respect to UPO. Denote p_0 to be the probability that $(\mathcal{R}_{\mathcal{B}}, \mathcal{R}_{\mathcal{C}})$ output $(0, 0)$.
- *Case 2. Challenge bit is* $b = 1$. Here, $\mathcal{R}_{\mathcal{A}}$ receives as input obfuscation of the circuit f_k punctured at x. Similarly, denote p_1 to be the probability that $(\mathcal{R}_{\mathcal{B}}, \mathcal{R}_{\mathcal{C}})$ output $(1, 1)$.

From the security of UPO, we have the following: $\frac{p_0+p_1}{2} \leq \frac{1}{2} + \mu(\lambda)$, for some negligible function $\mu(\cdot)$. From the puncturing security of $\mathcal{F}$, $p_1 \geq 1 - \nu(\lambda)$, for some negligible function ν. From this, we can conclude, p_0 is negligible which proves the security of the copy-protection scheme.

Perhaps surprisingly (at least to the authors), we do not know how to make the above reduction work if $\mathcal{R}_{\mathcal{B}}$ (resp., $\mathcal{R}_{\mathcal{C}}$) instead output bit 1 in the case when $f_k(x) = y_{\mathcal{B}}$ (resp., $f_k(x) = y_{\mathcal{C}}$). This is because we only get an upper bound for p_1 which cannot be directly used to determine an upper bound for p_0.

Generalizing to Puncturable Cryptographic Schemes. We present two generalizations of the above approach. We first generalize the above approach to handle puncturable circuit classes see the full version for more details. A circuit class $\mathfrak{C}$, equipped with an efficient puncturing algorithm $\mathsf{Puncture}$, is said to be puncturable[6] if given a circuit $C \in \mathfrak{C}$, we can puncture C on a point x to obtain a punctured circuit G such that given a punctured circuit G, it is computationally infeasible to predict $C(x)$. As we can see, puncturable pseudorandom functions are a special case of puncturable circuit classes. The template to copy-protect an arbitrary puncturable circuit class, say $\mathfrak{C}$, is essentially the same as the above template to copy-protect puncturable pseudorandom functions. To copy-protect C, obfuscate C using the scheme UPO. The evaluation process and the proof of security proceed along the same lines as above.

We then generalize this further to handle puncturable[7] cryptographic schemes. We consider an abstraction of a cryptographic scheme consisting of efficient algorithms $(\mathsf{Gen}, \mathsf{Eval}, \mathsf{Puncture}, \mathsf{Verify})$ with the following correctness guarantee: the verification algorithm on input (pk, x, y) outputs 1, where $\mathsf{Gen}(1^\lambda)$ produces the secret key-public key pair $(\mathsf{sk}, \mathsf{pk})$ and the value y is the output of

[6] We need a slightly more general version than this. Formally, we puncture the circuit at two points (and not one), and then we require the adversary to predict the output of the circuit on one of the points, see the full version for more details.

[7] We again consider a general version where the circuit is punctured at two points.

Eval on input (sk, x). The algorithm $\mathsf{Puncture}$ on input (sk, x) outputs a punctured circuit that has the same functionality as $\mathsf{Eval}(\mathsf{sk}, \cdot)$ on all the points except x. The security property roughly states that predicting the output $\mathsf{Eval}(\mathsf{sk}, x)$ given the punctured circuit should be computationally infeasible. The above template of copy-protecting PRFs can similarly be adopted for copy-protecting puncturable cryptographic schemes.

Copy-Protecting Evasive Functions. Using UPO to construct copy-protection for evasive functions turns out to be more challenging. To understand the difficulty, let us compare both the notions below:

- In a UPO scheme, $\mathcal{A}$ gets as input an obfuscation of a circuit C (if the challenge bit is $b = 0$) or a circuit C punctured at two points $x^{\mathcal{B}}$ and $x^{\mathcal{C}}$ (if $b = 1$). In the challenge phase, $\mathcal{B}$ gets $x^{\mathcal{B}}$ and $\mathcal{C}$ gets $x^{\mathcal{C}}$.
- In a copy-protection scheme for evasive functions, $\mathcal{A}$ gets as input copy-protection of C, where C is a circuit implements an evasive function. In the challenge phase, $\mathcal{B}$ gets $x^{\mathcal{B}}$ and $\mathcal{C}$ gets $x^{\mathcal{C}}$, where $(x^{\mathcal{B}}, x^{\mathcal{C}})$ is sampled from an input distribution that depends on the challenge bit b. As example, we can sample $(x^{\mathcal{B}}, x^{\mathcal{C}}) = (x, x)$ as follows: x is sampled uniformly at random (if challenge bit is $b = 0$), otherwise x is sampled uniformly at random from the set of points on which C outputs 1 (if the challenge bit is $b = 1$).

In other words, the distribution from which $\mathcal{A}$ gets its input from depends on the bit b in UPO but the challenges given to $\mathcal{B}$ and $\mathcal{C}$ are sampled from a distribution that does not depend on b. The setting in the case of copy-protection is the opposite: the distribution from which $\mathcal{A}$ gets its input does not depend on b while the challenge distribution depends on b.

Preimage Sampleable Property: To handle this discrepancy, we consider a class of evasive functions called preimage sampleable evasive functions. The first condition we require is that there is a distribution $\mathcal{D}$ from which we can efficiently sample a circuit C (representing an evasive function) together with an input x such that $C(x) = 1$. The second condition states that there exists another distribution $\mathcal{D}'$ from which we can sample (C', x'), where x' is sampled uniformly at random and then a punctured circuit C' is sampled conditioned on $C'(x') = 1$, satisfying the following property: the distributions $\mathcal{D}$ and $\mathcal{D}'$ are computationally indistinguishable. The second condition is devised precisely to ensure that we can reduce the security of copy-protection to UPO.

Construction and Proof Idea: But first, let us discuss the construction of copy-protection: to copy-protect a circuit C, compute two layers of obfuscation of C. First, obfuscate C using a post-quantum iO scheme and then obfuscate the resulting circuit using UPO. To argue security, we view the obfuscated state given to $\mathcal{A}$ as follows: first sample C from $\mathcal{D}$ and then do the following: (a) give ρ_C to $\mathcal{A}$ if $b = 0$ and, (b) ρ_C to $\mathcal{A}$ if $b = 1$, where ρ_C is the copy-protected state and b is the challenge bit that is used in the challenge phase. So far, we have not changed the distribution. Now, we will modify (b). We will leverage

the above conditions to modify (b) as follows: we will instead jointly sample the circuit and the challenge input from $\mathcal{D}'$. Since $\mathcal{D}$ and $\mathcal{D}'$ are computationally indistinguishable, the adversary will not notice the change. Now, let us examine the modified experiment: if $b = 0$, the adversary receives ρ_C (defined above), where (C, x) is sampled from $\mathcal{D}$ and if $b = 1$, the adversary receives $\rho_{C'}$, where (C', x') is sampled from $\mathcal{D}'$. We can show that this precisely corresponds to the UPO experiment and thus, we can successfully carry out the reduction.

Single-Decryptor Encryption. A natural attempt to construct single-decryptor encryption would be to leverage UPO for puncturable cryptographic schemes. After all, it would seem that identifying a public-key encryption scheme where the decryption key can be punctured at the challenge ciphertexts would be helpful to achieve our desired result. A UPO obfuscation of the decryption algorithm would be the quantum decryption key of the single-decryptor encryption scheme.

Unfortunately, this does not quite work: the reason lies in the challenge distribution of UPO. In this work, we only consider challenge distributions whose marginals correspond to the uniform distribution. On the other hand, the public-key encryption scheme we start with might not have pseudorandom ciphertexts which would in turn make it incompatible with combining it with the UPO scheme as suggested above. Of course, we could have considered more general challenge distributions but the techniques we have developed is limited to achieving challenge distributions with uniform marginals. This suggests that we need to start with a public-key encryption scheme with pseudorandom ciphertexts.

We start with the public-key encryption scheme due to Sahai and Waters [SW14]. The advantage of this scheme is that the ciphertexts are pseudorandom. First, we show that this public-key encryption scheme can be made puncturable. Once we show this, using UPO for puncturable cryptographic schemes (and standard iO tricks), we construct single-decryptor encryption schemes of two flavors:

- First, we consider search security. In this security definition, $\mathcal{B}$ and $\mathcal{C}$ receive ciphertexts of random messages and they win if they are able to predict the messages.
- Next, we consider selective security. In this security definition, $\mathcal{B}$ and $\mathcal{C}$ receive encryptions of one of two messages adversarially chosen and they are supposed to predict which of the two messages was used in the encryption. Moreover, the adversarially chosen messages need to be declared before the security experiment begins and hence, the term selective security. Once we achieve this, we propose a generic lifting theorem to lift SDE security satisfying selective security to full adaptive security, where the challenge messages can be chosen later in the experiment.

Construction of UPO We move on to the construction of UPO.

Starting Point: Decoupling Unclonability and Computation. We consider the following template to design UPO. To obfuscate a circuit C, we build

two components. The first component is an unclonable quantum state that serves the purpose of authentication. The second component is going to aid in the computation of C once the authentication passes. Specifically, given an input x, we first use the unclonable quantum state to authenticate x and then execute the second component on the authenticated tag along with x to get the output $C(x)$.

The purpose of designing the obfuscation scheme this way is two-fold. Firstly, the fact that the first component is an unclonable quantum state means that an adversary cannot create multiple copies of this. And by design, without this state, it is not possible to execute the second component. Secondly, decoupling the unclonability and the computation part allows us to put less burden on the unclonable state, and in particular, only require the first component for authentication. This is in turn allows us to leverage existing classical tools to instantiate the second component.

To implement the above approach, we use a copy-protection scheme for pseudorandom functions [CLLZ21], denoted by CP, and a post-quantum indistinguishability obfuscation scheme, denoted by iO. In the UPO scheme, to obfuscate C, we do the following:

1. Copy-protect a pseudorandom function $f_k(\cdot)$ and,
2. Obfuscate a circuit, with the PRF key k hardcoded in it, that takes as input (x, y) and outputs $C(x)$ if and only if $f_k(x) = y$.

First Issue. While syntactically the above template makes sense, when proving security we run into an issue. To invoke the security of CP, we need to argue that the obfuscated circuit does not reveal any information about the PRF key k. This suggests that we need a much stronger object like virtual black box obfuscation instead of iO which is in general known to be impossible [BGI+01]. Taking a closer look, we realize that this issue arose because we wanted to completely decouple the CP part and the iO part.

Second Issue. Another issue that arises when attempting to work out the proof. At a high level, in the security proof, we reach a hybrid where we need to hardwire the outputs of the PRF on the challenge inputs $x^{\mathcal{B}}$ and $x^{\mathcal{C}}$ in the obfuscated circuit (i.e., in bullet 2 above). This creates an obstacle when we need to invoke the security of copy-protection: the outputs of the PRF are only available in the challenge phase (i.e., *after* $\mathcal{A}$ splits) whereas we need to know these outputs in order to generate the input to $\mathcal{A}$.

Addressing the Above Issues. We first address the second issue. We introduce a new security notion of copy-protection for PRFs, referred to as copy-protection with *preponed security*. Roughly speaking, in the preponed security experiment, $\mathcal{A}$ receives the outputs of the PRF on the challenge inputs instead of being delayed until the challenge phase. By design, this stronger security notion solves the second issue.

In order to resolve the first issue, we pull back and only partially decouple the two components. In particular, we tie both the CP and iO parts together by making non-black-box use of the underlying copy-protection scheme. Specifically, we

rely upon the scheme by Colandangelo et al. [CLLZ21]. Moreover, we show that Colandangelo et al. [CLLZ21] scheme satisfies preponed security by reducing their security to the security of their single-decryptor encryption construction; our proof follows along the same lines as theirs. Unfortunately, we do not know how to go further. While they did show that their single-decryptor encryption construction can be based on well studied cryptographic assumptions, the type of single-decryptor encryption schemes we need have a different flavor. In more detail, in [CLLZ21], they consider *independent* challenge distribution (i.e., both $\mathcal{B}$ and $\mathcal{C}$ receive ciphertexts where the challenge bit is picked independently), whereas we consider *identical* challenge distribution (i.e., the challenge bit for both $\mathcal{B}$ and $\mathcal{C}$ is identical). We show how to modify their construction to satisfy security with respect to different challenge distributions based on the two different versions of the simultaneous inner product conjecture.

Summary. To summarise, we design UPO for keyed circuit classes in P/poly as follows:

- We show that if the copy-protection scheme of [CLLZ21] satisfies preponed security, UPO for P/poly exists. This step makes heavy use of iO techniques.
- We reduce the task of proving preponed security for the copy-protection scheme of [CLLZ21] to the task of proving that the single-decryptor encryption construction of [CLLZ21] is secure in the identical challenge setting.

2 Unclonable Puncturable Obfuscation: Definition

Next, we present the definition of an unclonable puncturable obfuscation scheme.

Keyed Circuit Class. A class of classical circuits of the form $\mathfrak{C} = \{\mathfrak{C}_\lambda\}_{\lambda\in\mathbb{N}}$ is said to be a keyed circuit class if the following holds: $\mathfrak{C}_\lambda = \{C_k : k \in \mathcal{K}_\lambda\}$, where C_k is a (classical) circuit with input length $n(\lambda)$, output length $m(\lambda)$ and $\mathcal{K} = \{\mathcal{K}_\lambda\}_{\lambda\in\mathbb{N}}$ is the key space. We refer to C_k as a keyed circuit. We note that any circuit class can be represented as a keyed circuit class using universal circuits. We will be interested in the setting when C_k is a polynomial-sized circuit; henceforth, unless specified otherwise, all keyed circuit classes considered in this work will consist only of polynomial-sized circuits. We will also make a simplifying assumption that C_k and $C_{k'}$ have the same size, where $k, k' \in \mathcal{K}_\lambda$.

Syntax. An unclonable puncturable obfuscation (UPO) scheme $(\mathsf{Obf}, \mathsf{Eval})$ for a keyed circuit class $\mathfrak{C} = \{\mathfrak{C}_\lambda\}_{\lambda\in\mathbb{N}}$, consists of the following QPT algorithms:

- $\mathsf{Obf}(1^\lambda, C)$: on input a security parameter λ and a keyed circuit $C \in \mathfrak{C}_\lambda$ with input length $n(\lambda)$, it outputs a quantum state ρ_C.
- $\mathsf{Eval}(\rho_C, x)$: on input a quantum state ρ_C and an input $x \in \{0,1\}^{n(\lambda)}$, it outputs (ρ'_C, y).

Correctness. An unclonable puncturable obfuscation scheme $(\mathsf{Obf}, \mathsf{Eval})$ for a keyed circuit class $\mathfrak{C} = \{\mathfrak{C}_\lambda\}_{\lambda\in\mathbb{N}}$ is δ-correct, if for every $C \in \mathfrak{C}_\lambda$ with input length $n(\lambda)$, and for every $x \in \{0,1\}^{n(\lambda)}$,

$$\Pr\left[C(x) = y \;\middle|\; \begin{matrix}\rho_C \leftarrow \mathsf{Obf}(1^\lambda, C) \\ (\rho'_C, y) \leftarrow \mathsf{Eval}(\rho_C, x)\end{matrix}\right] \geq \delta$$

If δ is negligibly close to 1 then we say that the scheme is correct (i.e., we omit mentioning δ).

Remark 1. If $(1-\delta)$ is a negligible function in λ, by invoking the almost as good as new lemma [Aar16], we can evaluate ρ'_C on another input x' to get $C(x')$ with probability negligibly close to 1. We can repeat this process polynomially many times and each time, due to the quantum union bound [Gao15], we get the guarantee that the output is correct with probability negligibly close to 1.

2.1 Security

Puncturable Keyed Circuit Class. Consider a keyed circuit class $\mathfrak{C} = \{\mathfrak{C}_\lambda\}_{\lambda\in\mathbb{N}}$, where $\mathfrak{C}_\lambda$ consists of circuits of the form $C_k(\cdot)$, where $k \in \mathcal{K}_\lambda$, the input length of $C_k(\cdot)$ is $n(\lambda)$ and the output length is $m(\lambda)$. We say that $\mathfrak{C}_\lambda$ is said to be puncturable if there exists a deterministic polynomial-time puncturing algorithm $\mathsf{Puncture}$ such that the following holds: on input $k \in \{0,1\}^\lambda$, strings $x^{\mathcal{B}} \in \{0,1\}^{n(\lambda)}, x^{\mathcal{C}} \in \{0,1\}^{n(\lambda)}$, it outputs a circuit G_{k^*}. Moreover, the following holds: for every $x \in \{0,1\}^{n(\lambda)}$,

$$G_{k^*}(x) = \begin{cases} C_k(x), x \neq x^{\mathcal{B}}, x \neq x^{\mathcal{C}}, \\ \bot, \quad x \in \{x^{\mathcal{B}}, x^{\mathcal{C}}\}. \end{cases}$$

Without loss of generality, we can assume that the size of G_{k^*} is the same as the size of C_k. Note that for every keyed circuit class, there exists a trivial $\mathsf{Puncture}$ algorithm. The trivial $\mathsf{Puncture}$ algorithm on any input $k, x_1, x_2, \mu_1, \mu_2$, constructs the circuit C_k and then outputs the circuit G that on input x, if $x = x_0$ or x_1 outputs $\bot$, else if $x \notin \{x_1, x_2\}$ outputs $C_k(x)$[8].

Definition 1 (UPO Security). *We say that a pair of QPT algorithms* $(\mathsf{Obf}, \mathsf{Eval})$ *for a puncturable keyed circuit class* $\mathfrak{C}$*, associated with puncturing procedure* $\mathsf{Puncture}$*, satisfies* **UPO security** *with respect to a distribution* $\mathcal{D}_{\mathcal{X}}$ *on* $\{0,1\}^{n(\lambda)} \times \{0,1\}^{n(\lambda)}$ *if for every QPT* $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ *in* $\mathsf{UPO.Expt}$ *(see Fig. 2), there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\mathsf{Pr}\left[1 \leftarrow \mathsf{UPO.Expt}^{(\mathcal{A},\mathcal{B},\mathcal{C}),\mathcal{D}_{\mathcal{X}},\mathfrak{C}}\left(1^\lambda, b\right) \;:\; b \xleftarrow{\$} \{0,1\}\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

[8] The output circuit G_{k^*} is not of the same size as C_k, but this issue can be resolved by sufficient padding of the circuit class.

$\underline{\mathsf{UPO.Expt}^{(\mathcal{A},\mathcal{B},\mathcal{C}),\mathcal{D}_{\mathcal{X}},\mathfrak{C}}\left(1^{\lambda},b\right)}$:

- $\mathcal{A}$ sends k, where $k \in \mathcal{K}_{\lambda}$, to the challenger Ch.
- Ch samples $(x^{\mathcal{B}}, x^{\mathcal{C}}) \leftarrow \mathcal{D}_{\mathcal{X}}(1^{\lambda})$ and generates $G_{k^*} \leftarrow \mathsf{Puncture}(k, x^{\mathcal{B}}, x^{\mathcal{C}})$.
- Ch generates ρ_b as follows:
  - $\rho_0 \leftarrow \mathsf{Obf}(1^{\lambda}, C_k(\cdot))$,
  - $\rho_1 \leftarrow \mathsf{Obf}(1^{\lambda}, G_{k^*}(\cdot))$

  It sends ρ_b to $\mathcal{A}$.
- Apply $(\mathcal{B}(x^{\mathcal{B}},\cdot) \otimes \mathcal{C}(x^{\mathcal{C}},\cdot))(\sigma_{\mathcal{B},\mathcal{C}})$ to obtain $(b_{\mathbf{B}}, b_{\mathbf{C}})$.
- Output 1 if $b = b_{\mathbf{B}} = b_{\mathbf{C}}$.

Fig. 2. Security Experiment

Generalized Security. For most applications, the security definition discussed in Sect. 2.1 suffices, but for a couple of applications, we need a generalized definition as follows. We allow the adversary to choose the outputs of the circuit generated by $\mathsf{Puncture}$ on the punctured points. Previously, the circuit generated by the puncturing algorithm was such that on the punctured points, it output $\perp$. Instead, we allow the adversary to decide the values that need to be output on the points that are punctured. We emphasize that the adversary still would not know the punctured points itself until the challenge phase. Formally, the (generalized) puncturing algorithm $\mathsf{GenPuncture}$ now takes as input $k \in \mathcal{K}_{\lambda}$, polynomial-sized circuits $\mu^{\mathcal{B}} : \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}$, $\mu^{\mathcal{C}} : \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}$, strings $x^{\mathcal{B}} \in \{0,1\}^{n(\lambda)}, x^{\mathcal{C}} \in \{0,1\}^{n(\lambda)}$, if $x^{\mathcal{B}} \neq x^{\mathcal{C}}$, it outputs a circuit G_{k^*} such that for every $x \in \{0,1\}^{n(\lambda)}$,

$$G_{k^*}(x) = \begin{cases} C_k(x), & x \neq x^{\mathcal{B}}, x \neq x^{\mathcal{C}} \\ \mu_{\mathcal{B}}(x^{\mathcal{B}}), & x = x^{\mathcal{B}} \\ \mu_{\mathcal{C}}(x^{\mathcal{C}}), & x = x^{\mathcal{C}}, \end{cases}$$

else it outputs a circuit G_{k^*} such that for every $x \in \{0,1\}^{n(\lambda)}$,

$$G_{k^*}(x) = \begin{cases} C_k(x), & x \neq x^{\mathcal{B}} \\ \mu_{\mathcal{B}}(x^{\mathcal{B}}), & x = x^{\mathcal{B}}. \end{cases}$$

As before, we assume that without loss of generality, the size of G_{k^*} is the same as the size of C_k. A keyed circuit class $\mathfrak{C}$ associated with a generalized puncturing algorithm $\mathsf{GenPuncture}$ is referred to as a *generalized puncturable keyed circuit class*. Note that for every keyed circuit class $\mathfrak{C} = \{C_k\}_k$, there exists a trivial $\mathsf{GenPuncture}$ algorithm, which on any input $k, x_1, x_2, \mu_1, \mu_2$, constructs

the circuit C_k and then outputs the circuit G_{k^*}[9] that on input x, if $x = x_i$ for any $i \in \{0,1\}$, outputs $\mu_i(x_i)$, else if $x \notin \{x_1, x_2\}$ outputs $C_k(x)$.

Definition 2 (Generalized UPO security). *We say that a pair of QPT algorithms* $(\mathsf{Obf}, \mathsf{Eval})$ *for a generalized keyed circuit class* $\mathfrak{C} = \{\mathfrak{C}_\lambda\}_{\lambda\in\mathbb{N}}$ *equipped with a puncturing algorithm* $\mathsf{GenPuncture}$*, satisfies* **generalized UPO security** *with respect to a distribution* $\mathcal{D}_{\mathcal{X}}$ *on* $\{0,1\}^{n(\lambda)} \times \{0,1\}^{n(\lambda)}$ *if the following holds for every QPT* $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ *in* $\mathsf{GenUPO.Expt}$ *defined in Fig. 3:*

$$\Pr\left[1 \leftarrow \mathsf{GenUPO.Expt}^{(\mathcal{A},\mathcal{B},\mathcal{C}),\mathcal{D}_{\mathcal{X}},\mathfrak{C}}\left(1^\lambda, b\right) \ : \ b \xleftarrow{\$} \{0,1\}\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

$\underline{\mathsf{GenUPO.Expt}^{(\mathcal{A},\mathcal{B},\mathcal{C}),\mathcal{D}_{\mathcal{X}},\mathfrak{C}}\left(1^\lambda, b\right)}$:

- $\mathcal{A}$ sends $(k, \mu_{\mathcal{B}}, \mu_{\mathcal{C}})$, where $k \in \mathcal{K}_\lambda, \mu_{\mathcal{B}} : \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}, \mu_{\mathcal{C}} : \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}$, to the challenger Ch.
- Ch samples $(x^{\mathcal{B}}, x^{\mathcal{C}}) \leftarrow \mathcal{D}_{\mathcal{X}}(1^\lambda)$ and generates $G_{k^*} \leftarrow \mathsf{Puncture}(k, x^{\mathcal{B}}, x^{\mathcal{C}}, \mu_{\mathcal{B}}, \mu_{\mathcal{C}})$.
- Ch generates ρ_b as follows:
  - $\rho_0 \leftarrow \mathsf{Obf}(1^\lambda, C_k)$,
  - $\rho_1 \leftarrow \mathsf{Obf}(1^\lambda, G_{k^*})$

  It sends ρ_b to $\mathcal{A}$.
- Apply $(\mathcal{B}(x^{\mathcal{B}}, \cdot) \otimes \mathcal{C}(x^{\mathcal{C}}, \cdot))(\sigma_{\mathcal{B},\mathcal{C}})$ to obtain $(b_{\mathbf{B}}, b_{\mathbf{C}})$.
- Output 1 if $b = b_{\mathbf{B}} = b_{\mathbf{C}}$.

Fig. 3. Generalized Security Experiment

Instantiations of $\mathcal{D}_{\mathcal{X}}$*.* In the applications, we will be considering the following two distributions:

1. $\mathcal{U}_{\{0,1\}^{2n}}$: the uniform distribution on $\{0,1\}^{2n}$. When the context is clear, we simply refer to this distribution as $\mathcal{U}$.
2. $\mathsf{Id}_{\mathcal{U}}\{0,1\}^n$: identical distribution on $\{0,1\}^n \times \{0,1\}^n$ with uniform marginals. That is, the sampler for $\mathsf{Id}_{\mathcal{U}}\{0,1\}^n$ is defined as follows: sample x from $\mathcal{U}_{\{0,1\}^n}$ and output (x, x). When the context is clear, we simply refer to this distribution as $\mathsf{Id}_{\mathcal{U}}$.

[9] As before, the output circuit G_{k^*} may not have the same size as C_k, but this can be resolved by sufficient padding of the complexity class.

3 Conjectures

To show that our construction satisfies the UPO security notions, we rely upon some novel conjectures. Towards understanding our conjectures, consider the following problem: suppose say an adversary $\mathcal{B}$ is given a state $\rho_{\mathbf{x}}$ that is generated using a secret vector $\mathbf{x} \in \mathbb{Z}_Q^n$, where $Q, n \in \mathbb{N}$ and Q is prime. We are given the guarantee that just given $\rho_{\mathbf{x}}$, it should be infeasible to compute $\mathbf{x}$ with inverse polynomial probability over the randomness of sampling $\mathbf{x}$. Now, the goal of $\mathcal{B}$ is to predict $(\mathbf{u}, \langle \mathbf{u}, \mathbf{x} \rangle)$ versus $(\mathbf{u}, \langle \mathbf{u}, \mathbf{x} \rangle + m)$, where $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_Q^n$ and $m \xleftarrow{\$} \mathbb{Z}_Q$. The quantum Goldreich-Levin theorem [AC02, CLLZ21] states that, for the case when $Q = 2$, the probability that $\mathcal{B}$ succeeds is negligibly close to $\frac{1}{2}$. The quantum Goldreich-Levin theorem has been generalized [APV23, STHY23] to the case when Q is large.

We study a generalized version of this problem where there are two non-communicating but entangled parties $\mathcal{B}$ and $\mathcal{C}$ and both are simultaneously participating in the above distinguishing experiment. Depending on the entangled state shared by $\mathcal{B}$ and $\mathcal{C}$ and the distributions from which the samples are generated, we obtain many generalizations. We conjecture that in some of these generalized versions, the prediction probability is close to $\frac{1}{2}$. But first, we will capture all the generalizations by defining the following problem.

$(\mathcal{D}_{\mathcal{X}}, \mathcal{D}_{\mathsf{Ch}}, \mathcal{D}_{\mathsf{bit}})$-Simultaneous Inner Product Problem ($(\mathcal{D}_{\mathcal{X}}, \mathcal{D}_{\mathsf{Ch}}, \mathcal{D}_{\mathsf{bit}})$-simultIP). Let $\mathcal{D}_{\mathcal{X}}$ be a distribution on $\mathbb{Z}_Q^n \times \mathbb{Z}_Q^n$, $\mathcal{D}_{\mathsf{Ch}}$ be a distribution on $\mathbb{Z}_Q^{n+1} \times \mathbb{Z}_Q^{n+1}$ and finally, let $\mathcal{D}_{\mathsf{bit}}$ be a distribution on $\{0,1\} \times \{0,1\}$, for prime $Q \in \mathbb{N}$. Let $\mathcal{B}'$ and $\mathcal{C}'$ be QPT algorithms. Let $\rho = \{\rho_{\mathbf{x}^{\mathcal{B}}, \mathbf{x}^{\mathcal{C}}}\}_{\mathbf{x}^{\mathcal{B}}, \mathbf{x}^{\mathcal{C}} \in \mathbb{Z}_Q^n}$ be a set of bipartite states. If $\mathbf{x}^{\mathcal{B}} = \mathbf{x}^{\mathcal{C}} = \mathbf{x}$ then we denote $\rho_{\mathbf{x}^{\mathcal{B}}, \mathbf{x}^{\mathcal{C}}}$ by $\rho_{\mathbf{x}}$.

Consider the following game.

- Sample $(\mathbf{x}^{\mathcal{B}}, \mathbf{x}^{\mathcal{C}}) \leftarrow \mathcal{D}_{\mathcal{X}}$,
- Sample $\left(\left(\mathbf{u}^{\mathcal{B}}, m^{\mathcal{B}}\right), \left(\mathbf{u}^{\mathcal{C}}, m^{\mathcal{C}}\right)\right) \leftarrow \mathcal{D}_{\mathsf{Ch}}$,
- Set $z_0^{\mathcal{B}} = \langle \mathbf{u}^{\mathcal{B}}, \mathbf{x}^{\mathcal{B}} \rangle, z_0^{\mathcal{C}} = \langle \mathbf{u}^{\mathcal{C}}, \mathbf{x}^{\mathcal{C}} \rangle, z_1^{\mathcal{B}} = m^{\mathcal{B}} + \langle \mathbf{u}^{\mathcal{B}}, \mathbf{x}^{\mathcal{B}} \rangle, z_1^{\mathcal{C}} = m^{\mathcal{C}} + \langle \mathbf{u}^{\mathcal{C}}, \mathbf{x}^{\mathcal{C}} \rangle$,
- Sample $(b^{\mathcal{B}}, b^{\mathcal{C}}) \leftarrow \mathcal{D}_{\mathsf{bit}}$,
- $(\widehat{b}^{\mathcal{B}}, \widehat{b}^{\mathcal{C}}) \leftarrow (\mathcal{B}'(\mathbf{u}^{\mathcal{B}}, z_{b^{\mathcal{B}}}^{\mathcal{B}}, \cdot) \otimes \mathcal{C}'(\mathbf{u}^{\mathcal{C}}, z_{b^{\mathcal{C}}}^{\mathcal{C}}, \cdot))(\rho_{\mathbf{x}^{\mathcal{B}}, \mathbf{x}^{\mathcal{C}}})$.

We say that $(\mathcal{B}', \mathcal{C}')$ succeeds if $\widehat{b}^{\mathcal{B}} = b^{\mathcal{B}}$ and $\widehat{b}^{\mathcal{C}} = b^{\mathcal{C}}$.

Our goal is to upper bound the optimal success probability in the above problem. We are primarily interested in the following setting: $\mathcal{D}_{\mathsf{bit}}$ is a distribution on $\{0,1\} \times \{0,1\}$, where (b, b) is sampled with probability $\frac{1}{2}$, for $b \in \{0,1\}$. In this case, we simply refer to the above problem as $(\mathcal{D}_{\mathcal{X}}, \mathcal{D}_{\mathsf{Ch}})$-simultIP problem.

Conjectures. We state the following conjectures. In the conjectures, we assume that the order of the field is $Q \geq 2^{\lambda}$. We are interested in the following distributions:

– We define $\mathcal{D}^{\mathsf{ind}}_{\mathsf{Ch}}$ as follows: it samples $\left(\left(\mathbf{u}^{\mathcal{B}}, m^{\mathcal{B}}\right), \left(\mathbf{u}^{\mathcal{C}}, m^{\mathcal{C}}\right)\right)$, where $\mathbf{u}^{\mathcal{B}} \xleftarrow{\$} \mathbb{Z}_Q^n, \mathbf{u}^{\mathcal{C}} \xleftarrow{\$} \mathbb{Z}_Q^n, m^{\mathcal{B}} \xleftarrow{\$} \mathbb{Z}_Q, m^{\mathcal{C}} \xleftarrow{\$} \mathbb{Z}_Q$. We define $\mathcal{D}^{\mathsf{id}}_{\mathsf{Ch}}$ as follows: it samples $\left((\mathbf{u}, m), (\mathbf{u}, m)\right)$, where $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_Q^n, m \xleftarrow{\$} \mathbb{Z}_Q$.
– Similarly, we define $\mathcal{D}^{\mathsf{ind}}_{\mathcal{X}}$ as follows: it samples $\left(\mathbf{x}^{\mathcal{B}}, \mathbf{x}^{\mathcal{C}}\right)$, where $\mathbf{x}^{\mathcal{B}} \xleftarrow{\$} \mathbb{Z}_Q^n, \mathbf{x}^{\mathcal{C}} \xleftarrow{\$} \mathbb{Z}_Q^n$. We define $\mathcal{D}^{\mathsf{id}}_{\mathcal{X}}$ as follows: it samples $(\mathbf{x}, \mathbf{x})$, where $\mathbf{x} \xleftarrow{\$} \mathbb{Z}_Q^n$.

*Conjecture 1 ($\left(\mathcal{D}^{\mathsf{id}}_{\mathcal{X}}, \mathcal{D}^{\mathsf{id}}_{\mathsf{Ch}}\right)$-*simultIP* Conjecture).* Consider a set of bipartite states $\rho = \{\rho_{\mathbf{x}}\}_{\mathbf{x} \in \mathbb{Z}_Q^n}$ satisfying the following property: for any two QPT adversaries $\mathcal{B}, \mathcal{C}$,

$$\mathsf{Pr}\left[(\mathbf{x}, \mathbf{x}) \leftarrow (\mathcal{B} \otimes \mathcal{C})(\rho_{\mathbf{x}}) \ : \ (\mathbf{x}, \mathbf{x}) \leftarrow \mathcal{D}^{\mathsf{id}}_{\mathcal{X}}\right] \leq \nu(n)$$

for some negligible function $\nu(\lambda)$.

Any QPT non-local solver for the $\left(\mathcal{D}^{\mathsf{id}}_{\mathcal{X}}, \mathcal{D}^{\mathsf{id}}_{\mathsf{Ch}}\right)$-simultIP problem succeeds with probability at most $\frac{1}{2} + \varepsilon(n)$, where ε is a negligible function.

*Conjecture 2 ($\left(\mathcal{D}^{\mathsf{ind}}_{\mathcal{X}}, \mathcal{D}^{\mathsf{ind}}_{\mathsf{Ch}}\right)$-*simultIP* Conjecture).* Consider a set of bipartite states $\rho = \{\rho_{\mathbf{x}^{\mathcal{B}}, \mathbf{x}^{\mathcal{C}}}\}_{\mathbf{x}^{\mathcal{B}}, \mathbf{x}^{\mathcal{C}} \in \mathbb{Z}_Q^n}$ satisfying the following property: for any two QPT adversaries $\mathcal{B}, \mathcal{C}$,

$$\mathsf{Pr}\left[\left(\mathbf{x}^{\mathcal{B}}, \mathbf{x}^{\mathcal{C}}\right) \leftarrow (\mathcal{B} \otimes \mathcal{C})\left(\rho_{\mathbf{x}^{\mathcal{B}}, \mathbf{x}^{\mathcal{C}}}\right) \ : \ \left(\mathbf{x}^{\mathcal{B}}, \mathbf{x}^{\mathcal{C}}\right) \leftarrow \mathcal{D}^{\mathsf{ind}}_{\mathcal{X}}\right] \leq \nu(n)$$

for some negligible function $\nu(\lambda)$.

Any QPT non-local solver for the $\mathcal{D}^{\mathsf{id}}_{\mathsf{Ch}}$-simultIP problem succeeds with probability at most $\frac{1}{2} + \varepsilon(n)$, where ε is a negligible function.

3.1 Discussion

Special Cases. Variants of the above conjectures, obtained by modifying the input and challenge distributions, have been proven to be true by considering different flavors of the simultaneous Goldreich-Levin theorem. We mention three such special cases below.

	Field Size	Input distribution	Challenge sample distibution	Challenge bit distribution
[AKL23]	$Q = 2$	$\mathcal{D}_{\mathcal{X}} = \mathcal{D}^{\mathsf{id}}_{\mathcal{X}}$	$\mathcal{D}_{\mathsf{Ch}} = \widetilde{\mathcal{D}}^{\mathsf{ind}}_{\mathsf{Ch}}$	$\mathcal{D}_{\mathsf{bit}} = \mathcal{D}^{\mathsf{ind}}_{\mathsf{bit}}$
[KT22]	$Q \in \{2, 3\}$	$\mathcal{D}_{\mathcal{X}} = \mathcal{D}^{\mathsf{ind}}_{\mathcal{X}}$	$\mathcal{D}_{\mathsf{Ch}} = \widetilde{\mathcal{D}}^{\mathsf{ind}}_{\mathsf{Ch}}$	$\mathcal{D}_{\mathsf{bit}} = \mathcal{D}^{\mathsf{ind}}_{\mathsf{bit}}$
[AKY24]	$Q = 2$	$\mathcal{D}_{\mathcal{X}} = \mathcal{D}^{\mathsf{id}}_{\mathcal{X}}$	$\mathcal{D}_{\mathsf{Ch}} = \widetilde{\mathcal{D}}^{\mathsf{ind}}_{\mathsf{Ch}}$	$\mathcal{D}_{\mathsf{bit}} = \mathcal{D}^{\mathsf{id}}_{\mathsf{bit}}$

We define $\widetilde{D}^{\mathsf{ind}}_{\mathsf{Ch}}$, for the case when $Q = 2$, to be the distribution that samples $((\mathbf{u}^{\mathcal{B}}, 1), (\mathbf{u}^{\mathcal{C}}, 1))$, where $\mathbf{u}^{\mathcal{B}} \xleftarrow{\$} \mathbb{Z}_Q^n$ and $\mathbf{u}^{\mathcal{C}} \xleftarrow{\$} \mathbb{Z}_Q^n$. Note that this is similar to $D^{\mathsf{ind}}_{\mathsf{Ch}}$ except that $m^{\mathcal{B}}$ and $m^{\mathcal{C}}$ is always set to 1.

Although not explicitly stated, the generic framework of upgrading classical reductions to non-local reductions, introduced in [AKL23], can be leveraged to

extend the above result to large values of Q. Finally, the work of [CG23] considers a similar simultaneous Goldreich-Levin theorem as [AKL23] except that Bob's and Charlie's challenge messages consists of multiple Goldreich-Levin samples.

Among all the works so far, [AKY24] (which was subsequent to our work) is the only work that can handle *identical* challenge bit distributions (i.e. $\mathcal{D}_{\mathsf{bit}}^{\mathsf{id}}$) but for the case when $Q = 2$.

Proving our conjecture: Challenges. Unfortunately, it is unclear how to leverage the techniques used in the aforementioned works to prove our conjectures. To understand the difficulties, let us look at each of the two conjectures separately.

Let us start with the $(\mathcal{D}_{\mathcal{X}}^{\mathsf{ind}}, \mathcal{D}_{\mathsf{Ch}}^{\mathsf{ind}})$-$\mathsf{simultIP}$ conjecture (Conjecture 2). Recall that this conjecture is defined for large fields (of size $\geq 2^{\lambda}$). When $Q = 2$, a version of this conjecture was proven in a subsequent work by [AKY24]. Their proof is sensitive to the case that they are dealing with binary fields and their techniques do not seem to readily generalize to the case of large fields.

Proving $(\mathcal{D}_{\mathcal{X}}^{\mathsf{id}}, \mathcal{D}_{\mathsf{Ch}}^{\mathsf{id}})$-$\mathsf{simultIP}$ conjecture seems much harder although this is incomparable to the $(\mathcal{D}_{\mathcal{X}}^{\mathsf{ind}}, \mathcal{D}_{\mathsf{Ch}}^{\mathsf{ind}})$-$\mathsf{simultIP}$ conjecture. Let us illustrate its difficulty using the example of simultaneous Goldreich-Levin theorem proven in [AKL23,KT22]. They consider the independent setting where both Bob and Charlie receive independent Goldreich-Levin samples (i.e. $\mathcal{D}_{\mathsf{Ch}} = \widetilde{\mathcal{D}}_{\mathsf{Ch}}^{\mathsf{ind}}$ and $\mathcal{D}_{\mathsf{bit}} = \mathcal{D}_{\mathsf{bit}}^{\mathsf{ind}}$). To recall, the quantum Goldreich-Levin extractor (for a single party), as proven by [AC02,CLLZ21], proceeds as follows: it creates a superposition over all the challenge messages, coherently computes the distinguisher on it, applies a phase flip operation, uncomputes and finally, measures the answer in the Fourier basis. In the simultaneous version, we are required to extract from two parties, say, Bob and Charlie, simultaneously. In the independent setting, Bob's extractor and Charlie's extractor can each independently run the (single party) Goldreich-Levin extractor, and the analysis for the single party case smoothly extends to the simultaneous case as well. However, in the identical setting, this approach does not work. This is due to the fact that Bob and Charlie, instead of applying the Fourier basis measurement independently, would have to apply an entangled measurement jointly on their system. Since the two extractors are not allowed to communicate, it is not at all clear if such a measurement operation can be implemented. Another, and perhaps a more serious problem, is that the phase flip operations done by both Bob and Charlie cancel each other out, making the rest of the extraction process useless. Even though these difficulties are in the context of proving the simultaneous Goldreich-Levin theorem in the identical challenge setting, similar issues seem to exist with other non-local approaches to proving the identical simultaneous Goldreich-Levin theorem.

4 Direct Construction

In this section, we construct unclonable puncturable obfuscation for all efficiently computable generalized puncturable keyed circuit classes, with respect to $\mathcal{U}$ and $\mathsf{Id}_{\mathcal{U}}$ challenge distribution (see Sect. 2.1). Henceforth, we assume that any keyed

circuit class we consider will consist of circuits that are efficiently computable. We present the construction in three steps.

1. In the first step (Sect. 4.1), we construct a single decryptor encryption (SDE) scheme based on the CLLZ scheme [CLLZ21] (see Fig. 4) and show that it satisfies $\mathcal{D}_{\mathsf{ind\text{-}msg}}$-indistinguishability from random anti-piracy (and $\mathcal{D}_{\mathsf{ind\text{-}msg}}$-indistinguishability from random anti-piracy respectively) (for more details on the definition, see the full version), based on the conjectures, Conjectures 1 and 2 .
2. In the second step (Sect. 4.2), we define a variant of the security definition considered in [CLLZ21] with respect to two different challenge distributions and prove that the copy-protection construction for PRFs in [CLLZ21] (see Fig. 8) satisfies this security notion, based on the indistinguishability from random anti-piracy guarantees of the SDE scheme considered in the first step.
3. In the third step (Sect. 4.3), we show how to transform the copy-protection scheme obtained from the first step into UPO for a keyed circuit class with respect to the $\mathcal{U}$ and $\mathsf{Id}_{\mathcal{U}}$ challenge distribution.

4.1 A New Public-Key Single-Decryptor Encryption Scheme

The first step is to construct a SDE scheme of the suitable form. While SDE schemes have been studied [GZ20, CLLZ21], we require a weaker version of security called indistinguishability from random anti-piracy (for more details on the definition, see the full version), which has not been considered in prior works.

Our construction is based on the SDE scheme in [CLLZ21, Section 6.3] which we recall in Fig. 4. From here on, we will refer to it as the CLLZ SDE scheme, given in Fig. 4. Next, we define a family of SDE schemes based on the CLLZ SDE, called CLLZ *post-processing* schemes, as follows.

CLLZ Post-processing Single Decryptor Encryption Scheme: Definition. We call a SDE scheme $(\mathsf{Gen}, \mathsf{QKeyGen}, \mathsf{Enc}, \mathsf{Dec})$ a CLLZ post-processing if there exists polynomial time classical algorithms $(\mathsf{EncPostProcess}, \mathsf{DecPostProcess})$, such that DecPostProcess is a deterministic algorithm. For correctness of a CLLZ *post-processing* SDE scheme (see Fig. 5) we require that for every string r, m,

$$c' \leftarrow \mathsf{EncPostProcess}(m, r), m' \leftarrow \mathsf{DecPostProcess}(c', r) \implies m = m'. \quad (1)$$

It is easy to verify that assuming Eq. (1), δ-correctness of the CLLZ SDE implies δ-correctness of a CLLZ *post-processing* SDE for every $\delta \in [0, 1]$.

Construction. We next consider the following CLLZ *post-processing* scheme given in Fig. 6. As mentioned before, we will assume that the message length is at least polynomial in the security parameter. Note that the algorithms $(\mathsf{EncPostProcess}, \mathsf{DecPostProcess})$ in Fig. 6 satisfies Eq. (1), and hence if the CLLZ

Tools: post-quantum indistinguishability obfuscation iO.

$\mathsf{Gen}(1^\lambda)$:

1. Sample ℓ_0 uniformly random subspaces $\{A_i\}_{i\in[\ell_0]}$ of dimension $\frac{\lambda}{2}$ from $\mathbb{Z}_2^\lambda$ and for each $i \in [\ell_0]$, sample vectors $s_i, s'_i \xleftarrow{\$} \mathbb{Z}_2^\lambda$, where $\ell_0 = \ell_0(\lambda)$ is a polynomial in λ.
2. Compute $\{R_i^0, R_i^1\}_{i\in\ell_0}$, where for every $i \in [\ell_0]$, $R_i^0 \leftarrow \mathsf{iO}(A_i + s_i)$ and $R_i^1 \leftarrow \mathsf{iO}(A_i^\perp + s'_i)$ are the membership oracles.
3. Output $\mathsf{sk} = \{\{A_{is_i,s'_i}\}_i\}$ and $\mathsf{pk} = \{R_i^0, R_i^1\}_{i\in\ell_0}$

$\mathsf{QKeyGen}(\mathsf{sk})$:

1. Interprete sk as $\{\{A_{is_i,s'_i}\}_i\}$.
2. Output $\rho_{\mathsf{sk}} = \{\{|A_{is_i,s'_i}\rangle\}_i\}$.

$\mathsf{Enc}(\mathsf{pk}, m)$:

1. Interprete $\mathsf{pk} = \{R_i^0, R_i^1\}_{i\in\ell_0}$.
2. Sample $r \xleftarrow{\$} \{0,1\}^n$.
3. Generate $\tilde{Q} \leftarrow \mathsf{iO}(Q_{m,r})$ where $Q_{m,r}$ has $\{R_i^0, R_i^1\}_{i\in\ell_0}$ hardcoded inside, and on input $v_1, \ldots, v_{\ell_0} \in \{0,1\}^{n\ell_0}$, checks if $R_i^{r_i}(v_i) = 1$ for every $i \in [\ell_0]$ and if the check succeeds, outputs m, otherwise output $\perp$.
4. Output $\mathsf{ct} = (r, \tilde{Q})$

$\mathsf{Dec}(\rho_{\mathsf{sk}}, \mathsf{ct})$

1. Interprete $\mathsf{ct} = (r, \tilde{Q})$.
2. For every $i \in [\ell_0]$, if $r_i = 1$ apply $H^{\otimes n}$ on $|A_{is_i,s'_i}\rangle$. Let the resulting state be $|\psi_x\rangle$.
3. Run the circuit $\tilde{Q}$ in superposition on the state $|\psi_x\rangle$ and measure the output register and output the measurement result m.

Fig. 4. The CLLZ single decryptor encryption scheme, see [CLLZ21, Construction 1].

SDE scheme (depicted in Fig. 4) satisfies δ-correctness so does the SDE scheme in Fig. 6. It is also easy to see that $\mathsf{DecPostProcess}$ is a determinisitc algorithm. Next, we prove security for the SDE scheme in Fig. 6 based on the simultaneous inner product conjectures.

Remark 2. By the definition of the randomized embedding Embed_Q defined in the algorithm $\mathsf{EncPostProcess}$ given in Fig. 6, it is easy to see that the ensemble

$$\{\mathsf{Embed}_Q(m)\}_{m\xleftarrow{\$}\{0,1\}^M} = \{\tilde{m}_Q\}_{\tilde{m}_Q\xleftarrow{\$}\{0,1,\ldots,LM-1\}} \approx_s \{\tilde{m}_Q\}_{\tilde{m}_Q\xleftarrow{\$}\mathbb{Z}_Q},$$

Tools: CLLZ SDE scheme given in Figure 4.

$\mathsf{Gen}(1^\lambda)$: Same as $\mathsf{CLLZ.Gen}(1^\lambda)$.

$\mathsf{QKeyGen}(\mathsf{sk})$: Same as $\mathsf{CLLZ.QKeyGen}(\mathsf{sk})$.

$\mathsf{Enc}(\mathsf{pk}, m)$:

1. Sample $r \xleftarrow{\$} \mathbb{Z}_Q^\lambda$, where Q is the smallest prime greater than or equal to $M \cdot 2^\lambda$ with M being the size of the message space, i.e., $M = 2^{|m|}$ and $|m|$ is the bit-size of the message m.
2. Generate $c \leftarrow \mathsf{EncPostProcess}(m, r)$ and generate $c' \leftarrow \mathsf{CLLZ.Enc}(\mathsf{pk}, r)$[10].
3. Output $\mathsf{ct} = (c, c')$.

$\mathsf{Dec}(\rho_{\mathsf{sk}}, \mathsf{ct})$

1. Interprete $\mathsf{ct} = (c, c')$.
2. Generate $r \leftarrow \mathsf{CLLZ.Dec}(\rho_{\mathsf{sk}}, c')$.
3. Output $m \leftarrow \mathsf{DecPostProcess}(c, r)$.

Fig. 5. Definition of a CLLZ *post-processing* SDE scheme.

$\mathsf{EncPostProcess}(m, r)$:

1. Sample $u \xleftarrow{\$} \mathbb{Z}_Q^\lambda$, where Q is the smallest prime number greater than $2^{|m|+\lambda}$, and $|m|$ is the bit-size of the binary string m.
2. Generate $\tilde{m} \leftarrow \mathsf{Embed}_Q(m)$, where Embed_Q randomly embeds the binary string m in $\mathbb{Z}_Q$, i.e., $\tilde{m}_Q \equiv kM + m_Q$ where $k \xleftarrow{\$} \{0, 1, \ldots, L-1\}$, $M \equiv 2^{|m|}$, $L \equiv [Q/M]$, and m_Q is the canonical embedding of m in $\mathbb{Z}_Q$.
3. Output $u, \tilde{m}_Q + \langle u, r \rangle$, where the addition and inner product uses the product over the field $\mathbb{Z}_Q$.

$\mathsf{DecPostProcess}(c, r)$:

1. Interprete c as u, z.
2. Generate $\tilde{m}_Q \leftarrow z + \langle u, r \rangle$.
3. Output m where m is the binary representation of $\tilde{m}_Q \mod M$.

Fig. 6. Construction of a CLLZ *post-processing* SDE scheme.

because $Q - L < M$ by definition of L, and hence, $\frac{Q-L}{Q} < \frac{M}{Q}$ which is at most $\frac{M}{M \cdot 2^\lambda} = \frac{1}{2^\lambda}$, by our choice of Q.

[10] We would like to note that the obfuscated circuit may be padded more than what is required in the CLLZ SDE scheme, for the security proofs of the CLLZ *post-processing* SDE.

Theorem 6. *Assuming Conjecture 2, the existence of post-quantum sub-exponentially secure* iO *and one-way functions, and the quantum hardness of Learning-with-errors problem (LWE), the* CLLZ *post-processing* SDE *as defined in Fig. 5 given in Fig. 6 satisfies* $\mathcal{D}_{\mathsf{ind\text{-}msg}}$*-indistinguishability from random anti-piracy (for more details on the definition, see the full version).*

Theorem 7. *Assuming Conjecture 1, the existence of post-quantum sub-exponentially secure* iO *and one-way functions, and quantum hardness of Learning-with-errors problem (LWE), the* CLLZ *post-processing* SDE *(as defined in Fig. 5) given in Fig. 6 satisfies* $\mathcal{D}_{\mathsf{identical\text{-}cipher}}$*-indistinguishability from random anti-piracy (for more details on the definition, see the full version).*

The proof of Theorems 6 and 7 is given in full version.

4.2 Copy-Protection for PRFs with Preponed Security

We first introduce the definition of *preponed security* in Sect. 4.2 and then we present the constructions of copy-protection in Sect. 4.2.

Definition. We introduce a new security notion for copy-protection called *preponed security*.

Consider a pseudorandom function family $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$, where $\mathcal{F}_\lambda = \{f_k : \{0,1\}^{\ell(\lambda)} \to \{0,1\}^{\kappa(\lambda)} \ : \ k \in \{0,1\}^\lambda\}$. Moreover, f_k can be implemented using a polynomial-sized circuit, denoted by C_k.

Definition 3 (Preponed Security). *A copy-protection scheme* $\mathsf{CP} = (\mathsf{CopyProtect}, \mathsf{Eval})$ *for* $\mathcal{F}$ *(see the full version for the formal definition) satisfies* $\mathcal{D}_{\mathcal{X}}$*-preponed security if for any QPT* $(\mathcal{A}, \mathcal{B}, \mathcal{C})$*, there exists a negligible function* negl *such that:*

$$\Pr[\mathsf{PreponedExpt}^{(\mathcal{A},\mathcal{B},\mathcal{C}),\mathcal{F},\mathcal{U}}\left(1^\lambda\right) = 1] \leq \frac{1}{2} + \mathsf{negl}.$$

where PreponedExpt *is defined in Fig. 7.*

We consider two instantiations of $\mathcal{D}_{\mathcal{X}}$*:*

1. $\mathcal{U}$ *which is the product of uniformly random distribution on* $\{0,1\}^\ell$*, meaning* $x_1, x_2 \leftarrow \mathcal{U}(1^\lambda)$ *where* $x_1, x_2 \xleftarrow{\$} \{0,1\}^\ell$ *independently.*
2. $\mathsf{Id}_{\mathcal{U}}$ *which is the perfectly correlated distribution on* $\{0,1\}^\ell$ *with uniform marginals, meaning* $x, x \leftarrow \mathsf{Id}_{\mathcal{U}}(1^\lambda)$ *where* $x \xleftarrow{\$} \{0,1\}^\ell$.

Construction. The CLLZ copy-protection scheme is given in Figs. 8 and 9.

$\underline{\mathsf{PreponedExpt}^{(\mathcal{A},\mathcal{B},\mathcal{C}),\mathsf{CP},\mathcal{D}_{\mathcal{X}}}(1^\lambda)}$:

1. Ch samples $k \leftarrow \mathsf{KeyGen}(1^\lambda)$, then generates $\rho_{C_k} \leftarrow \mathsf{CopyProtect}(1^\lambda, C_k)$ and sends ρ_{f_k} to $\mathcal{A}$.
2. Ch samples $x^{\mathcal{B}}, x^{\mathcal{C}} \leftarrow \mathcal{D}_{\mathcal{X}}(1^\lambda)$, $b \xleftarrow{\$} \{0,1\}$. Let $y_1^{\mathcal{B}} = f(x^{\mathcal{B}}), y_1^{\mathcal{C}} = f(x^{\mathcal{C}})$, and $y_0^{\mathcal{B}} = y_1, y_0^{\mathcal{C}} = y_2$ where $y_1, y_2 \xleftarrow{\$} \{0,1\}^{\kappa(\lambda)}$. Ch gives $(y_b^{\mathcal{B}}, y_b^{\mathcal{C}})$ to Alice.
3. $\mathcal{A}(\rho_{C_k})$ outputs a bipartite state $\sigma_{\mathcal{B},\mathcal{C}}$.
4. Apply $(\mathcal{B}(x^{\mathcal{B}}, \cdot) \otimes \mathcal{C}(x^{\mathcal{C}}, \cdot))(\sigma_{\mathcal{B},\mathcal{C}})$ to obtain $(b_{\mathbf{B}}, b_{\mathbf{C}})$.
5. Output 1 if $b_{\mathbf{B}} = b_{\mathbf{C}} = b$.

Fig. 7. Preponed security experiment for copy-protection of PRFs with respect to the distribution $\mathcal{D}_{\mathcal{X}}$.

Tools: Punctrable and extractable PRF family $F_1 = (\mathsf{KeyGen}, \mathsf{Eval})$ (represented as $F_1(k,x) = \mathsf{PRF.Eval}(k,\cdot)$) and secondary PRF family F_2, F_3 with some special properties as noted in [CLLZ21]

$\mathsf{CopyProtect}(K_1)$:

1. Sample secondary keys K_2, K_3, and $\{\{|A_{i s_i, s'_i}\rangle\}_i\}$, and compute the coset state $\{\{|A_{i s_i, s'_i}\rangle\}_i\}$.
2. Compute $\tilde{P} \leftarrow \mathsf{iO}(P)$ where P is as given in Figure 9.
3. Output $\rho = (\tilde{P}, \{\{|A_{i s_i, s'_i}\rangle\}_i\})$.

$\mathsf{Eval}(\rho, x)$:

1. Interprete $\rho = (\tilde{P}, \{\{|A_{i s_i, s'_i}\rangle\}_i\})$.
2. Let $x = x_0 \| x_1 \| x_2$, where $x_0 = \ell_0$. For every $i \in [\ell_0]$, if $x_{0,i} = 1$ apply $H^{\otimes n}$ on $|A_{i s_i, s'_i}\rangle$. Let the resulting state be $|\psi_x\rangle$.
3. Run the circuit $\tilde{C}$ in superposition on the input registers (X, V) with the initial state $(x, |\psi_x\rangle)$ and measure the output register to get an output y.

Fig. 8. CLLZ copy-protection for PRFs.

Construction of Copy-Protection.

Proposition 1. *Assuming the existence of post-quantum* iO*, and one-way functions, and if there exists a* CLLZ *post-processing* SDE *scheme that satisfies* $\mathcal{D}_{\mathsf{ind\text{-}msg}}$*-indistinguishability from random anti-piracy (for more details on the definition, see the full version), then the* CLLZ *copy-protection construction*

$\underline{P}$:

Hardcoded keys $K_1, K_2, K_3, R_i^0, R_i^1$ for every $i \in [\ell_0]$ On input $x = x_0 \| x_1 \| x_2$ and vectors $v = v_1, \ldots v_{\ell_0}$.

1. If $F_3(K_3, x_1) \oplus x_2 = x_0 \| Q$ and $x_1 = F_2(K_2, x_0 \| Q)$:
   Hidden trigger mode: Treat Q as a classical circuit and output $Q(v)$.
2. Otherwise, check if the following holds: for all $i \in \ell_0$, $R^{x_{0,i}}(v_i) = 1$ (where $x_{0,i}$ is the i^{th} coordinate of x_0).
   Normal mode: If so, output $F_1(K_1, x)$ where $F_1() = \mathsf{PRF.Eval}()$ is the primary pseudorandom function family that is being copy-protected. Otherwise output $\perp$.

Fig. 9. Circuit P in CLLZ copy-protection of PRF.

in [CLLZ21, Section 7.3] (see Fig. 8) satisfies $\mathcal{U}$-preponed security (Definition 3).

Proposition 2. *Assuming the existence of post-quantum* iO*, and one-way functions, and if there exists a* CLLZ *post-processing* SDE *scheme that satisfies $\mathcal{D}_{\mathsf{identical\text{-}cipher}}$-indistinguishability from random anti-piracy, (for more details on the definition, see the full version), then the* CLLZ *copy-protection construction in [CLLZ21, Section 7.3] (see Fig. 8) satisfies $\mathsf{Id}_{\mathcal{U}}$-preponed security (Definition 3).*

The proofs of Propositions 1 and 2 can be found in full version.

4.3 UPO for Keyed Circuits from Copy-Protection with Preponed Security

Theorem 8. *Assuming Conjecture 2, the existence of post-quantum sub-exponentially secure* iO *and injective one-way functions, and the quantum hardness of Learning-with-errors problem (LWE), there is a construction of unclonable puncturable obfuscation satisfying $\mathcal{U}$-generalized* UPO *security (see Definition 2), for any generalized keyed puncturable circuit class $\mathfrak{C}$ in* P/poly*, see Sect. 2.1.*

Proof. The proof follows by combining Lemma 1 and Theorem 10. □

Theorem 9. *Assuming Conjecture 1, the existence of post-quantum sub-exponentially secure* iO *and injective one-way functions, and the quantum hardness of Learning-with-errors problem (LWE), there is a construction of unclonable puncturable obfuscation satisfying $\mathsf{Id}_{\mathcal{U}}$-generalized* UPO *security (see Definition 2), for any generalized keyed puncturable circuit class $\mathfrak{C}$ in* P/poly*, see Sect. 2.1.*

Proof. The proof follows by combining Lemma 1 and Theorem 11. □

In the construction given in Fig. 10, the PRF family $(\mathsf{KeyGen}, \mathsf{Eval})$ satisfies the requirements as in [CLLZ21] and has input length $n(\lambda)$ and output length m; PRG is a length-doubling injective pseudorandom generator with input length m, which can be constructed based on injective one-way functions.

Tools: PRF family $(\mathsf{KeyGen}, \mathsf{Eval})$ with same properties as needed in [CLLZ21], PRG, CLLZ copy-protection scheme $(\mathsf{CopyProtect}, \mathsf{Eval})$.

$\mathsf{Obf}(1^\lambda, W)$:

1. Sample a random key $k \leftarrow \mathsf{PRF.KeyGen}(1^\lambda)$.
2. Compute $\mathsf{iO}(P), \{\{|A_{i_{s_i,s'_i}}\rangle\}_i\} \leftarrow \mathsf{CLLZ.CopyProtect}(k)$.
3. Compute $\tilde{C} \leftarrow \mathsf{iO}(C)$ where $C = \mathsf{PRG} \cdot \mathsf{iO}(P)$.
4. Compute $\mathsf{iO}(D)$ where D takes as input x, v, y, and runs C on x, v to get y' and outputs $\perp$ if $y' \neq y$ or $y' = \perp$, else it runs the circuit W on x to output $W(x)$.
5. Output $\rho = (\{\{|A_{i_{s_i,s'_i}}\rangle\}_i\}, \tilde{C}, \mathsf{iO}(D))$.

$\mathsf{Eval}(\rho, x)$

1. Interprete $\rho = (\{\{|A_{i_{s_i,s'_i}}\rangle\}_i\}, \tilde{C}, \mathsf{iO}(D))$.
2. Let $x = x_0 \| x_1 \| x_2$, where $x_0 = \ell_0$. For every $i \in [\ell_0]$, if $x_{0,i} = 1$ apply $H^{\otimes n}$ on $|A_{i_{s_i,s'_i}}\rangle$. Let the resulting state be $|\psi_x\rangle$.
3. Run the circuit $\tilde{C}$ in superposition on the input registers (X, V) with the initial state $(x, |\psi_x\rangle)$ and then measure the output register to get an output y. Let the resulting state quantum state on register V be σ.
4. Run $\mathsf{iO}(D)$ on the registers X, V, Y in superposition where registers X, Y are initialized to classical values x, y and then measure the output register to get an output z. Output z.

Fig. 10. Construction of a UPO scheme.

Lemma 1. *The construction given in Fig. 10 satisfies* $(1-\mathsf{negl})$-UPO *correctness for any generalized puncturable keyed circuit class in* $\mathsf{P/poly}$ *for some negligible function* negl.

The proof is given in the full version.

Theorem 10. *Assuming Conjecture 2, post-quantum sub-exponentially secure* iO *and injective one-way functions, and the quantum hardness of Learning-with-errors problem (LWE), the construction given in Fig. 10 satisfies* $\mathcal{U}$*-generalized*

unclonable puncturable obfuscation security (see Sect. 2.1) for any generalized puncturable keyed circuit class in $\mathsf{P/poly}$.

Proof. The proof follows by combining Lemma 2 and Proposition 1, and Theorem 6. □

Theorem 11. *Assuming Conjecture 1, the existence of post-quantum sub-exponentially secure* iO *and injective one-way functions, and the quantum hardness of Learning-with-errors problem (LWE), the construction given in Fig. 10 satisfies* $\mathsf{Id}_{\mathcal{U}}$*-generalized unclonable puncturable obfuscation security (see Sect. 2.1) for any generalized puncturable keyed circuit class in* $\mathsf{P/poly}$.

Proof. The proof follows by combining Lemma 3 , Proposition 2, and Theorem 7. □

Lemma 2. *Assuming the existence of post-quantum* iO*, injective one-way functions, and that* CLLZ *copy protection construction for* PRF*s given in Fig. 8, satisfies* $\mathcal{U}$*-preponed security (defined in Definition 3, the construction given in Fig. 10 for* $\mathcal{W}$ *satisfies* $\mathcal{U}$*-generalized* UPO *security guarantee (see Sect. 2.1), for any puncturable keyed circuit class* $\mathcal{W} = \{\{W_s\}_{s\in\mathcal{K}_\lambda}\}_\lambda$ *in* $\mathsf{P/poly}$.

Lemma 3. *Assuming the existence of post-quantum* iO*, injective one-way functions, and that* CLLZ *copy protection construction for* PRF*s given in Fig. 8, satisfies* $\mathsf{Id}_{\mathcal{U}}$*-preponed security (defined in Definition 3), the construction given in Fig. 10 for* $\mathcal{W}$ *satisfies* $\mathsf{Id}_{\mathcal{U}}$*-generalized* UPO *security guarantee (see Sect. 2.1), for any puncturable keyed circuit class* $\mathcal{W} = \{\{W_s\}_{s\in\mathcal{K}_\lambda}\}_\lambda$ *in* $\mathsf{P/poly}$.

The proof of the Lemmas 2 and 3 can be found in the full version.

5 Construction of UPO from Quantum State iO

Recently, Coladangelo and Gunn [CG23] proposed the definition of quantum state indistinguishability obfuscation (qsiO) and presented a candidate construction of qsiO. In this section, we show how to construct UPO from qsiO, assuming unclonable encryption and injective one-way functions. As an intermediate tool, we consider a variant of private-key unclonable encryption introduced in [CG23], called key-testable (private-key) unclonable encryption.

Key-Testable Unclonable Encryption. A key-testable unclonable encryption scheme [CG23] is an unclonable encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ where, given a ciphertext ρ and a key sk', we can efficiently determine with probability 1 whether ρ was generated using the secret key sk' or not.

Formally, a key-testable private-key unclonable encryption is associated with an additional QPT algorithm Test that takes as input a key $\mathsf{sk} \in \{0,1\}^\lambda$, a quantum ciphertext ρ and outputs a bit b such that for every pair of keys $\mathsf{sk}, \mathsf{sk}' \in \{0,1\}^\lambda$, $\mathsf{sk} \neq \mathsf{sk}'$, a message $m \in \{0,1\}^n$,

$$\Pr\left[b \leftarrow \mathsf{Test}(\mathsf{sk}', \rho) : \begin{smallmatrix} \mathsf{sk}\leftarrow\mathsf{Gen}(1^\lambda), \\ \rho\leftarrow\mathsf{Enc}(\mathsf{sk},m), \\ b=\delta_{\mathsf{sk}}(\mathsf{sk}') \end{smallmatrix}\right] = 1,$$

where δ_{sk} is the function that is 1 at sk and 0 everywhere else, and Enc is the encryption algorithm for the unclonable encryption scheme.

Unclonable Encryption Schemes with Uniform Key-Generation. In addition to the key-testable property, for the purpose of our construction of UPO, we also require that the key generation algorithm of the underlying unclonable encryption scheme samples the secret key uniformly at random from $\{0,1\}^\lambda$. We need this restriction on the key-generation algorithm because, in our construction, the output distribution of the key-generation algorithm determines the challenge distribution $\mathcal{D}_{\mathcal{X}}$, i.e., the distribution of the point to be punctured.

We next show that given an unclonable encryption scheme, we can generically transform it into another scheme satisfying the above-mentioned restriction.

Theorem 12. *An unclonable encryption scheme* $\mathsf{UE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *can be transformed into another unclonable encryption scheme* $\mathsf{UE}' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ *such that the output distribution of* Gen' *is uniform. Moreover,* UE' *supports messages of the same length as* UE.

Proof. Given $\mathsf{UE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, we define $\mathsf{UE}' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ as follows.

- $\mathsf{Gen}'(1^\lambda)$: Sample $k' \xleftarrow{\$} \{0,1\}^\lambda$, and output k'.
- $\mathsf{Enc}'(k', m)$: Generate $k \leftarrow \mathsf{Gen}(1^\lambda)$, and then generate $\rho \leftarrow \mathsf{Enc}(k, m)$. Output $\rho' = (\rho, k \oplus k')$.
- $\mathsf{Dec}'(k', \rho')$: Interprete $\rho' = (\rho, c)$. Generate $k = k' \oplus c$, and then generate $m \leftarrow \mathsf{Dec}(k, \rho)$. Output m.

Clearly, the correctness of UE' is immediate from the correctness of UE. Furthermore, Gen' satisfies the property mentioned in the theorem.

To argue security, let $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ be an adversary that violates unclonable indistinguishability security of UE' (see the formal definition in the full version). Consider the following reduction $(\mathcal{R}_{\mathcal{A}}, \mathcal{R}_{\mathcal{B}}, \mathcal{R}_{\mathcal{C}})$ that uses $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ to violate the unclonable indistinguishability security of UE.

1. $\mathcal{R}_{\mathcal{A}}$ runs $\mathcal{A}$ on the security parameter and get backs a message pair (m_0, m_1), which she sends to the challenger Ch.
2. Ch sends a ciphertext ρ.
3. $\mathcal{R}_{\mathcal{A}}$ samples $r \xleftarrow{\$} \{0,1\}^\lambda$, and feeds $\rho' = (\rho, r)$ to $\mathcal{A}$ who then outputs a bipartite state $\sigma_{(\mathcal{B},\mathcal{C})}$. $\mathcal{R}_{\mathcal{A}}$ outputs $(r_{\mathcal{B}}, \sigma_{(\mathcal{B},\mathcal{C})}, r_{\mathcal{C}})$ where $r_{\mathcal{B}} = r_{\mathcal{C}} = r$.
4. $\mathcal{R}_{\mathcal{B}}$ (respectively, $\mathcal{R}_{\mathcal{C}}$) on receiving $(r_{\mathcal{B}}, \sigma_{\mathcal{B}})$ (respectively, $(r_{\mathcal{C}}, \sigma_{\mathcal{C}})$) from $\mathcal{R}_{\mathcal{A}}$ and a key k from the challenger, runs $\mathcal{B}$ on $(r_{\mathcal{B}} \oplus k, \sigma_{\mathcal{B}})$ (respectively, $\mathcal{C}$ on $(r_{\mathcal{C}} \oplus k, \sigma_{\mathcal{C}})$), and outputs $\mathcal{B}$'s output (respectively, $\mathcal{C}$'s output).

It follows that the success probability of $(\mathcal{R}_{\mathcal{A}}, \mathcal{R}_{\mathcal{B}}, \mathcal{R}_{\mathcal{C}})$ is the same as that of $(\mathcal{A}, \mathcal{B}, \mathcal{C})$, which completes the proof of the theorem. □

It was shown in [CG23] that assuming qsio, the key testable property can be generically attached to any unclonable encryption scheme that satisfies the above restriction on the key generation algorithm.

Theorem 13 (Adapted from [CG23, Theorem 16]). *If injective one-way functions and* qsio *exist, then any unclonable bit encryption scheme (with the key generation algorithm outputting a uniformly random key from* $\{0,1\}^\lambda$*) can be compiled into one with key testing (with the same key generation algorithm).*

For the rest of the section, for any key testable unclonable encryption scheme, we will assume that the Gen algorithm has uniform output distribution.Hence we will use a triplet of algorithms $(\mathsf{Enc}, \mathsf{Dec}, \mathsf{Test})$ to represent a key testable unclonable encryption scheme and in particular, we omit Gen from the description.

UPO from qsiO. We consider the following tools:

- A key-testable unclonable bit encryption scheme $\mathsf{UE} = (\mathsf{Enc}, \mathsf{Dec}, \mathsf{Test})$.
- Quantum state iO scheme, denoted by $\mathsf{qsio} = (\mathsf{Obf}, \mathsf{Eval})$.

Theorem 14. *Suppose there exists a key-testable unclonable bit encryption scheme,* $\mathsf{UE} = (\mathsf{Enc}, \mathsf{DecTest})$*. Then, any* qsio *scheme* $(\mathsf{Obf}, \mathsf{Eval})$ *for* P/poly *is also a* UPO *scheme satisfying* $\mathsf{Id}_{\mathcal{U}}$*-generalized* UPO *security guarantee (see Sect. 2.1), for any puncturable keyed circuit class* $\mathcal{W} = \{\{W_s\}_{s\in\mathcal{K}_\lambda}\}_\lambda$ *in* P/poly*.*

Proof. The correctness is immediate from the correctness of the qsio scheme.

Next, we prove security. Let $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ be a QPT adversary in the generalized UPO security experiment given in Fig. 3 with $\mathcal{D}_{\mathcal{X}} = \mathsf{Id}_{\mathcal{U}}$.

Hybrid_0 : Same as the security experiment given in Fig. 3.

1. $\mathcal{A}$ sends a key $s \in \mathcal{K}_\lambda$ and function μ[11] to Ch.
2. Ch samples $x^* \xleftarrow{\$} \{0,1\}^{n(\lambda)}$, and a bit $b \xleftarrow{\$} \{0,1\}$.
3. Ch generates $\tilde{\rho}_0 \leftarrow \mathsf{Obf}(1^\lambda, W_s)$, and $\tilde{\rho}_1 \leftarrow \mathsf{Obf}(1^\lambda, W_{s,x^*,\mu})$, where $W_{s,x^*,\mu} \leftarrow \mathsf{GenPuncture}(s, x^*, x^*, \mu, \mu)$.
4. Ch sends $\tilde{\rho}_b$ to $\mathcal{A}$.
5. $\mathcal{A}(\tilde{\rho}_b)$ outputs a bipartite state $\sigma_{\mathcal{B},\mathcal{C}}$.
6. Apply $(\mathcal{B}(x^*,\cdot) \otimes \mathcal{C}(x^*,\cdot))(\sigma_{\mathcal{B},\mathcal{C}})$ to obtain $(b_{\mathbf{B}}, b_{\mathbf{C}})$.
7. Output 1 if $b_{\mathbf{B}} = b_{\mathbf{C}} = b$.

Hybrid_1 :

1. $\mathcal{A}$ sends a key $s \in \mathcal{K}_\lambda$ and function μ to Ch.
2. Ch samples $x^* \xleftarrow{\$} \{0,1\}^{n(\lambda)}$, and a bit $b \xleftarrow{\$} \{0,1\}$.
3. Ch generates $\tilde{\rho}_b \leftarrow \mathsf{Obf}(1^\lambda, (C, \rho_b))$ where $\rho_b \leftarrow \mathsf{UE.Enc}(x^*, b)$ and C is the circuit that on input (x, ρ_b), first checks if $\mathsf{UE.Test}(x, \rho_b)$ rejects, in which case, C outputs $W_s(x)$. Else, C runs $d \leftarrow \mathsf{UE.Dec}(x, \rho_b)$ and if $d = 0$ outputs $W_s(x)$ else outputs $\mu(x)$.

[11] In the security experiment in Fig. 3, $\mathcal{A}$ sends two functions $\mu_{\mathcal{B}}$, $\mu_{\mathcal{C}}$ but since in this proof, $\mathcal{D}_{\mathcal{X}} = \mathsf{Id}_{\mathcal{U}}$, the second function $\mu_{\mathcal{C}}$ is redundant. Hence, for the sake of the proof, we can assume, without loss of generality, that $\mathcal{A}$ sends a *single* function μ to Ch.

4. Ch sends $\tilde{\rho}_b$ to $\mathcal{A}$.
5. $\mathcal{A}(\tilde{\rho}_b)$ outputs a bipartite state $\sigma_{\mathcal{B},\mathcal{C}}$.
6. Apply $(\mathcal{B}(x^*,\cdot)\otimes\mathcal{C}(x^*,\cdot))(\sigma_{\mathcal{B},\mathcal{C}})$ to obtain $(b_{\mathbf{B}}, b_{\mathbf{C}})$.
7. Output 1 if $b_{\mathbf{B}} = b_{\mathbf{C}} = b$.

Observe that W_s and (C,ρ_0) are functionally equivalent. Here, (C,ρ_0) represents an implementation of a classical function that maps x to $C(\rho,x)$. Similarly, $W_{s,x^*,\mu}$ and (C,ρ_1) are functionally equivalent. From the security of qsio, it follows that the hybrids Hybrid_0 and Hybrid_1 are computationally indistinguishable.

Next, we give a reduction $(R_{\mathcal{A}}, R_{\mathcal{B}}, R_{\mathcal{C}})$ from Hybrid_1 to the unclonable indistinguishability experiment for UE as follows.

- $R_{\mathcal{A}}$ sends $(0,1)$ as the challenge message pair to the challenger.
- Challenger sends a ciphertext ρ.
- $R_{\mathcal{A}}$ generates (C,ρ) (as described in Hybrid_1), and then computes $\tilde{\rho} \leftarrow \mathsf{Obf}(1^\lambda,(C,\rho))$.
- $R_{\mathcal{A}}$ feeds $\tilde{\rho}$ to $\mathcal{A}$ and outputs a bipartite state $\sigma_{B,C}$.
- $R_{\mathcal{B}}$ (respectively, $R_{\mathcal{C}}$) on receiving x from the challenger, runs $\mathcal{B}$ (respectively, $\mathcal{C}$) on $\sigma_{\mathcal{B}}$ (respectively, $\sigma_{\mathcal{C}}$) and x, and outputs $\mathcal{B}'s$ output (respectively, $\mathcal{C}$'s output).

It follows that the advantage of the QPT adversary $(\mathcal{A},\mathcal{B},\mathcal{C})$ in breaking UPO security is within a negligible additive factor of the advantage of the QPT adversary in breaking the unclonable indistinguishability of UE. This completes the proof of generalized UPO security for $(\mathsf{Obf},\mathsf{Eval})$. □

Combining Theorems 12 to 14, we conclude the following.

Corollary 3. *Suppose there exists a post-quantum injective one-way function and an unclonable bit encryption scheme* UE*. Then, any* qsio *scheme* $(\mathsf{Obf},\mathsf{Eval})$ *is also a* UPO *scheme satisfying* $\mathsf{Id}_{\mathcal{U}}$*-generalized* UPO *security guarantee (see Sect. 2.1), for any puncturable keyed circuit class* $\mathcal{W} = \{\{W_s\}_{s\in\mathcal{K}_\lambda}\}_\lambda$ *in* $\mathsf{P/poly}$.

Acknowledgements. A.B. has received funding from 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.

P.A. is supported in part by the National Science Foundation under Grant No. 2329938 and Grant No. 2341004.

We thank Supartha Podder for discussions during the early stages of the project. We are grateful to the anonymous reviewers of Crypto 2024 for various suggestions to improve the paper and specifically, for suggesting the use of the key-testable unclonable encryption [CG23] in constructing unclonable puncturable obfuscation from quantum state indistinguishability obfuscation.

References

[Aar09] Aaronson, S.: Quantum copy-protection and quantum money. In: 2009 24th Annual IEEE Conference on Computational Complexity, pp. 229–242. IEEE (2009)

[Aar16] Aaronson, S.: The Complexity of Quantum States and Transformations: From Quantum Money to Black Holes (2016). arXiv:1607.05256 [quant-ph]

[AC02] Adcock, M., Cleve, R.: A quantum Goldreich-Levin theorem with cryptographic applications. In: Alt, H., Ferreira, A. (eds.) STACS 2002. LNCS, vol. 2285, pp. 323–334. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45841-7_26

[AC12] Aaronson, S., Christiano, P.: Quantum money from hidden subspaces. In: Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing. STOC '12, pp. 41–60. Association for Computing Machinery. New York, New York, USA (2012). ISBN 9781450312455. https://doi.org/10.1145/2213977.2213983

[AK21] Ananth, P., Kaleoglu, F.: Unclonable encryption, revisited. In: Nissim, K., Waters, B. (eds.) TCC 2021. LNCS, vol. 13042, pp. 299–329. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-90459-3_11

[AKL+22] Ananth, P., Kaleoglu, F., Li, X., Liu, Q., Zhandry, M.: On the feasibility of unclonable encryption, and more. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022. LNCS, vol. 13508, pp. 212–241. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-15979-4_8

[AKL23] Ananth, P., Kaleoglu, F., Liu, Q.: Cloning games: a general framework for unclonable primitives. arXiv preprint arXiv:2302.01874 (2023)

[AKY24] Ananth, P., Kaleoglu, F., Yuen, H.: Simultaneous haar indistinguishability with applications to unclonable cryptography. arXiv preprint arXiv:2405.10274 (2024)

[AL21] Ananth, P., La Placa, R.L.: Secure software leasing. In: Canteaut, A., Standaert, F.-X. (eds.) EUROCRYPT 2021. LNCS, vol. 12697, pp. 501–530. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77886-6_17

[APV23] Ananth, P., Poremba, A., Vaikuntanathan, V.: Revocable cryptography from learning with errors. In: Rothblum, G., Wee, H. (eds.) TCC 2023. LNCS, vol. 14372, pp. 93–122. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-48624-1_4

[BGI+01] Barak, B., et al.: On the (im)possibility of obfuscating programs. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 1–18. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_1

[BGI14] Boyle, E., Goldwasser, S., Ivan, I.: Functional signatures and pseudorandom functions. In: Krawczyk, H. (ed.) PKC 2014. LNCS, vol. 8383, pp. 501–519. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54631-0_29

[BGS13] Broadbent, A., Gutoski, G., Stebila, D.: Quantum one-time programs. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8043, pp. 344–360. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40084-1_20

[BKL23] Broadbent, A., Karvonen, M., Lord, S.: Uncloneable quantum advice. arXiv preprint arXiv:2309.05155 (2023)

[BL20] Broadbent, A., Lord, S.: Uncloneable quantum encryption via oracles. en. In: Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020). https://doi.org/10.4230/LIPICS.TQC.2020.4, https://drops.dagstuhl.de/opus/volltexte/2020/12063/

[BPR15] Bitansky, N., Paneth, O., Rosen, A.: On the cryptographic hardness of finding a Nash equilibrium. In: 2015 IEEE 56th Annual Symposium on Foundations of Computer Science, pp. 1480–1498. IEEE (2015)

[BS16] Ben-David, S., Sattath, O.: Quantum Tokens for Digital Signatures (2016). https://doi.org/10.48550/ARXIV.1609.09047, https://arxiv.org/abs/1609.09047

[BS20] Behera, A., Sattath, O.: Almost public quantum coins. arXiv preprint arXiv:2002.12438 (2020)

[BW13] Boneh, D., Waters, B.: Constrained pseudorandom functions and their applications. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013. LNCS, vol. 8270, pp. 280–300. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-42045-0_15

[BZ17] Boneh, D., Zhandry, M.: Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. Algorithmica **79**, 1233–1285 (2017)

[CG23] Coladangelo, A., Gunn, S.: How to use quantum indistinguishability obfuscation. arXiv preprint arXiv:2311.07794 (2023)

[CHV23] Chevalier, C., Hermouet, P., Vu, Q.-H.: Semi-quantum copy-protection and more. Cryptology ePrint Archive (2023)

[CLLZ21] Coladangelo, A., Liu, J., Liu, Q., Zhandry, M.: Hidden cosets and applications to unclonable cryptography. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021, Part I. LNCS, vol. 12825, pp. 556–584. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-84242-0_20

[Die82] Dieks, D.G.B.J.: Communication by EPR devices. Phys. Lett. A **92**(6), 271–272 (1982)

[Gao15] Gao, J.: Quantum union bounds for sequential projective measurements. Phys. Rev. A **92**(5), 052331 (2015)

[GGH+16] Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. SIAM J. Comput. **45**(3), 882–929 (2016)

[GGHR14] Garg, S., Gentry, C., Halevi, S., Raykova, M.: Two-round secure MPC from indistinguishability obfuscation. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 74–94. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54242-8_4

[GMR23] Goyal, V., Malavolta, G., Raizes, J.: Unclonable commitments and proofs. Cryptology ePrint Archive (2023)

[Got02] Gottesman, D.: Uncloneable encryption (2002). https://doi.org/10.48550/ARXIV.QUANT-PH/0210062. url: https://arxiv.org/abs/quant-ph/0210062

[GZ20] Georgiou, M., Zhandry, M.: Unclonable decryption keys. IACR Cryptology ePrint Archive https://eprint.iacr.org/2020/877 (2020)

[JK23] Jawale, R., Khurana, D.: Unclonable non-interactive zero-knowledge. arXiv preprint arXiv:2310.07118 (2023)

[JLS21] Jain, A., Lin, H., Sahai, A.: Indistinguishability obfuscation from well-founded assumptions. In: Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing, pp. 60–73 (2021)

[KN23] Kitagawa, F., Nishimaki, R.: One-out-of-many unclonable cryptography: definitions, constructions, and more. arXiv preprint arXiv:2302.09836 (2023)

[KT22] Kundu, S., Tan, E.Y.-Z.: Device-independent uncloneable encryption. In: arXiv preprint arXiv:2210.01058 (2022)

[LLQZ22] Liu, J., Liu, Q., Qian, L., Zhandry, M.: Collusion resistant copy-protection for watermarkable functionalities. In: Kiltz, E., Vaikuntanathan, V. (eds.) TCC 2022, Part I. Lecture Notes in Computer Science, vol. 13747, pp. 294–323. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-22318-1_11

[LMZ23] Liu, J., Montgomery, H., Zhandry, M.: Another round of breaking and making quantum money: how to not build it from lattices, and more. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023. LNCS, vol. 14004, pp. 611–638. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-30545-0_21

[RS19] Radian, R., Sattath, O.: Semi-quantum money. In: Proceedings of the 1st ACM Conference on Advances in Financial Technologies, pp. 132–146 (2019)

[RZ21] Roberts, B., Zhandry, M.: Franchised quantum money. In: Tibouchi, M., Wang, H. (eds.) ASIACRYPT 2021. LNCS, vol. 13090, pp. 549–574. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-92062-3_19

[Shm22] Shmueli, O.: Public-key Quantum money with a classical bank. In: Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing, pp. 790–803 (2022)

[STHY23] Sudo, K., Tezuka, M., Hara, K., Yoshida, Y.: Quantum search-to-decision reduction for the LWE problem. In: El Mrabet, N., De Feo, L., Duquesne, S. (eds.) AFRICACRYPT 2023. LNCS, vol. 14064, pp. 395–413. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-37679-5_17

[SW14] Sahai, A., Waters, B.: How to use indistinguishability obfuscation: deniable encryption, and more. In: Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing, pp. 475–484 (2014)

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

[WZ82] Wootters, W.K., Zurek, W.H.: A single quantum cannot be cloned. Nature **299**(5886), 802–803 (1982)

[Zha19] Zhandry, M.: Quantum lightning never strikes the same state twice. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019. LNCS, vol. , pp. 408–438. Springer, Cham (2019). ISBN 978-3-030-17659-4, https://doi.org/10.1007/978-3-030-17659-4_14

[Zha23] Zhandry, M.: Quantum money from abelian group actions. arXiv preprint arXiv:2307.12120 (2023)