

MIT Open Access Articles

The Impossibility of Obfuscation with Auxiliary Input or a Universal Simulator

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

Citation: Bitansky, Nir, Canetti, Ran, Cohn, Henry, Goldwasser, Shafi, Kalai, Yael Tauman et al. 2014. "The Impossibility of Obfuscation with Auxiliary Input or a Universal Simulator."

As Published: 10.1007/978-3-662-44381-1_5

Publisher: Springer Nature

Persistent URL: <https://hdl.handle.net/1721.1/137558>

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Terms of use: Creative Commons Attribution-Noncommercial-Share Alike



The Impossibility of Obfuscation with Auxiliary Input or a Universal Simulator

Nir Bitansky* Ran Canetti† Henry Cohn‡ Shafi Goldwasser§
Yael Tauman Kalai¶ Omer Paneth|| Alon Rosen**

February 9, 2014

Abstract

In this paper we show that the existence of general indistinguishability obfuscators conjectured in a few recent works implies, somewhat counterintuitively, strong impossibility results for virtual black box obfuscation. In particular, we show that indistinguishability obfuscation for all circuits implies:

- The impossibility of average-case virtual black box obfuscation with auxiliary input for any circuit family with super-polynomial pseudo-entropy. Such circuit families include all pseudo-random function families, and all families of encryption algorithms and randomized digital signatures that generate their required coin flips pseudo-randomly. Impossibility holds even when the auxiliary input depends only on the public circuit family, and not the specific circuit in the family being obfuscated.
- The impossibility of average-case virtual black box obfuscation with a universal simulator (with or without any auxiliary input) for any circuit family with super-polynomial pseudo-entropy.

These bounds significantly strengthen the impossibility results of Goldwasser and Kalai (STOC 2005).

*Tel Aviv University, nirbitan@tau.ac.il. Supported by an IBM Ph.D. Fellowship, and the Check Point Institute for Information Security.

†Boston University and Tel Aviv University, canetti@bu.edu. Supported by the Check Point Institute for Information Security, an NSF EAGER grant, and an NSF Algorithmic Foundations grant 1218461.

‡Microsoft Research, One Memorial Drive, Cambridge, MA 02142, cohn@microsoft.com.

§MIT and the Weizmann Institute of Science, shafi@theory.csail.mit.edu.

¶Microsoft Research, One Memorial Drive, Cambridge, MA 02142, yael@microsoft.com.

||Boston University, omer@bu.edu. Supported by the Simons award for graduate students in theoretical computer science and an NSF Algorithmic foundations grant 1218461.

**Efi Arazi School of Computer Science, IDC Herzliya, Israel, alon.rosen@idc.ac.il. Supported by ISF grant no. 1255/12 and by the ERC under the EU's Seventh Framework Programme (FP/2007-2013) ERC Grant Agreement n. 307952.

1 Introduction

The study of *program obfuscation*—a method that transforms a program (say, described as a Boolean circuit) into a form that is executable, but otherwise completely unintelligible—has been a longstanding research direction in cryptography. It was formalized by Barak et al. [BGI⁺01], who formulated a number of security notions for this task. The strongest and most applicable of these notions is *virtual black box (VBB) obfuscation*, which requires that any adversary trying to learn information from an obfuscated program cannot do better than a simulator that is given only black-box access to the program. Barak et al. constructed contrived function families that cannot be VBB obfuscated, thus ruling out a universal obfuscator, but they left open the possibility that large classes of programs might still be obfuscated. Subsequently, VBB obfuscators were produced only for a number of restricted (and mostly simple) classes of programs [Can97, CD08, CRV10, BR13a, BBC⁺14]. To date, the classification of which programs can or cannot be VBB obfuscated is still not well understood.

In contrast, recent progress for more relaxed notions of obfuscation suggests a much more positive picture: Garg et al. [GGH⁺13] proposed a candidate construction for *indistinguishability obfuscation* for *all* circuits. This notion requires only that it is hard to distinguish an obfuscation of C_0 from an obfuscation of C_1 , where C_0 and C_1 are circuits of the same size that compute the same function [BGI⁺01]. Indeed, unlike the case of VBB obfuscation, there are no known impossibility theorems for indistinguishability obfuscation. Furthermore, the Garg et al. construction and variants thereof were shown to satisfy the VBB guarantee in ideal algebraic oracle models [CV13, BR13b, BGK⁺13], although these results have not proved useful so far in achieving VBB obfuscation in the standard model of computation.

Although indistinguishability obfuscation might initially sound arcane, it is surprisingly powerful. For example, it amounts to *best possible* obfuscation [GR07], in the sense that anything that can be hidden by some obfuscator will be hidden by every indistinguishability obfuscator. Subsequent to [GGH⁺13], a flood of results have appeared showing that indistinguishability obfuscation suffices for many applications, such as the construction of public-key encryption from private-key encryption, the existence of deniable encryption, the existence of multi-input functional encryption, and more [SW13, GGH⁺13, HSW13, GGJS13].

Still, for many program classes the meaningfulness and applicability of indistinguishability obfuscation is unclear. Thus, understanding which classes of programs are VBB obfuscatable remains of central importance. Aiming towards such a characterization, Goldwasser and Kalai [GK05] proved strong limitations on VBB obfuscation for a broad class of *pseudo-entropic programs*, including many cryptographic functions, such as pseudo-random functions and certain natural instances of encryption and signatures. They showed the impossibility of a form of VBB security with respect to adversaries that have some a priori *auxiliary information*. When the auxiliary information depends on the actual obfuscated program, they showed that no class of pseudo-entropic functions can be obfuscated, assuming VBB obfuscation for a simple class of *point-filter functions*. For auxiliary information that depends only on the class of programs to be obfuscated, they gave an unconditional result, but only for a restricted class of programs (those that evaluate *NP-filter functions*).

This work in a nutshell. We strengthen the known impossibility results for VBB obfuscation with auxiliary input, and we suggest a different, compelling interpretation of auxiliary-input obfuscation. In a somewhat strange twist, our negative results on VBB obfuscation are based on the existence of indistinguishability obfuscation, which is typically viewed positively. Specifically:

- We weaken the conditions for the impossibility of *dependent* auxiliary-input VBB obfuscation to *witness encryption*, which in turn follows from *indistinguishability obfuscation*.
- We extend the impossibility of *independent* auxiliary-input VBB obfuscation to *all* pseudo-entropic

functions, assuming *indistinguishability obfuscation*.

- We observe that auxiliary-input VBB obfuscation is equivalent to a very natural formulation of VBB obfuscation with universal simulation. This equivalence provides a clear conceptual argument for the significance of our extended impossibility results.

In the rest of the introduction, we introduce the notion of universal simulation and further discuss the notion of auxiliary-input VBB obfuscation. Then, we provide an overview of the results and sketch the proof techniques involved.

Universal simulators. The definition of VBB obfuscation as proposed by Barak et al. requires that for each PPT adversary A , there exists a PPT simulator S that succeeds in simulating the output of A when A is given the obfuscation $\mathcal{O}(f)$ but S is given only black-box access to f . This definition does not say how hard (or easy) it is to find the corresponding simulator S for a given adversary A . When security with black-box access to the function depends on computational hardness assumptions, this definition leaves open the possibility that the obfuscation could be broken in practice without providing an algorithm that breaks these assumptions.

A stronger and arguably more meaningful definition requires that there exist an efficient transformation from an adversary to its corresponding simulator, or equivalently a *universal* PPT simulator capable of simulating any PPT adversary A given the code of A . We will refer to such a definition as VBB obfuscation with a *universal simulator*.

As we said above, we will show that VBB obfuscation with a universal simulator is impossible for function families with super-polynomial pseudo-entropy if general indistinguishability obfuscation is possible.

Auxiliary input. The definition of VBB security with auxiliary inputs, originally considered in [GK05], is a strengthening of VBB security, which corresponds to a setting in which the adversary may have some additional a priori information.

Allowing auxiliary input is crucial when obfuscation is used together with other components in a larger scheme or protocol. Consider, for example, a zero-knowledge protocol in which one of the prover’s messages to the verifier contains an obfuscated program $\mathcal{O}(f)$. To prove that the protocol is zero-knowledge, we would like to show that every verifier V has a zero-knowledge simulator S_{zk} that can simulate V ’s view of the protocol. Intuitively, S_{zk} would rely on the security of \mathcal{O} by thinking of V as an “obfuscation adversary” that is trying to learn information from $\mathcal{O}(f)$. Such an adversary has an “obfuscation simulator” $S_{\mathcal{O}}$ that can learn the same information given only black-box access to f , and S_{zk} can try to use $S_{\mathcal{O}}$. The problem is that the view of V does not depend only on the code of V , but also on auxiliary input to V , such as other prover messages and the statement being proven. An obfuscation definition that does not allow auxiliary input is insufficient to handle this case.

The problem can be avoided by using a definition that guarantees the existence of an obfuscation simulator that can simulate the view of V given any auxiliary input. If the obfuscated program f depends on other prover messages or on the statement, then we require security with respect to *dependent* auxiliary input. Otherwise *independent* auxiliary input suffices. The paper [GK05] considered both of these notions. In the case of dependent auxiliary input, the virtual black box property is required to hold even when the auxiliary input given to the adversary and simulator depends on the actual, secret circuit being obfuscated. In the case of independent auxiliary input, this requirement is weakened: the auxiliary input may depend only on the family of circuits, which is public. The actual circuit to be obfuscated is chosen randomly from the family, independently of the auxiliary input given to the adversary and simulator.

More precisely, an obfuscator \mathcal{O} for a function family \mathcal{F} is (worst-case) VBB secure with *dependent* auxiliary inputs if for every probabilistic polynomial-time (PPT) adversary A , there exists a PPT simulator S

such that for every $f \in \mathcal{F}$ and every auxiliary input aux (which may depend on the function f), the output of $A(\mathcal{O}(f), \text{aux}(f))$ is computationally indistinguishable from $S^f(\text{aux}(f))$. The average-case analogue of this definition requires that the output of $A(\mathcal{O}(f), \text{aux}(f))$ be computationally indistinguishable from $S^f(\text{aux}(f))$ for a *random* function $f \leftarrow \mathcal{F}$.

VBB security with *independent* auxiliary inputs is defined only with respect to an average-case definition.¹ An obfuscator \mathcal{O} for a function family \mathcal{F} is average-case VBB secure with *independent* auxiliary inputs if for every PPT adversary A , there exists a PPT simulator S such that for every auxiliary input aux and for a random $f \leftarrow \mathcal{F}$, the output of $A(\mathcal{O}(f), \text{aux})$ is computationally indistinguishable from $S^f(\text{aux})$.

For the case of dependent auxiliary input, Goldwasser and Kalai [GK05] showed that functions with super-polynomial pseudo-entropy cannot be VBB obfuscated, assuming that a different class of *point filter functions* can be VBB obfuscated. For the weaker notion of VBB obfuscation with independent auxiliary input, they showed a more restricted impossibility result for a subclass of functions called *filter functions*. Our results extend these theorems, assuming indistinguishability obfuscators exist.

1.1 Overview of results and techniques

First we prove that VBB security with a universal simulator is equivalent to VBB security with auxiliary inputs, which is the obfuscation version of the known equivalence for zero-knowledge proofs [Ore87]. More specifically, we consider both *worst-case* VBB security and *average-case* VBB security. In the former the simulator is required to successfully simulate the output of A for every function in the family \mathcal{F} , whereas in the latter the simulator is required to successfully simulate the output of A only for a *random* function in the family.

We prove that worst-case VBB security with a universal simulator is equivalent to worst-case VBB security with *dependent* auxiliary inputs, and that average-case VBB security with a universal simulator is equivalent to average-case VBB security with *independent* auxiliary inputs. To be consistent with the literature, when we refer to VBB security we always consider the worst-case version. When we would like to consider the average-case version we refer to it as average-case VBB.

Informal Lemma 1. *A candidate obfuscator is a (worst-case) VBB obfuscator with a universal simulator for a class of functions \mathcal{F} if and only if it is a (worst-case) VBB obfuscator for \mathcal{F} with dependent auxiliary inputs.*

Informal Lemma 2. *A candidate obfuscator is an average-case VBB obfuscator with a universal simulator for a class of functions \mathcal{F} if and only if it is an average-case VBB obfuscator for \mathcal{F} with independent auxiliary inputs.*

We state and prove these results as Lemmas 3.1 and 3.2 in Section 3.

The above two lemmas imply that in order to obtain negative results for VBB obfuscation with a universal simulator, it suffices to obtain negative results for VBB obfuscation with auxiliary inputs.

New impossibility results. We show that indistinguishability obfuscation implies that any function family with super-polynomial pseudo-entropy *cannot* be VBB obfuscated with auxiliary input. Loosely speaking, a function family \mathcal{F} has super-polynomial pseudo-entropy if it is difficult to distinguish a genuine function in \mathcal{F} from one that has been randomly modified in some locations: for every polynomial p there exists a polynomial-size set I of inputs such that no efficient adversary can distinguish between a random function $f \leftarrow \mathcal{F}$ and such a function with its values on I replaced with another random variable with min-entropy p . We refer the reader to Definition 2.7 for the precise definition, but note that such families include all pseudo-random function families. They also include all semantically secure secret-key or public-key encryption

¹It is not clear how to enforce that the auxiliary input is independent of the function in a worst-case definition.

schemes or secure digital signature schemes, provided that the randomness is generated by using a (secret) pseudo-random function. (See Claim 4.0.1 in [GK05].)

Recently, the notion of witness encryption was put forth by Garg et al. [GGSW13]. It was observed by Goldwasser et al. [GKP⁺13] that an extractable version of witness encryption can be used to obfuscate the class of point-filter functions with respect to dependent auxiliary inputs. Thus, together with [GK05], this shows that the existence of an extractable witness encryption scheme implies that *any* function with super-polynomial pseudo-entropy cannot be obfuscated with respect to dependent auxiliary inputs.

Here we show that the proof of [GK05] actually implies that witness encryption, *without* the extractability property, suffices to prove that all functions with super-polynomial pseudo-entropy are not obfuscatable with respect to dependent auxiliary inputs.

Informal Theorem 3. *Assume the existence of a witness encryption scheme. Then no function family with super-polynomial pseudo-entropy has an average-case VBB obfuscator with respect to dependent auxiliary input.*

The idea behind the proof is that functions with high pseudo-entropy cannot be efficiently compressed; i.e., given oracle access to such a function, one cannot produce a small circuit for it. The reason is that functions with genuinely high entropy cannot be compressed at all (let alone efficiently), and no efficient algorithm can distinguish them from those with high pseudo-entropy.

Using this observation, the proof works as follows. Suppose we wish to construct an obfuscation $\mathcal{O}(f)$ of a function f that has high pseudo-entropy on a polynomial-size set I of inputs. We use witness encryption to encrypt a random bit b so that it can be read only by someone who knows a circuit of size at most $|\mathcal{O}(f)|$ for the values of f on I . Given this encryption of b as auxiliary input, knowledge of the circuit $\mathcal{O}(f)$ suffices to decrypt b . However, black-box access to f is not enough to produce any small circuit, and so VBB security is violated.

We note that this theorem is true in the strong sense: for *any* secret predicate $\pi(f)$ that is not learnable from black-box access to f , there exists an adversary and auxiliary input $\text{aux}(f)$ such that given $\mathcal{O}(f)$ and $\text{aux}(f)$, the adversary efficiently recovers $\pi(f)$, whereas given $\text{aux}(f)$ and oracle access to f , it is computationally hard to recover $\pi(f)$. Moreover, the theorem holds even if we restrict $\text{aux}(f)$ to be an efficiently computable function of f .

It was shown by Garg et al. [GGSW13] (using different terminology) that indistinguishability obfuscation for point-filter functions implies the existence of witness encryption. Thus, the informal theorem above can be restated as follows: assuming the existence of indistinguishability obfuscation for point-filter functions, functions with super-polynomial pseudo-entropy are not average-case VBB obfuscatable with respect to dependent auxiliary inputs.

For independent auxiliary input, we use of a different hypothesis, namely indistinguishability obfuscation for *puncturable pseudo-random functions* (see Definition 2.6). Roughly speaking, these are pseudo-random functions for which we can produce alternate keys that effectively randomize the output for a specified input while leaving the rest of the function unchanged.

Informal Theorem 4. *Assume the existence of indistinguishability obfuscation for a class of puncturable pseudo-random functions. Then no function family with super-polynomial pseudo-entropy has an average-case VBB obfuscator with respect to independent auxiliary input.*

The proof of this theorem is a little more subtle than the previous proof. Suppose we are trying to obfuscate a circuit family with high pseudo-entropy on a set I of inputs. The auxiliary input will be $i\mathcal{O}(K_s)$, where $i\mathcal{O}$ denotes indistinguishability obfuscation and K_s is a circuit that takes another circuit \tilde{C} as input and applies a puncturable pseudo-random function G_s to the values $\tilde{C}(I)$ of \tilde{C} on I . Here, s is a random key.

Now, let $\mathcal{O}(C)$ be a candidate obfuscation of a circuit C . By definition, applying the auxiliary circuit $i\mathcal{O}(K_s)$ to $\mathcal{O}(C)$ yields $K_s(C)$ (i.e., $G_s(C(I))$), but we will show that $K_s(C)$ cannot be computed using only black-box access to C . If it could, then we could replace the C oracle with suitable random values Y on I and still get the answer $G_s(Y)$, by the definition of pseudo-entropy. Then we could modify the auxiliary input to be $i\mathcal{O}(K_s^*)$, where the pseudo-random function in K_s^* has been punctured to randomize its value at Y . The reason this modification is allowable is that with high probability, K_s and K_s^* define the same function (Y has entropy too high to be compressible to any small circuit, so no input \tilde{C} to K_s^* will ever satisfy $\tilde{C}(I) = Y$). Thus, $i\mathcal{O}(K_s)$ and $i\mathcal{O}(K_s^*)$ are indistinguishable. However, by construction K_s^* does not determine the value $G_s(Y)$, which is a contradiction.

We state and prove these results more formally as Theorems 4.1 and 4.2. Together with Lemmas 3.1 and 3.2, they immediately yield impossibility results for VBB obfuscation with a universal simulator. In particular, Theorem 4.1 and Lemma 3.1 imply the following corollary.

Corollary 1. *Assume the existence of a witness encryption scheme. Then no function family with super-polynomial pseudo-entropy has a VBB obfuscator with a universal simulator.*

As was the case for Theorem 4.1, this corollary is true in the strong sense: for *any* secret predicate $\pi(f)$ that is not learnable from black-box access to f , there exists an adversary that efficiently recovers $\pi(f)$ given $\mathcal{O}(f)$, whereas given the code of the adversary and given oracle access to f , it is computationally hard to recover $\pi(f)$.

Theorem 4.2 and Lemma 3.2 imply the following corollary.

Corollary 2. *Assume the existence of indistinguishability obfuscation for a class of puncturable pseudo-random functions. Then no function family with super-polynomial pseudo-entropy has an average-case VBB obfuscator with a universal simulator.*

2 Preliminaries

Let $\mathcal{F} = \{f_s\}$ be a family of polynomial-size circuits. In what follows, we write $\mathcal{F} = \bigcup_{k \in \mathbb{N}} \mathcal{F}_k$ with $\mathcal{F}_k = \{f_s\}_{s \in \{0,1\}^k}$. Each circuit f_s will have size $\text{poly}(|s|)$, where poly denotes an unspecified, polynomially-bounded function.

Definition 2.1 (VBB obfuscation with universal simulator). *Let $\mathcal{F} = \{f_s\}$ be a family of polynomial-size circuits. We say that a probabilistic algorithm \mathcal{O} (mapping circuits to circuits) is an obfuscation of \mathcal{F} with a universal simulator if the following conditions hold:*

- **Correctness:** For every function $f_s \in \mathcal{F}$ and every possible input x ,

$$\mathcal{O}(f_s)(x) = f_s(x).$$

I.e., the random variable $\mathcal{O}(f_s)$ defines the same function as f_s with probability 1.

- **Polynomial slowdown:** There exists a polynomial p such that for every $f_s \in \mathcal{F}$,

$$|\mathcal{O}(f_s)| \leq p(|f_s|).$$

- **Security with a universal simulator:** There exists a (possibly non-uniform) PPT S such that for every (possibly non-uniform) PPT A , every predicate π , every $k \in \mathbb{N}$, and every $s \in \{0,1\}^k$,

$$\left| \Pr[A(\mathcal{O}(f_s)) = \pi(s)] - \Pr[S^{f_s}(A) = \pi(s)] \right| = \text{negl}(k), \quad (1)$$

where the probabilities are over the random coin tosses of A and S . Here $\text{negl}(k)$ denotes an unspecified, negligible function (i.e., $|\text{negl}(k)| = O(1/k^c)$ for each constant $c > 0$).

We say that \mathcal{O} is an **average-case** obfuscation of \mathcal{F} with a universal simulator if Equation (1) holds for **random** $s \leftarrow \{0, 1\}^k$; in other words, it means there exists a (possibly non-uniform) PPT S such that for every (possibly non-uniform) PPT A , every predicate π , and every $k \in \mathbb{N}$,

$$\left| \Pr[A(\mathcal{O}(f_s)) = \pi(s)] - \Pr[S^{f_s}(A) = \pi(s)] \right| = \text{negl}(k),$$

where the probabilities are over $s \leftarrow \{0, 1\}^k$ and over the random coin tosses of A and S .

Note that we do not assume $\mathcal{O}(f_s)$ can be efficiently computed given f_s . Our negative results rule out the existence of obfuscations, and not merely the possibility of finding them.

When A is non-uniform, the notation $S^{f_s}(A)$ of course means that S is given a circuit for A for inputs of the appropriate size. When A is uniform, it means the same thing as in the non-uniform case; equivalently, S is given the code for A together with $1^{\text{time}(A(\mathcal{O}(f_s)))}$ to ensure that it is allowed enough time.

In Definition 2.1, we have conflated the circuit size parameter k and the security parameter of the obfuscation method. One could distinguish between them at the cost of more notation, but this conflation is of course harmless for proving impossibility theorems.

Definition 2.2 (VBB obfuscation with auxiliary inputs). Let $\mathcal{F} = \{f_s\}$ be a family of polynomial-size circuits. We say that a probabilistic algorithm \mathcal{O} is an obfuscation of \mathcal{F} with (dependent) auxiliary inputs if it satisfies the correctness and polynomial slowdown conditions of Definition 2.1, and in addition it satisfies the following security requirement:

- **Security with auxiliary inputs:** For every (possibly non-uniform) PPT A , there exists a (possibly non-uniform) PPT S such that for every predicate π , every $k \in \mathbb{N}$, every $s \in \{0, 1\}^k$, and every auxiliary input $\text{aux}(s)$ of size $\text{poly}(k)$,

$$\left| \Pr[A(\mathcal{O}(f_s), \text{aux}(s)) = \pi(s, \text{aux}(s))] - \Pr[S^{f_s}(\text{aux}(s)) = \pi(s, \text{aux}(s))] \right| = \text{negl}(k), \quad (2)$$

where the probabilities are over the random coin tosses of A and S . We write $\text{aux}(s)$ as a function of s for clarity, but this is not strictly necessary since the quantification automatically allows dependence on s .

We say that \mathcal{O} is an **average-case** obfuscation of \mathcal{F} with (dependent) auxiliary inputs if Equation (2) holds for **random** $s \leftarrow \{0, 1\}^k$; namely, if for every (possibly non-uniform) PPT A there exists a (possibly non-uniform) PPT S such that for every predicate π , every $k \in \mathbb{N}$, and every auxiliary input $\text{aux}(s)$ of size $\text{poly}(s)$ (and allowed to depend on s),

$$\left| \Pr[A(\mathcal{O}(f_s), \text{aux}(s)) = \pi(s, \text{aux}(s))] - \Pr[S^{f_s}(\text{aux}(s)) = \pi(s, \text{aux}(s))] \right| = \text{negl}(k),$$

where the probabilities are over $s \leftarrow \{0, 1\}^k$ and over the random coin tosses of A and S .

In the definition above we allowed the auxiliary input to depend on the function being obfuscated. In what follows we define VBB obfuscation with *independent* auxiliary inputs, where we restrict the auxiliary input to be *independent* of the function being obfuscated. For this definition, only the average-case version makes sense, since in the worst-case version it is not clear how to ensure that the auxiliary input is independent of the function being obfuscated.

Definition 2.3 (Average-case VBB obfuscation with independent auxiliary inputs). Let $\mathcal{F} = \{f_s\}$ be a family of polynomial-size circuits. We say that \mathcal{O} is an obfuscation of \mathcal{F} with independent auxiliary inputs if it satisfies the correctness and polynomial slowdown conditions of Definition 2.1, and in addition it satisfies the following security requirement:

- **Average-case security with independent auxiliary input:** For every (possibly non-uniform) PPT A , there exists a (possibly non-uniform) PPT S such that for every predicate π , every $k \in \mathbb{N}$, and every auxiliary input $\text{aux} \in \{0, 1\}^{\text{poly}(k)}$,

$$\left| \Pr[A(\mathcal{O}(f_s), \text{aux}) = \pi(s, \text{aux})] - \Pr[S^{f_s}(\text{aux}) = \pi(s, \text{aux})] \right| = \text{negl}(k),$$

where the probabilities are over $s \leftarrow \{0, 1\}^k$ and over the random coin tosses of A and S .

Definition 2.4 (Witness encryption). A witness encryption scheme for an NP language \mathcal{L} with corresponding witness relation $\mathcal{R}_{\mathcal{L}}$ is a pair of PPT algorithms (Enc, Dec) such that the following conditions hold:

- **Correctness:** For all $(x, w) \in \mathcal{R}_{\mathcal{L}}$ and every $b \in \{0, 1\}$,

$$\Pr[\text{Dec}(\text{Enc}_x(1^k, b), w) = b] = 1 - \text{negl}(k).$$

- **Semantic Security:** For every $x \notin \mathcal{L}$ and every (possibly non-uniform) PPT adversary A ,

$$\left| \Pr[A(\text{Enc}_x(1^k, 0)) = 1] - \Pr[A(\text{Enc}_x(1^k, 1)) = 1] \right| = \text{negl}(k),$$

where the probability is over the random coin tosses of Enc and A .

Definition 2.5 (Indistinguishability obfuscation). Let \mathcal{C} be a family of polynomial-size circuits. A PPT algorithm $i\mathcal{O}$ is said to be an indistinguishability obfuscator for \mathcal{C} if it satisfies the correctness and polynomial slowdown conditions of Definition 2.1, and in addition it satisfies the following security requirement:

- **Indistinguishability:** For all $C, C' \in \mathcal{C}$ that are of the same size and define the same function, $i\mathcal{O}(C)$ and $i\mathcal{O}(C')$ are computationally indistinguishable. More formally, for every (possibly non-uniform) PPT distinguisher D ,

$$\left| \Pr[D(i\mathcal{O}(C)) = 1] - \Pr[D(i\mathcal{O}(C')) = 1] \right| = \text{negl}(k),$$

where the probability is over the random coin tosses of $i\mathcal{O}$ and D .

Although Definition 2.1 did not require VBB obfuscation to be efficiently computable (to obtain stronger impossibility results), we require $i\mathcal{O}$ to be efficiently computable in Definition 2.5, because inefficient indistinguishability obfuscation is trivial.

We next define puncturable pseudo-random functions. We consider a simple case in which any PRF might be punctured at a single point. The definition is formulated as in [SW13].

Definition 2.6 (Puncturable PRFs). Let ℓ, m be polynomially bounded length functions. An efficiently computable family of functions

$$\mathcal{G} = \left\{ G_s : \{0, 1\}^{m(k)} \rightarrow \{0, 1\}^{\ell(k)} \mid s \in \{0, 1\}^k, k \in \mathbb{N} \right\},$$

associated with an efficient (probabilistic) key sampler $\text{Gen}_{\mathcal{G}}$, is a puncturable PRF if there exists a puncturing algorithm Punc that takes as input a key $s \in \{0, 1\}^k$ and a point $x^* \in \{0, 1\}^{m(k)}$ and outputs a punctured key s_{x^*} so that the following conditions are satisfied:

- **Functionality is preserved under puncturing:** For every $x^* \in \{0, 1\}^{m(k)}$, if we sample s from $\text{Gen}_G(1^k)$ and let $s_{x^*} = \text{Punc}(s, x^*)$, then G_s and $G_{s_{x^*}}$ have the same values at every point other than x^* with probability 1.
- **Indistinguishability at punctured points:** The two ensembles

$$\left\{ (x^*, s_{x^*}, G_s(x^*)) \mid s \leftarrow \text{Gen}_G(1^k), s_{x^*} = \text{Punc}(s, x^*) \right\}_{x^* \in \{0, 1\}^{m(k)}, k \in \mathbb{N}},$$

$$\left\{ (x^*, s_{x^*}, u) \mid s \leftarrow \text{Gen}_G(1^k), s_{x^*} = \text{Punc}(s, x^*), u \leftarrow \{0, 1\}^{\ell(k)} \right\}_{x^* \in \{0, 1\}^{m(k)}, k \in \mathbb{N}}$$

are computationally indistinguishable by (possibly non-uniform) PPT distinguishers.

To be explicit, we include x^* in the distribution; throughout, we shall assume for simplicity that a punctured key s_{x^*} includes x^* in the clear. As shown in [BGI13, BW13, KPTZ13], the pseudo-random functions from [GGM86] yield puncturable PRFs as defined above.

Definition 2.7 (Pseudo-entropy of a circuit class). Let $p = p(k)$ be a polynomial. We say that a class of circuits $\mathcal{C} = \bigcup_{k \in \mathbb{N}} \mathcal{C}_k$ has pseudo-entropy at least $p = p(k)$, if there exists a polynomial $t = t(k)$ and a subset $I_k \subseteq \{0, 1\}^k$ of size $t(k)$, and for every $C \in \mathcal{C}_k$ there exists a random variable $Y^C = (Y_i)_{i \in I_k} \in \{0, 1\}^{I_k}$, such that the following conditions hold:

1. The random variable Y^C has statistical min-entropy at least $p(k)$. In other words, each of its values occurs with probability at most $2^{-p(k)}$.
2. For every (possibly non-uniform) PPT distinguisher D ,

$$\left| \Pr[D^C(1^k) = 1] - \Pr[D^{C \circ Y^C}(1^k) = 1] \right| = \text{negl}(k),$$

where $C \circ Y^C$ denotes an oracle that agrees with C except that Y^C replaces the values of C for inputs in I_k . Here the probabilities are over $C \leftarrow \mathcal{C}_k$, the random variable Y^C , and the random coin tosses of D .

We say that \mathcal{C} has super-polynomial pseudo-entropy if it has pseudo-entropy at least p for every polynomial p , and we then call the circuits in \mathcal{C} pseudo-entropic.

3 Equivalence between a universal simulator and auxiliary inputs

In this section we show that VBB obfuscation with a universal simulator is equivalent to VBB obfuscation with auxiliary inputs. Specifically, we prove the following two lemmas.

Lemma 3.1. Let $\mathcal{F} = \{f_s\}$ be a family of polynomial-size circuits. Then \mathcal{O} is a VBB obfuscator for \mathcal{F} with a universal simulator if and only if it is a VBB obfuscator for \mathcal{F} with dependent auxiliary inputs.

Lemma 3.2. Let $\mathcal{F} = \{f_s\}$ be a family of polynomial-size circuits. Then \mathcal{O} is an average-case VBB obfuscator for \mathcal{F} with a universal simulator if and only if it is an average-case VBB obfuscator for \mathcal{F} with independent auxiliary inputs.

Proof of Lemma 3.1.

(\Rightarrow): Suppose that \mathcal{O} is a VBB obfuscator for \mathcal{F} with a universal simulator. Namely, there exists a (possibly non-uniform) PPT S such that for every (possibly non-uniform) PPT A , every predicate π , every $k \in \mathbb{N}$ and every $s \in \{0, 1\}^k$,

$$\left| \Pr[A(\mathcal{O}(f_s)) = \pi(s)] - \Pr[S^{f_s}(A) = \pi(s)] \right| = \text{negl}(k),$$

where the probabilities are over the random coin tosses of A and S .

We will prove that \mathcal{O} is a VBB obfuscator for \mathcal{F} with dependent auxiliary inputs. To this end, fix any (possibly non-uniform) PPT adversary A . Let S_A be the PPT simulator defined as follows: for every auxiliary input $\text{aux}(s)$, $S_A^{f_s}(\text{aux}(s))$ runs the universal simulator S^{f_s} on input $A_{\text{aux}(s)}$, where $A_{\text{aux}(s)}$ is the (non-uniform) adversary that simulates A with auxiliary input $\text{aux}(s)$. We need to prove that for every predicate π , every $k \in \mathbb{N}$, and every $s \in \{0, 1\}^k$,

$$\left| \Pr[A(\mathcal{O}(f_s), \text{aux}(s)) = \pi(s, \text{aux}(s))] - \Pr[S_A^{f_s}(\text{aux}(s)) = \pi(s, \text{aux}(s))] \right| = \text{negl}(k),$$

where the probabilities are over the random coin tosses of A and S .

To do so, we check that

$$\begin{aligned} & \left| \Pr[A(\mathcal{O}(f_s), \text{aux}(s)) = \pi(s, \text{aux}(s))] - \Pr[S_A^{f_s}(\text{aux}(s)) = \pi(s, \text{aux}(s))] \right| \\ &= \left| \Pr[A(\mathcal{O}(f_s), \text{aux}(s)) = \pi(s, \text{aux}(s))] - \Pr[S^{f_s}(A_{\text{aux}(s)}) = \pi(s, \text{aux}(s))] \right| \\ &\leq \left| \Pr[A(\mathcal{O}(f_s), \text{aux}(s)) = \pi(s, \text{aux}(s))] - \Pr[A_{\text{aux}(s)}(\mathcal{O}(f_s)) = \pi(s, \text{aux}(s))] \right| \\ &\quad + \left| \Pr[A_{\text{aux}(s)}(\mathcal{O}(f_s)) = \pi(s, \text{aux}(s))] - \Pr[S^{f_s}(A_{\text{aux}(s)}) = \pi(s, \text{aux}(s))] \right| \\ &= \text{negl}(k), \end{aligned}$$

where the first equation follows by the definition of S_A , the inequality follows from the triangle inequality, and the last equation follows from the definition of $A_{\text{aux}(s)}$ and from the fact that \mathcal{O} is VBB secure with the universal simulator S .

(\Leftarrow): Suppose that \mathcal{O} is a VBB obfuscator for \mathcal{F} with dependent auxiliary inputs. Namely, for every (possibly non-uniform) PPT A there exists a (possibly non-uniform) PPT S such that for every predicate π , every $k \in \mathbb{N}$, every $s \in \{0, 1\}^k$, and every auxiliary input $\text{aux}(s)$ of size $\text{poly}(k)$,

$$\left| \Pr[A(\mathcal{O}(f_s), \text{aux}(s)) = \pi(s, \text{aux}(s))] - \Pr[S^{f_s}(\text{aux}(s)) = \pi(s, \text{aux}(s))] \right| = \text{negl}(k),$$

where the probabilities are over the random coin tosses of A and S . We prove that \mathcal{O} is a VBB obfuscator for \mathcal{F} with a universal simulator. To this end, let A^* be a universal PPT adversary that interprets its auxiliary input $\text{aux} = \text{aux}(s)$ as a (possibly non-uniform) PPT adversary and runs this adversary. (As pointed out after Definition 2.1, we must interpret this carefully regarding running times in the uniform case.) The fact that \mathcal{O} is a VBB obfuscator with dependent auxiliary inputs implies that there is a PPT simulator S such that for every predicate π , every $k \in \mathbb{N}$, every $s \in \{0, 1\}^k$, and every auxiliary input $\text{aux}(s)$ of size $\text{poly}(k)$,

$$\left| \Pr[A^*(\mathcal{O}(f_s), \text{aux}(s)) = \pi(s, \text{aux}(s))] - \Pr[S^{f_s}(\text{aux}(s)) = \pi(s, \text{aux}(s))] \right| = \text{negl}(k), \quad (3)$$

where the probabilities are over the random coin tosses of A^* and S . We claim that S is a universal simulator for \mathcal{O} . Namely, we claim that for every (possibly non-uniform) PPT adversary A , every predicate π , every $k \in \mathbb{N}$, and every $s \in \{0, 1\}^k$,

$$\left| \Pr[A(\mathcal{O}(f_s)) = \pi(s)] - \Pr[S^{f_s}(A) = \pi(s)] \right| = \text{negl}(k).$$

To see why, note that

$$\begin{aligned} & \left| \Pr[A(\mathcal{O}(f_s)) = \pi(s)] - \Pr[S^{f_s}(A) = \pi(s)] \right| \\ & \leq \left| \Pr[A(\mathcal{O}(f_s)) = \pi(s)] - \Pr[A^*(\mathcal{O}(f_s), A) = \pi(s)] \right| \\ & \quad + \left| \Pr[A^*(\mathcal{O}(f_s), A) = \pi(s)] - \Pr[S^{f_s}(A) = \pi(s)] \right| \\ & = \left| \Pr[A^*(\mathcal{O}(f_s), A) = \pi(s)] - \Pr[S^{f_s}(A) = \pi(s)] \right| \\ & = \text{negl}(k), \end{aligned}$$

where the inequality follows from the triangle inequality, the next equation follows from the definition of A^* , and the last equation follows from Equation (3). \square

Proof of Lemma 3.2.

(\Rightarrow): Suppose that \mathcal{O} is an average-case VBB obfuscator for \mathcal{F} with a universal simulator. Namely, there exists a (possibly non-uniform) PPT S such that for every (possibly non-uniform) PPT A , every predicate π , and every $k \in \mathbb{N}$,

$$\left| \Pr[A(\mathcal{O}(f_s)) = \pi(s)] - \Pr[S^{f_s}(A) = \pi(s)] \right| = \text{negl}(k),$$

where the probabilities are over $s \leftarrow \{0, 1\}^k$ and over the random coin tosses of A and S .

We will prove that \mathcal{O} is an average-case VBB obfuscator for \mathcal{F} with independent auxiliary inputs. To this end, fix any (possibly non-uniform) PPT adversary A . Let S_A be the PPT simulator defined as follows: for every auxiliary input aux , $S_A^{f_s}(\text{aux})$ runs the universal simulator S^{f_s} on input A_{aux} , where A_{aux} is the (non-uniform) adversary that simulates A with auxiliary input aux . We need to prove that for every predicate π , every $k \in \mathbb{N}$, and every $\text{aux} \in \{0, 1\}^{\text{poly}(k)}$,

$$\left| \Pr[A(\mathcal{O}(f_s), \text{aux}) = \pi(s, \text{aux})] - \Pr[S_A^{f_s}(\text{aux}) = \pi(s, \text{aux})] \right| = \text{negl}(k),$$

where the probabilities are over $s \leftarrow \{0, 1\}^k$ and over the random coin tosses of A and S . To see why this is true, note that

$$\begin{aligned} & \left| \Pr[A(\mathcal{O}(f_s), \text{aux}) = \pi(s, \text{aux})] - \Pr[S_A^{f_s}(\text{aux}) = \pi(s, \text{aux})] \right| \\ & = \left| \Pr[A(\mathcal{O}(f_s), \text{aux}) = \pi(s, \text{aux})] - \Pr[S^{f_s}(A_{\text{aux}}) = \pi(s, \text{aux})] \right| \\ & \leq \left| \Pr[A(\mathcal{O}(f_s), \text{aux}) = \pi(s, \text{aux})] - \Pr[A_{\text{aux}}(\mathcal{O}(f_s)) = \pi(s, \text{aux})] \right| \\ & \quad + \left| \Pr[A_{\text{aux}}(\mathcal{O}(f_s)) = \pi(s, \text{aux})] - \Pr[S^{f_s}(A_{\text{aux}}) = \pi(s, \text{aux})] \right| \\ & = \text{negl}(k), \end{aligned}$$

where the first equation follows from the definition of S_A , the inequality follows from the triangle inequality, and the last equation follows from the definition of A_{aux} and from the fact that \mathcal{O} is average-case VBB secure

with the universal simulator S .

(\Leftarrow): Suppose that \mathcal{O} is an average-case VBB obfuscator for \mathcal{F} with independent auxiliary inputs. Namely, for every (possibly non-uniform) PPT A , there exists a (possibly non-uniform) PPT S such that for every predicate π , every $k \in \mathbb{N}$, and every auxiliary input $\text{aux} \in \{0, 1\}^{\text{poly}(k)}$,

$$\left| \Pr[A(\mathcal{O}(f_s), \text{aux}) = \pi(s, \text{aux})] - \Pr[S^{f_s}(\text{aux}) = \pi(s, \text{aux})] \right| = \text{negl}(k),$$

where the probabilities are over $s \leftarrow \{0, 1\}^k$ and over the random coin tosses of A and S .

We will prove that \mathcal{O} is an average-case VBB obfuscator for \mathcal{F} with a universal simulator. To this end, let A^* be a universal PPT adversary that interprets its auxiliary input aux as a (possibly non-uniform) PPT adversary and runs this adversary, as in the previous proof. The fact that \mathcal{O} is an average-case VBB obfuscator with independent auxiliary inputs implies that there is a PPT simulator S such that for every predicate π , every $k \in \mathbb{N}$, and every auxiliary input $\text{aux} \in \{0, 1\}^{\text{poly}(k)}$,

$$\left| \Pr[A^*(\mathcal{O}(f_s), \text{aux}) = \pi(s, \text{aux})] - \Pr[S^{f_s}(\text{aux}) = \pi(s, \text{aux})] \right| = \text{negl}(k), \quad (4)$$

where the probabilities are over $s \leftarrow \{0, 1\}^k$ and over the random coin tosses of A^* and S . We claim that S is an average-case universal simulator for \mathcal{O} . Namely, we claim that for every (possibly non-uniform) PPT adversary A , every predicate π , and every $k \in \mathbb{N}$,

$$\left| \Pr[A(\mathcal{O}(f_s)) = \pi(s)] - \Pr[S^{f_s}(A) = \pi(s)] \right| = \text{negl}(k),$$

where the probabilities are over $s \leftarrow \{0, 1\}^k$, and over the random coin tosses of A and S .

To see why, note that

$$\begin{aligned} & \left| \Pr[A(\mathcal{O}(f_s)) = \pi(s)] - \Pr[S^{f_s}(A) = \pi(s)] \right| \\ & \leq \left| \Pr[A(\mathcal{O}(f_s)) = \pi(s)] - \Pr[A^*(\mathcal{O}(f_s), A) = \pi(s)] \right| \\ & \quad + \left| \Pr[A^*(\mathcal{O}(f_s), A) = \pi(s)] - \Pr[S^{f_s}(A) = \pi(s)] \right| \\ & = \left| \Pr[A^*(\mathcal{O}(f_s), A) = \pi(s)] - \Pr[S^{f_s}(A) = \pi(s)] \right| \\ & = \text{negl}(k), \end{aligned}$$

where the inequality follows from the triangle inequality, the next equation follows from the definition of A^* , and the last equation follows from Equation (4). □

4 Impossibility for obfuscation with auxiliary inputs

As mentioned in the introduction, Goldwasser and Kalai [GK05] proved that either point-filter functions are not obfuscatable with dependent auxiliary inputs or *all* function families with sufficient pseudo-entropy are not obfuscatable with dependent auxiliary inputs. It was recently observed by Goldwasser et al. [GKP⁺13] that extractable witness encryption implies that point-filter functions are obfuscatable with dependent auxiliary inputs, and thus that any function family with sufficient pseudo-entropy is not obfuscatable with dependent auxiliary inputs. We now show that the same impossibility result (with essentially the same proof as in [GK05]) can be obtained assuming the existence of witness encryption, without any extractability property.

Theorem 4.1. *Assume the existence of a witness encryption scheme for an NP-complete language. Then no function family with super-polynomial pseudo-entropy has an average-case VBB obfuscator with respect to dependent auxiliary input.*

In fact, the proof rules out average-case obfuscation if we restrict the auxiliary input to be efficiently computable given the function (or even oracle access to the function).

Theorem 4.2. *Assume the existence of indistinguishability obfuscation for a class of puncturable pseudo-random functions. Then no function family with super-polynomial pseudo-entropy has an average-case VBB obfuscator with respect to independent auxiliary input.*

We describe the specific class for which we need indistinguishability obfuscation in the proof of the theorem.

Theorems 4.1 and 4.2, together with Lemmas 3.1 and 3.2, immediately yield impossibility results for VBB obfuscation with a universal simulator. In particular, Theorem 4.1 and Lemma 3.1 imply the following corollary.

Corollary 4.3. *Assume the existence of a witness encryption scheme for an NP-complete language. Then no function family with super-polynomial pseudo-entropy has a VBB obfuscator with a universal simulator.*

Theorem 4.2 and Lemma 3.2 imply the following corollary.

Corollary 4.4. *Assume the existence of indistinguishability obfuscation for a class of puncturable pseudo-random functions. Then no function family with super-polynomial pseudo-entropy has an average-case VBB obfuscator with a universal simulator.*

All that remains is to prove Theorems 4.1 and 4.2. For notation in both proofs, let $\mathcal{C} = \bigcup_{k \in \mathbb{N}} \mathcal{C}_k$ be a class of circuits with super-polynomial pseudo-entropy such that each $C \in \mathcal{C}_k$ maps $\{0, 1\}^{\ell(k)}$ to $\{0, 1\}^{\ell'(k)}$. Let \mathcal{O} be any candidate obfuscator for \mathcal{C} , and let $m(k)$ be a polynomial such that $|\mathcal{O}(C)| \leq m(k)$ for every $C \in \mathcal{C}_k$.

4.1 Proof of Theorem 4.1

The fact that \mathcal{C} has super-polynomial pseudo-entropy implies that it has pseudo-entropy at least $m(k) + k$. In particular, recalling Definition 2.7, this implies that there exists a polynomial $t = t(k)$ and a subset $I_k \subseteq \{0, 1\}^k$ of size $t(k)$ such that for every C there exists a random variable $Y^C = (Y_1, \dots, Y_t)$ such that the following conditions hold:

1. The random variable Y^C has statistical min-entropy at least $m(k) + k$.
2. For every (possibly non-uniform) PPT distinguisher D ,

$$\left| \Pr[D^C(1^k) = 1] - \Pr[D^{C \circ Y^C}(1^k) = 1] \right| = \text{negl}(k),$$

where $C \circ Y^C$ denotes an oracle that agrees with C except that Y^C replaces the values of C for inputs in I_k . Here the probabilities are over $C \leftarrow \mathcal{C}_k$, the random variable Y^C , and the random coin tosses of D .

We define an NP language \mathcal{L} by

$$\mathcal{L} = \{(x_i)_{i \in I_k} \mid k \in \mathbb{N} \text{ and there exists a circuit } C \text{ of size } |C| \leq p(k) \text{ such that } C(i) = x_i \text{ for all } i \in I_k\}.$$

Set $x = (C(i))_{i \in I_k}$ and let $\text{aux}(C) = \text{Enc}_x(1^k, b)$, where $b \leftarrow \{0, 1\}$ is a random bit and Enc is a witness encryption for the language \mathcal{L} . Note that the fact that there is a witness encryption for an NP-complete language implies that there is a witness encryption for every NP language, and in particular for \mathcal{L} .

Given $\mathcal{O}(C)$ and $\text{aux}(C) = \text{Enc}_x(1^k, b)$, one can efficiently decrypt b with probability $1 - \text{negl}(k)$, since $\mathcal{O}(C)$ is a valid witness of x . It remains to prove the following claim.

Claim 4.5. *For any (possibly non-uniform) PPT adversary S which takes as input $\text{aux}(s) = \text{Enc}_x(1^k, b)$ and has black-box access to C ,*

$$\Pr[S^C(\text{Enc}_x(1^k, b)) = b] \leq \frac{1}{2} + \text{negl}(k).$$

Proof. Suppose for the sake of contradiction that there exists a PPT adversary S such that

$$\Pr[S^C(\text{Enc}_x(1^k, b)) = b] \geq \frac{1}{2} + \epsilon(k)$$

for some non-negligible function ϵ , where the probability is over random $C \leftarrow \mathcal{C}_k$, the choice of b , and the randomness of Enc .

Let D be the distinguisher that, given oracle access to C , does the following. First, it computes $x = (C(i))_{i \in I_k}$ by querying the oracle $t(k)$ times. Then it computes $\text{Enc}_x(1^k, b)$ and simulates $S^C(\text{Enc}_x(1^k, b))$ to arrive at its output.

By assumption,

$$\Pr[D^C(1^k) = b] \geq \frac{1}{2} + \epsilon(k).$$

Thus, because \mathcal{C} has super-polynomial pseudo-entropy,

$$\Pr[D^{C \circ Y^C}(1^k) = b] \geq \frac{1}{2} + \epsilon(k) + \text{negl}(k). \quad (5)$$

When it is given oracle access to $C \circ Y^C$, D replaces x with $x^* = Y^C$, and at the end it is trying to recover b from $\text{Enc}_{x^*}(1^k, b)$.

Note however that x^* has min-entropy $m(k) + k$, and so the probability that it is in \mathcal{L} is at most 2^{-k} . (For each of the at most $2^{m(k)}$ circuits of size $m(k)$ in the definition of \mathcal{L} , the probability of obtaining x^* is at most $2^{-m(k)-k}$.) Thus, Equation (5) contradicts the semantic security of the underlying witness-encryption scheme. \square

Remark 4.6. Note that for any secret predicate π that is not learnable from black-box access to the circuit, we could have taken the auxiliary input to be $\text{aux}(C) = \text{Enc}_x(1^k, b)$ where $b = \pi(C)$ (as opposed to being truly random). In this case, there exists a PPT adversary A that given the obfuscated circuit $\mathcal{O}(C)$ and the auxiliary input $\text{aux}(C)$ outputs $\pi(C)$ with probability 1, whereas any PPT simulator cannot learn $\pi(C)$ from $\text{aux}(C)$ and black-box access to C .

Using Lemma 3.1, we conclude that for any secret predicate π that is not learnable from black-box access to the circuit and for any circuit C there exists an adversary $A_{\text{aux}(C)}$ that outputs $\pi(C)$ with probability 1, whereas any universal simulator S , which is given black box access to C and takes as input the code of $A_{\text{aux}(C)}$, cannot learn the predicate $\pi(C)$.

Thus our negative result is a strong one: VBB obfuscation with a universal simulator cannot conceal *any* secret predicate that is not learnable from black-box access to the circuit.

4.2 Proof of Theorem 4.2

We first describe an auxiliary-input distribution ensemble \mathcal{Z} and a PPT adversary A such that given $z \leftarrow \mathcal{Z}$ and an obfuscation of $C \leftarrow \mathcal{C}$, A always learns some predicate $\pi(C, z)$. Then, we show that any PPT simulator that is only given oracle access to C fails to learn the predicate.

The auxiliary input distribution \mathcal{Z} . By assumption, \mathcal{C} has pseudo-entropy at least $m(k) + k$. Let $\{I_k\}_{k \in \mathbb{N}}$ be the sets guaranteed by Definition 2.7, where I_k is of polynomial size $t(k)$, and let \mathcal{G} be a puncturable one-bit PRF family

$$\mathcal{G} = \left\{ G_s : \{0, 1\}^{\ell'(k) \cdot t(k)} \rightarrow \{0, 1\} \mid s \in \{0, 1\}^k, k \in \mathbb{N} \right\}.$$

We define two circuit families

$$\begin{aligned} \mathcal{K} &= \left\{ K_s : \{0, 1\}^{m(k)} \rightarrow \{0, 1\} \mid s \in \{0, 1\}^k, k \in \mathbb{N} \right\}, \\ \mathcal{K}^* &= \left\{ K_{s_{x^*}}^* : \{0, 1\}^{m(k)} \rightarrow \{0, 1\} \mid s \in \{0, 1\}^k, x^* \in \{0, 1\}^{\ell'(k) \cdot t(k)}, k \in \mathbb{N} \right\}. \end{aligned}$$

Given a circuit $\tilde{C} : \{0, 1\}^\ell \rightarrow \{0, 1\}^{\ell'}$ of size m as input, the circuit K_s computes $x := \tilde{C}(I_k) := (\tilde{C}(i))_{i \in I_k}$ and outputs $G_s(x)$. See Figure 1.

Hardwired: a PRF key $s \in \{0, 1\}^k$ and the set I_k .

Input: a circuit $\tilde{C} : \{0, 1\}^\ell \rightarrow \{0, 1\}^{\ell'}$, where $|\tilde{C}| = m(k)$.

1. Compute $x = \tilde{C}(I_k)$.
2. Return $G_s(x)$.

Figure 1: The circuit K_s .

The circuit $K_{s_{x^*}}^*$, has a hardwired PRF key s_{x^*} that was derived from s by puncturing it at the point x^* . It operates the same as K_s , except that when $x = x^*$, it outputs an arbitrary bit, say, 0. See Figure 2. In particular, if $x^* \neq \tilde{C}(I_k)$ for all circuits $\tilde{C} \in \{0, 1\}^{m(k)}$, then $K_{s_{x^*}}^*$ and K_s compute the exact same function.

We are now ready to define our auxiliary-input distribution $\mathcal{Z} = \{Z_k\}_{k \in \mathbb{N}}$. Let $d = d(k)$ be the maximal size of circuits in either \mathcal{K} or \mathcal{K}^* , corresponding to security parameter k . Denote by $[K]_d$ a circuit K padded with zeros to size d , and by $[\mathcal{K}]_d$ the class of circuits where every circuit $K \in \mathcal{K}$ is replaced with $[K]_d$. Let $i\mathcal{O}$ be an indistinguishability obfuscator for the class $[\mathcal{K} \cup \mathcal{K}^*]_d$.

The distribution Z_k simply consists of an obfuscated (padded) circuit K_s for a randomly generated s . See Figure 3.

The adversary A and predicate π . The adversary A , given auxiliary input $z = [i\mathcal{O}(K_s)]_{d(k)}$ and an obfuscation $\mathcal{O}(C)$ with $C \in \mathcal{C}_k$, outputs

$$z(\mathcal{O}(C)) = K_s(\mathcal{O}(C)) = G_s(\mathcal{O}(C)(I_k)) = G_s(C(I_k)),$$

where the above follows by the definition of K_s and the functionality of $i\mathcal{O}$ and \mathcal{O} .

Hardwired: a punctured PRF key $s_{x^*} = \text{Punc}(s, x^*)$ and the set I_k .

Input: a circuit $\tilde{C}: \{0, 1\}^\ell \rightarrow \{0, 1\}^{\ell'}$, where $|\tilde{C}| = m(k)$.

1. Compute $x = \tilde{C}(I_k)$.
2. If $x \neq x^*$, return $G_{s_{x^*}}(x)$.
3. If $x = x^*$, return 0.

Figure 2: The circuit $K_{s_{x^*}}^*$.

1. Sample $s \leftarrow \text{Gen}_{\mathcal{G}}(1^k)$.
2. Sample an obfuscation $z \leftarrow i\mathcal{O}([K_s]_{d(k)})$.
3. Output z .

Figure 3: The auxiliary input distribution Z_k .

Thus, A always successfully outputs the predicate

$$\pi(C, K_s) = K_s(C) = G_s(C(I_k)).$$

Adversary A cannot be simulated. We prove the following claim implying that the candidate obfuscator \mathcal{O} for the class \mathcal{C} fails to meet the virtual black box requirement:

Claim 4.7. *For any PPT simulator S ,*

$$\Pr_{\substack{C \leftarrow \mathcal{C}_k \\ z \leftarrow Z_k}} [S^C(z) = \pi(C, z)] \leq \frac{1}{2} + \text{negl}(k).$$

Proof. Assume towards contradiction that there exists a PPT simulator S that learns $\pi(C, z)$ with probability $\frac{1}{2} + \epsilon(k)$, for some non-negligible ϵ . We show how to use S to break either the pseudo-entropy of \mathcal{C} or the pseudo-randomness at punctured points of \mathcal{G} .

According to the definition of Z_k ,

$$\Pr [S^C(i\mathcal{O}([K_s]_d)) = G_s(C(I_k))] \geq \frac{1}{2} + \epsilon(k),$$

where the probability is over $C \leftarrow \mathcal{C}_k$, $s \leftarrow \text{Gen}_{\mathcal{G}}(1^k)$, and the random coin tosses of S .

Now, for every $C \in \mathcal{C}_k$, let $Y^C = (Y_1, \dots, Y_t)$ be the random variable guaranteed by the pseudo-entropy of values in I_k (Definition 2.7). We first consider an alternative experiment in which the oracle C is replaced with an oracle $C \circ Y^C$ that behaves like C on all points outside I_k , and on points in I_k answers according to Y^C . We claim that

$$\Pr [S^{C \circ Y^C}(i\mathcal{O}([K_s]_d)) = G_s(Y^C)] \geq \frac{1}{2} + \epsilon(k) - \text{negl}(k),$$

where the probability is over $C \leftarrow \mathcal{C}_k$, the random variable Y^C , $s \leftarrow \text{Gen}_{\mathcal{G}}(1^k)$, and the coin tosses of S . Indeed, this follows directly from the pseudo-entropy guarantee (Definition 2.7), together with the fact that a distinguisher can sample s and compute $i\mathcal{O}([K_s]_d)$ on its own.

Next, we change the above experiment so that instead of an indistinguishability obfuscation of K_s , the simulator gets an indistinguishability obfuscation of the circuit $K_{s_{x^*}}^*$, where s is punctured at the point $x^* = Y^C$. We claim that

$$\Pr \left[S^{C \circ Y^C} (i\mathcal{O}([K_{s_{x^*}}^*]_d)) = G_s(Y^C) \right] \geq \frac{1}{2} + \epsilon(k) - \text{negl}(k),$$

where the probability is over $C \leftarrow \mathcal{C}_k$, the random variable Y^C , $s \leftarrow \text{Gen}_{\mathcal{G}}(1^k)$, and the coin tosses of S , $x^* = Y^C$, and $s_{x^*} = \text{Punc}(s, x^*)$. Indeed, recalling that Y^C has min-entropy $m(k) + k$ for every $C \in \mathcal{C}_k$, there does not exist a circuit \tilde{C} such that $x^* := Y^C = \tilde{C}(I_k)$, except with negligible probability 2^{-k} . However, recall that in this case K_s and $K_{s_{x^*}}^*$ have the exact same functionality, and thus the above follows by the indistinguishability obfuscation guarantee.

It is now left to note that S predicts with noticeable advantage the value of G_s at the punctured point x^* , and thus violates the pseudo-randomness at punctured points requirement (Definition 2.6). \square

References

- [BBC⁺14] Boaz Barak, Nir Bitansky, Ran Canetti, Yael Tauman Kalai, Omer Paneth, and Amit Sahai. Obfuscation for evasive functions. In Yehuda Lindell, editor, *Theory of Cryptography (TCC 2014)*, volume 8349 of *Lecture Notes in Computer Science*, pages 26–51. Springer, 2014.
- [BGI⁺01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2001.
- [BGI13] Elette Boyle, Shafi Goldwasser, and Ioana Ivan. Functional signatures and pseudorandom functions. Cryptology ePrint Archive, Report 2013/401, 2013. <http://eprint.iacr.org/>.
- [BGK⁺13] Boaz Barak, Sanjam Garg, Yael Tauman Kalai, Omer Paneth, and Amit Sahai. Protecting obfuscation against algebraic attacks. Cryptology ePrint Archive, Report 2013/631, 2013. <http://eprint.iacr.org/>.
- [BR13a] Zvika Brakerski and Guy N. Rothblum. Black-box obfuscation for d -cnfs. Cryptology ePrint Archive, Report 2013/557, 2013. <http://eprint.iacr.org/>.
- [BR13b] Zvika Brakerski and Guy N. Rothblum. Virtual black-box obfuscation for all circuits via generic graded encoding. Cryptology ePrint Archive, Report 2013/563, 2013. <http://eprint.iacr.org/>.
- [BW13] Dan Boneh and Brent Waters. Constrained pseudorandom functions and their applications. Cryptology ePrint Archive, Report 2013/352, 2013. <http://eprint.iacr.org/>.
- [Can97] Ran Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In Burton S. Kaliski, Jr., editor, *Advances in Cryptology – CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 455–469. Springer, 1997.

- [CD08] Ran Canetti and Ronny Ramzi Dakdouk. Extractable perfectly one-way functions. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfssdóttir, and Igor Walukiewicz, editors, *Proceedings of the 35th International Colloquium on Automata, Languages and Programming, Part II*, volume 5126 of *Lecture Notes in Computer Science*, pages 449–460. Springer, 2008.
- [CRV10] Ran Canetti, Guy N. Rothblum, and Mayank Varia. Obfuscation of hyperplane membership. In Daniele Micciancio, editor, *Theory of Cryptography (TCC 2010)*, volume 5978 of *Lecture Notes in Computer Science*, pages 72–89. Springer, 2010.
- [CV13] Ran Canetti and Vinod Vaikuntanathan. Obfuscating branching programs using black-box pseudo-free groups. Cryptology ePrint Archive, Report 2013/500, 2013. <http://eprint.iacr.org/>.
- [GGH⁺13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2013)*, pages 40–49. IEEE Computer Society, 2013.
- [GGJS13] Shafi Goldwasser, Vipul Goyal, Abhishek Jain, and Amit Sahai. Multi-input functional encryption. Cryptology ePrint Archive, Report 2013/727, 2013. <http://eprint.iacr.org/>.
- [GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *J. ACM*, 33(4):792–807, 1986.
- [GGSW13] Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th Annual ACM Symposium on Theory of Computing (STOC 2013)*, pages 467–476. ACM, 2013.
- [GK05] Shafi Goldwasser and Yael Tauman Kalai. On the impossibility of obfuscation with auxiliary input. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005)*, pages 553–562. IEEE Computer Society, 2005.
- [GKP⁺13] Shafi Goldwasser, Yael Tauman Kalai, Raluca A. Popa, Vinod Vaikuntanathan, and Nikolai Zeldovich. Reusable garbled circuits and succinct functional encryption. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th Annual ACM Symposium on Theory of Computing (STOC 2013)*, pages 555–564. ACM, 2013.
- [GR07] Shafi Goldwasser and Guy N. Rothblum. On best-possible obfuscation. In Salil P. Vadhan, editor, *Theory of Cryptography (TCC 2007)*, volume 4392 of *Lecture Notes in Computer Science*, pages 194–213. Springer, 2007.
- [HSW13] Susan Hohenberger, Amit Sahai, and Brent Waters. Replacing a random oracle: Full domain hash from indistinguishability obfuscation. Cryptology ePrint Archive, Report 2013/509, 2013. <http://eprint.iacr.org/>.
- [KPTZ13] Aggelos Kiayias, Stavros Papadopoulos, Nikos Triandopoulos, and Thomas Zacharias. Delegatable pseudorandom functions and applications. Cryptology ePrint Archive, Report 2013/379, 2013. <http://eprint.iacr.org/>.

- [Ore87] Yair Oren. On the cunning power of cheating verifiers: Some observations about zero knowledge proofs. In *28th Annual IEEE Symposium on Foundations of Computer Science*, pages 462–471. IEEE Computer Society, 1987.
- [SW13] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: Deniable encryption, and more. Cryptology ePrint Archive, Report 2013/454, 2013. <http://eprint.iacr.org/>.