Lower Bounds on the Overhead of Indistinguishability Obfuscation

Zhenjian Lu* Noam Mazor† Igor C. Oliveira‡ Rafael Pass§

Abstract

We consider indistinguishability obfuscation (i$\mathcal{O}$) for multi-output circuits $C\colon \{0,1\}^n \to \{0,1\}^n$ of size s, where s is the number of AND/OR/NOT gates in C. Under the worst-case assumption that $\mathsf{NP} \not\subseteq \mathsf{BPP}$, we establish that there is no efficient indistinguishability obfuscation scheme that outputs circuits of size $s + o(s/\log s)$. In other words, to be secure, an efficient i$\mathcal{O}$ scheme must incur an $\Omega(s/\log s)$ additive overhead in the size of the obfuscated circuit. The hardness assumption under which this negative result holds is minimal since an optimal i$\mathcal{O}$ scheme with no circuit size overhead exists if $\mathsf{NP} \subseteq \mathsf{BPP}$.

Expanding on this result, we also rule out i$\mathcal{O}$ for single-output database-aided circuits with an arbitrary polynomial overhead in circuit size. This strengthens an impossibility result by Goldwasser and Rothblum [GR07], which considered circuits with access to an exponential-length database that the obfuscator has oracle access to; in contrast, our impossibility result holds even w.r.t. polynomial-size databases and even w.r.t. obfuscators that may run in time polynomial in the size of the database (and thus may read the whole database).

The proof of our main result builds on a connection between obfuscation and meta-complexity put forward by Mazor and Pass [MP24], and on the NP-hardness of circuit minimization for multi-output circuits established by Loff, Ilango, and Oliveira [ILO20], together with other techniques from cryptography and complexity theory.

*University of Warwick. E-mail: zhenjian.lu@warwick.ac.uk

†Tel Aviv University. E-mail: noammaz@gmail.com

‡University of Warwick. E-mail: igor.oliveira@warwick.ac.uk

§Cornell Tech, Technion, and Tel Aviv University. E-mail: rafael@cs.cornell.edu

Contents

1 Introduction

The goal of program obfuscation is to "scramble" a computer program, hiding its implementation details (making it hard to "reverse-engineer"), while preserving its functionality (i.e., its input/output behavior). During the last decade, the notion of *indistinguishability obfuscation* ($i\mathcal{O}$) [BGI+12, GGH+13] has emerged as a powerful, and useful, way of defining the security of program obfuscation. Roughly speaking, this notion of obfuscation requires that for any two functionally equivalent programs C_1 and C_2 (viewed in this work as Boolean circuits), their obfuscated versions are computationally indistinguishable. This concept was first introduced by Barak et al. [BGI+12], with the first general-purpose $i\mathcal{O}$ candidate proposed by Garg et al. [GGH+13]. On the one hand, this notion strong enough to imply both a wide range of standard cryptographic primitives (from the worst-case assumption that $\mathsf{NP} \not\subseteq \mathsf{P/poly}$) and furthemore enables the construction of various more advanced tasks for which other instantations are not known, such as functional encryption [GGH+13], software watermarking [CHN+18], deniable encryption [SW21], among many others (see [JLS21, BCP14, BZ17, GGHR14, KNY17, BGL+15, CHJV14, KLW15, CLP15, BPR15, BPW15, BP15] and references therein).

The initial $i\mathcal{O}$ construction proposed by Garg et al. [GGH+13] was based on a computational hardness assumption related to multilinear maps, which was later shown to be insecure. Building on a sequence of subsequent works ([PST14, GLSW15, Lin16, LV16, Lin17, LT17, AJS18, JLMS19, AJL+19, GJLS21]), Jain, Lin, and Sahai [JLS21, JLS22] introduced a celebrated construction based on assumptions that have been well-studied by cryptographers (see [JLS24] for an overview). Additional constructions that rely on a variety of (more heuristic) assumptions, but with simpler constructions, were suggested by [BDGM23, GP21, BDGM22, GJK18, BIJ+20, Agr19, APM20, CCMR24].

How much overhead is needed to obfuscate a circuit? Alongside the research on the theoretical foundations of indistinguishability obfuscation, considerable efforts have been made to propose mechanisms that enhance the efficiency of $i\mathcal{O}$ schemes. A key measure of this efficiency is the overhead in the circuit size of the obfuscated circuit. In this context, ideally we would like to ensure the obfuscation of a circuit C of size s is a new circuit $\tilde{C}$ with size $s + \mathsf{poly}(n)$; we will refer to such obfuscators are *rate*-1, or simply *near-optimal*. (We highlight that the study of the overhead of the $i\mathcal{O}$ is not just a question of theoretical interest; it directly impacts the efficiency of the cryptographic applications of $i\mathcal{O}$.) Indeed, for many cryptographic primitives, such as e.g., secret and public-key encryption schemes, rate-1 construction have been known for decades [Gol04], and more recently, rate-1 construction have also been developed for more advanced cryptographic primitives like fully-homomorphic encryption (under standard cryptographic assumptions) [BDGM19].

Towards addresssing this problem, Bitansky and Vaikuntanathan [BV15] and Ananth, Jain, and Sahai [AJS17] investigated the possibility of achieving $i\mathcal{O}$ with constant multiplicative overhead in size. In particular, [AJS17] proposed a general bootstrapping mechanism able to achieve obfuscated programs of length $2 \cdot s + \mathsf{poly}(n)$ on an input circuit of length s.[1] Constant-size overhead is also known to be achievable in the context of obfuscation for Turing machines [AJS17] and RAM programs [JLL23]. These constructions suggest the possibility that near-optimal $i\mathcal{O}$ schemes may be possible.

Motivated by these results, we consider the following basic question:

What is the minimal overhead required to obfuscate a program?

[1] We note that their notion of size seems to depend to the description length of a representation of the circuit, which is different than the standard notion of size using number of gates that we consider in this work.

Our main contribution is the first negative result showing limits on the size efficiency of i$\mathcal{O}$ schemes for general circuits. Moreover, we obtain our impossibility result under a minimal hardness assumption. In essence, our results will show—in a number of settings—that:

"Low overhead i$\mathcal{O}$*" exists if and only if we live in Algorithmica (i.e., if* NP *is easy).*

In the next section, we describe our contributions in detail.

1.1 Results

As alluded to above, our results show that, unless $\mathsf{NP} \subseteq \mathsf{BPP}$, any efficient procedure that securely obfuscates a program must output a program with a larger number of instructions. Before formally stating our results, we introduce appropriate notation.

Notation. We consider i$\mathcal{O}$ schemes for multi-output Boolean circuits $C\colon \{0,1\}^\ell \to \{0,1\}^\ell$, where $\ell \geq 1$ is arbitrary. We measure the size s of C by its number of (fan-in two) AND, OR, and NOT gates (see Section 2.1 for an example). We write $\mathsf{i}\mathcal{O}(1^\lambda, C)$ to denote the obfuscation of C, where λ is the security parameter. We consider two standard variants of i$\mathcal{O}$ in our results. In a perfect i$\mathcal{O}$ scheme, the obfuscated circuit $\mathsf{i}\mathcal{O}(1^\lambda, C)$ is functionally equivalent to C with probability 1 over the internal randomness of the i$\mathcal{O}$ procedure. On the other hand, in an imperfect i$\mathcal{O}$ scheme, we allow a negligible probability that $\mathsf{i}\mathcal{O}(1^\lambda, C)$ and C compute different functions. We consider security against non-uniform circuits. The definition of i$\mathcal{O}$ is reviewed in Section 2.

Impossibility of Near-Optimal Obfuscation. We establish the following hardness results on the size efficiency of indistinguishability obfuscation.

Theorem 1 (Hardness of Near-Optimal Imperfect i$\mathcal{O}$)**.** *There exists a universal constant $b > 0$ such that the following statements are equivalent.*

1. $\mathsf{NP} \not\subseteq \mathsf{BPP}$.
2. *There exists no imperfect indistinguishability obfuscator for multi-output circuits with output size σ, for any constant $c > 0$ and $\sigma(\lambda, s) = s + s/(b \cdot \log s) + \lambda^c$.*

Theorem 2 (Hardness of Near-Optimal Perfect i$\mathcal{O}$)**.** *There exists a universal constant $b > 0$ such that the following statements are equivalent.*

1. $\mathsf{NP} \not\subseteq \mathsf{ZPP}$.
2. *There exists no perfect indistinguishability obfuscator for multi-output circuits with output size σ, for any constant $c > 0$ and $\sigma(\lambda, s) = s + s/(b \cdot \log s) + \lambda^c$.*

Since we obtain an equivalence in each case, we rule out near-optimal size-efficient i$\mathcal{O}$ under a minimal worst-case assumption. We note that, under the stronger assumption that one-way functions exist, our argument rules out i$\mathcal{O}$ with output size $s + s^{1-o(1)} + \mathsf{poly}(\ell', \lambda)$ for input circuits $C\colon \{0,1\}^{\ell'} \to \{0,1\}^\ell$ of size $s = \ell^{\Theta(1)}$ and input length $\ell' = O(\log \ell)$, i.e., even when the number of input bits is exponentially smaller than the size of the circuit. This can be seen as a lower bound on the overhead of (multi-output) Xi$\mathcal{O}$ [LPST16] – a weaker kind of obfuscator that is allowed to run in polynomial time in the length of the truth table of the circuit (rather than the size of the circuit).

From Near-Optimal to Optimal i$\mathcal{O}$. An immediate consequence of the results above is that imperfect computationally secure i$\mathcal{O}$ with small size overhead yields the easiness of NP. We note that before our results it was not even clear if statically secure i$\mathcal{O}$ with no size overhead implied the easiness of NP. Using that optimal i$\mathcal{O}$ schemes with no circuit size overhead exist under the worst-case easiness of NP, we derive the following unexpected consequence from Theorem 1 and Theorem 2.

Corollary 3 (Bootstrapping Near-Optimal i$\mathcal{O}$ to Optimal i$\mathcal{O}$)**.** *If for some constant $c > 0$ there is a perfect (resp. imperfect) indistinguishability obfuscation scheme for multi-output circuits with output size $\sigma(\lambda, s) = s + s/(b \cdot \log s) + \lambda^c$, where b is the constant in Theorem 2 (resp. Theorem 1), then there is a perfect (resp. imperfect) indistinguishability obfuscation scheme for multi-output circuits with output size $\sigma(\lambda, s) = s$.*

Our results can be extended to other computational models (see Section 7) and are robust to the set of gates employed by the circuits. However, we currently do not know how to establish them for single-output Boolean circuits nor for multi-output circuits where we allow unbounded fan-in gates when measuring circuit size. These are interesting directions for further research (see also Section 1.4).

Next, we present a natural setting where we can prove negative results for single-output circuits and with respect to an arbitrarily large polynomial overhead in the size of the input circuit.

Impossibility of Obfuscation for (Single-Output Bit) Circuits with Data-Access. In real-life applications of program obfuscation, it would be desirable to be able to obfuscate also a program that has access to some "external" (large) database. Of course, a simple way to do this is to simply include the database as part of the description of the program and obfuscate the new program, but this would induce a huge overhead (e.g., requiring to duplicate the external database in an obfuscated form). A natural question is thus whether we can obfuscate just the original program, and let the new obfuscated code still have access to the external database. The obfuscator is allowed to read the whole external database (during the time of obfuscation), but we require (just like for standard i$\mathcal{O}$) that the size of the obfuscated program is polynomial in the size of the original program (i.e., does not depend on the size of the external database). We refer to Section 7.1 for the formal definition.

A generalized version of such a notion of obfuscation was first studied by Goldwasser and Rothblum [GR07], where instead of databases, a notion of oracle-aided circuits was considered—in essence, an oracle can be thought of as an exponentially long database. In our terminology, it was shown in [GR07] that if (1) the outside database is allowed to be exponentially large, and (2) the obfuscator only gets black-box access to it, then obfuscation is impossible (even if the database is simply a random oracle). But their work leaves open the question of whether obfuscation of database-aided circuits is possible if the database is "feasible" (i.e., of polynomial size) and the obfuscator may read the whole database. Indeed, if $\mathsf{NP} \subseteq \mathsf{BPP}$, then it is easy to optimally obfuscate such database-aided circuits (if the obfuscator gets access to the database).

As our final result, we show how to extend their result to fully rule out obfuscation of database-aided circuits (assuming $\mathsf{NP} \not\subseteq \mathsf{BPP}$) .

Theorem 4. *The following statements are equivalent.*

1. $\mathsf{NP} \not\subseteq \mathsf{BPP}$ (*resp.* $\mathsf{NP} \not\subseteq \mathsf{ZPP}$).
2. *There exists no imperfect (resp. perfect) indistinguishability obfuscator for database-aided circuits with output size $\sigma(\lambda, s)$, for any $\sigma \in \mathsf{poly}$.*

In contrast to [GR07] who showed an unconditional lower bound, to derive our lower bound below we need to assume the worst-case hardness of NP. This is necessary since, as mentioned, assuming NP is easy, there is an optimal obfuscation for database-aided circuits. Unlike the result of [GR07] (which worked given a random oracle), part of our database has a certain structure, and we do not know if the result can be extended to a fully random oracle.

We prove a result similar to Theorem 4 for randomized encoding for Turing machines with database access (see Section 7.2).

Note that removing the access to a database from the result in Theorem 4 would show that i$\mathcal{O}$ in the standard sense does not exist (unless NP is easy). This motivates the investigation of the limits of our approach and techniques. We refer to Section 1.4 for further discussion on this and additional research directions.

1.2 Techniques

In this section, we discuss our techniques and their relation to previous work. In order to explain our techniques in more detail, we focus on the proof of Theorem 1.

The Key Conceptual Insight. Our proofs rely on a recently established connection between indistinguishability obfuscation and meta-complexity introduced by Mazor and Pass [MP24]. Meta-complexity refers to the complexity of computational problems and tasks that are themselves about computations and their complexity. A central example is the Minimum Circuit Size Problem (MCSP), i.e., given the truth table of a Boolean function F, compute the minimum circuit size of F. [MP24] showed, among other results, that the existence of indistinguishably obfuscation and of sub-exponentially secure one-way functions imply that a gap version of MCSP is not NP-hard under Levin reductions. They argue that this result provides strong evidence that the gap version of MCSP is not NP-complete under such reductions.

An important insight of this work is to turn the connection introduced by [MP24] on its head by viewing it as a concrete approach to showing impossibility results for indistinguishably obfuscation. In other words, under the widely believed assumption that sub-exponentially secure one-way functions exist, their result establishes that either indistinguishably obfuscation does not exist or a gap version of MCSP is not NP-hard under Levin reductions. While we do not make progress towards showing the NP-hardness of MCSP and its variants, we show that the connection from [MP24] can be extended to some meta-computational problems for which NP-hardness results are known. This change of perspective, together with other ideas, allow us to derive negative results for indistinguishability obfuscation in a natural setting and under a minimal assumption. (Note that, while our approach relies on techniques from meta-complexity, the statement of Theorem 1 does not refer to meta-complexity.)

Overview of the Proof of Theorem 1. First, we briefly review the easier direction of the equivalence, which follows from a standard argument. From the assumption that NP $\subseteq$ BPP, the polynomial-time hierarchy PH collapses to BPP. Since the problem of finding the lexicographic-first minimum-size multi-output circuit that is equivalent to a given multi-output circuit can be encoded as a language in PH, we obtain a probabilistic polynomial-time algorithm for this problem that fails with exponentially small probability. This in turn yields an (imperfect) i$\mathcal{O}$ scheme with optimal output size $\sigma(\lambda, C) = \mathsf{size}(C)$. This scheme is statistically secure, and thus secure against non-uniform adversaries.

For the other direction, we argue that the assumption that NP $\not\subseteq$ BPP and the existence of a perfect i$\mathcal{O}$ scheme for multi-output circuits with output size $\sigma(\lambda, s) = s + s^{1-o(1)} + \mathsf{poly}(\lambda)$ lead to a contradiction.

To achieve this, we rely on a combination of results, which we describe next.

1. Connection Between $i\mathcal{O}$ and Hardness of Multi-MCSP. As alluded to above, we reinterpret and adapt the connection between $i\mathcal{O}$ and meta-complexity from [MP24]. In order to explain the result, we need a few standard definitions from cryptography and complexity theory.

Recall that a universal one-way hash function (UOWHF) is a function H that maps n bits to m bits, where for us $m = n - \omega(1)$, and such that given a random n-bit input x, it is difficult for an efficient probabilistic algorithm to find $x' \neq x$ such that $H(x) = H(x')$.

The Multi-MCSP problem [ILO20] is defined as follows. We are given as inputs parameters ℓ and m, the truth-table of a function $F\colon \{0,1\}^\ell \to \{0,1\}^m$, and a parameter s. The goal is to decide if there is a multi-output circuit of size at most s that computes F. We will also consider a "gap" or "approximate" version of the problem, where we only need to approximate the minimum size s.

Let $\mathcal{R}_1$ and $\mathcal{R}_2$ be relations contained in $\{0,1\}^* \times \{0,1\}^*$, where we view the first coordinate as an instance and the second coordinate as a potential solution. Recall that a triplet (f, g, h) of efficiently computable functions is a Levin reduction from $\mathcal{R}_1$ to $\mathcal{R}_2$ if $(x, w) \in \mathcal{R}_1$ implies $(f(x), g(x, w)) \in \mathcal{R}_2$, and if $(f(x), w) \in \mathcal{R}_2$ implies $(x, h(x, w)) \in \mathcal{R}_1$. Informally, f maps an instance of $\mathcal{R}_1$ to an instance of $\mathcal{R}_2$, g maps solutions of $\mathcal{R}_1$ to solutions of $\mathcal{R}_2$, and h maps solutions of $\mathcal{R}_2$ to solutions of $\mathcal{R}_1$.

Formally, we show that if a gap version of Multi-MCSP is NP-hard under randomized Levin reductions and a secure $i\mathcal{O}$ scheme for multi-output circuits $C\colon \{0,1\}^\ell \to \{0,1\}^m$ with near-optimal circuit size overhead exists, then UOWHFs do not exist. As in [MP24], a larger gap between the positive instances and negative instances of Multi-MCSP in the assumed Levin reduction allows us to relax the circuit size overhead in the $i\mathcal{O}$ assumption, while maintaining the same conclusion that UOWHFs do not exist.

The proof of this result follows the strategy of [MP24], observing that their approach also holds for the Multi-MCSP problem under the assumption that we have $i\mathcal{O}$ for multi-output circuits. In our setting, we only assume the existence of an imperfect $i\mathcal{O}$ scheme, i.e., there is a negligible probability that it outputs a circuit that is not equivalent to the input circuit. This weaker assumption can be accommodated with a careful argument.

While the technique of [MP24] is reasonably general, we note that it does not easily extend to meta-computational problems over partial functions, such as the one considered in [Hir22]. The fact that we consider the minimum circuit size problem for total functions is crucial for the correctness of the argument.

2. NP-Hardness of Multi-MCSP Under Randomized Approximate Levin Reductions. Loff, Ilango, and Oliveira [ILO20] established that Multi-MCSP is NP-hard under randomized reductions. In order to exploit the connection described in Item 1 above, we extend their result in two directions. Firstly, using a modified construction and a tighter analysis, we show that a gap variant of Multi-MCSP remains NP-hard. In other words, approximating the minimum circuit size s of a given multi-output function up to an additive term of $s^{1-o(1)}$ is hard. Secondly, we show that the NP-hardness result holds under the stronger notion of (randomized) Levin reductions.

3. NP-Hardness of Approximating Set-Cover Under Levin Reductions. The argument from [ILO20] relies on a reduction from certain structured instances of the Set-Cover problem. Recall that in this problem where we are given n, a collection $\mathcal{S}$ of subsets of $[n]$, and a parameter ℓ, and must decide if there are ℓ sets $S_1, \ldots, S_\ell$ in $\mathcal{S}$ whose union is $[n]$. In order to simultaneously achieve a gap and a Levin reduction in Item 2, we must verify that a similar hardness result holds for an appropriate subset of instances of the Set-Cover problem.

Due to the parameters in our approximate randomized Levin reduction from Set-Cover to Multi-MCSP, it is sufficient to show that Set Cover is NP-hard to approximate under Levin reductions up to a constant factor of the form $(1+\varepsilon)$ for some $\varepsilon > 0$. Note that an even stronger hardness of approximation result for Set-Cover is known [Fei98], but to our knowledge there is no explicit proof in the literature that the hardness holds under Levin reductions.

For completeness, we verify that this is indeed the case with an observation that might be of independent interest. Pich [Pic15] established that the PCP Theorem can be formalized in the bounded arithmetic theory PV_1. We note that any NP-hardness result established in PV_1 under the standard notion of Karp reductions immediately implies the NP-hardness of the same problem under Levin reductions. This follows in a generic way using the constructiveness of PV_1. We then simply check that existing reductions from the k-SAT instances derived from the PCP Theorem to the Set-Cover problem maintain the approximation gap, parameters, and hardness under Levin reductions needed in Item 2. To optimize the parameters in our final result about the circuit size overhead in obfuscated circuits, we employ Set-Cover instances where each set is of size $O(1)$, which requires a slightly more involved intermediate reduction using expander graphs from [PY91].

Putting together the results from Items 1,2, and 3, we get that if an i$\mathcal{O}$ scheme for general multi-output circuits $C\colon \{0,1\}^\ell \to \{0,1\}^m$ with near-optimal circuit size overhead exists, then UOWHFs do not exist. In other words, under the cryptographic hardness assumption that UOWHFs exist, we establish the hardness of i$\mathcal{O}$ for general multi-output circuits $C\colon \{0,1\}^\ell \to \{0,1\}^m$.

A caveat is that the parameters ℓ and m obtained from this argument are arbitrary. This is because in the NP-hardness result from [ILO20] and in its strengthening from Item 2, ℓ can be exponentially small in m. Consequently, the argument does not rule out the possibility of having i$\mathcal{O}$ for circuits $C\colon \{0,1\}^\ell \to \{0,1\}^m$ where m is polynomial in ℓ, which is a natural setting in applications.

Next, we discuss how to rule out i$\mathcal{O}$ for input instances $C\colon \{0,1\}^n \to \{0,1\}^n$ where C is of polynomial size, and how to reduce the required assumption from cryptographic hardness to worst-case hardness.

4. Indistinguishability Obfuscation Versus Multi-MCSP: A Simple Reduction. In order to relax the i$\mathcal{O}$ assumption from Item 1 to circuits $C\colon \{0,1\}^n \to \{0,1\}^n$, it is sufficient to show in Item 2 above that Multi-MCSP retains its hardness on input functions $F\colon \{0,1\}^n \to \{0,1\}^n$. This remains a challenging open problem, and can be seen as an important step towards establishing the NP-hardness of MCSP for single-output Boolean functions. The key difficulty is that the instances of Multi-MCSP produced in the reduction from Set-Cover are of the form $F\colon \{0,1\}^\ell \to \{0,1\}^m$ with m of order roughly 2^ℓ, where the reduction runs in time polynomial in m. A natural idea is to "pad" the number of input bits of F to m. However, this would require the reduction to print a truth-table where each coordinate of the resulting function $F\colon \{0,1\}^m \to \{0,1\}^m$ is described by a string of size 2^m, which is prohibitively large.

Interestingly, we can remove this drawback in the setting of i$\mathcal{O}$. Crucially, in contrast with Multi-MCSP, the input of the i$\mathcal{O}$ procedure is a *circuit* instead of a *truth-table*. This allows us to establish thorough a simple padding argument that the hardness of i$\mathcal{O}$ for arbitrary multi-output circuits $C\colon \{0,1\}^\ell \to \{0,1\}^m$ implies its hardness for circuits of the form $C\colon \{0,1\}^n \to \{0,1\}^n$.

5. Hardness Under a Minimal Assumption. Finally, we employ standard results to establish a link between i$\mathcal{O}$ and worst-case hardness. By a result of Rompel [Rom90], the existence of UOWHFs can be based on the existence of one-way functions (OWF). On the other hand, the existence of an imperfect i$\mathcal{O}$ scheme and the assumption that $\mathsf{NP} \not\subseteq \mathsf{BPP}$ yield OWFs [KMN+22].

Items 4 and 5, together with the discussion above, imply that the existence of (imperfect) $\mathsf{i}\mathcal{O}$ for multi-output circuits $C\colon \{0,1\}^n \to \{0,1\}^n$ with bounded circuit size overhead and the assumption that $\mathsf{NP} \not\subseteq \mathsf{BPP}$ lead to a contradiction. This completes the overview of the techniques employed in the proof of Theorem 1.

In Section 6, we present an alternative proof of our main results that might be of independent interest. In particular, this proof does not rely on the PCP Theorem. It bypasses the general framework described above by relying instead on pseudorandom encryption schemes and on a direct adaptation of a technique from [ILO20]. However, while the alternative proof is shorter and more direct, it achieves somewhat weaker parameters, only ruling out $\mathsf{i}\mathcal{O}$ schemes whose output circuits have size of order $s + o(s^{1/2})$ instead of $s + s^{1-o(1)}$. In contrast, the general framework described above via meta-complexity is more modular and can be applied in more general settings. For instance, it can rule out size-efficient $\mathsf{i}\mathcal{O}$ for a weak circuit model under Levin hardness of a corresponding meta-computational problem. Conversely, an advantage of the alternative proof is that it is essentially self-contained and easier to adapt to various stronger computational models when applicable (see Section 7).

The Alternative Approach Through Pseudorandom Encryptions. We rely on encryption schemes with pseudorandom ciphers, which we describe below. Such schemes are equivalent to one-way functions via [HILL99, GGM86] (see Section 6). Consequently, the argument sketched next rules out size-optimal $\mathsf{i}\mathcal{O}$ under OWFs. It is then possible to derive our main results (with somewhat weaker parameters) by combining the argument with the ideas described in Items 4 and 5 above.

Informally, a triple $(\mathsf{Enc}, \mathsf{Dec}, \mathsf{LD})$ of efficient algorithms is a *local rate*-1 *pseudorandom encryption scheme* if the following properties hold:

1. (Rate) For every choice of the key k and message m, the length of the ciphertext $\mathsf{Enc}(m,k)$ equals the length of m.
2. (Correctness) The scheme is correct, i.e., we always have $\mathsf{Dec}(\mathsf{Enc}(m,k),k) = m$.
3. (Pseudorandomness) No PPT algorithm can distinguish $(m, \mathsf{Enc}(m,k))$ from (m,y), where y is a random string of length $|m|$.
4. (Local Decoding) LD runs in time $\mathsf{poly}(|k|, \log|m|)$ and recovers the i-th bit of the message, i.e., $\mathsf{LD}(k, |m|, i, \mathsf{Enc}(m,k)_i) = m_i$.

Let λ be the security parameter for an obfuscator $\mathsf{i}\mathcal{O}$ with output size $\sigma(\lambda, s) = s + o(s^{1/2}) + \mathsf{poly}(\lambda)$. We use the triple $(\mathsf{Enc}, \mathsf{Dec}, \mathsf{LD})$ to construct an ensemble of distributions $\{\mathcal{P}_\lambda\}_{\lambda\in\mathbb{N}}$ over pairs (C_1, C_2) of multi-output circuits and an advice string $a \in \{0,1\}^*$ of bounded length, together with an efficient deterministic algorithm A, such that $\mathbf{Pr}_{(C_1,C_2,a)\leftarrow\mathcal{P}_\lambda}[C_1 \equiv C_2] = 1$ but

$$\left| \mathop{\mathbf{Pr}}_{(C_1,C_2,a)\leftarrow\mathcal{P}_\lambda,\, \mathsf{i}\mathcal{O}}[A(1^\lambda, a, \mathsf{i}\mathcal{O}(1^\lambda, C_1)) = 1] - \mathop{\mathbf{Pr}}_{(C_1,C_2,a)\leftarrow\mathcal{P}_\lambda,\, \mathsf{i}\mathcal{O}}[A(1^\lambda, a, \mathsf{i}\mathcal{O}(1^\lambda, C_2)) = 1] \right| = \Omega(1). \quad (1)$$

By a standard averaging argument that non-uniformly fixes the required advice bits, this contradicts the security of the $\mathsf{i}\mathcal{O}$ scheme.

Informal Description of $\mathcal{P}_\lambda$. We next describe how to sample the circuits $C_1, C_2 \colon \{0,1\}^{\log n} \to \{0,1\}^n$. and the advice a. We start with an informal description of the procedure that samples the truth table of the function computed by C_1 and C_2.

Let $j_1, j_2 \leftarrow [n]$ be two distinct random indices, and let $r \colon \{0,1\}^{\log n} \to \{0,1\}$ be a random function (represented by a random string of length n). Let k_1 and k_2 be two random keys for the pseudorandom encryption scheme $(\mathsf{Enc}, \mathsf{Dec}, \mathsf{LD})$. We next sample n additional functions $T_1, \ldots, T_n \colon \{0,1\}^{\log n} \to \{0,1\}$ as follows: For every $i \notin \{j_1, j_2\}$, we let T_i be a random function. We let T_{j_1} and T_{j_2} be the functions whose truth tables are $\mathsf{Enc}(r, k_1)$ and $\mathsf{Enc}(r, k_2)$, respectively. Finally, we let $T \colon \{0,1\}^{\log n} \to \{0,1\}^n$ be the multi-output function defined by the concatenation of $T_1, \ldots, T_n$. That is, $T(x) = T_1(x) \bullet \cdots \bullet T_n(x)$.

We next rely on a technique from [ILO20]. Roughly speaking, [ILO20] shows a way to encode a function T in a new function $\widehat{T}$ (with a similar input length but longer output) that has a canonical circuit C_T that computes $\widehat{T}$, with the property that a copy of C_T can be found in any circuit that computes $\widehat{T}$. Moreover, $\widehat{T}$ admits a "chain rule": for any function $f \colon \{0,1\}^{\log n} \to \{0,1\}$, and any circuit C that computes the concatenation $(\widehat{T} \bullet f)(x) = \widehat{T}(x) \bullet f(x)$, it is possible to efficiently decompose C into two circuits: the circuit C_T that computes $\widehat{T}$, and a circuit D of size $s = \mathsf{size}(C) - \mathsf{size}(C_T)$ such that $D(x, \widehat{T}(x)) = f(x)$. For simplicity, in the following we assume that this decomposition work when we take $\widehat{T}$ to be simply T. Given this simplified assumption, the truth table of C_1 and C_2 is given by the function $F(x) = T(x) \bullet r(x)$.

We next define the circuits C_1 and C_2. The circuit C_1 is simply composed by the circuit C_T that computes the function T, and the local decoding circuit of the encryption scheme, $\mathsf{LD}(k_1, n, x, T_{j_1}(x))$, where the key k_1 is hard-coded to it, and the value of T_{j_1} is computed by the right output wire of the circuit C_T. By the construction of $T_{j_1} = \mathsf{Enc}(r, k_1)$ and the correctness of the local decoding, we get that C_1 indeed computes the function F. Moreover, by a careful choice of parameters, the size of C_1 is bounded by $\mathsf{size}(C_T) + o(n)$. The circuit C_2 is defined similarly, but replaces k_1 and j_1 in the above construction of C_1 with k_2 and j_2.

Lastly, the advice string a is simply composed by the index j_1 and the truth table of the function T.

Informal Description of the Distinguisher A. Next we describe the distinguisher. Given the advice string containing the index j_1 and a circuit C that computes $T \circ r$, A decomposes C into the circuit C_T and a circuit D such that $D(x, T(x)) = r(x)$. Importantly, by carefully choosing the parameters, when $C = \mathsf{i}\mathcal{O}(C_b)$ for $b \in \{1,2\}$, the size of D is much smaller than $n = |T(x)|$. The distinguisher A simply checks if D touches the j_1-th bit of $T(x)$, and if so it outputs 1.

The upper bound on the size of D relies on the upper bound on the size of $\mathsf{i}\mathcal{O}(C_b)$. This in turn depends on the size of C_b, which can be bounded due to the use of the local decoder, and on the size overhead of the $\mathsf{i}\mathcal{O}$ scheme.

Intuition Behind Equation (1). We finally give some intuition on why A succeeds. We rely on the pseudorandomness property of the encryption scheme, i.e., that T_{j_1} and T_{j_2} are indistinguishable from random functions.

- First, it follows that A cannot distinguish between the circuit C_2 and a circuit $\widehat{C}_2$ which is constructed similarly, but in which T_{j_1} is a random function (instead of an encryption of r). However, in the circuit $\widehat{C}_2$, T_{j_1} is distributed exactly as every other output wire T_i, and since D touches only $o(n)$ of the input wires of C_T, it touches T_{j_1} with low probability. This can be used to show that A outputs 1 with low probability on $\mathsf{i}\mathcal{O}(C_2)$.

- On the other hand, we argue that on $\mathsf{i}\mathcal{O}(C_1)$, A must output 1 with high probability. Let $\widehat{C}_1$ be a circuit which is constructed similarly to C_1, but in which T_{j_2} is a random function (instead of an encryption of r). Since $\widehat{C}_1$ is indistinguishable from C_1, A outputs 1 on $\mathsf{i}\mathcal{O}(C_1)$ and on $\mathsf{i}\mathcal{O}(\widehat{C}_1)$ with essentially the same probability. Assume towards a contradiction that A outputs 0 with noticeable probability on $\mathsf{i}\mathcal{O}(\widehat{C}_1)$. Then there is a small circuit D, such that $D(x, T(x)) = r(x)$, and D does not touch the j_1-th bit of $T(x)$. Since all the other bits in the output of T are independent of r, we are able to show in this case that r can be compressed. But r is random, which implies that A outputs 1 with high probability on $\mathsf{i}\mathcal{O}(\widehat{C}_1)$.

The argument is somewhat subtle because the obfuscations of C_1 and $\widehat{C}_1$ are not indistinguishable by non-uniform adversaries (indeed, these circuits compute different functions). However, we only need to fool the fixed distinguisher A described above, which is sufficient for the purpose of establishing Equation (1).

The proof of Theorem 2 is similar to the proof of Theorem 1. On the other hand, the proof of Theorem 4 combines ideas from the alternative approach describe above and from NP-hardness results for conditional time-bounded Kolmogorov complexity ([Ila20, ACM+21, LP22]).

1.3 Related Work

Although there is extensive literature on positive results and potential constructions of indistinguishability obfuscation schemes, the area of impossibility results remains less explored.

Hada [Had00] provided impossibility results regarding the obfuscation of pseudorandom functions under a strong definition of obfuscation.

Barak et al. [BGI+12] established the impossibility of virtual black-box obfuscation for some contrived functions. Goldwasser and Kalai [GK05] considered the notion of virtual black-box obfuscation with respect to an auxiliary input, and proved that there exist natural classes of functions that cannot be obfuscated in this setting (see also [BCC+14, CKP15, PS15, MMN15, LPST16]).

As mentioned above, Goldwasser and Rothblum [GR07] showed that indistinguishably obfuscation is impossible in the random oracle model, i.e., when the circuit, the adversary, and the obfuscator all share access to a random oracle.

Boyle et al. [BIM+23] investigated negative results in the context of $\mathsf{i}\mathcal{O}$ for weak circuit classes $\mathcal{C}$. In more detail, a proper obfuscation scheme obfuscates circuits from a circuit class $\mathcal{C}$ by circuits from the same class. For instance, assuming the existence of one-way functions, they showed that there is no proper $\mathsf{i}\mathcal{O}$ scheme for k-CNF formulas for any constant $k \geq 3$. They make use of existing PAC learning algorithms to obtain negative results for obfuscation, which are not available in the context of more expressive circuit classes, such as DNFs.[2]

1.4 Concluding Remarks

As alluded to above (see also Section 4.1.1), any proof of an NP-hardness result in Cook's theory PV implies that the same NP-hardness result holds under Levin reductions. Consequently, the provability of the hardness of meta-computational problems with a gap between yes and no instances in this theory is directly

[2] While we have not explored this direction in our work, we believe that our techniques can be adapted to show negative results on the size-efficiency of proper $\mathsf{i}\mathcal{O}$ schemes for DNFs and other weak classes (via existing results on the NP-hardness of the minimum circuit size problem for such classes of functions).

connected to the (non)existence of $i\mathcal{O}$ with bounded output size for the corresponding class of computational devices. This exhibits another link between bounded arithmetic and $i\mathcal{O}$ (see also [JJ22, ILW23]).

Our results show that the Mazor-Pass connection [MP24] can be leveraged to show impossibility results for $i\mathcal{O}$ for a certain range of parameters. Intriguingly, there is currently no barrier preventing this framework from ruling out $i\mathcal{O}$ (in the standard sense) under a worst-case hardness assumption.[3] Note that this would also follow if one could prove a bootstrapping result showing that an $i\mathcal{O}$ scheme with output size $s^{O(1)}$ yields an $i\mathcal{O}$ scheme with output size $s + o(s/\log s)$. Moreover, if the results of [AJS17] generalize to the multi-output setting and for a notion of size corresponding to number of gates, it would be sufficient to bootstrap from output size $(2 + o(1)) \cdot s$ to output size $s + o(s/\log s)$.

As our main open problems, we leave the extension of our (database-free) impossibility results to the setting of single-output circuits, the investigation of $i\mathcal{O}$ for multi-output circuits with output size $(1+\Omega(1)) \cdot s$, and understanding the limits of our approach. It would also be interesting to understand if our techniques can be used to show negative results on the efficiency of additional cryptographic primitives. We hope that our work will motivate further research in these directions.

Acknowledgements. This work received support from the Royal Society University Research Fellowship URF\R1\191059; the UKRI Frontier Research Guarantee Grant EP/Y007999/1; and the Centre for Discrete Mathematics and its Applications (DIMAP) at the University of Warwick. The research of the second author is partly supported by NSF CNS-2149305, AFOSR Award FA9550-23-1-0312 and AFOSR Award FA9550-23-1-0387. The last author is supported in part by NSF Award CNS 2149305, AFOSR Award FA9550-23-1-0387, AFOSR Award FA9550-23-1-0312 and AFOSR Award FA9550-24-1-0267. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Government, or the AFOSR.

2 Preliminaries

2.1 Basic Definitions and Notation

For a string $x \in \{0,1\}^*$, we use $|x|$ to denote the length of x. For strings $x, y \in \{0,1\}^*$, we let $x \bullet y$ denote the concatenated string.

We say that a function $\alpha\colon \mathbb{N} \to \mathbb{N}$ is *negligible* if for every constant $c \geq 1$, there is $n_0 \geq 1$ such that, for every $n \geq n_0$, $\alpha(n) \leq n^{-c}$.

We let PPT stand for probabilistic polynomial time.

Boolean Circuits. For concreteness, we consider Boolean circuits consisting of AND, OR gates of fan-in two and NOT gates of fan-in one, with access to input variables $x_1, \ldots, x_n$ and constants 0 and 1. (Our results are robust to the set of fan-in two gates allowed in the circuit.) We measure the size of a circuit C (denoted $\mathsf{size}(C)$) by the number of AND, OR, and NOT gates in the circuit. When considering multi-output circuits, we assign to each output coordinate some internal gate or input variable present in the circuit.

[3] In order to get such result, it would be enough to establish hardness under Levin reductions under the assumption that $i\mathcal{O}$ exists. We refer to [HIR23] for examples of the usefulness of various cryptographic assumptions when proving NP-hardness of meta-computational problems.

$$
\begin{array}{lll}
 & & \boldsymbol{x_1} \\
 & & \boldsymbol{x_2} \\
\boldsymbol{o_1} & G_1 \leftarrow & \boldsymbol{x_1} \text{ AND } \boldsymbol{x_2} \\
 & G_2 \leftarrow & \text{NOT } \boldsymbol{x_1} \\
 & G_3 \leftarrow & \text{NOT } \boldsymbol{x_2} \\
 & G_4 \leftarrow & G_2 \text{ AND } \boldsymbol{x_2} \\
 & G_5 \leftarrow & \boldsymbol{x_1} \text{ AND } G_3 \\
\boldsymbol{o_2} & G_6 \leftarrow & G_4 \text{ OR } G_5
\end{array}
$$

In the example above, we have a circuit $C\colon \{0,1\}^2 \to \{0,1\}^2$ represented as a straight-line program that outputs the sum (o_1, o_2) of the input bits (x_1, x_2). In this example, the circuit C has size 6.

For two circuits $C, C'\colon \{0,1\}^\ell \to \{0,1\}^m$, we let $C \equiv C'$ denote that $C(x) = C'(x)$ for every $x \in \{0,1\}^\ell$. In this case, we say that the circuits are functionally equivalent.

The following fact will be useful for us.

Proposition 5 (See, e.g., [Juk12])**.** *There is a polynomial-time algorithm that, given a binary string of length 2^n representing the truth table of a function $f\colon \{0,1\}^n \to \{0,1\}$, outputs a circuit of size at most $5 \cdot 2^n/n$ that computes f.*

Infinitely-Often One-Way Functions (i.o.OWF). We say that a function family $f = \{f_n\}_{n\geq 1}$, where each $f_n\colon \{0,1\}^n \to \{0,1\}^{m(n)}$, is an *infinitely-often one-way function* (i.o.OWF) if for every PPT algorithm A and for every constant $c \geq 1$, there are infinitely many input lengths n such that

$$\Pr_{x\leftarrow\{0,1\}^n,\, A}\left[A(1^n, f(x)) \in f^{-1}(f(x))\right] \leq n^{-c}.$$

Infinitely-Often Universal One-Way Hash Function (i.o.UOWHF). We say that a function family $h = \{h_n\}_{n\geq 1}$, where each $h_n\colon \{0,1\}^n \to \{0,1\}^{n-\zeta(n)}$, is an *infinitely-often universal one-way hash function* (i.o.UOWHF) if $\zeta(n) \geq 1$ and for every PPT algorithm A and for every constant $c \geq 1$, there are infinitely many input lengths n such that

$$\Pr_{x\leftarrow\{0,1\}^n,\, A}\left[x' \leftarrow A(x);\ f_n(x) = f_n(x') \text{ and } x \neq x'\right] \leq n^{-c}.$$

Theorem 6 (OWF to UOWHF, [Rom90])**.** *Assume the existence of i.o.OWFs. Then i.o.UOWHFs exist.*

2.2 Indistinguishability Obfuscation

Definition 7 (Indistinguishability Obfuscation for Multi-Output Circuits)**.** *A probabilistic polynomial-time algorithm $i\mathcal{O}$ is an* indistinguishability obfuscator for multi-output circuits with output size $\sigma\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ *if the following hold.*

- ***Perfect/Imperfect Functionality:*** *There exists a negligible function α such that for all $\lambda, n \in \mathbb{N}$ and any circuit $C\colon \{0,1\}^n \to \{0,1\}^n$,*

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

We say that $\mathsf{i}\mathcal{O}$ is perfect if $\alpha(\cdot) = 0$; otherwise it is imperfect.

- ***Indistinguishability:*** *For any polynomial-size circuit family $\{A_\lambda\}_\lambda$ and polynomial p, there exists a negligible function μ such that for all $\lambda, n \in \mathbb{N}$ and any pair of circuits $C, C'\colon \{0,1\}^n \to \{0,1\}^n$ satisfying $C \equiv C'$ and $n \leq \mathsf{size}(C) = \mathsf{size}(C') \leq p(\lambda)$, it holds that*

$$\left|\Pr_{\mathsf{i}\mathcal{O}}[A_\lambda(\mathsf{i}\mathcal{O}(1^\lambda, C)) = 1] - \Pr_{\mathsf{i}\mathcal{O}}[A_\lambda(\mathsf{i}\mathcal{O}(1^\lambda, C')) = 1]\right| \leq \mu(\lambda).$$

- ***σ-Output-Size:*** *For all $\lambda, n \in \mathbb{N}$ and every circuit $C\colon \{0,1\}^n \to \{0,1\}^n$,*

$$\Pr_{\mathsf{i}\mathcal{O}}\Big[\mathsf{size}(\mathsf{i}\mathcal{O}(1^\lambda, C)) \leq \sigma(\lambda, \mathsf{size}(C))\Big] = 1.$$

We often implicitly assume that $\mathsf{size}(C) = \mathsf{size}(C') = p(\lambda)$, since dummy gates can be added to the circuit so it has size exactly $p(\lambda)$.

2.3 Randomized Levin Reductions for Promise Problems

For a relation $\mathcal{R} \subseteq \{0,1\}^* \times \{0,1\}^*$, let $\mathcal{L}(\mathcal{R}) = \{x \in \{0,1\}^*\colon \exists w \in \{0,1\}^* \; s.t. \; (x,w) \in \mathcal{R}\}$. We say that a relation $\mathcal{R}$ is the witness relation of a language $\mathcal{L} \subseteq \{0,1\}^*$ if $\mathcal{L}(\mathcal{R}) = \mathcal{L}$.

Definition 8 (Levin Reduction)**.** *Let $\mathcal{R}_1$ and $\mathcal{R}_2$ be relations. A triplet (f, g, h) of efficiently computable functions is a* Levin reduction *from $\mathcal{R}_1$ to $\mathcal{R}_2$ if*

- *For every $(x,w) \in \mathcal{R}_1$, $(f(x), g(x,w)) \in \mathcal{R}_2$.*
- *If $(f(x), w) \in \mathcal{R}_2$ then $(x, h(x,w)) \in \mathcal{R}_1$.*

Remark 9. *Notice that if (f, g, h) is a Levin reduction from $\mathcal{R}_1$ to $\mathcal{R}_2$, then f is a Karp reduction from $\mathcal{L}(\mathcal{R}_1)$ to $\mathcal{L}(\mathcal{R}_2)$. Indeed, the first item above implies that if $x \in \mathcal{L}(\mathcal{R}_1)$ then $f(x) \in \mathcal{L}(\mathcal{R}_2)$, and the second item implies the other direction.*

A Levin reduction (f, g, h) is *honest* if there exists a constant $\delta > 0$ such that for every large enough $n \in \mathbb{N}$ and every $x \in \{0,1\}^n$, $|f(x)| \geq n^\delta$.

When for two languages $\mathcal{L}_1$ and $\mathcal{L}_2$ we fix canonical relations $\mathcal{R}_1$ and $\mathcal{R}_2$, we say that there is a Levin reduction from $\mathcal{L}_1$ to $\mathcal{L}_2$ if there is a Levin reduction from $\mathcal{R}_1$ to $\mathcal{R}_2$. We say that $\mathcal{L} \in \mathsf{NP}$ is NP-hard under Levin reductions if there exists a Levin reduction from SAT to $\mathcal{L}$, where the canonical relation for SAT is

$$\mathcal{R}_{\mathsf{SAT}} = \{(\phi, v)\colon \phi \text{ is a Boolean formula and } \phi[v] = 1\}.$$

We also define Levin reductions in the more general context of promise problems. This is useful when considering the NP-hardness of problems for which there is a gap between positive and negative instances.

In the following, we consider a promise problem $(\mathcal{Y}, \mathcal{N})$ that is associated with two relations $(\mathcal{R}_\mathcal{Y}, \mathcal{R}_{\overline{\mathcal{N}}})$ such that $\mathcal{R}_\mathcal{Y} \subseteq \mathcal{R}_{\overline{\mathcal{N}}}$, where $\mathcal{R}_\mathcal{Y}$ is the witness relation for $\mathcal{Y}$, and $\mathcal{R}_{\overline{\mathcal{N}}}$ is the witness relation for $\overline{\mathcal{N}}$. That is, $(\mathcal{Y}, \mathcal{N}) = (\mathcal{L}(\mathcal{R}_\mathcal{Y}), \overline{\mathcal{L}(\mathcal{R}_{\overline{\mathcal{N}}})})$.

Definition 10 (Levin Reduction for Promise Problems)**.** *Let* $(\mathcal{R}^1_{\mathcal{Y}}, \mathcal{R}^1_{\mathcal{N}})$ *and* $(\mathcal{R}^2_{\mathcal{Y}}, \mathcal{R}^2_{\mathcal{N}})$ *be pairs of relations such that* $\mathcal{R}^1_{\mathcal{Y}} \subseteq \mathcal{R}^1_{\mathcal{N}}$ *and* $\mathcal{R}^2_{\mathcal{Y}} \subseteq \mathcal{R}^2_{\mathcal{N}}$*. A triplet* (f, g, h) *of efficiently computable functions is a* Levin reduction *from* $(\mathcal{R}^1_{\mathcal{Y}}, \mathcal{R}^1_{\mathcal{N}})$ *to* $(\mathcal{R}^2_{\mathcal{Y}}, \mathcal{R}^2_{\mathcal{N}})$ *if*

- *For every* $(x, w) \in \mathcal{R}^1_{\mathcal{Y}}$*,* $(f(x), g(x, w)) \in \mathcal{R}^2_{\mathcal{Y}}$*.*
- *If* $(f(x), w) \in \mathcal{R}^2_{\mathcal{N}}$ *then* $(x, h(x, w)) \in \mathcal{R}^1_{\mathcal{N}}$*.*

We also consider randomized Levin reductions. In this case, $f(x; r)$ can be a randomized function (that uses randomness r), and both g and h get access to r. We require that the above conditions hold with high probability over the choice of r. The following definition suffices for our purposes.

Definition 11 (Randomized Levin Reduction for Promise Problems)**.** *Let* $(\mathcal{R}^1_{\mathcal{Y}}, \mathcal{R}^1_{\mathcal{N}})$ *and* $(\mathcal{R}^2_{\mathcal{Y}}, \mathcal{R}^2_{\mathcal{N}})$ *be pairs of relations such that* $\mathcal{R}^1_{\mathcal{Y}} \subseteq \mathcal{R}^1_{\mathcal{N}}$ *and* $\mathcal{R}^2_{\mathcal{Y}} \subseteq \mathcal{R}^2_{\mathcal{N}}$*. A triplet of efficiently computable functions* (f, g, h) *is an* ε-error randomized Levin reduction *from* $(\mathcal{R}^1_{\mathcal{Y}}, \mathcal{R}^1_{\mathcal{N}})$ *to* $(\mathcal{R}^2_{\mathcal{Y}}, \mathcal{R}^2_{\mathcal{N}})$ *if on every input instance* x*, with probability at least* $1 - \varepsilon$ *over the choice of* r*, the following holds*:

1. *for every* w *such that* $(x, w) \in \mathcal{R}^1_{\mathcal{Y}}$*,* $(f(x; r), g(x, w; r)) \in \mathcal{R}^2_{\mathcal{Y}}$*, and*
2. *for every* w' *such that* $(f(x; r), w') \in \mathcal{R}^2_{\mathcal{N}}$*,* $(x, h(x, w'; r)) \in \mathcal{R}^1_{\mathcal{N}}$*.*

In order to improve readability, in many occasions we explicit describe the properties of the triple (f, g, h) describing a Levin reduction between promise problems.

3 Indistinguishability Obfuscation

3.1 Indistinguishability Obfuscation Under the Easiness of NP

Proposition 12. *If* $\mathsf{NP} \subseteq \mathsf{ZPP}$ (*resp.* $\mathsf{NP} \subseteq \mathsf{BPP}$)*, then there exist perfect* (*resp. imperfect*) *indistinguishability obfuscators for multi-output circuits with output size* $\sigma(\lambda, \mathsf{size}(C)) = \mathsf{size}(C)$*.*

Proof Sketch. This is a well know result that is essentially due to [BGI+12]. If $\mathsf{NP} \subseteq \mathsf{ZPP}$, then $\mathsf{PH} \subseteq \mathsf{ZPP}$ (see, e.g., [For17]). By standard search-to-decision techniques, there exists a *zero-error* probabilistic polynomial-time algorithm such that, given a (multi-output) circuit C, outputs the lexicographic-first minimum-size circuit that computes the same function as C with probability exponentially close to 1 and $\perp$ otherwise. By replacing the output $\perp$ with the input circuit C, we get a (statistically secure) indistinguishability obfuscator with *perfect* functionality and no overhead.

The proof for the case of $\mathsf{NP} \subseteq \mathsf{BPP}$ is similar except that now we get a probabilistic polynomial-time algorithm that can output, with exponentially small probability, some circuit other than the lexicographic-first minimum-size circuit computing the same function as C. As a result, we only get an indistinguishability obfuscator with *imperfect* functionality. □

3.2 Reduction for Multi-Output Circuits

It is not hard to see that if we can obfuscate circuits mapping n bits to n bits, then we can also obfuscate circuits mapping ℓ bits to m bits for any $\ell, m \in \mathbb{N}$, by adding dummy input or output wires. We make this formal in this subsection.

Analogously to Definition 7, we define indistinguishability obfuscation for *general* multi-output circuits.

Definition 13 (Indistinguishability Obfuscation for General Multi-Output Circuits)**.** *A probabilistic polynomial-time algorithm* $\mathsf{i}\mathcal{O}$ *is an* indistinguishability obfuscator for general multi-output circuits with output size $\sigma\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$ *if the following hold.*

- ***Perfect/Imperfect Functionality:*** *There exists a negligible function* α *such that for all* $\lambda,\ell,m\in\mathbb{N}$ *and any circuit* $C\colon\{0,1\}^{\ell}\to\{0,1\}^{m}$*,*

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

  We say that $\mathsf{i}\mathcal{O}$ *is perfect if* $\alpha(\cdot)=0$*; otherwise it is imperfect.*

- ***Indistinguishability:*** *For any polynomial-size circuit family* $\{A_\lambda\}_\lambda$ *and polynomial* p*, there exists a negligible function* μ *such that for all* $\lambda,\ell,m\in\mathbb{N}$ *and any pair of circuits* $C,C'\colon\{0,1\}^{\ell}\to\{0,1\}^{m}$ *satisfying* $\max\{\ell,m\}\leq C\equiv C'$ *and* $\mathsf{size}(C)=\mathsf{size}(C')\leq p(\lambda)$*, it holds that*

$$\left|\Pr_{\mathsf{i}\mathcal{O}}[A_\lambda(\mathsf{i}\mathcal{O}(1^{\lambda},C))=1]-\Pr_{\mathsf{i}\mathcal{O}}[A_\lambda(\mathsf{i}\mathcal{O}(1^{\lambda},C'))=1]\right|\leq\mu(\lambda).$$

- σ***-Output-Size:*** *For all* $\lambda,\ell,m\in\mathbb{N}$ *and every circuit* $C\colon\{0,1\}^{\ell}\to\{0,1\}^{m}$*,*

$$\Pr_{\mathsf{i}\mathcal{O}}\Big[\mathsf{size}(\mathsf{i}\mathcal{O}(1^{\lambda},C))\leq\sigma(\lambda,\mathsf{size}(C))\Big]=1.$$

Lemma 14. *For any output size function* σ*, if there exist indistinguishability obfuscators for multi-output circuits with output size* σ *(in the sense of Definition 7), then there also exist indistinguishability obfuscators for general multi-output circuits with output size* σ *(in the sense of Definition 13).*

Proof. Let σ be an output size function and $\mathsf{i}\mathcal{O}$ be an indistinguishability obfuscator for multi-output circuits with output size σ.

Consider the following algorithm $\mathsf{i}\mathcal{O}'$,

> On input $(1^{\lambda},C)$, where C is a circuit mapping ℓ bits to m bits. Assume without loss of generality that $m>\ell$. We first obtain a circuit C' mapping m bits to m bits, obtained from C by adding $m-\ell$ dummy input wires, i.e., they are not connected to any internal gate. We then obtain a circuit $C''\coloneqq\mathsf{i}\mathcal{O}(1^{\lambda},C')$. Finally, we set the last $m-\ell$ bits of C'' to be 0 and output the resulting circuit, which maps ℓ bits to m bits.

We claim that $\mathsf{i}\mathcal{O}'$ is an indistinguishability obfuscator for general multi-output circuits, as defined in Definition 13.

First of all, since the circuit C' is obtained by adding dummy input wires to the input circuit C (which maps ℓ bits to m bits), it has the same functionality as C when restricted to the first ℓ input bits. If $\mathsf{i}\mathcal{O}$ preserves the functionality of C', then $C''=\mathsf{i}\mathcal{O}(1^{\lambda},C')$ also has the same functionality as C (again, when restricted to the first ℓ input bits). This further implies that the final output circuit has the same functionality as C. As a result, $\mathsf{i}\mathcal{O}'$ preserves functionality as $\mathsf{i}\mathcal{O}$ does (either perfectly or imperfectly).

For the output size, since we only add dummy input wires to the input circuit C, its size does not change, i.e., $\mathsf{size}(C)=\mathsf{size}(C')$. Then we have $\mathsf{size}(\mathsf{i}\mathcal{O}'(1^{\lambda},C))\leq\mathsf{size}(\mathsf{i}\mathcal{O}(1^{\lambda},C'))\leq\sigma(\lambda,\mathsf{size}(C'))=\sigma(\lambda,\mathsf{size}(C))$.

Finally, for the security, suppose, towards a contradiction, there exist a polynomial-size circuit family $\{A_\lambda\}_\lambda$, a polynomial p, a constant $c>1$ and an infinite set $I\subseteq\mathbb{N}$ such that for every $\lambda\in I$, there

are $\ell, m \in \mathbb{N}$ and a pair of circuits $C_1, C_2 \colon \{0,1\}^\ell \to \{0,1\}^m$ satisfying $C_1 \equiv C_2$ and $\max\{\ell, m\} \leq \mathsf{size}(C_1) = \mathsf{size}(C_2) \leq p(\lambda)$ for which

$$\left| \Pr_{\mathsf{i}\mathcal{O}}[A_\lambda(\mathsf{i}\mathcal{O}'(1^\lambda, C_1)) = 1] - \Pr_{\mathsf{i}\mathcal{O}}[A_\lambda(\mathsf{i}\mathcal{O}'(1^\lambda, C_2)) = 1] \right| > \lambda^{-c}.$$

Then consider the following polynomial-size circuit family $\{A'_\lambda\}_\lambda$ obtained from $\{A_\lambda\}_\lambda$ as follows.

> For every $\lambda \in I$, A'_λ takes an additional advice (ℓ, m) and given a circuit $C'' \colon \{0,1\}^m \to \{0,1\}^m$, sets the last $m - \ell$ input wires of C'' to 0, and outputs A_λ applying to the resulting circuit.

By the definition of $\mathsf{i}\mathcal{O}'$, the above implies that for every $\lambda \in I$, there exist $m \in \mathbb{N}$ and a pair of circuits $C'_1, C'_2 \colon \{0,1\}^m \to \{0,1\}^m$ satisfying $C'_1 \equiv C'_2$ and $m \leq \mathsf{size}(C'_1) = \mathsf{size}(C'_2) \leq p(\lambda)$ such that

$$\left| \Pr_{\mathsf{i}\mathcal{O}}[A'_\lambda(\mathsf{i}\mathcal{O}(1^\lambda, C'_1)) = 1] - \Pr_{\mathsf{i}\mathcal{O}}[A'_\lambda(\mathsf{i}\mathcal{O}(1^\lambda, C'_2)) = 1] \right| > \lambda^{-c},$$

which contradicts the security of $\mathsf{i}\mathcal{O}$. □

As an immediate consequence of Lemma 14, we get that the existence of indistinguishability obfuscators for multi-output circuits implies that for single-output circuits. It was shown in [KMN+22] that assuming NP is hard, the existence of indistinguishability obfuscators for single-output circuits (with polynomial output size) implies the existence of one-way functions. As a result, we derive the following consequence.

Theorem 15 (Following [KMN+22])**.** *Assume* $\mathsf{NP} \not\subseteq \mathsf{ZPP}$ (*resp.* $\mathsf{NP} \not\subseteq \mathsf{BPP}$)*. If there exist perfect* (*resp. imperfect*) *indistinguishability obfuscators for multi-output circuits with polynomial output size, then infinitely-often one-way functions exist.*

3.3 Connection Between Indistinguishability Obfuscation and Meta-Complexity

In this section, we employ a technique from [MP24] to establish the following result.

Theorem 16. *Let* $\beta > \alpha > 0$*, and let* $\gamma, \delta > 0$*. Consider the following assumptions*:

(*i*) *There is an honest randomized Levin reduction from* Circuit-Sat *to* Gap-Multi-MCSP$[s, s + s/(\alpha \cdot \log s)]$*. More precisely, there are deterministic polynomial-time functions* f*,* g*, and* h *satisfying the following conditions:*

(1) *Given a sufficiently large* n*-variable Boolean circuit* $\varphi \in \{0,1\}^{\mathsf{poly}(n)}$ *and a random string* $r \in \{0,1\}^{\mathsf{poly}(n)}$*,* $f(\varphi, r)$ *outputs* $(1^\ell, 1^m, 1^s, \mathsf{tt}(F))$*, where* $\ell, m, s \in \mathbb{N}$ *and* $F \colon \{0,1\}^\ell \to \{0,1\}^m$ *is a multi-output function. Moreover,* $s = s(\varphi, r) \geq n^\delta$ *and* $s \geq \max\{\ell, m\}$*.*

(2) *With probability at least* $1 - o(1)$ *over the choice of* r*, the following holds*:

(*a*) *for any input* x *such that* $\varphi(x) = 1$*,* $g(\varphi, x, r)$ *outputs a circuit* C *of size at most* s *that computes* F*, and*

(*b*) *for any circuit* D *of size at most* $s/(\alpha \cdot \log s)$ *that computes* F*,* $\varphi(h(\varphi, D, r)) = 1$*.*

(*ii*) *There exist imperfect indistinguishability obfuscators for general multi-output circuits with output size* $\sigma(\lambda, s) = s + s/(\beta \cdot \log s) + \lambda^\gamma$*.*

Then there is no infinitely-often universal one-way hash function $H = \{H_n\}_{n\geq 1}$ with $H_n\colon \{0,1\}^n \to \{0,1\}^{n-\zeta(n)}$ and $\zeta(n) = \omega(1)$.

Proof. Let $H := \{H_n\}_{n\geq 1}$ with $H_n\colon \{0,1\}^n \to \{0,1\}^{n-\zeta(n)}$ and $\zeta(n) = \omega(1)$ be an arbitrary polynomial-time function. Let α, β, γ, and δ be constants as specified in the statement of the theorem, (f, g, h) be polynomial-time functions describing an honest randomized Levin reduction from the circuit satisfiability problem Circuit-Sat to Gap-Multi-MCSP$[s, s+s/(\alpha\cdot\log s)]$, and let $\mathsf{i}\mathcal{O}$ be an imperfect indistinguishability obfuscator for general multi-output circuits with output size $\sigma(\lambda, s) = s + s/(\beta\cdot\log s) + \lambda^\gamma$.

We describe a PPT algorithm $\mathcal{A}(x)$ that, given a random input $x \in \{0,1\}^n$, attempts to find a different pre-image of $H_n(x)$.

Input: $x \in \{0,1\}^n$ and random strings $r, r_{\mathsf{i}\mathcal{O}} \in \{0,1\}^{\mathsf{poly}(n)}$, for some $n \geq 1$.

1 Let $y_x := H_n(x)$;

2 Let $\varphi_{y_x}(w)$ be a Boolean circuit that outputs 1 on an input w if and only if $H_n(w) = y_x$;
`// Note that the circuit` φ_{y_x} `depends only on` y_x `and not on` x.

3 Let $(1^\ell, 1^m, 1^s, \mathsf{tt}(F)) := f(\varphi_{y_x}, r)$;
`// Observe that` x `satisfies` $\varphi_{y_x}(w)$.

4 Let $C^{x,r} := g(\varphi_{y_x}, x, r)$ be a circuit with at most s gates (if size $> s$ **return** $\perp$);

5 Let $C^{x,r}_{\mathsf{pad}}$ be a padded version of $C^{y_x,r}$ with exactly s gates;

6 Let $\lambda := n^{\frac{\delta}{2\gamma}}$ and $D := \mathsf{i}\mathcal{O}(1^\lambda, C^{x,r}_{\mathsf{pad}}; r_{\mathsf{i}\mathcal{O}})$;

7 Let $x' := h(\varphi_{y_x}, D, r)$. Then **return** x' if $\varphi_{y_x}(x') = 1$; otherwise **return** $\perp$;

Algorithm 1: Randomized algorithm $\mathcal{A}(x)$ attempts to find a different pre-image of $H_n(x)$.

We establish the following lemma about the correctness of $\mathcal{A}(x)$ on a random input x.

Lemma 17. *The following holds for every sufficiently large n:*

$$\Pr_{x\leftarrow\{0,1\}^n,\,\mathcal{A}}\left[x' \leftarrow \mathcal{A}(x);\ H_n(x') = H_n(x) \text{ and } x \neq x'\right] = \Omega(1).$$

Proof. We adapt the argument from [MP24] to our setting. Let $x \in \{0,1\}^n$, and consider the output $x' = \mathcal{A}(x, r, r_{\mathsf{i}\mathcal{O}})$, where r and $r_{\mathsf{i}\mathcal{O}}$ are random strings. Let y_x, φ_{y_x}, F, s, $C^{x,r}$, $C^{x,r}_{\mathsf{pad}}$, λ, and D be as in the description of $\mathcal{A}$.

We consider a distribution $\mathcal{D}$ over pairs (φ, x), where φ is the description of a Boolean circuit on n-variables and x is an assignment. For a given n, we sample from $\mathcal{D}$ by sampling a string $x \leftarrow \{0,1\}^n$ and setting $\varphi = \varphi_{y_x}$. Note that φ_{y_x} depends only on y_x and not on x.

Since $\zeta(n) = \omega(1)$, for a random $x \leftarrow \{0,1\}^n$, with probability at least $1 - o(1)$, there is $x' \neq x$ such that $H_n(x) = H_n(x')$. Let

$$\mathcal{G} := \{x \in \{0,1\}^n \ \mid\ |H_n^{-1}(H_n(x))| > 1\}.$$

Equivalently, a formula $\varphi_{y_x}(w)$ obtained from $x \in \mathcal{G}$ has more than one satisfying assignment. We might abuse notation and write $\varphi_{y_x} \in \mathcal{G}$ when considering whether $x \in \mathcal{G}$.

Let $(\varphi_{y_x}, x) \in \mathsf{Support}(\mathcal{G})$. We say that a random string r is *good* for φ_{y_x} if Items (*a*) and (*b*) in the statement of Theorem 16 hold for r:

(*a*) Given any satisfying assignment x' of φ_{y_x}, $g(\varphi_{y_x}, x', r)$ outputs a circuit $C^{x',r}$ of size at most s that computes F, and[4]

(*b*) for any circuit D of size at most $s + s/(\alpha \cdot \log s)$ that computes F, $\varphi_{y_x}(h(\varphi_{y_x}, D, r)) = 1$.

Note that for any fixed φ_{y_x} as above, r is good for φ_{y_x} with probability at least $1 - o(1)$ over the choice of r. We let

$$\mathcal{R}_{\varphi_{y_x}} := \{r \in \{0,1\}^{\mathsf{poly}(n)} \mid r \text{ is good for } \varphi_{y_x}\}.$$

Now observe that, for every $x \in \mathcal{G}$ and $r \in \mathcal{R}_{\varphi_{y_x}}$, since x satisfies φ_{y_x}, the output circuit $C^{x,r} = g(\varphi_{y_x}, x, r)$ has size at most s and $C^{x,r}_{\mathsf{pad}}$ is a circuit of size exactly s. Consequently, $D = \mathsf{i}\mathcal{O}(1^\lambda, C^{x,r}_{\mathsf{pad}}; r_{\mathsf{i}\mathcal{O}})$ is a circuit of size at most

$$\begin{aligned}
\sigma(\lambda, s) &= s + s/(\beta \cdot \log s) + \lambda^\gamma \\
&= s + s/(\beta \cdot \log s) + (n^\delta)^{1/2} \\
&\leq s + s/(\beta \cdot \log s) + s^{1/2} \\
&\leq s + s/(\alpha \cdot \log s),
\end{aligned}$$

where the last inequality uses that $\beta > \alpha$ and $s \geq n^\delta$ is sufficiently large. Moreover, for any circuit $C^{x,r}_{\mathsf{pad}}$, with probability at least $1 - o(1)$ over the choice of $r_{\mathsf{i}\mathcal{O}}$, D is equivalent to $C^{x,r}$ and $C^{x,r}_{\mathsf{pad}}$. Let

$$\mathcal{R}^{\mathsf{i}\mathcal{O}}_{x,r} := \left\{r_{\mathsf{i}\mathcal{O}} \in \{0,1\}^{\mathsf{poly}(n)} \mid D = \mathsf{i}\mathcal{O}(1^\lambda, C^{x,r}_{\mathsf{pad}}; r_{\mathsf{i}\mathcal{O}}) \text{ is of size } \leq s + s/(\alpha \cdot \log s) \text{ and computes } F\right\}.$$

As a consequence, for $x \in \mathcal{G}$, $r \in \mathcal{R}_{\varphi_{y_x}}$, and $r_{\mathsf{i}\mathcal{O}} \in \mathcal{R}^{\mathsf{i}\mathcal{O}}_{x,r}$, $x' = h(\varphi_{y_x}, D, r)$ satisfies φ_{y_x}, i.e., $H_n(x') = y_x = H_n(x)$.

It remains for us to argue that we have $x' \neq x$, which we will should to hold with probability $\Omega(1)$ over $x \leftarrow \{0,1\}^n$ and $\mathcal{A}$'s internal randomness r and $r_{\mathsf{i}\mathcal{O}}$. For this, we rely on the security of the $\mathsf{i}\mathcal{O}$ scheme.

In more detail, for fixed $x \in \mathcal{G}$ and $r \in \mathcal{R}_{\varphi_{y_x}}$, and for every x' such that $\varphi_{y_x}(x') = 1$, the corresponding circuits $C^{x,r}_{\mathsf{pad}}$ and $C^{x',r}_{\mathsf{pad}}$ are of the same size and equivalent. This is because F and s depend on r and on $\varphi_{y_x} = \varphi_{y_{x'}}$. For convenience, let $\varphi := \varphi_{y_x} = \varphi_{y_{x'}}$. By the security of the obfuscator and our choice of $\lambda = n^{\Omega(1)}$, the distributions $\mathsf{i}\mathcal{O}(1^\lambda, C^{x,r}_{\mathsf{pad}})$ and $\mathsf{i}\mathcal{O}(1^\lambda, C^{x',r}_{\mathsf{pad}})$ are indistinguishable against non-uniform adversaries of size $\mathsf{poly}(n)$, assuming that n is large enough. Since h performs a polynomial-time computation, by data processing, the distributions

$$h\Big(\varphi, \mathsf{i}\mathcal{O}(1^\lambda, C^{x,r}_{\mathsf{pad}}), r\Big) \quad \text{and} \quad h\Big(\varphi, \mathsf{i}\mathcal{O}(1^\lambda, C^{x',r}_{\mathsf{pad}}), r\Big)$$

(generated from a random string $r_{\mathsf{i}\mathcal{O}}$) are also indistinguishable, even when given as advice r, φ, x, and x'. Therefore, by the definition of $\mathcal{A}$, for any x and x' as above, we get that

$$\begin{aligned}
\mathop{\mathbf{Pr}}_{r, r_{\mathsf{i}\mathcal{O}}}[\mathcal{A}(x, r, r_{\mathsf{i}\mathcal{O}}) = x \mid (r, r_{\mathsf{i}\mathcal{O}}) \in \mathcal{E}_{\varphi,x,x'}] &\leq \mathop{\mathbf{Pr}}_{r, r_{\mathsf{i}\mathcal{O}}}[\mathcal{A}(x', r, r_{\mathsf{i}\mathcal{O}}) = x \mid (r, r_{\mathsf{i}\mathcal{O}}) \in \mathcal{E}_{\varphi,x,x'}] + o(1) \\
&\leq \mathop{\mathbf{Pr}}_{r, r_{\mathsf{i}\mathcal{O}}}[\mathcal{A}(x', r, r_{\mathsf{i}\mathcal{O}}) \neq x' \mid (r, r_{\mathsf{i}\mathcal{O}}) \in \mathcal{E}_{\varphi,x,x'}] + 1/3,
\end{aligned}$$

[4] On a related note, observe that x' and r fully specify $C^{x',r}$ in the computation $\mathcal{A}$, since the circuits φ_{y_x} and $\varphi_{y_{x'}}$ are the same when $H_n(x) = H_n(x')$.

where for convenience we employed the event $\mathcal{E}_{\varphi,x,x'} := \{(r, r_{\mathsf{i}\mathcal{O}}) \mid r \in \mathcal{R}_\varphi \wedge r_{\mathsf{i}\mathcal{O}} \in \mathcal{R}^{\mathsf{i}\mathcal{O}}_{x,r} \cap \mathcal{R}^{\mathsf{i}\mathcal{O}}_{x',r}\}$. This implies that

$$1 - \mathop{\mathbf{Pr}}_{r,r_{\mathsf{i}\mathcal{O}}}[\mathcal{A}(x, r, r_{\mathsf{i}\mathcal{O}}) \neq x \mid (r, r_{\mathsf{i}\mathcal{O}}) \in \mathcal{E}_{\varphi,x,x'}] \;\leq\; \mathop{\mathbf{Pr}}_{r,r_{\mathsf{i}\mathcal{O}}}[\mathcal{A}(x', r, r_{\mathsf{i}\mathcal{O}}) \neq x' \mid (r, r_{\mathsf{i}\mathcal{O}}) \in \mathcal{E}_{\varphi,x,x'}] + 1/3,$$

or equivalently,

$$\frac{1}{2} \cdot \Big(\mathop{\mathbf{Pr}}_{r,r_{\mathsf{i}\mathcal{O}}}[\mathcal{A}(x, r, r_{\mathsf{i}\mathcal{O}}) \neq x \mid (r, r_{\mathsf{i}\mathcal{O}}) \in \mathcal{E}_{\varphi,x,x'}] + \mathop{\mathbf{Pr}}_{r,r_{\mathsf{i}\mathcal{O}}}[\mathcal{A}(x', r, r_{\mathsf{i}\mathcal{O}}) \neq x' \mid (r, r_{\mathsf{i}\mathcal{O}}) \in \mathcal{E}_{\varphi,x,x'}] \Big) \;\geq\; \frac{1}{3}. \quad (2)$$

Finally, recall the definition of the distribution $\mathcal{D}$ over pairs (φ, x). For convenience, we also consider the distribution $\mathcal{D}'$, where we first sample $(\varphi, x_0) \leftarrow \mathcal{D}$, and then if $x_0 \in \mathcal{G}$ we sample a random $x_1 \neq x_0$ such that $\varphi(x_1) = 1$ (otherwise we let $x_1 = x_0$). We then output the triple (φ, x_0, x_1). The following holds:

$$\mathop{\mathbf{Pr}}_{x \leftarrow \{0,1\}^n,\, \mathcal{A}}[x' \leftarrow \mathcal{A}(x);\; H_n(x') = H_n(x) \text{ and } x \neq x'] = \mathop{\mathbf{Pr}}_{(\varphi,x) \leftarrow \mathcal{D}, r, r_{\mathsf{i}\mathcal{O}}}[\mathcal{A}(x, r, r_{\mathsf{i}\mathcal{O}}) \notin \{x, \perp\}]$$

$$= \mathop{\mathbf{Pr}}_{(\varphi,x_0,x_1) \leftarrow \mathcal{D}', r, r_{\mathsf{i}\mathcal{O}}}[\mathcal{A}(x_0, r, r_{\mathsf{i}\mathcal{O}}) \notin \{x_0, \perp\}]$$

$$\geq \mathbf{Pr}[\mathcal{A}(x_0, r, r_{\mathsf{i}\mathcal{O}}) \notin \{x_0, \perp\} \mid \varphi \in \mathcal{G} \wedge (r, r_{\mathsf{i}\mathcal{O}}) \in \mathcal{E}_{\varphi,x_0,x_1}] \cdot \mathbf{Pr}[(r, r_{\mathsf{i}\mathcal{O}}) \in \mathcal{E}_{\varphi,x_0,x_1} \mid \varphi \in \mathcal{G}] \cdot \mathbf{Pr}[\varphi \in \mathcal{G}]$$

$$\geq \mathop{\mathbf{Pr}}_{(\varphi,x_0,x_1) \leftarrow \mathcal{D}', r, r_{\mathsf{i}\mathcal{O}}}[\mathcal{A}(x_0, r, r_{\mathsf{i}\mathcal{O}}) \notin \{x_0, \perp\} \mid \varphi \in \mathcal{G} \wedge (r, r_{\mathsf{i}\mathcal{O}}) \in \mathcal{E}_{\varphi,x_0,x_1}] \cdot (1 - o(1)) \quad (3)$$

$$= \mathop{\mathbf{Pr}}_{(\varphi,x_0,x_1) \leftarrow \mathcal{D}', r, r_{\mathsf{i}\mathcal{O}}, b \leftarrow \{0,1\}}[\mathcal{A}(x_b, r, r_{\mathsf{i}\mathcal{O}}) \notin \{x_b, \perp\} \mid \varphi \in \mathcal{G} \wedge (r, r_{\mathsf{i}\mathcal{O}}) \in \mathcal{E}_{\varphi,x_0,x_1}] \cdot (1 - o(1)) \quad (4)$$

$$= \frac{1}{2} \sum_{b \in \{0,1\}} \mathop{\mathbf{Pr}}_{(\varphi,x_0,x_1) \leftarrow \mathcal{D}', r, r_{\mathsf{i}\mathcal{O}}}[\mathcal{A}(x_b, r, r_{\mathsf{i}\mathcal{O}}) \notin \{x_b, \perp\} \mid \varphi \in \mathcal{G} \wedge (r, r_{\mathsf{i}\mathcal{O}}) \in \mathcal{E}_{\varphi,x_0,x_1}] \cdot (1 - o(1))$$

$$= \frac{1}{2} \sum_{b \in \{0,1\}} \mathop{\mathbf{Pr}}_{(\varphi,x_0,x_1) \leftarrow \mathcal{D}', r, r_{\mathsf{i}\mathcal{O}}}[\mathcal{A}(x_b, r, r_{\mathsf{i}\mathcal{O}}) \neq x_b\} \mid \varphi \in \mathcal{G} \wedge (r, r_{\mathsf{i}\mathcal{O}}) \in \mathcal{E}_{\varphi,x_0,x_1}] \cdot (1 - o(1)) \quad (5)$$

$$\geq \frac{1}{3} \cdot (1 - o(1)) \quad (6)$$

$= \Omega(1)$, where the non-trivial equations are justified as follows:

- Equation (3) uses the definition of the events $\mathcal{G}$ and $\mathcal{E}_{\varphi,x_0,x_1}$ the estimates above, and the union bound;
- Equation (4) employs that the distributions (φ, x_0) and (φ, x_1) are identical for $(\varphi, x_0, x_1) \leftarrow \mathcal{D}'$;
- Equation (5) uses that $\mathcal{A}(x_b, r, r_{\mathsf{i}\mathcal{O}}) \neq \perp$ under events $\varphi \in \mathcal{G}$ and $(r, r_{\mathsf{i}\mathcal{O}}) \in \mathcal{E}_{\varphi,x_0,x_1}$; and
- Equation (6) relies on Equation (2), which is available since $\varphi \in \mathcal{G}$.

This completes the proof of the lemma. ◇

Finally, the lemma implies that there are no infinitely-often one-way hash functions under the assumptions in Theorem 16, which completes the proof. □

4 Hardness of Multi-MCSP Under Randomized Approximate Levin Reductions

The *components* of a multi-output Boolean function $f\colon \{0,1\}^\ell \to \{0,1\}^m$ are the single-output functions that compute the i-th output bit of f for $i \in [m]$. We say a Boolean circuit C computes a multi-output Boolean function f if for each component f_i of f, there is a gate or input wire in C that computes f_i.

In this section, we establish the following result.

Theorem 18 (Hardness of Approximating Multi-MCSP under Levin Reductions)**.** *There exist a constant $a > 0$ such that the following holds. Let $L \in \mathsf{NP}$, and V_L be an NP-verifier for L. There are deterministic polynomial-time functions f, g, and h satisfying the following conditions*:

- *Given a sufficiently large instance x of L and $r \in \{0,1\}^{|x|^c}$, where $c > 0$ is a constant depending only on L, $f(x;r)$ outputs $(1^\ell, 1^m, 1^s, \mathtt{tt}(F))$, where $\ell, m, s \in \mathbb{N}$ and $F\colon \{0,1\}^\ell \to \{0,1\}^m$ is a multi-output function. Moreover, $s \geq \max\{\ell, m\}$ and $s \geq |x|^\kappa$ for some universal constant $\kappa > 0$.*
- *With probability at least $1 - o(1)$ over the choice of r, the following holds*:
  1. *if there is y such that $V_L(x,y) = 1$, then $g(x,y;r)$ outputs a circuit of size at most s that computes F, and*
  2. *if there is a circuit C of size at most $s + s/(a \cdot \log s)$ that computes F, then $V_L(x, h(x,C;r)) = 1$.*

To prove Theorem 18, we will use the following two results. The first states that the Set-Cover problem is hard to approximate.

Lemma 19 (Hardness of Approximating Set-Cover under Levin Reductions)**.** *There exist constants $0 < \lambda < \delta < 1/2$ and an integer $q > 0$ such that the following holds. Let $L \in \mathsf{NP}$, and let V_L be an NP-verifier for L. There are polynomial-time functions f, g, and h satisfying the following conditions:*

- *Given any input instance x for L, $f(x)$ outputs a Set-Cover instance $\varphi_x = (n, S_1, \ldots, S_t)$ such that $n/q \leq t \leq n$ and $|S_j| \leq q$ for every $j \in [t]$. Moreover, $t \geq |x|^\kappa$ for some constant $\kappa > 0$.*
- *If there is y such that $V_L(x,y)$ holds, then there exists a set cover of size $(1-\lambda) \cdot t$ for φ_x. Moreover, $T = g(x,y)$ is such a set cover.*
- *If there is a set cover $T \subseteq [t]$ for φ_x such that $|T| < (1 - \lambda + \delta) \cdot t$, then $x \in L$. Moreover, $V_L(x,y)$ holds for $y := h(x,T)$.*

The second lemma establishes a Levin reduction from Set-Cover to Multi-MCSP.

Lemma 20 (A Randomized Approximate Levin-Reduction from Set-Cover to Multi-MCSP)**.** *Let λ, δ and q be the constants from Lemma 19. There exist a constant $a > 0$ and deterministic polynomial-time functions $\widehat{f}$, $\widehat{g}$, and $\widehat{h}$ satisfying the following conditions*:

- *Given a sufficiently large Set-Cover instance $z := (n, S_1, \ldots, S_t)$ satisfying $n/q \leq t \leq n$ and $|S_j| \leq q$ for every $j \in [t]$, and a string $r \in \{0,1\}^{\mathsf{poly}(n)}$, $\widehat{f}(z;r)$ outputs $(1^{n_{\mathsf{in}}}, 1^{n_{\mathsf{out}}}, 1^s, \mathtt{tt}(F))$, where $n_{\mathsf{in}}, n_{\mathsf{out}}, s \in \mathbb{N}$ and $F\colon \{0,1\}^{n_{\mathsf{in}}} \to \{0,1\}^{n_{\mathsf{out}}}$ is a multi-output function. Moreover, $s \geq \max\{n_{\mathsf{in}}, n_{\mathsf{out}}\}$ and $s > t/2$.*
- *With probability at least $1 - o(1)$ over the choice of r, the following holds*:

1. *if there is a set cover $S \subseteq [t]$ for z such that $|S| \leq (1-\lambda)\cdot t$, then $\widehat{g}(z, S; r)$ outputs a circuit of size at most s that computes F, and*
2. *if there is a circuit C of size at most $s + s/(a \cdot \log s)$ that computes F, then $\widehat{h}(z, C; r)$ outputs a set cover $S \subseteq [t]$ for z such that $|S| < (1-\lambda+\delta)\cdot t$.*

We prove Lemma 19 in Section 4.1 and Lemma 20 in Section 4.2, but first let us use Lemma 19 and Lemma 20 to prove Theorem 18.

Proof of Theorem 18. Let $a > 0$ be the constant specified in Lemma 20.

Fix an $L \in \mathsf{NP}$ with NP-verifier V_L, and let (f_0, g_0, h_0) be the Levin reduction from L to Set-Cover promised by Lemma 19. Let $(\widehat{f}, \widehat{g}, \widehat{h})$ be the randomized reduction from Set-Cover to Multi-output MCSP promised by Lemma 20. Define $f(x; r) = \widehat{f}(f_0(x); r)$, $g(x, y; r) = \widehat{g}(f_0(x), g_0(x, y); r)$ and $h(x, C; r) = h_0(x, \widehat{h}(f_0(x), C; r))$.

By Lemma 19, $f_0(x)$ outputs a Set-Cover instance $(n, S_1, \ldots, S_t)$ with $n/q \leq t \leq n$, $|S_j| \leq q$ for every $j \in [t]$ and $t \geq |x|^\kappa$ for some constant $\kappa > 0$. Thus, by Lemma 20, $f(x; r) = \widehat{f}(f_0(x); r)$ outputs a Multi-MCSP instance $(1^{n_{\mathsf{in}}}, 1^{n_{\mathsf{out}}}, 1^s, \mathtt{tt}(F))$ with $s > t/2 \geq |x|^\kappa/2$ and $s \geq \max\{n_{\mathsf{in}}, n_{\mathsf{out}}\}$.

Moreover, for every y such that $V_L(x, y)$ holds, $g_0(x, y)$ outputs a set cover of size $(1-\lambda)\cdot t$. By Lemma 20, with probability $1 - o(1)$ over r, $g(x, y; r) = \widehat{g}(f_0(x), g_0(x, y); r)$ outputs a circuit of size at most s that computes F. Finally, with the same probability over r, if there is a circuit C of size at most $s + s/(a \cdot \log s)$ that computes F, then $\widehat{h}(f_0(x), C; r)$ outputs a set cover $S \subseteq [t]$ for $f_0(x)$ such that $|S| < (1-\lambda+\delta)\cdot t$. Thus, by Lemma 19, $h(x, C; r) = h_0(x, \widehat{h}(f_0(x), C; r))$ outputs y such that $V_L(x, y)$ holds. □

4.1 Hardness of Approximating Set-Cover Under Levin Reductions

4.1.1 Hardness of Approximating k-SAT Under Levin Reductions

Let k be a sufficiently large positive integer. The PCP Theorem states that, given a k-CNF $\varphi(x_1, \ldots x_n)$, it is NP-hard to distinguish the following two cases:

- φ is satisfiable; and
- no assignment satisfies more than an $(1-\gamma_k)$-fraction of clauses in φ, where $\gamma_k > 0$ is a universal constant.

It is a folklore result that this computational task remains NP-hard under Levin reductions. In this section, we provide more details about this claim.

Instead of inspecting an existing proof of the PCP Theorem, we rely in a black-box way on a result of [Pic15] showing that the PCP Theorem can be formalized in the bounded arithmetic theory PV_1. To achieve this, we use that PV_1 is a theory for polynomial-time computations, and that the proof of correctness of the PCP Theorem in this theory implies the existence of polynomial-time functions that exactly match the definition of a Levin reduction. While we assume basic familiarity with bounded arithmetic (see, e.g., [Bus97]), the argument described below is not difficult to follow.

We will need the following definition (expressible in PV_1).

Definition 21 ([Pic15, Definition 3.5])**.** *Let k, c, q, ℓ be constants, $m \in \mathsf{Log}$, $r \in [\ell \cdot m^\ell]$, $\pi \in \{0,1\}^{c \cdot m^c}$. Let D be an algorithm that runs in time at most $k \cdot m^k$ on an input y of length m. We let $D^\pi(y, r) \in \{0,1\}$ denote the output of the computation of D on y with oracle access to π using r as a source of randomness, which proceeds as follows:*

1. *Given y and r, D makes at most q non-adaptive queries to π, represented by the coordinates $i_1, \ldots, i_q \in [|\pi|]$.*

2. *Given y and the bits $\pi(i_1), \ldots, \pi(i_q)$, D computes its output in $\{0,1\}$.*

Theorem 22 (PCP Theorem in PV_1 [Pic15, Theorem 7])**.** *There are positive constants k, c, q, and ℓ and a $k \cdot m^k$-time algorithm D (given as a PV_1-function) computing according to Definition 21 There are positive constants k, c, q, and ℓ and a $k \cdot m^k$-time algorithm D (given as a PV_1-function) computing according to Definition 21 such that PV_1 proves that for any $m \in \mathsf{Log}$ and Boolean formula $\psi \in \{0,1\}^m$,*

(*i*) $\exists x\, \mathsf{SAT}(\psi, x) \to \exists \pi \in \{0,1\}^{c \cdot m^c}\ \mathbf{Pr}_{r \in [\ell \cdot m^\ell]}[D^\pi(\psi, r) = 1] = 1\,.$

(*ii*) $\forall x\, \neg\mathsf{SAT}(\psi, x) \to \forall \pi \in \{0,1\}^{c \cdot m^c}\ \mathbf{Pr}_{r \in [\ell \cdot m^\ell]}[D^\pi(\psi, r) = 1] \leq 1/2\,.$

We derive the following consequence from Theorem 22.

Corollary 23 (PCP Theorem Under Levin Reductions (Folklore))**.** *Let k, c, q, ℓ be the constants and D be the polynomial-time algorithm from Theorem 22. There are polynomial-time functions g_{PCP} and h_{PCP} for which the following conditions hold*:

1. *Given a formula $\psi \in \{0,1\}^m$ and an assignment x that satisfies ψ, $\pi := g_{\mathsf{PCP}}(\psi, x) \in \{0,1\}^{c \cdot m^c}$ satisfies $\mathbf{Pr}_{r \in [\ell \cdot m^\ell]}[D^\pi(\psi, r) = 1] = 1\,.$*

2. *Given a formula $\psi \in \{0,1\}^m$ and $\pi \in \{0,1\}^{c \cdot m^c}$ such that $\mathbf{Pr}_{r \in [\ell \cdot m^\ell]}[D^\pi(\psi, r) = 1] > 1/2$, the assignment $x := h_{\mathsf{PCP}}(\psi, \pi)$ satisfies ψ.*

Proof. The result follows from Theorem 22 and the witnessing theorem for $\forall\Sigma^b_1$-sentences available in PV_1. More generally, any proof of NP-hardness under Karp-reductions that can be formalized in PV_1 implies NP-hardness under Levin reductions. We provide more details below.

First, consider Item (*i*) in Theorem 22. The condition $\mathbf{Pr}_{r \in [\ell \cdot m^\ell]}[D^\pi(\psi, r) = 1] = 1$ is expressed as a polynomial-time predicate $A(\psi, \pi)$, since there are at most polynomially many choices of r. Consequently, the correctness of this direction of the reduction is captured by a PV_1-sentence of the following form:

$$\forall 1^m\, \forall \psi \in \{0,1\}^m\, \exists x\ (|x| \leq m \wedge \mathsf{SAT}(\psi, x)) \to (\exists \pi\ (|\pi| \leq c \cdot m^c \wedge\ A(\psi, \pi))\ .$$

This can be equivalently expressed by the following sentence:

$$\forall 1^m\, \forall \psi \in \{0,1\}^m\, \forall x \in \{0,1\}^{\leq m}\, \exists \pi \in \{0,1\}^{c \cdot m^c}\ (\neg\mathsf{SAT}(\psi, x) \vee A(\psi, \pi))\ .$$

Since the latter is a $\forall\Sigma^b_1$-sentence, by the witnessing theorem for PV_1 its provability yields a polynomial-time function $g(1^m, \psi, x)$ such that $A(\psi, \pi)$ holds with $\pi := g(1^m, \psi, x)$ whenever $\mathsf{SAT}(\psi, x)$ holds.

A similar argument in the other direction (i.e., Item (*ii*)) establishes the existence of a polynomial-time function $x = h(\psi, \pi)$ with the desired property. □

Hardness of approximating k-SAT under Levin reductions. We now combine the Cook-Levin reduction and the standard translation between the PCP Theorem and the hardness of approximating CSPs to derive the NP-hardness of approximating k-SAT under Levin reductions.

Given a k-CNF ψ and an assignment x, we let $\mathsf{val}(\psi, x) \in [0,1]$ denote the fraction of clauses of ψ that are satisfied by x. We will use the same notation for CSPs.

Theorem 24 (Hardness of Approximating k-SAT under Levin Reductions)**.** *There is an integer $k \geq 1$ such that the following holds. Let $L \in \mathsf{NP}$, and let V_L be an NP-verifier for L. There are polynomial-time functions f_1, g_1, and h_1 satisfying the following conditions:*

1. *Given any input string x, $\varphi_x := f_1(x)$ is a k-CNF with $|x|^{\Omega(1)}$ clauses.*
2. *If there is y such that $V_L(x, y)$ holds, then φ_x is satisfiable. Moreover, $z := g_1(x, y)$ satisfies φ_x.*
3. *If there is an assignment w such that $\mathsf{val}(\varphi_x, w) > 1 - 1/2^{k+1}$, then $x \in L$. Moreover, $V_L(x, y)$ holds for $y := h_1(x, w)$.*

Proof. Since the proof employs standard ideas, we only sketch the argument. Given any L and V_L as above, the Cook-Levin Theorem provides a triple f_{CL}, g_{CL}, and h_{CL} that describes a Levin reduction to the Boolean formula satisfiability problem. We compose this Levin reduction with the reduction from Corollary 23, obtaining polynomial-time functions $\widetilde{f}$, $\widetilde{g}$, and $\widetilde{h}$ with the following properties:

1. Given any input string $x \in \{0,1\}^n$, $\psi_x := \widetilde{f}(x)$ is a Boolean formula of description length $m = \mathsf{poly}(n)$.
2. If $V_L(x, y)$ holds for some y, then $\pi_x := \widetilde{g}(x, y) \in \{0,1\}^{c \cdot m^c}$ satisfies $\mathbf{Pr}_{r \in [\ell \cdot m^\ell]}[D^{\pi_x}(\psi_x, r)] = 1$.
3. Given any $\pi \in \{0,1\}^{c \cdot m^c}$ such that $\mathbf{Pr}_{r \in [\ell \cdot m^\ell]}[D^{\pi}(\psi_x, r) = 1] > 1/2$, $V_L(x, y)$ holds for $y := \widetilde{h}(x, \pi)$.

Next, the standard correspondence between PCPs and CSPs maintains the gap between yes and no instances of L, i.e., an instance x is mapped to a Boolean CSP instance $\widetilde{\varphi}_x$ with constraints of arity k such that $\mathsf{val}(\widetilde{\varphi}_x) = 1$ if $x \in L$ and $\mathsf{val}(\widetilde{\varphi}_x) \leq 1/2$ if $x \notin L$. Finally, we convert $\widetilde{\varphi}_x$ into a k-SAT formula φ_x with the properties stated in Theorem 24. It is not difficult to check that the transformation from PCPs to CSPs and k-SAT instances remains a Levin reduction. □

4.1.2 Proof of Lemma 19

We continue the proof with the chain of reductions from k-SAT to *bounded* Set-Cover. Since the arguments are rather standard, we only sketch some of them.

Lemma 25 (From k-SAT to k-SAT with Bounded Variable Occurrences; Following [PY91])**.** *There exist a constant $d \geq 1$ and polynomial-time functions f_2, g_2 and h_2 such that the following hold.*

- *For any integer $k > 1$, given a k-CNF formula φ with m clauses, $f_2(\varphi)$ is a k-CNF formula with $m' := m + kdm$ clauses such that each variable occurs in at most $2d + 1$ clauses.*
- *If v is an assignment such that $\mathsf{val}(\varphi, v) = 1$, then $g_2(\varphi, v)$ outputs an assignment u such that $\mathsf{val}(f_2(\varphi), u) = 1$.*
- *Given an assignment u such that $\mathsf{val}(f_2(\varphi), u) > 1 - \frac{1}{2^{k+1} \cdot (kd+1)}$, $h_2(\varphi, S)$ outputs an assignment v such that $\mathsf{val}(\varphi, v) > 1 - \frac{1}{2^{k+1}}$.*

Proof. The reduction follows closely the one presented in [PY91, Theorem 2]. We verify that it yields a Levin reduction. We will need the following construction of expanders.

Claim 26 (See, e.g., [RVW00])**.** *There exist a constant $d > 0$ and a polynomial-time algorithm A such that, given as input 1^m where $m \in \mathbb{N}$, A outputs a d-regular graph $G = (V, E)$ such that $|V| = m$ and for every subset $S \subseteq V$ with $|S| \leq |V|/2$, the number of edges $E(S, V - S)$ having one endpoint in S and one in $V - S$ is at least $|S|$.*

The reduction f_2. Let φ be a k-CNF formula with m clauses and variables $x_1, x_2, \cdots, x_n$. Without loss of generality, assume that each clause has exactly k literals. For $i \in [n]$, let ℓ_i be such that x_i (either positively or negatively) occurs in ℓ_i clauses. We construct the new formula $f_2(\varphi)$ as follows. For each variable x_i, we introduce the set of ℓ_i variables

$$V_{x_i} := \{x_i^1, \cdots, x_i^{\ell_i}\}.$$

Then using Claim 26, we construct a graph G_{x_i} whose vertices are V_{x_i}. Next, for each edge (x_i^u, x_i^v) in G_{x_i}, we create two clauses

$$(\overline{x_i^u} \vee x_i^v) \text{ and } (x_i^u \vee \overline{x_i^v}).$$

Note that if $x_i^u = x_i^v$ then both clauses are satisfied and otherwise exactly one clause is satisfied.

We do the above for each x_i, $i \in [n]$, and obtain a formula ϕ with variables $\cup_{i\in[n]} V_{x_i}$. Note that by construction and by the fact that each G_i is d-regular, the number of clauses in ϕ is

$$\sum_{i\in[n]} \frac{\ell_i \cdot d}{2} \cdot 2 = kdm.$$

Also, each variable occurs in at most $2d$ clauses.

Next, we take the original formula φ, and for each variable x_i, we replace every occurrence of x_i in φ by a distinct variable from V_{x_i}. Let ϕ' be the resulting formula.

Finally, $f_2(\varphi)$ outputs

$$\phi' \wedge \phi.$$

It is easy to verify that the above formula has $m' := m + kdm$ clauses and that each variable occurs in at most $2d + 1$ clauses.

The function g_2. If v is an assignment for the variables in φ that satisfies all the m clauses, then by letting the variables in V_{x_i} be $v(x_i)$ for each x_i, we obtain an assignment that satisfies all the m' clauses in $f_2(\varphi)$.

The function h_2. Let u be an assignment for the variables $\cup_{i\in[n]} V_{x_i}$ that satisfies more than

$$\alpha := \left(1 - \frac{1}{2^{k+1} \cdot (kd+1)}\right) \cdot m'$$

clauses in $f_2(\varphi)$. We can obtain an assignment for the variables $x_1, x_2, \cdots, x_n$ in φ as follows. We take the assignment u for $\cup_{i\in[n]} V_{x_i}$, and for each x_i, we set the values of the variables in V_{x_i} to be the majority of $(u(x_i^1), \cdots, u(x_i^{\ell_i}))$, breaking ties arbitrarily for each V_{x_i} if the number of 1s is exactly $\ell_i/2$. Note that this allows us to obtain a single truth value for each x_i and hence an assignment for the variables $x_1, x_2, \cdots, x_n$ in φ.

The key observation here is that changing the values of the variables in each V_{x_i} in this way does not decrease the number of clauses satisfied in $f_2(\varphi)$. To see this, let S be the set of variables in V_{x_i} that do not have the majority-truth-value. Then flipping the values of variables in S can make at most $|S|$ clauses in ϕ' that were previously satisfied become unsatisfied. However, by the expander property of G_i, there are also at least $|S|$ of the clauses in ϕ that were previously unsatisfied (which correspond to the edges in $E(S, V_{x_i} - S)$) and that become satisfied. As a result, we can obtain a new assignment u' for the variables

$\cup_{i\in[n]} V_{x_i}$, where for each $i \in [n]$ all the variables in V_{x_i} have the same truth value, that satisfies more than α clauses in $f_2(\varphi)$. In particular, less than

$$m' - \alpha = \frac{m'}{2^{k+1} \cdot (kd+1)}$$

clauses in ϕ' are unsatisfied. Such an u' yields an assignment for the variables $x_1, x_2, \cdots, x_n$ that satisfies more than

$$m - \frac{m'}{2^{k+1} \cdot (kd+1)} = m \cdot \left(1 - \frac{1}{2^{k+1}}\right)$$

clauses in φ. □

Lemma 27 (From k-SAT with Bounded Variable Occurrences to Independent-Set with Bounded Degree)**.** *There exist polynomial-time functions f_3, g_3 and h_3 such that the following hold.*

- *For any integers $k, B > 1$, given a k-CNF formula φ with m clauses such that each variable occurs in at most B clauses, $f_3(\varphi)$ is is a graph $G = (V, E)$ with degree at most $\Delta := k + B - 2$. Moreover $|V| \geq |\varphi|^\delta$ for some constant δ.*
- *If v is an assignment such that $\mathsf{val}(\varphi, v) = 1$, then $g_3(\varphi, v)$ outputs a set of vertices $S \subseteq V$ of size $|V|/k$ such that S is an independent set in G.*
- *For any $0 \leq \varepsilon \leq 1$, given any independent set $S \subseteq V$ in G of size greater than $(1-\varepsilon) \cdot |V|/k$, $h_2(\varphi, S)$ outputs an assignment v such that $\mathsf{val}(\varphi, v) > 1 - \varepsilon$.*

Proof Sketch. The lemma follows from the textbook reduction from k-SAT to Independent-Set. Given a k-CNF formula φ with m clauses such that each variable occurs in at most B clauses, we construct a graph G with one vertex for every occurrence of every literal (i.e., there are km vertices). We add an edge between each pair of vertices whose corresponding literals are within the same clauses (i.e., the vertices within each clauses form a k-clique). Also, we add an edge between two vertices u and v, if the literals ℓ_u and ℓ_v are complementary to each other. Note that since every variable occurs in at most B clauses, the degree is at most $(k-1)+(B-1) = k+B-2$. Given an assignment that satisfies t clauses in φ, it is easy to construct an independent set of size t for G, and vice versa. □

Lemma 28 (From Independent-Set with Bounded Degree to Vertex-Cover with Bounded Degree)**.** *There exist polynomial-time functions f_4, g_4 and h_4 such that the following hold.*

- *Given a graph $G = (V, E)$, $f_4(G)$ outputs a graph $G' = (V', E')$ such that $|V'| = |V|$ and $|E'| = |E|$.*
- *Given an independent set $S \subseteq V$ in G, $g_4(G, S)$ outputs a set $S' \subseteq V'$ such that $|S'| = |V'| - |S|$ and S' is a vertex cover of G'.*
- *Given a vertex cover $S' \subseteq V'$ of G', $h_4(G, S')$ outputs an independent set $S \subseteq V$ in G of size $|S| = |V| - |S'|$.*

Proof Sketch. The functions f_4, g_4, h_4 are defined as follows: $f_4(G) = G$, $g_4(G, S) = V \setminus S$, $h_4(G, S') = V' \setminus S'$. □

Lemma 29 (From Vertex-Cover with Bounded Degree to Set-Cover with Bounded Set Size)**.** *There exist polynomial-time functions f_5, g_5 and h_5 such that the following holds.*

- *Given a graph $G = (V, E)$ with degree at most $\Delta \geq 1$, $f_5(G)$ outputs a set cover instance $z = (n, S_1, \ldots, S_t \subseteq [n])$ where $n = |E|$, $t = |V|$ and $|S_i| \leq \Delta$ for all $i \in [t]$.*
- *Given a vertex cover $S \subseteq V$ in G, $g_5(G, S)$ outputs a set cover T of size $|S|$ of z.*
- *Given a set cover $T \subseteq [t]$ of z, $h_5(G, T)$ outputs a vertex cover $S \subseteq V$ in G of size $|T|$.*

Proof Sketch. The function f_5 is defined as follows: Given a graph G, let $n = |E|$ and $t = |V|$, and assume without loss of generality that $V = [t]$. For every vertex $v \in [t]$, let S_v be the set of all edges in E that touch V. f_5 outputs $\varphi = (n, S_1, \ldots, S_t)$. Let $g_5(G, S) = S$ and $h_5(G, T) = T$. □

We are now ready to complete the proof of Lemma 19.

Proof Sketch of Lemma 19. Let k be the integer promised by Theorem 24, and let $(f_1, g_1, h_1), \ldots, (f_5, g_5, h_5)$ be the functions promised by Theorem 24 and lemmas 25 and 27 to 29 respectively. The proof follows by composition of Levin reductions.

Define $f(x) = f_5(\ldots f_2(f_1(x)))$. Let $x_1 = f_1(x), x_2 = f_2(x_1), \ldots, x_5 = f_5(x_4)$ (so that $f(x) = x_5$). Similarly, given a witness y for x, let $y_1 = g_1(x, y), y_2 = g_2(x_1, y_1), \ldots, y_5 = g_5(x_4, y_4)$ and define $g(x, y) = y_5$. Finally, given T, let $T_4 = h_5(x_4, T), T_3 = h_4(x_3, T_4), \ldots, T_0 = h_1(x, T_1)$. Define $h(x, T) = T_0$.

It is straightforward to verify the conditions stated in the theorem by inspecting the statements of Theorem 24 and Lemmas 25 and 27 to 29. We omit the details here. □

4.2 A Randomized Approximate Levin-Reduction from Set-Cover to Multi-MCSP

This subsection is devoted to proving Lemma 20. We begin with some preliminaries.

4.2.1 Preliminaries

Let $n, m \in \mathbb{N}$ be powers of two and $m > n$. Let $T \in \{0,1\}^m$. We identify T with a function mapping $\log m$ bits to 1 bit whose truth table is given by T.

We will first define a canonical partition of the domain of T, which we identify with $[m]$, into n parts. Let $\mathcal{P} = (P_1, \ldots, P_n)$ be the partition of $[m]$ into n parts given by

$$P_i = \left\{ j + (i-1) \cdot \frac{m}{n} \mid j = 1, \ldots, \frac{m}{n} \right\}.$$

Note that such a partition allows us to divide the m-bit string T into n consecutive *segments*, each of which is an (m/n)-bit string.

For each $i \in [n]$, we define $T_{\langle i \rangle} \colon \{0,1\}^{\log m} \to \{0,1\}$ to be the following function.

$$T_{\langle i \rangle}(x) = \begin{cases} T(x) & \text{if } x \in P_i, \\ 0 & \text{otherwise.} \end{cases}$$

That is, the truth table of $T_{\langle i \rangle}$ is an m-bit string whose i-th segment is identical to that of T and 0 elsewhere.

We generalize the above to a subset of $[n]$. For $S \subseteq [n]$, we define $T_{\langle S_i \rangle} \colon \{0,1\}^{\log m} \to \{0,1\}$ to be the following function.

$$T_{\langle S \rangle}(x) = \begin{cases} T(x) & \text{if } x \in \bigcup_{i \in S} P_i, \\ 0 & \text{otherwise.} \end{cases}$$

That is, the truth table of $T_{\langle S \rangle}$ is an m-bit string whose i-th segment is identical to that of T *for every* $i \in S$ and 0 elsewhere.

Canonical Circuits. For $S \subseteq [n]$, we describe a single "canonical" Boolean circuit for computing $T_{\langle S \rangle}$, denoted as $\mathbf{CKT}_{T_{\langle S \rangle}}$. More specifically, we construct $\mathbf{CKT}_{T_{\langle S \rangle}}$ as follows.

> For each $i \in S$, we construct a circuit C_i computing $T_{\langle i \rangle}$ as follows. We first construct a circuit C' that, given $x \in \{0,1\}^{\log m}$, outputs 1 if and only if $x \in P_i$. Note that C' only needs to look at the most significant $\log n$ bits of x and can be constructed using at most $O(\log n)$ gates. We then construct a circuit C'' that takes the least significant $\log(m/n)$ bits of x and computes a function $f \colon \{0,1\}^{\log(m/n)} \to \{0,1\}$ whose truth table is given by the i-th segment of T. Note that by Proposition 5, C'' can be constructed using $O(m/(n \cdot \log(m/n)))$ gates. We then let
>
> $$C_i(x) := C'(x) \wedge C''(x).$$
>
> Note that each C_i has size at most $O(m/n + \log n)$. After we have C_i for every $i \in S$, we let
>
> $$\mathbf{CKT}_{T_{\langle S \rangle}}(x) := \bigvee_{i \in S} C_i(x).$$

It is easy to see that $\mathbf{CKT}_{T_{\langle S \rangle}}$ can be obtained in polynomial-time given the truth table of $T_{\langle S \rangle}$.

4.2.2 Proof of Lemma 20

In this subsection we prove Lemma 20.

Proof of Lemma 20. We give a probabilistic polynomial-time many-one reduction with one-sided error from an $(1 + \Omega(1))$-approximation of Set-Cover to an *additive* approximation of Multi-MCSP.

Let λ, δ and q be the constants from Lemma 19. Let $a, a' > 0$ be two sufficiently large constants specified later.

First of all, given a Set-Cover instance $z' := (n', S_1, \ldots, S_t)$, we can transform it into another instance $z := (n, S_1, \ldots, S_t, S_{t+1})$, where n is the least power of two that is at least n', and $S_{t+1} := [n] \setminus [n']$. Then given a set cover for z' of some size ℓ, we can construct a set cover for z of size $\ell + 1$, and vice versa. Therefore, it suffices to show polynomial-time functions $\widehat{f}$, $\widehat{g}$ and $\widehat{f}$ such that for every sufficiently large set cover instance $z := (n, S_1, \ldots, S_t, S_{t+1})$ such that n is a power of two, $n/(2q) \leq t \leq n$ and $|S_i| \leq q$ for every $i \in [t]$, with probability at least $1 - o(1)$ over the choice of r, for $(1^{n_{\mathsf{in}}}, 1^{n_{\mathsf{out}}}, 1^s, \mathtt{tt}(F)) := \widehat{f}(z; r)$,

1. if there is a set cover (encoded by a subset S of indexes) for z of size at most $(1 - \lambda) \cdot t + 1$, then $\widehat{g}(z, S; r)$ outputs a circuit of size at most s that computes F, and
2. if there is a circuit C of size at most $s + s/(a \cdot \log s)$ that computes F, then $\widehat{h}(z, C; r)$ outputs a set cover for z of size less than $(1 - \lambda + \delta) \cdot t + 1$.

We proceed to show the above.

The reduction $\widehat{f}$. Given an instance $(n, \mathcal{S} := (S_1, \ldots, S_{t+1}))$ of the Set-Cover problem and a string $r \in \mathsf{poly}(n)$, the reduction $\widehat{f}$ proceeds as follows. Let m be the least power of two greater than $a' \cdot n \log n$. Note that $m \leq 2a' \cdot n \log n$. Also, let $T \in \{0,1\}^m$ be obtained from the first m bits of r. We view T as the truth table of a function mapping $\log m$ bits to 1 bit.

Next, we define a multi-output function G. Let $C_{\langle T, \mathcal{S} \rangle}$ be the circuit built as follows.

We start by letting $C_{(T,\mathcal{S})}$ be the empty circuit. For each $S \in \mathcal{S}$, we first construct $\mathbf{CKT}_{T_{\langle S \rangle}}$. We then iterate through the gates u in $\mathbf{CKT}_{T_{\langle S \rangle}}$ in *topological* order. Let $\diamond \in \{\wedge, \vee, \neg\}$ be the gate type of u. If u computes a function that is already computed by $C_{(T,\mathcal{S})}$, then ignore it. Otherwise, add a $\diamond$ gate to $C_{(T,\mathcal{S})}$ that takes as input(s) those gate(s) in $C_{(T,\mathcal{S})}$ that compute the function(s) which are fed as inputs to u in $\mathbf{CKT}_{T_{\langle S \rangle}}$. (Note that we are guaranteed to find such gates in $C_{(T,\mathcal{S})}$ since we are iterating in topological order).

We denote by $\mathcal{O}_S$ the (ordered) set of newly added gates during the iteration that corresponds to the set $S \in \mathcal{S}$.

We then define

$$G(x) := x_1 \bullet \cdots \bullet x_{\log m} \bigbullet_{u \in \mathcal{O}_{S_1}} u(x) \ldots \bigbullet_{u \in \mathcal{O}_{S_{t+1}}} u(x). \tag{7}$$

That is, on input $x \in \{0,1\}^{\log m}$, G outputs x together with the value of each gate in $C_{(T,\mathcal{S})}$ when evaluating on x.

Let γ be the number of components of G that are not functions computed by an input wire. That is,

$$\gamma := |\{G_i : G_i \text{ is a component of } G \text{ and } G_i \neq x_j \text{ for all } j \in [\log m]\}|.$$

Note that by construction γ is the number of gates in the circuit $C_{(T,\mathcal{S})}$ that defines G. Also, we observe the following upper bound for γ, which will be useful later.

Claim 30. *We have $\gamma \leq O(m) + q \cdot t + n$.*

Proof of Claim 30. Recall the definition of a canonical circuit from the previous subsection. Note that by construction, γ is the number of internal gates in the circuit $C_{(T,\mathcal{S})}$. In constructing $C_{(T,\mathcal{S})}$, we start with empty circuit, and build the circuits $\mathbf{CKT}_{T_{\langle S_1 \rangle}}, \cdots, \mathbf{CKT}_{T_{\langle S_{t+1} \rangle}}$, each of which computes $T_{\langle S_j \rangle}$ and is of the form $\bigvee_{i \in S_j} C_i$, where each C_i computes $T_{\langle i \rangle}$. We then add their gates into $C_{(T,\mathcal{S})}$. However, since we never add two gates that compute the same function, we will never add the gates in C_i more than once for each $i \in [n]$. Then the total number of gates that need to be added for computing $\{C_i\}_{i \in [n]}$ is at most

$$O\left(\frac{m}{n \cdot \log(m/n)} + \log n\right) \cdot n \leq O(m).$$

In addition to $\{C_i\}_{i \in [n]}$, we have gates that are used to compute $T_{\langle S_j \rangle}$, for all $j \in [t+1]$. For this, we add at most qt gates for each S_j, where $j \leq t$, since $|S_j| \leq q$ for each such S_j, and at most n gates for S_{t+1}, giving that we need at most $qt + n$ more gates. The total number gates is upper bounded by $O(m) + qt + n$, as desired. $\diamond$

Finally, let $F \colon \{0,1\}^{\log m} \to \{0,1\}^{1+\log m+\gamma}$ be the function defined as

$$F(x) := T(x) \bullet G(x).$$

The reduction then outputs $(1^{\log m}, 1^{1+\log m+\gamma}, 1^s, \mathtt{tt}(F))$ for $s := \gamma + \lceil (1-\lambda) \cdot t \rceil + 1$. Observe that s is bounded by a polynomial in n, t.

First, we argue that this procedure runs in polynomial time. In particular, we argue that we can compute the truth table of F and γ efficiently. To show the former, it suffices to show that we can compute the truth table of G efficiently. This is indeed the case since we can construct the circuit $C_{(T,\mathcal{S})}$ in time $\mathsf{poly}(m,t)$. Then to compute the truth table of G, we just need to evaluate $C_{(T,\mathcal{S})}$ on every $x \in \{0,1\}^{\log m}$. Note that this also implies that we can compute γ efficiently, since by construction γ is the number of gates in $C_{(T,\mathcal{S})}$.

Now, we will argue for the correctness of the reduction.

The completeness and the function $\widehat{g}$. We show the following regarding the completeness of our reduction.

Claim 31. *There is a deterministic polynomial-time function $\widehat{g}$ such that if there is a set cover S for the instance $(n, \mathcal{S})$ of size at most $\ell := (1-\lambda)\cdot t + 1$, then $\widehat{g}(n, \mathcal{S}, S; T)$ outputs a circuit of size s that computes F.*

Proof of Claim 31. Suppose there is a set cover $S_1, \ldots, S_\ell$ for $(n, \mathcal{S})$. From such a set cover, we construct a circuit of size at most s that computes F.

We first compute a circuit for G. In particular, we will construct the circuit $C_{(T,\mathcal{S})}$ described above. Note that $C_{(T,\mathcal{S})}$ can be built efficiently in time $\mathsf{poly}(m, t)$. Also, it has size γ.

To compute T, note that by construction and the fact that $(S_1, \ldots, S_\ell)$ covers $[n]$, we have that

$$T(x) = T_{\langle S_1 \rangle}(x) \vee \cdots \vee T_{\langle S_\ell \rangle}(x).$$

Since $T_{\langle S_1 \rangle}, \ldots, T_{\langle S_\ell \rangle}$ are components of G, using additional ℓ gates, we can compute T as well. This yields a circuit of size at most $\gamma + \ell \leq s$ for F, as desired. ◇

The soundness and the function $\widehat{h}$. Next, we show the following regarding the soundness of our reduction.

Claim 32. *There is a deterministic polynomial-time function $\widehat{h}$ such that the following holds with probability $1 - o(1)$ over the choice of T. Given any circuit C of size at most $s + s/(a \cdot \log s)$ that computes F, $\widehat{h}(n, \mathcal{S}, C; T)$ outputs a set cover of size less than $(1 - \lambda + \delta)\cdot t + 1$.*

Proof of Claim 32. We start with the description of $\widehat{h}$.

Fix any T, and let C be a circuit computing F using at most $s + s/(a \cdot \log s)$ gates.

We first argue that γ of the gates in C must compute the components of G that are not functions computed by an input wire. Suppose we have a circuit that computes G. Then every distinct component of G has a (necessarily distinct) input wire or gate from C that computes that component. Therefore, since G has γ distinct components that are not computed by an input wire, C must have at least γ distinct gates computing components of G.

It follows that there is a circuit D that takes $(\log m + \gamma)$ input bits and has at most

$$s + s/(a \cdot \log s) - \gamma = \lceil (1-\lambda)\cdot t \rceil + 1 + s/(a \cdot \log s) =: \alpha$$

gates such that

$$D(x, G_1(x), \ldots, G_\gamma(x)) = T(x)$$

for all $x \in \{0,1\}^{\log m}$, where $G_1, \ldots, G_\gamma$ are the unique components of G. Moreover, since D has only α gates of fan-in 2, it uses at most $\alpha + 1$ of the components of G in the circuit. This follows from the fact that a tree with α internal nodes has at most $\alpha + 1$ leaves. Thus, after a possible relabeling of $G_1, \ldots, G_\gamma$, we can assume D takes at most $(\log m + \alpha + 1)$ input bits and that

$$D(x, G_1(x), \ldots, G_{\alpha+1}(x)) = T(x). \tag{8}$$

For each $j \in [\alpha + 1]$, let $S_j \in \mathcal{S}$ be a set such that G_j is a component that corresponds to a gate in $\mathcal{O}_{S_j}$ (recall the definition of G in Equation (7)). Note that D, $\{G_j\}$ and $\{S_j\}$ can be computed efficiently in time $\mathsf{poly}(m, t)$ given the inputs of $\widehat{h}$. Finally, we use $S_1, \ldots, S_{\alpha+1}$ to construct a set cover. Let $I := \cup_{j \in [\alpha+1]} S_j$.

Then we can add $n-|I|$ sets $S_{\alpha+2},\ldots,S_{\alpha+n+1-|I|}\in\mathcal{S}$, such that $S_1,\ldots,S_{\alpha+n+1-|I|}$ is a set cover. Define $\widehat{h}$ to output $S_1,\ldots,S_{\alpha+n+1-|I|}$.

To show the correctness of $\widehat{h}$, we first specify the condition for T under which $\widehat{h}$ satisfies the property stated in the claim. We say that $T\in\{0,1\}^m$ is *good* if the Kolmogorov complexity of T is at least $m-n$, i.e., $\mathsf{K}(T)\geq m-n$. By a simple counting argument, we have that a uniformly random T is good with probability at least $1-2^{-n}$.

We next show that if T is good, then for any circuit C of size at most $s+s/(a\cdot\log s)$ that computes F and the resulting set I obtained in the description of $\widehat{h}$ (note that I depends on C), it holds that $|I|\geq n-\delta t/2$. Observe that this implies that $\widehat{h}$ outputs a set cover of size

$$\begin{aligned}\alpha+n+1-|I| &= \lceil(1-\lambda)\cdot t\rceil+1+s/(a\cdot\log s)+\delta t/2\\ &\leq (1-\lambda)\cdot t+2+s/(a\cdot\log s)+\delta t/2.\end{aligned}$$

Since δ, ε and q are fixed constants, using Claim 30, we have

$$\begin{aligned}\frac{s}{a\cdot\log s} &= \frac{\gamma+\lceil(1-\lambda)\cdot t\rceil+1}{a\cdot\log s}\\ &\leq \frac{O(m+qt+n)}{a\cdot\log s}\\ &\leq \frac{O(a'\cdot n\log n)}{a\cdot\log s}\\ &\leq \frac{O(a'\cdot(qt)\log(qt))}{a\cdot\log t}\\ &< \delta t/2-1,\end{aligned}\tag{9}$$

where the second last inequality uses that $t\geq n/(2q)$, and the last inequality holds by letting a be a sufficiently large constant. Therefore, in this case we can obtain a set cover of size less than $(1-\lambda+\delta)\cdot t+1$, as desired.

We proceed to show that $|I|\geq n-\delta t/2$ if T is good. Fix a good T and (for the sake of contradiction) consider any circuit C and resulting set I with $|I|<n-\delta t/2$. Observe that by Equation (8), to describe T, it suffices to have a description for D and the truth tables of $G_1,\ldots,G_{\alpha+1}$.

First of all, the circuit D, which has α gates and $(\log(m)+\alpha+1)=O(n+\alpha)$ input bits, can be described using $O(\alpha\log(n+\alpha))$ bits. Using Equation (9), we have

$$\alpha=\lceil(1-\lambda)\cdot t\rceil+1+s/(a\cdot\log s)\leq O(t).\tag{10}$$

Therefore, D can be described using $O(t\log n)$ bits.

Next, we describe how to obtain the truth tables of $G_1,\ldots,G_{\alpha+1}$ using a short description. We need the following encodings.

- For each $i\in I\subseteq[n]$, we encode $(i,V_i)\in[n]\times\{0,1\}^{m/n}$, where V_i is the i-th segment of T.

  Using that $|I|<n-\delta t/2$, the number of bits that is required for all these encodings is at most

  $$\begin{aligned}(n-\delta t/2)\cdot(\log n+m/n) &\leq m-\frac{m}{n}\cdot\frac{\delta t}{2}+O(n\log n)\\ &\leq m-(a'/2)\cdot n\log n+O(n\log n),\end{aligned}$$

  where in the second inequality we use that $m\geq a'\cdot n\log n$, $t\geq n/(2q)$, and a' is a sufficiently large constant.

- We encode the subsets $S_1, \ldots, S_{\alpha+1} \in \mathcal{S}$ for which, G_j, where $j \in [\alpha+1]$, is a component that corresponds to a gate in $\mathcal{O}_{S_j}$, which also corresponds to some gate in $\mathbf{CKT}_{T_{\langle S_j \rangle}}$ (recall the definition of G in Equation (7)). Note that each $S_j \subseteq I$. Then given the encodings of (i, V_i) for all $i \in I$ (which allow us to recover the i-th segment of T for each $i \in I$), we can recover the truth table of $T_{\langle S_j \rangle}$.

  Note that all but at most one of the sets in $S_1, \ldots, S_{\alpha+1}$ contain at most q elements. Therefore, encoding the subsets $S_1, \ldots, S_{\alpha+1} \subseteq [n]$ can be done using at most $O(\alpha \cdot q \cdot \log n + n \cdot \log n) \leq O(n \log n)$ bits, using that $\alpha \leq O(t) \leq O(n)$.

- Let $u_1, \ldots, u_{\alpha+1} \in [O(m)]$ be such that G_j is the function computed by the u_j-th gate of $\mathbf{CKT}_{T_{\langle S_j \rangle}}$. We then encodes these gate numbers.

  Encoding the gate numbers $u_1, \ldots, u_{\alpha+1}$ requires at most $O(\alpha \cdot \log m) = O(t \log n)$ bits, using that $\alpha \leq O(t)$ and that $m = n^{O(1)}$.

With the above encodings, we can describe $G_1, \ldots, G_{\alpha+1}$ as follows. Using the encodings of (i, V_i) for all $i \in I$ and the encodings of the subsets $S_1, \ldots, S_{\alpha+1}$, we recover the truth table of $T_{\langle S_j \rangle}$ for each $j \in [\alpha+1]$. Given these, we construct $\mathbf{CKT}_{T_{\langle S_j \rangle}}$ for each $j \in [\alpha+1]$. Finally, using the encodings of the gate numbers $u_1, \ldots, u_{\alpha+1}$, we construct the truth table of G_j for each $j \in [\alpha+1]$.

Finally, note that the total number of bits needed for the above description (for both the circuit D and $G_1, \ldots, G_{\alpha+1}$) is

$$\begin{aligned} &\leq \big(m - (a'/2) \cdot n \log n) + O(n \log n)\big) + O(n \log n) + O(t \log n) \\ &\leq m - (a'/2) \cdot n \log n + O(n \log n). \end{aligned}$$

For a' is sufficiently large, the above implies that $\mathsf{K}(T) < m - n$, which gives a contradiction as desired. ◇

This completes the proof of Lemma 20. □

5 Proofs of Theorem 1 and Theorem 2

Here we complete the proofs of Theorem 1 and Theorem 2.

Proof of Theorem 1 and Theorem 2. Let $b > a$, where $a > 0$ is the constant in Theorem 18. We consider each implication below.
(2 $\implies$ 1). This follows directly from Proposition 12.
(1 $\implies$ 2). Assume $\mathsf{NP} \not\subseteq \mathsf{BPP}$ (resp. $\mathsf{NP} \not\subseteq \mathsf{ZPP}$). Towards a contradiction, suppose there exist a constant $c > 0$ and an imperfect (resp. perfect) indistinguishability obfuscator for multi-output circuits with output size $\sigma(\lambda, s) = s + s/(b \cdot \log s) + \lambda^c$. By Lemma 14, there exist indistinguishability obfuscators for general multi-output circuits with the same output size. Then by Theorem 15, there exist i.o.OWFs, and by Theorem 6 there exist i.o.UOWHFs. On the other hand, by combining Theorem 18 and Theorem 16 there is no i.o.UOWHF. This contradiction completes the proof. □

Remark 33 (Overhead on Circuit Size in Theorem 1 and Theorem 2)**.** *The proof described above allows for an additive overhead on circuit size of order $s/\log s$. Note that, in principle, Set-Cover might be hard to approximate under Levin reductions for any constant factor C when each set has size $O_C(1)$. Moreover, the proof of Theorem 16 does not impose a constraint on the circuit size overhead. The bottleneck lies in the proof of Lemma 20, i.e., on the gap between positive and negative instances of the* NP*-hardness of*

Multi-MCSP *under Levin reductions. The choice of parameters governing this gap are constrained by the encoding argument presented near the end of the proof, which does not seem to allow an additive gap of the form* $\Omega(s)$.

Also, we note that it is possible to slightly improve the additive overhead from $\Omega(s/\log s)$ *in Theorem 1 and Theorem 2 to* $\Omega(s \cdot \log\log s/\log s)$. *More specifically, we can first improve the upper bound for the quantity* γ *in Claim 30. To do this, we employ a technique in* [Juk12, *Claim 1.16*] *to obtain a circuit of size* $O(n)$, *rather than* $O(n \log n)$, *that simultaneously computes, for* all $i \in [n]$, *whether a given* $x \in \{0,1\}^{\log m}$ *belongs to* P_i. *This can be used to construct circuits for computing* $T_{\langle S_j \rangle}$ *for every* $j \in [m/n]$ (*recall the definition of a canonical circuit in Section 4.2.1*). *Then by modifying the construction of* $C_{(T,\mathcal{S})}$, *we can obtain a circuit that computes* $\{T_{\langle S_j \rangle}\}_{j \in [m/n]}$, *whose size can be upper bounded by* $O\left(\frac{n \log n}{\log\log n}\right)$ *instead of* $O(n \log n)$ *in Claim 30. This will allow us to get the claimed improvement.*

6 Alternative Proof via Pseudorandom Encryption Schemes

In this part we give a more direct proof to a weaker version of Theorems 1 and 2.
We prove the following results.

Theorem 34 (Hardness of Near-Optimal Imperfect i$\mathcal{O}$ for Multi-Output Circuits)**.** *The following are equivalent.*

1. NP $\not\subseteq$ BPP.
2. *There exists no imperfect indistinguishability obfuscator for multi-output circuits with output size* σ, *for any constant* $c > 0$ *and* $\sigma(\lambda, s) = s + o(s^{1/2}) + \lambda^c$.

Theorem 35 (Hardness of Near-Optimal Perfect i$\mathcal{O}$ for Multi-Output Circuits)**.** *The following are equivalent.*

1. NP $\not\subseteq$ ZPP.
2. *There exists no perfect indistinguishability obfuscator for multi-output circuits with output size* σ, *for any constant* $c > 0$ *and* $\sigma(\lambda, s) = s + o(s^{1/2}) + \lambda^c$.

To prove Theorem 34 and Theorem 35 we use an encryption scheme with pseudorandom ciphers, defined below. For simplicity we define here security with respect to a randomly chosen message m, which is weaker than the standard security definition but enough for our proof.

Definition 36 (Pseudorandom Encryption Schemes)**.** *A pair of efficient algorithms* $(\mathsf{Enc}, \mathsf{Dec})$ *is an i.o.-rate-1 pseudorandom encryption scheme if the following holds*:

- ***Rate-1:*** *For every* $\lambda \in \mathbb{N}$, $k \in \{0,1\}^\lambda$, *and* $m \in \{0,1\}^*$, $|(\mathsf{Enc}(m,k))| = |m|$.
- ***Correctness:*** *For every* $\lambda \in \mathbb{N}$, *and* $m \in \{0,1\}^*$ $\mathbf{Pr}_{k \leftarrow \{0,1\}^\lambda}[\mathsf{Dec}(\mathsf{Enc}(m,k),k) = m] = 1$.
- ***Security:*** *For every polynomial* p, *PPT* A *and constant* c, *the following holds for infinitely many* $\lambda \in \mathbb{N}$,

$$\left| \mathop{\mathbf{Pr}}_{A,\, k \leftarrow \{0,1\}^\lambda,\, m \leftarrow \{0,1\}^{p(\lambda)}} [A(m, \mathsf{Enc}(m,k)) = 1] - \mathop{\mathbf{Pr}}_{A,\, y \leftarrow \{0,1\}^{p(\lambda)},\, m \leftarrow \{0,1\}^{p(\lambda)}} [A(m,y) = 1] \right| \leq n^{-c}.$$

We say that $(\mathsf{Enc}, \mathsf{Dec})$ *has local decryption if there exists an algorithm* LD *of running time* $\mathsf{poly}(|k|, \log|m|)$ *such that for every* $k \in \{0,1\}^*$, $m \in \{0,1\}^*$, *and* $i \in [|m|]$, *we have* $\mathsf{LD}(k, |m|, i, \mathsf{Enc}(m,k)_i) = m_i$.

Such pseudorandom encryption schemes can be constructed from one-way function. We use the following result.

Proposition 37. *Assuming i.o.-one-way functions, there exists an i.o.-rate-1 pseudorandom encryption scheme with local decryption.*

Proof Sketch. By [HILL99, GGM86] the existence of i.o.-one-way functions implies i.o.-pseudorandom functions. Let $F = \{f_k\}_{k\in\{0,1\}^\lambda}$ be such a family of pseudorandom functions. We let $\mathsf{Enc}(m,k)_i = f_k(i) \oplus m_i$ and $\mathsf{LD}(k, |m|, i, b) = f_k(i) \oplus b$. It is not difficult to check that the required conditions hold. □

Our main result in this part is the following theorem, stating that if there exists a near-optimal obfuscator then there is no pseudorandom encryption scheme.

Theorem 38. *Assume the existence of an* (*imperfect*) *obfuscator for general multi-output circuits with output size* $\sigma(\lambda, s) \leq s + s^{1/2}/30 + \mathsf{poly}(\lambda)$. *Then there is no i.o.-pseudorandom encryption scheme.*

We prove Theorem 38 below, but first we use it to prove Theorem 34 and Theorem 35.

Proofs of Theorem 34 and Theorem 35. We consider each implication below.
(2 $\Longrightarrow$ 1). This follows directly from Proposition 12.
(1 $\Longrightarrow$ 2). Assume $\mathsf{NP} \not\subseteq \mathsf{BPP}$ (resp. $\mathsf{NP} \not\subseteq \mathsf{ZPP}$). Towards a contradiction, suppose there exist a constant $c > 0$ and an imperfect (resp. perfect) indistinguishability obfuscator for multi-output circuits with output size $\sigma(\lambda, s) = s + o(s^{1/2}) + \lambda^c$. By Lemma 14, there exist indistinguishability obfuscators for general multi-output circuit with the same output size. Then by Theorem 15, there exist i.o.OWFs, and by Proposition 37 there exists an i.o.-pseudorandom encryption scheme. On the other hand, by Theorem 38 there is no i.o.-pseudorandom encryption scheme. This contradiction completes the proof. □

6.1 Proof of Theorem 38

Let $\sigma(\lambda, s) \leq s + s^{1/2}/30 + \lambda^c$ for some constant c. Let $(\mathsf{Enc}, \mathsf{Dec})$ be a rate-1 pseudorandom encryption scheme with local decryption algorithm LD that can be implemented as a circuit of size $(\lambda + \log|m|)^{\alpha/2}$ for some constant α, and assume without loss of generality that $\alpha/2 > c$. Let $\varepsilon = 1/4$, and let $p(\lambda) = \lambda^{\alpha/\varepsilon}$. Let $n = n_\lambda$ be the first power of two larger than $p(\lambda)$, and let $t(\lambda) = n/(10\log n)$. In the following we construct an ensemble of distributions $\mathcal{P}_\lambda$ over pairs (C_1, C_2) of multi-output circuits and an advice string $a \in \{0,1\}^*$ of bounded length, together with an efficient deterministic algorithm A, such that $\mathbf{Pr}_{(C_1,C_2,a)\leftarrow\mathcal{P}_\lambda}[C_1 \equiv C_2] = 1$ but

$$\left| \mathop{\mathbf{Pr}}_{(C_1,C_2,a)\leftarrow\mathcal{P}_\lambda, \mathsf{i}\mathcal{O}}[A(1^\lambda, a, \mathsf{i}\mathcal{O}(1^\lambda, C_1)) = 1] - \mathop{\mathbf{Pr}}_{(C_1,C_2,a)\leftarrow\mathcal{P}_\lambda, \mathsf{i}\mathcal{O}}[A(1^\lambda, a, \mathsf{i}\mathcal{O}(1^\lambda, C_2)) = 1] \right| \geq 1/3$$

for any obfuscator $\mathsf{i}\mathcal{O}$ with output size σ and for infinitely many $\lambda \in \mathbb{N}$. By a standard averaging argument that non-uniformly fixes the required advice bits, this contradicts the security of the assumed $\mathsf{i}\mathcal{O}$ scheme.

The distribution $\mathcal{P}_\lambda$. We describe below how to sample from $\mathcal{P}_\lambda$. In the following we fix λ and omit it from the notation. We also need the following definition of a canonical circuit. For a string $T \in \{0,1\}^n$, viewed as the truth table of a Boolean function mapping $\log n$ bits to 1 bit, let $\mathbf{CKT}_T$ denote the circuit computing T that is constructed using Proposition 5.

1. We define a sequence $(T_1, \ldots, T_t)$, where each $T_i \in \{0,1\}^n$, as follows. Sample $r \leftarrow \{0,1\}^n$, $k_1, k_2 \leftarrow \{0,1\}^\lambda$, and $j_1 \neq j_2 \leftarrow [t]$. For every $i \in [t] \setminus \{j_1, j_2\}$, let $T_i \leftarrow \{0,1\}^n$ be a random function. Let $T_{j_1} = \mathsf{Enc}(r, k_1)$ and $T_{j_2} = \mathsf{Enc}(r, k_2)$.

2. Next, we define a multi-output function G. Let $C_{(T_1,\ldots,T_t)}$ be the circuit built as follows.

   We start by letting $C_{(T_1,\ldots,T_t)}$ be the empty circuit. For each $i \in [t]$, we first construct $\mathbf{CKT}_{T_i}$. We then iterate through the gates u in $\mathbf{CKT}_{T_i}$ in *topological* order. Let $\diamond \in \{\wedge, \vee, \neg\}$ be the gate type of u. If u computes a function that is already computed by $C_{(T_1,\ldots,T_t)}$, then ignore it. Otherwise, add a $\diamond$ gate to $C_{(T_1,\ldots,T_t)}$ that takes as input(s) those gate(s) in $C_{(T_1,\ldots,T_t)}$ that compute the function(s) which are fed as inputs to u in $\mathbf{CKT}_{T_i}$. (Note that we are guaranteed to find such gates in $C_{(T_1,\ldots,T_t)}$ since we are iterating in topological order).

   We denote by $\mathcal{O}_i$ the (ordered) set of newly added gates during the iteration that corresponds to T_i.

   We then define

   $$G(x) := x_1 \bullet \cdots \bullet x_{\log n} \bigbullet_{u \in \mathcal{O}_1} u(x) \ldots \bigbullet_{u \in \mathcal{O}_t} u(x). \tag{11}$$

   That is, on input $x \in \{0,1\}^{\log n}$, G outputs x together with the value of each gate in $C_{(T_1,\ldots,T_t)}$ when evaluating on x.

3. Let γ be the number of components of G that are not functions computed by an input wire. That is,

   $$\gamma := |\{G_i : G_i \text{ is a component of } G \text{ and } G_i \neq x_j \text{ for all } j \in [\log n]\}|.$$

   Note that by construction γ is the number of gates in the circuit $C_{(T_1,\ldots,T_t)}$ that defines G. Moreover, by Proposition 5 it holds that

   $$\gamma \leq t \cdot 5n/\log n. \tag{12}$$

4. Finally, let $F \colon \{0,1\}^{\log n} \to \{0,1\}^{1+\log n+\gamma}$ be the function defined as

   $$F(x) := r(x) \bullet G(x),$$

   where we think on r as a truth table of a function $r \colon \{0,1\}^{\log n} \to \{0,1\}$.

5. We construct two circuits C_1, C_2 of size $\gamma + n^\varepsilon$ that compute the function F:

   - C_1 is the circuit composed by the circuit computing G, together with the local decryption circuit $\mathsf{LD}(k_1, n, x, T_{j_1}(x))$, with k_1 hardcoded to it, where $T_{j_1}(x)$ is computed by $G(x)$.
   - C_2 is defined similarly with respect to k_2 and T_{j_2}.

   Note that the size of each circuit is indeed bounded by $\gamma + n^\varepsilon$ due to our choice of parameters.

6. Let $a = (\vec{T}, j_1)$, where we use $\vec{T}$ to denote $(T_1, \ldots, T_t)$.

The following claim follows directly from the construction.

Claim 39. *For every $\lambda \in \mathbb{N}$,*

$$\Pr_{(C_1,C_2,a)\leftarrow\mathcal{P}_\lambda}[C_1 \equiv C_2] = 1.$$

Proof. The claim follows by the correctness of the encryption scheme, and since $T_{j_1} = \mathsf{Enc}(r, k_1)$ and $T_{j_2} = \mathsf{Enc}(r, k_2)$. □

Before describing the distinguisher, we prove the following claim on the size of the output of $\mathsf{i}\mathcal{O}$.

Claim 40. *We have*

$$\mathsf{size}(\mathsf{i}\mathcal{O}(1^\lambda, C_1)) \leq \gamma + t/3 \quad \textit{and} \quad \mathsf{size}(\mathsf{i}\mathcal{O}(1^\lambda, C_2)) \leq \gamma + t/3.$$

Proof. We prove the claim for C_1. The proof for C_2 is symmetric. First notice that by construction and Equation (12)

$$\mathsf{size}(C_1) \leq \gamma + n^\varepsilon \leq t \cdot 5n/\log n + n^\varepsilon \leq 6tn/\log n.$$

Thus, by our assumption on the overhead of the obfuscator,

$$\mathsf{size}(\mathsf{i}\mathcal{O}(1^\lambda, C_1)) \leq \gamma + n^\varepsilon + o((tn/\log n)^{1/2}).$$

Let $\delta \in o(1)$ be a monotony decreasing function such that $\delta(n) \geq 1/\log n$ and

$$\mathsf{size}(\mathsf{i}\mathcal{O}(1^\lambda, C_1)) \leq \gamma + n^\varepsilon + \frac{1}{30} \cdot \sqrt{\frac{6tn}{\log n}} \leq \gamma + \frac{n}{30 \log n} = \gamma + t/3 \tag{13}$$

for large enough n. □

The distinguisher. Let $(C_1, C_2, a) \leftarrow \mathcal{P}_\lambda$. We next define a distinguisher A that takes as input the security parameter 1^λ, a circuit $C\colon \{0,1\}^{\log n} \to \{0,1\}^{1+\log n+\gamma}$ of size at most $\gamma + t/3$ that computes F, and the advice a. Its goal is to distinguish $C \leftarrow \mathsf{i}\mathcal{O}(1^\lambda, C_1)$ from $C \leftarrow \mathsf{i}\mathcal{O}(1^\lambda, C_2)$.

Before defining A, we first argue that γ of the gates in C must compute the components of G. Suppose we have a circuit that computes G. Then every distinct component of G has a (necessarily distinct) input wire or gate from C that computes that component. Therefore, since G has γ distinct components that are not computed by an input wire, C must have at least γ distinct gates computing components of G.

It follows that there is a circuit D that takes $(\log n + \gamma)$ input bits and has at most $t/3$ gates such that

$$D(x, G_1(x), \ldots, G_\gamma(x)) = r(x)$$

for all $x \in \{0,1\}^{\log n}$, where $G_1, \ldots, G_\gamma$ are the unique components of G. Moreover, since D has only $t/3$ gates of fan-in 2, it uses at most $t/3 + 1$ of the components of G in the circuit. Thus, after a possible relabeling of $G_1, \ldots, G_\gamma$, we can assume D takes at most $(\log n + t/3 + 1)$ input bits and that

$$D(x, G_1(x), \ldots, G_{t/3+1}(x)) = r(x). \tag{14}$$

For each $j \in [t/3 + 1]$, let $i_j \in [t]$ be the first index such that G_j is a component that corresponds to a gate in $\mathcal{O}_{i_j}$ (recall the definition of G in Equation (7)). Note that these values can be efficiently computed given the input of A. The distinguisher A outputs 1 if j_1 is among those indices, and 0 otherwise.

We prove the following lemma.

Lemma 41. *Assume that $\mathsf{i}\mathcal{O}$ is an obfuscator for general multi-output circuits with output size $\sigma(\lambda, s)$. Then for infinitely many $\lambda \in \mathbb{N}$,*

$$\left| \Pr_{(C_1,C_2,a)\leftarrow\mathcal{P}_\lambda,\, \mathsf{i}\mathcal{O}}[A(1^\lambda, a, \mathsf{i}\mathcal{O}(1^\lambda, C_1)) = 1] - \Pr_{(C_1,C_2,a)\leftarrow\mathcal{P}_\lambda,\, \mathsf{i}\mathcal{O}}[A(1^\lambda, a, \mathsf{i}\mathcal{O}(1^\lambda, C_2)) = 1] \right| \geq 1/3.$$

We prove Lemma 41 below, but first let us conclude the proof of the main theorem.

Proof of Theorem 38. The proof follows by combining Claim 39 and Lemma 41, which together contradict the security of $\mathsf{i}\mathcal{O}$. □

To prove Lemma 41, we define two new distributions $\widehat{\mathcal{P}}_{\lambda,1}$ and $\widehat{\mathcal{P}}_{\lambda,2}$:

- $\widehat{\mathcal{P}}_{\lambda,1}$ outputs a pair $(\widehat{C}_1, a)$ and is defined similarly to $\mathcal{P}_\lambda$, where when constructing $\widehat{C}_1$ we follow the construction of C_1 in $\mathcal{P}_\lambda$, but changing the first step as follows: replacing T_{j_2} with a random function (instead of $\mathsf{Enc}(r, k_2)$). The advice string a is set to $(\vec{T}, j_1)$, where the j_2-entry of $\vec{T}$ now contains this random function.
- $\widehat{\mathcal{P}}_{\lambda,2}$ outputs a pair $(\widehat{C}_2, a)$ and is also defined similarly to $\mathcal{P}_\lambda$, but when constructing $\widehat{C}_2$ we follow the construction of C_2 but replacing T_{j_1} with a random function. The rest of the construction is left the same, and the advice string a is set to $(\vec{T}, j_1)$, where the j_1-entry of $\vec{T}$ now contains this random function.

We remark that in $\widehat{\mathcal{P}}_{\lambda,2}$ the distribution of j_1 is independent from the circuit $\widehat{C}_2$ and can be chosen (uniformly from $[n] \setminus \{j_2\}$) *after* $\widehat{C}_2$ is constructed. Additionally, note that C_1 and $\widehat{C}_1$ (similarly, C_2 and $\widehat{C}_2$) are not functionally equivalent, and might not produce the same number of output bits. Yet, we prove the following claim for the algorithm A fixed above.

Claim 42. *Assume that $(\mathsf{Enc}, \mathsf{Dec})$ is an i.o.-pseudorandom encryption scheme. Then for every general multi-output obfuscator $\mathsf{i}\mathcal{O}$ and for every constant c, the following hold for infinitely many $\lambda \in \mathbb{N}$,*

$$\left| \Pr_{(C_1,C_2,a)\leftarrow\mathcal{P}_\lambda,\, \mathsf{i}\mathcal{O}}[A(1^\lambda, a, \mathsf{i}\mathcal{O}(1^\lambda, C_1)) = 1] - \Pr_{(\widehat{C}_1,a)\leftarrow\widehat{\mathcal{P}}_{\lambda,1},\, \mathsf{i}\mathcal{O}}[A(1^\lambda, a, \mathsf{i}\mathcal{O}(1^\lambda, \widehat{C}_1)) = 1] \right| \leq \lambda^{-c},$$

and,

$$\left| \Pr_{(C_1,C_2,a)\leftarrow\mathcal{P}_\lambda,\, \mathsf{i}\mathcal{O}}[A(1^\lambda, a, \mathsf{i}\mathcal{O}(1^\lambda, C_2)) = 1] - \Pr_{(\widehat{C}_2,a)\leftarrow\widehat{\mathcal{P}}_{\lambda,2},\, \mathsf{i}\mathcal{O}}[A(1^\lambda, a, \mathsf{i}\mathcal{O}(1^\lambda, \widehat{C}_2)) = 1] \right| \leq \lambda^{-c}.$$

Proof. Fix a general multi-output obfuscator $\mathsf{i}\mathcal{O}$, and assume towards a contradiction that the claim does not hold. That is, there exists a polynomial g such that for every large enough $\lambda \in \mathbb{N}$, the following holds for at least one value of $b \in \{1, 2\}$.

$$\left| \Pr_{(C_1,C_2,a)\leftarrow\mathcal{P}_\lambda,\, \mathsf{i}\mathcal{O}}[A(1^\lambda, a, \mathsf{i}\mathcal{O}(1^\lambda, C_b)) = 1] - \Pr_{(\widehat{C}_b,a)\leftarrow\widehat{\mathcal{P}}_{\lambda,b},\, \mathsf{i}\mathcal{O}}[A(1^\lambda, a, \mathsf{i}\mathcal{O}(1^\lambda, \widehat{C}_b)) = 1] \right| \geq 1/g(\lambda). \quad (15)$$

For every large enough $\lambda \in \mathbb{N}$, let b_λ be a value of $b \in \{1, 2\}$ such that

$$\left| \Pr_{(C_1,C_2,a)\leftarrow\mathcal{P}_\lambda,\, \mathsf{i}\mathcal{O}}[A(1^\lambda, a, \mathsf{i}\mathcal{O}(1^\lambda, C_b)) = 1] - \Pr_{(\widehat{C}_b,a)\leftarrow\widehat{\mathcal{P}}_{\lambda,b},\, \mathsf{i}\mathcal{O}}[A(1^\lambda, a, \mathsf{i}\mathcal{O}(1^\lambda, \widehat{C}_b)) = 1] \right| \geq 1/2g(\lambda). \quad (16)$$

(Fix $b_\lambda = 1$ if the above holds for both $b = 1$ and $b = 2$.) Note that by sampling a polynomial number of samples from the distributions $\mathcal{P}_\lambda$,$\widehat{\mathcal{P}}_{\lambda,1}$, and $\widehat{\mathcal{P}}_{\lambda,2}$, we can find a value of b_λ as above with probability at least $1 - 1/6g(\lambda)$. Consider the following algorithm that breaks the security of the encryption scheme on every large enough security parameter λ.

Input: 1^λ, $m \in \{0,1\}^n$, $w \in \{0,1\}^n$;
// The goal is to distinguish between $w = \mathsf{Enc}(m, K)$ and $w \leftarrow \{0,1\}^n$.
1 Estimate b_λ. Let $b \in \{1,2\}$ be the estimation;
2 Sample $j_1 \neq j_2 \leftarrow [t]$, $k_b \leftarrow \{0,1\}^\lambda$, $T_1, \ldots, T_t \leftarrow \{0,1\}^n$;
3 Set $T_{j_b} = \mathsf{Enc}(m, k_b)$ and $T_{j_{\mathsf{flip}(b)}} = w$, where $\mathsf{flip}(b) \in \{1,2\}$ gets the opposite value of $b \in \{1,2\}$;
4 Follow Steps 2-5 in the description of $\mathcal{P}_\lambda$ to construct the circuit C_b using strings $T_1, \ldots, T_t$, index j_b, and the string m for the role of r. Let C be the resulting circuit;
5 **return** $A(1^\lambda, a, \mathsf{i}\mathcal{O}(1^\lambda, C))$, *where* $a = (\vec{T}, j_1)$;

Algorithm 2: Description of the reduction.

The proof now follows by the simple observation that C is distributed as C_b when $w = \mathsf{Enc}(m, K)$, and distributed as $\widehat{C}_b$ when w is a random string. Thus, when $b = b_\lambda$, Algorithm 2 distinguishes between a random string and an encryption of m with advantage $1/2g(\lambda)$. Since Algorithm 2 guesses the correct value of b_λ with probability at least $1 - 1/6g(\lambda)$, we get that Algorithm 2 distinguishes between a random string and an encryption of m with advantage $1/6g(\lambda)$. □

Given the above claim, it is enough to show that A distinguishes between $\mathsf{i}\mathcal{O}(\widehat{C}_1)$ and $\mathsf{i}\mathcal{O}(\widehat{C}_2)$. Towards this, for an advice string $a = (\vec{T}, j_1)$, let $\gamma(a)$ be the value computed in the definition of $\mathcal{P}_\lambda$, when using $\vec{T}$ as the values of $T_1, \ldots, T_t$ in the construction of G. It follows that for $(C_1, C_2, a) \leftarrow \mathcal{P}_\lambda$, $\gamma(a)$ is the value of γ used in the description of the distribution $\mathcal{P}_\lambda$. Similarly to Claim 40, it is not hard to see that the following claim holds.

Claim 43. *We have*

$$\Pr_{(\widehat{C}_1,a)\leftarrow\widehat{\mathcal{P}}_{\lambda,1},\mathsf{i}\mathcal{O}}[\mathsf{size}(\mathsf{i}\mathcal{O}(1^\lambda, \widehat{C}_1)) \leq \gamma(a) + t/3] = 1$$

and

$$\Pr_{(\widehat{C}_2,a)\leftarrow\widehat{\mathcal{P}}_{\lambda,2},\mathsf{i}\mathcal{O}}[\mathsf{size}(\mathsf{i}\mathcal{O}(1^\lambda, \widehat{C}_2)) \leq \gamma(a) + t/3] = 1.$$

The next claim states that A outputs 1 with high probability on $\mathsf{i}\mathcal{O}(\widehat{C}_1)$.

Claim 44. *For every* $\lambda \in \mathbb{N}$,

$$\Pr_{(\widehat{C}_1,a)\leftarrow\widehat{\mathcal{P}}_{\lambda,1},\,\mathsf{i}\mathcal{O}}[A(1^\lambda, a, \mathsf{i}\mathcal{O}(1^\lambda, \widehat{C}_1)) = 1] \geq 1 - \mathsf{negl}(n).$$

Proof. We first describe the randomness used by the construction of $\widehat{C}_1$. The construction samples $j_1 \in [t]$ and $M \in \{0,1\}^{n\times t}$, where $M_1, \ldots, M_{t-1} \in \{0,1\}^n$ are used to determine $T_1, \ldots, T_{j_1-1}, T_{j_1+1}, \ldots, T_n$, and M_t is used to determine r. Then the construction samples k_1 and let $T_{j_1} = \mathsf{Enc}(r, k_1)$.

In the following, fix the value of $j_1 \in [t]$ and $M \in \{0,1\}^{n\times t}$, and assume that for some k_1, and $T_{j_1} = \mathsf{Enc}(r, k_1)$, it holds that with some positive probability over the randomness of $\mathsf{i}\mathcal{O}$, $A(1^\lambda, a, \mathsf{i}\mathcal{O}(1^\lambda, \widehat{C}_1))$

outputs 0 and $\mathsf{i}\mathcal{O}(1^\lambda, \widehat{C}_1)$ is functionally equivalent to $\widehat{C}_1$. Then we claim that (j_1, M) can be compressed to length $nt - n/100$, which implies that the claim holds.

By the assumption above and Claim 43, there exists a circuit C of size at most $\gamma(a) + t/3$ which is equivalent to $\widehat{C}_1$, and such that $A(1^\lambda, a, C) = 0$. We next describe how to use such C to compress (j_1, M). By the definition of the distinguisher A, there is a circuit D of size $t/3$ such that

$$D(x, G_1(x), \ldots, G_{t/3+1}(x)) = r(x),$$

where $G_1(x), \ldots, G_{t/3+1}(x)$ are components computed by the circuits

$$\mathbf{CKT}_{T_1}, \ldots, \mathbf{CKT}_{T_{j_1-1}}, \mathbf{CKT}_{T_{j_1+1}}, \ldots, \mathbf{CKT}_{T_n}$$

(and importantly can be computed without knowing the value of T_{j_1}). The compressed description is as follows:

- Describe $j_1, M_1, \ldots, M_{t-1}$ (using $\lceil \log t \rceil + n(t-1)$ bits). These values determine the values of $T_1, \ldots, T_{j_1-1}, T_{j_1+1}, \ldots, T_n$ and thus the values of $\mathbf{CKT}_{T_1}, \ldots, \mathbf{CKT}_{T_{j_1-1}}, \mathbf{CKT}_{T_{j_1+1}}, \ldots, \mathbf{CKT}_{T_n}$.
- Describe the circuit D and the components $G_1(x), \ldots, G_{t/3+1}(x)$. Each component can be described using $\log(O(t \cdot n \log n)) = \log t + \log n + \log\log n + O(1)$ bits, and the circuit D can be described using $t \log t$ bits (here we use the fact that a circuit of size s can be described using $3s \log s$ bits).

The correctness follows since the output of D is equal to r and thus to M_t. The overall compression size is of length

$$\lceil \log t \rceil + n(t-1) + (t/3+1) \cdot (\log t + \log n + \log\log n + O(1)) + t \log t \;\leq\; nt - n/3, \tag{17}$$

as long as $t \leq n/(10 \log n)$. □

Finally, it is not hard to see that A outputs 1 with low probability given $\mathsf{i}\mathcal{O}(\widehat{C}_2)$.

Claim 45. *For every $\lambda \in \mathbb{N}$,*

$$\mathop{\mathbf{Pr}}_{(\widehat{C}_2, a) \leftarrow \widehat{\mathcal{P}}_{\lambda,2},\, \mathsf{i}\mathcal{O}} [A(1^\lambda, a, \mathsf{i}\mathcal{O}(1^\lambda, \widehat{C}_2)) = 1] \leq 1/2.$$

Proof. The proof follows since the corresponding circuit D described in the specification of A touches at most $t/3 + 1$ components of G and by symmetry. □

Proof of Lemma 41. We are now ready to prove Lemma 41.

Proof of Lemma 41. The proof follows by Claim 42, Claim 44, Claim 45, and the triangular inequality. □

Remark 46 (Overhead on Circuit Size in Theorem 38)**.** *The proof described above allows for an additive overhead on circuit size of order $o(s^{1/2})$. The choice of parameters leading to this overhead are constrained by Equation (13) and Equation (17). While the size overhead can be made larger by picking $t \gg n$ in Equation (13), we must use $t \ll n$ in Equation (17). This leads to a choice of $t = \widetilde{\Theta}(n)$.*

7 Lower Bounds on Obfuscation for Other Computational Models

7.1 Obfuscation of Circuits with Database Access

In this part we prove a lower bound on the size overhead for circuits with database access. While we assume database access, the circuits considered in this section produce a single output bit.

Goldwasser and Rothblum [GR07] showed that obfuscation is impossible when the program has (oracle) access to a random database. Their proof relies on the fact that the database is too long for the obfuscator to read. We show here that obfuscation is impossible even if the obfuscator can run in polynomial time in the size of the database, as long as the size overhead is polynomial only in the security parameter and the program size.

We start with the definition of obfuscation for single-output circuits with database access. In the following, given a string $z \in \{0,1\}^*$, a circuit $C\colon \{0,1\}^\ell \to \{0,1\}$ with oracle access to a database z, denoted by C^z, is a circuit with oracle gates to the function for which z is its truth table. The size of C is defined as its total number of AND, OR, NOT, and ORACLE gates. We note that this definition is very similar to that used by [GR07]; the key (and important) difference is that we allow the obfuscator to run in time polynomial in the length of the database/oracle z (whereas in their notion, the running time was independent of z, and thus z could be exponentially long).

Definition 47 (Indistinguishability Obfuscation for Database-Aided Circuits)**.** *A probabilistic polynomial-time algorithm* $\mathsf{i}\mathcal{O}$ *is an* indistinguishability obfuscator for database-aided circuits with output size $\sigma\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ *if the following hold.*

- ***Perfect/Imperfect Functionality:*** *There exists a negligible function α such that for all $\lambda, \ell \in \mathbb{N}$, any $z \in \{0,1\}^*$ and any database-aided circuit $C\colon \{0,1\}^\ell \to \{0,1\}$,*

$$\Pr_{\mathsf{i}\mathcal{O}}\left[\widehat{C}^z \equiv C^z \colon \widehat{C} \leftarrow \mathsf{i}\mathcal{O}(1^\lambda, C, z)\right] \geq 1 - \alpha(\lambda).$$

  We say that $\mathsf{i}\mathcal{O}$ is perfect if $\alpha(\cdot) = 0$; otherwise it is imperfect.

- ***Indistinguishability:*** *For any polynomial-size circuit family $\{A_\lambda\}_\lambda$ and polynomial p, there exists a negligible function μ such that for all $\lambda, \ell \in \mathbb{N}$ and $z \in \{0,1\}^*$ with $\ell, |z| \leq p(\lambda)$ and any pair of circuits $C, C'\colon \{0,1\}^\ell \to \{0,1\}$ satisfying $C^z \equiv C'^z$ and $\mathsf{size}(C^z) = \mathsf{size}(C'^z) \leq p(\lambda)$, it holds that*

$$\left|\Pr_{\mathsf{i}\mathcal{O}}[A_\lambda(\mathsf{i}\mathcal{O}(1^\lambda, C, z)) = 1] - \Pr_{\mathsf{i}\mathcal{O}}[A_\lambda(\mathsf{i}\mathcal{O}(1^\lambda, C', z)) = 1]\right| \leq \mu(\lambda).$$

- ***σ-Output-Size:*** *For all $\lambda, \ell \in \mathbb{N}$, $z \in \{0,1\}^*$ and every circuit $C\colon \{0,1\}^\ell \to \{0,1\}$,*

$$\Pr_{\mathsf{i}\mathcal{O}}\left[\mathsf{size}(\mathsf{i}\mathcal{O}(1^\lambda, C, z)) \leq \sigma(\lambda, \mathsf{size}(C))\right] = 1.$$

We prove the following theorem.

Theorem 48. *Assume the existence of an (imperfect) obfuscator for database-aided circuits with output size $\sigma(\lambda, s) \in \mathsf{poly}$. Then there is no i.o.-Pseudorandom encryption.*

Since obfuscation for database-aided circuits is stronger than standard obfuscation, its existence together with $\mathsf{NP} \not\subseteq \mathsf{ZPP}$ implies the existence of one-way functions. On the other hand, by a similar argument to the proof of Proposition 12, if $\mathsf{NP} \subseteq \mathsf{ZPP}$ (resp. $\mathsf{NP} \subseteq \mathsf{BPP}$), there exist perfect (resp. imperfect) obfuscations for database-aided circuits. We get the following theorems by an argument similar to the proofs of Theorem 34 and Theorem 35.

Theorem 49. *The following are equivalent.*

1. $\mathsf{NP} \not\subseteq \mathsf{ZPP}$.
2. *There exists no perfect indistinguishability obfuscator for database-aided circuits with output size σ, for any $\sigma \in \mathsf{poly}$.*

Theorem 50. *The following are equivalent.*

1. $\mathsf{NP} \not\subseteq \mathsf{BPP}$.
2. *There exists no imperfect indistinguishability obfuscator for database-aided circuits with output size σ, for any $\sigma \in \mathsf{poly}$.*

The proof of Theorem 48 follows the same lines of the proof of Theorem 38. Assume towards a contradiction that there exists an indistinguishability obfuscator $\mathsf{i}\mathcal{O}$ for database-aided circuits with output size $\sigma \in \mathsf{poly}$, and let $(\mathsf{Enc}, \mathsf{Dec})$ be a rate-1 pseudorandom encryption scheme with local decryption algorithm LD that can be implemented as a circuit of size $(\lambda + \log|m|)^\alpha$ for some constant $\alpha > 1$. Let $p(\lambda) = (\sigma(\lambda, 10\lambda^\alpha))^2$ be a polynomial, and let $n = n_\lambda = p(\lambda)$, and $t(\lambda) = n$. We construct a distribution $\mathcal{P}_\lambda$ over the database z, a pair (C_1, C_2) of database-aided circuits, and advice string a, such that C_1^z computes the same function as C_2^z but $\mathsf{i}\mathcal{O}(C_1, z)$ is distinguishable from $\mathsf{i}\mathcal{O}(C_1, z)$.

The distribution $\mathcal{P}_\lambda$. We describe below how to sample from $\mathcal{P}_\lambda$. In the following we fix λ and omit it from the notation.

1. Sample $r \leftarrow \{0,1\}^n$, $k_1, k_2 \leftarrow \{0,1\}^\lambda$ and $j_1 \neq j_2 \leftarrow [t]$. For every $i \in [t] \setminus \{j_1, j_2\}$, let $T_i \leftarrow \{0,1\}^n$ be a random function. Let $T_{j_1} = \mathsf{Enc}(r, k_1)$ and $T_{j_2} = \mathsf{Enc}(r, k_2)$. Let $z = T_1 || \ldots || T_t$. We construct two circuits C_1, C_2 of size at most $10\lambda^\alpha$ such that each of C_1^z, C_2^z computes the function r:

   C_1 is the circuit with the values for k_1 and j_1 hardcoded to it, that given an input x (seen as a number in $[n]$) computes $\mathsf{LD}(k_1, n, x, T_{j_1}(x))$, where $T_{j_1}(x) = z(n \cdot (j_1 - 1) + x)$.

   C_2 is defined similarly with respect to k_2 and T_{j_2}.

2. Let $a = (j_1, z)$.

The following claim follows directly by the construction.

Claim 51. *For every $\lambda \in \mathbb{N}$,*

$$\Pr_{(z, C_1, C_2, a) \leftarrow \mathcal{P}_\lambda}[C_1^z \equiv C_2^z] = 1.$$

Proof. The claim follows by the correctness of the encryption scheme, and since $T_{j_1} = \mathsf{Enc}(r, k_1)$ and $T_{j_2} = \mathsf{Enc}(r, k_2)$. □

The distinguisher. We define a (randomized) distinguisher A that takes as input the security parameter 1^λ, a circuit C that computes r, and the advice a: The distinguisher A executes C on 100 random inputs. If C queries z on the j_1-th block in at least one of these inputs, A outputs 1. Otherwise, A outputs 0.

We prove the following lemma.

Lemma 52. *Assume that $\mathsf{i}\mathcal{O}$ is an obfuscator for database-aided circuits with overhead $\sigma \in \mathsf{poly}$. Then for infinitely many $\lambda \in \mathbb{N}$,*

$$\left| \Pr_{(z,C_1,C_2,a)\leftarrow\mathcal{P}_\lambda,\, \mathsf{i}\mathcal{O}}[A(1^\lambda, a, \mathsf{i}\mathcal{O}(1^\lambda, C_1, z)) = 1] - \Pr_{(z,C_1,C_2,a)\leftarrow\mathcal{P}_\lambda,\, \mathsf{i}\mathcal{O}}[A(1^\lambda, a, \mathsf{i}\mathcal{O}(1^\lambda, C_2, z)) = 1] \right| \geq 1/3.$$

We prove Lemma 52 below, but first let us conclude the proof of the main theorem.

Proof of Theorem 48. The proof follows by combining Claim 51 and Lemma 52, which together contradict the security of $\mathsf{i}\mathcal{O}$. □

To prove Lemma 52, we define two new distributions $\widehat{\mathcal{P}}_{\lambda,1}$ and $\widehat{\mathcal{P}}_{\lambda,2}$:

- $\widehat{\mathcal{P}}_{\lambda,1}$ outputs a triplet $(\widehat{z}_1, C_1, a)$ and is defined similarly to $\mathcal{P}_\lambda$, where when constructing $\widehat{z}_1$ we follow the construction of z in $\mathcal{P}_\lambda$, but replacing T_{j_2} with a random function (instead of $\mathsf{Enc}(r, k_2)$). The advice string a is set to $(j_1, \widehat{z}_1)$, where the j_2-block of $\widehat{z}_1$ now contains this random function. The circuit C_1 is defined in the same way as in $\mathcal{P}_\lambda$.
- $\widehat{\mathcal{P}}_{\lambda,2}$ outputs a triplet $(\widehat{z}_2, C_2, a)$ and is also defined similarly to $\mathcal{P}_\lambda$, but when constructing $\widehat{z}_2$ we follow the construction of z but replacing T_{j_1} with a random function. The rest of the construction is left the same, and the advice string a is set to $(j_1, \widehat{z}_2)$, where the j_1-block of $\widehat{z}_2$ now contains this random function. The circuit C_2 is defined in the same way as in $\mathcal{P}_\lambda$.

Importantly, due to our choice of parameters, it holds that

$$\mathsf{size}(\mathsf{i}\mathcal{O}(1^\lambda, C_1, \widehat{z}_1)) \leq \sqrt{n} \quad \text{and} \quad \mathsf{size}(\mathsf{i}\mathcal{O}(1^\lambda, C_2, \widehat{z}_2)) \leq \sqrt{n}.$$

The proof of the next claim is identical to the proof of Claim 42.

Claim 53. *Assume that $(\mathsf{Enc}, \mathsf{Dec})$ is an i.o.-pseudorandom encryption scheme. Then for every obfuscator $\mathsf{i}\mathcal{O}$ for database-aided circuits, and for every constant c, the following hold for infinitely many $\lambda \in \mathbb{N}$,*

$$\left| \Pr_{(z,C_1,C_2,a)\leftarrow\mathcal{P}_\lambda,\, \mathsf{i}\mathcal{O}}[A(1^\lambda, a, \mathsf{i}\mathcal{O}(1^\lambda, C_1, z)) = 1] - \Pr_{(\widehat{z}_1,C_1,a)\leftarrow\widehat{\mathcal{P}}_{\lambda,1},\, \mathsf{i}\mathcal{O}}[A(1^\lambda, a, \mathsf{i}\mathcal{O}(1^\lambda, C_1, \widehat{z}_1)) = 1] \right| \leq \lambda^{-c},$$

and,

$$\left| \Pr_{(z,C_1,C_2,a)\leftarrow\mathcal{P}_\lambda,\, \mathsf{i}\mathcal{O}}[A(1^\lambda, a, \mathsf{i}\mathcal{O}(1^\lambda, C_2, z)) = 1] - \Pr_{(\widehat{z}_2,C_2,a)\leftarrow\widehat{\mathcal{P}}_{\lambda,2},\, \mathsf{i}\mathcal{O}}[A(1^\lambda, a, \mathsf{i}\mathcal{O}(1^\lambda, C_2, \widehat{z}_2)) = 1] \right| \leq \lambda^{-c}.$$

We prove the following claim.

Claim 54. *For every $\lambda \in \mathbb{N}$,*

$$\Pr_{(\widehat{z}_1,C_1,a)\leftarrow\widehat{\mathcal{P}}_\lambda,\, \mathsf{i}\mathcal{O}}[A(1^\lambda, a, \mathsf{i}\mathcal{O}(1^\lambda, C_1, \widehat{z}_1)) = 1] \geq 0.99.$$

Proof. Fix the (partial) randomness used by the construction of $\widehat{z}_1$. That is, fix $j_1 \in [t]$ and $M \in \{0,1\}^{n \times t}$, where $M_1, \ldots, M_{t-1} \in \{0,1\}^n$ are used to determine the values of $T_1, \ldots, T_{j_1-1}, T_{j_1+1}, \ldots, T_n$, and M_t is used to determine the value of r. Assume that for some k_1, and $T_{j_1} = \mathsf{Enc}(r, k_1)$, it holds that with some positive probability over the randomness of $\mathsf{i}\mathcal{O}$, $A(1^\lambda, a, \mathsf{i}\mathcal{O}(1^\lambda, C_1, \widehat{z}_1))$ outputs 0 with probability 0.01. Then we claim that (j_1, M) can be compressed to length $nt - n/100$, which implies that the claim holds.

Indeed, let $\widehat{C}$ be the output of $\mathsf{i}\mathcal{O}(1^\lambda, C_1, \widehat{z}_1)$, and observe that $\mathsf{size}(\widehat{C}) \leq \sigma(\lambda, 10\lambda^\alpha) \leq \sqrt{n}$. If $A(\widehat{C})$ outputs 0 with probability 0.01, it follows that for on least $1/2$ of the possible inputs x, $\widehat{C}^z$ does not query T_{j_1}. We thus can describe M by describing $j_1, M_1, \ldots, M_{t-1}, \widehat{C}$, and every entry in $\widehat{C}$ for which $\widehat{C}^z$ queries T_{j_1} (note that we do not need to describe the indices themselves, as this can be checked using everything we described so far). To decode the rest of the values we can simply execute $\widehat{C}^z$. The length of the encoding is

$$(t-1)n + \log n + \mathsf{size}(\widehat{C}) + \lambda + n/2,$$

which by our choice of parameters is bounded by $tn - n/100$. □

Claim 55. *For every $\lambda \in \mathbb{N}$,*

$$\mathop{\mathbf{Pr}}_{(\widehat{z}_2, C_2, a) \leftarrow \widehat{\mathcal{P}}_{\lambda,2},\, \mathsf{i}\mathcal{O}}[A(1^\lambda, a, \mathsf{i}\mathcal{O}(1^\lambda, C_2, \widehat{z}_2)) = 1] \leq 1/2.$$

Proof. The proof follows since by the definition of A, A executes $\mathsf{i}\mathcal{O}(1^\lambda, C_2, \widehat{z}_2)$ 100 times on random inputs. Since $\mathsf{size}(\mathsf{i}\mathcal{O}(1^\lambda, C_2, \widehat{z}_2)) \leq \sqrt{n}$, A makes at most $100 \cdot \sqrt{n}$ queries to $\widehat{z}_2$. It follows that A touches at most $100 \cdot \sqrt{n} \leq n/100$ blocks of z, and thus the claim follows by symmetry. □

Proof of Lemma 52. We are now ready to prove Lemma 52.

Proof of Lemma 52. The proof follows by Claim 53, Claim 54, Claim 55, and the triangular inequality. □

7.2 Randomized Encoding for TMs with Database Access

We next define randomized encoding for Turing machines with (oracle) access to a database. In the following we do not allow the overhead to depend on the length of the database nor on the running time of $M(x)$.

For simplicity we only define randomized encoding with perfect functionality.

Definition 56 (Randomized encoding for database-aided TM)**.** *A pair* $(\mathsf{RE.Enc}, \mathsf{RE.Dec})$ *of efficient randomized algorithms is a* randomized encoding for database-aided TMs with output size $\sigma : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ *if the following holds. Let M be a TM and $x \in \{0,1\}^*$ be an input, $z \in \{0,1\}^*$ an database, $\lambda \in \mathbb{N}$ be a security parameter and let $T \in \mathbb{N}$ be a bound on the running time of $M(x)$. Then*

- ***Perfect Functionality:*** $\mathbf{Pr}[\mathsf{RE.Dec}^z(\mathsf{RE.Enc}(1^\lambda, M, x, T, z)) = M^z(x)] = 1$.
- ***Efficiency:*** $\mathsf{RE.Enc}(1^\lambda, M, x, T, z)$ *runs in time* $\mathsf{poly}(\lambda, |M|, |x|, |z|)$ *and* $\mathsf{RE.Dec}^z(\widehat{M(x)})$ *runs in time* $\mathsf{poly}(\lambda, |M|, |x|, t)$ *for* $\widehat{M(x)} \leftarrow \mathsf{RE.Enc}(1^\lambda, M, x, T, z)$ *and where $t \leq T$ is the running time of $M^z(x)$.*

- ***Indistinguishability:*** *For any polynomial-size circuit family $\{A_\lambda\}_\lambda$ and polynomial p, there exists a negligible function μ such that for all TM M and two inputs x_0, x_1 such that $M^z(x_0) = M^z(x_1)$, $|M| \leq p(\lambda), |x_0| \leq p(\lambda), |x_1| \leq p(\lambda)$ and the running time of M^z on x_0 at most $p(\lambda)$ and is the same as the running time of M^z on x_1, the following holds*:

$$|\mathbf{Pr}[A(z, \mathsf{RE.Enc}(1^\lambda, M, x_0, T, z)) = 1] - \mathbf{Pr}[A(z, \mathsf{RE.Enc}(1^\lambda, M, x_1, T, z)) = 1]| = \mu(\lambda).$$

- ***σ-Output-Size:*** *For all $\lambda, \ell \in \mathbb{N}$, $z \in \{0,1\}^*$, TM M and input x, $|\mathsf{RE.Enc}(1^\lambda, M, x, T, z)| \leq \sigma(\lambda, |M|, |x|, \log T)$.*

Importantly, note that the running time of $\mathsf{RE.Dec}^z(\widehat{M(x)})$ should not be dependent on the length of z. In contrast to i$\mathcal{O}$, we do not know if the existence of randomized encoding for TM implies OWFs. We prove the following theorem.

Theorem 57. *Assume the existence of one-way functions. Then there exists no perfect randomized encoding for database-aided TMs with output size σ, for any $\sigma \in \mathsf{poly}$.*

The proof of Theorem 57 follows the same lines of the proofs of Theorem 48 and Theorem 38. Below we describe the distribution $\mathcal{P}_\lambda$ and the distinguisher and omit the rest of the proof. Assume towards a contradiction that there exists randomized encoding RE for database-aided TMs with output size $\sigma \in \mathsf{poly}$, and let $\tau \in \mathsf{poly}$ be the polynomial that bounds the running time of $\mathsf{RE.Dec}$. Let $(\mathsf{Enc}, \mathsf{Dec})$ be a rate-1 pseudorandom encryption scheme with local decryption algorithm LD that can be implemented as a circuit of size $(\lambda + \log |m|)^\alpha$ for some constant α. Let $p(\lambda) = (\sigma(\lambda, \lambda, 2\lambda, (2\lambda)^\alpha))^2$ be a polynomial, and let $n = n_\lambda = p(\lambda)$. Let $t = t_\lambda = 100 \cdot \tau(\lambda, \lambda, 2\lambda, n \cdot (2\lambda)^\alpha)$. We construct a distribution $\mathcal{P}_\lambda$ over database z, TM M, pair of inputs (x_1, x_2) and an advice a, such that $M^z(x_1) = M^z(x_2)$ but $\mathsf{RE.Enc}(1^\lambda, M, x_1, T, z)$ is distinguishable from $\mathsf{RE.Enc}(1^\lambda, M, x_2, T, z)$.

The distribution $\mathcal{P}_\lambda$. We describe below how to sample from $\mathcal{P}_\lambda$. In the following we fix λ and omit it from the notation.

1. Sample $r \leftarrow \{0,1\}^n$, $k_1, k_2 \leftarrow \{0,1\}^\lambda$ and $j_1 \neq j_2 \leftarrow [t]$. For every $i \in [t] \setminus \{j_1, j_2\}$, let $T_i \leftarrow \{0,1\}^n$ be a random function. Let $T_{j_1} = \mathsf{Enc}(r, k_1)$ and $T_{j_2} = \mathsf{Enc}(r, k_2)$. Let $z = T_1 || \ldots || T_t$.
2. Let M be the program that given an index j and key k, reads the j-th block of z and outputs $\mathsf{Dec}(T_j, k)$. Let $x_1 = (j_1, k_1)$ and $x_2 = (j_2, k_2)$. By construction $M^z(x_1) = M^z(x_2) = r$.
3. Let $a = (j_1, z)$.

The following claim follows directly by the construction.

Claim 58. *For every $\lambda \in \mathbb{N}$,*

$$\mathbf{Pr}_{(z,M,x_1,x_2,a) \leftarrow \mathcal{P}_\lambda}[M^z(x_1) = M^z(x_2)] = 1.$$

The distinguisher. We define a distinguisher A that takes as input the security parameter 1^λ, a program $\widehat{M(x)}$ such that $|\widehat{M(x)}| \ll r$ and $\mathsf{RE.Dec}^z(\widehat{M(x)}) = r$ and the index j_1: The distinguisher A executes $\mathsf{RE.Dec}^z(\widehat{M(x)})$. If $\mathsf{RE.Dec}$ queries z on the j_1 block, A outputs 1. Otherwise, A outputs 0.

Observe that by the bound on the running time of $\mathsf{RE.Dec}^z(\widehat{M(x)})$, $\mathsf{RE.Dec}^z(\widehat{M(x)})$ queries at most $t/100$ blocks out of the t blocks of z. Using this observation, the proof of the next lemma follows similarly to the proof of Lemma 41.

Lemma 59. *Assume that* RE *is a perfect randomized encoding for database-aided TMs with output size* σ*. Then for infinitely many* $\lambda \in \mathbb{N}$*,*

$$\Big| \Pr_{(z,M,x_1,x_2,a)\leftarrow\mathcal{P}_\lambda}[A(1^\lambda, z, a, \mathsf{RE.Enc}(1^\lambda, M, x_1, T, z)) = 1] - \Pr_{(z,M,x_1,x_2,a)\leftarrow\mathcal{P}_\lambda}[A(1^\lambda, z, a, \mathsf{RE.Enc}(1^\lambda, M, x_2, T, z)) = 1]\Big| \geq 1/3.$$

We now conclude the proof of the main theorem.

Proof of Theorem 57. The proof follows by combining Claim 58 and Lemma 59, which together contradict the security of RE. □

References

[ACM+21] Eric Allender, Mahdi Cheraghchi, Dimitrios Myrisiotis, Harsha Tirumala, and Ilya Volkovich. One-way functions and a conditional variant of MKTP. In *Conference on Foundations of Software Technology and Theoretical Computer Science,* (FSTTCS), pages 7:1–7:19, 2021. 11

[Agr19] Shweta Agrawal. Indistinguishability obfuscation without multilinear maps: new methods for bootstrapping and instantiation. In *International Conference on the Theory and Applications of Cryptographic Techniques* (EUROCRYPT), pages 191–225, 2019. 3

[AJL+19] Prabhanjan Ananth, Aayush Jain, Huijia Lin, Christian Matt, and Amit Sahai. Indistinguishability obfuscation without multilinear maps: new paradigms via low degree weak pseudorandomness and security amplification. In *International Cryptology Conference* (CRYPTO), pages 284–332. Springer, 2019. 3

[AJS17] Prabhanjan Ananth, Abhishek Jain, and Amit Sahai. Indistinguishability obfuscation for turing machines: Constant overhead and amortization. In *International Cryptology Conference* (CRYPTO), pages 252–279, 2017. 3, 12

[AJS18] Prabhanjan Ananth, Aayush Jain, and Amit Sahai. Indistinguishability obfuscation without multilinear maps: iO from LWE, bilinear maps, and weak pseudorandomness. *Cryptology ePrint Archive*, 2018. 3

[APM20] Shweta Agrawal and Alice Pellet-Mary. Indistinguishability obfuscation without maps: attacks and fixes for noisy linear FE. In *International Conference on the Theory and Applications of Cryptographic Techniques* (EUROCRYPT), pages 110–140, 2020. 3

[BCC+14] Nir Bitansky, Ran Canetti, Henry Cohn, Shafi Goldwasser, Yael Tauman Kalai, Omer Paneth, and Alon Rosen. The impossibility of obfuscation with auxiliary input or a universal simulator. In *Annual Cryptology Conference* (CRYPTO), pages 71–89, 2014. 11

[BCP14] Elette Boyle, Kai-Min Chung, and Rafael Pass. On extractability obfuscation. In *Theory of cryptography conference*, pages 52–73. Springer, 2014. 3

[BDGM19] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Leveraging linear decryption: Rate-1 fully-homomorphic encryption and time-lock puzzles. In *Theory of Cryptography Conference*, pages 407–437. Springer, 2019. 3

[BDGM22] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Factoring and pairings are not necessary for IO: circular-secure LWE suffices. In *International Colloquium on Automata, Languages, and Programming* (ICALP), volume 229, pages 28:1–28:20, 2022. 3

[BDGM23] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Candidate io from homomorphic encryption schemes. *Journal of Cryptology*, 36(3):27, 2023. 3

[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. *J. ACM*, 59(2):6:1–6:48, 2012. 3, 11, 15

[BGL+15] Nir Bitansky, Sanjam Garg, Huijia Lin, Rafael Pass, and Sidharth Telang. Succinct randomized encodings and their applications. In *Symposium on Theory of Computing* (STOC), pages 439–448, 2015. 3

[BIJ+20] James Bartusek, Yuval Ishai, Aayush Jain, Fermi Ma, Amit Sahai, and Mark Zhandry. Affine determinant programs: A framework for obfuscation and witness encryption. In *Innovations in Theoretical Computer Science Conference* (ITCS), pages 82:1–82:39, 2020. 3

[BIM+23] Elette Boyle, Yuval Ishai, Pierre Meyer, Robert Robere, and Gal Yehuda. On low-end obfuscation and learning. In *Innovations in Theoretical Computer Science Conference* (ITCS), pages 23:1–23:28, 2023. 11

[BP15] Nir Bitansky and Omer Paneth. ZAPs and non-interactive witness indistinguishability from indistinguishability obfuscation. In *Theory of Cryptography Conference* (TCC), pages 401–427, 2015. 3

[BPR15] Nir Bitansky, Omer Paneth, and Alon Rosen. On the cryptographic hardness of finding a Nash equilibrium. In *Symposium on Foundations of Computer Science* (FOCS), pages 1480–1498, 2015. 3

[BPW15] Nir Bitansky, Omer Paneth, and Daniel Wichs. Perfect structure on the edge of chaos: Trapdoor permutations from indistinguishability obfuscation. In *Theory of Cryptography Conference* (TCC), pages 474–502. Springer, 2015. 3

[Bus97] Samuel R. Buss. Bounded arithmetic and propositional proof complexity. In *Logic of Computation*, pages 67–121. Springer Berlin Heidelberg, 1997. 22

[BV15] Nir Bitansky and Vinod Vaikuntanathan. Indistinguishability obfuscation from functional encryption. In *Symposium on Foundations of Computer Science* (FOCS), pages 171–190, 2015. 3

[BZ17] Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. *Algorithmica*, 79:1233–1285, 2017. 3

[CCMR24] Ran Canetti, Claudio Chamon, Eduardo R. Mucciolo, and Andrei E. Ruckenstein. Towards general-purpose program obfuscation via local mixing. *IACR Cryptol. ePrint Arch.*, page 6, 2024. 3

[CHJV14] Ran Canetti, Justin Holmgren, Abhishek Jain, and Vinod Vaikuntanathan. Indistinguishability obfuscation of iterated circuits and RAM programs. *Cryptology ePrint Archive*, 2014. 3

[CHN+18] Aloni Cohen, Justin Holmgren, Ryo Nishimaki, Vinod Vaikuntanathan, and Daniel Wichs. Watermarking cryptographic capabilities. *SIAM J. Comput.*, 47(6):2157–2202, 2018. 3

[CKP15] Ran Canetti, Yael Tauman Kalai, and Omer Paneth. On obfuscation with random oracles. In *Theory of Cryptography Conference* (TCC), pages 456–467, 2015. 11

[CLP15] Kai-Min Chung, Huijia Lin, and Rafael Pass. Constant-round concurrent zero-knowledge from indistinguishability obfuscation. In *International Cryptology Conference* (CRYPTO), pages 287–307, 2015. 3

[Fei98] Uriel Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998. 8

[For17] Lance Fortnow. NP in ZPP implies PH in ZPP, 2017. Post available at (accessed on 20/6/2024): https://blog.computationalcomplexity.org/2017/03/np-in-zpp-implies-ph-in-zpp.html. 15

[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 *Symposium on Foundations of Computer Science* (FOCS), pages 40–49, 2013. 3

[GGHR14] Sanjam Garg, Craig Gentry, Shai Halevi, and Mariana Raykova. Two-round secure mpc from indistinguishability obfuscation. In *Theory of Cryptography Conference*, pages 74–94. Springer, 2014. 3

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986. 9, 34

[GJK18] Craig Gentry, Charanjit S. Jutla, and Daniel Kane. Obfuscation using tensor products. Cryptology ePrint Archive, Paper 2018/756, 2018. 3

[GJLS21] Romain Gay, Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from simple-to-state hard problems: New assumptions, new techniques, and simplification. In *International Conference on the Theory and Applications of Cryptographic Techniques* (EUROCRYPT), pages 97–126, 2021. 3

[GK05] Shafi Goldwasser and Yael Tauman Kalai. On the impossibility of obfuscation with auxiliary input. In *Symposium on Foundations of Computer Science* (FOCS), pages 553–562, 2005. 11

[GLSW15] Craig Gentry, Allison Bishop Lewko, Amit Sahai, and Brent Waters. Indistinguishability obfuscation from the multilinear subgroup elimination assumption. In *Symposium on Foundations of Computer Science* (FOCS), pages 151–170, 2015. 3

[Gol04] Oded Goldreich. *Foundations of Cryptography: Basic Applications*. Cambridge University Press, 2004. 3

[GP21] Romain Gay and Rafael Pass. Indistinguishability obfuscation from circular security. In *Symposium on Theory of Computing* (STOC), pages 736–749, 2021. 3

[GR07] Shafi Goldwasser and Guy N. Rothblum. On best-possible obfuscation. In *Theory of Cryptography Conference* (TCC), pages 194–213, 2007. 1, 5, 6, 11, 40

[Had00] Satoshi Hada. Zero-knowledge and code obfuscation. In *International Conference on the Theory and Application of Cryptology and Information Security* (ASIACRYPT), pages 443–457, 2000. 11

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999. 9, 34

[Hir22] Shuichi Hirahara. NP-hardness of learning programs and partial MCSP. In *Symposium on Foundations of Computer Science* (FOCS), pages 968–979, 2022. 7

[HIR23] Yizhi Huang, Rahul Ilango, and Hanlin Ren. NP-hardness of approximating meta-complexity: A cryptographic approach. In *Symposium on Theory of Computing* (STOC), pages 1067–1075, 2023. 12

[Ila20] Rahul Ilango. Approaching MCSP from above and below: Hardness for a conditional variant and $\mathrm{AC}^0[p]$. In *Innovations in Theoretical Computer Science Conference* (ITCS), pages 34:1–34:26, 2020. 11

[ILO20] Rahul Ilango, Bruno Loff, and Igor C. Oliveira. NP-hardness of circuit minimization for multi-output functions. In *Computational Complexity Conference* (CCC), pages 22:1–22:36, 2020. 1, 7, 8, 9, 10

[ILW23] Rahul Ilango, Jiatu Li, and Ryan Williams. Indistinguishability obfuscation, range avoidance, and bounded arithmetic. In *Symposium on Theory of Computing* (STOC), pages 1076–1089, 2023. 12

[JJ22] Abhishek Jain and Zhengzhong Jin. Indistinguishability obfuscation via mathematical proofs of equivalence. In *Symposium on Foundations of Computer Science* (FOCS), pages 1023–1034, 2022. 12

[JLL23] Aayush Jain, Huijia Lin, and Ji Luo. On the optimal succinctness and efficiency of functional encryption and attribute-based encryption. In *International Conference on the Theory and Applications of Cryptographic Techniques* (EUROCRYPT), pages 479–510, 2023. 3

[JLMS19] Aayush Jain, Huijia Lin, Christian Matt, and Amit Sahai. How to leverage hardness of constant-degree expanding polynomials over R to build iO. In *International Conference on the Theory and Applications of Cryptographic Techniques* (EUROCRYPT), pages 251–281, 2019. 3

[JLS21] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In *Symposium on Theory of Computing* (STOC), pages 60–73, 2021. 3

[JLS22] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from LPN over $\mathbb{F}_p$, DLIN, and PRGs in NC^0. In *International Conference on the Theory and Applications of Cryptographic Techniques* (EUROCRYPT), pages 670–699, 2022. 3

[JLS24] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. *Commun. ACM*, 67(3):97–105, 2024. 3

[Juk12] Stasys Jukna. *Boolean function complexity: advances and frontiers*, volume 27. Springer, 2012. 13, 33

[KLW15] Venkata Koppula, Allison Bishop Lewko, and Brent Waters. Indistinguishability obfuscation for turing machines with unbounded memory. In *Symposium on Theory of Computing* (STOC), pages 419–428, 2015. 3

[KMN+22] Ilan Komargodski, Tal Moran, Moni Naor, Rafael Pass, Alon Rosen, and Eylon Yogev. One-way functions and (im)perfect obfuscation. *SIAM J. Comput.*, 51(6):1769–1795, 2022. 8, 17

[KNY17] Ilan Komargodski, Moni Naor, and Eylon Yogev. Secret-sharing for NP. *Journal of Cryptology*, 30(2):444–469, 2017. 3

[Lin16] Huijia Lin. Indistinguishability obfuscation from constant-degree graded encoding schemes. In *International Conference on the Theory and Applications of Cryptographic Techniques* (EUROCRYPT), pages 28–57, 2016. 3

[Lin17] Huijia Lin. Indistinguishability obfuscation from SXDH on 5-linear maps and locality-5 PRGs. In *International Cryptology Conference* (CRYPTO), pages 599–629, 2017. 3

[LP22] Yanyi Liu and Rafael Pass. On one-way functions from NP-complete problems. In *Computational Complexity Conference* (CCC), pages 36:1–36:24, 2022. 11

[LPST16] Huijia Lin, Rafael Pass, Karn Seth, and Sidharth Telang. Indistinguishability obfuscation with non-trivial efficiency. In *International Conference on Practice and Theory in Public-Key Cryptography* (PKC), pages 447–462, 2016. 4, 11

[LT17] Huijia Lin and Stefano Tessaro. Indistinguishability obfuscation from trilinear maps and block-wise local prgs. In *International Cryptology Conference* (CRYPTO), pages 630–660. Springer, 2017. 3

[LV16] Huijia Lin and Vinod Vaikuntanathan. Indistinguishability obfuscation from DDH-like assumptions on constant-degree graded encodings. In *Symposium on Foundations of Computer Science* (FOCS), pages 11–20, 2016. 3

[MMN15] Mohammad Mahmoody, Ameer Mohammed, and Soheil Nematihaji. More on impossibility of virtual black-box obfuscation in idealized models. *IACR Cryptol. ePrint Arch.*, 2015:632, 2015. 11

[MP24] Noam Mazor and Rafael Pass. Gap MCSP is not (Levin) NP-complete in Obfustopia. In *Computational Complexity Conference* (CCC), 2024. 1, 6, 7, 12, 17, 18

[Pic15] Ján Pich. Logical strength of complexity theory and a formalization of the PCP theorem in bounded arithmetic. *Logical Methods in Computer Science*, 11, 2015. 8, 22, 23

[PS15] Rafael Pass and Abhi Shelat. Impossibility of VBB obfuscation with ideal constant-degree graded encodings. In *Theory of Cryptography Conference* (TCC), pages 3–17, 2015. 11

[PST14] Rafael Pass, Karn Seth, and Sidharth Telang. Indistinguishability obfuscation from semantically-secure multilinear encodings. In *International Cryptology Conference* (CRYPTO), pages 500–517, 2014. 3

[PY91] Christos H. Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. *J. Comput. Syst. Sci.*, 43(3):425–440, 1991. 8, 24

[Rom90] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *Symposium on Theory of Computing* (STOC), pages 387–394, 1990. 8, 13

[RVW00] Omer Reingold, Salil P. Vadhan, and Avi Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors. In *Symposium on Foundations of Computer Science* (FOCS), pages 3–13, 2000. 24

[SW21] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: Deniable encryption, and more. *SIAM J. Comput.*, 50(3):857–908, 2021. 3