Point-Function Obfuscation:
A Framework and Generic Constructions

Mihir Bellare[1] Igors Stepanovs[2]

Abstract

We unify the many prior variants of point-function obfuscation via a definitional framework in which security is parameterized by a class of algorithms we call target generators, with different notions corresponding to different choices of this class. This leads to an elegant question, namely whether it is possible to provide a generic construction, meaning one that takes an arbitrary class of target generators and returns a point-function obfuscator secure for it. We answer this in the affirmative with three generic constructions, the first based on indistinguishability obfuscation, the second on deterministic public-key encryption and the third on universal computational extractors. By exploiting known constructions of the primitives assumed, we obtain a host of new point-function obfuscators, including many under standard assumptions.

Keywords: Obfuscation, Point-function obfuscation, One-way functions, Deterministic encryption, UCE.

[1] Department of Computer Science & Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. Email: mihir@eng.ucsd.edu. URL: http://cseweb.ucsd.edu/~mihir/. Supported in part by NSF grants CNS-1116800 and CNS-1228890. This work was done in part while the author was visiting the Simons Institute for the Theory of Computing, supported by the Simons Foundation and by the DIMACS/Simons Collaboration in Cryptography through NSF grant CNS-1523467.

[2] Department of Computer Science & Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. Email: istepano@eng.ucsd.edu. Supported in part by NSF grants CNS-1116800 and CNS-1228890.

1 Introduction

This paper has no deep technical results. It aims, rather, to bring some order and unity to what felt, to us at least, like a difficult to navigate area, namely the related but different notions that go under the broad name of point-function obfuscation. We provide a definitional framework parameterized by a class $\mathbf{X}$ of objects we call *target generators* that allows us to recover different notions in the literature as each corresponding to a choice of $\mathbf{X}$. This taxonomy leads to a compelling and general new question: Is it possible to find a *generic construction*, meaning a compiler that given an arbitrary $\mathbf{X}$ returns a point function obfuscator secure relative to it? We answer this in the affirmative by providing three such generic constructions, contributing to a simpler and more modular use of point-function obfuscation in applications and also yielding as special cases many new point-function obfuscators for particular classes of interest.

Background and notions. The most desirable form of obfuscation is VBBO (Virtual Black Box Obfuscation), where the obfuscated circuit is no more useful than an oracle for the circuit. Unfortunately, VBBO for all circuits is impossible [2, 33, 14]. The natural question then was, are there particular classes of circuits that one can obfuscate? Point functions emerged as the canonical candidate. Indeed, not only are point functions a basic and natural starting point but also their obfuscation has many applications [22, 35, 41, 33, 24, 25, 10, 13, 36, 20].

A point function with target $k \in \{0,1\}^*$ is the circuit $\mathbf{I}_k$ that on input $k' \in \{0,1\}^{|k|}$ returns 1 if $k' = k$ and 0 otherwise. A point-function obfuscator Obf takes input $\mathbf{I}_k$ and returns another circuit $\overline{\mathrm{P}}$ that is functionally equivalent to $\mathbf{I}_k$. (On input $k' \in \{0,1\}^{|k|}$ it also returns 1 if $k' = k$ and 0 otherwise.) Security requires that $\overline{\mathrm{P}}$ hides k. The most basic formalization [13] is that the adversary, given $\overline{\mathrm{P}}$ and auxiliary information a, be unable to tell whether $\overline{\mathrm{P}}$ is an obfuscation of $\mathbf{I}_{k_1}$ or of $\mathbf{I}_{k_0}$ when (k_1, a) is drawn from some known distribution and k_0 is drawn at random. This is not achievable if the distribution is arbitrary. For example, it could always pick k_1 to be the string of all zeroes, and the adversary could test whether or not $\overline{\mathrm{P}}$ returns 1 on input that string. The minimal requirement for security is that k_1 is unpredictable given a. Formalizations in a VBBO style have also been given but we will stick to this one because it is simpler and the one used by modern applications.

As work on the obfuscation of point functions evolved, we saw the introduction, consideration and use of a rather large number of variants of the basic notion. Early works [22, 26, 35, 41] did not have auxiliary information. The latter was introduced by GK [33] and is important for applications [13, 20, 21]. Unpredictability sometimes means that polynomial-time adversaries have negligible advantage [22, 41, 13], sometimes that polynomial-time adversaries have sub-exponential advantage [28] and sometimes that unbounded adversaries have negligible advantage [6]. Sometimes, a single point function is being obfuscated and at other times many [24]. (This was called composable point function obfuscation. Note that many does not reduce to one by a hybrid argument since the points may be related so it must be considered a separate notion.) And so on.

Constructions. The simplicity of point functions raised the hope that obfuscating them would be easy. Surprisingly, this has not been true even for the basic case (one point, no auxiliary input) and even less when auxiliary input is present. Indeed, there are few known constructions and those that exist rely on strong assumptions.

A primary construction of AIPO (Auxiliary-Input Point-Function Obfuscation) is from the AI-DHI (Auxiliary-Input Diffie-Hellman Inversion) assumption [22, 13]. The assumption states that there is a prime-order group $\mathbb{G}$ such that if we pick random r, s from $\mathbb{G}$ then it is hard to distinguish between (r, r^k) and (r, s), even when given auxiliary information a about k, as long as this information a is k-prediction-precluding. Obtaining AIPO from this is direct. Namely, to obfuscate $\mathbf{I}_k$, we pick a random r from $\mathbb{G}$, let $s = r^k$ and return the circuit $\mathrm{C}_{r,s}$ that on input k' returns 1 if $r^{k'} = s$ and 0 otherwise. But this is not entirely satisfactory because security amounts exactly to the AI-DHI assumption and is thus effectively assumed rather than proved. The benefit of a security proof appears when the assumption is simpler or weaker than the goal, which is not the case here. Another issue is that if VGBO (Virtual Grey Box Obfuscation) for all circuits is possible then the AI-DHI-based AIPO is insecure [9]. That is, either VGBO or AI-DHI-based AIPO must fail. This does not necessarily mean that the AI-DHI-based AIPO fails (it could well be VGBO that fails) but this still motivates finding new constructions not subject to even such a conditional impossibility result, in particular ones that can co-exist with VGBO.

Wee [41] provides a point-function obfuscator based on a fixed permutation about which a novel strong uninvertibility assumption is made. He only proves security in the absence of auxiliary information, and GK [33] show that the construction does not in fact provide security in the presence of auxiliary information. However BP [13] specify an extension of Wee's construction with a family of permutations rather than a fixed one, and show, under a novel assumption called Assumption 2.1 in their paper, that it achieves security with auxiliary inputs. BP [13] explain that Assumption 2.1 asks for (a weak form of) extractability, making it a strong assumption in light of the impossibility of related extractable primitives [12].

There are simple constructions in the ROM [35]. DKL [28] give a construction for sub-exponentially hard to predict target points under a novel assumption called LSN. BHK [6] give a UCE-based construction for statistically hard to predict targets and no auxiliary information.

In summary, there are few constructions and they all use strong and novel assumptions. Also, each construction achieves a different variant of the notion and it is hard to sort out, or say in a precise yet concise way, what has been done. The latter is due to lack of language which is provided by the framework that we now discuss.

FRAMEWORK. We define a *target generator* X as a polynomial-time algorithm that on input the security parameter returns a vector $\mathbf{k}$ of target points together with auxiliary information a. To measure security of a candidate point-function obfuscator Obf relative to X, we associate to an adversary $\mathcal{A}$ its advantage $\mathsf{Adv}^{\mathsf{pfo}}_{\mathsf{Obf},\mathsf{X},\mathcal{A}}(\cdot)$ in guessing the challenge bit b in the following game. We run X to get $(\mathbf{k}, a)$. We let $\overline{\mathbf{P}}$ be the vector obtained by obfuscating the targets in $\mathbf{k}$ ($b = 1$) or by obfuscating the same number of random, independent targets ($b = 0$). The input to $\mathcal{A}$ is $\overline{\mathbf{P}}$ and a. Now we let $\mathbf{X}$ be a class (set) of target generators X and say that obfuscator Obf is PFO[$\mathbf{X}$]-secure if $\mathsf{Adv}^{\mathsf{pfo}}_{\mathsf{Obf},\mathsf{X},\mathcal{A}}(\cdot)$ is negligible for all polynomial time $\mathcal{A}$ and all $\mathsf{X} \in \mathbf{X}$.

What we have here (for a formal definition see Section 2) is a notion of point-function obfuscation parameterized by a class of target generators. We view these as knobs. By turning these knobs (defining specific classes) we can capture specific restrictions, and by intersecting classes we can combine them, allowing us to speak precisely yet concisely about different variant notions that are unified in this way. In particular, in Section 2 we formalize a prediction game and advantage so that we can define the classes $\mathbf{X}^{\mathrm{cup}}$, $\mathbf{X}^{\mathrm{seup}}$ and $\mathbf{X}^{\mathrm{sup}}$ of computationally, sub-exponentially and statistically unpredictable target generators. We let $\mathbf{X}^{q(\cdot)}$ denote the class of target generators outputting $q(\cdot)$ target points and $\mathbf{X}^{\varepsilon}$ the class of target generators that produce no auxiliary information. (Formally it is the empty string.) Already we can characterize prior work in a precise way. For example PFO[$\mathbf{X}^{\mathrm{cup}} \cap \mathbf{X}^{\varepsilon} \cap \mathbf{X}^{1}$] is plain point function obfuscation where there is just one target point, no auxiliary information, and unpredictability is computational. This is achieved in [22, 26, 35, 41]. PFO[$\mathbf{X}^{\mathrm{cup}} \cap \mathbf{X}^{1}$] is AIPO, where there is again one target point, but auxiliary information is now present, while unpredictability continues to be computational [13]. PFO[$\mathbf{X}^{\mathrm{cup}}$] is composable AIPO, where there are many target points, auxiliary information is present, and unpredictability is computational. Other prior notions can be captured in obvious ways, and many natural new ones emerge for consideration.

GENERIC CONSTRUCTIONS. As we saw above, constructions so far have been ad hoc, targeting different security goals and using strong, novel assumptions to achieve them. The above framework allows us to frame a compelling question, namely whether there are generic constructions. By this we mean that we are handed an arbitrary class $\mathbf{X}$ of target generators and asked to craft an obfuscator that is PFO[$\mathbf{X}$]-secure. If we can do this, we can, in one unified swoop, obtain constructions for a wide variety of forms of PFO, not only ones considered in the past, but also new ones.

In this paper we provide three such generic constructions. The first is based on indistinguishability obfuscation, the second on deterministic public-key encryption and the third on UCE.

One natural objection at this point is that we know that PFO[$\mathbf{X}$] is not achievable for some choices of $\mathbf{X}$. For example, assuming iO, this is true for $\mathbf{X} = \mathbf{X}^{\mathrm{cup}}$, meaning composable PFO. (This follows by combining [19, 23].) So how can our constructions achieve PFO[$\mathbf{X}$] for any given $\mathbf{X}$? In fact, they do, and this, interestingly, yields new negative results, ruling out the primitives we start from for those particular values of $\mathbf{X}$. We will explain further below.

PFO from iO. The emergence of candidate constructions for iO (indistinguishability obfuscation) [31, 38, 11, 32] raised a natural hope, namely that one could obtain PFO from iO. But this has not happened. Despite the many powerful applications of iO, constructing point-function obfuscation from it has surprisingly evaded effort.

We show that iO plus a OWF yields PFO. More precisely, we show $\mathrm{iO} + \mathrm{OWF}[\mathbf{X}] \Rightarrow \mathrm{PFO}[\mathbf{X}]$: Given iO and a family of functions that is one-way relative to $\mathbf{X}$ as defined in Section 4.1 we can construct an obfuscator that is PFO[$\mathbf{X}$]-secure. The construction, result and proof are in Section 4.1. The idea is that to obfuscate $\mathbf{I}_k$ we pick at random a key fk for the OWF F (formally, the latter is a family of functions) and let $y = \mathsf{F}(fk, k)$. We consider the circuit C that hardwires fk, y and on input k' returns 1 if $\mathsf{F}(fk, k') = y$ and 0 otherwise. We then apply an indistinguishability obfuscator to C to produce the obfuscated point function. The security proof is a sequence of hybrids. Although we assume only iO, we exploit diO [2, 16, 1] in the proof in a manner similar to [8]. We will need it for circuits that differ only on one input, and in this case the result of BCP [16] says that an iO-secure obfuscator is also diO-secure, so the assumption remains iO. As part of the proof we state and prove a lemma reducing (d)iO on polynomially-many, related circuits to the usual single-circuit case.

We highlight the simplest case of this result as still being novel and of interest. Namely, given iO and an ordinary OWF, we achieve plain point-function obfuscation, PFO[$\mathbf{X}^{\mathrm{cup}} \cap \mathbf{X}^{\varepsilon} \cap \mathbf{X}^{1}$] in our notation. Previous constructions have been under assumptions that at this point seem less accepted than iO, and Wee [41] gives various arguments as to why this goal is hard under standard assumptions. Also on the negative side, combing our result with [19, 23] allows us, under iO, to rule out OWF[$\mathbf{X}^{\mathrm{cup}}$], one-way functions secure for polynomially-many, computationally unpredictable correlated inputs.

PFO from DPKE. Deterministic public key encryption (DPKE) [3] was motivated by applications to efficient searchable encryption [3]. It cannot provide IND-CPA security. Instead, BBO [3] provide a definition of a goal called PRIV which captures the best-possible security that encryption can provide subject to being deterministic. At this point many constructions of DPKE are known for various variant goals [3, 15, 4, 17, 30, 5, 17, 42, 44].

We show how to leverage these for point-function obfuscation via our second generic construction. We show that $\mathrm{PRIV1}[\mathbf{X}] \Rightarrow \mathrm{PFO}[\mathbf{X}]$. That is, given a deterministic public-key encryption scheme that is PRIV1 secure relative to $\mathbf{X}$ we can build a point-function obfuscator secure relative to the same class in a simple and natural way. Namely to obfuscate $\mathbf{I}_k$ we pick at random a public key pk and the associated secret key sk for the DPKE scheme and let c be the encryption of k under pk. The point-function obfuscation is the circuit C that hardwires pk, c and on input k', returns 1 if the encryption of k' under pk equals c, and 0 otherwise. The fact that the encryption is deterministic is used crucially to define the circuit. (The latter must be deterministic.) The secret key sk is discarded and not used in the construction. We note that we only require security of the DPKE scheme for a single message (PRIV1) so the negative result of Wichs [43] does not apply. The construction, result and proof are in Section 4.2.

From the LTDF-based DPKE scheme of BFO [15] and LTDFs from [39, 29, 42, 34, 45] we now get PFO[$\mathbf{X}^{\mathrm{sup}} \cap \mathbf{X}^{\varepsilon} \cap \mathbf{X}^{1}$]-secure obfuscators under a large number of standard assumptions. We also get PFO[$\mathbf{X}^{\mathrm{seup}} \cap \mathbf{X}^{1}$]-secure obfuscators under the DLIN, Subgroup Indistinguishability and LWE assumptions via [17, 44, 42]. On the negative side we can rule out PRIV1[$\mathbf{X}^{\mathrm{cup}}$]-secure DPKE under iO via [19, 23].

PFO from UCE. UCE [6] is a class of assumptions on function families crafted to allow instantiation of random oracles in certain settings. UCE security is parameterized so that we have UCE[$\mathbf{S}$] security of a family of functions for different choices of classes $\mathbf{S}$ of algorithms called sources. The parameterization is necessary because security is not achievable for the class of all sources. Different applications rely on UCE relative to different classes of sources [6, 18, 20, 37, 5, 27].

We show how to associate to any given class $\mathbf{X}$ of target generators a class $\mathbf{S}^{\mathbf{X}}$ of sources such that UCE[$\mathbf{S}^{\mathbf{X}}$] $\Rightarrow$ PFO[$\mathbf{X}$], meaning we can build a point-function obfuscator secure for $\mathbf{X}$ given a family of functions that is UCE[$\mathbf{S}^{\mathbf{X}}$]-secure. The definition of $\mathbf{S}^{\mathbf{X}}$ is given in Section 4.3, but what is most relevant here is that the strength of a UCE assumption is very sensitive to the choice of class of sources that parameterizes the assumption, and $\mathbf{S}^{\mathbf{X}}$ has good properties in this regard. The sources are what are called "split" in [6], and they inherit the unpredictability attributes of the target generators. UCE[$\mathbf{S}^{\mathbf{X}}$]-security is not achievable for all choices of $\mathbf{X}$ but the assumption is valid as far as we know for many choices of $\mathbf{X}$, yielding new constructions.

Discussion and further related work. Point-function obfuscation is sometimes formalized in a VBBO-style. An obvious critique of our framework is that it does not cover this. We don't believe there is benefit in doing so at this point. The definitions used by modern applications of point-function obfuscation are the ones from our framework and the indistinguishability-based formalism is easier to work with.

Target generators in our framework output a vector of targets, meaning we are in general considering the obfuscaton of multiple, related targets. (Intersecting with $\mathbf{X}^1$ gets us back to the single target case.) One may ask why bother since composable AIPO —PFO[$\mathbf{X}^{\mathrm{cup}}$] in our framework— is not possible assuming iO [19, 23]. But other forms of PFO involving multiple points, such as PFO[$\mathbf{X}^{\mathrm{sup}}$], are still of interest, and indeed we reach this. Also PFO[$\mathbf{X}$] is of interest for subsets of $\mathbf{X}^{\mathrm{cup}}$ such as $\mathbf{X}^{\mathrm{cup}} \cap \mathbf{X}^q$ for constant q, or even for polynomial q and generators which are block sources.

In concurrent and independent work, BM [21] take first steps towards a parameterized definition for point-function obfuscation. Ours goes further by allowing multiple targets and captures more existing notions as special cases. They also show that UCE for computationally unpredictable split sources making one oracle query implies AIPO, which is a special case of our UCE result.

2 Point-function obfuscation framework

The literature considers many different variants of point function obfuscation. Here we provide a definitional framework that unifies these concepts and allows us to obtain not just known but also new variants of point function obfuscation as special cases. The framework parameterizes the security of a point-obfuscator by a class of algorithms we call target generators. Different notions of point obfuscation then correspond to different choices of this class. We start by defining target generators. Please refer to Appendix A for standard notation and definitions.

Target generators. A *target generator* X specifies a PT algorithm $\mathsf{X.Ev}$ that takes 1^λ to return a *target vector* $\mathbf{k}$ and *auxiliary information* $a \in \{0,1\}^*$. The entries of $\mathbf{k}$ are the targets, each of length $\mathsf{X.tl}(\lambda)$, and the vector itself has length $\mathsf{X.vl}(\lambda)$, where $\mathsf{X.tl}, \mathsf{X.vl} \colon \mathbb{N} \to \mathbb{N}$ are the target length and target-vector length functions associated to X, respectively.

Point-function obfuscation. We now define security of point-function obfuscator relative to a class of target generators. We will then consider various choices of these classes.

If k is a bit-string then $\mathbf{I}_k\colon \{0,1\}^{|k|} \to \{0,1\}$ denotes a canonical representation of the circuit that on input $k' \in \{0,1\}^{|k|}$ returns 1 if $k = k'$ and 0 otherwise. It is assumed that given $\mathbf{I}_k$, one can compute k in time linear in $|k|$. A circuit C is called a *point circuit* if there is a k, called the circuit target, such that $\mathrm{C} \equiv \mathbf{I}_k$. If $\mathbf{k}$ is an n-vector of strings then we let $\mathbf{I}_{\mathbf{k}} = (\mathbf{I}_{\mathbf{k}[1]}, \ldots, \mathbf{I}_{\mathbf{k}[n]})$.

Let Obf be an obfuscator, as defined in Appendix A. Its correctness condition guarantees that on input $1^\lambda, \mathbf{I}_k$, it returns a point circuit with target k, which is the condition for calling it a *point-function obfuscator*. We say that Obf has target length $\mathsf{Obf.tl}\colon \mathbb{N} \to \mathbb{N}$ if its correctness condition is only required

Game $\mathrm{PFO}^{\mathcal{A}}_{\mathsf{Obf},\mathsf{X}}(\lambda)$	Game $\mathrm{PRED}^{\mathcal{Q}}_{\mathsf{X}}(\lambda)$	Game $\mathrm{MDIFF}^{\mathcal{D}}_{\mathsf{S}}(\lambda)$	Game $\mathrm{MIO}^{\mathcal{O}}_{\mathsf{Obf},\mathsf{S}}(\lambda)$
$b \leftarrow\$ \{0,1\}$	$(\mathbf{k}, a) \leftarrow\$ \mathsf{X.Ev}(1^\lambda)$	$(\mathbf{C}_0, \mathbf{C}_1, aux) \leftarrow\$ \mathsf{S}(1^\lambda)$	$b \leftarrow\$ \{0,1\}$
$(\mathbf{k}_1, a_1) \leftarrow\$ \mathsf{X.Ev}(1^\lambda)$	$k \leftarrow\$ \mathcal{Q}(1^\lambda, a)$	$x \leftarrow\$ \mathcal{D}(\mathbf{C}_0, \mathbf{C}_1, aux)$	$(\mathbf{C}_0, \mathbf{C}_1, aux) \leftarrow\$ \mathsf{S}(1^\lambda)$
For $i = 1, \ldots, \mathsf{X.vl}(\lambda)$ do	Return $(\exists i : \mathbf{k}[i] = k)$	Return $(\exists i : \mathbf{C}_0[i](x) \neq \mathbf{C}_1[i](x))$	$\overline{\mathbf{C}} \leftarrow\$ \mathsf{Obf}(1^\lambda, \mathbf{C}_b)$
$\quad \mathbf{k}_0[i] \leftarrow\$ \{0,1\}^{\mathsf{X.tl}(\lambda)}$			$b' \leftarrow\$ \mathcal{O}(1^\lambda, \overline{\mathbf{C}}, aux)$
$\overline{\mathbf{P}} \leftarrow\$ \mathsf{Obf}(1^\lambda, \mathbf{I}_{\mathbf{k}_b})$			Return $(b = b')$
$b' \leftarrow\$ \mathcal{A}(1^\lambda, \overline{\mathbf{P}}, a_1)$			
Return $(b = b')$			

Figure 1: Games defining PFO security of obfuscator Obf relative to target generator X, unpredictabilty of target generator X, difference-security of multi-circuit sampler S, and iO-security of obfuscator Obf relative to multi-circuit sampler S.

on inputs $\mathbf{I}_k$ with $k \in \{0,1\}^{\mathsf{Obf.tl}(\lambda)}$.

Consider game PFO of Fig. 1 associated to a point-function obfuscator Obf, a target generator X and an adversary $\mathcal{A}$, such that $\mathsf{Obf.tl} = \mathsf{X.tl}$. For $\lambda \in \mathbb{N}$ let $\mathsf{Adv}^{\mathsf{pfo}}_{\mathsf{Obf},\mathsf{X},\mathcal{A}}(\lambda) = 2\Pr[\mathrm{PFO}^{\mathcal{A}}_{\mathsf{Obf},\mathsf{X}}(\lambda)] - 1$. The game generates a target vector $\mathbf{k}_1$ and corresponding auxiliary information a_1 via X. It also samples a target vector $\mathbf{k}_0$ uniformly at random, containing $\mathsf{X.vl}(\lambda)$ elements each of length $\mathsf{X.tl}(\lambda)$. It then obfuscates the targets in the challenge vector $\mathbf{k}_b$ via Obf to produce $\overline{\mathbf{P}}$ which, as per our notation, will be the vector $(\mathsf{Obf}(1^\lambda, \mathbf{I}_{\mathbf{k}_b[1]}), \ldots, \mathsf{Obf}(1^\lambda, \mathbf{I}_{\mathbf{k}_b[\mathsf{X.vl}(\lambda)]}))$ formed by independently obfuscating the targets in the target vector. Given $\overline{\mathbf{P}}$ and a_1, adversary $\mathcal{A}$ outputs a bit b', and wins the game if this equals b, meaning it guesses whether the target vector that was obfuscated was the one corresponding to auxiliary information a_1 or one independent of it.

Let $\mathbf{X}$ be a class (set) of target generators. We say that Obf is PFO[$\mathbf{X}$]-secure if $\mathsf{Adv}^{\mathsf{pfo}}_{\mathsf{Obf},\mathsf{X},\mathcal{A}}(\cdot)$ is negligible for every PT $\mathcal{A}$ and every $\mathsf{X} \in \mathbf{X}$. Now we can capture different notions in the literature, as well as new ones, by considering particular classes $\mathbf{X}$.

Classes. One important (and necessary) condition on a target generator is unpredictability. To define this, consider game PRED of Fig. 1 associated to X and a predictor adversary $\mathcal{Q}$. For $\lambda \in \mathbb{N}$ let $\mathsf{Adv}^{\mathsf{pred}}_{\mathsf{X},\mathcal{Q}}(\lambda) = \Pr[\mathrm{PRED}^{\mathcal{Q}}_{\mathsf{X}}(\lambda)]$. The game generates a target vector $\mathbf{k}$ and associated auxiliary information a. The adversary $\mathcal{Q}$ gets a and wins if it can predict any entry of the vector $\mathbf{k}$.

The first dimension along which point-function obfuscators are classified is the type of unpredictability, within which there are two sub-dimensions: the success probability of predictors (may be required to be negligible or sub-exponential) and their computational power (PT and computationally unbounded are the popular choices, but one could also consider sub-exponential time). Some relevant classes are the following:

— $\mathbf{X}^{\mathrm{cup}}$ — Class of computationally unpredictable target generators — $\mathsf{X} \in \mathbf{X}^{\mathrm{cup}}$ if $\mathsf{Adv}^{\mathsf{pred}}_{\mathsf{X},\mathcal{Q}}(\cdot)$ is negligible for all PT predictor adversaries $\mathcal{Q}$.

— $\mathbf{X}^{\mathrm{seup}}$ — Class of sub-exponentially unpredictable target generators — $\mathsf{X} \in \mathbf{X}^{\mathrm{seup}}$ if there exists $0 < \epsilon < 1$ such that for every PT predictor adversary $\mathcal{Q}$ there is a $\lambda_{\mathcal{Q}}$ such that $\mathsf{Adv}^{\mathsf{pred}}_{\mathsf{X},\mathcal{Q}}(\lambda) \leq 2^{-\lambda^\epsilon}$ for all $\lambda \geq \lambda_{\mathcal{Q}}$.

— $\mathbf{X}^{\mathrm{sup}}$ — Class of statistically unpredictable target generators — $\mathsf{X} \in \mathbf{X}^{\mathrm{sup}}$ if $\mathsf{Adv}^{\mathsf{pred}}_{\mathsf{X},\mathcal{Q}}(\cdot)$ is negligible for all (even computationally unbounded) predictor adversaries $\mathcal{Q}$.

Another dimension is the number of target points in the target vector, to capture which, for any polynomial $q\colon \mathbb{N} \to \mathbb{N}$, we let

— $\mathbf{X}^{q(\cdot)}$ — Class of generators producing $q(\cdot)$ target points — $\mathsf{X} \in \mathbf{X}^{q(\cdot)}$ if $\mathsf{X.vl} = q$. An important special case is $q(\cdot) = 1$.

Another important dimension is auxiliary information, which may be present or absent (the latter, formally means it is the empty string), to capture which we let

— $\mathbf{X}^{\varepsilon}$ — Class of generators with no auxiliary information — $\mathsf{X} \in \mathbf{X}^{\varepsilon}$ if $a = \varepsilon$ for all $(\mathbf{k}, a) \in [\mathsf{X}.\mathsf{Ev}(1^{\lambda})]$ and all $\lambda \in \mathbb{N}$.

We can recover notions in the literature as follows:

— PFO$[\mathbf{X}^{\mathrm{cup}} \cap \mathbf{X}^{\varepsilon} \cap \mathbf{X}^{1}]$ — This is basic point-function obfuscation, where there is just one target point, no auxiliary information, and unpredictability is only required relative to PT predictors. It is achieved in [22, 26, 35, 41].

— PFO$[\mathbf{X}^{\mathrm{cup}} \cap \mathbf{X}^{1}]$ — This is AIPO [33, 13], where there is just one target point, auxiliary information is present, and unpredictability is only required relative to PT predictors. It is achieved under AI-DHI [22] and by the extended construction of [41] from [13].

— PFO$[\mathbf{X}^{\mathrm{cup}}]$ — This is composable AIPO [23], where there are many target points, auxiliary information is present, and unpredictability is only required relative to PT predictors.

3 (d)iO for multi-circuit samplers

We state and prove a lemma we will use that may be of independent interest. We extend the standard definition of circuit samplers from Appendix A to get *multi-circuit samplers*, which are samplers that may produce a vector of circuit pairs (but still only a single auxiliary information string). We also extend the security definition of differing-inputs obfuscation to work with respect to multi-circuit samplers. We then use a hybrid argument to show that the security of the latter is implied by the standard definition of differing-inputs obfuscation for circuit samplers that produce only a single pair of circuits. This result will be used for our iO-based construction of a point-function obfuscator, BCP [16] being applied to move from diO to iO. (We stress that diO is used as a tool but not as an assumption in our results.)

<u>IO FOR MULTI-CIRCUIT SAMPLERS.</u> A multi-circuit sampler is a PT algorithm S with an associated circuit-vector length function $\mathsf{S}.\mathsf{vl}\colon \mathbb{N} \to \mathbb{N}$. Algorithm S on input 1^{λ} returns a triple $(\mathbf{C}_0, \mathbf{C}_1, aux)$ where aux is a string and $\mathbf{C}_0, \mathbf{C}_1$ are circuit vectors of length $\mathsf{S}.\mathsf{vl}(\lambda)$, such that circuits $\mathbf{C}_0[i]$ and $\mathbf{C}_1[i]$ are of the same size, number of inputs and number of outputs for every $i \in \{1, \ldots, \mathsf{S}.\mathsf{vl}(\lambda)\}$.

Consider game MIO of Fig. 1 associated to an obfuscator Obf, a multi-circuit sampler S and an adversary $\mathcal{O}$. For $\lambda \in \mathbb{N}$ let $\mathsf{Adv}^{\mathsf{m\text{-}io}}_{\mathsf{Obf},\mathsf{S},\mathcal{O}}(\lambda) = 2\Pr[\mathrm{MIO}^{\mathcal{O}}_{\mathsf{Obf},\mathsf{S}}(\lambda)] - 1$. Let $\boldsymbol{S}$ be a class of multi-circuit samplers. We say that Obf is $\boldsymbol{S}$-secure if $\mathsf{Adv}^{\mathsf{m\text{-}io}}_{\mathsf{Obf},\mathsf{S},\mathcal{O}}(\cdot)$ is negligible for every multi-circuit sampler $\mathsf{S} \in \boldsymbol{S}$ and every PT adversary $\mathcal{O}$.

Consider game MDIFF of Fig. 1 associated to a multi-circuit sampler S and an adversary $\mathcal{D}$. For $\lambda \in \mathbb{N}$ let $\mathsf{Adv}^{\mathsf{m\text{-}diff}}_{\mathsf{S},\mathcal{D}}(\lambda) = \Pr[\mathrm{MDIFF}^{\mathcal{D}}_{\mathsf{S}}(\lambda)]$. We say that a multi-circuit sampler S is difference secure if $\mathsf{Adv}^{\mathsf{m\text{-}diff}}_{\mathsf{S},\mathcal{D}}(\cdot)$ is negligible for every PT adversary $\mathcal{D}$. Let $\boldsymbol{S}_{\text{m-diff}}$ be the class of all difference-secure multi-circuit samplers and let $d\colon \mathbb{N} \to \mathbb{N}$. We say that multi-circuit sampler S produces d-differing circuits if circuits $\mathrm{C}_0[i]$ and $\mathrm{C}_1[i]$ differ on at most $d(\lambda)$ inputs with an overwhelming probability over $(\mathbf{C}_0, \mathbf{C}_1, aux) \in [\mathsf{S}(1^{\lambda})]$, for all $\lambda \in \mathbb{N}$ and all $i \in \{1, \ldots, \mathsf{S}.\mathsf{vl}(\lambda)\}$. Let $\boldsymbol{S}_{\text{m-diff}}(d)$ be the class of all difference-secure multi-circuit samplers that produce d-differing circuits.

Lemma 3.1 *Let $d\colon \mathbb{N} \to \mathbb{N}$. Let Obf be an $\boldsymbol{S}_{\text{diff}}(d)$-secure obfuscator. Then Obf is also an $\boldsymbol{S}_{\text{m-diff}}(d)$-secure obfuscator.*

Proof of Theorem 3.1: Let $\mathsf{S}^{\mathsf{m}} \in \boldsymbol{S}_{\text{m-diff}}(d)$ be a multi-circuit sampler. Let $\mathcal{O}^{\mathsf{m}}$ be a PT adversary. Let $\lambda \in \mathbb{N}$ and $n = \mathsf{S}.\mathsf{vl}(\lambda)$. Consider the games G_{ℓ} of Fig. 2 for $\ell \in \{0, \ldots, n\}$. By construction, we have

$$\Pr[\mathrm{G}_0] = \Pr\left[\,\mathrm{MIO}^{\mathcal{O}^{\mathsf{m}}}_{\mathsf{Obf},\mathsf{S}^{\mathsf{m}}}(\lambda) \,|\, b = 1\,\right] \quad \text{and} \quad \Pr[\mathrm{G}_n] = \Pr\left[\,\mathrm{MIO}^{\mathcal{O}^{\mathsf{m}}}_{\mathsf{Obf},\mathsf{S}^{\mathsf{m}}}(\lambda) \,|\, b = 0\,\right] .$$

It follows that $\mathsf{Adv}^{\mathsf{m\text{-}io}}_{\mathsf{Obf},\mathsf{S}^{\mathsf{m}},\mathcal{O}^{\mathsf{m}}}(\lambda) = \Pr[\mathrm{G}_0] - \Pr[\mathrm{G}_n] = \sum_{i=1}^{n}(\Pr[\mathrm{G}_{i-1}] - \Pr[\mathrm{G}_i])$. We will show that this sum is bounded by a negligible function, therefore proving the lemma. We construct a circuit sampler S^{s}

```
Game $\mathrm{G}_\ell$
$b \leftarrow_{\$} \{0,1\}$ ; $(\mathbf{C}_0, \mathbf{C}_1, aux^{\mathsf{m}}) \leftarrow_{\$} \mathsf{S}^{\mathsf{m}}(1^\lambda)$ ; $n \leftarrow \mathsf{S.vl}(\lambda)$
For $i = 1, \ldots, \ell$ do $\overline{\mathbf{C}}[i] \leftarrow_{\$} \mathsf{Obf}(1^\lambda, \mathbf{C}_0[i])$
For $i = \ell+1, \ldots, n$ do $\overline{\mathbf{C}}[i] \leftarrow_{\$} \mathsf{Obf}(1^\lambda, \mathbf{C}_1[i])$
$b' \leftarrow_{\$} \mathcal{O}^{\mathsf{m}}(1^\lambda, \overline{\mathbf{C}}, aux^{\mathsf{m}})$ ; Return $(b = b')$
```

Figure 2: **Games for proof of Lemma 3.1.**

and a PT adversary $\mathcal{O}^{\mathsf{s}}$ as follows:

```
Circuit Sampler $\mathsf{S}^{\mathsf{s}}(1^\lambda)$
$n \leftarrow \mathsf{S.vl}(\lambda)$ ; $\ell \leftarrow_{\$} \{1, \ldots, n\}$ ; $(\mathbf{C}_0, \mathbf{C}_1, aux^{\mathsf{m}}) \leftarrow_{\$} \mathsf{S}^{\mathsf{m}}(1^\lambda)$
For $i = 1, \ldots, \ell - 1$ do $\overline{\mathbf{C}}[i] \leftarrow_{\$} \mathsf{Obf}(1^\lambda, \mathbf{C}_0[i])$
For $i = \ell+1, \ldots, n$ do $\overline{\mathbf{C}}[i] \leftarrow_{\$} \mathsf{Obf}(1^\lambda, \mathbf{C}_1[i])$
$e \leftarrow_{\$} \{0,1\}$ ; $\overline{\mathbf{C}}_l \leftarrow \overline{\mathbf{C}}[1], \ldots, \overline{\mathbf{C}}[\ell-1]$ ; $\overline{\mathbf{C}}_r \leftarrow \overline{\mathbf{C}}[\ell+1], \ldots, \overline{\mathbf{C}}[n]$
$aux^{\mathsf{s}} \leftarrow (\overline{\mathbf{C}}_l, \overline{\mathbf{C}}_r, e, aux^{\mathsf{m}})$ ; Return $(\mathbf{C}_0[\ell], \mathbf{C}_1[\ell], aux^{\mathsf{s}})$
```

```
Adversary $\mathcal{O}^{\mathsf{s}}(1^\lambda, \overline{\mathrm{C}}, aux^{\mathsf{s}})$
$(\overline{\mathbf{C}}_l, \overline{\mathbf{C}}_r, e, aux^{\mathsf{m}}) \leftarrow aux^{\mathsf{s}}$
$\overline{\mathbf{C}} \leftarrow (\overline{\mathbf{C}}_l, \overline{\mathrm{C}}, \overline{\mathbf{C}}_r)$
$e' \leftarrow_{\$} \mathcal{O}^{\mathsf{m}}(1^\lambda, \overline{\mathbf{C}}, aux^{\mathsf{m}})$
If $(e = e')$ then return 1
Else return 0
```

Let ℓ be the value sampled by S^{s} in game $\mathrm{IO}^{\mathcal{O}^{\mathsf{s}}}_{\mathsf{Obf},\mathsf{S}^{\mathsf{s}}}(\lambda)$. For any $i \in \{1, \ldots, n\}$ we have $\Pr[\mathrm{G}_{i-1}] - \Pr[\mathrm{G}_i] = \Pr\left[\, \mathsf{Adv}^{\mathsf{io}}_{\mathsf{Obf},\mathsf{S}^{\mathsf{s}},\mathcal{O}^{\mathsf{s}}}(\lambda) \;:\; \ell = i \,\right]$, and hence $\sum_{i=1}^{n}(\Pr[\mathrm{G}_{i-1}] - \Pr[\mathrm{G}_i]) = n \cdot \mathsf{Adv}^{\mathsf{io}}_{\mathsf{Obf},\mathsf{S}^{\mathsf{s}},\mathcal{O}^{\mathsf{s}}}(\lambda)$. We will now prove that $\mathsf{Adv}^{\mathsf{io}}_{\mathsf{Obf},\mathsf{S}^{\mathsf{s}},\mathcal{O}^{\mathsf{s}}}(\lambda)$ is negligible by showing that $\mathsf{S}^{\mathsf{s}} \in \boldsymbol{S}_{\mathrm{diff}}(d)$. Since $n = \mathsf{S.vl}(\lambda)$ is a polynomial, it will follow that the above sum is negligible. Given a PT adversary $\mathcal{D}^{\mathsf{s}}$ we construct a PT adversary $\mathcal{D}^{\mathsf{m}}$ such that $\mathsf{Adv}^{\mathsf{m\text{-}diff}}_{\mathsf{S}^{\mathsf{m}},\mathcal{D}^{\mathsf{m}}}(\lambda) \geq \mathsf{Adv}^{\mathsf{diff}}_{\mathsf{S}^{\mathsf{s}},\mathcal{D}^{\mathsf{s}}}(\lambda)$.

```
Adversary $\mathcal{D}^{\mathsf{m}}(1^\lambda, \mathbf{C}_0, \mathbf{C}_1, aux^{\mathsf{m}})$
$n \leftarrow \mathsf{S.vl}(\lambda)$ ; $\ell \leftarrow_{\$} \{1, \ldots, n\}$
For $i = 1, \ldots, \ell - 1$ do $\overline{\mathbf{C}}[i] \leftarrow_{\$} \mathsf{Obf}(1^\lambda, \mathbf{C}_0[i])$
For $i = \ell+1, \ldots, n$ do $\overline{\mathbf{C}}[i] \leftarrow_{\$} \mathsf{Obf}(1^\lambda, \mathbf{C}_1[i])$
$e \leftarrow_{\$} \{0,1\}$ ; $\overline{\mathbf{C}}_l \leftarrow \overline{\mathbf{C}}[1], \ldots, \overline{\mathbf{C}}[\ell-1]$ ; $\overline{\mathbf{C}}_r \leftarrow \overline{\mathbf{C}}[\ell+1], \ldots, \overline{\mathbf{C}}[n]$
$aux^{\mathsf{s}} \leftarrow (\overline{\mathbf{C}}_l, \overline{\mathbf{C}}_r, e, aux^{\mathsf{m}})$ ; $x \leftarrow_{\$} \mathcal{D}^{\mathsf{s}}(\mathbf{C}_0[\ell], \mathbf{C}_1[\ell], aux^{\mathsf{s}})$ ; Return $x$
```

Now, $\mathsf{S}^{\mathsf{s}} \in \boldsymbol{S}_{\mathrm{diff}}(d)$ follows from the assumption that $\mathsf{S}^{\mathsf{m}} \in \boldsymbol{S}_{\mathrm{m\text{-}diff}}(d)$, which concludes the proof. ▌

4 Generic constructions of PFO

Prior constructions have targeted PFO[$\mathbf{X}$] for specific choices of $\mathbf{X}$ in ad hoc ways and used strong assumptions. In this section we provide constructions that are generic. This means they take an arbitrary, given class $\mathbf{X}$ of target generators and return a point-function obfuscator that is PFO[$\mathbf{X}$]-secure.

4.1 PFO from iO

OWFs. Consider game OWF of Fig. 3 associated to a function family F, a target generator X with $\mathsf{X.tl} = \mathsf{F.il}$, and an adversary $\mathcal{F}$. For $\lambda \in \mathbb{N}$ let $\mathsf{Adv}^{\mathsf{owf}}_{\mathsf{F},\mathsf{X},\mathcal{F}}(\lambda) = \Pr[\mathrm{OWF}^{\mathcal{F}}_{\mathsf{F},\mathsf{X}}(\lambda)]$. Let $\mathbf{X}$ be a class of target generators with target length $\mathsf{F.il}$. Let $\mathsf{X}^{\mathsf{1ur}}$ be the target generator with $\mathsf{X}^{\mathsf{1ur}}.\mathsf{vl}(\cdot) = 1$ and $\mathsf{X}^{\mathsf{1ur}}.\mathsf{tl} = \mathsf{F.il}$, where the target is sampled from a uniform distribution and the auxiliary information is always empty, meaning $a = \varepsilon$. We say that F is OWF[$\mathbf{X}$]-secure if $\mathsf{Adv}^{\mathsf{owf}}_{\mathsf{F},\mathsf{X},\mathcal{F}}(\cdot)$ is negligible for all PT adversaries $\mathcal{F}$ and all $\mathsf{X} \in \mathbf{X} \cup \{\mathsf{X}^{\mathsf{1ur}}\}$. Relevant classes $\mathbf{X}$ are the same as for PFO. The standard notion of a OWF is recovered as $\mathbf{X} = \emptyset$, meaning that F is secure only with respect to $\mathsf{X}^{\mathsf{1ur}}$.

```
Game $\mathrm{OWF}^{\mathcal{F}}_{\mathsf{F},\mathsf{X}}(\lambda)$
$(\mathbf{k}, a) \xleftarrow{\$} \mathsf{X.Ev}(1^\lambda)$
For $i = 1, \ldots, \mathsf{X.vl}(\lambda)$ do
    $\mathbf{fk}[i] \xleftarrow{\$} \mathsf{F.Kg}(1^\lambda)$
    $\mathbf{y}[i] \leftarrow \mathsf{F.Ev}(1^\lambda, \mathbf{fk}[i], \mathbf{k}[i])$
$k \xleftarrow{\$} \mathcal{F}(1^\lambda, \mathbf{fk}, \mathbf{y}, a)$
Return $(\exists i : \mathsf{F.Ev}(1^\lambda, \mathbf{fk}[i], k) = \mathbf{y}[i])$
```

```
Game $\mathrm{PRIV1}^{\mathcal{A}}_{\mathsf{DPKE},\mathsf{X}}(\lambda)$
$b \xleftarrow{\$} \{0,1\}$ ; $(\mathbf{k}_1, a) \xleftarrow{\$} \mathsf{X.Ev}(1^\lambda)$
For $i = 1, \ldots, \mathsf{X.vl}(\lambda)$ do
    $\mathbf{k}_0[i] \xleftarrow{\$} \{0,1\}^{\mathsf{DPKE.ml}(\lambda)}$
    $(\mathbf{pk}[i], \mathbf{sk}[i]) \xleftarrow{\$} \mathsf{DPKE.Kg}(1^\lambda)$
    $\mathbf{c}[i] \leftarrow \mathsf{DPKE.Enc}(1^\lambda, \mathbf{pk}[i], \mathbf{k}_b[i])$
$b' \xleftarrow{\$} \mathcal{A}(1^\lambda, \mathbf{pk}, \mathbf{c}, a)$ ; Return $(b = b')$
```

Figure 3: Games defining one-wayness of function family F relative to target generator X and PRIV1-security of deterministic public-key encryption scheme DPKE relative to target generator X.

```
Games $\mathrm{G}_0$, $\mathrm{G}_1$
$b \xleftarrow{\$} \{0,1\}$ ; $(\mathbf{k}_1, a_1) \xleftarrow{\$} \mathsf{X.Ev}(1^\lambda)$
For $i = 1, \ldots, \mathsf{X.vl}(\lambda)$ do
    $\mathbf{k}_0[i] \xleftarrow{\$} \{0,1\}^{\mathsf{X.tl}(\lambda)}$ ; $\mathbf{fk}[i] \xleftarrow{\$} \mathsf{F.Kg}(1^\lambda)$ ; $\mathbf{y}[i] \leftarrow \mathsf{F.Ev}(1^\lambda, \mathbf{fk}[i], \mathbf{k}_b[i])$
    $\overline{\mathbf{P}}[i] \xleftarrow{\$} \mathsf{Obf_{io}}(\mathrm{C}^1_{1^\lambda,\mathbf{fk}[i],\mathbf{y}[i]})$            // $\mathrm{G}_0$
    $\overline{\mathbf{P}}[i] \xleftarrow{\$} \mathsf{Obf_{io}}(\mathsf{Pad}_{s(\lambda)}(\mathrm{C}^2))$            // $\mathrm{G}_1$
$b' \xleftarrow{\$} \mathcal{A}(1^\lambda, \overline{\mathbf{P}}, a_1)$ ; Return $(b = b')$

Circuit $\mathrm{C}^1_{1^\lambda,fk,y}(k)$
If $(y = \mathsf{F.Ev}(1^\lambda, fk, k))$ then return 1
Else return 0

Circuit $\mathrm{C}^2(k)$
Return 0
```

Figure 4: **Games for proof of Theorem 4.1.**

The definition of CD [23] is the special case of ours with vectors of length one. That of FOR [30], like ours, considers evaluations of the function on multiple inputs, but in their case the key for the evaluations is the same and there is no auxiliary input, while in our case the key is independently chosen for each evaluation and auxiliary inputs may be present. We stress that we require only one-wayness; we do *not* require extractability. The latter is a much stronger assumption [12].

We now show that indistinguishability obfuscation can be used to build a PFO[$\mathbf{X}$]-secure point-function obfuscator for an arbitrary target generator class $\mathbf{X}$ from any OWF[$\mathbf{X}$]-secure function family.

CONSTRUCTION. Let F be a family of functions. Let $\mathsf{Obf_{io}}$ be an obfuscator. We construct a point-function obfuscator Obf with $\mathsf{Obf.tl} = \mathsf{F.il}$ as follows:

```
Algorithm $\mathsf{Obf}(1^\lambda, \mathbf{I}_k)$
$fk \xleftarrow{\$} \mathsf{F.Kg}(1^\lambda)$ ; $y \leftarrow \mathsf{F.Ev}(1^\lambda, fk, k)$
$\overline{\mathrm{P}} \xleftarrow{\$} \mathsf{Obf_{io}}(\mathrm{C}_{1^\lambda,fk,y})$ ; Return $\overline{\mathrm{P}}$

Circuit $\mathrm{C}_{1^\lambda,fk,y}(k')$
If $(y = \mathsf{F.Ev}(1^\lambda, fk, k'))$ then return 1
Else return 0
```

Theorem 4.1 *Let* F *be an injective family of functions. Let* $\mathbf{X}$ *be a class of target generators with target length* $\mathsf{F.il}$*. Assume that* F *is* OWF[$\mathbf{X}$]*-secure. Let* $\mathsf{Obf_{io}}$ *be an indistinguishability obfuscator. Then* Obf *constructed above from* F *and* $\mathsf{Obf_{io}}$ *is a* PFO[$\mathbf{X}$]*-secure point-function obfuscator.*

Proof of Theorem 4.1: The injectivity of F implies that Obf satisfies the correctness condition of a point-function obfuscator. We now prove security.

Let $\mathsf{X} \in \mathbf{X}$ be a target generator. Let $\mathcal{A}$ be a PT adversary. Consider the games and the associated circuits of Fig. 4, where s is defined as follows. For any λ let $s(\lambda)$ be a polynomial upper bound on $\max(|\mathrm{C}^1_{1^\lambda,fk,y}|)$, where the maximum is over all $fk \in [\mathsf{F.Kg}(1^\lambda)]$ and $y \in \{0,1\}^{\mathsf{F.ol}(\lambda)}$. Lines not annotated with comments are common to all games.

Game G_0 is equivalent to $\mathrm{PFO}^{\mathcal{A}}_{\mathsf{Obf},\mathsf{X}}(\lambda)$. The inputs to adversary $\mathcal{A}$ in game G_1 do not depend on the challenge bit b, so we have $\Pr[\mathrm{G}_1] = 1/2$. It follows that

$$\mathsf{Adv}^{\mathsf{pfo}}_{\mathsf{Obf},\mathsf{X},\mathcal{A}}(\lambda) = 2 \cdot (\Pr[\mathrm{G}_1] + \Pr[\mathrm{G}_0] - \Pr[\mathrm{G}_1]) - 1 = 2 \cdot (\Pr[\mathrm{G}_0] - \Pr[\mathrm{G}_1]).$$

We now show that $\Pr[\mathrm{G}_0] - \Pr[\mathrm{G}_1]$ is negligible, meaning that $\mathsf{Adv}^{\mathsf{pfo}}_{\mathsf{Obf},\mathsf{X},\mathcal{A}}(\cdot)$ is also negligible. This proves the the theorem.

We construct a multi-circuit sampler S and a PT iO-adversary $\mathcal{O}$ as follows:

```
Multi-circuit Sampler $\mathsf{S}(1^\lambda)$
$d \leftarrow_{\$} \{0,1\}$ ; $(\mathbf{k}_1, a_1) \leftarrow_{\$} \mathsf{X.Ev}(1^\lambda)$
For $i = 1, \ldots, \mathsf{X.vl}(\lambda)$ do
    $\mathbf{k}_0[i] \leftarrow_{\$} \{0,1\}^{\mathsf{X.tl}(\lambda)}$
    $\mathbf{fk}[i] \leftarrow_{\$} \mathsf{F.Kg}(1^\lambda)$ ; $\mathbf{y}[i] \leftarrow \mathsf{F.Ev}(1^\lambda, \mathbf{fk}[i], \mathbf{k}_d[i])$
    $\mathbf{C}_1[i] \leftarrow \mathrm{C}^1_{1^\lambda,\mathbf{fk}[i],\mathbf{y}[i]}$ ; $\mathbf{C}_0[i] \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathrm{C}^2)$
$aux \leftarrow (d, a_1)$ ; Return $(\mathbf{C}_0, \mathbf{C}_1, aux)$
```

```
Adversary $\mathcal{O}(1^\lambda, \overline{\mathbf{C}}, aux)$
$(d, a_1) \leftarrow aux$
$d' \leftarrow_{\$} \mathcal{A}(1^\lambda, \overline{\mathbf{C}}, a_1)$
If $(d = d')$ then return 1
Else return 0
```

We have $\Pr[\mathrm{G}_0] - \Pr[\mathrm{G}_1] = \mathsf{Adv}^{\mathsf{m\text{-}io}}_{\mathsf{Obf}_{\mathsf{io}},\mathsf{S},\mathcal{O}}(\lambda)$ by construction. Next, we show that $\mathsf{S} \in \boldsymbol{S}_{\text{m-diff}}(1)$. According to Proposition A.1 (the result of BCP [16]), any indistinguishability obfuscator is also an $\boldsymbol{S}_{\text{diff}}(1)$-secure obfuscator. And according to Lemma 3.1, any $\boldsymbol{S}_{\text{diff}}(1)$-secure obfuscator is an $\boldsymbol{S}_{\text{m-diff}}(1)$-secure obfuscator. It follows that $\mathsf{Adv}^{\mathsf{m\text{-}io}}_{\mathsf{Obf}_{\mathsf{io}},\mathsf{S},\mathcal{O}}(\cdot)$ is negligible by the iO-security of $\mathsf{Obf}_{\mathsf{io}}$.

Let X^{ur} be the target generator with $\mathsf{X}^{\mathsf{ur}}.\mathsf{vl} = \mathsf{X.vl}$ and $\mathsf{X}^{\mathsf{ur}}.\mathsf{tl} = \mathsf{F.il}$, where the targets are sampled independently, from a uniform distribution and auxiliary information is always $a = \varepsilon$. Given any PT difference adversary $\mathcal{D}$ against multi-circuit sampler S, we build PT adversaries $\mathcal{F}_0$ and $\mathcal{F}_1$ against the OWF-security of F relative to target generators X^{ur} and X, respectively. The constructions are as follows:

```
Adversary $\mathcal{F}_0(1^\lambda, \mathbf{fk}, \mathbf{y}, a)$
$d \leftarrow 0$ ; $(\mathbf{k}_1, a_1) \leftarrow_{\$} \mathsf{X.Ev}(1^\lambda)$
For $i = 1, \ldots, |\mathbf{y}|$ do
    $\mathbf{C}_1[i] \leftarrow \mathrm{C}^1_{1^\lambda,\mathbf{fk}[i],\mathbf{y}[i]}$ ; $\mathbf{C}_0[i] \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathrm{C}^2)$
$aux \leftarrow (d, a_1)$ ; $x \leftarrow_{\$} \mathcal{D}(\mathbf{C}_1, \mathbf{C}_0, aux)$ ; Return $x$
```

```
Adversary $\mathcal{F}_1(1^\lambda, \mathbf{fk}, \mathbf{y}, a)$
$d \leftarrow 1$
For $i = 1, \ldots, |\mathbf{y}|$ do
    $\mathbf{C}_1[i] \leftarrow \mathrm{C}^1_{1^\lambda,\mathbf{fk}[i],\mathbf{y}[i]}$ ; $\mathbf{C}_0[i] \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathrm{C}^2)$
$aux \leftarrow (d, a)$ ; $x \leftarrow_{\$} \mathcal{D}(\mathbf{C}_1, \mathbf{C}_0, aux)$ ; Return $x$
```

Let d denote the value sampled by multi-circuit sampler S in game $\mathrm{MDIFF}^{\mathcal{D}}_{\mathsf{S}}(\lambda)$. Then we have

$$\Pr[\,\mathrm{MDIFF}^{\mathcal{D}}_{\mathsf{S}}(\lambda) \,|\, d = 0\,] = \Pr[\mathrm{OWF}^{\mathcal{F}_0}_{\mathsf{F},\mathsf{X}^{\mathsf{ur}}}(\lambda)] \quad \text{and} \quad \Pr[\,\mathrm{MDIFF}^{\mathcal{D}}_{\mathsf{S}}(\lambda) \,|\, d = 1\,] = \Pr[\mathrm{OWF}^{\mathcal{F}_1}_{\mathsf{F},\mathsf{X}}(\lambda)]\,,$$

and $\mathsf{Adv}^{\mathsf{m\text{-}diff}}_{\mathsf{S},\mathcal{D}}(\lambda) = \frac{1}{2}(\mathsf{Adv}^{\mathsf{owf}}_{\mathsf{F},\mathsf{X}^{\mathsf{ur}},\mathcal{F}_0}(\lambda) + \mathsf{Adv}^{\mathsf{owf}}_{\mathsf{F},\mathsf{X},\mathcal{F}_1}(\lambda))$. We note that $\mathsf{Adv}^{\mathsf{owf}}_{\mathsf{F},\mathsf{X}^{\mathsf{ur}},\mathcal{F}_0}(\lambda)$ is negligible according to the OWF[$\mathbf{X}$]-security of F, part of which requires that $\mathsf{Adv}^{\mathsf{owf}}_{\mathsf{F},\mathsf{X}^{\mathsf{1ur}},\mathcal{F}}(\lambda)$ is negligible for all PT adversaries $\mathcal{F}$. The former can be proved using a standard hybrid argument based on the latter. It follows that the multi-circuit sampler S is difference-secure. The injectivity of F also implies that S produces 1-differing circuits. Therefore, $\mathsf{S} \in \boldsymbol{S}_{\text{m-diff}}(1)$. ▮

4.2 PFO from DPKE

Our next generic construction is based on deterministic public-key encryption [3]. As before we aim to provide point-function obfuscation secure for any given class of target generators. We are able to do this assuming the existence of a deterministic public-key encryption scheme that is secure relative to the same class viewed as a class of message generators. We can then exploit known constructions of deterministic public-key encryption to get a slew of point-function obfuscators based on standard assumptions. We begin with a parameterized definition of security for deterministic public-key encryption.

DPKE. A deterministic public-key encryption scheme DPKE [3] specifies the following. PT key generation algorithm $\mathsf{DPKE.Kg}$ takes 1^λ to return a public encryption key pk and a secret decryption key sk. Deterministic PT encryption algorithm $\mathsf{DPKE.Enc}$ takes 1^λ, pk and a plaintext message $k \in \{0,1\}^{\mathsf{DPKE.ml}(\lambda)}$ to return a ciphertext c, where $\mathsf{DPKE.ml}\colon \mathbb{N} \to \mathbb{N}$ is the message length function associated to DPKE. Deterministic decryption algorithm $\mathsf{DPKE.Dec}$ takes 1^λ, sk, c to return plaintext message k. We do not require the decryption algorithm to be PT but we do require decryption correctness, namely that for all $\lambda \in \mathbb{N}$, all $(pk, sk) \in [\mathsf{DPKE.Kg}(1^\lambda)]$ and all $k \in \{0,1\}^{\mathsf{DPKE.ml}(\lambda)}$ we have $\mathsf{DPKE.Dec}(1^\lambda, sk, \mathsf{DPKE.Enc}(1^\lambda, pk, k)) = k$.

Now consider game PRIV1 of Fig. 3 associated to a deterministic public-key encryption scheme DPKE, a target generator X satisfying $\mathsf{X.tl} = \mathsf{DPKE.ml}$, and an adversary $\mathcal{A}$. For $\lambda \in \mathbb{N}$ let $\mathsf{Adv}^{\mathsf{priv1}}_{\mathsf{DPKE},\mathsf{X},\mathcal{A}}(\lambda) = 2\Pr[\mathrm{PRIV1}^{\mathcal{A}}_{\mathsf{DPKE},\mathsf{X}}(\lambda)] - 1$. If $\mathbf{X}$ is a class of target generators then we say that DPKE is PRIV1[$\mathbf{X}$]-secure if $\mathsf{Adv}^{\mathsf{priv1}}_{\mathsf{DPKE},\mathsf{X},\mathcal{A}}(\cdot)$ is negligible for all PT adversaries $\mathcal{A}$ and all $\mathsf{X} \in \mathbf{X}$.

This definition reflects what BBO [3] call the multi-user setting where there are many, independent public keys. However, in our case, only a single message is encrypted under each key. The single-key version of this is called PRIV1 in the literature, so we retained the name in moving to the multi-user setting. The definition is in the indistinguishability style of [4, 15] rather than the semantic security style of [3]. These definitions however did not allow auxiliary inputs. We are allowing those following BS [17]. Finally, while prior definitions require unpredictability of the message distribution, ours is simply parameterized by the latter. Prior definitions are captured as special cases, meaning they can be recovered as PRIV1[$\mathbf{X}$] for some choice of $\mathbf{X}$.

Construction. Let DPKE be a deterministic public-key encryption scheme. We construct an obfuscator Obf with $\mathsf{Obf.tl} = \mathsf{DPKE.ml}$ as follows:

```
Algorithm $\mathsf{Obf}(1^\lambda, \mathbf{I}_k)$
$(pk, sk) \leftarrow\$ \mathsf{DPKE.Kg}(1^\lambda)$
$c \leftarrow \mathsf{DPKE.Enc}(1^\lambda, pk, k)$ ; Return $\mathrm{C}_{1^\lambda, pk, c}$

Circuit $\mathrm{C}_{1^\lambda, pk, c}(k)$
If $(\mathsf{DPKE.Enc}(1^\lambda, pk, k) = c)$
Then return 1 else return 0
```

The construction is simple. To obfuscate $\mathbf{I}_k$ we pick a new key pair for the deterministic public-key encryption scheme and return a circuit that embeds the public key pk as well as the encryption c of the target point k. The circuit, given a candidate target point k', re-encrypts it under the embedded public key pk and checks that the ciphertext so obtained matches the embedded ciphertext c. Note that the determinism of $\mathsf{DPKE.Enc}$ is used crucially to ensure that the circuit is deterministic. For randomized encryption, one cannot check that a message corresponds to a ciphertext by re-encryption. The secret key sk is discarded and not used in the construction, but its existence will guarantee correctness of the point-function obfuscator.

Result. We show that this is a generic construction. Namely, a point-function obfuscator for a given class $\mathbf{X}$ of target generators can be obtained if we have a deterministic public-key encryption scheme secure for the same class.

Theorem 4.2 *Let* DPKE *be a deterministic public-key encryption scheme and* $\mathbf{X}$ *a class of target generators such that* $\mathsf{X.tl} = \mathsf{DPKE.ml}$ *for all* $\mathsf{X} \in \mathbf{X}$. *Assume* DPKE *is* PRIV1[$\mathbf{X}$]*-secure. Let* Obf *be as defined above. Then* Obf *is a* PFO[$\mathbf{X}$]*-secure point-function obfuscator.*

Proof of Theorem 4.2: The correctness of Obf follows from the decryption correctness of DPKE, and it does not require the decryption algorithm $\mathsf{DPKE.Dec}$ to be PT. We now prove that Obf is PFO[$\mathbf{X}$]-secure.

Let $\mathsf{X} \in \mathbf{X}$ be a target generator with $\mathsf{X.tl} = \mathsf{DPKE.ml}$. Let $\mathcal{A}$ be PT adversary against the PFO security of Obf relative to X. We construct a PT adversary $\mathcal{B}$ against the PRIV1 security of DPKE relative to X as follows:

```
Adversary $\mathcal{B}(1^\lambda, \mathbf{pk}, \mathbf{c}, a)$
For $i = 1, \ldots, |\mathbf{c}|$ do $\overline{\mathbf{P}}[i] \leftarrow \mathrm{C}_{1^\lambda, \mathbf{pk}[i], \mathbf{c}[i]}$
$b' \leftarrow\!\!\$\; \mathcal{A}(1^\lambda, \overline{\mathbf{P}}, a)$ ; Return $b'$
```

```
Circuit $\mathrm{C}_{1^\lambda, pk, c}(k)$
If $(\mathsf{DPKE.Enc}(1^\lambda, pk, k) = c)$
Then return 1 else return 0
```

We have $\mathsf{Adv}^{\mathsf{priv1}}_{\mathsf{DPKE},\mathsf{X},\mathcal{B}}(\lambda) = \mathsf{Adv}^{\mathsf{pfo}}_{\mathsf{Obf},\mathsf{X},\mathcal{A}}(\lambda)$ by construction. Hence, for any $\mathsf{X} \in \mathbf{X}$ the PFO-security of Obf relative to X follows from the assumed PRIV1-security of DPKE relative to X. ▌

In applying Theorem 4.2 to get point function obfuscators, the first case of interest is $\mathbf{X} = \mathbf{X}^{\mathrm{sup}} \cap \mathbf{X}^{\varepsilon} \cap \mathbf{X}^{1}$. In this case, PRIV1[$\mathbf{X}$]-secure deterministic public-key encryption is a standard form of the latter for which many constructions are known. The central construction, due to BFO [15], is from lossy trapdoor functions (LTDFs). But the latter can be built from a wide variety of standard assumptions [39, 29, 42, 34, 45]. Thus we get PFO[$\mathbf{X}^{\mathrm{sup}} \cap \mathbf{X}^{\varepsilon} \cap \mathbf{X}^{1}$]-secure point-function obfuscators under the same assumptions. The second case of interest is $\mathbf{X} = \mathbf{X}^{\mathrm{seup}} \cap \mathbf{X}^{1}$. Unlike in the first case, there is now auxiliary information, but it leaves the targets sub-exponentially unpredictable. Constructions of PRIV1[$\mathbf{X}$]-secure deterministic public-key encryption are known under standard assumptions including DLIN, Subgroup Indistinguishability and LWE [17, 44, 42]. Accordingly we get PFO[$\mathbf{X}^{\mathrm{seup}} \cap \mathbf{X}^{1}$]-secure point-function obfuscators under the same assumptions.

Theorem 4.2 also yields negative results. Assume iO exists. Then we know that there do not exist point function obfuscators that are PFO[$\mathbf{X}^{\mathrm{cup}}$]-secure [19]. Theorem 4.2 then implies that there also do not exist deterministic public-key encryption schemes that are PRIV1[$\mathbf{X}^{\mathrm{cup}}$]-secure.

4.3 PFO from UCE

Our next generic construction is based on UCE, a class of assumptions on function families from [6]. The definitions are recalled in Appendix A. As before we aim to provide point-function obfuscation secure for any given class of target generators. We are able to do this with UCE by associating to the class of target generators a class of sources such that the existence of a UCE-secure function family relative to the latter suffices to construct a point-function obfuscator secure relative to the former.

Construction. Let H be a family of functions as per Appendix A. Associate to it a point-function obfuscator Obf defined as follows. Let $\mathsf{Obf.tl} = \mathsf{H.il}$, and

```
Algorithm $\mathsf{Obf}(1^\lambda, \mathbf{I}_k)$
$hk \leftarrow\!\!\$\; \mathsf{H.Kg}(1^\lambda)$ ; $y \leftarrow \mathsf{H.Ev}(1^\lambda, hk, k)$
Return $\mathrm{C}_{1^\lambda, hk, y}$
```

```
Circuit $\mathrm{C}_{1^\lambda, hk, y}(k')$
$y' \leftarrow \mathsf{H.Ev}(1^\lambda, hk, k')$
If $(y = y')$ then return 1 else return 0
```

The construction is simple and natural. The point-function obfuscation of $\mathbf{I}_k$ is a circuit that embeds the hash y of k under a freshly-chosen key hk also embedded in the circuit, and, given a candidate target k', checks whether its hash under hk equals the embedded hash value.

Source classes. To state the result, we need a few definitions. Associate to a target generator X a source $\mathcal{S}^{\mathsf{X}}$ defined as follows:

```
Source $\mathcal{S}^{\mathsf{X}}(1^\lambda)$
$(\mathbf{k}, a) \leftarrow\!\!\$\; \mathsf{X.Ev}(1^\lambda)$
For $i = 1, \ldots, |\mathbf{k}|$ do $\mathbf{y}[i] \leftarrow\!\!\$\; \mathrm{HASH}(i, \mathbf{k}[i])$
$L \leftarrow (\mathbf{y}, a)$ ; Return $L$
```

The number of keys for this source is $\mathcal{S}^{\mathsf{X}}.\mathsf{nk} = \mathsf{X.vl}$, the number of points in the target vector. Now let $\mathbf{X}$ be a class of target generators and let $\mathbf{S}^{\mathbf{X}} = \{\, \mathcal{S}^{\mathsf{X}} \,:\, \mathsf{X} \in \mathbf{X} \,\}$ be the corresponding class of sources. We will show that the construction above is PFO[$\mathbf{X}$]-secure assuming H is UCE[$\mathbf{S}^{\mathbf{X}}$]-secure. To appreciate what this provides we now discuss the assumption further.

UCE security is very sensitive to the class of sources for which security is assumed. Accordingly one tries to restrict sources in different ways. In this regard $\mathbf{S}^{\mathbf{X}} = \{\, \mathcal{S}^{\mathsf{X}} : \mathsf{X} \in \mathbf{X} \,\}$ has some good attributes as we now discuss, referring to definitions of classes of UCE sources recalled in Appendix A.

The first attribute is that the sources in $\mathbf{S}^{\mathbf{X}}$ are what BHK [6] call "split," so that $\mathbf{S}^{\mathbf{X}} \subseteq \mathbf{S}^{\mathrm{splt}}$. "Split" means that the leakage is a function of the oracle queries and answers separately, but not both together. (Above, a depends only on the oracle queries, and $\mathbf{y}$ depends only on the answers.) The second attribute is that the sources make only one query per key. (In particular when there is only one target point, the source makes only one query overall.) That is, $\mathbf{S}^{\mathbf{X}} \subseteq \mathbf{S}^{n,1}$ if $\mathcal{S}.\mathsf{nk}(\cdot) \leq n(\cdot)$ for all $\mathcal{S} \in \mathbf{S}^{\mathbf{X}}$. The third attribute is that the source class inherits the unpredictability properties of the target generator class. Thus if $\mathbf{X} \subseteq \mathbf{X}^{\mathrm{cup}}$ then $\mathbf{S}^{\mathbf{X}} \subseteq \mathbf{S}^{\mathrm{cup}}$ consists of computationally unpredictable sources; if $\mathbf{X} \subseteq \mathbf{X}^{\mathrm{sup}}$ then $\mathbf{S}^{\mathbf{X}} \subseteq \mathbf{S}^{\mathrm{sup}}$ consists of statistically unpredictable sources; and if $\mathbf{X} \subseteq \mathbf{X}^{\mathrm{seup}}$ then $\mathbf{S}^{\mathbf{X}} \subseteq \mathbf{S}^{\mathrm{seup}}$ consists of sources that are sub-exponentially unpredictable.

We warn that UCE[$\mathbf{S}^{\mathbf{X}}$]-security is not achievable for all choices of $\mathbf{X}$. The value of our result is that it is entirely general, reducing PFO security for a given $\mathbf{X}$ to a question of UCE security for a related class of sources, and we can then investigate the latter separately. In this way we get many new constructions.

Result. The following theorem shows that our construction above provides secure point-function obfuscation in a very general and modular way, namely the point-function obfuscator is secure relative to a class of target generators if H is UCE-secure relative to the corresponding class of sources. After stating and proving this general result we will look at some special cases of interest.

Theorem 4.3 *Let* H *be an injective family of functions and* $\mathbf{X}$ *a class of target generators such that* $\mathsf{X.tl} = \mathsf{H.il}$ *for all* $\mathsf{X} \in \mathbf{X}$. *Assume* H *is* UCE[$\mathbf{S}^{\mathbf{X}}$]*-secure. Let* Obf *be as defined above. Then* Obf *is a* PFO[$\mathbf{X}$]*-secure point-function obfuscator.*

The injectivity of H is assumed in order to meet the correctness condition of a point-function obfuscator. It is not important for security. We note that the perfect correctness we have required for point-function obfuscators can be relaxed to a computational correctness requirement, namely that given an obfuscation $\overline{\mathrm{P}}$ of a point function $\mathbf{I}_k$, no PT adversary can find $k' \neq k$ such that $\overline{\mathrm{P}}(k') = 1$ holds with better than a negligible probability. The relaxed form of correctness can be achieved assuming nothing but UCE for the corresponding class of sources, meaning the injectivity requirement can be dropped.

Proof of Theorem 4.3: Correctness of the obfuscator follows from the assumed injectivity of H, meaning that the output of $\mathsf{Obf}(1^\lambda, \mathbf{I}_k)$ is always a point circuit with target k. We now prove that Obf is PFO[$\mathbf{X}$]-secure.

Let $\mathsf{X} \in \mathbf{X}$ be a target generator with $\mathsf{X.tl} = \mathsf{H.il}$. Let $\mathcal{S}^{\mathsf{X}}$ be the source as defined above. Let $\mathcal{A}$ be a PT adversary against the PFO-security of Obf relative to X and define PT distinguisher $\mathcal{D}$ via

```
Distinguisher $\mathcal{D}(1^\lambda, \mathbf{hk}, L)$
$(\mathbf{y}, a) \leftarrow L$
For $i = 1, \ldots, |\mathbf{y}|$ do $\overline{\mathbf{P}}[i] \leftarrow \mathrm{C}_{1^\lambda, \mathbf{hk}[i], \mathbf{y}[i]}$
$b' \leftarrow\$ \mathcal{A}(1^\lambda, \overline{\mathbf{P}}, a)$ ; Return $b'$
```

```
Circuit $\mathrm{C}_{1^\lambda, hk, y}(k')$
$y' \leftarrow \mathsf{H.Ev}(1^\lambda, hk, k')$
If $(y = y')$ then return 1
Else return 0
```

Then $\mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},\mathcal{S}^{\mathsf{X}},\mathcal{D}}(\lambda) = \mathsf{Adv}^{\mathsf{pfo}}_{\mathsf{Obf},\mathsf{X},\mathcal{A}}(\lambda)$. So for any $\mathsf{X} \in \mathbf{X}$ the PFO security of Obf relative to X follows from the assumed UCE[$\{\mathcal{S}^{\mathsf{X}}\}$]-security of H. ▮

One special case of this we mention is when $\mathbf{X} = \mathbf{X}^{\mathrm{cup}} \cap \mathbf{X}^1$, so that PFO[$\mathbf{X}$] is AIPO. Theorem 4.3 and the remarks preceding it imply that we get this assuming UCE[$\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}} \cap \mathbf{S}^{1,1}$]-security. This special case of our result was independently and concurrently obtained in [21]. Note that BM [20] showed that UCE[$\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}} \cap \mathbf{S}^{1,1}$]-security is achievable assuming iO and AIPO. It follows from our result that UCE[$\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}} \cap \mathbf{S}^{1,1}$] and AIPO are equivalent, assuming iO.

References

[1] P. Ananth, D. Boneh, S. Garg, A. Sahai, and M. Zhandry. Differing-inputs obfuscation and applications. Cryptology ePrint Archive, Report 2013/689, 2013. `http://eprint.iacr.org/2013/689`. 3, 15, 16

[2] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. In J. Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 1–18. Springer, Aug. 2001. 1, 3, 15, 16

[3] M. Bellare, A. Boldyreva, and A. O'Neill. Deterministic and efficiently searchable encryption. In A. Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 535–552. Springer, Aug. 2007. 3, 9, 10

[4] M. Bellare, M. Fischlin, A. O'Neill, and T. Ristenpart. Deterministic encryption: Definitional equivalences and constructions without random oracles. In D. Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 360–378. Springer, Aug. 2008. 3, 10

[5] M. Bellare and V. T. Hoang. Resisting randomness subversion: Fast deterministic and hedged public-key encryption in the standard model. In E. Oswald and M. Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 627–656. Springer, Apr. 2015. 3, 4

[6] M. Bellare, V. T. Hoang, and S. Keelveedhi. Instantiating random oracles via UCEs. Cryptology ePrint Archive, Report 2013/424, 2013. Preliminary version in CRYPTO 2013. 1, 2, 3, 4, 11, 12, 16, 17

[7] M. Bellare and P. Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In S. Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 409–426. Springer, May / June 2006. 15

[8] M. Bellare, I. Stepanovs, and S. Tessaro. Poly-many hardcore bits for any one-way function and a framework for differing-inputs obfuscation. In P. Sarkar and T. Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 102–121. Springer, Dec. 2014. 3, 15

[9] M. Bellare, I. Stepanovs, and S. Tessaro. Contention in cryptoland: Obfuscation, leakage and UCE. Cryptology ePrint Archive, Report 2015/487, 2015. `http://eprint.iacr.org/2015/487`. 2

[10] N. Bitansky and R. Canetti. On strong simulation and composable point obfuscation. In T. Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 520–537. Springer, Aug. 2010. 1

[11] N. Bitansky, R. Canetti, Y. T. Kalai, and O. Paneth. On virtual grey box obfuscation for general circuits. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Part II*, volume 8617 of *LNCS*, pages 108–125. Springer, Aug. 2014. 3

[12] N. Bitansky, R. Canetti, O. Paneth, and A. Rosen. On the existence of extractable one-way functions. In D. B. Shmoys, editor, *46th ACM STOC*, pages 505–514. ACM Press, May / June 2014. 2, 8

[13] N. Bitansky and O. Paneth. Point obfuscation and 3-round zero-knowledge. In R. Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 190–208. Springer, Mar. 2012. 1, 2, 6

[14] N. Bitansky and O. Paneth. On the impossibility of approximate obfuscation and applications to resettable cryptography. In D. Boneh, T. Roughgarden, and J. Feigenbaum, editors, *45th ACM STOC*, pages 241–250. ACM Press, June 2013. 1

[15] A. Boldyreva, S. Fehr, and A. O'Neill. On notions of security for deterministic encryption, and efficient constructions without random oracles. In D. Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 335–359. Springer, Aug. 2008. 3, 10, 11

[16] E. Boyle, K.-M. Chung, and R. Pass. On extractability obfuscation. In Y. Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 52–73. Springer, Feb. 2014. 3, 6, 9, 15, 16

[17] Z. Brakerski and G. Segev. Better security for deterministic public-key encryption: The auxiliary-input setting. In P. Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 543–560. Springer, Aug. 2011. 3, 10, 11

[18] C. Brzuska, P. Farshim, and A. Mittelbach. Indistinguishability obfuscation and UCEs: The case of computationally unpredictable sources. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 188–205. Springer, Aug. 2014. 4, 17

[19] C. Brzuska and A. Mittelbach. Indistinguishability obfuscation versus multi-bit point obfuscation with auxiliary input. In P. Sarkar and T. Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 142–161. Springer, Dec. 2014. 3, 4, 11

[20] C. Brzuska and A. Mittelbach. Using indistinguishability obfuscation via UCEs. In P. Sarkar and T. Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 122–141. Springer, Dec. 2014. 1, 4, 12

[21] C. Brzuska and A. Mittelbach. Universal computational extractors and the superfluous padding assumption for indistinguishability obfuscation. Cryptology ePrint Archive, Report 2015/581, 2015. `http://eprint.iacr.org/2015/581`. 1, 4, 12

[22] R. Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In B. S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 455–469. Springer, Aug. 1997. 1, 2, 6

[23] R. Canetti and R. R. Dakdouk. Extractable perfectly one-way functions. In L. Aceto, I. Damgård, L. A. Goldberg, M. M. Halldórsson, A. Ingólfsdóttir, and I. Walukiewicz, editors, *ICALP 2008, Part II*, volume 5126 of *LNCS*, pages 449–460. Springer, July 2008. 3, 4, 6, 8

[24] R. Canetti and R. R. Dakdouk. Obfuscating point functions with multibit output. In N. P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 489–508. Springer, Apr. 2008. 1

[25] R. Canetti, Y. T. Kalai, M. Varia, and D. Wichs. On symmetric encryption and point obfuscation. In D. Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 52–71. Springer, Feb. 2010. 1

[26] R. Canetti, D. Micciancio, and O. Reingold. Perfectly one-way probabilistic hash functions (preliminary version). In *30th ACM STOC*, pages 131–140. ACM Press, May 1998. 1, 2, 6

[27] Y. Dodis, C. Ganesh, A. Golovnev, A. Juels, and T. Ristenpart. A formal treatment of backdoored pseudorandom generators. In E. Oswald and M. Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 101–126. Springer, Apr. 2015. 4

[28] Y. Dodis, Y. T. Kalai, and S. Lovett. On cryptography with auxiliary input. In M. Mitzenmacher, editor, *41st ACM STOC*, pages 621–630. ACM Press, May / June 2009. 1, 2

[29] D. M. Freeman, O. Goldreich, E. Kiltz, A. Rosen, and G. Segev. More constructions of lossy and correlation-secure trapdoor functions. In P. Q. Nguyen and D. Pointcheval, editors, *PKC 2010*, volume 6056 of *LNCS*, pages 279–295. Springer, May 2010. 3, 11

[30] B. Fuller, A. O'Neill, and L. Reyzin. A unified approach to deterministic encryption: New constructions and a connection to computational entropy. *Journal of Cryptology*, 28(3):671–717, July 2015. 3, 8

[31] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49. IEEE Computer Society Press, Oct. 2013. 3, 16

[32] C. Gentry, A. Lewko, A. Sahai, and B. Waters. Indistinguishability obfuscation from the multilinear subgroup elimination assumption. Cryptology ePrint Archive, Report 2014/309, 2014. `http://eprint.iacr.org/2014/309`. 3

[33] S. Goldwasser and Y. T. Kalai. On the impossibility of obfuscation with auxiliary input. In *46th FOCS*, pages 553–562. IEEE Computer Society Press, Oct. 2005. 1, 2, 6

[34] B. Hemenway and R. Ostrovsky. Building lossy trapdoor functions from lossy encryption. In K. Sako and P. Sarkar, editors, *ASIACRYPT 2013, Part II*, volume 8270 of *LNCS*, pages 241–260. Springer, Dec. 2013. 3, 11

[35] B. Lynn, M. Prabhakaran, and A. Sahai. Positive results and techniques for obfuscation. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 20–39. Springer, May 2004. 1, 2, 6

[36] T. Matsuda and G. Hanaoka. Chosen ciphertext security via point obfuscation. In Y. Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 95–120. Springer, Feb. 2014. 1

[37] T. Matsuda and G. Hanaoka. Chosen ciphertext security via UCE. In H. Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 56–76. Springer, Mar. 2014. 4

[38] R. Pass, K. Seth, and S. Telang. Indistinguishability obfuscation from semantically-secure multilinear encodings. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 500–517. Springer, Aug. 2014. 3

[39] C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In R. E. Ladner and C. Dwork, editors, *40th ACM STOC*, pages 187–196. ACM Press, May 2008. 3, 11

[40] A. Sahai and B. Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In D. B. Shmoys, editor, *46th ACM STOC*, pages 475–484. ACM Press, May / June 2014. 16

[41] H. Wee. On obfuscating point functions. In H. N. Gabow and R. Fagin, editors, *37th ACM STOC*, pages 523–532. ACM Press, May 2005. 1, 2, 3, 6

[42] H. Wee. Dual projective hashing and its applications - lossy trapdoor functions and more. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 246–262. Springer, Apr. 2012. 3, 11

[43] D. Wichs. Barriers in cryptography with weak, correlated and leaky sources. In R. D. Kleinberg, editor, *ITCS 2013*, pages 111–126. ACM, Jan. 2013. 3

[44] X. Xie, R. Xue, and R. Zhang. Deterministic public key encryption and identity-based encryption from lattices in the auxiliary-input setting. In I. Visconti and R. D. Prisco, editors, *SCN 12*, volume 7485 of *LNCS*, pages 1–18. Springer, Sept. 2012. 3, 11

[45] H. Xue, B. Li, X. Lu, D. Jia, and Y. Liu. Efficient lossy trapdoor functions based on subgroup membership assumptions. In M. Abdalla, C. Nita-Rotaru, and R. Dahab, editors, *CANS 13*, volume 8257 of *LNCS*, pages 235–250. Springer, Nov. 2013. 3, 11

A Notation and standard definitions

NOTATION. We denote by $\lambda \in \mathbb{N}$ the security parameter and by 1^λ its unary representation. We let ε denote the empty string. If s is an integer then $\mathsf{Pad}_s(\mathrm{C})$ denotes circuit C padded to have size s. We say that circuits $\mathrm{C}_0, \mathrm{C}_1$ are equivalent, written $\mathrm{C}_0 \equiv \mathrm{C}_1$, if they agree on all inputs. If $\mathbf{x}$ is a vector then $|\mathbf{x}|$ denotes the number of its coordinates and $\mathbf{x}[i]$ denotes its i-th coordinate. We write $x \in \mathbf{x}$ as shorthand for $x \in \{\mathbf{x}[1], \ldots, \mathbf{x}[|\mathbf{x}|]\}$. If X is a finite set, we let $x \leftarrow\!\!\$\; X$ denote picking an element of X uniformly at random and assigning it to x. Algorithms may be randomized unless otherwise indicated. Running time is worst case. "PT" stands for "polynomial-time," whether for randomized algorithms or deterministic ones. If A is an algorithm, we let $y \leftarrow A(x_1, \ldots ; r)$ denote running A with random coins r on inputs $x_1, \ldots$ and assigning the output to y. We let $y \leftarrow\!\!\$\; A(x_1, \ldots)$ be the result of picking r at random and letting $y \leftarrow A(x_1, \ldots ; r)$. We let $[A(x_1, \ldots)]$ denote the set of all possible outputs of A when invoked with inputs $x_1, \ldots$. We say that $f \colon \mathbb{N} \to \mathbb{R}$ is negligible if for every positive polynomial p, there exists $\lambda_p \in \mathbb{N}$ such that $f(\lambda) < 1/p(\lambda)$ for all $\lambda > \lambda_p$. We use the code based game playing framework of [7]. (See Fig. 1 for an example.) By $\mathrm{G}^{\mathcal{A}}(\lambda)$ we denote the event that the execution of game G with adversary $\mathcal{A}$ and security parameter λ results in the game returning true.

OBFUSCATORS. An *obfuscator* is a PT algorithm Obf that on input 1^λ and a circuit C returns a circuit $\overline{\mathrm{C}}$ such that $\overline{\mathrm{C}} \equiv \mathrm{C}$. (That is, $\overline{\mathrm{C}}(x) = \mathrm{C}(x)$ for all x.) We refer to the latter as the *correctness condition.* If $\mathbf{c}$ is an n-vector of circuits then $\mathsf{Obf}(1^\lambda, \mathbf{c})$ denotes the vector $(\mathsf{Obf}(1^\lambda, \mathbf{c}[1]), \ldots, \mathsf{Obf}(1^\lambda, \mathbf{c}[n]))$ formed by applying Obf independently to each coordinate of $\mathbf{c}$. We consider various notions of security for obfuscators, namely iO and variants of point-function obfuscation, including AIPO.

INDISTINGUISHABILITY OBFUSCATION. Although our results need only iO, we use diO [2, 16, 1] in the proof, applying BCP [16] to then reduce the assumption to iO. To give the definitions compactly, we use the definitional framework of BST [8] which allows us to capture iO variants (including diO) via classes of circuit samplers. Let Obf be an obfuscator. A *sampler* in this context is a PT algorithm S that on input 1^λ returns a triple $(\mathrm{C}_0, \mathrm{C}_1, aux)$ where $\mathrm{C}_0, \mathrm{C}_1$ are circuits of the same size, number of inputs and number of outputs, and aux is a string. If $\mathcal{O}$ is an adversary and $\lambda \in \mathbb{N}$ we let $\mathsf{Adv}^{\mathsf{io}}_{\mathsf{Obf},\mathsf{S},\mathcal{O}}(\lambda) = 2\Pr[\mathrm{IO}^{\mathcal{O}}_{\mathsf{Obf},\mathsf{S}}(\lambda)] - 1$

Game $\mathrm{DIFF}_{\mathsf{S}}^{\mathcal{D}}(\lambda)$	Game $\mathrm{IO}_{\mathsf{Obf},\mathsf{S}}^{\mathcal{O}}(\lambda)$
$(\mathrm{C}_0, \mathrm{C}_1, aux) \xleftarrow{\$} \mathsf{S}(1^\lambda)$	$b \xleftarrow{\$} \{0,1\}$; $(\mathrm{C}_0, \mathrm{C}_1, aux) \xleftarrow{\$} \mathsf{S}(1^\lambda)$
$x \xleftarrow{\$} \mathcal{D}(\mathrm{C}_0, \mathrm{C}_1, aux)$	$\overline{\mathrm{C}} \xleftarrow{\$} \mathsf{Obf}(1^\lambda, \mathrm{C}_b)$; $b' \xleftarrow{\$} \mathcal{O}(1^\lambda, \overline{\mathrm{C}}, aux)$
Return $(\mathrm{C}_0(x) \neq \mathrm{C}_1(x))$	Return $(b = b')$

Figure 5: Games defining difference-security of circuit sampler S and iO-security of obfuscator Obf relative to circuit sampler S.

where game $\mathrm{IO}_{\mathsf{Obf},\mathsf{S}}^{\mathcal{O}}(\lambda)$ is defined in Fig. 5. Now let $\boldsymbol{S}$ be a class (set) of circuit samplers. We say that Obf is $\boldsymbol{S}$-secure if $\mathsf{Adv}^{\mathsf{io}}_{\mathsf{Obf},\mathsf{S},\mathcal{O}}(\cdot)$ is negligible for every PT adversary $\mathcal{O}$ and every circuit sampler $\mathsf{S} \in \boldsymbol{S}$. We say that circuit sampler S produces equivalent circuits if there exists a negligible function ν such that $\Pr\left[\, \mathrm{C}_0 \equiv \mathrm{C}_1 \;:\; (\mathrm{C}_0, \mathrm{C}_1, aux) \xleftarrow{\$} \mathsf{S}(1^\lambda) \,\right] \geq 1 - \nu(\lambda)$ for all $\lambda \in \mathbb{N}$. Let $\boldsymbol{S}_{\mathrm{eq}}$ be the class of all circuit samplers that produce equivalent circuits. We say that Obf is an indistinguishability obfuscator if it is $\boldsymbol{S}_{\mathrm{eq}}$-secure [2, 31, 40].

We say that a circuit sampler S is difference secure if $\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{S},\mathcal{D}}(\cdot)$ is negligible for every PT adversary $\mathcal{D}$, where $\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{S},\mathcal{D}}(\lambda) = \Pr[\mathrm{DIFF}_{\mathsf{S}}^{\mathcal{D}}(\lambda)]$ and game $\mathrm{DIFF}_{\mathsf{S}}^{\mathcal{D}}(\lambda)$ is defined in Fig. 5. Difference security of S means that given $\mathrm{C}_0, \mathrm{C}_1, aux$ it is hard to find an input on which the circuits differ [2, 16, 1]. Let $\boldsymbol{S}_{\mathrm{diff}}$ be the class of all difference-secure circuit samplers. We say that circuit sampler S produces d-differing circuits, where $d\colon \mathbb{N} \to \mathbb{N}$, if for all $\lambda \in \mathbb{N}$ circuits C_0 and C_1 differ on at most $d(\lambda)$ inputs with an overwhelming probability over $(\mathrm{C}_0, \mathrm{C}_1, aux) \xleftarrow{\$} \mathsf{S}(1^\lambda)$. Let $\boldsymbol{S}_{\mathrm{diff}}(d)$ be the class of all difference-secure circuit samplers that produce d-differing circuits, so that $\boldsymbol{S}_{\mathrm{eq}} \subseteq \boldsymbol{S}_{\mathrm{diff}}(d) \subseteq \boldsymbol{S}_{\mathrm{diff}}$. The interest of this definition is the following result of BCP [16] that we use:

Proposition A.1 *If d is a polynomial then any $\boldsymbol{S}_{\mathrm{eq}}$-secure circuit obfuscator is also an $\boldsymbol{S}_{\mathrm{diff}}(d)$-secure circuit obfuscator.*

<u>Function families.</u> A family of functions F specifies the following. PT key generation algorithm $\mathsf{F.Kg}$ takes 1^λ to return a key $fk \in \{0,1\}^{\mathsf{F.kl}(\lambda)}$, where $\mathsf{F.kl}\colon \mathbb{N} \to \mathbb{N}$ is the key length function associated to F. Deterministic, PT evaluation algorithm $\mathsf{F.Ev}$ takes 1^λ, key $fk \in [\mathsf{F.Kg}(1^\lambda)]$ and an input $x \in \{0,1\}^{\mathsf{F.il}(\lambda)}$ to return an output $\mathsf{F.Ev}(1^\lambda, fk, x) \in \{0,1\}^{\mathsf{F.ol}(\lambda)}$, where $\mathsf{F.il}, \mathsf{F.ol}\colon \mathbb{N} \to \mathbb{N}$ are the input and output length functions associated to F, respectively. We say that F is *injective* if the function $\mathsf{F.Ev}(1^\lambda, fk, \cdot)\colon \{0,1\}^{\mathsf{F.il}(\lambda)} \to \{0,1\}^{\mathsf{F.ol}(\lambda)}$ is injective for every $\lambda \in \mathbb{N}$ and every $fk \in [\mathsf{F.Kg}(1^\lambda)]$. Notions of security for function families that we use are OWF and UCE, defined in Section 4.1 and Appendix A respectively.

<u>UCE security.</u> We recall the Universal Computational Extractor (UCE) framework of BHK [6]. Let H be a family of functions. Let $\mathcal{S}$ be an adversary called the *source* and $\mathcal{D}$ an adversary called the *distinguisher*. We associate to them and H the game $\mathrm{UCE}_{\mathsf{H}}^{\mathcal{S},\mathcal{D}}(\lambda)$ in the left panel of Fig. 6. We will use what BHK [6] call the multi-key version of UCE, so that associated to $\mathcal{S}$ is a polynomial $\mathcal{S}.\mathsf{nk}$ that indicates how many keys $\mathcal{S}$ uses. The source has access to an oracle HASH, and a query to HASH consists of an index i of a key and the actual input x, which is a string required to have length $\mathsf{H.il}(\lambda)$. When the challenge bit b is 1 (the "real" case) the oracle responds via $\mathsf{H.Ev}$ under a key $\mathbf{hk}[i]$ that is chosen by the game and *not* given to the source. When $b = 0$ (the "random" case) it responds as a random oracle. The source then leaks a string L to its accomplice distinguisher. The latter *does* get the key vector $\mathbf{hk}$ as input and must now return its guess $b' \in \{0,1\}$ for b. The game returns true iff $b' = b$, and the uce-advantage of $(\mathcal{S}, \mathcal{D})$ is defined for $\lambda \in \mathbb{N}$ via $\mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},\mathcal{S},\mathcal{D}}(\lambda) = 2\Pr[\mathrm{UCE}_{\mathsf{H}}^{\mathcal{S},\mathcal{D}}(\lambda)] - 1$. If $\mathbf{S}$ is a class (set) of sources, we say that H is UCE[$\mathbf{S}$]-secure if $\mathsf{Adv}^{\mathsf{uce}}_{\mathsf{H},\mathcal{S},\mathcal{D}}(\cdot)$ is negligible for all sources $\mathcal{S} \in \mathbf{S}$ and all PT distinguishers $\mathcal{D}$.

It is easy to see that UCE[$\mathbf{S}$]-security is not achievable if $\mathbf{S}$ is the class of all PT sources [6]. To obtain meaningful notions of security, BHK [6] impose restrictions on the source. A central restriction is

```
Game $\mathrm{UCE}_{\mathsf{H}}^{\mathcal{S},\mathcal{D}}(\lambda)$
$b \leftarrow\!\!\$\ \{0,1\}$
For $i = 1, \ldots, \mathcal{S}.\mathsf{nk}(\lambda)$ do $\mathbf{hk}[i] \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$
$L \leftarrow\!\!\$\ \mathcal{S}^{\mathrm{HASH}}(1^\lambda)$
$b' \leftarrow\!\!\$\ \mathcal{D}(1^\lambda, \mathbf{hk}, L)$
Return $(b' = b)$

$\mathrm{HASH}(i, x)$
If not $(1 \leq i \leq \mathcal{S}.\mathsf{nk}(\lambda))$ then return $\perp$
If $T[i, x] = \perp$ then
    If $b = 0$ then $T[i, x] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.ol}(\lambda)}$
    Else $T[i, x] \leftarrow \mathsf{H.Ev}(1^\lambda, \mathbf{hk}[i], x)$
Return $T[i, x]$
```

```
Game $\mathrm{PRED}_{\mathcal{S}}^{\mathcal{P}}(\lambda)$
$X \leftarrow \emptyset$
$L \leftarrow\!\!\$\ \mathcal{S}^{\mathrm{HASH}}(1^\lambda)$
$x \leftarrow\!\!\$\ \mathcal{P}(1^\lambda, L)$
Return $(x \in X)$

$\mathrm{HASH}(i, x)$
If not $(1 \leq i \leq \mathcal{S}.\mathsf{nk}(\lambda))$ then return $\perp$
If $T[i, x] = \perp$ then
    $T[i, x] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.ol}(\lambda)}$
$X \leftarrow X \cup \{x\}$
Return $T[i, x]$
```

Figure 6: Games defining UCE security of function family H and unpredictability of source $\mathcal{S}$.

unpredictability. A source is unpredictable if it is hard to guess the source's HASH queries even given the leakage, in the *random case* of the UCE game. Formally, let $\mathcal{S}$ be a source and $\mathcal{P}$ an adversary called a predictor and consider game $\mathrm{PRED}_{\mathcal{S}}^{\mathcal{P}}(\lambda)$ in Fig. 6. For $\lambda \in \mathbb{N}$ we let $\mathsf{Adv}_{\mathcal{S},\mathcal{P}}^{\mathsf{pred}}(\lambda) = \Pr[\mathrm{PRED}_{\mathcal{S}}^{\mathcal{P}}(\lambda)]$. We say that $\mathcal{S}$ is computationally unpredictable if $\mathsf{Adv}_{\mathcal{S},\mathcal{P}}^{\mathsf{pred}}(\cdot)$ is negligible for all PT predictors $\mathcal{P}$, and let $\mathbf{S}^{\mathrm{cup}}$ be the class of all PT computationally unpredictable sources. We say that $\mathcal{S}$ is statistically unpredictable if $\mathsf{Adv}_{\mathcal{S},\mathcal{P}}^{\mathsf{pred}}(\cdot)$ is negligible for all (not necessarily PT) predictors $\mathcal{P}$, and let $\mathbf{S}^{\mathrm{sup}} \subseteq \mathbf{S}^{\mathrm{cup}}$ be the class of all PT statistically unpredictable sources. We say that $\mathcal{S}$ is sub-exponentially unpredictable if there is an $\epsilon > 0$ such that for any PT predictor $\mathcal{P}$ there is a $\lambda_{\mathcal{P}}$ such that $\mathsf{Adv}_{\mathcal{S},\mathcal{P}}^{\mathsf{pred}}(\lambda) \leq 2^{-\lambda^\epsilon}$ for all $\lambda \geq \lambda_{\mathcal{P}}$ and let $\mathbf{S}^{\mathrm{seup}} \subseteq \mathbf{S}^{\mathrm{cup}}$ be the class of all PT sub-exponentially unpredictable sources.

BFM [18] show that $\mathrm{UCE}[\mathbf{S}^{\mathrm{cup}}]$-security is not achievable assuming that indistinguishability obfuscation is possible. This has lead applications to either be based on $\mathrm{UCE}[\mathbf{S}^{\mathrm{sup}}]$ or on subsets of $\mathrm{UCE}[\mathbf{S}^{\mathrm{cup}}]$, meaning to impose further restrictions on the source. $\mathrm{UCE}[\mathbf{S}^{\mathrm{sup}}]$, introduced in [6, 18], seems at this point to be a viable assumption. In order to restrict the computational case, one can consider split sources as defined in BHK [6]. We let $\mathbf{S}^{\mathrm{splt}}$ denote the class of split sources. Another way to restrict a UCE source is by limiting the number of queries it can make. Let $\mathbf{S}^{n,q}$ be the class of sources $\mathcal{S}$ such that $\mathcal{S}.\mathsf{nk}(\cdot) \leq n(\cdot)$ and $\mathcal{S}$ makes at most $q(\cdot)$ queries to each key. In particular $\mathbf{S}^{1,1}$ is the class of sources that use only one key and make only one query to it.