One-Way Functions and (Im)perfect Obfuscation*

Ilan Komargodski† Tal Moran‡ Moni Naor† Rafael Pass§ Alon Rosen¶
Eylon Yogev†

Abstract

A program obfuscator takes a program and outputs a "scrambled" version of it, where the goal is that the obfuscated program will not reveal much about its structure beyond what is apparent from executing it. There are several ways of formalizing this goal. Specifically, in indistinguishability obfuscation, first defined by Barak et al. (CRYPTO 2001), the requirement is that the results of obfuscating any two functionally equivalent programs (circuits) will be computationally indistinguishable. Recently, a fascinating candidate construction for indistinguishability obfuscation was proposed by Garg et al. (FOCS 2013). This has led to a flurry of discovery of intriguing constructions of primitives and protocols whose existence was not previously known (for instance, fully deniable encryption by Sahai and Waters, STOC 2014). Most of them explicitly rely on additional hardness assumptions, such as one-way functions.

Our goal is to get rid of this extra assumption. We cannot argue that indistinguishability obfuscation of all polynomial-time circuits implies the existence of one-way functions, since if $\mathsf{P} = \mathsf{NP}$, then program obfuscation (under the indistinguishability notion) is possible. Instead, the ultimate goal is to argue that if $\mathsf{P} \neq \mathsf{NP}$ and program obfuscation is possible, then one-way functions exist.

Our main result is that if $\mathsf{NP} \not\subseteq \mathsf{io\text{-}BPP}$ and there is an efficient (even imperfect) indistinguishability obfuscator, then there are one-way functions. In addition, we show that the existence of an indistinguishability obfuscator implies (unconditionally) the existence of SZK-arguments for NP. This, in turn, provides an alternative version of our main result, based on the assumption of hard-on-the average NP problems. To get some of our results we need obfuscators for simple programs such as $3\mathsf{CNF}$ formulas.

*A preliminary version of this work appeared in Proceedings of the 55th Annual Symposium on Foundations of Computer Science (FOCS 2014). This paper incorporates the manuscript of Moran and Rosen [MR13].

†Weizmann Institute of Science. Email: {ilan.komargodski,moni.naor,eylon.yogev}@weizmann.ac.il. Supported in part by a grant from the I-CORE Program of the Planning and Budgeting Committee, the Israel Science Foundation, BSF, IMOS and the Citi Foundation. Moni Naor is the incumbent of the Judith Kleeman Professorial Chair.

‡IDC Herzliya. Email: talm@idc.ac.il. Supported by ISF grant no. 1790/13 and by the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 293843

§Cornell University. Email: rafael@cs.cornell.edu. Supported in part by a Alfred P. Sloan Fellowship, Microsoft New Faculty Fellowship, NSF Award CNS- 1217821, NSF CAREER Award CCF-0746990, NSF Award CCF-1214844, AFOSR YIP Award FA9550-10-1-0093, and DARPA and AFRL under contract FA8750-11-2- 0211. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the US Government.

¶IDC Herzliya. Email: 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 goal of program obfuscation is to transform a given program (say described as a boolean circuit) into another "scrambled" circuit which is functionally equivalent by "hiding" its implementation details (making it hard to "reverse-engineer"). The theoretical study of obfuscation was initiated by Barak et al. [BGI+01, BGI+12]. They studied several notions of obfuscation, primarily focusing on virtual black-box obfuscation (henceforth VBB). Virtual black-box obfuscation requires that anything that can be efficiently computed from the obfuscated program, can also be computed efficiently from black-box (i.e., input-output) access to the program. Their main result was that this notion of obfuscation cannot be achieved for all circuits. Moreover, the existence of virtual black-box obfuscators for various restricted families of functions is still a major open problem.

As a way to bypass their general impossibility result, Barak et al. [BGI+12] introduced the notion of indistinguishability obfuscation (henceforth $i\mathcal{O}$). An indistinguishability obfuscator is an algorithm that guarantees that if two circuits compute the same function, then their obfuscations are computationally indistinguishable.

Recently, there have been two significant developments regarding indistinguishability obfuscation: first, candidate constructions for obfuscators for all polynomial-time programs were proposed [GGH+13, BR14, BGK+14, PST14, AGIS14, GLSW14] and second, intriguing applications of $i\mathcal{O}$ have been demonstrated, e.g., general-purpose functional encryption scheme [GGH+13], deniable encryption with negligible advantage [SW14], two-round secure MPC [GGHR14], traitor-tracing schemes with very short messages [BZ14], secret-sharing for NP [KNY14] and more. However, essentially all these applications (and others) explicitly rely on some additional hardness assumption (such as one-way functions).[1] This should not come as a surprise: As noted already by Barak et al. [BGI+12], if P = NP, then there are no one-way functions but $i\mathcal{O}$ *does* exist.[2]

We consider both "perfect" obfuscators with perfect functionality (i.e., the obfuscator always preserves the functionality of the input circuit) and "imperfect" obfuscators, where the functionality is preserved only with overwhelming probability. Our goal is to deepen our understanding of the relation between several notions of obfuscation and one-way functions. We ask the following question:

Under which assumptions is it redundant to assume one-way functions on top of an efficient and possibly imperfect obfuscator?

Our Main Result. In this paper, we provide an answer to the above question. We show that if NP $\not\subseteq$ io-BPP and there is an efficient, even *imperfect*, $i\mathcal{O}$, then one-way functions exist, where io-BPP is the class of languages that can be decided by a probabilistic polynomial-time algorithm for infinitely many input lengths.[3]

In addition, we also provide a completely different proof of a somewhat weaker statement. We first show that the existence of efficient indistinguishability obfuscators for 3CNF formulas implies (unconditionally) the existence of SZK-arguments for NP. Then, we use a result of Ostrovsky [Ost91] which states that SKZ-arguments for hard-on-the average languages implies the

[1] Two notable exceptions are *witness encryption* [GGSW13] and *functional witness encryption* [BCP14]. However, Boyle et al. [BCP14] showed that these can be viewed as special cases of $i\mathcal{O}$.

[2] If P = NP, then the polynomial hierarchy collapses to P, thus we can efficiently find the lexicographically first circuit that has the same functionality as some given circuit.

[3] If we assume efficient and *perfect* $i\mathcal{O}$, then we give a simple argument that proves that NP $\not\subseteq$ io-coRP implies one-way functions. See Section 2 and appendix A for further details.

existence of one-way functions. Thus, we get that the existence of one-way functions can be based on the existence of a hard-on-the average NP-problem and, even *imperfect*, $i\mathcal{O}$ for 3CNFs. This result is weaker than the result above since the existence of hard-on-the average NP-problems implies that $\mathsf{NP} \not\subseteq \mathsf{io\text{-}BPP}$ (however, it only requires an obfuscator for 3CNF formulas, as opposed to all polynomial-size circuits).

Finally, we generalize a result of [BGI+12] and show that even if *imperfect* VBB obfuscators exist (even for a very simple family of functions such as point functions[4]), then one-way functions exist. We summarize our results in the following theorem.

Main Theorem. *Any of the following three conditions implies that one-way functions exist:*

1. $\mathsf{NP} \not\subseteq \mathsf{io\text{-}BPP}$ *and an efficient, even imperfect,* $i\mathcal{O}$ *for polynomial-size circuits exists.*
2. *Hard-on-the average functions in* NP *exist and an efficient, even imperfect,* $i\mathcal{O}$ *for* 3CNF *formulas exists.*
3. *An efficient, even imperfect, VBB obfuscator for point functions exists.*

A corollary of our main theorem is that many applications that assume (even imperfect) $i\mathcal{O}$ and one-way functions can be obtained by assuming $i\mathcal{O}$ and $\mathsf{NP} \not\subseteq \mathsf{io\text{-}BPP}$. Two notable examples are the construction of deniable encryption of Sahai and Waters [SW14] and the construction of a traitor-tracing scheme of Boneh and Zhandry [BZ14]. In addition, we view our results as making the claim of Sahai and Waters [SW14] that $i\mathcal{O}$ is a "central hub" of cryptography more cohesive.

Borrowing from Impagliazzo's terminology [Imp95], if (even imperfect) $i\mathcal{O}$ exists, then our result rules out *Pessiland*, where hard-on-the average languages exist but one-way functions do not. We observe that if $\mathsf{NP} \subseteq \mathsf{BPP}$, then one-way functions do not exist but $i\mathcal{O}$ does. Therefore, ignoring the issue of infinitely-often input lengths, we can state Item 1 of our main result as follows: $\mathsf{NP} \subseteq \mathsf{BPP}$ *if and only if* there exists an efficient indistinguishability obfuscator *and* one-way functions do not exist.

More Related Work. Subsequently to [BGI+12], Goldwasser and Kalai [GK05] and Goldwasser and Rothblum [GR07] introduced other variants of definitions of obfuscation and proved that they are also impossible to achieve in general.

Recently, a work of Garg et al. [GGH+13] proposed the first candidate construction of indistinguishability obfuscators relying on multilinear graded encodings. Different variants of this construction that are secure in idealized algebraic models have been proposed in [BR14, BGK+14, AGIS14], and [PST14] presents a construction of an $i\mathcal{O}$ whose security can be reduced to the assumption that semantically-secure graded encodings exist.

Paper Organization. In Section 2 we give a high level overview of our main techniques. In Section 3 we provide preliminary definitions and set up notation. In Sections 4 to 6 we prove Items 1 to 3 of our main theorem, respectively. In Section 7 we summarize and state some open problems. In Appendix B we prove that an approximate notion of $i\mathcal{O}$ is equivalent to the imperfect notion of $i\mathcal{O}$, thus one can get similar results to Items 1 and 2 of our main theorem while assuming approximate $i\mathcal{O}$.

[4] A Boolean function is a point function if it is the constant 0 function or it assumes the value 1 at exactly one point (and 0 everywhere else).

2 Our Techniques

We focus on Item 1 of the main theorem and present our main ideas and techniques. We say that an indistinguishability obfuscator $i\mathcal{O}$ is perfect if it perfectly preserves functionality (i.e., it always outputs a circuit that agrees with the input circuit on every input), and we say that $i\mathcal{O}$ is imperfect if it preserves functionality with overwhelming probability (i.e., with overwhelming probability it outputs a circuit that agrees with the input circuit on every input). For the exact definition we refer to Definition 3.4. By default, we assume that an indistinguishability obfuscator is *imperfect* (i.e., if we require it to be perfect, we explicitly say so).

Our starting observation is that if we assume the existence of an efficient *perfect* indistinguishability obfuscator, then assuming that $\mathsf{NP} \not\subseteq \mathsf{io\text{-}coRP}$ there are one-way functions, where io-coRP is the class of languages that can be coRP-decided (i.e., efficiently and probabilistically with a one-sided error) for infinitely many input lengths.

Observation 2.1. *Assume that* $\mathsf{NP} \not\subseteq \mathsf{io\text{-}coRP}$. *If there exists an efficient* perfect *indistinguishability obfuscator for* 3CNF *formulas, then one-way functions exist.*

The idea behind the proof of Observation 2.1 is simple and borrows the construction from [GR07, Theorem 4.1]. Given an efficient and *perfect* indistinguishability obfuscation scheme $i\mathcal{O}(C;x)$ (that uses randomness x to obfuscate an input 3CNF formula C), our candidate one-way function is defined as

$$f(x) = i\mathcal{O}(Z;x), \tag{2.1}$$

where Z is a circuit of appropriate size and input length that always outputs zero. Assuming that $i\mathcal{O}$ satisfies both *perfect* functionality and indistinguishability, we show how to use an adversary A that can (infinitely-often) invert the function f with non-negligible advantage (over the choice of a random input x) in order to (one-sided, infinitely-often) probabilistically decide the circuit (un)satisfiability of a given 3CNF formula C. This is done by simply observing whether A succeeds in inverting or not. The key observations in our argument are the following:

- If C is unsatisfiable, then by the indistinguishability of the $i\mathcal{O}$ scheme, A inverts f with non-negligible advantage even if we replace $f(x) = i\mathcal{O}(Z;x)$ with $f(x) = i\mathcal{O}(C;x)$.
- If C is satisfiable, then by the *perfect* functionality of the $i\mathcal{O}$ scheme, $i\mathcal{O}(Z;x)$ can never be a satisfiable circuit. Thus, no inverse of $i\mathcal{O}(C;x)$ exists and A fails to invert f when we replace $f(x) = i\mathcal{O}(Z;x)$ with $f(x) = i\mathcal{O}(C;x)$.

The full proof of Observation 2.1 can be found in Appendix A. We note that the intuition above (as well as the formal proof in Appendix A) makes strong use of the *perfect* functionality required from $i\mathcal{O}$.

Indeed, if the obfuscator is *imperfect*, then we cannot claim that if C is satisfiable, then $i\mathcal{O}(C;x)$ cannot be a circuit that always outputs zero: By the imperfect functionality of $i\mathcal{O}$ we are only guaranteed that for a *random* string x, with overwhelming probability, it will be the case that $i\mathcal{O}(C;x)$ is functionally equivalent to C. Therefore, for *every* satisfiable circuit C it is possible that there exists a string x such that $i\mathcal{O}(Z;x)$ is functionally equivalent to C. In this case, the inverter A can just output that x, causing us to answer incorrectly.

Remark 2.2. *Observe that all we need for Observation 2.1 is an indistinguishability obfuscator for* 3CNF formulas*. However, for Item 1 of our main theorem (see Theorem 4.1 for a formal statement) we require $i\mathcal{O}$ for polynomial-size circuits. It is a very interesting open problem to get a similar result to that of Theorem 4.1 but only relying on an obfuscator for* 3CNF*s.*

2.1 Going Beyond Perfect $i\mathcal{O}$

As we noted above, the simple construction given in Equation (2.1) does not work when $i\mathcal{O}$ is only guaranteed to be *imperfect*. We continue the overview by introducing a useful notation: For a circuit C we denote by $\widehat{C} \leftarrow i\mathcal{O}(C)$ a random variable that corresponds to a random obfuscation of C. Moreover, for two circuits C and $\widehat{C}$ we denote by $\varphi(C, \widehat{C})$ the set of random strings x for which $i\mathcal{O}(C; x) = \widehat{C}$.

Observe that, with the new set of notation, the inverter A of f from above is given a circuit $\widehat{C}$ and, if successful, finds an x such that $x \in \varphi(Z, \widehat{C})$. Thus, with high enough probability for any *unsatisfiable* circuit C it holds that $|\varphi(Z, \widehat{C})| \geq 1$, however, by the perfect functionality of $i\mathcal{O}$, for any *satisfiable* circuit C, it holds that $|\varphi(Z, \widehat{C})| = 0$. Hence, using A we can efficiently determine if the set $\varphi(Z, \widehat{C})$ is empty or not, that is, whether C is satisfiable or not.

Unfortunately, as we have said, when $i\mathcal{O}$ is imperfect this difference no longer holds. Thus, we seek for a stronger separation by φ of satisfiable and unsatisfiable circuits.

Towards a Strong Separation. One of our main observations (see Lemma 4.4) is that if C is a *satisfiable* circuit, then with high probability it holds that

$$|\varphi(C, \widehat{Z})| \ll |\varphi(Z, \widehat{Z})|. \tag{2.2}$$

At this point we wish to prove a complementary inequality, that is, if C is *unsatisfiable*, then with high probability it holds that

$$|\varphi(C, \widehat{Z})| \gg |\varphi(Z, \widehat{Z})|. \tag{2.3}$$

If this were true, then φ would act as a measure that can separate satisfiable and unsatisfiable circuits. Then, we would be left proving that there exists an efficient procedure $\varphi_{\approx}$ that can estimate the value of $|\varphi(\cdot, \widehat{Z})|$. We would decide satisfiability of a given circuit C by computing $\widehat{Z} \leftarrow i\mathcal{O}(Z)$, $\varphi_{\approx}(C, \widehat{Z})$ and $\varphi_{\approx}(Z, \widehat{Z})$, and comparing them.

However, the complementary inequality (Equation (2.3)) does not seem to follow from the basic properties of $i\mathcal{O}$.[5] Moreover, it seems hard to establish the estimator $\varphi_{\approx}$ (as defined above) for reasons we will discuss later.

A Strong Separation via Double Obfuscation. Our main idea, that solves both problems raised in the previous paragraph, is to consider the *double obfuscation* of a circuit.

Denote by $\widehat{\widehat{Z}}$ the *double* obfuscation of the circuit Z (i.e., $\widehat{\widehat{Z}} \leftarrow i\mathcal{O}(\widehat{Z})$). By the functionality property of $i\mathcal{O}$, a natural corollary of Equation (2.2) is that for a satisfiable circuit C with high probability it holds that

$$|\varphi(\widehat{C}, \widehat{\widehat{Z}})| \ll |\varphi(\widehat{Z}, \widehat{\widehat{Z}})|. \tag{2.4}$$

[5] If we assume there are one-way functions, then the complementary inequality is false. However, in a world without one-way functions, it is unclear.

Now assume that we have an estimator $\varphi_{\approx}$ that can efficiently estimate $\varphi(\cdot, \widehat{\widehat{Z}})$. Unlike before, by the indistinguishability property of $i\mathcal{O}$, we show that a weak version of the complementary inequality *is* true. Moreover, we show that this weaker version suffices for completing our proof. In particular, we show that if C is *unsatisfiable*, then with probability roughly $1/2$ it holds that

$$|\varphi(\widehat{C}, \widehat{\widehat{Z}})| \geq |\varphi(\widehat{Z}, \widehat{\widehat{Z}})|. \tag{2.5}$$

Indeed, given two independent samples from $|\varphi(\widehat{Z}, \widehat{\widehat{Z}})|$ we know that the first will be smaller than the second with probability $1/2$. Since $\widehat{C}$ is indistinguishable from $\widehat{Z}$, it must hold that any efficient algorithm that estimates $|\varphi(\cdot, \widehat{\widehat{Z}})|$ is unable to distinguish between whether it was given $\widehat{C}$ or $\widehat{Z}$. Thus, by the indistinguishability described above, we get that the same holds when one sample is from $|\varphi(\widehat{C}, \widehat{\widehat{Z}})|$ and the other is from $|\varphi(\widehat{Z}, \widehat{\widehat{Z}})|$.

At this point, given the two inequalities we can decide satisfiability of a given circuit C: compute $\widehat{Z} \leftarrow i\mathcal{O}(Z), \widehat{\widehat{Z}} \leftarrow i\mathcal{O}(\widehat{Z}), \widehat{C} \leftarrow i\mathcal{O}(C)$, $K_1 \leftarrow \varphi_{\approx}(\widehat{C}, \widehat{\widehat{Z}})$ and $K_2 \leftarrow \varphi_{\approx}(\widehat{Z}, \widehat{\widehat{Z}})$, and compare K_1 and K_2. If C is satisfiable, then with high probability K_1 will be much smaller than K_2. If C is unsatisfiable, then with probability roughly $1/2$, K_1 will be larger than K_2. Finally, we repeat this test many times to amplify the success probability. We are left to prove that $\varphi(\cdot, \widehat{\widehat{Z}})$ can be efficiently estimated.

Towards Efficiently Estimating φ. We start with a standard trick for estimating the size of such sets, that was originally used by Impagliazzo and Luby [IL89] (see also [Imp92]). Recall Equation (2.1) which defines the function f. We append to f a description of a (pairwise independent) hash function h and its evaluation on x. That is, we define the function

$$f'(x, h, k) = Z \circ i\mathcal{O}(Z; x) \circ h \circ h(x)|_k,$$

where $\circ$ denotes string concatenation operator and $h(x)|_k$ is the k bit long prefix of $h(x)$. Assuming that f' is not one-way, we have an efficient algorithm A' that inverts f' on random inputs with non-negligible probability. Using the leftover hash lemma [ILL89, HILL99], the inverter A' and the indistinguishability feature of $i\mathcal{O}$, one can obtain an efficient procedure $\varphi_{\approx}$ that estimates $|\varphi(Z, \widehat{C})|$ for any circuit C.

Unfortunately, as we have noted, we are interested in estimating $|\varphi(\widehat{C}, \widehat{\widehat{Z}})|$ and not $|\varphi(Z, \widehat{C})|$. A possible direction that might be useful is to try and estimate $|\varphi(C, \widehat{Z})|$. Recall that this is not what we ultimately want, however, the following example emphasizes a step towards the final solution. To do this, consider an inverter for the function defined as follows

$$f'_C(x, h, k) = C \circ i\mathcal{O}(C; x) \circ h \circ h(x)|_k. \tag{2.6}$$

This direction, however, has an immediate drawback: for each circuit C, f'_C might have a different inverter A'_C, which cannot be found efficiently, thus yielding a non-uniform estimator $\varphi_{\approx}$. We remark that if we assume that deciding circuit satisfiability is hard-on-the-average, then this problem can be solved. This is true since, in this case, C is sampled at random and can be thought of as an input to the function and not part of its description,[6] which results in having only a single inverter.

[6] That is, we can define the function $f'(C, x, h, k) \triangleq f'_C(x, h, k)$.

Estimating φ via Double Obfuscation. This step can intuitively be seen as a worst-case to average-case reduction. Roughly speaking, the *double* obfuscation allows us to re-randomize unsatisfiable instances while maintaining the separation by φ, yielding a uniform estimator $\varphi_{\approx}$ for the measures in Equations (2.4) and (2.5).

The idea is, as we discussed above, to *obfuscate the obfuscation* of Z. That is, we define the following variant of f' which is our final construction:

$$f''(x, y, h, k) = i\mathcal{O}(Z; y) \circ i\mathcal{O}(i\mathcal{O}(Z; y); x) \circ h \circ h(x)|_k,$$

Assuming that f'' is not one-way, then, there exists an inverter A'' for f''. As opposed to the previous construction, here we have a single inverter A'' that can be used for any circuit C. Using similar estimation techniques as before (sampling combined with the leftover hash lemma), we are able to use A'' to construct an estimator $\varphi_{\approx}$ that can estimate $|\varphi(\widehat{Z}, \widehat{\widehat{Z}})|$ and $|\varphi(\widehat{C}, \widehat{\widehat{Z}})|$ for *satisfiable* circuits C (we remark that we only require and achieve estimation in some suffice sense). For *unsatisfiable* circuits C, in this case, *any* efficient estimator for $|\varphi(\widehat{Z}, \widehat{\widehat{Z}})|$ is also a good estimator for $|\varphi(\widehat{C}, \widehat{\widehat{Z}})|$, since $\widehat{C}$ and $\widehat{Z}$ are indistinguishable.

At this point, we have all the ingredients. Given a circuit C, we can use $\varphi_{\approx}$ to efficiently estimate $K_C = |\varphi(\widehat{C}, \widehat{\widehat{Z}})|$ and $K_Z = |\varphi(\widehat{Z}, \widehat{\widehat{Z}})|$. Using the guarantees of Equations (2.4) and (2.5) we can determine if C is satisfiable or not by the difference between K_C and K_Z. For the exact details and the full proof we refer to Section 4.

3 Preliminaries

We start with some general notation. We denote by $[n]$ the set of numbers $\{1, 2, \ldots, n\}$. We denote by $\mathsf{neg} : \mathbb{N} \to \mathbb{R}$ a function such that for every positive integer c there exists an integer N_c such that for all $n > N_c$, $\mathsf{neg}(n) < 1/n^c$. For two strings $x \in \{0,1\}^n$ and $y \in \{0,1\}^m$ we denote by $x \circ y$ the string concatenation of x and y.

For a set S, we let $\mathbf{U}_S$ denote the uniform distribution over S. For an integer $m \in \mathbb{N}$, we let $\mathbf{U}_m$ denote the uniform distribution over $\{0,1\}^m$, the bit-strings of length m. For a distribution or random variable X we write $x \leftarrow X$ to denote the operation of sampling a random x according to X. For a set S, we write $s \leftarrow S$ as shorthand for $s \leftarrow \mathbf{U}_S$. For a randomized algorithm A, we write $\Pr_A[\cdot]$ (resp., $\mathbb{E}_A[\cdot]$) to state that the probability (resp., expectation) is over the internal randomness of the algorithm A. Finally, throughout this paper we denote by log the base 2 logarithm and we define $\log 0 = 0$.

Throughout this paper we deal with Boolean circuits. We denote by $|C|$ the size of a circuit C and define it as the number of wires in C.

3.1 Computational Indistinguishability

Definition 3.1 (Computational Indistinguishability)**.** *Two sequences of random variables $X = \{X_n\}_{n\in\mathbb{N}}$ and $Y = \{Y_n\}_{n\in\mathbb{N}}$ are computationally indistinguishable if for every probabilistic polynomial time algorithm A there exists an integer N such that for all $n \geq N$,*

$$|\Pr[A(X_n) = 1] - \Pr[A(Y_n) = 1]| \leq \mathsf{neg}(n).$$

where the probabilities are over X_n, Y_n and the internal randomness of A.

3.2 One-Way Functions

Definition 3.2 (One-Way Functions)**.** *A function f is said to be* one-way *if the following two conditions hold:*

1. *There exists a polynomial-time algorithm A such that $A(x) = f(x)$ for every $x \in \{0,1\}^*$.*
2. *For every probabilistic polynomial-time algorithm A and all sufficiently large n,*

$$\Pr[A'(1^n, f(x)) \in f^{-1}(f(x))] < \mathsf{neg}(n),$$

where the probability is taken uniformly over all possible $x \in \{0,1\}^n$ and the internal randomness of A'.

Definition 3.3 (Weak One-Way Functions)**.** *A function f is said to be* weakly one-way *if the following two conditions hold:*

1. *There exists a polynomial-time algorithm A such that $A(x) = f(x)$ for every $x \in \{0,1\}^*$.*
2. *There exists a polynomial p such that for every probabilistic polynomial-time algorithm A and all sufficiently large n,*

$$\Pr[A'(1^n, f(x)) \in f^{-1}(f(x))] < 1 - \frac{1}{p(n)},$$

where the probability is taken uniformly over all possible $x \in \{0,1\}^n$ and the internal randomness of A'.

3.3 Obfuscation

We say that two circuits C and C' are *equivalent* and denote it by $C \equiv C'$ if they compute the same function (i.e., $\forall x : C(x) = C'(x)$).

Indistinguishability Obfuscation

Definition 3.4 (Perfect/Imperfect Indistinguishability Obfuscator)**.** *Let $\mathcal{C} = \{\mathcal{C}_n\}_{n\in\mathbb{N}}$ be a class of polynomial-size circuits, where $\mathcal{C}_n$ is a set of circuits operating on inputs of length n. A uniform algorithm $i\mathcal{O}$ is called an (*imperfect*)* indistinguishability obfuscator *for the class $\mathcal{C}$ if it takes as input a security parameter and a circuit in $\mathcal{C}$ and outputs a new circuit so that following properties are satisfied:*

1. *(Perfect/Imperfect) Preserving Functionality:*

   There exists a negligible function α such that for any input length $n \in \mathbb{N}$, any λ and any $C \in \mathcal{C}_n$ it holds that

$$\Pr_{i\mathcal{O}}\left[C \equiv i\mathcal{O}(1^\lambda, C)\right] \geq 1 - \alpha(\lambda),$$

   where the probability is over the internal randomness of $i\mathcal{O}$. If $\alpha(\cdot) = 0$, then we say that $i\mathcal{O}$ is perfect*.*

2. *Polynomial Slowdown:*

   There exists a polynomial $p(\cdot)$ such that: For any input length $n \in \mathbb{N}$, any λ and any circuit $C \in \mathcal{C}_n$ it holds that $\left|i\mathcal{O}(1^\lambda, C)\right| \leq p(|C|)$.

3. *Indistinguishable Obfuscation:*

   For any probabilistic polynomial-time algorithm D, any $n \in \mathbb{N}$, any two equivalent *circuits $C_1, C_2 \in \mathcal{C}_n$ of the same size and large enough λ, it holds that*

$$\left|\Pr_{i\mathcal{O},D}\left[D\left(i\mathcal{O}\left(1^\lambda, C_1\right)\right) = 1\right] - \Pr_{i\mathcal{O},D}\left[D\left(i\mathcal{O}\left(1^\lambda, C_2\right)\right) = 1\right]\right| \leq \mathsf{neg}(\lambda).$$

We say that $i\mathcal{O}$ is efficient *if it runs in polynomial-time.*

Virtual Black-Box Obfuscation

Definition 3.5 (Perfect/Imperfect VBB Obfuscator)**.** *Let $\mathcal{C} = \{\mathcal{C}_n\}_{n\in\mathbb{N}}$ be a class of polynomial-size circuits, where $\mathcal{C}_n$ is a set of circuits operating on inputs of length n. A uniform algorithm $\mathcal{O}$ is called an (*imperfect*) VBB obfuscator for the class $\mathcal{C}$ if it takes as input a security parameter and a circuit in $\mathcal{C}$ and outputs a new circuit so that following properties are satisfied:*

1. *(Perfect/Imperfect) Preserving Functionality:*

   There exists a negligible function α such that for any input length $n \in \mathbb{N}$, any λ and any $C \in \mathcal{C}_n$ it holds that

$$\Pr_{\mathcal{O}}\left[C \equiv \mathcal{O}(1^\lambda, C)\right] \geq 1 - \alpha(\lambda),$$

   where the probability is over the internal randomness of $\mathcal{O}$. If $\alpha(\cdot) = 0$, then we say that $\mathcal{O}$ is perfect*.*

2. *Polynomial Slowdown:*

   There exists a polynomial $p(\cdot)$ such that: For any input length $n \in \mathbb{N}$, any λ and any circuit $C \in \mathcal{C}_n$ it holds that $\left|\mathcal{O}(1^\lambda, C)\right| \leq p(|C|)$.

3. *Virtual Black-Box:*

   For any probabilistic polynomial-time algorithm D, any predicate $\pi : \mathcal{C}_n \to \{0,1\}$, any $n \in \mathbb{N}$ and any circuit $C \in \mathcal{C}_n$, there is a polynomial-size simulator S such that for large enough λ it holds that

$$\left|\Pr_{\mathcal{O},D}\left[D\left(\mathcal{O}\left(1^\lambda, C\right)\right) = \pi(C)\right] - \Pr_{S}\left[D\left(S^C\left(1^\lambda\right)\right) = \pi(C)\right]\right| \leq \mathsf{neg}(\lambda).$$

We say that $\mathcal{O}$ is efficient *if it runs in polynomial-time.*

Notation. For ease of notation, 1^λ, the first parameter of $i\mathcal{O}$ and $\mathcal{O}$, is sometimes omitted when it is clear from the context.

3.4 Leftover Hash Lemma

Definition 3.6 (Statistical Distance)**.** *The statistical distance between two random variables X, Y is defined by*

$$\mathsf{SD}(X,Y) \triangleq \frac{1}{2} \cdot \left(\sum_x |\Pr[X = x] - \Pr[Y = x]| \right)$$

Definition 3.7 (Pairwise Independence)**.** *A family $\mathcal{H}_n^k : \{h : \{0,1\}^n \to \{0,1\}^k\}$ of functions is called pairwise independent if for all distinct $x, y \in \{0,1\}^n$ and every $a_1, a_2 \in \{0,1\}^k$, it holds that*

$$\Pr_{h \leftarrow \mathcal{H}_n^k}[h(x) = a_1 \ \wedge \ h(y) = a_2] = 2^{-2k}.$$

The following formulation of the leftover hash lemma is taken from [Gol08, Theorem D.5].

Theorem 3.8 (Leftover Hash Lemma)**.** *Let $\mathcal{H}_n^k$ be a family of pairwise independent hash functions and $S \subseteq \{0,1\}^n$. Let $\varepsilon = \sqrt[3]{2^k / |S|}$. Consider random variables X and H that is uniformly distributed on S and $\mathcal{H}_n^k$, respectively. Then,*

$$\mathsf{SD}\left(H \circ H(X), H \circ \mathbf{U}_k\right) \leq 2\varepsilon.$$

4 From Imperfect $i\mathcal{O}$ to One-Way Functions

In this section we prove Item 1 of our main theorem and show that if an efficient indistinguishability obfuscator exists and $\mathsf{NP} \not\subseteq \mathsf{io\text{-}BPP}$, then one-way functions exist.

Theorem 4.1. *Assume that $\mathsf{NP} \not\subseteq \mathsf{io\text{-}BPP}$. If there exists an efficient (even imperfect) indistinguishability obfuscator for polynomial-size circuits, then one-way functions exist.*

To prove Theorem 4.1, we assume towards contradiction that there are no one-way functions (and, in particular, there are no *weakly* one-way functions (see e.g., [Gol01, Theorem 2.3.2]). Note, however, that the latter only guarantees that for every function there is an efficient inverter that succeeds on *infinitely many* inputs length. We use the existence of this efficient inverter to solve an NP-complete problem in probabilistic polynomial-time with two sided error. Thus, we get that an algorithm that solves the NP-complete problem *infinitely-often* (io), and thus $\mathsf{NP} \subseteq \mathsf{io\text{-}BPP}$ contradicting our assumption. In the rest of the proof, for simplicity, we ignore this infinitely-often issue.

Let $i\mathcal{O}(1^\lambda, C; r)$ be an efficient indistinguishability obfuscator, where λ is a security parameter, C is the input circuit and r is the randomness used by the obfuscator. Let $Z_{s,n}$ be the canonical zero circuit of size s that accepts n inputs.

Throughout the proof, we use several parameters: λ the security parameter, n the number of input bits, s the size of the circuit and $|r|$ the number of random bits used by the obfuscator (the latter might depend on λ and s). For simplicity of exposition, we will assume that they are all equal and denote them by n (otherwise, one could always increase the security parameter and add dummy inputs to make them equal).

Let $\mathcal{H}_m = \{h : \{0,1\}^m \to \{0,1\}^m\}$ be a pairwise independent hash function family (see Definition 3.7). For a function $h \in \mathcal{H}_m$, an input $x \in \{0,1\}^m$ and an integer $k \in [m]$ we denote by

$h(x)|_k$ the k bit long prefix of $h(x)$. Define the function family $\mathcal{F} = \{f_n \colon \{0,1\}^n \times \{0,1\}^n \times \mathcal{H}_n \times \{0,1,\ldots,n\} \to \{0,1\}^*\}_{n\in\mathbb{N}}$ where

$$f_n(r_1, r_2, h, k) = i\mathcal{O}(1^n, Z_{n,n}; r_2) \circ i\mathcal{O}\left(1^s, i\mathcal{O}(1^n, Z_{n,n}; r_2); r_1\right) \circ h \circ k \circ h(r_1)|_k.$$

Note that since $i\mathcal{O}$ is efficiently computable then so is f_n.

Suppose, towards contradiction, that $\mathcal{F}$ is not weakly one-way. Then, there exists a probabilistic polynomial-time adversary A that can invert outputs of f_n on random inputs with probability at least $1 - 1/n^{50}$.[7] We show that using A we are able to (probabilistically and) efficiently solve circuit satisfiability. Let $f = f_n$.

Notation. Recall that for every two circuits C and C' we define

$$\varphi(C, C') \triangleq \{r \in \{0,1\}^n \mid i\mathcal{O}(C; r) = C'\}.$$

That is, $\varphi(C, C')$ is the set of random strings r for which applying $i\mathcal{O}$ on C with randomness r leads to C'. For a circuit C, we denote by $\widehat{C}_r \triangleq i\mathcal{O}(C; r)$ a shorthand for the obfuscation of the circuit C when applied with randomness r. Moreover, we denote by $\widehat{\widehat{C}}_{r_1,r_2} = i\mathcal{O}(i\mathcal{O}(C; r_2); r_1)$ the shorthand for the (double) obfuscation of the circuit C when applied with randomness r_2 and then applied with randomness r_1.

Proof Overview. Roughly speaking, the proof follows the ideas presented in Section 2. In what follows, we give an overview of these main steps and how they are used to prove our main result. Let C be a circuit. Let $\widehat{C}$ be a uniform obfuscation of C and $\widehat{\widehat{Z}}$ be a uniform obfuscation of a uniform obfuscation of the canonical zero circuit Z. Our main claims are the following:

1. Lemma 4.2 - We prove that there exists a procedure that gives a good estimation for $|\varphi(\widehat{C}, \widehat{\widehat{Z}})|$ (see Lemma 4.2 for the exact details). This result uses the assumption that f is not one-way in a very strong way.

2. Lemma 4.3 - We prove that since we can efficiently estimate $|\varphi(\widehat{C}, \widehat{\widehat{Z}})|$ (by the previous item), then it must be that case that with probability $1/2$ for every *unsatisfiable* circuit it holds that

   $$|\varphi(\widehat{C}, \widehat{\widehat{Z}})| \geq |\varphi(\widehat{Z}, \widehat{\widehat{Z}})|.$$

   This is true since otherwise we get an efficient algorithm that breaks the indistinguishability feature of $i\mathcal{O}$.

3. Corollary 4.5 - We prove that if C is a *satisfiable* circuit, then with very high probability

   $$|\varphi(\widehat{C}, \widehat{\widehat{Z}})| \ll |\varphi(\widehat{Z}, \widehat{\widehat{Z}})|.$$

   We emphasize that this inequality is unconditional and follows from the (possibly imperfect) functionality feature of $i\mathcal{O}$.

Using Items 1,2 and 3 it is easy to get an algorithm that distinguishes between a satisfiable and an unsatisfiable circuit: we compute $\varphi_{\approx}(\widehat{C}, \widehat{\widehat{Z}})$ and $\varphi_{\approx}(\widehat{Z}, \widehat{\widehat{Z}})$ and compare them.

[7] More precisely, we are only guaranteed that A is able to invert random outputs of f_n infinitely-often (i.e., for infinitely many n's). However, as we said, in order not to complicate the proof, we ignore this issue throughout the analysis.

The Full Proof. We begin by showing that although we cannot compute exactly $|\varphi(C, C')|$ for any two circuits, in some cases we can approximate it quite well.

Lemma 4.2. *Let C be a circuit. Let $\widehat{C} \leftarrow i\mathcal{O}(C)$ and $\widehat{\widehat{Z}} \leftarrow i\mathcal{O}(i\mathcal{O}(Z))$ be random variables. There exists a procedure $\varphi_{\approx}$ that gets as input $\widehat{C}$ and $\widehat{\widehat{Z}}$ and with probability at least $1 - 1/n^{10}$ over $\widehat{C}, \widehat{\widehat{Z}}$ and the internal randomness of $\varphi_{\approx}$ satisfies that:*

1. $\varphi_{\approx}(\widehat{C}, \widehat{\widehat{Z}}) \leq \log |\varphi(\widehat{C}, \widehat{\widehat{Z}})| + 90 \log n$.
2. *If C is* unsatisfiable*, then* $\varphi_{\approx}(\widehat{C}, \widehat{\widehat{Z}}) \geq \log |\varphi(\widehat{C}, \widehat{\widehat{Z}})| - 90 \log n$.

Proof. We describe the procedure $\varphi_{\approx}$ that gets two obfuscated circuit as input $\widehat{C}, \widehat{\widehat{Z}}$ and estimates $|\varphi(\widehat{C}, \widehat{\widehat{Z}})|$. The procedure $\varphi_{\approx}$ approximates the maximum k on which A is able to invert on inputs of the form $\widehat{C} \circ \widehat{\widehat{Z}} \circ h \circ k \circ s$ where s is a random string of length k. If $|\varphi(\widehat{C}, \widehat{\widehat{Z}})|$ is small, then there is a small number of random string r' that A can find, thus the probability that one of them also satisfies $h(r') = s$ is small. Therefore, A will fail to invert with high probability for large enough values of k. The formal description appears in Figure 1. In the rest of the proof we analyze the procedure $\varphi_{\approx}$.

The $\varphi_{\approx}$ Procedure

Input: A circuit $\widehat{C} \leftarrow i\mathcal{O}(C)$ and a circuit $\widehat{\widehat{Z}} \leftarrow i\mathcal{O}(i\mathcal{O}(Z))$.

1. Initialize $\mathsf{max}_k \leftarrow -\infty$.
2. For $k = 0 \ldots n$ do:
   (a) Sample uniformly at random a hash function $h \in \mathcal{H}_n$ and a random strings s of length k.
   (b) Set $y \leftarrow \widehat{C} \circ \widehat{\widehat{Z}} \circ h \circ k \circ s$.
   (c) Run $r'_1, r'_2, h', k' \leftarrow A(y)$.
   (d) If $f(r'_1, r'_2, h', k') = y$, set $\mathsf{max}_k \leftarrow k$.
3. Return max_k.

Figure 1: φ Estimation Procedure.

Point 1 of the lemma (Upper Bound). We show that with very high probability it holds that the output of $\varphi_{\approx}(\widehat{C}, \widehat{\widehat{Z}})$ is (roughly) upper bounded by $\log |\varphi(\widehat{C}, \widehat{\widehat{Z}})|$. Let $\widehat{C} \leftarrow i\mathcal{O}(C)$, $\widehat{\widehat{Z}} \leftarrow i\mathcal{O}(i\mathcal{O}(Z))$ and let $k \geq \log |\varphi(\widehat{C}, \widehat{\widehat{Z}})| + 90 \log n$. Then, for any h, there are at most $|\varphi(\widehat{C}, \widehat{\widehat{Z}})|$ strings of the form $h(z)$ for $z \in \varphi(\widehat{C}, \widehat{\widehat{Z}})$. Since there are $n^{90} \cdot \varphi(\widehat{C}, \widehat{\widehat{Z}})$ strings of this length (i.e., of length k), and since we select s at random, the probability that it is in this set is at most $1/n^{90}$. Clearly, if there is no inverse to y, then A cannot successfully invert. Therefore, the probability that max_k is updated to be k in Step 2d (i.e., A successfully inverts) is at most $1/n^{90}$. Since there are at most n different values for k as above, this gives a bound of at most $1/n^{89}$ probability of any such k being the final value of max_k.

Point 2 of the lemma (Lower Bound). We show that if C is unsatisfiable, then with high probability it holds that the output of $\varphi_{\approx}(\widehat{C}, \widehat{\widehat{Z}})$ is (roughly) lower bounded by $\log|\varphi(\widehat{C}, \widehat{\widehat{Z}})|$.

Intuitively, the proof of the lower bound goes as follows. We begin with the assumption that A is a very good inverter for random images of f that are of the form $\widehat{Z} \circ \widehat{\widehat{Z}} \circ h \circ k \circ h(x)|_k$ (since f is not weakly one-way). In particular, A is a good inverter when we fix k and the other inputs are random, as above. It follows, by the leftover hash lemma, that for small enough values of k, the inverter A must also invert inputs of the form $\widehat{Z} \circ \widehat{\widehat{Z}} \circ h \circ k \circ s$. By the indistinguishability feature of $i\mathcal{O}$, for every unsatisfiable circuit C the inverter A must also invert (with high enough probability) given inputs sampled uniformly at random from the distribution $\widehat{C} \circ \widehat{\widehat{Z}} \circ h \circ k \circ s$, which proves our claim. This intuition is made precise in the rest of the proof.

By our assumption, A fails to invert a random image of f with probability at most $1/n^{50}$. Namely,

$$\Pr_{r_1,r_2,h,k,A}[A(f(r_1,r_2,h,k)) \notin f^{-1}(f(r_1,r_2,h,k))] \leq 1/n^{50}.$$

Denote $\widehat{\widehat{Z}}_{r_{\text{outer}},r_{\text{inner}}} \triangleq i\mathcal{O}(i\mathcal{O}(Z; r_{\text{inner}}); r_{\text{outer}})$. By Markov's inequality we have that for a randomly chosen r_1, r_2 with probability at least $1 - 1/n^{25}$ the conditional probability that A succeeds on $f_n(r_1', r_2, h, k)$ over a random choice of h, k and $r_1' \in \varphi(\widehat{Z}_{r_2}, \widehat{\widehat{Z}}_{r_1,r_2}) \triangleq \Gamma$ is at least $1 - 1/n^{25}$. Namely,

$$\Pr_{r_1,r_2}[\Pr_{r_1' \leftarrow \Gamma,h,k,A}[A(f(r_1',r_2,h,k)) \in f^{-1}(f(r_1',r_2,h,k))] \geq 1 - 1/n^{25}] \geq 1 - 1/n^{25}.$$

Since every possible value of k is chosen with probability $1/n$ we get that given any fixed value of k it holds that

$$\Pr_{r_1,r_2}[\Pr_{r_1' \leftarrow \Gamma,h,A}[A(f(r_1',r_2,h,k)) \in f^{-1}(f(r_1',r_2,h,k))] \geq 1 - 1/n^{24}] \geq 1 - 1/n^{25}.$$

Recall that the procedure $\varphi_{\approx}(\widehat{C}, \widehat{\widehat{Z}})$ emulates the execution of A on input of the form $\widehat{C} \circ \widehat{\widehat{Z}} \circ h \circ k \circ s$ where h is chosen uniformly at random from $\mathcal{H}_n$ and s is chosen uniformly at random from $\{0,1\}^k$. Using the leftover hash lemma (see Theorem 3.8) we have that

$$\mathsf{SD}\left(H \circ H(X'), H \circ \mathbf{U}_k\right) \leq \varepsilon,$$

where X' is uniformly distributed over $\varphi(\widehat{C}, \widehat{\widehat{Z}})$, H is uniformly distributed over $\mathcal{H}_n$ and $\varepsilon = 2 \cdot \sqrt[3]{2^k/|\varphi(\widehat{C}, \widehat{\widehat{Z}})|}$. We get that, for any possible value of k it holds that

$$\Pr_{r_1,r_2}[\Pr_{r_1' \leftarrow \Gamma,h,A,s}[A(\widehat{Z}_{r_2} \circ \widehat{\widehat{Z}}_{r_1',r_2} \circ h \circ k \circ s) \in f^{-1}(f(r_1',r_2,h,k))] \geq 1 - 1/n^{24} - \varepsilon] \geq 1 - 1/n^{25}.$$

Plugging in $k^* = \log|\varphi(\widehat{Z}_{r_1}, \widehat{\widehat{Z}}_{r_1,r_2})| - 90\log n$ we get that

$$\Pr_{r_1,r_2}[\Pr_{r_1' \leftarrow \Gamma,h,A,s}[A(\widehat{Z}_{r_2} \circ \widehat{\widehat{Z}}_{r_1',r_2} \circ h \circ k^* \circ s) \in f^{-1}(f(r_1',r_2,h,k^*))] \geq 1 - 1/n^{23}] \geq 1 - 1/n^{25}.$$

Since C is unsatisfiable (i.e., functionally equivalent to Z), then A is able to distinguish between the distribution $\widehat{Z}_U$ and the distribution $\widehat{C}_U$ only with negligible probability. Moreover, the joint

distribution $(\widehat{Z}_{U_1}, \widehat{\widehat{Z}}_{U_1,U_2})$ is computationally indistinguishable from $(\widehat{C}_{U_1}, \widehat{\widehat{Z}}_{U_2,U_3})$, where U_1, U_2, U_3 are distributed uniformly at random from $\{0,1\}^n$. Therefore, since A is efficient, it must also invert inputs of the form $\widehat{C} \circ \widehat{\widehat{Z}} \circ h \circ k^* \circ s$. Namely, for $\Gamma' \triangleq \varphi(\widehat{C}_{r_3}, \widehat{\widehat{Z}}_{r_1,r_2})$ we have

$$\Pr_{r_1,r_2,r_3}[\Pr_{r'_1 \leftarrow \Gamma', h, A, s}[A(\widehat{C}_{r_3} \circ \widehat{\widehat{Z}}_{r'_1,r_2} \circ h \circ k^* \circ s) \in f^{-1}(f(r'_1, r_2, h, k^*))] \geq 1 - 1/n^{23}] \geq 1 - 1/n^{24}.$$

The claim follows. □

Next, we show that for every unsatisfiable circuit C, with probability roughly $1/2$, the value $|\varphi(\widehat{C}, \widehat{\widehat{Z}})|$ cannot be much smaller than $|\varphi(\widehat{Z}, \widehat{\widehat{Z}})|$.

Lemma 4.3. *Let C be any* unsatisfiable *circuit whose size is equal to the size of Z. Let $\widehat{Z} \leftarrow i\mathcal{O}(Z)$, $\widehat{\widehat{Z}} \leftarrow i\mathcal{O}(\widehat{Z})$ and $\widehat{C} \leftarrow i\mathcal{O}(C)$ be random variables. Then, with probability $1/2 - 1/n^9$ over the internal randomness of $i\mathcal{O}$, it holds that*

$$\log|\varphi(\widehat{C}, \widehat{\widehat{Z}})| \geq \log|\varphi(\widehat{Z}, \widehat{\widehat{Z}})| - 200\log n.$$

Proof. Let $\widehat{Z}, \widehat{Z}' \leftarrow i\mathcal{O}(Z)$ and $\widehat{\widehat{Z}} \leftarrow i\mathcal{O}(\widehat{Z})$ be random variables. Since $\widehat{Z}$ and $\widehat{Z}'$ are functionally equivalent (with probability $1-\mathsf{neg}(n)$), the distribution $(\widehat{Z}, \widehat{\widehat{Z}})$ is computationally indistinguishable from $(\widehat{Z}', \widehat{\widehat{Z}})$. Therefore, since $\varphi_\approx$ is a polynomial-time algorithm, it must be that the distribution of the output of $\varphi_\approx(\widehat{Z}, \widehat{\widehat{Z}})$ is computationally indistinguishable from the output of $\varphi_\approx(\widehat{Z}', \widehat{\widehat{Z}})$. That is,

$$\varphi_\approx(\widehat{Z}, \widehat{\widehat{Z}}) \approx_c \varphi_\approx(\widehat{Z}', \widehat{\widehat{Z}}).$$

Similarly, for C as in the statement and the random variable $\widehat{C} \leftarrow i\mathcal{O}(C)$, it holds that

$$\varphi_\approx(\widehat{Z}', \widehat{\widehat{Z}}) \approx_c \varphi_\approx(\widehat{C}, \widehat{\widehat{Z}})$$

Together, we get that

$$\varphi_\approx(\widehat{Z}, \widehat{\widehat{Z}}) \approx_c \varphi_\approx(\widehat{C}, \widehat{\widehat{Z}}). \tag{4.1}$$

Let $K_1 \leftarrow \log \varphi_\approx(\widehat{Z}, \widehat{\widehat{Z}})$ and $K_2 \leftarrow \log \varphi_\approx(\widehat{Z}, \widehat{\widehat{Z}})$ be two independent samples from the distribution of the output of $\varphi_\approx(\widehat{Z}, \widehat{\widehat{Z}})$ (where both $\widehat{Z}$'s are fresh samples and $\widehat{\widehat{Z}} \leftarrow i\mathcal{O}(\widehat{Z})$ of the appropriate $\widehat{Z}$). Therefore, with probability at least $1/2$, it holds that

$$K_1 \leq K_2.$$

Thus, from Equation (4.1), it holds that for $K_3 \leftarrow \log \varphi_\approx(\widehat{C}, \widehat{\widehat{Z}})$, with probability $1/2 - \mathsf{neg}(n)$:

$$K_1 \leq K_3.$$

Finally, using the fact that the procedure $\varphi_\approx$ gives a good estimation to φ (see Lemma 4.2), we get that with probability $1/2 - \mathsf{neg}(n) - 1/n^{10}$, it holds that

$$\log|\varphi(\widehat{C}, \widehat{\widehat{Z}})| \geq \log|\varphi(\widehat{Z}, \widehat{\widehat{Z}})| - 200\log n.$$

□

Next, we show that for every two unsatisfiable circuits C and D, with high probability the value $|\varphi(C, \widehat{D})|$ is much smaller than $|\varphi(D, \widehat{D})|$.

Lemma 4.4. *Let C and D be any two equal-size functionally non-equivalent circuits and let $p(\cdot)$ be any polynomial. Let $\widehat{D} \leftarrow i\mathcal{O}(D)$. Then, with probability $1 - \mathsf{neg}(n)$ over the internal randomness of $i\mathcal{O}$, it holds that*

$$p(n) \cdot |\varphi(C, \widehat{D})| < |\varphi(D, \widehat{D})|.$$

Proof. Let $p(\cdot)$ be a polynomial and let $\widehat{D}_r = \widehat{D}$, where r is the randomness used to generate $\widehat{D}$. Assume towards contradiction that there exists a polynomial $q(\cdot)$ such that

$$\Pr_r \left[p(n) \cdot |\varphi(C, \widehat{D}_r)| \geq |\varphi(D, \widehat{D}_r)| \right] \geq \frac{1}{q(n)}. \tag{4.2}$$

Denote by Bad the set of r's for which $p(n) \cdot |\varphi(C, \widehat{D}_r)| \geq |\varphi(D, \widehat{D}_r)|$. By Equation (4.2) we have that $\Pr_r[r \in \mathsf{Bad}] \geq 1/q(n)$. From the completeness of $i\mathcal{O}$ we have that $\Pr_r[r \in \mathsf{Bad} \ \wedge \ \widehat{D}_r \equiv D] \geq 1/q(n) - \mathsf{neg}(n)$. Denote by Bad' the set of all $r \in \mathsf{Bad}$ for which $\widehat{D}_r \equiv D$. In particular, for any $r \in \mathsf{Bad}'$ it holds that $|\{y \in \{0,1\}^n \mid i\mathcal{O}(C; y) = \widehat{D}_r\}| \geq |\{y \in \{0,1\}^n \mid i\mathcal{O}(D; y) = \widehat{D}_r\}|/p(n)$. Then,

$$\begin{aligned}
\Pr_y[i\mathcal{O}(C; y) \not\equiv C] &\geq \Pr_y[i\mathcal{O}(C; y) \in \{\widehat{D}_r \mid r \in \mathsf{Bad}'\}] \\
&\geq \Pr_y[i\mathcal{O}(D; y) \in \{\widehat{D}_r \mid r \in \mathsf{Bad}'\}] \cdot \frac{1}{p(n)} \\
&\geq \frac{1}{p(n)} \cdot \left(\frac{1}{q(n)} - \mathsf{neg}(n) \right) \geq \frac{1}{2p(n) \cdot q(n)}.
\end{aligned}$$

Clearly, this is a contradiction to the completeness of $i\mathcal{O}$ which proves the claim. □

Since for every circuit C it holds that $\widehat{C} \leftarrow i\mathcal{O}(C)$ is functionally equivalent to C with probability $1 - \mathsf{neg}(n)$, we get the following corollary.

Corollary 4.5. *Let C be any* satisfiable *circuit whose size is the size of Z. Let $\widehat{C} \leftarrow i\mathcal{O}(C)$, $\widehat{Z} \leftarrow i\mathcal{O}(Z)$ and $\widehat{\widehat{Z}} \leftarrow i\mathcal{O}(\widehat{Z})$. For any constant $c \in \mathbb{N}$, with probability at least $1 - \mathsf{neg}(n)$ it holds that $c \log n + \log |\varphi(\widehat{C}, \widehat{\widehat{Z}})| < \log |\varphi(\widehat{Z}, \widehat{\widehat{Z}})|$.*

4.1 Proof of Theorem 4.1

We prove Theorem 4.1 by showing how to combine Lemmas 4.2 and 4.3 and Corollary 4.5 in order to devise an efficient (probabilistic) algorithm $\mathsf{SolveSAT}$ that gets a circuit C as input and satisfies the following (infinitely-often):

1. If C is satisfiable, then $\Pr_{\mathsf{SolveSAT}}[\mathsf{SolveSAT}(C) = \text{“SAT”}] \geq 1 - \mathsf{neg}(n)$.
2. If C is unsatisfiable, then $\Pr_{\mathsf{SolveSAT}}[\mathsf{SolveSAT}(C) = \text{“UNSAT”}] \geq 1/2 - 1/n^8$.

The SolveSAT Procedure

Input: A circuit C that receives n inputs.

Let $\varphi_{\approx}$ be the procedure from Lemma 4.2.

Algorithm:

1. Sample $\widehat{Z} \leftarrow i\mathcal{O}(Z)$, $\widehat{\widehat{Z}} \leftarrow i\mathcal{O}(\widehat{Z})$ and $\widehat{C} \leftarrow i\mathcal{O}(C)$.
2. Compute $K_Z \leftarrow \varphi_{\approx}(\widehat{Z}, \widehat{\widehat{Z}})$ and $K_C \leftarrow \varphi_{\approx}(\widehat{C}, \widehat{\widehat{Z}})$.
3. If $K_Z - K_C < 400 \log n$, output "UNSAT" and halt.
4. Otherwise, output "SAT".

Figure 2: SAT Solver.

Then, one can amplify the probabilities by a standard BPP amplification technique (i.e., repeat multiple times).

The algorithm SolveSAT runs as follows. It samples $\widehat{Z} \leftarrow i\mathcal{O}(Z)$, $\widehat{\widehat{Z}} \leftarrow i\mathcal{O}(\widehat{Z})$ and $\widehat{C} \leftarrow i\mathcal{O}(C)$ and uses $\varphi_{\approx}$ to estimate $\varphi_{\approx}(\widehat{Z}, \widehat{\widehat{Z}})$ and $\varphi_{\approx}(\widehat{C}, \widehat{\widehat{Z}})$. If the distance between the two is small, then it outputs "UNSAT". Otherwise, in the end it outputs "SAT". The formal description appears in Figure 2.

By Lemma 4.2 we know that with probability at least $1 - 1/n^{10}$ it holds that

$$|K_Z - \varphi(\widehat{Z}, \widehat{\widehat{Z}})| \leq 90 \log n.$$

Assume that C is an unsatisfiable circuit. By Lemma 4.2 we get that with probability $1 - 1/n^{10}$ it holds that

$$|K_C - \varphi(\widehat{C}, \widehat{\widehat{Z}})| \leq 90 \log n.$$

Using Lemma 4.3 we also know that with probability $1/2 - 1/n^9$

$$|\varphi(\widehat{Z}, \widehat{\widehat{Z}}) - \varphi(\widehat{C}, \widehat{\widehat{Z}})| \leq 200 \log n.$$

Therefore, using the triangle inequality, with probability $1/2 - 1/n^8$ it holds that

$$K_Z - K_C < 400 \log n,$$

and the procedure will output "UNSAT".

Next, assume that C is a satisfiable circuit. Using Corollary 4.5 we know that with probability $1 - \mathsf{neg}(n)$ it holds that

$$\varphi(\widehat{Z}, \widehat{\widehat{Z}}) - \varphi(\widehat{C}, \widehat{\widehat{Z}}) \geq 800 \log n.$$

Using Item 2 of Lemma 4.2 we have that with probability at least $1 - 1/n^9$ it holds that

$$K_Z - K_C \geq \varphi(\widehat{Z}, \widehat{\widehat{Z}}) - 90 \log n - (\varphi(\widehat{C}, \widehat{\widehat{Z}}) + 90 \log n) > 600 \log n.$$

Therefore, in this case, SolveSAT outputs "SAT" with probability at least $1 - \mathsf{neg}(n)$, as required.

5 From Imperfect $i\mathcal{O}$ to One-Way Functions Through SZK

In this section we prove Item 2 of our main theorem. We assume the existence of an (imperfect) indistinguishability obfuscator for 3CNF formulas. We show that assuming the existence of hard-on-the average NP problems, one-way functions exist.

Theorem 5.1. *Assume the existence of a hard-on-the average* NP*-problem. If there exists an efficient imperfect indistinguishability obfuscator for* 3CNF *formulas, then one-way functions exist.*

In order to prove Theorem 5.1 we need the following theorem (that might be interesting in its own right) that states that $i\mathcal{O}$ implies *unconditionally* SZK-arguments for NP.

Theorem 5.2. *If there exists an efficient (and even imperfect) indistinguishability obfuscators for* 3CNF *formulas, then there exists a statistical zero-knowledge argument for* NP*.*

Theorem 5.1 follows by combining Theorem 5.2 with a result of Ostrovsky [Ost91] - showing that honest-verifier statistical zero-knowledge arguments for hard-on-the average languages implies the existence of one-way functions.[8]

Proof of Theorem 5.2. We first prove the theorem assuming that $i\mathcal{O}$ is an indistinguishability obfuscator for polynomial-size circuits and then show how to modify the protocol to get the same result but only assuming that $i\mathcal{O}$ is an indistinguishability obfuscator for 3CNF formulas.

We first observe that $i\mathcal{O}$ for polynomial-size circuits implies a two-round perfect honest-verifier zero-knowledge argument for NP.[9] Let $i\mathcal{O}$ be an efficient indistinguishability obfuscator and consider an NP-language L with an associated witness relation R_L. Let $\Pi^s_x(w)$ be a circuit that outputs s if $w \in R_L(x)$; otherwise, it outputs $\perp$. The verifier V on input a statement $x \in \{0,1\}^n$ picks a random $s \leftarrow \{0,1\}^n$, generates $C \leftarrow i\mathcal{O}(\Pi^s_x)$ and sends it to the prover. The prover P, on input x, a witness w, and receiving C from V, lets $s' \leftarrow C(w)$ and sends s' back to V. V accepts if and only if $s = s'$.

The protocol is clearly *complete* and perfect honest-verifier zero-knowledge (a simulator knowing the random tape of V simply outputs s).

To show *soundness*, consider some cheating prover P^* that convinces V with inverse polynomial probability $1/p(|x|)$ for infinitely many $x \notin L$. Consider some $x \notin L$. Note that Π^s_x is functionally equivalent to the "dummy" circuit $\Pi^\perp$ that always outputs $\perp$. Thus, by the indistinguishability property of $i\mathcal{O}$, C is indistinguishable from $C' = i\mathcal{O}(\Pi^\perp)$. It follows that in a modified experiment where V sends C' instead of C, P^* also convinces V with inverse polynomial probability $1/p'(|x|)$ for infinitely many $x \notin L$. However, in this experiment P^*'s view is independent of s and it can thus only guess s with probability $2^{-|s|}$, which is a contradiction.

Finally, to conclude the first part of our theorem we use a result by Ong and Vadhan [OV07] showing that the existence of a statistical honest-verifier zero-knowledge argument for a language L implies the existence of a statistical zero-knowledge argument for L.

[8] Alternatively, using a result of Ostrovsky and Wigderson [OW93], if we assume NP $\not\subseteq$ io-BPP, then we can deduce the existence of "auxiliary input" one-way functions, which are not sufficient for many cryptographic applications. However, the result of Theorem 4.1 shows that under the same assumption (i.e., NP $\not\subseteq$ io-BPP) we can deduce a stronger result (i.e., that one-way functions exist).

[9] A similar observation was made informally and independently by Pandey et al. [PPS13] using exactly the same construction. We thank Kai-Min Chung for pointing this out.

Using $i\mathcal{O}$ for 3CNFs. We show how the protocol from above can be modified to get an SZK-argument while assuming $i\mathcal{O}$ for 3CNF formulas only. Let the length of the secret s be ℓ and denote $s = s_1, \ldots, s_\ell$.

Assume that the NP language L is 3SAT. Then, we observe that R_{3SAT} can be implemented using a 3CNF formula. Moreover, for every $i \in [\ell]$ it holds that $\Pi_x^{s_i}$ can be implemented using a 3CNF formula, as well. Thus, instead of sending $i\mathcal{O}(\Pi_x^s)$ to the prover, we send the circuits $i\mathcal{O}(\Pi_x^{s_1}), \ldots, i\mathcal{O}(\Pi_x^{s_\ell})$. The prover acts similarly to the prover from the previous protocol on each obfuscated circuit separately. It is easy to see that this protocol is also a honest-verifier SZK-argument for 3SAT, which uses only an obfuscator for 3CNFs.

In the general case, where L is an arbitrary language in NP, since 3SAT is NP-complete, we can apply the previous protocol after reducing the instance of L to an instance of 3SAT. That is, we add an additional step in which the verifier (resp., the prover) convert its input x (resp., witness w) for L into an input x' (resp., witness w') for 3SAT. Then, instead of sending $i\mathcal{O}(\Pi_x^{s_i})$, the verifier sends $i\mathcal{O}(\Pi'^{s_i}_{x'})$ (for every $i \in [\ell]$), where Π' implements the relation R_{3SAT}. Thus, it is enough to use an obfuscator for Π', where Π' can be implemented using a 3CNF formula, as discussed above. □

Remark. In the above proof of Theorem 5.2, the only thing we require from C is that it is a "witness encryption" [GGSW13] (at least according to the definition from [BCP14]) of the string s. Recall that a witness encryption scheme enables one to encrypt a message m with respect to an NP-language L, an instance x and a function f, such that anyone that has, and only those that have, a witness w for $x \in L$ can recover $f(m, w)$. Therefore, we have actually shown that witness encryption for NP (even with imperfect correctness) implies statistical zero-knowledge arguments for NP.

6 From Imperfect VBB to One-Way Functions

In this section we prove Item 3 of our main theorem. We show that the existence of efficient, even imperfect, VBB obfuscators implies (unconditionally) the existence of one-way functions.

Barak et al. [BGI+12, Lemma 3.8] proved that *perfect* efficient VBB obfuscators imply one-way functions. Their proof strongly relies on the assumption that $\mathcal{O}$ is a *perfect* VBB obfuscator. In the rest of this section we generalize their result and prove the following theorem.

Theorem 6.1. *If an efficient, even* imperfect*, VBB obfuscator for point functions exists, then one-way functions exist.*

Proof. For $\alpha \in \{0,1\}^n$ and $b \in \{0,1\}$, let $C_{\alpha,b} : \{0,1\}^n \to \{0,1\}$ be the circuit defined by

$$C_{\alpha,b}(x) \triangleq \begin{cases} b & \text{if } x = \alpha \\ 0 & \text{otherwise.} \end{cases}$$

Suppose that $\mathcal{O}(1^\lambda, C; r)$ is an efficient (possibly imperfect) VBB obfuscator, where λ is a security parameter, C is the input circuit and r is the randomness used by the obfuscator. Assume that $|r| = p(\lambda)$ for a polynomial p. For $m \in \mathbb{N}$ let $\mathcal{H}_m = \{h : \{0,1\}^m \to \{0,1\}^m\}$ be a pairwise independent hash function family (see Definition 3.7). As in Section 4, for a function $h \in \mathcal{H}_m$, an

input $r \in \{0,1\}^m$ and an integer $k \in [m]$ we denote by $h(r)|_k$ the k-bit long prefix of $h(r)$. For $n \in \mathbb{N}$ let $n' = n + 1 + p(n)$ and define $f_n : \{0,1\}^{n'} \times \mathcal{H}_{n'} \times [n'] \to \{0,1\}^*$ as

$$f_n(\alpha, b, r, h, k) = \mathcal{O}(1^n, C_{\alpha,b}; r) \circ h \circ k \circ h(\alpha \circ b \circ r)|_k, \tag{6.1}$$

i.e., the obfuscation of the circuit $C_{\alpha,b}$ using randomness r followed by a description of a hash function $h \in \mathcal{H}_n$ and the k-bit long prefix of $h(\alpha \circ b \circ r)$. We prove that the function $f = \bigcup_{n\in\mathbb{N}} f_n$ is a weak one-way function.

Notation. For a circuit $\widehat{C}$ define

$$\psi(\widehat{C}) = \{(\alpha, b, r) \in \{0,1\}^{n'} \mid \mathcal{O}(C_{\alpha,b}; r) = \widehat{C}\},$$

i.e., $\psi(\widehat{C})$ is the set of tuples (α, b, r) for which the obfuscation of $C_{\alpha,b}$ with randomness r is exactly the circuit $\widehat{C}$.

The following claim states with non-negligible probability the input (α, b, r, h, k) to f_n is *good* in the following sense: (1) k is chosen to be the "correct" value, and (2) the only valid string α', b', r' that is consistent with $h(\alpha \circ b \circ r)$ (i.e., $h(\alpha' \circ b' \circ r')|_k = h(\alpha \circ b \circ r)|_k$) is $\alpha \circ b \circ r$ itself.

Claim 6.2. *With non-negligible probability over a random input (α, b, r, h, k) to f_n the following two conditions hold simultaneously:*

1. $k = \max\{1, \log|\psi(\mathcal{O}(C_{\alpha,b}; r))| - 10\}$.[10]
2. $|\psi(\mathcal{O}(C_{\alpha,b}; r)) \cap \{(\alpha', b', r') \mid h(\alpha' \circ b' \circ r')|_k = h(\alpha \circ b \circ r)|_k\}| = 1$.

Proof. We prove Item 1 of the claim. Notice that $\log|\psi(\mathcal{O}(C_{\alpha,b}; r))| \leq n'$. Thus, with probability at least $1/n'$ it holds that $k = \max\{1, \log|\psi(\mathcal{O}(C_{\alpha,b}; r))| - 10\}$.

We proceed with the proof of Item 2 of the claim. Let $t \triangleq |\psi(\mathcal{O}(C_{\alpha,b}; r))|$. It is clear that the tuple (α, b, r) itself is in the intersection of $\psi(\mathcal{O}(C_{\alpha,b}; r))$ and $\{(\alpha', b', r') \mid h(\alpha' \circ b' \circ r')|_k = h(\alpha \circ b \circ r)|_k\}$.

We are left to show that with non-negligible probability there is no $(\alpha', b', r') \in \psi(\mathcal{O}(C_{\alpha,b}; r))$ such that $(\alpha', b', r') \neq (\alpha, b, r)$ and $h(\alpha' \circ b' \circ r')|_k = h(\alpha \circ b \circ r)|_k$. From the pairwise independence of $\mathcal{H}_n$ we have that for any $(\alpha', b', r') \neq (\alpha, b, r)$ it holds that $\Pr_h[h(\alpha' \circ b' \circ r')|_k \neq h(\alpha \circ b \circ r)|_k] = 1 - 2^{-k}$. Therefore, the probability that for all $(\alpha', b', r') \in \psi(\mathcal{O}(C_{\alpha,b}; r))$ such that $(\alpha', b', r') \neq (\alpha, b, r)$ it holds that $h(\alpha' \circ b' \circ r')|_k \neq h(\alpha \circ b \circ r)|_k$ is at least $(1 - 2^{-k})^t \geq \Omega(1)$ by Item 1 of the claim. □

Assume that the event from Claim 6.2 holds. We show that, in this case, every probabilistic polynomial-time algorithm cannot invert f_n with probability larger than $1 - 1/\mathsf{poly}(n)$. More precisely, we prove the following claim.

Claim 6.3. *Assume that the event from Claim 6.2 holds. Then, for any probabilistic polynomial-time algorithm A it holds that*

$$\Pr[A(f_n(\alpha, b, r, h, k)) \notin f_n^{-1}(f_n(\alpha, b, r, h, k))] \geq 1/\mathsf{poly}(n).$$

[10] As before, for simplicity, we ignore integrality issues.

Since the event from Claim 6.2 holds with non-negligible probability Claim 6.3 implies that any efficient algorithm fails to invert f_n with probability larger than $1/\mathsf{poly}(n)$. Therefore, the function f is weakly one-way, which implies that one-way functions exist (see e.g., [Gol01, Theorem 2.3.2]). We conclude with the proof of Claim 6.3.

Proof of Claim 6.3. Clearly for any simulator S and random string s (that is independent of α and b) of length k, we have that

$$\Pr[S^{\mathcal{O}(C_{\alpha,b};r)}(h \circ k \circ s) = b] \leq 1/2 + \mathsf{neg}(n).$$

From the security guarantee of $\mathcal{O}$ (recall that $\mathcal{O}$ is a VBB obfuscator) it holds that

$$\Pr[A(\mathcal{O}(C_{\alpha,b};r) \circ h \circ k \circ s) = b] \leq 1/2 + \mathsf{neg}(n).$$

By Markov's inequality we get that for a randomly chosen (α, b, r) with probability at least $1/5$ the conditional probability that A succeeds on $f_n(\alpha', b', r', h, k)$ over a random choice of h, k and $(\alpha', b', r') \in \psi(C_{\alpha,b};r) \triangleq \Gamma$ is at least $1/3$. Namely,

$$\Pr_{\alpha,b,r}[\Pr_{h,k}[A(\mathcal{O}(C_{\alpha,b};r) \circ h \circ k \circ s) \neq b] \geq 1/3] \geq 1 - (3/4 + \mathsf{neg}(n)) \geq 1/5.$$

From Item 1 of Claim 6.2 and the leftover hash lemma (see Theorem 3.8), we get that

$$\Pr_{\alpha,b,r}[\Pr_{(\alpha',b',r')\in\Gamma,h,k}[A(\mathcal{O}(C_{\alpha',b'};r') \circ h \circ k \circ h(\alpha', b', r')|_k) \neq b] \geq 1/3 - \varepsilon] \geq 1/5,$$

where $\varepsilon = 2 \cdot \sqrt[3]{2^k/|\psi(\mathcal{O}(C_{\alpha,b};r))|} \leq 1/5$. Therefore,

$$\Pr_{\alpha,b,r,h,k}[A(\mathcal{O}(C_{\alpha,b};r) \circ h \circ k \circ h(\alpha, b, r)|_k) \neq b] \geq \Omega(1),$$

Using Item 2 of Claim 6.2 we have that any inverter A must, in particular, find the "correct" value of b. That is,

$$\begin{aligned}\Pr[A(f_n(\alpha, b, r, h, k)) \in f_n^{-1}(f_n(\alpha, b, r, h, k))] \leq \\ \Pr[A(\mathcal{O}(C_{\alpha,b};r) \circ h \circ k \circ h(\alpha \circ b \circ r)|_k) = b] \leq 1 - \Omega(1),\end{aligned}$$

which completes the proof of the claim. □

□

7 Summary and Open Problems

We have shown that several definitions of obfuscation essentially imply one-way functions. In particular, we showed that (even imperfect) efficient indistinguishability obfuscators together with complexity-theoretic assumptions imply one-way functions (see Theorems 4.1 and 5.1), and imperfect VBB obfuscators (unconditionally) imply one-way functions (see Theorem 6.1).

Theorem 4.1 assumes $\mathsf{NP} \not\subseteq \mathsf{io\text{-}BPP}$ and $i\mathcal{O}$ for polynomial-size circuits, and Theorem 5.1 assumes hard-on-the average NP-problems and $i\mathcal{O}$ for $\mathsf{3CNF}$ formulas (or, better yet, witness encryption for NP). That is, Theorem 4.1 assumes a weaker complexity-theoretic assumption but a stronger obfuscator than Theorem 5.1. It would be very interesting to fully characterize the minimal class of functions whose indistinguishability obfuscation (or witness encryption) implies one-way functions under the weakest complexity-theoretic assumption (cf., Theorems 4.1 and 5.1).

Acknowledgements

We are grateful to Huijia Lin for pointing out several mistakes in a previous version of the paper. We thank Nir Bitansky, Zvika Brakerski and Ron Rothblum for many helpful discussions.

References

[AGIS14] Prabhanjan Ananth, Divya Gupta, Yuval Ishai, and Amit Sahai. Optimizing obfuscation: Avoiding barringtons theorem. *IACR Cryptology ePrint Archive*, 2014:222, 2014.

[BCP14] Elette Boyle, Kai-Min Chung, and Rafael Pass. On extractability obfuscation. In *TCC*, volume 8349 of *Lecture Notes in Computer Science*, pages 52–73. 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 *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2001.

[BGI+12] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. *Journal of the ACM*, 59(2):6, 2012. Preliminary version [BGI+01].

[BGK+14] Boaz Barak, Sanjam Garg, Yael Tauman Kalai, Omer Paneth, and Amit Sahai. Protecting obfuscation against algebraic attacks. In *EUROCRYPT*, volume 8441 of *Lecture Notes in Computer Science*, pages 221–238. Springer, 2014.

[BR14] Zvika Brakerski and Guy N. Rothblum. Virtual black-box obfuscation for all circuits via generic graded encoding. In *TCC*, pages 1–25, 2014.

[BZ14] Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In *CRYPTO (1)*, volume 8616 of *Lecture Notes in Computer Science*, pages 480–499. Springer, 2014.

[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 *FOCS*, pages 40–49, 2013.

[GGHR14] Sanjam Garg, Craig Gentry, Shai Halevi, and Mariana Raykova. Two-round secure mpc from indistinguishability obfuscation. In *TCC*, volume 8349 of *Lecture Notes in Computer Science*, pages 74–94. Springer, 2014.

[GGSW13] Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In *STOC*, pages 467–476. ACM, 2013.

[GK05] Shafi Goldwasser and Yael Tauman Kalai. On the impossibility of obfuscation with auxiliary input. In *FOCS*, pages 553–562. IEEE Computer Society, 2005.

[GLSW14] Craig Gentry, Allison B. Lewko, Amit Sahai, and Brent Waters. Indistinguishability obfuscation from the multilinear subgroup elimination assumption. *IACR Cryptology ePrint Archive*, 2014:309, 2014.

[Gol01] Oded Goldreich. *The Foundations of Cryptography - Volume 1, Basic Techniques*. Cambridge University Press, 2001.

[Gol08] Oded Goldreich. *Computational Complexity - A Conceptual Perspective*. Cambridge University Press, 2008.

[GR07] Shafi Goldwasser and Guy N. Rothblum. On best-possible obfuscation. In *TCC*, volume 4392 of *Lecture Notes in Computer Science*, pages 194–213. Springer, 2007.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudo-random generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[IL89] Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography (extended abstract). In *FOCS*, pages 230–235. IEEE Computer Society, 1989.

[ILL89] Russell Impagliazzo, Leonid A. Levin, and Michael Luby. Pseudo-random generation from one-way functions (extended abstracts). In *STOC*, pages 12–24. ACM, 1989.

[Imp92] Russell Impagliazzo. *Pseudo-random generators for cryptography and for randomized algorithms*. PhD thesis, University of California, Berkeley, 1992. http://cseweb.ucsd.edu/users/russell/format.ps.

[Imp95] Russell Impagliazzo. A personal view of average-case complexity. In *Structure in Complexity Theory Conference*, pages 134–147. IEEE Computer Society, 1995.

[KNY14] Ilan Komargodski, Moni Naor, and Eylon Yogev. Secret-sharing for NP. *IACR Cryptology ePrint Archive*, 2014:213, 2014.

[MR13] Tal Moran and Alon Rosen. There is no indistinguishability obfuscation in pessiland. *IACR Cryptology ePrint Archive*, 2013:643, 2013.

[Ost91] Rafail Ostrovsky. One-way functions, hard on average problems, and statistical zero-knowledge proofs. In *Structure in Complexity Theory Conference*, pages 133–138. IEEE Computer Society, 1991.

[OV07] Shien Jin Ong and Salil P. Vadhan. Zero knowledge and soundness are symmetric. In *EUROCRYPT*, volume 4515 of *Lecture Notes in Computer Science*, pages 187–209. Springer, 2007.

[OW93] Rafail Ostrovsky and Avi Wigderson. One-way fuctions are essential for non-trivial zero-knowledge. In *ISTCS*, pages 3–17, 1993.

[PPS13] Omkant Pandey, Manoj Prabhakaran, and Amit Sahai. Obfuscation-based non-black-box simulation and four message concurrent zero knowledge for NP. *IACR Cryptology ePrint Archive*, 2013:754, 2013.

[PST14] Rafael Pass, Karn Seth, and Sidharth Telang. Indistinguishability obfuscation from semantically-secure multilinear encodings. In *CRYPTO (1)*, volume 8616 of *Lecture Notes in Computer Science*, pages 500–517. Springer, 2014.

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In *STOC*, pages 475–484. ACM, 2014.

A From Perfect $i\mathcal{O}$ to One-Way Functions

In this section we assume the existence of a perfect indistinguishability obfuscator. We show that assuming $\mathsf{NP} \not\subseteq \mathsf{io\text{-}coRP}$, one-way functions exist.

Observation 2.1 (Restated)**.** *Assume that* $\mathsf{NP} \not\subseteq \mathsf{io\text{-}coRP}$*. If there exists an efficient* perfect *indistinguishability obfuscator for* $\mathsf{3CNF}$ *formulas, then one-way functions exist.*

In the rest of this section we make the intuition given in Section 2 precise and prove Observation 2.1. Let $i\mathcal{O}(1^\lambda, C; r)$ be a *perfect* indistinguishability obfuscator for $\mathsf{3CNF}$s, where C is the input circuit and r the randomness used in obfuscation. As in Section 4 we denote by $Z_{s,n}$ a canonical constant zero circuit with inputs of n bits of size s gates. Moreover, for simplicity of writing, we assume that the size of the circuits s, the number of inputs n and the security parameter λ are all equal and denote them by n (otherwise, one could always increase the security parameter and add dummy inputs to make them equal).

For every $n \in \mathbb{N}$ define

$$f_n(x) \triangleq i\mathcal{O}(1^n, Z_{n,n}; x)$$

and let $\mathcal{F} = \{f_n : \{0,1\}^n \to \{0,1\}^*\}_{n\in\mathbb{N}}$ be the corresponding (efficiently computable) family of functions. Observation 2.1 follows immediately from the following lemma.

Lemma A.1. *If* $\mathsf{NP} \not\subseteq \mathsf{io\text{-}coRP}$*, then* $\mathcal{F}$ *is a family of one-way functions.*

Proof. Suppose, in contradiction, that $\mathcal{F}$ is not weakly one-way. Then there exists a probabilistic polynomial time adversary A who can invert f_n (for infinitely many n's) with probability $1-1/p(n)$ for any polynomial p. Let $f = f_n$. Since A cannot distinguish between an obfuscation of Z and any circuit C that is unsatisfiable we get that

$$\left|\Pr_{x,A}[A(1^n, i\mathcal{O}(C;x)) \in f^{-1}(x)] - \Pr_{x,A}[A(1^n, f(x)) \in f^{-1}(x)]\right| \leq \mathsf{neg}(n)$$

On the other hand, for any satisfiable circuit C, there will *never* be a pre-image under f. Thus we get that

$$\Pr_{x,A}[A(1^n, i\mathcal{O}(C;x)) \in f^{-1}(x)] = 0.$$

Given a $\mathsf{3CNF}$ instance C on n variables with n gates, we will now use A to (one-sided) probabilistically decide if C is satisfiable with very high probability:

Assume that C is unsatisfiable. Then, by the indistinguishability feature of $i\mathcal{O}$ and the inversion feature of A, we get that

$$\Pr[\mathsf{SolveSAT}(C) = \text{“UNSAT”}] \geq 1 - 1/p(n) - \mathsf{neg}(n) \geq 2/3.$$

On the other hand, when C is satisfiable, it is impossible to find an x' such that $f(x') = \widehat{C}$ and, hence

$$\Pr[\mathsf{SolveSAT}(C) = \text{“UNSAT”}] = 0.$$

□

The SolveSAT Algorithm

Input: A circuit C that gets inputs of length n.

1. Compute $\widehat{C} \leftarrow i\mathcal{O}(C)$.
2. Run $x' \leftarrow A(1^n, \widehat{C})$.
3. If $f(x') = \widehat{C}$ output "UNSAT"; Otherwise, output "SAT".

Figure 3: SolveSAT Algorithm from Perfect $i\mathcal{O}$.

B From Approximate $i\mathcal{O}$ to One-Way Functions

A natural variant of Definition 3.4 is to consider *approximate* indistinguishability obfuscators.[11] In this variant we require from $i\mathcal{O}$ the second and third requirements from Definition 3.4 (i.e., *polynomial slowdown* and *indistinguishability*) but replace the first requirement with the following:

1. (Approximate) Preserving Functionality:

   There exists a negligible function α such that for any input length $n \in \mathbb{N}$, any λ, any $C \in \mathcal{C}_n$ and every $x \in \{0,1\}^n$ it holds that

$$\Pr_{i\mathcal{O}}\left[C(x) = i\mathcal{O}(1^\lambda, C)(x)\right] \geq 1 - \alpha(\lambda).$$

We observe that by standard error amplification we have that if *approximate* indistinguishability obfuscators exist, then *imperfect* indistinguishability obfuscators exist. We note that the other direction is trivial.

Lemma B.1. *If there is an* approximate *indistinguishability obfuscator, then there exists an* imperfect *indistinguishability obfuscator, and vice-versa.*

As a corollary, we obtain that our main results (Theorems 4.1 and 5.1) are true even if we assume the existence of *approximate* $i\mathcal{O}$ instead of imperfect $i\mathcal{O}$.

Proof of Lemma B.1. One direction (from imperfect $i\mathcal{O}$ to approximate $i\mathcal{O}$) is trivial. We proceed with the other direction. Let approx-$i\mathcal{O}$ be an approximate $i\mathcal{O}$ algorithm as defined above. We construct an algorithm imperfect-$i\mathcal{O}$ that is an imperfect $i\mathcal{O}$ according to Definition 3.4. Given a circuit C as input to imperfect-$i\mathcal{O}$ it outputs a circuit which is described in Figure 4.

Our goal is to prove that imperfect-$i\mathcal{O}$ is an imperfect indistinguishability obfuscator. The first thing that we prove is the imperfect functionality feature of imperfect-$i\mathcal{O}$.

Claim B.2. imperfect-$i\mathcal{O}$ *is functionality preserving.*

[11] This definition is inspired by the definition of approximate virtual black-box obfuscation defined and studied by Barak et al. [BGI+12]. In that work, they also proved an impossibility result for general-purpose approximate virtual black-box obfuscators.

The imperfect-$i\mathcal{O}$ Algorithm

Input: A circuit C that gets inputs of length n.

Let approx-$i\mathcal{O}$ be an approximate $i\mathcal{O}$.

1. Compute n^2 independent obfuscation of C using approx-$i\mathcal{O}$: $C_1, \ldots, C_{n^2} \leftarrow \text{approx-}i\mathcal{O}(C)$.
2. Output a circuit that gets $x \in \{0,1\}^n$ as input and outputs the circuit that implements

$$\mathsf{MAJORITY}(C_1(x), C_2(x), \ldots, C_{n^2}(x)).$$

Figure 4: imperfect-$i\mathcal{O}$ Obfuscator.

Proof. Assume that for every circuit $C \in \mathcal{C}_n$ and every $x \in \{0,1\}^n$ it holds that

$$\Pr_{\text{approx-}i\mathcal{O}}[\text{approx-}i\mathcal{O}(C)(x) = C(x)] \geq 1 - \varepsilon(n).$$

Fix $x \in \{0,1\}^n$. Observe that, by Chernoff's bound, since we output the majority of n^2 independent executions, we get that for this x

$$\Pr_{\text{approx-}i\mathcal{O}}[\text{approx-}i\mathcal{O}(C)(x) \neq C(x)] \leq \varepsilon(n) \cdot 2^{-n},$$

which implies, by a union bound (since the latter is true for every $x \in \{0,1\}^n$), that

$$\Pr_{\text{imperfect-}i\mathcal{O}}[\forall x \in \{0,1\}^n : \text{ imperfect-}i\mathcal{O}(C)(x) \neq C(x)] \leq \varepsilon(n),$$

as required. □

Second, we show that imperfect-$i\mathcal{O}$ is still secure.

Claim B.3. imperfect-$i\mathcal{O}$ *is secure.*

Proof. The majority gate in the root of the output circuit of imperfect-$i\mathcal{O}$ is public and appears in *every* output of imperfect-$i\mathcal{O}$. Thus, all we are left to do is to show that the fact that the output of imperfect-$i\mathcal{O}$ consists of n independent obfuscations of the same circuit does not help an adversary trying to break the indistinguishability of our construction. This follows by a standard hybrid argument given for completeness in Claim B.4. □

This completes the proof of the lemma. □

Claim B.4. *Let $i\mathcal{O}$ be an efficient indistinguishability obfuscator. For any probabilistic polynomial time algorithm D, large enough security parameter $\lambda > 0$, any $n = \mathsf{poly}(\lambda)$ and any two functionally equivalent circuits C_1, C_2 of the same size it holds that*

$$|\Pr[D(\underbrace{i\mathcal{O}(1^\lambda, C_1), \ldots, i\mathcal{O}(1^\lambda, C_1)}_{n \text{ times}}) = 1] - \Pr[D(\underbrace{i\mathcal{O}(1^\lambda, C_2), \ldots, i\mathcal{O}(1^\lambda, C_2)}_{n \text{ times}}) = 1]| \leq \mathsf{neg}(\lambda),$$

where $i\mathcal{O}(1^\lambda, C), \ldots, i\mathcal{O}(1^\lambda, C)$ are n independent *executions of $i\mathcal{O}$ on the circuit C with a security parameter λ.*

Proof. Assume that there exists a polynomial-time algorithm D and some ε such that

$$|\Pr[D(\underbrace{i\mathcal{O}(1^\lambda, C_1), \ldots, i\mathcal{O}(1^\lambda, C_1)}_{n \text{ times}}) = 1] - \Pr[D(\underbrace{i\mathcal{O}(1^\lambda, C_2), \ldots, i\mathcal{O}(1^\lambda, C_2)}_{n \text{ times}}) = 1]| \geq \varepsilon, \tag{B.1}$$

For $\sigma \in \{1, 2\}$ let $i\mathcal{O}_\sigma$ be a random variable sampled according to the distribution $i\mathcal{O}(C_\sigma)$. With this notation, eq. (B.1) can be rewritten as

$$\Big|\Pr[D(i\mathcal{O}_1, \ldots, i\mathcal{O}_1) = 1] - \Pr[D(i\mathcal{O}_2, \ldots, i\mathcal{O}_2) = 1]\Big| \geq \varepsilon. \tag{B.2}$$

For $0 \leq i \leq n$ let $\mathcal{C}^{(i)}$ be the distribution induced by the sequence $\underbrace{i\mathcal{O}_1, \ldots, i\mathcal{O}_1}_{n-i \text{ times}}, \underbrace{i\mathcal{O}_2, \ldots, i\mathcal{O}_2}_{i \text{ times}}$. Using this notation, eq. (B.2) can be rewritten as

$$\Big|\Pr[D(\mathcal{C}^{(0)}) = 1] - \Pr[D(\mathcal{C}^{(n)}) = 1]\Big| \geq \varepsilon.$$

By a hybrid argument, there exists an index $i \in [n]$ for which

$$\Big|\Pr[D(\mathcal{C}^{(i-1)}) = 1] - \Pr[D(\mathcal{C}^{(i)}) = 1]\Big| \geq \varepsilon/n.$$

Expanding the definition of $\mathcal{C}^{(i)}$,

$$\begin{aligned}|\Pr[D(\underbrace{i\mathcal{O}_1, \ldots, i\mathcal{O}_1}_{n-i \text{ times}}, i\mathcal{O}_2, \underbrace{i\mathcal{O}_2, \ldots, i\mathcal{O}_2}_{i-1 \text{ times}}) = 1] - \\ \Pr[D(\underbrace{i\mathcal{O}_1, \ldots, i\mathcal{O}_1}_{n-i-1 \text{ times}}, i\mathcal{O}_1, \underbrace{i\mathcal{O}_2, \ldots, i\mathcal{O}_2}_{i \text{ times}}) = 1]| \geq \varepsilon/n.\end{aligned}$$

At this point, it follows that there exists D' that distinguishes between $i\mathcal{O}_1$ and $i\mathcal{O}_2$. Namely, it holds that

$$\Big|\Pr[D'(i\mathcal{O}(C_1)) = 1] - \Pr[D'(i\mathcal{O}(C_2)) = 1]\Big| \geq \varepsilon/n,$$

as required. □