2014 IEEE Annual Symposium on Foundations of Computer Science

One-Way Functions and (Im)perfect Obfuscation

Ilan Komargodski*, Tal Moran†, Moni Naor*, Rafael Pass‡, Alon Rosen† and Eylon Yogev*
*Weizmann Institute of Science, Rehovot, Israel,
Email: {ilan.komargodski,moni.naor,eylon.yogev}@weizmann.ac.il
†IDC, Herzliya, Israel,
Email: {talm, alon.rosen}@idc.ac.il
‡Cornell University, Ithaca, NY,
Email: rafael@cs.cornell.edu

***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.

Ilan Komargodski, Moni Naor and Eylon Yogev are supported in part by grants 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.

Tal Moran is supported by ISF grant no. 1790/13 and by the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 293843

Rafael Pass is 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.

Alon Rosen is 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.

I. 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. [1], [2]. 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. [2] 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 [3], [4], [5], [6], [7], [8] and second, intriguing applications of $i\mathcal{O}$ have been demonstrated, e.g., general-purpose functional encryption scheme [3], deniable encryption with negligible advantage [9], two-round secure MPC [10], traitor-tracing schemes with very short messages [11], secret-sharing for NP [12] 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. [2], if $\mathsf{P} = \mathsf{NP}$, then there are no

[1]Two notable exceptions are *witness encryption* [13] and *functional witness encryption* [14]. However, Boyle et al. [14] showed that these can be viewed as special cases of $i\mathcal{O}$.

0272-5428/14 $31.00 © 2014 IEEE
DOI 10.1109/FOCS.2014.47

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 $\mathsf{NP} \not\subseteq \mathsf{io\text{-}BPP}$ 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 [15] which states that SKZ-arguments for hard-on-the average languages implies the 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 [2] 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.*

[2]If $\mathsf{P} = \mathsf{NP}$, 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 $\mathsf{NP} \not\subseteq \mathsf{io\text{-}coRP}$ implies one-way functions. See Section II for further details.

[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).

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 [9] and the construction of a traitor-tracing scheme of Boneh and Zhandry [11]. In addition, we view our results as making the claim of Sahai and Waters [9] that $i\mathcal{O}$ is a "central hub" of cryptography more cohesive.

Borrowing from Impagliazzo's terminology [16], 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 [2], Goldwasser and Kalai [17] and Goldwasser and Rothblum [18] 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. [3] 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 [4], [5], [7], and [6] 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 II we give a high level overview of our main techniques. In Section III we provide preliminary definitions and set up notation. In Sections IV to VI we prove Items 1 to 3 of our main theorem, respectively. In Section VII 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 $i\mathcal{O}$.

II. 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 III.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 II.1. *Assume that* NP $\not\subseteq$ io-coRP. *If there exists an efficient* perfect *indistinguishability obfuscator for* 3CNF *formulas, then one-way functions exist.*

The idea behind the proof of Observation II.1 is simple and borrows the construction from [18, 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

$$f(x) = i\mathcal{O}(Z;x), \tag{1}$$

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 II.1 can be found in [19]. We note that the intuition above (as well as the formal proof in [19]) 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 II.2. *Observe that all we need for Observation II.1 is an indistinguishability obfuscator for* 3CNF formulas. *However, for Item 1 of our main theorem (see Theorem IV.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 IV.1 but only relying on an obfuscator for* 3CNF*s.*

A. Going Beyond Perfect $i\mathcal{O}$

As we noted above, the simple construction given in Equation (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 $\widehat{C}$ 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 IV.4) is that if C is a *satisfiable* circuit, then with high probability it holds that

$$|\varphi(C,\widehat{Z})| \ll |\varphi(Z,\widehat{Z})|. \tag{2}$$

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

$$|\varphi(C,\widehat{Z})| \approx |\varphi(Z,\widehat{Z})|. \tag{3}$$

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 checking whether the latter two are roughly the same or not.

However, the complementary inequality (Equation (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 $\widehat{\widehat{C}}$ the *double* obfuscation of a circuit C (i.e., $\widehat{\widehat{C}} \leftarrow i\mathcal{O}(i\mathcal{O}(C))$). By the functionality property of $i\mathcal{O}$,

[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.

a natural corollary of Equation (2) is that for a satisfiable circuit C with high probability it holds that

$$|\varphi(\widehat{C},\widehat{\widehat{Z}})| \ll |\varphi(\widehat{Z},\widehat{\widehat{Z}})|. \tag{4}$$

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 $i\mathcal{O}$, the complementary inequality of this corollary *is* true! That is, if C is *unsatisfiable*, then with high probability it holds that

$$|\varphi(\widehat{C},\widehat{\widehat{Z}})| \approx |\varphi(\widehat{Z},\widehat{\widehat{Z}})|. \tag{5}$$

Indeed, 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 $\widehat{Z}$.

At this point, we can decide satisfiability of a given circuit C by computing $\widehat{Z} \leftarrow i\mathcal{O}(Z), \widehat{\widehat{Z}} \leftarrow i\mathcal{O}(i\mathcal{O}(Z)), \widehat{C} \leftarrow i\mathcal{O}(C)$, $\varphi_{\approx}(\widehat{C},\widehat{\widehat{Z}})$ and $\varphi_{\approx}(\widehat{Z},\widehat{\widehat{Z}})$, and checking whether the latter two are roughly the same or not. 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 [20] (see also [21]). Recall Equation (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

$$f'(x,h,k) = Z \circ i\mathcal{O}(Z;x) \circ h \circ h(x)|_k, \tag{6}$$

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 [22], [23], 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

$$f'_C(x,h,k) = C \circ i\mathcal{O}(C;x) \circ h \circ h(x)|_k. \tag{7}$$

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.

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 separating by φ, yielding a uniform estimator $\varphi_{\approx}$ for the measures in Equations (4) and (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:

$$f''(x,y,h,k) = \\ i\mathcal{O}(Z;y) \circ i\mathcal{O}(i\mathcal{O}(Z;y);x) \circ h \circ h(x)|_k, \tag{8}$$

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 (4) and (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 IV.

III. 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 $\log 0 = 0$.

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

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.

A. Computational Indistinguishability

Definition III.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.

B. One-Way Functions

Definition III.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 III.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'.

C. 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 III.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 III.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.

D. Leftover Hash Lemma

Definition III.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 III.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 [24, Theorem D.5].

Theorem III.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,*

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

IV. 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 $\mathsf{NP} \not\subseteq$ io-BPP, then one-way functions exist.

Theorem IV.1. *Assume that $\mathsf{NP} \not\subseteq$ io-BPP. If there exists an efficient (even imperfect) indistinguishability obfuscator for polynomial-size circuits, then one-way functions exist.*

To prove Theorem IV.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., [25, 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 $\mathsf{NP} \subseteq$ io-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 III.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

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

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 II. 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 IV.2 - We prove that there exists a procedure that gives a good estimation for $|\varphi(\widehat{C}, \widehat{\widehat{Z}})|$ (see Lemma IV.2 for the exact details). This result uses the assumption that f is not one-way in a very strong way.
2) Lemma IV.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 high probability for every *unsatisfiable* circuit it holds that

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

[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.

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

3) Corollary IV.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 check if they are close or far, and answer accordingly.

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 IV.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. The procedure $\varphi_{\approx}$ from Figure 1 (that gets as input $\widehat{C}$ and $\widehat{\widehat{Z}}$) satisfies that with probability at least $1-1/n^{10}$ it holds 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* $\varphi_{\approx}(\widehat{C},\widehat{\widehat{Z}}) \geq \log|\varphi(\widehat{C},\widehat{\widehat{Z}})| - 90\log n$,

where the probability is over $\widehat{C},\widehat{\widehat{Z}}$ and the internal randomness of $\varphi_{\approx}$.

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.

The proof of Lemma IV.2 can be found in the full version [19].

Next, we show that for every two unsatisfiable circuits Z and C with high probability $|\varphi(\widehat{C},\widehat{\widehat{Z}})|$ and $|\varphi(\widehat{Z},\widehat{\widehat{Z}})|$ are roughly the same.

Lemma IV.3. *Let Z and C be any two unsatisfiable circuits that are of the same size. Let $\widehat{Z} \leftarrow i\mathcal{O}(Z)$, $\widehat{C} \leftarrow i\mathcal{O}(C)$ and $\widehat{\widehat{Z}} \leftarrow i\mathcal{O}(i\mathcal{O}(Z))$ be random variables. Then, with probability $1-\mathsf{neg}(n)$ over the internal randomness of $i\mathcal{O}$ it holds that*

$$\left|\log|\varphi(\widehat{C},\widehat{\widehat{Z}})| - \log|\varphi(\widehat{Z},\widehat{\widehat{Z}})|\right| \leq 400\log n.$$

Proof: Assume towards a contradiction that the claim is false. That is, there is a polynomial $q(\cdot)$ such that with probability $1/q(n)$ it holds that $|\log|\varphi(\widehat{C},\widehat{\widehat{Z}})| - \log|\varphi(\widehat{Z},\widehat{\widehat{Z}})|| > 400\log n$.

We show that we can use the procedure $\varphi_{\approx}$ from Lemma IV.2 to distinguish an obfuscation of C from an obfuscation of Z with high probability. According to the security guarantee of $i\mathcal{O}$ (see Item 3 of Definition III.4), the latter is a contradiction. Recall that the procedure $\varphi_{\approx}$ from Lemma IV.2 is given two circuits as input: $\widehat{C} \leftarrow i\mathcal{O}(C)$ and $\widehat{Z} \leftarrow i\mathcal{O}(i\mathcal{O}(Z))$.

Given Z and C as in the claim, and an obfuscation of one of them $\widehat{W}$, we define a procedure Break-i$\mathcal{O}$ that is able to decide whether this obfuscation is an obfuscation of Z or of C with non-negligible probability. The procedure Break-i$\mathcal{O}$ is defined as follows: First, it samples $\widehat{Z} \leftarrow i\mathcal{O}(Z)$ and $\widehat{\widehat{Z}} \leftarrow i\mathcal{O}(i\mathcal{O}(Z))$. Then it uses $\varphi_{\approx}$ to estimate $K_Z = \varphi_{\approx}(\widehat{Z},\widehat{\widehat{Z}})$ and $K_W = \varphi_{\approx}(\widehat{W},\widehat{\widehat{Z}})$ and output "Z'' if and only if these two estimates are close. The formal description of Break-i$\mathcal{O}$ is given in Figure 2.

The Break-i$\mathcal{O}$ Procedure

Input: An obfuscation $\widehat{W}$ of either Z or C.

Let $\varphi_{\approx}$ be the procedure from Lemma IV.2.

1) Sample $\widehat{Z} \leftarrow i\mathcal{O}(Z)$, $\widehat{\widehat{Z}} \leftarrow i\mathcal{O}(i\mathcal{O}(Z))$.
2) Compute $K_Z \leftarrow \varphi_{\approx}(\widehat{Z},\widehat{\widehat{Z}})$.
3) Compute $K_W \leftarrow \varphi_{\approx}(\widehat{W},\widehat{\widehat{Z}})$.
4) If $|K_W - K_Z| \leq 200\log n$, then output "Z"; Otherwise, output "C".

Figure 2. Break-i$\mathcal{O}$ Procedure.

If $\widehat{W}$ is an obfuscation of Z, then by Lemma IV.2 with probability $1-1/n^9$ it holds that

$$|K_Z - K_W| \leq \log|\varphi(\widehat{Z},\widehat{\widehat{Z}})| + 90\log n - (\log|\varphi(\widehat{Z},\widehat{\widehat{Z}})| - 90\log n) = 180\log n.$$

Therefore, in this case, Break-i$\mathcal{O}$ will output "Z" with high probability, as required.

If $\widehat{W}$ is an obfuscation of C, then by Lemma IV.2 and the assumption, with probability $1-1/n^9-1/q(n)$ it holds that

$$|K_Z - K_W| \geq \left|\log|\varphi(\widehat{Z},\widehat{\widehat{Z}})| - \log|\varphi(\widehat{C},\widehat{\widehat{Z}})|\right| - 180\log n \geq 400\log n - 180\log n > 200\log n.$$

Therefore, in this case, Break-i$\mathcal{O}$ will output "C". In conclusion, Break-i$\mathcal{O}$ can distinguish between the obfuscations of C and Z with high probability, breaking the security guarantee of $i\mathcal{O}$. ∎

Next, we show that for every circuit C that is functionally different from the zero circuit Z, with high probability $|\varphi(C,\widehat{Z})|$ is much smaller than $|\varphi(Z,\widehat{Z})|$.

Lemma IV.4. *For any two non-equivalent circuits Z, C and for any polynomial $p(\cdot)$ it holds that*

$$\Pr_r\left[p(n)\cdot|\varphi(C,\widehat{Z}_r)| < |\varphi(Z,\widehat{Z}_r)|\right] > 1-\mathsf{neg}(n).$$

Proof: Let $p(\cdot)$ be a polynomial. Assume towards contradiction that there exists a polynomial $q(\cdot)$ such that

$$\Pr_r\left[p(n)\cdot|\varphi(C,\widehat{Z}_r)| \geq |\varphi(Z,\widehat{Z}_r)|\right] \geq \frac{1}{q(n)}. \quad (9)$$

Denote by Bad the set of r's for which $p(n)\cdot|\varphi(C,\widehat{Z}_r)| \geq |\varphi(Z,\widehat{Z}_r)|$. By Equation (9) we have that $\Pr_r[r\in\mathsf{Bad}] \geq 1/q(n)$. From the completeness of $i\mathcal{O}$ we have that $\Pr_r[r\in \mathsf{Bad} \wedge \widehat{Z}_r \equiv Z] \geq 1/q(n) - \mathsf{neg}(n)$. Denote by Bad' the set of all $r\in\mathsf{Bad}$ for which $\widehat{Z}_r \equiv Z$. In particular, for any $r\in\mathsf{Bad}'$ it holds that $|\{y\in\{0,1\}^n \mid i\mathcal{O}(C;y)=\widehat{Z}_r\}| \geq |\{y\in\{0,1\}^n \mid i\mathcal{O}(Z;y)=\widehat{Z}_r\}|/p(n)$. Then,

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

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 IV.5. *Let C and Z be two non-equivalent circuits of the same size. Let $\widehat{C}\leftarrow i\mathcal{O}(C)$, $\widehat{Z}\leftarrow i\mathcal{O}(Z)$ and $\widehat{\widehat{Z}}\leftarrow i\mathcal{O}(i\mathcal{O}(Z))$. For any positive constant $c\in\mathbb{N}$, with probability at least $1-\mathsf{neg}(n)$ it holds that $c\log n + \log|\varphi(\widehat{C},\widehat{\widehat{Z}})| < \log|\varphi(\widehat{Z},\widehat{\widehat{Z}})|$.*

A. Proof of Theorem IV.1

We prove Theorem IV.1 by showing how to combine Lemmas IV.2 and IV.3 and Corollary IV.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 $\Pr_{\mathsf{SolveSAT}}[\mathsf{SolveSAT}(C) = \text{"SAT"}] \geq 2/3$.
2) If C is unsatisfiable, then $\Pr_{\mathsf{SolveSAT}}[\mathsf{SolveSAT}(C) = \text{"UNSAT"}] \geq 2/3$.

$\mathsf{SolveSAT}$ samples $\widehat{Z}\leftarrow i\mathcal{O}(Z)$, $\widehat{\widehat{Z}}\leftarrow i\mathcal{O}(i\mathcal{O}(Z))$ 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 large, then it outputs "SAT". The formal description appears in Figure 3.

The $\mathsf{SolveSAT}$ Procedure

Input: A circuit C that receives n inputs.

Let $\varphi_\approx$ be the procedure from Lemma IV.2.

1) Sample $\widehat{Z}\leftarrow i\mathcal{O}(Z)$, $\widehat{\widehat{Z}}\leftarrow i\mathcal{O}(i\mathcal{O}(Z))$ 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 $K_Z-K_C > 600\log n$, output "SAT"; Otherwise, output "UNSAT".

Figure 3. SAT Solver.

By Lemma IV.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 IV.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 IV.3 we also know that with probability $1-\mathsf{neg}(n)$

$$|\varphi(\widehat{Z},\widehat{\widehat{Z}}) - \varphi(\widehat{C},\widehat{\widehat{Z}})| \leq 400\log n.$$

Therefore, using the triangle inequality, with probability $1-1/n^9$ it holds that

$$K_Z - K_C \leq 600\log n.$$

Thus, in this case, $\mathsf{SolveSAT}$ outputs "UNSAT" with high probability, as required.

Next, assume that C is a satisfiable circuit. Using Corollary IV.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 IV.2 we have that with probability at least $1-1/n^9$ it holds that

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

Therefore, in this case, $\mathsf{SolveSAT}$ outputs "SAT" with high probability, as required.

V. 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 V.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 V.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 V.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 V.1 follows by combining Theorem V.2 with a result of Ostrovsky [15] - showing that honest-verifier statistical zero-knowledge arguments for hard-on-the average languages implies the existence of one-way functions.[8] The proof of Theorem V.2 can be found in the full version [19].

Remark: In the proof of Theorem V.2, the only thing we require from C is that it is a "witness encryption" [13] (at least according to the definition from [14]) 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.

VI. 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. [2, 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 VI.1. *If an efficient, even* imperfect*, VBB obfuscator for point functions exists, then one-way functions exist.*

[8] Alternatively, using a result of Ostrovsky and Wigderson [26], 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 IV.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).

The proof of Theorem VI.1 can be found in the full version [19].

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

A natural variant of Definition III.4 is to consider *approximate* indistinguishability obfuscators.[9] In this variant we require from $i\mathcal{O}$ the second and third requirements from Definition III.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 VII.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 IV.1 and V.1) are true even if we assume the existence of *approximate* $i\mathcal{O}$ instead of imperfect $i\mathcal{O}$. The proof of Lemma VII.1 can be found in the full version [19].

Acknowledgment

We thank Nir Bitansky, Zvika Brakerski and Ron Rothblum for many helpful discussions.

References

[1] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, and K. Yang, "On the (im)possibility of obfuscating programs," in *CRYPTO*, ser. Lecture Notes in Computer Science, vol. 2139. Springer, 2001, pp. 1–18.

[2] ——, "On the (im)possibility of obfuscating programs," *Journal of the ACM*, vol. 59, no. 2, p. 6, 2012, preliminary version [1].

[3] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters, "Candidate indistinguishability obfuscation and functional encryption for all circuits," in *FOCS*, 2013, pp. 40–49.

[4] Z. Brakerski and G. N. Rothblum, "Virtual black-box obfuscation for all circuits via generic graded encoding," in *TCC*, 2014, pp. 1–25.

[9] This definition is inspired by the definition of approximate virtual black-box obfuscation defined and studied by Barak et al. [2]. In that work, they also proved an impossibility result for general-purpose approximate virtual black-box obfuscators.

[5] B. Barak, S. Garg, Y. T. Kalai, O. Paneth, and A. Sahai, "Protecting obfuscation against algebraic attacks," in *EUROCRYPT*, ser. Lecture Notes in Computer Science, vol. 8441. Springer, 2014, pp. 221–238.

[6] R. Pass, K. Seth, and S. Telang, "Indistinguishability obfuscation from semantically-secure multilinear encodings," in *CRYPTO (1)*, ser. Lecture Notes in Computer Science, vol. 8616. Springer, 2014, pp. 500–517.

[7] P. Ananth, D. Gupta, Y. Ishai, and A. Sahai, "Optimizing obfuscation: Avoiding barringtons theorem," *IACR Cryptology ePrint Archive*, vol. 2014, p. 222, 2014.

[8] C. Gentry, A. B. Lewko, A. Sahai, and B. Waters, "Indistinguishability obfuscation from the multilinear subgroup elimination assumption," *IACR Cryptology ePrint Archive*, vol. 2014, p. 309, 2014.

[9] A. Sahai and B. Waters, "How to use indistinguishability obfuscation: deniable encryption, and more," in *STOC*. ACM, 2014, pp. 475–484.

[10] S. Garg, C. Gentry, S. Halevi, and M. Raykova, "Two-round secure mpc from indistinguishability obfuscation," in *TCC*, ser. Lecture Notes in Computer Science, vol. 8349. Springer, 2014, pp. 74–94.

[11] D. Boneh and M. Zhandry, "Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation," in *CRYPTO (1)*, ser. Lecture Notes in Computer Science, vol. 8616. Springer, 2014, pp. 480–499.

[12] I. Komargodski, M. Naor, and E. Yogev, "Secret-sharing for NP," *IACR Cryptology ePrint Archive*, vol. 2014, p. 213, 2014.

[13] S. Garg, C. Gentry, A. Sahai, and B. Waters, "Witness encryption and its applications," in *STOC*. ACM, 2013, pp. 467–476.

[14] E. Boyle, K.-M. Chung, and R. Pass, "On extractability obfuscation," in *TCC*, ser. Lecture Notes in Computer Science, vol. 8349. Springer, 2014, pp. 52–73.

[15] R. Ostrovsky, "One-way functions, hard on average problems, and statistical zero-knowledge proofs," in *Structure in Complexity Theory Conference*. IEEE Computer Society, 1991, pp. 133–138.

[16] R. Impagliazzo, "A personal view of average-case complexity," in *Structure in Complexity Theory Conference*. IEEE Computer Society, 1995, pp. 134–147.

[17] S. Goldwasser and Y. T. Kalai, "On the impossibility of obfuscation with auxiliary input," in *FOCS*. IEEE Computer Society, 2005, pp. 553–562.

[18] S. Goldwasser and G. N. Rothblum, "On best-possible obfuscation," in *TCC*, ser. Lecture Notes in Computer Science, vol. 4392. Springer, 2007, pp. 194–213.

[19] I. Komargodski, T. Moran, M. Naor, R. Pass, A. Rosen, and E. Yogev, "One-way functions and (im)perfect obfuscation," *IACR Cryptology ePrint Archive*, vol. 2014, p. 347, 2014.

[20] R. Impagliazzo and M. Luby, "One-way functions are essential for complexity based cryptography (extended abstract)," in *FOCS*. IEEE Computer Society, 1989, pp. 230–235.

[21] R. Impagliazzo, "Pseudo-random generators for cryptography and for randomized algorithms," Ph.D. dissertation, University of California, Berkeley, 1992, http://cseweb.ucsd.edu/users/russell/format.ps.

[22] R. Impagliazzo, L. A. Levin, and M. Luby, "Pseudo-random generation from one-way functions (extended abstracts)," in *STOC*. ACM, 1989, pp. 12–24.

[23] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby, "A pseudorandom generator from any one-way function," *SIAM J. Comput.*, vol. 28, no. 4, pp. 1364–1396, 1999.

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

[25] ——, *The Foundations of Cryptography - Volume 1, Basic Techniques*. Cambridge University Press, 2001.

[26] R. Ostrovsky and A. Wigderson, "One-way fuctions are essential for non-trivial zero-knowledge," in *ISTCS*, 1993, pp. 3–17.