Point-Function Obfuscation: A Framework and Generic Constructions

Mihir Bellare[1] and Igors Stepanovs[2]

[1] Department of Computer Science and Engineering, University of California San Diego, USA. http://cseweb.ucsd.edu/~mihir/
[2] Department of Computer Science and Engineering, University of California San Diego, USA. https://sites.google.com/site/igorsstepanovs/

Abstract. We give a definitional framework for point-function obfuscation in which security is parameterized by a class of algorithms we call target generators. Existing and new notions are captured and explained as corresponding to different choices of this class. This leads to an elegant question: Is it 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 new point-function obfuscators, including many under standard assumptions. We end with a broader look that relates different known and possible notions of point function obfuscation to each other and to ours.

1 Introduction

In the theory of point-function obfuscation (PO), there are many different goals and definitions. It is (at least to us) hard territory to navigate. Meanwhile, there are few constructions; indeed, there are fewer constructions than there are definitions. And the ones that exist use strong assumptions. We try to bring some structure and unity to this area via a parameterized definitional framework, generic constructions and relations between definitions.

1.1 The state of point-function obfuscation

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 $\mathbf{I}_k$, meaning 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. We now discuss the state of the area with regard to both definitions and constructions.

Definitions. The theory of PO contains a large number of different goals and definitions. Sometimes there is auxiliary information [35, 14, 21], other times

not [23, 27, 39, 47]. Sometimes security pertains to a single target, other times to many [25]. Sometimes the formalization is a VBB-style simulation based one, other times indistinguishability based. Within each category, there are variants, for example, for indistinguishability, whether the necessary unpredictability condition on targets should be for polynomial-time or unbounded adversaries, and with negligible or sub-exponential advantage. And this list is not complete.

While from one perspective there are too many definitions, from other perspectives there are too few. Think of different elements that have been considered (for example whether or not auxiliary information is present, one target or many, polynomial-time or unbounded predictability adversaries, ... , in the context of an indistinguishability-based definition) as dimensions or axes in a multi-dimensional space. Then definitions in the literature can be seen as capturing some points in this space. But there is no systematic attempt to look in some unified way at all the points in this space. There is a connection that does not seem to have been explicitly made and pursued, namely that definitionally, there is little to no difference between PO and deterministic public-key encryption DPKE [3, 4, 17] or other forms of entropic security [30, 36]. Existing systematic and in-depth consideration of DPKE definitions and relations between them [4, 17] can be exploited to obtain semantic-security formalizations of PO that address issues with current definitions, and also to obtain definitional relations.

Constructions. Existing constructions use strong assumptions and achieve only some of the goals. A primary construction is from the AI-DHI (Auxiliary-Input Diffie-Hellman Inversion) assumption [23, 14]. Calling it a construction is a bit of a stretch; the security just amounts to the assumption. The latter cannot co-exist with VGBO (Virtual Grey Box Obfuscation) [10]. That doesn't mean it is wrong (perhaps VGBO does not exist) but it would be preferable to base PO on assumptions not in contention with VGBO. Wee [47] provides a construction 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 [35] show that the construction does not in fact provide security in the presence of auxiliary information. However BP [14] 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 targets that are hard to predict given the auxiliary information. BP [14] 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 [13]. DKL [29] use a novel assumption they call LSN to give a construction for targets that are exponentially hard to predict given the auxiliary information. BHK [6] give a construction for statistically hard to predict targets and no auxiliary information based on a multi-key version of their UCE assumption. There are simple constructions in the ROM [39].

In summary, there are few (standard-model) constructions and those that exist all use strong and sometimes novel assumptions. Also, each construction achieves a different variant of the goal and it is hard to visualize, or say in a

concise way, what has been done. The framework that we now discuss provides language to do this.

1.2 Contributions in brief

We pick one, simple indistinguishability-based definitional template. Using this, we provide a framework parameterized by a class $\mathbf{X}$ of objects we call *target generators*, giving a definition of what it means for a point-function obfuscator to be IND[$\mathbf{X}$] secure. This allows us to recover and explain different notions in the literature as each corresponding to a choice of $\mathbf{X}$, and also obtain many natural new ones, points in the above-mentioned multi-dimensional space that had not been explicitly considered.

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. As a consequence we obtain new constructions for both old and new forms of PO.

We then step back to consider other definitions of PO. These include existing simulation and indistinguishability style notions, as well as new, semantic security style ones emanating from the above-mentioned connection to DPKE. We formulate these also in a parameterized framework and then provide relations (implications and separations) between these notions and our IND notion.

We now look at these three contributions in more detail.

1.3 Definitional framework

Recall that a point-function obfuscator Obf takes input $\mathbf{I}_k$ and returns another circuit $\overline{\mathrm{P}}$ that is functionally equivalent to $\mathbf{I}_k$. Security requires that $\overline{\mathrm{P}}$ hides k. 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. We measure security of a candidate point-function obfuscator Obf relative to X. To do this, we associate to an adversary $\mathcal{A}$ its advantage $\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{Obf},\mathsf{X},\mathcal{A}}(\cdot)$ 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 independently obfuscating $\mathbf{I}_k$ for each of the targets k from $\mathbf{k}$ ($b = 1$) 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 IND[$\mathbf{X}$]-secure if $\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{Obf},\mathsf{X},\mathcal{A}}(\cdot)$ is negligible for all polynomial time $\mathcal{A}$ and all $\mathsf{X} \in \mathbf{X}$. See Section 3 for a formal definition.

What we have here is a notion of point-function obfuscation parameterized by a class of target generators. We view the latter 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.

IND[$\mathbf{X}$]-security is not achievable for all $\mathbf{X}$. For example, X could pick $\mathbf{k}[1]$ 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 the target points produced by X are unpredictable given a. In Section 3 we formalize a prediction game and advantage so that we can define the classes $\mathbf{X}^{\rm cup}, \mathbf{X}^{\rm seup}$ and $\mathbf{X}^{\rm sup}$ 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. IND$[\mathbf{X}^{\rm cup} \cap \mathbf{X}^{\varepsilon} \cap \mathbf{X}^{1}]$ is plain point-function obfuscation [23, 27, 39, 47], where there is just one target point, no auxiliary information, and unpredictability is computational. IND$[\mathbf{X}^{\rm cup} \cap \mathbf{X}^{1}]$ is AIPO [14, 20], where there is again one target point, but auxiliary information is now present, while unpredictability continues to be computational. IND$[\mathbf{X}^{\rm cup}]$ is composable AIPO [25], where there are many arbitrarily correlated target points, auxiliary information is present, and unpredictability is computational. DKL [29] achieve IND$[\mathbf{X}^{\rm sup} \cap \mathbf{X}^{1}]$, where there is a single target that is statistically hard to predict given the auxiliary information. BHK [6] achieve IND$[\mathbf{X}^{\rm sup} \cap \mathbf{X}^{\varepsilon}]$, where there are multiple targets, unpredictability is statistical, and there is no auxiliary information. Other prior notions can be captured in similar ways, and many natural new notions emerge as well.

1.4 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 IND$[\mathbf{X}]$-secure. If we can do this, we can, in one unified swoop, obtain constructions for a wide variety of forms of PO, 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 (multi-key) UCE.

One natural objection at this point is that we know that IND$[\mathbf{X}]$ is not achievable for some choices of $\mathbf{X}$. For example, assuming iO, this is true for $\mathbf{X} = \mathbf{X}^{\rm cup}$, meaning composable PO. (This follows by combining [20, 24].) So how can our constructions achieve IND$[\mathbf{X}]$ 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.

PO FROM IO. The emergence of candidate constructions for iO (indistinguishability obfuscation) [33, 43, 12, 34] raised a natural hope, namely that one could obtain PO 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 PO. More precisely, we show iO + OWF$[\mathbf{X}] \Rightarrow$ IND$[\mathbf{X}]$: Given iO and a family of functions that is one-way relative

to $\mathbf{X}$ as defined in Section 5.1 we can construct an obfuscator that is IND[$\mathbf{X}$]-secure. The construction, result and proof are in Section 5.1. 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 [9]. 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 note that to guarantee the usual (perfect) correctness condition of a PO, we require the OWF to be injective.

We highlight the simplest case of this result as still being novel and of interest. Namely, given iO and an ordinary injective OWF, we achieve plain point-function obfuscation, IND[$\mathbf{X}^{\mathrm{cup}} \cap \mathbf{X}^{\varepsilon} \cap \mathbf{X}^{1}$] in our notation. Previous constructions have been under assumptions that at this point seem less accepted than iO, and Wee [47] gives various arguments as to why this goal is hard under standard assumptions. Also on the negative side, combining our result with [20, 24] allows us, under iO, to rule out OWF[$\mathbf{X}^{\mathrm{cup}}$] (one-way functions secure for polynomially-many, computationally unpredictable correlated inputs), at least in the injective case.

PO 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, 32, 5, 48, 50, 45, 42, 38].

We show how to leverage these for point-function obfuscation via our second generic construction. We show that PRIV1[$\mathbf{X}$] $\Rightarrow$ IND[$\mathbf{X}$]. 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 [49] does not apply. The construction, result and proof are in Section 5.2.

From the LTDF-based DPKE scheme of BFO [15] and LTDFs from [44, 31, 48, 37, 51] we now get IND[$\mathbf{X}^{\mathrm{sup}} \cap \mathbf{X}^{\varepsilon} \cap \mathbf{X}^{1}$]-secure obfuscators under a large number of standard assumptions. We also get IND[$\mathbf{X}^{\mathrm{seup}} \cap \mathbf{X}^{1}$]-secure obfuscators

under the DLIN, Subgroup Indistinguishability and LWE assumptions via [17, 50, 48]. On the negative side we can rule out PRIV1[$\mathbf{X}^{\text{cup}}$]-secure DPKE under iO via [20, 24].

PO 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, 21, 41, 5, 28].

In this work we use the multi-key version of UCE, abbreviated mUCE [6]. We show how to associate to any given class $\mathbf{X}$ of target generators a class $\mathbf{S}^{\mathbf{X}}$ of sources such that mUCE[$\mathbf{S}^{\mathbf{X}}$] $\Rightarrow$ IND[$\mathbf{X}$], meaning we can build a point-function obfuscator secure for $\mathbf{X}$ given a family of functions that is mUCE[$\mathbf{S}^{\mathbf{X}}$]-secure. The definition of $\mathbf{S}^{\mathbf{X}}$ is given in Section 5.3. But what is most relevant here is that the strength of UCE-framework assumptions is very sensitive to the choice of class of sources that parameterizes the particular 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. mUCE[$\mathbf{S}^{\mathbf{X}}$]-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.

1.5 Alternative notions and relations between notions

Above, we fixed one, basic definitional template, which we called IND, and then parameterized it by classes $\mathbf{X}$ of target generators to get notions IND[$\mathbf{X}$]. However, there are other possible choices for the basic template, some emanating from the literature, and others from the definitional similarity of PO with DPKE. We consider parameterized versions of some of these and relate them to each other and to IND. Specifically we define and consider the following (see Section 6 for formal definitions):

- SIM[$\mathbf{X}$]: (Simulation) The first definitions for PO simply restricted VBB security [2] to the class of point functions [39, 47, 35, 25]. With SSS[$\mathbf{X}$] we give an $\mathbf{X}$-parameterized version of this.
- SIND[$\mathbf{X}$]: (Strong Indistinguishability) Recall that in IND[$\mathbf{X}$], the adversary decision bit is produced as a function of the vector $\overline{\mathbf{P}}$ of obfuscated point functions and the auxiliary information a. In SIND[$\mathbf{X}$], this bit is not the final decision, but is passed to another adversary who produces the final decision based on it and the target vector itself. This is a parameterized version of the definition of [23].
- CSS[$\mathbf{X}$]: (Comparison-based semantic security) This is an analogue of comparision based semantic security for boolean functions for DPKE [4] in which the adversary needs to compute some predicate on the target vector and auxiliary information.

- SSS[$\mathbf{X}$]: (Simulation-based semantic security) This is an analogue of simulation based semantic security for boolean functions for DPKE [4] in which a simulator with an oracle for the point functions must compute a predicate on the target vectors and auxiliary information.

Fig. 8 shows the relations between five parameterized notions of PO, namely the four above and our original IND[$\mathbf{X}$].

1.6 Discussion and further related work

In concurrent and independent work, BM3 [22] take first steps towards a parameterized definition for point-function obfuscation, with separate definitions for the basic and composable cases. They also show that injective mUCE-secure function families for strongly unpredictable sources making one oracle query per key implies composable AIPO (both for computational and statistical unpredictability), which is a special case of our mUCE result.

Multi-bit auxiliary-input point-function obfuscation (MB-AIPO) [25, 11, 40] allows one to obfuscate the circuit $\mathbf{I}_{k,m}$ that on input k' returns m if $k = k'$ and $\perp$ otherwise, where k, m are strings. CD [25] show that composable AIPO implies MB-AIPO. MB-AIPO was subsequently used in BP [14] and MH [40]. BM1 [20] show that if iO is possible then MB-AIPO is not. MB-AIPO seems to be quite a bit stronger than AIPO itself and in particular this result does not rule out AIPO.

In Section 5.1 we define OWF[$\mathbf{X}$], one-wayness of a function family relative to a class of target generators, the targets here being the inputs to the OWF. We note that OWF[$\mathbf{X}^{\mathrm{sup}} \cap \mathbf{X}^{\varepsilon}$] (inputs are statistically unpredictable and there is no auxiliary information) is the notion of a one-way correlation intractable hash (CIH) function family as per GOR [36].

Our parameterized PRIV1[$\mathbf{X}$] notions of security for DPKE schemes apply equally to function families and thus recover, via particular choices of $\mathbf{X}$, some of the security notions for CIH function families from GOR [36]. In these cases, since our DPKE-based constructions of PO do not require that decryption in the DPKE scheme is polynomial-time, CIH function families meeting the corresponding notions suffice as well.

Seeing that prior work can be characterized in terms of intersections of certain basic classes in our framework makes apparent that so far the literature has considered only a few points from the larger space of all possible intersections. A systematic consideration of the full space (which is lacking) would surface other notions of interest and give a coherent picture of the area.

2 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. 3 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 $\overline{\mathrm{C}}$. 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 $\mathbf{C}$. The *correctness condition* of obfuscator Obf requires that for every circuit C, every $\lambda \in \mathbb{N}$ and every $\overline{\mathrm{C}} \in [\mathsf{Obf}(1^\lambda, \mathrm{C})]$ we have $\overline{\mathrm{C}} \equiv \mathrm{C}$ (meaning $\overline{\mathrm{C}}(x) = \mathrm{C}(x)$ for all x). We also call the latter a *perfect* correctness condition and we require that it holds for all obfuscators. We consider various notions of security for obfuscators, namely indistinguishability obfuscation 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 [9] 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 $\mathsf{Adv}^{\mathsf{io}}_{\mathsf{Obf},\mathsf{S},\mathcal{O}}(\lambda) = 2\Pr[\mathrm{IO}^{\mathcal{O}}_{\mathsf{Obf},\mathsf{S}}(\lambda)] - 1$ where game $\mathrm{IO}^{\mathcal{O}}_{\mathsf{Obf},\mathsf{S}}(\lambda)$ is defined in Fig. 1. 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 $\Pr[\mathrm{C}_0 \equiv \mathrm{C}_1 \ :\ (\mathrm{C}_0, \mathrm{C}_1, aux) \leftarrow\!\!\$\ \mathsf{S}(1^\lambda)] \geq 1 - \nu(\lambda)$ 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, 33, 46].

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 $\mathsf{Adv}^{\mathsf{diff}}_{\mathsf{S},\mathcal{D}}(\lambda) = \Pr[\mathrm{DIFF}^{\mathcal{D}}_{\mathsf{S}}(\lambda)]$ and game $\mathrm{DIFF}^{\mathcal{D}}_{\mathsf{S}}(\lambda)$ is defined in Fig. 1. 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

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

Fig. 1. Games defining difference-security of circuit sampler S and iO-security of obfuscator Obf relative to circuit sampler S.

most $d(\lambda)$ inputs with an overwhelming probability over $(\mathrm{C}_0, \mathrm{C}_1, aux) \leftarrow\!\!\$\ \mathsf{S}(1^\lambda)$. 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 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.*

Function families. 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 mUCE and OWF, the latter defined in Section 5.1.

UCE framework. We recall the Universal Computational Extractor (UCE) framework of BHK [6]. We will use what BHK call the multi-key version of UCE (mUCE). It is an extension of the more commonly used UCE notion for a single key, meaning that it implies the latter. Meanwhile, no implications in the other direction (from single-key to multi-key) are known.

Let H be a family of functions. Let $\mathcal{S}$ be an adversary called the *source* and $\mathcal{D}$ an adversary called the *distinguisher*. Consider game $\mathrm{mUCE}^{\mathcal{S},\mathcal{D}}_{\mathsf{H}}(\lambda)$ in the left panel of Fig. 2. 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. 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 $b' = b$. The advantage of $(\mathcal{S}, \mathcal{D})$ against the mUCE security of H is defined for $\lambda \in \mathbb{N}$ via $\mathsf{Adv}^{\mathsf{m\text{-}uce}}_{\mathsf{H},\mathcal{S},\mathcal{D}}(\lambda) = 2\Pr[\mathrm{mUCE}^{\mathcal{S},\mathcal{D}}_{\mathsf{H}}(\lambda)] - 1$. If $\mathbf{S}$ is a class (set) of sources, we say that H is mUCE[$\mathbf{S}$]-secure if $\mathsf{Adv}^{\mathsf{m\text{-}uce}}_{\mathsf{H},\mathcal{S},\mathcal{D}}(\cdot)$ is negligible for all sources $\mathcal{S} \in \mathbf{S}$ and all PT distinguishers $\mathcal{D}$.

$\underline{\mathrm{mUCE}^{\mathcal{S},\mathcal{D}}_{\mathsf{H}}(\lambda)}$	$\underline{\mathrm{mSPRED}^{\mathcal{P}}_{\mathcal{S}}(\lambda)}$
For $i = 1, \ldots, \mathcal{S}.\mathsf{nk}(\lambda)$ do $\mathbf{hk}[i] \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)$	$X \leftarrow \emptyset\ ;\ L \leftarrow\!\!\$\ \mathcal{S}^{\mathrm{HASH}}(1^\lambda)$
$b \leftarrow\!\!\$\ \{0,1\}\ ;\ L \leftarrow\!\!\$\ \mathcal{S}^{\mathrm{HASH}}(1^\lambda)$	$x \leftarrow\!\!\$\ \mathcal{P}(1^\lambda, L)$
$b' \leftarrow\!\!\$\ \mathcal{D}(1^\lambda, \mathbf{hk}, L)$	Return $(x \in X)$
Return $(b = b')$	
$\underline{\mathrm{HASH}(i,x)}$	$\underline{\mathrm{HASH}(i,x)}$
If not $(1 \le i \le \mathcal{S}.\mathsf{nk}(\lambda))$ then return $\perp$	If not $(1 \le i \le \mathcal{S}.\mathsf{nk}(\lambda))$ then
If $T[i,x] = \perp$ then	Return $\perp$
If $b = 0$ then $T[i,x] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.ol}(\lambda)}$	If $T[i,x] = \perp$ then
Else $T[i,x] \leftarrow \mathsf{H.Ev}(1^\lambda, \mathbf{hk}[i], x)$	$T[i,x] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.ol}(\lambda)}$
Return $T[i,x]$	$X \leftarrow X \cup \{x\}$
	Return $T[i,x]$

Fig. 2. Games defining mUCE security of function family H and unpredictability of source $\mathcal{S}$.

It is easy to see that mUCE[$\mathbf{S}$]-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 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 mUCE game. Formally, let $\mathcal{S}$ be a source and $\mathcal{P}$ an adversary called a predictor and consider game $\mathrm{mSPRED}^{\mathcal{P}}_{\mathcal{S}}(\lambda)$ in Fig. 2. For $\lambda \in \mathbb{N}$ we let $\mathsf{Adv}^{\mathsf{m\text{-}spred}}_{\mathcal{S},\mathcal{P}}(\lambda) = \Pr[\mathrm{mSPRED}^{\mathcal{P}}_{\mathcal{S}}(\lambda)]$. We say that $\mathcal{S}$ is computationally unpredictable if $\mathsf{Adv}^{\mathsf{m\text{-}spred}}_{\mathcal{S},\mathcal{P}}(\cdot)$ 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 $\mathsf{Adv}^{\mathsf{m\text{-}spred}}_{\mathcal{S},\mathcal{P}}(\cdot)$ 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 $\mathsf{Adv}^{\mathsf{m\text{-}spred}}_{\mathcal{S},\mathcal{P}}(\lambda) \le 2^{-\lambda^\epsilon}$ for all $\lambda \ge \lambda_{\mathcal{P}}$, 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 UCE-framework security notions (both single-key and multi-key) are not achievable for $\mathbf{S}^{\mathrm{cup}}$ assuming that indistinguishability obfuscation exists. This has lead applications to impose further restrictions on the source by using either $\mathbf{S}^{\mathrm{sup}}$ or subsets of $\mathbf{S}^{\mathrm{cup}}$. Assumptions based on $\mathbf{S}^{\mathrm{sup}}$, introduced in [6, 18], at this point seem to be a viable. In order to restrict the computational case, one can consider split sources as defined in BHK [6]. Such sources can leak information about oracle queries and answers separately, but not together. We let $\mathbf{S}^{\mathrm{splt}}$ denote the class of split sources. Another way to restrict a 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 $\mathcal{S}.\mathsf{nk}(\cdot) \le n(\cdot)$ and $\mathcal{S}$ makes at most $q(\cdot)$ queries

Game $\mathrm{IND}^{\mathcal{A}}_{\mathsf{Obf},\mathsf{X}}(\lambda)$	Game $\mathrm{PRED}^{\mathcal{Q}}_{\mathsf{X}}(\lambda)$	Game $\mathrm{TRIV}^{\mathcal{A}}_{\mathsf{X}}(\lambda)$
$b \leftarrow\!\!\$\ \{0,1\}$	$(\mathbf{k}, a) \leftarrow\!\!\$\ \mathsf{X.Ev}(1^\lambda)$	$b \leftarrow\!\!\$\ \{0,1\}$
$(\mathbf{k}_1, a_1) \leftarrow\!\!\$\ \mathsf{X.Ev}(1^\lambda)$	$k \leftarrow\!\!\$\ \mathcal{Q}(1^\lambda, a)$	$(\mathbf{k}_1, a_1) \leftarrow\!\!\$\ \mathsf{X.Ev}(1^\lambda)$
For $i = 1, \ldots, \mathsf{X.vl}(\lambda)$ do	Return $(\exists i : \mathbf{k}[i] = k)$	For $i = 1, \ldots, \mathsf{X.vl}(\lambda)$ do
$\quad \mathbf{k}_0[i] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{X.tl}(\lambda)}$		$\quad \mathbf{k}_0[i] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{X.tl}(\lambda)}$
$\overline{\mathbf{P}} \leftarrow\!\!\$\ \mathsf{Obf}(1^\lambda, \mathbf{I}_{\mathbf{k}_b})$		$b' \leftarrow\!\!\$\ \mathcal{A}(1^\lambda, \mathbf{k}_b, a_1)$
$b' \leftarrow\!\!\$\ \mathcal{A}(1^\lambda, \overline{\mathbf{P}}, a_1)$		Return $(b = b')$
Return $(b = b')$		

Fig. 3. Games defining IND security of point-function obfuscator Obf relative to target generator X, unpredictabilty of target generator X and triviality of target generator X.

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.

3 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.

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. 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 Section 2. 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 on inputs $\mathbf{I}_k$ with $k \in \{0,1\}^{\mathsf{Obf.tl}(\lambda)}$.

Security of 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.

Consider game IND of Fig. 3 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 $\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{Obf},\mathsf{X},\mathcal{A}}(\lambda) = 2\Pr[\mathrm{IND}^{\mathcal{A}}_{\mathsf{Obf},\mathsf{X}}(\lambda)] - 1$. 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 IND[$\mathbf{X}$]-secure if $\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{Obf},\mathsf{X},\mathcal{A}}(\cdot)$ is negligible for every PT $\mathcal{A}$ and every $\mathsf{X} \in \mathbf{X}$. We now capture different notions in the literature, as well as new ones, by considering particular classes $\mathbf{X}$. At the end of this section we will present what we call the triviality theorem, showing how the definition is vacuous for some classes, and discuss its implications. We will further discuss alternative security definitions for point-function obfuscation in Section 6.

Classes of target generators. One important (and necessary) condition on a target generator is unpredictability. To define this, consider game PRED of Fig. 3 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, encompassing 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 $(\mathbf{k}, a) \in [\mathsf{X.Ev}(1^{\lambda})]$ and all $\lambda \in \mathbb{N}$.

We can recover notions from the literature as follows:

— $\mathrm{IND}[\mathbf{X}^{\mathrm{cup}} \cap \mathbf{X}^{\varepsilon} \cap \mathbf{X}^{1}]$ — This is basic point-function obfuscation, secure for a single computationally unpredictable target point, and no auxiliary information is allowed. It is achieved in [23, 27, 39, 47].

— $\mathrm{IND}[\mathbf{X}^{\mathrm{cup}} \cap \mathbf{X}^{1}]$ — This is AIPO [35, 14], secure for a single computationally unpredictable target point in the presence of auxiliary information. It is achieved under the AI-DHI assumption by Canetti [23], and using the extended construction of Wee [47] by BP [14].

— $\mathrm{IND}[\mathbf{X}^{\mathrm{cup}}]$ — This is composable AIPO [25], meaning that it is secure for arbitrarily many correlated target points that are computationally unpredictable in the presence of auxiliary information. BM1 [20] showed that this notion cannot co-exit with iO in the presence of OWFs.

— $\mathrm{IND}[\mathbf{X}^{\mathrm{sup}} \cap \mathbf{X}^{\varepsilon}]$ — This is composable point-function obfuscation, secure for arbitrarily many correlated target points that are statistically unpredictable, and no auxiliary information is allowed. It is achieved from $\mathrm{mUCE}[\mathbf{S}^{\mathrm{sup}}]$ in BHK [6].

Furthermore, DKL [29] achieve $\mathrm{IND}[\mathbf{X}^{\mathrm{sup}} \cap \mathbf{X}^{1}]$ from the LSN (i.e. auxiliary-input LPN) assumption and BM3 [22] build $\mathrm{IND}[\mathbf{X}^{\mathrm{sup}}]$ from $\mathrm{mUCE}[\mathbf{S}^{\mathrm{s\text{-}sup}} \cap \mathbf{X}^{1}]$. Here $\mathbf{S}^{\mathrm{s\text{-}sup}}$ denotes a subclass of $\mathbf{S}^{\mathrm{sup}} \cap \mathbf{S}^{\mathrm{splt}}$ that is used to denote sources with "strong statistical unpredictability", as defined in BM2 [21]. We note that some of the above results achieve notions that are stronger than IND. Such notions are discussed and defined in Section 6.

Triviality theorem. The $\mathrm{IND}[\mathbf{X}]$ definition has the peculiar property of trivializing for some choices of $\mathbf{X}$. For example, let X be a target generator that returns a vector of random, independent targets and auxiliary information $a = \varepsilon$ the empty string. Then *any* point-function obfuscator Obf is $\mathrm{IND}[\{\mathsf{X}\}]$-secure. This is true because game IND in this case samples $\mathbf{k}_0, \mathbf{k}_1$ from the same distribution and the information provided to the adversary $\mathcal{A}$ is thus independent of the challenge bit. Before discussing and assessing what this means for the definition, we provide a general *triviality theorem* that characterizes for what choices of $\mathbf{X}$ this phenomenon happens.

Consider game TRIV of Fig. 3 associated to a target generator X and an adversary $\mathcal{A}$. For $\lambda \in \mathbb{N}$ let $\mathsf{Adv}^{\mathsf{triv}}_{\mathsf{X},\mathcal{A}}(\lambda) = 2\Pr[\mathrm{TRIV}^{\mathcal{A}}_{\mathsf{X}}(\lambda)] - 1$. We say that X is trivial if $\mathsf{Adv}^{\mathsf{triv}}_{\mathsf{X},\mathcal{A}}(\cdot)$ is negligible for every PT $\mathcal{A}$. An example of trivial X is the one given above. Let $\mathbf{X}^{\mathrm{triv}}$ be the class of all trivial target generators, and say that a class $\mathbf{X}$ is trivial if $\mathbf{X} \subseteq \mathbf{X}^{\mathrm{triv}}$. The proof of the following triviality theorem follows directly from the definitions of games IND and TRIV and is omitted.

Game $\mathrm{MDIFF}^{\mathcal{D}}_{\mathsf{S}}(\lambda)$	Game $\mathrm{MIO}^{\mathcal{O}}_{\mathsf{Obf},\mathsf{S}}(\lambda)$
$(\mathbf{C}_0, \mathbf{C}_1, aux) \leftarrow\!\!\$\ \mathsf{S}(1^\lambda)$	$b \leftarrow\!\!\$\ \{0,1\}$; $(\mathbf{C}_0, \mathbf{C}_1, aux) \leftarrow\!\!\$\ \mathsf{S}(1^\lambda)$
$x \leftarrow\!\!\$\ \mathcal{D}(\mathbf{C}_0, \mathbf{C}_1, aux)$	$\overline{\mathbf{C}} \leftarrow\!\!\$\ \mathsf{Obf}(1^\lambda, \mathbf{C}_b)$; $b' \leftarrow\!\!\$\ \mathcal{O}(1^\lambda, \overline{\mathbf{C}}, aux)$
Return $(\exists i\ :\ \mathbf{C}_0[i](x) \neq \mathbf{C}_1[i](x))$	Return $(b = b')$

Fig. 4. Games defining difference-security of multi-circuit sampler S and iO-security of obfuscator Obf relative to multi-circuit sampler S.

Theorem 2. *Let* $\mathbf{X} \subseteq \mathbf{X}^{\mathrm{triv}}$ *be a class of target generators. Let* Obf *be any point-function obfuscator. Then* Obf *is* $\mathrm{IND}[\mathbf{X}]$*-secure.*

This can be viewed as a defect of the IND definition, but whether or not this is true is debatable. The IND definition has been successfully employed in applications [14, 21]. In these cases, $\mathbf{X} = \mathbf{X}^{\mathrm{cup}} \cap \mathbf{X}^1$, a class to which Theorem 2 does not apply. This indicates that the classes of target generators arising in applications are naturally not trivial. And the constructions we give in Section 5 cover such non-trivial classes. Thus we are on the whole unsure whether or not Theorem 2 should be viewed as a definitional weakness. In Section 6 we will provide alternative security definitions for PO that avoid this type of triviality theorem and are meaningful for all choices of target generators. But if an application can be obtained via IND, then it seems preferable, since this definition is simpler and easier to use and, from Section 5, we have more constructions for it.

4 (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 Section 2 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.)

iO FOR MULTI-CIRCUIT SAMPLERS. A multi-circuit sampler is a PT algorithm S with an associated circuit-vector length function $\mathsf{S.vl}\colon \mathbb{N} \to \mathbb{N}$. 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 $\mathsf{S.vl}(\lambda)$, 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 $i \in \{1, \ldots, \mathsf{S.vl}(\lambda)\}$.

Consider game MIO of Fig. 4 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

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

Fig. 5. 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.

$\mathsf{Adv}^{\mathsf{m\text{-}io}}_{\mathsf{Obf},\mathsf{S},\mathcal{O}}(\cdot)$ is negligible for every multi-circuit sampler $\mathsf{S} \in \boldsymbol{S}$ and every PT adversary $\mathcal{O}$.

Consider game MDIFF of Fig. 4 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 $i \in \{1, \ldots, \mathsf{S.vl}(\lambda)\}$. Let $\boldsymbol{S}_{\text{m-diff}}(d)$ be the class of all difference-secure multi-circuit samplers that produce d-differing circuits. The proof of the following lemma is provided in [8].

Lemma 3. *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.*

5 Generic constructions of PO

Prior constructions have targeted IND[$\mathbf{X}$] for specific choices of $\mathbf{X}$ in ad hoc ways and used non-standard 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 IND[$\mathbf{X}$]-secure.

5.1 PO from iO

OWFs. Consider game OWF of Fig. 5 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

Games G_0, G_1

$b \leftarrow\!\!\$\ \{0,1\}$; $(\mathbf{k}_1, a_1) \leftarrow\!\!\$\ \mathsf{X.Ev}(1^\lambda)$
For $i = 1, \ldots, \mathsf{X.vl}(\lambda)$ do
  $\mathbf{k}_0[i] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{X.tl}(\lambda)}$; $\mathbf{fk}[i] \leftarrow\!\!\$\ \mathsf{F.Kg}(1^\lambda)$; $\mathbf{y}[i] \leftarrow \mathsf{F.Ev}(1^\lambda, \mathbf{fk}[i], \mathbf{k}_b[i])$
  $\overline{\mathbf{P}}[i] \leftarrow\!\!\$\ \mathsf{Obf}_{\mathsf{io}}(\mathrm{C}^1_{1^\lambda, \mathbf{fk}[i], \mathbf{y}[i]})$ // G_0
  $\overline{\mathbf{P}}[i] \leftarrow\!\!\$\ \mathsf{Obf}_{\mathsf{io}}(\mathsf{Pad}_{s(\lambda)}(\mathrm{C}^2))$ // G_1
$b' \leftarrow\!\!\$\ \mathcal{A}(1^\lambda, \overline{\mathbf{P}}, a_1)$; 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

Fig. 6. Games for proof of Theorem 4.

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 PO. The standard notion of a OWF is recovered as $\mathbf{X} = \emptyset$, meaning that F is secure only with respect to $\mathsf{X}^{\mathsf{1ur}}$.

The definition of CD [24] is the special case of ours with vectors of length one. That of FOR [32], 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 [13].

We now show that indistinguishability obfuscation can be used to build a IND[$\mathbf{X}$]-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 $\mathsf{Obf}_{\mathsf{io}}$ be an obfuscator. We construct a point-function obfuscator Obf with $\mathsf{Obf.tl} = \mathsf{F.il}$ as follows:

Algorithm $\mathsf{Obf}(1^\lambda, \mathbf{I}_k)$
$fk \leftarrow\!\!\$\ \mathsf{F.Kg}(1^\lambda)$; $y \leftarrow \mathsf{F.Ev}(1^\lambda, fk, k)$
$\overline{\mathrm{P}} \leftarrow\!\!\$\ \mathsf{Obf}_{\mathsf{io}}(\mathrm{C}_{1^\lambda, fk, y})$; 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. *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 $\mathsf{Obf}_{\mathsf{io}}$ be an indistinguishability obfuscator. Then Obf constructed above from F and $\mathsf{Obf}_{\mathsf{io}}$ is a* IND[$\mathbf{X}$]*-secure point-function obfuscator.*

Proof (Theorem 4). 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. 6, 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 $\mathrm{IND}^{\mathcal{A}}_{\mathsf{Obf},\mathsf{X}}(\lambda)$. 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

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

The first equality holds by the definition of IND, and the second equality holds because of $\Pr[\mathrm{G}_1] = 1/2$. We now show that $\Pr[\mathrm{G}_0] - \Pr[\mathrm{G}_1]$ is negligible, meaning that $\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{Obf},\mathsf{X},\mathcal{A}}(\cdot)$ 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 $\mathsf{S}(1^\lambda)$	Adversary $\mathcal{O}(1^\lambda, \overline{\mathbf{C}}, aux)$
$d \leftarrow\!\!\$\ \{0,1\}$; $(\mathbf{k}_1, a_1) \leftarrow\!\!\$\ \mathsf{X.Ev}(1^\lambda)$	$(d, a_1) \leftarrow aux$
For $i = 1, \ldots, \mathsf{X.vl}(\lambda)$ do	$d' \leftarrow\!\!\$\ \mathcal{A}(1^\lambda, \overline{\mathbf{C}}, a_1)$
$\quad \mathbf{k}_0[i] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{X.tl}(\lambda)}$	If $(d = d')$ then return 1
$\quad \mathbf{fk}[i] \leftarrow\!\!\$\ \mathsf{F.Kg}(1^\lambda)$; $\mathbf{y}[i] \leftarrow \mathsf{F.Ev}(1^\lambda, \mathbf{fk}[i], \mathbf{k}_d[i])$	Else return 0
$\quad \mathbf{C}_1[i] \leftarrow \mathrm{C}^1_{1^\lambda, \mathbf{fk}[i], \mathbf{y}[i]}$; $\mathbf{C}_0[i] \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathrm{C}^2)$	
$aux \leftarrow (d, a_1)$; Return $(\mathbf{C}_0, \mathbf{C}_1, aux)$	

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 1 (the result of BCP [16]), any indistinguishability obfuscator is also an $\boldsymbol{S}_{\text{diff}}(1)$-secure obfuscator. And according to Lemma 3, 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 $\mathcal{F}_0(1^\lambda, \mathbf{fk}, \mathbf{y}, a)$	Adversary $\mathcal{F}_1(1^\lambda, \mathbf{fk}, \mathbf{y}, a)$
$d \leftarrow 0$; $(\mathbf{k}_1, a_1) \leftarrow\!\!\$\ \mathsf{X.Ev}(1^\lambda)$	$d \leftarrow 1$
For $i = 1, \ldots, \lvert\mathbf{y}\rvert$ do	For $i = 1, \ldots, \lvert\mathbf{y}\rvert$ do
$\quad \mathbf{C}_1[i] \leftarrow \mathrm{C}^1_{1^\lambda, \mathbf{fk}[i], \mathbf{y}[i]}$	$\quad \mathbf{C}_1[i] \leftarrow \mathrm{C}^1_{1^\lambda, \mathbf{fk}[i], \mathbf{y}[i]}$
$\quad \mathbf{C}_0[i] \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathrm{C}^2)$	$\quad \mathbf{C}_0[i] \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathrm{C}^2)$
$aux \leftarrow (d, a_1)$; $x \leftarrow\!\!\$\ \mathcal{D}(\mathbf{C}_1, \mathbf{C}_0, aux)$	$aux \leftarrow (d, a)$; $x \leftarrow\!\!\$\ \mathcal{D}(\mathbf{C}_1, \mathbf{C}_0, aux)$
Return x	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

$$\Pr[\,\mathrm{MDIFF}^{\mathcal{D}}_{\mathsf{S}}(\lambda) \mid d = 0\,] = \Pr[\mathrm{OWF}^{\mathcal{F}_0}_{\mathsf{F},\mathsf{X}^{\mathsf{ur}}}(\lambda)],$$

$$\Pr[\,\mathrm{MDIFF}^{\mathcal{D}}_{\mathsf{S}}(\lambda) \mid d = 1\,] = \Pr[\mathrm{OWF}^{\mathcal{F}_1}_{\mathsf{F},\mathsf{X}}(\lambda)].$$

and $\mathsf{Adv}^{\mathsf{m\text{-}diff}}_{\mathsf{S},\mathcal{D}}(\lambda) = \frac{1}{2}(\mathsf{Adv}^{\mathsf{owf}}_{\mathsf{F},\mathsf{X}^{\mathsf{ur}},\mathcal{F}_0}(\lambda) + \mathsf{Adv}^{\mathsf{owf}}_{\mathsf{F},\mathsf{X},\mathcal{F}_1}(\lambda))$. Note that OWF[$\mathbf{X}$]-security of F requires that $\mathsf{Adv}^{\mathsf{owf}}_{\mathsf{F},\mathsf{X}^{1\mathsf{ur}},\mathcal{F}}(\lambda)$ is negligible for all PT adversaries $\mathcal{F}$. One can use the latter with a standard hybrid argument to further prove that $\mathsf{Adv}^{\mathsf{owf}}_{\mathsf{F},\mathsf{X}^{\mathsf{ur}},\mathcal{F}_0}(\lambda)$ is also negligible for all PT adversaries $\mathcal{F}_0$. It follows that the multi-circuit sampler S is difference-secure. The injectivity of F also implies that S produces 1-differing circuits. Therefore, $\mathsf{S} \in \boldsymbol{S}_{\text{m-diff}}(1)$.

5.2 PO 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. 5 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 $\mathsf{Obf}(1^\lambda, \mathbf{I}_k)$	Circuit $\mathrm{C}_{1^\lambda,pk,c}(k)$
$(pk, sk) \leftarrow\!\!\$\ \mathsf{DPKE.Kg}(1^\lambda)$	If $(\mathsf{DPKE.Enc}(1^\lambda, pk, k) = c)$
$c \leftarrow \mathsf{DPKE.Enc}(1^\lambda, pk, k)$; Return $\mathrm{C}_{1^\lambda,pk,c}$	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 5. *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* IND[$\mathbf{X}$]*-secure point-function obfuscator.*

Proof (Theorem 5). 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 IND[$\mathbf{X}$]-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 IND 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 $\mathcal{B}(1^\lambda, \mathbf{pk}, \mathbf{c}, a)$	Circuit $\mathrm{C}_{1^\lambda,pk,c}(k)$
For $i = 1, \ldots, \lvert\mathbf{c}\rvert$ do $\overline{\mathbf{P}}[i] \leftarrow \mathrm{C}_{1^\lambda,\mathbf{pk}[i],\mathbf{c}[i]}$	If $(\mathsf{DPKE.Enc}(1^\lambda, pk, k) = c)$
$b' \leftarrow\!\!\$\ \mathcal{A}(1^\lambda, \overline{\mathbf{P}}, a)$; Return b'	Then return 1 else return 0

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

In applying Theorem 5 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 [44, 31, 48, 37, 51]. Thus we get IND[$\mathbf{X}^{\mathrm{sup}} \cap \mathbf{X}^{\varepsilon} \cap \mathbf{X}^{1}$]-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, 50, 48]. Accordingly we get IND[$\mathbf{X}^{\mathrm{seup}} \cap \mathbf{X}^{1}$]-secure point-function obfuscators under the same assumptions. BH [5] obtain PRIV-secure DPKE from UCE[$\mathbf{S}^{\mathrm{sup}}$], which via Theorem 5 yields IND[$\mathbf{X}^{\mathrm{sup}} \cap \mathbf{X}^{\varepsilon} \cap \mathbf{X}^{1}$] under UCE[$\mathbf{S}^{\mathrm{sup}}$].

Theorem 5 also yields negative results. Assume iO exists. Then we know that there do not exist point function obfuscators that are IND[$\mathbf{X}^{\mathrm{cup}}$]-secure [20]. Theorem 5 then implies that there also do not exist deterministic public-key encryption schemes that are PRIV1[$\mathbf{X}^{\mathrm{cup}}$]-secure.

CIH function families as per GOR [36] do not seem to have a unique associated security notion. Rather the authors discuss a few choices. Our parameterized PRIV definitions above apply to function families as well and can be viewed as providing more security notions for CIH function families. These function families can also be used in our PO construction above as long as they are injective.

5.3 PO from UCE

Our next generic construction is based on UCE, a class of assumptions on function families from [6]. We use the multi-key version of the UCE assumption, denoted mUCE. As before we aim to provide point-function obfuscation secure for any given class of target generators. We are able to do this with mUCE by associating to the class of target generators a class of sources. The existence of an mUCE-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. Associate to it a point-function obfuscator Obf defined as follows. Let $\mathsf{Obf.tl} = \mathsf{H.il}$, and

$$\begin{array}{l|l}
\underline{\text{Algorithm } \mathsf{Obf}(1^\lambda, \mathbf{I}_k)} & \underline{\text{Circuit } \mathrm{C}_{1^\lambda, hk, y}(k')} \\
hk \leftarrow\!\!\$\ \mathsf{H.Kg}(1^\lambda)\ ;\ y \leftarrow \mathsf{H.Ev}(1^\lambda, hk, k) & y' \leftarrow \mathsf{H.Ev}(1^\lambda, hk, k') \\
\text{Return } \mathrm{C}_{1^\lambda, hk, y} & \text{If } (y = y') \text{ then return 1 else return 0}
\end{array}$$

The construction is simple and natural. The point-function obfuscation of $\mathbf{I}_k$ is a circuit that embeds the hash y of target 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:

$$\begin{array}{l}
\underline{\text{Source } \mathcal{S}^{\mathsf{X}}(1^\lambda)} \\
d \leftarrow\!\!\$\ \{0,1\}\ ;\ (\mathbf{k}_1, a_1) \leftarrow\!\!\$\ \mathsf{X.Ev}(1^\lambda) \\
\text{For } i = 1, \ldots, \mathsf{X.vl}(\lambda) \text{ do } \mathbf{k}_0[i] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{X.tl}(\lambda)} \\
\text{For } i = 1, \ldots, \mathsf{X.vl}(\lambda) \text{ do } \mathbf{y}[i] \leftarrow\!\!\$\ \mathrm{HASH}(i, \mathbf{k}_d[i]) \\
L \leftarrow ((d, a_1), \mathbf{y})\ ;\ \text{Return } L
\end{array}$$

The number of keys for this source is $\mathcal{S}^{\mathsf{X}}.\mathsf{nk} = \mathsf{X}.\mathsf{vl}$, 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 IND[$\mathbf{X}$]-secure assuming H is mUCE[$\mathbf{S}^{\mathbf{X}}$]-secure. To appreciate what this provides we now discuss the assumption further.

Assumptions in the UCE framework are 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 mUCE sources recalled in Section 2.

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, (d, a_1) 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 mUCE[$\mathbf{S}^{\mathbf{X}}$]-security is not achievable for all choices of $\mathbf{X}$. The value of our result is that it is entirely general, reducing IND security for a given $\mathbf{X}$ to a question of mUCE 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 mUCE-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 6. *Let* H *be an injective family of functions. Let* $\mathbf{X}$ *be a class of target generators such that* $\mathsf{X}.\mathsf{tl} = \mathsf{H}.\mathsf{il}$ *for all* $\mathsf{X} \in \mathbf{X}$*. Assume* H *is* mUCE[$\mathbf{S}^{\mathbf{X}}$]*-secure. Let* Obf *be as defined above. Then* Obf *is a* IND[$\mathbf{X}$]*-secure point-function obfuscator.*

Function family H is assumed to be injective in order to meet the perfect correctness condition of a point-function obfuscator, and it is not important for security. In [8] we show that *non-injective* mUCE is sufficient to construct a point-function obfuscator that satisfies a relaxed correctness condition and achieves the same security as above.

Proof (Theorem 6). 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 IND[$\mathbf{X}$]-secure.

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

Distinguisher $\mathcal{D}(1^\lambda, \mathbf{hk}, L)$	Circuit $\mathrm{C}_{1^\lambda, hk, y}(k')$
$((d, a_1), \mathbf{y}) \leftarrow L$	$y' \leftarrow \mathsf{H.Ev}(1^\lambda, hk, k')$
For $i = 1, \ldots, \lvert\mathbf{y}\rvert$ do $\overline{\mathbf{P}}[i] \leftarrow \mathrm{C}_{1^\lambda, \mathbf{hk}[i], \mathbf{y}[i]}$	If $(y = y')$ then return 1
$d' \leftarrow\!\!\$\ \mathcal{A}(1^\lambda, \overline{\mathbf{P}}, a_1)$	Else return 0
If $(d = d')$ then return 1 else return 0	

Let b denote the challenge bit in game $\mathrm{mUCE}^{\mathcal{S}^{\mathsf{X}}, \mathcal{D}}_{\mathsf{H}}(\lambda)$, and let b' denote the bit returned by $\mathcal{D}$ in the same game. We claim that

$$\Pr[\, b' = 1 \,|\, b = 1 \,] = \Pr[\mathrm{IND}^{\mathcal{A}}_{\mathsf{Obf},\mathsf{X}}(\lambda)] \qquad \text{and} \qquad \Pr[\, b' = 1 \,|\, b = 0 \,] = \frac{1}{2} \,.$$

The first equation holds by construction. The second equation is true because $\mathcal{D}$ runs $\mathcal{A}$ with inputs that are independent of the challenge bit d. Namely, for $b = 0$ the entries in $\mathbf{y}$ are uniform and independent, since the source $\mathcal{S}$ makes only one query per key index. We have $\mathsf{Adv}^{\mathsf{m\text{-}uce}}_{\mathsf{H},\mathcal{S}^{\mathsf{X}},\mathcal{D}}(\lambda) = \mathsf{Adv}^{\mathsf{ind}}_{\mathsf{Obf},\mathsf{X},\mathcal{A}}(\lambda)/2$. Therefore, for any $\mathsf{X} \in \mathbf{X}$ the IND security of Obf relative to X follows from the assumed $\mathrm{mUCE}[\{\mathcal{S}^{\mathsf{X}}\}]$-security of H.

Negative results for multi-key UCE. Let $n\colon \mathbb{N} \to \mathbb{N}$ be a polynomial such that $n(\cdot) \in \Omega((\cdot)^\epsilon)$. Theorem 6 allows us to conclude that $\mathrm{mUCE}[\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}} \cap \mathbf{S}^{n,1}]$-secure injective function families do not exist under certain assumptions. This is a simple corollary of the prior results which show that MB-AIPO can not co-exist with iO [25, 20]. We now explain our claim in more details.

Theorem 6 shows that the existence of $\mathrm{mUCE}[\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}} \cap \mathbf{S}^{n,1}]$-secure injective function families implies $\mathrm{IND}[\mathbf{X}^{\mathrm{cup}} \cap \mathbf{X}^n]$-secure point-function obfuscation. Note that the latter is a composable AIPO as per CD [25]. CD [25] show that composable AIPO can be used to construct MB-AIPO, which is an obfuscation that is secure for functions that map a target point to a multi-bit output (as opposed to an output in $\{0, 1\}$). Finally, BM1 [20] show that MB-AIPO cannot co-exist with iO, assuming one-way functions. These results imply the following:

Corollary 7. *Let* H *be an injective function family. Let* $n\colon \mathbb{N} \to \mathbb{N}$ *be a polynomial such that* $n(\cdot) \in \Omega((\cdot)^\epsilon)$ *for some constant* $\epsilon > 0$*. Assume the existence of one-way functions and indistinguishability obfuscation. Then* H *is not* $\mathrm{mUCE}[\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}} \cap \mathbf{S}^{n,1}]$*-secure.*

In a concurrent and independent work, BM3 [22] discuss a similar impossibility result for $\mathrm{mUCE}[\mathbf{S}^{\text{s-cup}} \cap \mathbf{S}^{n,1}]$-security. Here $\mathbf{S}^{\text{s-cup}}$ is a class of UCE sources introduced in (BM2) [21] who also show that $\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}} \subsetneq \mathbf{S}^{\text{s-cup}}$. We note that impossibility of $\mathrm{mUCE}[\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}} \cap \mathbf{S}^{n,1}]$-secure function families is a stronger result because it concerns a smaller class of sources.

No other impossibility results are known for mUCE exclusively, but any negative results for (single-key) UCE also apply to mUCE. Specifically, BFM [18] give

an iO-based attack on UCE[$\mathbf{S}^{\mathrm{cup}}$]. And BST [10] show that UCE[$\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}}$]-secure function families do not exist assuming the existence of OWFs and iO, which is a strictly stronger impossibility result than the latter. The result by BST [10] implies that mUCE[$\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}} \cap \mathbf{S}^{1,p}$]-secure function families do not exist for a polynomial $p(\cdot) \in \Omega((\cdot)^{\epsilon})$, but we currently do not know whether this notion is comparable to mUCE[$\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}} \cap \mathbf{S}^{n,1}$].

Related work. One special case of Theorem 6 is when $\mathbf{X} = \mathbf{X}^{\mathrm{cup}} \cap \mathbf{X}^{1}$, so that IND[$\mathbf{X}$] is AIPO. The theorem and the remarks preceding it imply that we get this assuming mUCE[$\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}} \cap \mathbf{S}^{1,1}$]-security. This special case of our result was independently and concurrently obtained in [22]. Note that BM2 [21] showed that mUCE[$\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}} \cap \mathbf{S}^{1,1}$]-security is achievable assuming iO and AIPO. It follows from our result that mUCE[$\mathbf{S}^{\mathrm{cup}} \cap \mathbf{S}^{\mathrm{splt}} \cap \mathbf{S}^{1,1}$] and AIPO are equivalent, assuming iO.

6 Alternative security notions for PO

In Section 3 we defined IND security of point-function obfuscation. It extends security notions that were used for variants of AIPO in BP [14], MH [40], MB1 [20] and MB3 [22]. The main difference is that IND is parameterized with a class of target generators, allowing us to unify the treatment of AIPO from the literature.

In this section we provide several alternative security notions for point-function obfuscation, and show relations between them and IND. Specifically, we extend the security notion introduced by Canetti [23] as well as the notions of average-case [29, 26] and worst-case [2, 39, 47, 35, 25] simulation-based security for point-function obfuscation. Similar to IND, our extended notions are parameterized with classes of target generators. We also define a novel security notion, called computational semantic security, by adapting the corresponding definition that was used for DPKE in [4] to the setting of point-function obfuscation and parameterizing it in the same way as above. Finally, we discuss the security achieved by our PO constructions from Section 5 with respect to the new notions.

Strong indistinguishability. Consider game SIND of Fig. 7 associated to a point-function obfuscator Obf, a target generator X, an adversary $\mathcal{A}$ and a distinguisher $\mathcal{D}$, such that $\mathcal{A}$ returns an output in $\{0,1\}$ and $\mathsf{Obf.tl} = \mathsf{X.tl}$. For $\lambda \in \mathbb{N}$ let $\mathsf{Adv}^{\mathsf{sind}}_{\mathsf{Obf},\mathsf{X},\mathcal{A},\mathcal{D}}(\lambda) = 2\Pr[\mathrm{SIND}^{\mathcal{A},\mathcal{D}}_{\mathsf{Obf},\mathsf{X}}(\lambda)] - 1$. Let $\mathbf{X}$ be a class of target generators. We say that Obf is SIND[$\mathbf{X}$]-secure if $\mathsf{Adv}^{\mathsf{sind}}_{\mathsf{Obf},\mathsf{X},\mathcal{A},\mathcal{D}}(\cdot)$ is negligible for every $\mathsf{X} \in \mathbf{X}$, every PT $\mathcal{A}$ and every PT $\mathcal{D}$. The difference between our definitions of IND and SIND is that the latter also runs a distinguisher in the last stage of the game, which makes this definition meaningful even for trivial target generators (as defined in Section 3). Our definition of SIND extends the security notion used for oracle hashing by Canetti [23], parameterizing it with classes of target generators. Another difference is that SIND samples target vectors $\mathbf{k}_0, \mathbf{k}_1$ from distributions that are potentially different, whereas [23] used the same distribution for both. Note that adversary $\mathcal{A}$ cannot be allowed to

Game $\mathrm{SIND}^{\mathcal{A},\mathcal{D}}_{\mathsf{Obf},\mathsf{X}}(\lambda)$	Game $\mathrm{CSS}^{\mathcal{A}}_{\mathsf{Obf},\mathsf{X}}(\lambda)$	Game $\mathrm{SSS}^{\mathcal{A},\mathcal{S},\mathcal{P}}_{\mathsf{Obf},\mathsf{X}}(\lambda)$
$b \leftarrow\!\!\$\ \{0,1\}$	$b \leftarrow\!\!\$\ \{0,1\}$	$b \leftarrow\!\!\$\ \{0,1\}$
$(\mathbf{k}_1, a_1) \leftarrow\!\!\$\ \mathsf{X.Ev}(1^\lambda)$	$(\mathbf{k}_1, a_1) \leftarrow\!\!\$\ \mathsf{X.Ev}(1^\lambda)$	$(\mathbf{k}, a) \leftarrow\!\!\$\ \mathsf{X.Ev}(1^\lambda)$
For $i = 1, \ldots, \mathsf{X.vl}(\lambda)$ do	For $i = 1, \ldots, \mathsf{X.vl}(\lambda)$ do	$\overline{\mathbf{P}} \leftarrow\!\!\$\ \mathsf{Obf}(1^\lambda, \mathbf{I}_{\mathbf{k}})$
$\quad \mathbf{k}_0[i] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{X.tl}(\lambda)}$	$\quad \mathbf{k}_0[i] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{X.tl}(\lambda)}$	$p \leftarrow\!\!\$\ \mathcal{P}(1^\lambda, \mathbf{k}, a)$
$\overline{\mathbf{P}} \leftarrow\!\!\$\ \mathsf{Obf}(1^\lambda, \mathbf{I}_{\mathbf{k}_b})$	$\overline{\mathbf{P}} \leftarrow\!\!\$\ \mathsf{Obf}(1^\lambda, \mathbf{I}_{\mathbf{k}_b})$	If $(b = 1)$ then
$d \leftarrow\!\!\$\ \mathcal{A}(1^\lambda, \overline{\mathbf{P}}, a_1)$	$t \leftarrow\!\!\$\ \mathcal{A}_1(1^\lambda, \mathbf{k}_1, a_1)$	$\quad p' \leftarrow\!\!\$\ \mathcal{A}(1^\lambda, \overline{\mathbf{P}}, a)$
$b' \leftarrow\!\!\$\ \mathcal{D}(1^\lambda, \mathbf{k}_1, a_1, d)$	$t' \leftarrow\!\!\$\ \mathcal{A}_2(1^\lambda, \overline{\mathbf{P}}, a_1)$	Else $p' \leftarrow\!\!\$\ \mathcal{S}^{\mathbf{I}_{\mathbf{k}}}(1^\lambda, a)$
Return $(b = b')$	If $(t = t')$ then $b' \leftarrow 1$	If $(p = p')$ then $b' \leftarrow 1$
	Else $b' \leftarrow 0$	Else $b' \leftarrow 0$
	Return $(b = b')$	Return $(b = b')$

Fig. 7. Games defining SIND security, CSS security and SSS security of point-function obfuscator Obf relative to target generator X.

return an output of an arbitrary length because then it would be able to return $\overline{\mathbf{P}}$, hence making the security trivially unachievable.

Computational semantic security. Consider game CSS of Fig. 7 associated to a point-function obfuscator Obf, a target generator X and an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ such that algorithms $\mathcal{A}_1, \mathcal{A}_2$ return outputs in $\{0,1\}$ and $\mathsf{Obf.tl} = \mathsf{X.tl}$. For $\lambda \in \mathbb{N}$ let $\mathsf{Adv}^{\mathsf{css}}_{\mathsf{Obf},\mathsf{X},\mathcal{A}}(\lambda) = 2\Pr[\mathrm{CSS}^{\mathcal{A}}_{\mathsf{Obf},\mathsf{X}}(\lambda)] - 1$. Let $\mathbf{X}$ be a class of target generators. We say that Obf is CSS[$\mathbf{X}$]-secure if $\mathsf{Adv}^{\mathsf{css}}_{\mathsf{Obf},\mathsf{X},\mathcal{A}}(\cdot)$ is negligible for every $\mathsf{X} \in \mathbf{X}$ and every PT $\mathcal{A}$. This is an adaptation of the definition of *computational semantic security* for DPKE from [4], which we further parameterize with classes of target generators. It asks that adversary $\mathcal{A}$ can not use an obfuscation $\overline{\mathbf{P}}$ of $\mathbf{k}_1$ to compute any partial information about the latter, even in the presence of auxiliary information a_1. This provides us with a better intuition about the desired security of point-function obfuscation, as opposed to the less intuitive definition of SIND.

Simulation-based semantic security. We consider two different definitions of simulation-based semantic security. Informally, both definitions require that for every PT adversary $\mathcal{A}$ that receives as input an obfuscation of some point-function $\mathbf{I}_k$, there exists a PT simulator with only an oracle access to $\mathbf{I}_k$, such that the output distribution of the former is indistinguishable from that of the latter. The two definitions differ in the way how $\mathbf{I}_k$ is chosen. One option is to quantify over all possible point-functions that can be produced by a particular target generator. For this purpose, we extend the definitions of *worst-case* security [2, 39, 47, 35, 25] for point-function obfuscation. We use SIM to denote our new security notion. An alternative approach is to use target generator X in order to sample point-functions. This follows the definitions of *average-case* security [29, 26] for point function obfuscation, and we use SSS to denote our extended security notion.

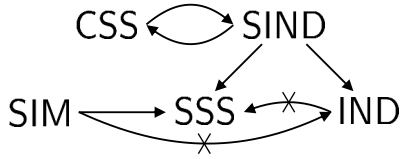

Fig. 8. Relations between security notions for point-function obfuscation.

Consider game SSS of Fig. 7 associated to a point-function obfuscator Obf, a target generator X, an adversary $\mathcal{A}$, a simulator $\mathcal{S}$ and a predicate algorithm $\mathcal{P}$, such that algorithms $\mathcal{A}, \mathcal{S}, \mathcal{P}$ return outputs in $\{0,1\}$ and $\mathsf{Obf.tl} = \mathsf{X.tl}$. For $\lambda \in \mathbb{N}$ let $\mathsf{Adv}^{\mathsf{sss}}_{\mathsf{Obf},\mathsf{X},\mathcal{A},\mathcal{S},\mathcal{P}}(\lambda) = 2\Pr[\mathrm{SSS}^{\mathcal{A},\mathcal{S},\mathcal{P}}_{\mathsf{Obf},\mathsf{X}}(\lambda)] - 1$. Let $\mathbf{X}$ be a class of target generators. We say that Obf is SSS[$\mathbf{X}$]-secure if for every target generator $\mathsf{X} \in \mathbf{X}$ and every PT $\mathcal{A}$ there exists PT $\mathcal{S}$ such that $\mathsf{Adv}^{\mathsf{sss}}_{\mathsf{Obf},\mathsf{X},\mathcal{A},\mathcal{S},\mathcal{P}}(\cdot)$ is negligible for every PT $\mathcal{P}$. Informally, this security notion requires that for every adversary $\mathcal{A}$ there exists a simulator $\mathcal{S}$ such that if $\mathcal{A}$ can use obfuscations $\mathbf{I}_{\mathbf{k}}$ to compute any property (function) $\mathcal{P}$ of $\mathbf{k}$, then $\mathcal{S}$ can do the same using only an oracle access to $\mathbf{I}_{\mathbf{k}}$ (meaning that $\mathcal{S}$ has oracle access to each of $\mathbf{I}_{\mathbf{k}[1]}, \ldots, \mathbf{I}_{\mathbf{k}[n]}$ for $n = |\mathbf{k}|$). This is required to hold even when $\mathcal{A}, \mathcal{S}, \mathcal{P}$ receive as input some auxiliary information a about $\mathbf{k}$.

SIM security. Next, we define the SIM-security of PO. Let $\mathbf{X}$ be a class of target generators. Let Obf be a point-function obfuscator. We say that Obf is SIM[$\mathbf{X}$]-secure if for every target generator $\mathsf{X} \in \mathbf{X}$ and every PT adversary $\mathcal{A}$ there exists a PT simulator $\mathcal{S}$ and a negligible function $\mu\colon \mathbb{N} \to \mathbb{N}$ such that

$$\left|\Pr[\mathcal{A}(1^\lambda, \mathsf{Obf}(1^\lambda, \mathbf{I}_{\mathbf{k}}), a) = \mathcal{P}(\mathbf{k}, a)] - \Pr[\mathcal{S}^{\mathbf{I}_{\mathbf{k}}}(1^\lambda, a) = \mathcal{P}(\mathbf{k}, a)]\right| \leq \mu(\lambda)$$

for every $\lambda \in \mathbb{N}$, every $(\mathbf{k}, a) \in [\mathsf{X.Ev}(1^\lambda)]$ and every PT predicate algorithm $\mathcal{P}$ that returns an output in $\{0,1\}$.

In the above definition of SIM-security, predicate $\mathcal{P}$ can be substituted with a constant function, resulting in an equivalent definition (as noted in [2, 47, 35]). In contrast, this is not true for the definition of SSS-security. Replacing $\mathcal{P}$ with a constant function will allow $\mathcal{S}$ to run X in order to generate fresh $(\mathbf{k}, a)$, obfuscate $\mathbf{I}_{\mathbf{k}}$ to get $\overline{\mathbf{P}}$, and simulate $\mathcal{A}$ on $\overline{\mathbf{P}}, a$. As a result, every obfuscator would be vaciously SSS-secure for any class of target generators $\mathbf{X}$.

Relations between security notions. Fig. 8 shows relations between the security notions for point-function obfuscation that are discussed in this paper. Consider any two security notions A and B. An arrow from A to B means that any A[$\mathbf{X}$]-secure point-function obfuscator is also B[$\mathbf{X}$]-secure, for every class of target generators $\mathbf{X}$. A crossed arrow going from A to B means that there exists an obfuscator Obf and a class of target generators $\mathbf{X}$ such that Obf is A[$\mathbf{X}$]-secure but not B[$\mathbf{X}$]-secure.

Implications SIM $\to$ SSS and SIND $\to$ IND trivially follow from our definitions of the corresponding security notions. The proofs for all other implications and separations shown in Fig. 8 are provided in [8]. The only relations that are

missing in the figure (and can not be deduced using transitivity) are those between SIM and both of SSS, CSS. We leave it as an open question to show the remaining relations between these security notions.

Security of our PO constructions. Let $\mathbf{X}$ be a class of target generators. In Section 5 we showed how to build a point-function obfuscator that is IND[$\mathbf{X}$]-secure, based on any of the following: a OWF[$\mathbf{X}$]-secure function family and an iO, or a PRIV1[$\mathbf{X}$]-secure DPKE, or an mUCE[$\mathbf{S}^{\mathbf{X}}$]-secure function family for $\mathbf{S}^{\mathbf{X}}$ as defined in Section 5.3. We do not know how to adapt our constructions to achieve SIM[$\mathbf{X}$]-security. But each of our construction achieves CSS[$\mathbf{X}$]-security, requiring only minimal changes in the used assumptions.

We now provide some intuition about our claim. Recall that game CSS computes $t \in \{0,1\}$ by running $\mathcal{A}_1(1^\lambda, \mathbf{k}_1, a_1)$, and subsequently compares it to the output of $\mathcal{A}_2(1^\lambda, \overline{\mathbf{P}}, a_1)$. This is different from game IND where the adversary consists only of an algorithm $\mathcal{A}(1^\lambda, \overline{\mathbf{P}}, a_1)$. The difficulty of adapting proofs of IND[$\mathbf{X}$]-security to achieve CSS[$\mathbf{X}$]-security is that in the latter $\mathbf{k}_1$ (required to run $\mathcal{A}_1$) and $\overline{\mathbf{P}}$ (required to run $\mathcal{A}_2$) are usually available in different stages of the security proof, meaning that one has to find a way to pass around the value of t (which depends on $\mathbf{k}_1$) across the stages. We resolve this by pushing t into the auxiliary information of target generators that parametrize our security notions.

Let $\mathbf{X}$ be a class of target generators. Let $\mathbf{P}$ be the set of all PT predicate algorithms $\mathcal{P}$ such that $\mathcal{P}(1^\lambda, \cdot, \cdot)\colon \{0,1\}^* \times \{0,1\}^* \to \{0,1\}$ for all $\lambda \in \mathbb{N}$. For any $\lambda \in \mathbb{N}$, $\mathsf{X} \in \mathbf{X}$ and $\mathcal{P} \in \mathbf{P}$ let $\mathsf{X}^{\mathcal{P}}$ be defined as follows:

$$\frac{\text{Source } \mathsf{X}^{\mathcal{P}}(1^\lambda)}{(\mathbf{k}, a) \leftarrow\!\!\$\ \mathsf{X}(1^\lambda)\ ;\ \beta \leftarrow\!\!\$\ \mathcal{P}(1^\lambda, \mathbf{k}, a)\ ;\ \text{Return } (\mathbf{k}, (a, \beta))}$$

where $\mathsf{X}^{\mathcal{P}}.\mathsf{vl} = \mathsf{X}.\mathsf{vl}$ and $\mathsf{X}^{\mathcal{P}}.\mathsf{tl} = \mathsf{X}.\mathsf{tl}$. We define a new class of target generators $\mathbf{X}' = \{\ \mathsf{X}^{\mathcal{P}}\ :\ \mathsf{X} \in \mathbf{X}, \mathcal{P} \in \mathbf{P}\ \}$. Then each of our constructions from Section 5 achieves CSS[$\mathbf{X}$]-security, based on either of the following: a OWF[$\mathbf{X}'$]-secure function family and an iO, or a PRIV1[$\mathbf{X}'$]-secure DPKE, or an mUCE[$\mathbf{S}^{\mathbf{X}'}$]-secure function family.

Note that for any $\mathsf{X} \in \mathbf{X}$ and $\mathcal{P} \in \mathbf{P}$, the construction of $\mathsf{X}^{\mathcal{P}}$ expands the auxiliary information of X only by a single bit. This means that $\mathsf{X}^{\mathcal{P}}$ inherits the unpredictability properties of X. Namely, for any $\lambda \in \mathbb{N}$, $\mathsf{X} \in \mathbf{X}$, $\mathcal{P} \in \mathbf{P}$ and any PT adversary $\mathcal{R}$ we can construct a PT adversary $\mathcal{Q}$ such that $\Pr[\mathrm{PRED}^{\mathcal{Q}}_{\mathsf{X}}(\lambda)] \geq \frac{1}{2}\Pr[\mathrm{PRED}^{\mathcal{R}}_{\mathsf{X}^{\mathcal{P}}}(\lambda)]$ for all $\lambda \in \mathbb{N}$. Adversary $\mathcal{Q}$ would attempt to guess the extra bit of information and then simulate $\mathcal{R}$. The same approach can be used to show that any OWF[$\mathbf{X}$]-secure function family is also OWF[$\mathbf{X}'$]-secure, recoving the construction of CSS[$\mathbf{X}$]-secure PO directly from a OWF[$\mathbf{X}$]-secure function family and an iO.

Definitional choices. All of our security notions for point-function obfuscation require that adversaries return single-bit outputs. This is consistent with the prior work. Specifically, simulation-based definitions in the prior literature always compare the outputs of adversary and simulator to either a predicate [27,

35] or a constant [2, 39, 47, 25, 29, 26]. However, it would be more intuitive to not restrict the size of outputs returned by adversaries in games CSS, SSS and SIM. The goal of these adversaries can be thought as to compute some "property" of the target vector, and there is no reason to limit it to a single bit.

The initial work on obfuscation [2] discusses various definitional choices and chooses to use the weakest of them to achieve stronger impossibility results. Subsequent work continues to use definitions of the same style even for positive results. We are not aware of any follow-up discussion on alternative definitions.

Some of our implications from Fig. 8 might change if adversaries in games CSS, SSS and SIM are allowed to return multiple-bit outputs. In particular, note that our definitions of CSS and SSS are similar to those that were used for DPKE schemes in BFOR [4], who showed them to be equivalent for multiple-bit outputs in their setting. We leave it as an open problem to extend our definitions to allow outputs of an arbitrary size.

Acknowledgments

Bellare and Stepanovs were supported in part by NSF grants CNS-1116800, CNS-1228890 and CNS-1526801. This work was done in part while Bellare 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. We thank Stefano Tessaro and Arno Mittelbach for discussions and insights. Extensive and insightful comments by the TCC 2016-A reviewers lead to considerable changes and additions to the paper including Theorem 2, Corollary 7 and Section 6.

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`.
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, Heidelberg, Aug. 2001.
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, Heidelberg, Aug. 2007.
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, Heidelberg, Aug. 2008.
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, Heidelberg, Apr. 2015.

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.
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, Heidelberg, May / June 2006.
8. M. Bellare and I. Stepanovs. Point-function obfuscation: A framework and generic constructions. Cryptology ePrint Archive, Report 2015/703, 2015. `http://eprint.iacr.org/2015/703`.
9. 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, Heidelberg, Dec. 2014.
10. M. Bellare, I. Stepanovs, and S. Tessaro. Contention in cryptoland: Obfuscation, leakage and uce. In *Theory of Cryptography, TCC 2016-A*. Springer, 2016.
11. 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, Heidelberg, Aug. 2010.
12. 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, Heidelberg, Aug. 2014.
13. 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.
14. 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, Heidelberg, Mar. 2012.
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, Heidelberg, Aug. 2008.
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, Heidelberg, Feb. 2014.
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, Heidelberg, Aug. 2011.
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, Heidelberg, Aug. 2014.
19. C. Brzuska, P. Farshim, and A. Mittelbach. Random-oracle uninstantiability from indistinguishability obfuscation. In Y. Dodis and J. B. Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 428–455. Springer, Heidelberg, Mar. 2015.
20. 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, Heidelberg, Dec. 2014.
21. 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, Heidelberg, Dec. 2014.

22. 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.
23. 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, Heidelberg, Aug. 1997.
24. 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, Heidelberg, July 2008.
25. 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, Heidelberg, Apr. 2008.
26. 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, Heidelberg, Feb. 2010.
27. 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.
28. 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, Heidelberg, Apr. 2015.
29. 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.
30. Y. Dodis and A. Smith. Entropic security and the encryption of high entropy messages. In J. Kilian, editor, *TCC 2005*, volume 3378 of *LNCS*, pages 556–577. Springer, Heidelberg, Feb. 2005.
31. 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, Heidelberg, May 2010.
32. 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.
33. 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.
34. 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.
35. 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.
36. V. Goyal, A. O'Neill, and V. Rao. Correlated-input secure hash functions. In Y. Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 182–200. Springer, Heidelberg, Mar. 2011.
37. 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, Heidelberg, Dec. 2013.

38. V. Koppula, O. Pandey, Y. Rouselakis, and B. Waters. Deterministic public-key encryption under continual leakage. Cryptology ePrint Archive, Report 2014/780, 2014. http://eprint.iacr.org/2014/780.
39. 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, Heidelberg, May 2004.
40. 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, Heidelberg, Feb. 2014.
41. T. Matsuda and G. Hanaoka. Chosen ciphertext security via UCE. In H. Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 56–76. Springer, Heidelberg, Mar. 2014.
42. I. Mironov, O. Pandey, O. Reingold, and G. Segev. Incremental deterministic public-key encryption. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 628–644. Springer, Heidelberg, Apr. 2012.
43. 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, Heidelberg, Aug. 2014.
44. 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.
45. A. Raghunathan, G. Segev, and S. P. Vadhan. Deterministic public-key encryption for adaptively chosen plaintext distributions. In T. Johansson and P. Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 93–110. Springer, Heidelberg, May 2013.
46. 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.
47. H. Wee. On obfuscating point functions. In H. N. Gabow and R. Fagin, editors, *37th ACM STOC*, pages 523–532. ACM Press, May 2005.
48. 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, Heidelberg, Apr. 2012.
49. D. Wichs. Barriers in cryptography with weak, correlated and leaky sources. In R. D. Kleinberg, editor, *ITCS 2013*, pages 111–126. ACM, Jan. 2013.
50. 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, Heidelberg, Sept. 2012.
51. 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, Heidelberg, Nov. 2013.