The Impossibility of Obfuscation with a Universal Simulator

Henry Cohn[*] Shafi Goldwasser[†] Yael Tauman Kalai[‡]

January 3, 2014

Abstract

We show that indistinguishability obfuscation implies that all functions with sufficient "pseudo-entropy" cannot be obfuscated under a virtual black box definition with a universal simulator. Let $\mathcal{F} = \{f_s\}$ be a circuit family with super-polynomial pseudo-entropy, and suppose $\mathcal{O}$ is a candidate obfuscator with universal simulator S. We demonstrate the existence of an adversary A that, given the obfuscation $\mathcal{O}(f_s)$, learns a predicate the simulator S cannot learn from the code of A and black-box access to f_s. Furthermore, this is true in a strong sense: for *any* secret predicate P that is not learnable from black-box access to f_s, there exists an adversary that given $\mathcal{O}(f_s)$ efficiently recovers $P(s)$, whereas given oracle access to f_s and given the code of the adversary, it is computationally hard to recover $P(s)$.

We obtain this result by exploiting a connection between obfuscation with a universal simulator and obfuscation with auxiliary inputs, and by showing new impossibility results for obfuscation with auxiliary inputs.

[*] 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

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$^+$01], who constructed (contrived) function families that are not obfuscatable under a very natural definition of obfuscation as well as various relaxations. The virtual black box (VBB) definition introduced in [BGI$^+$01] essentially requires that anything that can be efficiently computed given an obfuscation of a program could be efficiently computed from black box access to the program.

Following this work, much effort has been devoted to proving the existence of obfuscators for natural classes of programs. However, until the recent proposal of Garg et al. [GGH$^+$13], all known obfuscation candidates were for simple classes of functions, such as point functions [Can97], hyperplanes [CRV10], conjunctions [BR13b], and d-CNFs [BR13a]. The recent work of Garg et al. gave the first plausible candidate for general-purpose obfuscation, and they conjectured that it is a *best possible* [GR07] or *indistinguishability* obfuscator; i.e., given any two circuits C_1, C_2 of the same size and computing the same function, no polynomial-time adversary can distinguish between the obfuscation of C_1 and that of C_2. Followup works [BR13c, BGK$^+$13] proved security of variants of their scheme in the generic multi-linear group model. Subsequent to [GGH$^+$13], a flood of results have appeared showing that indistinguishability obfuscation suffices for many applications, such as the construction of public-key encryption from private-key encryption, the existence of deniable encryption, the existence of multi-input functional encryption, and more [SW13, GGH$^+$13, HSW13, GGJS13].

In contrast, we prove that the existence of indistinguishability obfuscation shows, in a somewhat strange twist, limitations on the possibility of VBB obfuscation with a universal simulator.

The definition of VBB obfuscation requires that for each probabilistic polynomial-time (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 this simulator S for a given adversary. It leaves open the possibility that the obfuscation could be broken in practice with no apparent simulator and thus without revealing how to disprove the underlying cryptographic assumptions.

A stronger and arguably more meaningful definition requires that there exist 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 with a *universal simulator*. All the proposed VBB obfuscators we are aware of in the literature, with the exception of that of Canetti [Can97], prove VBB security via a simulator which makes black-box use of the adversary code, and they therefore have universal simulators. Showing VBB obfuscation without universal simulation may require further development of simulation techniques in which the simulator uses the code of the adversary in an inefficient manner.

One of the motivations of indistinguishability obfuscation is that it is equivalent to "best possible" obfuscation [GR07]. In particular, indistinguishability obfuscation is VBB obfuscation whenever VBB obfuscation is possible. Thus, another interpretation of our results is that if we have even a limited degree of VBB obfuscation, then the scope of VBB obfuscation with a universal simulator is highly restricted. Thus, any attempt at VBB obfuscation will suffer from the shortcomings described below.

1.1 Our results

We show that indistinguishability obfuscation implies that any function family with super-polynomial "pseudo-entropy" *cannot* be VBB obfuscated with a universal simulator. Loosely speaking, a function family $\mathcal{F}$

has super-polynomial pseudo-entropy if it is difficult to distinguish a genuine function from $\mathcal{F}$ 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 PPT 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 2.5 for the precise definition, but note that such families include all pseudo-random function families, as well as every semantically secure secret-key or public-key encryption scheme, or any secure digital signature scheme in which randomness is generated by using a (secret) pseudo-random function.

We believe that VBB with universal simulation is a much more meaningful definition of obfuscation, as per our discussion above.

We obtain our results in two steps. First we prove that VBB security with a universal simulator is equivalent to VBB security with auxiliary inputs. 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 Theorem 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 Theorem 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 definition of VBB security with auxiliary inputs, originally considered in [GK05], is a strengthening of VBB security, which corresponds to the setting where the adversary may have some additional a priori information on the function being obfuscated. The motivation for considering auxiliary inputs is that they exist in just about any cryptographic usage of obfuscation. For example, security with respect to auxiliary input is needed when obfuscation is used together with other components in a larger scheme or protocol.

The paper [GK05] considered both *independent* and *dependent* auxiliary inputs. An obfuscator $\mathcal{O}$ for a function family $\mathcal{F}$ is (worst-case) VBB secure with *dependent* auxiliary inputs if for every 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 $\mathsf{A}(\mathcal{O}(f), \mathsf{aux}(f))$ is computationally indistinguishable from $\mathsf{S}^f(\mathsf{aux}(f))$. The average-case analogue of this definition requires that the output of $\mathsf{A}(\mathcal{O}(f), \mathsf{aux}(f))$ is computationally indistinguishable from $\mathsf{S}^f(\mathsf{aux}(f))$ 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 $\mathsf{A}(\mathcal{O}(f), \mathsf{aux})$ is computationally indistinguishable from $\mathsf{S}^f(\mathsf{aux})$.

The above two theorems 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.

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

1.1.1 Negative results for VBB obfuscation with auxiliary inputs

Goldwasser and Kalai [GK05] proved the existence of many "natural" classes of functions that are not obfuscatable with respect to (contrived) auxiliary inputs. For the case of dependent auxiliary inputs, they proved that every function family with super-polynomial pseudo-entropy cannot be obfuscated with respect to dependent auxiliary inputs, *assuming the class of point-filter functions are obfuscatable with respect to dependent auxiliary inputs*.[2] Thus, their result is a conditional one.

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 implies 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 in order to prove that all functions with super-polynomial pseudo-entropy are not obfuscatable with respect to dependent auxiliary inputs.

Informal Theorem. *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.*

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

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.

In followup work to a previous version of this paper (which proved Theorem 4.1 but not Lemmas 3.1 and 3.2), Bitansky et al. [BCPR13] strengthened this theorem,[3] and proved the same impossibility for the case of *independent* auxiliary inputs.

Informal Theorem [BCPR13]**.** *Assume the existence of indistinguishability obfuscation for a class of puncturable pseudo-random functions (PRFs).*[4] *Then no function family with super-polynomial pseudo-entropy has an average-case VBB obfuscator with respect to independent auxiliary input.*

We state 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.*

[2] We refer the reader to [GK05] for the definition of a point-filter function family.

[3] We note that technically the theorem of [BCPR13] is not a strengthening of our theorem, since our assumptions are incomparable. We assume witness encryption, which is implied by indistinguishability obfuscation for the class of point-filter functions, and they assume indistinguishability obfuscation for a specific class of pseudo-random functions (which they call puncturable PRFs).

[4] We refer the reader to [BCPR13] for the definition of puncturable PRFs.

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

Theorem 4.2 and Lemma 3.2 imply the following corollary.

Corollary 2. *Assume the existence of indistinguishability obfuscation for a class of puncturable pseudorandom 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 $\mathsf{poly}(|s|)$, 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 $s \in \{0,1\}^k$,*

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

where the probabilities are over the random coin tosses of A and S. Here $\mathsf{negl}(k)$ denotes an unspecified, negligible function (i.e., $|\mathsf{negl}(k)| = O(1/k^c)$ 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 $k \in \mathbb{N}$,*

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

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 $\mathsf{S}^{f_s}(\mathsf{A})$ of course means that S is given a circuit for A for inputs of the appropriate size.

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* $\mathsf{aux}(s)$ *of size* $\mathsf{poly}(k)$*,*

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

where the probabilities are over the random coin tosses of A *and* S*. We write* $\mathsf{aux}(s)$ *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* $\mathsf{aux}(s)$ *of size* $\mathsf{poly}(s)$ *(and allowed to depend on s),*

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

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* $\mathsf{aux} \in \{0,1\}^{\mathsf{poly}(k)}$*,*

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

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* $(\mathsf{Enc}, \mathsf{Dec})$ *such that the following conditions hold:*

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

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

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

$$\Pr[\mathsf{A}(\mathsf{Enc}_x(1^k, b)) = b] \leq \frac{1}{2} + \mathsf{negl}(k),$$

where the probability is over $b \leftarrow \{0,1\}$ and over the random coin tosses of Enc *and* A*.*

Definition 2.5 (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 $I \subseteq \{0,1\}^k$ of size $t(k)$, and for every $C \in \mathcal{C}_k$ there exists a random variable $Y^C = (Y_i)_{i\in I} \in \{0,1\}^I$, 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 $\mathcal{D}$,*

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

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. Here the probabilities are over $C \leftarrow \mathcal{C}_k$, the random variable Y^C, and the random coin tosses of $\mathcal{D}$.

We say that $\mathcal{C}$ has super-polynomial pseudo-entropy if it has pseudo-entropy at least p for every polynomial p.

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 $s \in \{0,1\}^k$,

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

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 $\mathsf{aux}(s)$, $\mathsf{S}_\mathsf{A}^{f_s}(\mathsf{aux}(s))$ runs the universal simulator S^{f_s} on input $\mathsf{A}_{\mathsf{aux}(s)}$, where $\mathsf{A}_{\mathsf{aux}(s)}$ is the (non-uniform) adversary that simulates A with auxiliary input $\mathsf{aux}(s)$. We need to prove that for every predicate π, every $k \in \mathbb{N}$, and every $s \in \{0,1\}^k$,

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

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

To do so, we check that

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

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 $\mathsf{A}_{\mathsf{aux}(s)}$ 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 $\mathsf{aux}(s)$ of size $\mathsf{poly}(k)$,

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

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 $\mathsf{aux} = \mathsf{aux}(s)$ as a (possibly non-uniform) PPT adversary and runs this adversary. 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 $\mathsf{aux}(s)$ of size $\mathsf{poly}(k)$,

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

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 $s \in \{0,1\}^k$,

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

To see why, note that

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

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). □

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 $k \in \mathbb{N}$,

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

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, $\mathsf{S}_\mathsf{A}^{f_s}(\mathsf{aux})$ 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 $\mathsf{aux} \in \{0,1\}^{\mathsf{poly}(k)}$,

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

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

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

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 $\mathsf{aux} \in \{0,1\}^{\mathsf{poly}(k)}$,

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

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. 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 $\mathsf{aux} \in \{0,1\}^{\mathsf{poly}(k)}$,

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

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 $k \in \mathbb{N}$,

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

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

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

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).

In followup work, Bitansky et al. [BCPR13] strengthened this theorem[5] and proved the same impossibility for the case of *independent* auxiliary inputs.

Theorem 4.2. *[BCPR13]: Assume the existence of indistinguishability obfuscation for a class of puncturable pseudo-random functions (PRFs).*[6] *Then no function family with super-polynomial pseudo-entropy has an average-case VBB obfuscator with respect to independent auxiliary input.*

[5] As explained in footnote 3, technically the assumptions are incomparable, but conceptually it is a strengthening.

[6] We refer the reader to [BCPR13] for the definition of puncturable PRFs.

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.*

The rest of this section is devoted for the proof of Theorem 4.1. For the sake of simplicity, we first prove it for a pseudo-random function family. We then show how the proof extends to any function family with super-polynomial pseudo-entropy.

Proof of Theorem 4.1 for pseudo-random functions. Assume the existence of a witness encryption scheme for some NP-complete language, and let $\mathcal{F} = \{f_s\}$ be any family of pseudo-random functions.

Suppose for the sake of contradiction that there exists an obfuscator $\mathcal{O}$ for $\mathcal{F}$ that takes as input $s \in \{0,1\}^k$ and outputs an obfuscated circuit $\mathcal{O}(f_s)$ of polynomial size $t = t(k)$. We can assume $t \geq k$.

Let $\mathcal{L}$ be the NP language defined by

$$\mathcal{L} = \{x = (x_1, \ldots, x_{2t}) : \text{there exists a circuit } C \text{ of size } |C| \leq t \text{ such that } C(i) = x_i \text{ for all } i \in [2t]\}.$$

Set $x = (f_s(1), \ldots, f_s(2t))$ and let $\mathsf{aux}(s) = \mathsf{Enc}_x(1^k, b)$, where $b \leftarrow \{0,1\}$ is a random bit and Enc is a witness encryption for $\mathcal{L}$. 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 $\mathcal{O}(f_s)$ and $\mathsf{aux}(s) = \mathsf{Enc}_x(1^k, b)$, one can efficiently decrypt b with probability $1 - \mathsf{negl}(k)$, since $\mathcal{O}(f_s)$ is a valid witness of x.

On the other hand, we prove the following claim, which contradicts the security of $\mathcal{O}$.

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

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

Proof of Claim 4.5 Suppose for the sake of contradiction that there exists a PPT adversary S such that

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

for some non-negligible function ϵ, where the probability is over random $(s, b) \leftarrow \{0,1\}^{k+1}$ and over the randomness of Enc.

The fact that f_s is a pseudo-random function implies that

$$\Pr[\mathsf{S}^{R}(\mathsf{Enc}_{x^*}(1^k, b)) = b] \geq \frac{1}{2} + \epsilon(k) + \mathsf{negl}(k), \tag{5}$$

where R is a truly random function and $x^* = (R(1), \ldots, R(2t))$. Otherwise the PPT adversary S could be used to distinguish R from f_s, contradicting the pseudo-randomness of f_s.

Note that $x^* \notin \mathcal{L}$ with probability at least $1 - 2^{-t}$ (since there are 2^{2t} choices of x^* and only 2^t circuits of size at most t). Therefore Equation (5) contradicts the semantic security of the underlying witness encryption scheme. □

□

4.1 Extending the proof of Theorem 4.1 to functions with super-polynomial pseudo-entropy.

Let $\mathcal{C}$ be any circuit class of polynomial size with super-polynomial pseudo-entropy. Suppose for the sake of contradiction that $\mathcal{C}$ has an obfuscator with auxiliary inputs, denoted by $\mathcal{O}$. Let $p = p(k)$ be a polynomial such that $|\mathcal{O}(C)| \leq p(k)$ for every $C \in \mathcal{C}_k$.

The fact that $\mathcal{C}$ has super-polynomial pseudo-entropy implies that it has pseudo-entropy at least $2p(k)$. In particular, recalling Definition 2.5, this implies that there exists a polynomial $t = t(k)$ and a subset $I \subseteq \{0,1\}^k$ of size $t(k)$ such that for every C there exists a random variable $Y^C = (Y_1, \ldots, Y_t)$ such that the following conditions hold:

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

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

   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. Here the probabilities are over $C \leftarrow \mathcal{C}_k$, the random variable Y^C, and the random coin tosses of $\mathcal{D}$.

We define an NP language $\mathcal{L}$ similarly to the previous case, by

$$\mathcal{L} = \{(x_i)_{i \in I} : \text{there exists a circuit } C \text{ of size } |C| \leq p \text{ such that } C(i) = x_i \text{ for all } i \in I\}.$$

Set $x = (C(i))_{i \in I}$ and let $\mathsf{aux}(C) = \mathsf{Enc}_x(1^k, b)$, where $b \leftarrow \{0,1\}$ is a random bit and Enc is a witness encryption for the language $\mathcal{L}$.

Note that given $\mathcal{O}(C)$ and $\mathsf{aux}(C) = \mathsf{Enc}_x(1^k, b)$, one can efficiently decrypt b with probability $1 - \mathsf{negl}(k)$, since $\mathcal{O}(C)$ is a valid witness of x. It remains to prove the following claim, which is analogous to Claim 4.5.

Claim 4.6. *For any (possibly non-uniform) PPT adversary* S *which takes as input* $\mathsf{aux}(s) = \mathsf{Enc}_x(1^k, b)$ *and has black-box access to* C*,*

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

Proof of Claim 4.6.

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

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

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

By assumption,

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

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

$$\Pr[\mathcal{D}^{C \circ Y^C}(1^k) = b] \geq \frac{1}{2} + \epsilon(k) + \mathsf{negl}(k). \tag{6}$$

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

Note however that x^* has min-entropy $2p(k)$ and so the probability that it is in $\mathcal{L}$ is at most $2^{-p(k)}$. Thus, Equation (6) contradicts the semantic security of the underlying witness-encryption scheme. □

□

Remark 4.7. Note that for any secret predicate P that is not learnable from black-box access to the circuit, we could have taken the auxiliary input to be $\mathsf{aux}(C) = \mathsf{Enc}_x(1^k, b)$ where $b = P(C)$ (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 $\mathsf{aux}(C)$ outputs $P(C)$ with probability 1, whereas any PPT simulator cannot learn $P(C)$ from $\mathsf{aux}(C)$ and black-box access to C.

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

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.

References

[BCPR13] Nir Bitansky, Ran Canetti, Omer Paneth, and Alon Rosen. More on the impossibility of virtual-black-box obfuscation with auxiliary input. Cryptology ePrint Archive, Report 2013/701, 2013. `http://eprint.iacr.org/`.

[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.

[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. Obfuscating conjunctions. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013*, volume 8043 of *Lecture Notes in Computer Science*, pages 416–434. Springer, 2013.

[BR13c] 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/`.

[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.

[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.

[GGH$^+$13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. Cryptology ePrint Archive, Report 2013/451, 2013. `http://eprint.iacr.org/`.

[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/`.

[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 Nickolai 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. IACR Cryptology ePrint Archive, Report 2013/509, 2013. `http://eprint.iacr.org/`.

[SW13] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: Deniable encryption, and more. IACR Cryptology ePrint Archive, Report 2013/454, 2013. `http://eprint.iacr.org/`.