Hardness vs. (Very Little) Structure in Cryptography: A Multi-Prover Interactive Proofs Perspective

Gil Segev
School of Computer Science and Engineering, Hebrew University of Jerusalem, Jerusalem, Israel
segev@cs.huji.ac.il

Ido Shahaf
School of Computer Science and Engineering, Hebrew University of Jerusalem, Jerusalem, Israel
ido.shahaf@cs.huji.ac.il

Abstract

The hardness of highly-structured computational problems gives rise to a variety of public-key primitives. On one hand, the structure exhibited by such problems underlies the basic functionality of public-key primitives, but on the other hand it may endanger public-key cryptography in its entirety via potential algorithmic advances. This subtle interplay initiated a fundamental line of research on whether structure is inherently necessary for cryptography, starting with Rudich's early work (PhD Thesis '88) and recently leading to that of Bitansky, Degwekar and Vaikuntanathan (CRYPTO '17).

Identifying the structure of computational problems with their corresponding complexity classes, Bitansky et al. proved that a variety of public-key primitives (e.g., public-key encryption, oblivious transfer and even functional encryption) cannot be used in a black-box manner to construct either any hard language that has NP-verifiers both for the language itself and for its complement, or any hard language (and even promise problem) that has a statistical zero-knowledge proof system – corresponding to hardness in the structured classes $\mathsf{NP} \cap \mathsf{coNP}$ or SZK, respectively, from a black-box perspective.

In this work we prove that the same variety of public-key primitives do not inherently require *even very little structure* in a black-box manner: We prove that they do not imply any hard language that has *multi-prover interactive proof systems* both for the language and for its complement – corresponding to hardness in the class $\mathsf{MIP} \cap \mathsf{coMIP}$ from a black-box perspective. Conceptually, given that $\mathsf{MIP} = \mathsf{NEXP}$, our result rules out languages with very little structure.

Already the cases of languages that have IP or AM proof systems both for the language itself and for its complement, which we rule out as immediate corollaries, lead to intriguing insights. For the case of IP, where our result can be circumvented using non-black-box techniques, we reveal a gap between black-box and non-black-box techniques. For the case of AM, where circumventing our result via non-black-box techniques would be a major development, we both strengthen and unify the proofs of Bitansky et al. for languages that have NP-verifiers both for the language itself and for its complement and for languages that have a statistical zero-knowledge proof system.

2012 ACM Subject Classification Theory of computation → Computational complexity and cryptography

Keywords and phrases Hardness vs. Structure, Black-box Constructions, Interactive Proofs

Digital Object Identifier 10.4230/LIPIcs.ITC.2020.10

Related Version A full version of the paper is available at `https://eprint.iacr.org/2020/330`.

Funding Supported by the European Union's Horizon 2020 Framework Program (H2020) via an ERC Grant (Grant No. 714253).
Ido Shahaf: Supported by the Clore Israel Foundation via the Clore Scholars Programme.

1st Conference on Information-Theoretic Cryptography (ITC 2020).
Editors: Yael Tauman Kalai, Adam D. Smith, and Daniel Wichs; Article No. 10; pp. 10:1–10:23
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Starting with the revolutionary invention of public-key cryptography [18, 32, 23], the hardness of highly-structured computational problems (e.g., factoring, discrete log, or various lattice-based problems) has given rise to a variety of public-key primitives. On one hand, the structure exhibited by such problems underlies the basic functionality of nearly all such primitives, but on the other hand it may also danger their conjectured hardness. As noted by Barak [5], this *"makes public-key cryptography somewhat of an endangered species that could wiped out by a surprising algorithmic advance"*.

This subtle interplay has led to the long-studied question of whether structure is inherently necessary for certain cryptographic primitives, and most notably for public-key primitives. While there may be different approaches for measuring or quantifying "structure", the main approach taken by the cryptography community over the years relies on computational complexity: Understanding which cryptographic primitives inherently require hardness in "structured" complexity classes such as $\mathsf{NP} \cap \mathsf{coNP}$, TFNP and SZK.

There are only a few known examples of cryptographic primitives that require hardness in such classes. Most notably, one-way permutations imply hardness in $\mathsf{NP} \cap \mathsf{coNP}$ [15], homomorphic encryption and non-interactive computational private-information retrieval imply hardness in SZK [14, 28], and indistinguishability obfuscation implies hardness in $\mathsf{PPAD} \subseteq \mathsf{TFNP}$ unless $\mathsf{NP} \subseteq \mathsf{ioBPP}$ [11, 21, 27].

Within the classic framework of black-box constructions, capturing "natural" cryptographic constructions [26, 31], Rudich [33] showed (based on [13, 25]) that a one-way function cannot be used in black-box manner to construct NP-verifiers for any hard language both for the language itself and for its complement – corresponding to hardness in $\mathsf{NP} \cap \mathsf{coNP}$ from a black-box perspective (we note that the known examples stated above all follow in such a black-box manner).

For several decades no progress has been made in extending Rudich's result to public-key primitives or to other complexity classes. This situation has recently changed dramatically with the work of Bitansky, Degwekar and Vaikuntanathan [10] (see also the refinements in the more recent work of Bitansky and Degwekar [9]): They showed that even indistinguishability obfuscation cannot be used in a black-box manner to construct any hard language that has NP verifiers both for the language itself and for its complement, or any hard language (and even a promise problem) that has a statistical zero-knowledge proof system – corresponding to hardness in $\mathsf{NP} \cap \mathsf{coNP}$ or SZK, respectively, from a black-box perspective. Proving their result within the framework of Asharov and Segev [2, 3] capturing indistinguishability obfuscation for oracle-aided computations, Bitansky et al. in fact proved their result for all primitives that can be based on indistinguishability obfuscation for circuits that access an injective one-way function in a black-box manner. These include, in particular, a variety of public-key primitives including public-key encryption, oblivious transfer and even functional encryption.

Focusing on the classes $\mathsf{NP} \cap \mathsf{coNP}$ and SZK , Bitansky et al. showed that, from a black-box perspective, public-key cryptography does not inherently require highly-structured hardness. However, going back to Barak's concern [5], even less stringent forms of structure may still endanger public-key cryptography in its entirety. This leads to the following fundamental question aiming at substantially refining our understanding of the interplay between hardness and structure:

Does public-key cryptography inherently require hardness in complexity classes that are "less structured" than $\mathsf{NP} \cap \mathsf{coNP}$ or SZK?

1.1 Our Contributions

In this work we show that a wide variety of public-key primitives do not inherently require even very little structure in a black-box manner. Specifically, we prove that such primitives do not naturally imply hard languages that have *multi-prover interactive proof systems* (MIP) [8] both for the language and for its complement.

Conceptually, given that MIP = NEXP [4], our result considers languages with very little structure. Already the cases of languages that have IP or AM proof systems both for the language itself and for its complement, which we obtain as immediate corollaries, lead to intriguing insights. For the case of IP, where our result can be circumvented using non-black-box techniques, we reveal a gap between black-box and non-black-box techniques (as we discuss below). For the case of AM, where circumventing our result via non-black-box techniques would be a major development, we both strengthen and unify the proofs of Bitansky et al. for languages that have NP-verifiers both for the language itself and for its complement and for languages that have a statistical zero-knowledge proof system (since $\mathsf{NP} \subseteq \mathsf{AM}$ by definition, and since $\mathsf{SZK} \subseteq \mathsf{AM} \cap \mathsf{coAM}$ in a black-box manner [19, 1]).[1]

The following is an informal statement of our main result. We refer the reader to Section 1.2 for an overview of our result, and to Sections 3 and 4 for a formal definition of the class of constructions to which our result applies and for a formal theorem statement, respectively.

▶ **Theorem 1** (Informal). *There is no fully black-box construction of a pair of multi-prover interactive proof systems, Π and $\overline{\Pi}$, corresponding to a worst-case hard language L and to its complement $\overline{L}$, respectively, from an injective one-way function f and an indistinguishability obfuscator for the class of all oracle-aided circuits C^f.*

Note that as our result rules out constructions of languages that are *worst-case hard*, then it rules out in particular constructions of languages that are *average-case hard*.

Black-box vs. non-black-box constructions. Our result might seem too strong and somewhat contradicting to the fact that any one-way function implies a hard (even on average) language in $\mathsf{NP} \subseteq \mathsf{IP}$ in a *black-box manner*. Given that IP is closed under complement [30, 36], then

$$\mathsf{NP} \subseteq \mathsf{IP} \cap \mathsf{coIP} \subseteq \mathsf{MIP} \cap \mathsf{coMIP}.$$

In particular, any one-way function implies a hard language that has IP proof systems both for the language itself and for its complement, which seemingly contradicts our result. However, this sequence of containments cannot be established via relativizing reductions, and thus there is in fact no contradiction (note that any black-box reduction relativizes [31]), but rather a gap between black-box and non-black-box techniques. Specifically, Chang et al. [16] showed that there exists an oracle Γ relative to which $\mathsf{NP}^\Gamma \not\subseteq \mathsf{coIP}^\Gamma$, and in particular IP is *not* closed under complement with respect to relativizing reductions. Still, as mentioned above, our impossibility result already applies to $\mathsf{AM} \cap \mathsf{coAM}$, for which circumventing our result via non-black-box techniques would be a major development. We discuss this in much more detail in Section 1.2 in the context of black-box representations of complexity classes.

[1] We note that the result of Bitansky et al. for SZK holds not only for languages but in fact also for promise problems. This, however, cannot be covered by our result since already a hard promise problem that has NP verifiers both for its "YES" instances and for its "NO" instances can be constructed based on any one-way function in a black-box manner.

Implications to public-key cryptography. Similarly to Bitansky et al. [10] we prove our result within the framework of Asharov and Segev [2, 3], capturing indistinguishability obfuscation for oracle-aided circuits. Indistinguishability obfuscation for such circuits suffices for realizing a variety of public-key primitives (e.g., public-key encryption, oblivious transfer and even functional encryption) in a fully black-box manner [34, 37, 2], and therefore as a corollary we obtain that there is no construction of the above form based on any of these primitives.

We strongly emphasize that our result is unconditional, and in particular does not depend on whether or not indistinguishability obfuscation actually exists. Even if it does not exist in the actual world, then within the framework of Asharov and Segev it does exist information theoretically, and it implies the above variety of public-key primitives to which our result applies (once again, in an unconditional manner).

1.2 Overview of Our Approach

In this section we provide a high-level overview of the framework in which we prove our impossibility result, and then describe the main ideas and challenges underlying our proof.

Black-box constructions. Our goal is to prove a statement along the lines of "a cryptographic primitive $\mathcal{P}$ does not naturally imply a hard language in a complexity class $\mathcal{C}$". However, it is not clear how to prove such a statement in an unconditional manner, as it may be the case that the class $\mathcal{C}$ (e.g., $\mathsf{NP} \cap \mathsf{coNP}$ as discussed by Bitansky et al. [10]) does not contain hard languages. One possible approach is to prove a result that is conditioned on a specific assumption, but then it may be the case that the assumption itself already rules out the existence of hard languages in the class $\mathcal{C}$. Obtaining substantial insight using such an approach requires a deep understanding of the interplay between the primitive $\mathcal{P}$, the complexity class $\mathcal{C}$ and the additional assumption – which is somewhat rare when considering cryptographic primitives and assumptions.

Faced with such difficulties, the cryptography community has relied over the years on the framework of black-box constructions [26, 31] for proving impossibility results for "natural" construction techniques. In our context, a fully black-box construction of a hard language $L \in \mathcal{C}$ based on a cryptographic primitive $\mathcal{P}$ consists of two ingredients. The first ingredient is a "construction" of a language $L^{\mathcal{P}}$ that completely ignores the internal implementation of $\mathcal{P}$ and only requires black-box access to any given implementation of $\mathcal{P}$. Here, the notion of a "construction" depends on the specific complexity class $\mathcal{C}$. For example, in a natural black-box interpretation of $\mathsf{NP} \cap \mathsf{coNP}$, Rudich [33] and Bitansky et al. [10] considered as a construction a pair of oracle-aided NP-verifiers, V and $\overline{V}$, for the language itself and for its complement, respectively, where the verifiers have black-box access to the primitive $\mathcal{P}$. That is, for any oracle realizing $\mathcal{P}$, the two verifiers must be valid in the sense that for any instance $x \in \{0,1\}^*$ either there exists a "yes" witness for $V^{\mathcal{P}}$ and there do not exist any "no" witnesses for $\overline{V}^{\mathcal{P}}$ (i.e., $x \in L^{\mathcal{P}}$), or there exists a "no" witness for $\overline{V}^{\mathcal{P}}$ and there do not exist any "yes" witnesses for $V^{\mathcal{P}}$ (i.e., $x \notin L^{\mathcal{P}}$) – but never both. The second ingredient, is a black-box proof of hardness, showing that for any implementation of the primitive $\mathcal{P}$, any algorithm that decides the language $L^{\mathcal{P}}$ can be efficiently used in a black-box manner for breaking the security of the given implementation of $\mathcal{P}$.

At this point we would like to already emphasize that a "black-box representation" of a complexity class is in fact not unique, and that different representations are not always equivalent from a black-box perspective. For example, a natural black-box representation for the class $\mathsf{IP} \cap \mathsf{coIP}$ relative to a given primitive $\mathcal{P}$ is to consider all languages that have interactive proof systems both for the language itself and for its complement, where the two proof systems access $\mathcal{P}$ in a black-box manner. However, since IP is closed under complement [30, 36] then $\mathsf{IP} \cap \mathsf{coIP} = \mathsf{IP}$ and therefore an additional representation is to consider all languages that have interactive proof systems for the language itself (without considering its complement) where the proof system accesses $\mathcal{P}$ in a black-box manner. As discussed in Section 1.1, these two representations are not equivalent from a black-box perspective since IP is not closed under complement with respect to relativizing reductions.

The structure of our proof. Following Bitansky et al. [10] we prove our result within the framework of Asharov and Segev [2] for capturing black-box constructions based on indistinguishability obfuscation, utilizing the latter as a "central hub" for deriving impossibility results for a variety of public-key primitives. As observed by Asharov and Segev, although constructions that are based on indistinguishability obfuscation are almost always *non-black-box*, most of their non-black-box techniques have essentially the same flavor: The obfuscator itself is used in a black-box manner and applied to circuits that can be constructed in a fully black-box manner from a low-level primitive, such as a one-way function. Thus, even though the obfuscator requires concrete implementations of such circuits, by introducing the stronger primitive of an indistinguishability obfuscator for *oracle-aided* circuits (see Section 2), Asharov and Segev showed that such non-black-box techniques in fact directly translate into black-box ones. These include, in particular, non-black-box techniques such as the punctured programming approach of Sahai and Waters [34] and Waters [37] leading to the construction of a variety of public-key primitive. Relying on the transitivity of black-box reductions, this enables to rule out black-box constructions based on all of these primitives by focusing only on indistinguishability obfuscation for oracle-aided circuits and one-way functions.

In order to prove our impossibility result within this framework, we present a distribution over oracles Γ relative to which we prove the following two properties:

- Relative to a random instance of Γ there exist an injective one-way function f and an indistinguishability obfuscator $i\mathcal{O}$ for the class of all oracle-aided circuits C^f.
- Relative to any instance of Γ, we can efficiently decide in the worst case any language that has multi-prover interactive proof systems, $\Pi^{f,i\mathcal{O}}$ and $\overline{\Pi}^{f,i\mathcal{O}}$, for the language itself and for its complement, respectively.[2]

Our oracle Γ is a pair of the form $(\Psi, \mathsf{Decide}^{\Psi})$, where Ψ is based on the oracle of Asharov and Segev that realizes a one-way function and an indistinguishability obfuscator, and Decide^{Ψ} is a generalization of the "decision oracle" introduced by Bitansky et al. for deciding languages that rely on Ψ in a black-box manner (more specifically, whose black-box representation as discussed above relies on Ψ). In the work of Bitansky et al. the decision oracle is defined in a manner that allows to easily decide any language L^{Ψ} that has NP-verifiers, V^{Ψ} and $\overline{V}^{\Psi}$, for the language itself and for its complement, and the main technical challenge underlying their work is proving that Ψ realizes a one-way function and an indistinguishability obfuscator relative to the decision oracle.

Our decision oracle is a natural generalization that allows to easily decide any language L^{Ψ} that has multi-prover proof systems, Π^{Ψ} and $\overline{\Pi}^{\Psi}$, for the language itself and for its complement. This decision oracle seems much more powerful than that of Bitansky et al.

[2] In fact, as discussed below we allow the honest provers to depend on the one-way function and the obfuscator in an arbitrary non-black-box manner, and only require that the verifiers are constructed in a black-box manner (this makes our result stronger).

as it decides a significantly larger class of languages, and our technical effort is devoted to proving that the oracle Ψ still realizes a one-way function and an indistinguishability obfuscator even relative to our generalized decision oracle.

In what follows we describe the decision oracle of Bitansky et al. (to which we refer as the BDV decision oracle) and discuss its key property that underlies their approach. Then, we describe our generalized oracle, relative to which this key property no longer seems to hold, and then describe our the main ideas underlying our proof.

The BDV decision oracle. For any oracle Ψ, taken from an appropriate family $\mathfrak{S}$ of oracles, the BDV decision oracle $\mathsf{Decide}_{\mathfrak{S}}^{\Psi}$ takes as input a triplet $(V, \overline{V}, x)$, where V and $\overline{V}$ are oracle-aided circuits. The oracle first checks whether or not the pair $(V, \overline{V})$ indeed consists of valid NP-verifiers for a language and for its complement in the standard black-box sense discussed above. That is, checks whether or not for any $\Psi' \in \mathfrak{S}$ and $x' \in \{0,1\}^n$ exactly one out of the following two cases holds: (1) There exists a "yes" witness w' such that $V^{\Psi'}(x', w') = 1$ and there do not exist any "no" witnesses w' such that $\overline{V}^{\Psi'}(x', w') = 1$; (2) there exists a "no" witness w' such that $\overline{V}^{\Psi'}(x', w') = 1$ and there do not exist any "yes" witnesses w' such that $V^{\Psi}(x', w') = 1$ (note that the witnesses are allowed to depend on Ψ'). If $(V, \overline{V})$ is not valid in this sense, then the oracle outputs $\perp$. If $(V, \overline{V})$ is valid, then the oracle outputs 1 if $x \in L^{\Psi}$ and 0 otherwise, where L^{Ψ} is the language defined by $(V^{\Psi}, \overline{V}^{\Psi})$.

Then, any language that has oracle-aided NP-verifiers both for the language itself and for its complement with respect to any $\Psi \in \mathfrak{S}$, can be easily decided in the worst case by an algorithm that issues a single query to the BDV decision oracle. The main challenge in the work of Bitansky et al. was in showing that a random instance of Ψ that is sampled from the family $\mathfrak{S}$ of oracles introduced by Asharov and Segev (or from any other appropriate family) realizes a one-way function and an indistinguishability obfuscator even relative to $\mathsf{Decide}_{\mathfrak{S}}^{\Psi}$.

The existence of small critical sets. The key property underlying the proof of Bitansky et al. is the following observation on the existence of "small critical sets". Fix an oracle $\Psi \in \mathfrak{S}$ and let $(V, \overline{V}, x)$ be a query to their decision oracle such that the pair $(V, \overline{V})$ is valid in the above sense, and V and $\overline{V}$ issue at most q oracle queries. Then, there exists a "critical set" of at most q queries, such that for any oracle $\Psi' \in \mathfrak{S}$ that agrees with Ψ on the outputs of all queries from the critical set it holds that $\mathsf{Decide}_{\mathfrak{S}}^{\Psi}(V, \overline{V}, x) = \mathsf{Decide}_{\mathfrak{S}}^{\Psi'}(V, \overline{V}, x)$.

The existence of such a small critical set follows from the $\mathsf{NP} \cap \mathsf{coNP}$ structure of the pair $(V, \overline{V})$. Specifically, assume without loss of generality that $x \in L^{\Psi}$, and let w be a witness such that $V^{\Psi}(x, w) = 1$. Define the set of critical queries as all Ψ-queries that are issued in the computation $V^{\Psi}(x, w)$, and let Ψ' by any oracle that agrees with Ψ on this set. Then clearly $V^{\Psi'}(x, w) = V^{\Psi}(x, w) = 1$, and the validity of the pair $(V, \overline{V})$ guarantees that there is no witness $\tilde{w}$ such that $\overline{V}^{\Psi'}(x, \tilde{w}) = 1$. Thus, $\mathsf{Decide}_{\mathfrak{S}}^{\Psi}(V, \overline{V}, x) = \mathsf{Decide}_{\mathfrak{S}}^{\Psi'}(V, \overline{V}, x) = 1$.

Relying on this key property, Bitansky et al. proved that Ψ realizes a one-way function and an indistinguishability obfuscator relative to their decision oracle via an elegant sequence of hybrids in each case. Specifically, in each sequence the first experiment is the actual security experiment of the one-way function or the indistinguishability obfuscator, the last experiment is one in which no algorithm can achieve any advantage, and the transition between each consecutive pair of experiment is enabled by this key property (or via standard arguments).

Representing MIP ∩ coMIP in a black-box manner. In order to describe our approach, we first need to describe our black-box representation of languages in the complexity class $\mathsf{MIP} \cap \mathsf{coMIP}$. Naturally generalizing the approach of Rudich and Bitansky et al. for $\mathsf{NP} \cap \mathsf{coNP}$, we consider pairs of polynomial-time oracle-aided MIP-verifiers, V and $\overline{V}$, for the language itself and for its complement, respectively, subject to a similar validity requirement of their black-box flavor: For any oracle Ψ taken from an appropriate family $\mathfrak{S}$ of oracles, there should exist a language L^Ψ such that the following two conditions are satisfied[3]:

- For every $x \in L^\Psi$ there exist computationally-unbounded provers $P_1, \ldots, P_N$ such that[4]

$$\Pr_{r \leftarrow \{0,1\}^{\mathsf{poly}(|x|)}}\left[\left\langle V^\Psi(x;r), P_1, \ldots, P_N \right\rangle = 1\right] \geq 2/3\ ,$$

  and for every computationally-unbounded provers $\overline{P}_1, \ldots, \overline{P}_N$ it holds that

$$\Pr_{r \leftarrow \{0,1\}^{\mathsf{poly}(|x|)}}\left[\left\langle \overline{V}^\Psi(x;r), \overline{P}_1, \ldots, \overline{P}_N \right\rangle = 1\right] \leq 1/3\ .$$

- For every $x \notin L^\Gamma$ there exist computationally-unbounded provers $\overline{P}_1, \ldots, \overline{P}_N$ such that

$$\Pr_{r \leftarrow \{0,1\}^{\mathsf{poly}(|x|)}}\left[\left\langle \overline{V}^\Psi(x;r), \overline{P}_1, \ldots, \overline{P}_N \right\rangle = 1\right] \geq 2/3\ ,$$

  and for every computationally-unbounded provers $P_1, \ldots, P_N$ it holds that

$$\Pr_{r \leftarrow \{0,1\}^{\mathsf{poly}(|x|)}}\left[\left\langle V^\Psi(x;r), P_1, \ldots, P_N \right\rangle = 1\right] \leq 1/3\ .$$

Note that instead of considering oracle-aided MIP proof systems we consider oracle-aided MIP verifiers, and allow the honest provers to depend on any given oracle in an arbitrary non-black-box manner (thus our result rules out, in particular, oracle-aided proof systems). We refer the reader to Section 3 where we formally describe the proof systems we consider and the class of constructions to which our result applies.

Our generalized decision oracle. For any oracle $\Psi \in \mathfrak{S}$ our generalized decision oracle $\mathsf{Decide}_{\mathfrak{S}}^{\Psi}$ takes as input a triplet $(V, \overline{V}, x)$, where V and $\overline{V}$ are oracle-aided MIP-verifiers and $x \in \{0,1\}^n$. The oracle first checks whether or not the pair $(V, \overline{V})$ indeed consists of MIP-verifiers for a language and for its complement with respect to all oracles in $\mathfrak{S}$ as discussed above. If $(V, \overline{V})$ is not valid in this sense, then the oracle outputs $\perp$. If $(V, \overline{V})$ is valid, then the oracle outputs 1 if $x \in L^\Psi$ and 0 otherwise, where L^Ψ is the language defined by $(V^\Psi, \overline{V}^\Psi)$.

At this point, we would ideally like to follow the approach of Bitansky et al. in proving that Ψ realizes a one-way function and an indistinguishability obfuscator relative to our generalized decision oracle. Recall that their proof consists of a sequence of hybrid experiments, where the transition between each consecutive pair of experiments is enabled by the existence of a small set of critical queries. Specifically, in each transition they modify Ψ on some set of queries into an oracle Ψ', and argue that unless these queries fall into the small critical set then the decision oracle behaves exactly the same.

[3] For an oracle Ψ, an instance x, a string r, a polynomial-time oracle-aided verifier V, and provers $P_1, \ldots, P_N$ we denote by $\left\langle V^\Psi(x;r), P_1, \ldots, P_N \right\rangle$ the output of V with oracle access to Ψ on input x and randomness r in the multi-prover execution with $P_1, \ldots, P_N$. Note that whenever the provers are computationally unbounded we can assume that they are deterministic.

[4] It is usually assumed that the same provers are used for every $x \in \{0,1\}^n$, and that they obtain x as input. However, since the provers are computationally unbounded, our definition is clearly equivalent and easier to work with for our purposes.

Are there small and useful critical query sets? Fix an oracle $\Psi \in \mathfrak{S}$, and fix a query $(V, \overline{V}, x)$ to our generalized decision oracle, where V and $\overline{V}$ are valid MIP-verifiers in the above sense. Unlike the case of NP-verifiers, when considering MIP-verifiers then at a first glance there does not seem to be a small set of queries that completely determines whether or not $x \in L^{\Psi}$. Specifically, assuming for the current discussion that $x \in L^{\Psi}$, in the case of NP-verifiers this is completely determined by the polynomial number of queries to the oracle Ψ in the execution $V^{\Psi}(x, w)$ where w is any specific witness (say, the lexicographically first such witness). However, in the case of MIP-verifiers, we are guaranteed that there exist provers $P_1, \ldots, P_N$ that lead the MIP-verifier $V^{\Psi}(x; r)$ to accept with probability at least $2/3$ over the randomness $r \leftarrow \{0,1\}^{\mathsf{poly}(|x|)}$ of the verifier – but this guarantee involves potentially exponentially-many executions and thus exponentially-many queries to the oracle Ψ. It may even be the case that any oracle Ψ' that agrees with Ψ on all of these queries, is in fact $\Psi' = \Psi$, and this is not very useful for the purpose of transitioning between two hybrid experiments.

Nevertheless, let us consider an oracle Ψ' that differs from Ψ on a *single* query z, and now suppose that suddenly $x \notin L^{\Psi'}$ although we started with $x \in L^{\Psi}$. Thus, no provers can now lead $V^{\Psi'}(x; r)$ to accept with probability larger than $1/3$ over the randomness $r \leftarrow \{0,1\}^{\mathsf{poly}(|x|)}$, and in particular this holds for the above provers $P_1, \ldots, P_N$ that led $V^{\Psi}(x; r)$ to accept with probability at least $2/3$. The only way that $V^{\Psi'}(x; r)$ can differ from $V^{\Psi}(x; r)$ in an execution with the same $P_1, \ldots, P_N$ is by having $V^{\Psi}(x; r)$ query Ψ on z – and we can deduce that with probability at least $1/3$ over the choice of $r \leftarrow \{0,1\}^{\mathsf{poly}(|x|)}$ it holds that $V^{\Psi}(x; r)$ queries Ψ on z when interacting with $P_1, \ldots, P_N$.

Therefore, it is quite tempting to fix a distance parameter $d \geq 1$, and then for an oracle $\Psi \in \mathfrak{S}$ and a query $(V, \overline{V}, x)$ such that $x \in L^{\Psi}$ to define the following "d-influential set" of queries: Let $P_1, \ldots, P_N$ be provers that lead $V^{\Psi}(x; r)$ to accept with probability at least $2/3$, then the d-influential set consists of all queries that $V^{\Psi}(x; r)$ issues to Ψ in at least a $1/(3d)$-fraction of these executions. Then, if V issues at most q queries in each execution, then this set consists of at most $3qd$ queries. Moreover, for any oracle Ψ' that differs from Ψ on at most d queries, and these queries are not in the d-influential set, then it must hold that $x \in L^{\Psi'}$ (the probability that $V(x; r)$ accepts cannot drop from $2/3$ to $1/3$ when switching from Ψ to Ψ' since they differ on at most d queries and each of these queries cannot affect more than a $1/(3d)$-fraction of the executions).

From influential queries to influential labels. Unfortunately, this observation is still insufficient for our purposes. In the proof of Bitansky et al. the number of differences between Ψ and Ψ' is irrelevant as long as these differences are not in the critical set. However, in our case more than d differences outside of the d-influential set may still cause the verifier's acceptance probability to drop from $2/3$ to below $1/3$.

Although our proof considers oracles Ψ and Ψ' that may differ on an exponential number of queries, we tailor the specific structure of our obfuscator in a way that enables us to "group together" related queries: We introduce labeling functions (depending on the specific structure of our oracles) that assign a label to each query to the oracle Ψ, where different queries may share the same label. We show that it now suffices to focus on the small number $d \leq 3$ of labels that result from the potentially-exponential number of differences between the oracles Ψ and Ψ'.

Specifically, we prove that for any Ψ and for any query $(V, \overline{V}, x)$ to our generalized decision oracle there exists a small set $\mathbf{I}$ of "d-influential labels" such that any changes to Ψ involving at most d labels outside of $\mathbf{I}$ do not change the answer to the query. That is, let $\Psi' \in \mathfrak{G}$ be any oracle for which there exists a set $\mathcal{D} \subseteq \mathcal{X} \setminus \mathbf{I}$ of at most d labels such that if $\Psi'(\alpha) \neq \Psi(\alpha)$ then $\mathsf{lab}(\alpha) \in \mathcal{D}$, where $\mathcal{X}$ is the set of all possible labels and lab is a labeling function. Then, it holds that $\mathsf{Decide}_{\mathfrak{G}}^{\Psi'}(V, \overline{V}, x) = \mathsf{Decide}_{\mathfrak{G}}^{\Psi}(V, \overline{V}, x)$. This is a simplified description of the key property on which we rely in order to prove that a random instance of Ψ realizes a one-way function and an indistinguishability obfuscator relative to our generalized decision oracle, and we refer the reader to Section 4 for the proof of our impossibility result.

1.3 Paper Organization

The remainder of this paper is organized as follows. In Section 2 we introduce some standard notation as well as the cryptographic primitives under consideration in this paper. In Section 3 we define the class of constructions to which our impossibility result applies, and in Section 4 we formally state and prove Theorem 1.

2 Preliminaries

In this section we present the notation and basic definitions that are used in this work. For a distribution X we denote by $x \leftarrow X$ the process of sampling a value x from the distribution X. Similarly, for a set $\mathcal{X}$ we denote by $x \leftarrow \mathcal{X}$ the process of sampling a value x from the uniform distribution over $\mathcal{X}$. For an integer $n \in \mathbb{N}$ we denote by $[n]$ the set $\{1, \ldots, n\}$. For every $n \in \mathbb{N}$ and $m \geq n$ we denote by $\mathsf{InjFunc}_n^m$ the set of all injective functions $f : \{0,1\}^n \to \{0,1\}^m$.

Oracle-aided languages and complexity classes. For a language $L \subseteq \{0,1\}^*$, we let $\chi_L : \{0,1\}^* \to \{0,1\}$ denote the characteristic function of L, that is, $\chi_L(x) = 1$ if and only if $x \in L$. A deterministic algorithm A decides a language L if for every $x \in \{0,1\}^*$ it holds that $A(x) = \chi_L(x)$.

We consider the standard notions of languages and complexity classes when naturally generalized to oracle-aided computations. In particular, an oracle-aided language L defines a set $L^\Gamma \subseteq \{0,1\}^*$ for any possible oracle $\Gamma : \{0,1\}^* \to \{0,1\}^*$. Our definitions throughout the paper follow the standard approach that was introduced in the classic complexity-theory literature for proving separations between complexity classes by considering type-2 languages and complexity classes (see, for example, [7, 17] and the references therein).

Indistinguishability obfuscation for oracle-aided circuits. We consider the standard notion of indistinguishability obfuscation [6, 20] when naturally generalized to oracle-aided circuits (i.e., circuits that may contain oracle gates in addition to standard gates) [2, 3]. We first define the notion of functional equivalence relative to a specific function (provided as an oracle), and then we define the notion of an indistinguishability obfuscation for a class of oracle-aided circuits. In what follows, when considering a class $\mathcal{C} = \{\mathcal{C}_n\}_{n \in \mathbb{N}}$ of oracle-aided circuits, we assume that each $\mathcal{C}_n$ consists of circuits of size at most n.

▶ **Definition 2.** *Let C_0 and C_1 be two oracle-aided circuits, and let f be a function. We say that C_0 and C_1 are functionally equivalent relative to f, denoted $C_0^f \equiv C_1^f$, if for any input x it holds that $C_0^f(x) = C_1^f(x)$.*

▶ **Definition 3.** *A probabilistic polynomial-time oracle-aided algorithm $i\mathcal{O}$ is an indistinguishability obfuscator relative to an oracle Γ for a class $\mathcal{C} = \{\mathcal{C}_n\}_{n \in \mathbb{N}}$ of oracle-aided circuits if the following conditions are satisfied:*

- ***Functionality.*** *For all $n \in \mathbb{N}$ and for all $C \in \mathcal{C}_n$ it holds that*

$$\Pr\left[C^\Gamma \equiv \widehat{C}^\Gamma : \widehat{C} \leftarrow i\mathcal{O}^\Gamma(1^n, C)\right] = 1.$$

- ***Indistinguishability.*** *For any probabilistic polynomial-time oracle-aided distinguisher $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ there exists a negligible function $\nu(\cdot)$ such that*

$$\mathsf{Adv}^{\mathit{iO}}_{\Gamma,i\mathcal{O},\mathcal{A},\mathcal{C}}(n) \stackrel{\mathit{def}}{=} \left|\Pr\left[\mathsf{Exp}^{\mathit{iO}}_{\Gamma,i\mathcal{O},\mathcal{A},\mathcal{C}}(n) = 1\right] - \frac{1}{2}\right| \le \nu(n)$$

for all sufficiently large $n \in \mathbb{N}$, where the random variable $\mathsf{Exp}^{\mathit{iO}}_{\Gamma,i\mathcal{O},\mathcal{A},\mathcal{C}}(n)$ is defined via the following experiment:

1. *$b \leftarrow \{0,1\}$.*
2. *$(C_0, C_1, \mathsf{state}) \leftarrow \mathcal{A}_1^\Gamma(1^n)$ where $C_0, C_1 \in \mathcal{C}_n$ and $C_0^\Gamma \equiv C_1^\Gamma$.*
3. *$\widehat{C} \leftarrow i\mathcal{O}^\Gamma(1^n, C_b)$.*
4. *$b' \leftarrow \mathcal{A}_2^\Gamma(\mathsf{state}, \widehat{C})$.*
5. *If $b' = b$ then output 1, and otherwise output 0.*

For simplicity, note that whenever the algorithm $\mathcal{A}_1$ is deterministic there is in fact no need for $\mathcal{A}_1$ to transfer any state information state to $\mathcal{A}_2$ as the state can be reconstructed if needed by invoking $\mathcal{A}_1$. Looking ahead, in this paper we consider computationally-unbounded algorithms (i.e., we limit their query complexity but we do not limit their internal computation), and such algorithms can be assumed without loss of generality to be deterministic.

3 The Class of Constructions

The proof systems we consider in this paper can be formalized in a variety of seemingly equivalent manners, and here we choose a specific definition that we find to simplify the proof of our impossibility result:

▶ **Definition 4.** *For functions $V, P : \{0,1\}^* \to \{0,1\}^*$, an integer $k \ge 0$ and a string $s \in \{0,1\}^*$, we denote by $\langle V(s), P\rangle_k$ the output of the following computation:*

- *Let $m_0 = P(V(s,0))$.*
- *For $1 \le i < k$, let $m_i = P(V(s,i,m_0,\dots,m_{i-1}))$.*
- *Output $V(s,k,m_0,\dots,m_{k-1}) \in \{0,1\}$.*

That is, we consider a sequential process that is executed by two parties, a verifier V that is given as input a string s, and a prover P that is not given any input. The process consists of k rounds, where in each round the verifier sends the prover a message that is computed as a function of its input s, the index i of the current round, and the prover's previous responses $m_0,\dots,m_{i-1}$. In turn, the prover replies with a response m_i, and following these k steps the verifier outputs a bit indicating acceptance or rejection.

A crucial property to notice is that the prover's response, m_i, in each step is a function of the verifier's ith message only, and not of the entire transcript which includes all of the verifier's previous messages as well (i.e., the prover is "memoryless"). A verifier may potentially include the entire transcript in each message, and then the definition would collapse to the class IP of languages that have an interactive proof system [24].

In general, however, a verifier need not send the entire transcript in each message, and this enables us to capture the class MIP of languages that have a multi-prover interactive proof system [8]. Specifically, any such proof system $\langle V, P_1, \dots, P_N\rangle$ in which each prover sends at most v messages can be transformed in a black-box manner into a proof system $\langle V, P\rangle_k$ of the above form with $k = v \cdot N$. This can be done, for example, by defining $P(i, \cdot) = P_i(\cdot)$ for every $i \in [T]$ (with P maintaining the local state of each prover if needed), and having the verifier include in each message the index of the prover to which this message is sent together with the entire transcript that this specific prover has seen so far. Although we have not yet defined the completeness and soundness properties for the above proof systems (these are defined as part of the following definition), we already note that this transformation naturally preserves them.

As discussed in Section 1.2, instead of considering oracle-aided MIP proof systems we consider oracle-aided MIP verifiers, and allow the honest provers to depend on any given oracle in an arbitrary (i.e., non-black-box) manner (thus our result rules out, in particular, oracle-aided proof systems). This is captured via the following definition:

▶ **Definition 5.** *A pair $\left(V, \overline{V}\right)$ of oracle-aided polynomial-time algorithms, together with polynomials $\ell_r(\cdot)$ and $k(\cdot)$, define a* (MIP, coMIP) protocol pair relative to an oracle $\Psi : \{0,1\}^* \to \{0,1\}^*$ *if there exists a language $L^\Psi \subseteq \{0,1\}^*$ and such that:*

- *For every $x \in L^\Psi$ there exists a function $P : \{0,1\}^* \to \{0,1\}$ such that*

$$\Pr_{r \leftarrow \{0,1\}^{\ell_r(|x|)}}\left[\left\langle V^\Psi(x,r), P\right\rangle_{k(|x|)} = 1\right] \geq 2/3 ,$$

and for every function $\overline{P} : \{0,1\}^* \to \{0,1\}$ it holds that

$$\Pr_{r \leftarrow \{0,1\}^{\ell_r(|x|)}}\left[\left\langle \overline{V}^\Psi(x,r), \overline{P}\right\rangle_{k(|x|)} = 1\right] \leq 1/3 .$$

- *For every $x \notin L^\Psi$ there exists a function $\overline{P} : \{0,1\}^* \to \{0,1\}$ such that*

$$\Pr_{r \leftarrow \{0,1\}^{\ell_r(|x|)}}\left[\left\langle \overline{V}^\Psi(x,r), \overline{P}\right\rangle_{k(|x|)} = 1\right] \geq 2/3 ,$$

and for every function $P : \{0,1\}^* \to \{0,1\}$ it holds that

$$\Pr_{r \leftarrow \{0,1\}^{\ell_r(|x|)}}\left[\left\langle V^\Psi(x,r), P\right\rangle_{k(|x|)} = 1\right] \leq 1/3 .$$

Note that the above definition considers provers that output only a single bit in each step. This is just for syntactical reasons, making sure that the verifier runs in polynomial-time with respect to the length of the input x. For example, if the prover was allowed to be a length-doubling function, then after $|x|$ rounds this would allow a polynomial-time verifier to run in time that is exponential in the length of $|x|$. There are naturally various ways in which this technical issue can be handled (e.g., providing the verifier with oracle access to the prover instead of direct communication), clearly without having any effect on our result.

The following definition is based on those of [2, 3, 10] (which, in turn, are motivated by [29, 22, 31]), and captures the class of construction that we consider in this paper. We remind the reader that two oracle-aided circuits, C_0 and C_1, are functionally equivalent relative to a function f, denoted $C_0^f \equiv C_1^f$, if for any input x it holds that $C_0^f(x) = C_1^f(x)$ (see Definition 2).

▶ **Definition 6.** *A fully black-box construction of a worst-case hard* (MIP, coMIP) *protocol pair from an injective one-way function f and an indistinguishability obfuscator for the class $\mathcal{C}$ of all oracle-aided circuits C^f, consists of a pair of oracle-aided polynomial-time algorithms $(V, \overline{V})$, polynomials $\ell_r(\cdot)$ and $k(\cdot)$, an oracle-aided polynomial-time algorithm M, and "security loss" functions $\epsilon_{M,1}(\cdot)$ and $\epsilon_{M,2}(\cdot)$, such that the following conditions hold:*

- ***Correctness:*** *For every ensemble* $f = \{f_n : \{0,1\}^n \to \{0,1\}^{n+1}\}_{n\in\mathbb{N}}$ *of injective functions, and for any function* $i\mathcal{O}$ *such that* $i\mathcal{O}(C;r)^f \equiv C^f$ *for any circuit* C *and* $r \in \{0,1\}^*$*, the pair* $(V, \overline{V})$*, together with the polynomials* $\ell_r(\cdot)$ *and* $k(\cdot)$*, define an* $(\mathsf{MIP}, \mathsf{coMIP})$ *protocol pair (with a corresponding language* $L^{f,i\mathcal{O}}$*) relative to the oracle* $(f, i\mathcal{O})$*.*
- ***Black-box proof of hardness:*** *For every ensemble* $f = \{f_n : \{0,1\}^n \to \{0,1\}^{n+1}\}_{n\in\mathbb{N}}$ *of injective functions, for any function* $i\mathcal{O}$ *such that* $i\mathcal{O}(C;r)^f \equiv C^f$ *for any circuit* C *and* $r \in \{0,1\}^*$*, and for any oracle-aided algorithm* $\mathcal{A}$ *that runs in time* $T_{\mathcal{A}}(\cdot)$*, if* $\mathcal{A}^{f,i\mathcal{O}}(x) = \chi_{L^{f,i\mathcal{O}}}(x)$ *for every* $x \in \{0,1\}^*$ *then either*

  $$\Pr\left[M^{f,i\mathcal{O},\mathcal{A}}(f(x)) = x\right] \geq \epsilon_{M,1}(T_{\mathcal{A}}(n)) \cdot \epsilon_{M,2}(n)$$

  for infinitely many values of $n \in \mathbb{N}$*, where the probability is taken over the choice of* $x \leftarrow \{0,1\}^n$ *and over the internal randomness of* M*, or*

  $$\left|\Pr\left[\mathsf{Exp}^{\mathsf{iO}}_{(f,i\mathcal{O}),i\mathcal{O},M^{\mathcal{A}},\mathcal{C}}(n) = 1\right] - \frac{1}{2}\right| \geq \epsilon_{M,1}(T_{\mathcal{A}}(n)) \cdot \epsilon_{M,2}(n)$$

  for infinitely many values of $n \in \mathbb{N}$*.*

Intuitively, a black-box proof of hardness for $L^{f,i\mathcal{O}}$ means that any algorithm that decides $L^{f,i\mathcal{O}}$ can be used to construct an adversary that breaks either the one-wayness of f or the indistinguishability property of $i\mathcal{O}$ in a black-box way.

Note that restricting $\mathcal{A}$ to be deterministic and to decide the language in the worst case (i.e., on all inputs) only makes our result stronger. Also note that, following Asharov and Segev [2, 3], we split the security loss in the above definition to an adversary-dependent security loss (the function $\epsilon_{M,1}(\cdot)$) and an adversary-independent security loss (the function $\epsilon_{M,2}(\cdot)$), as this allows us to also rule out constructions in which one of these losses is super-polynomial while the other is polynomial.

4 Our Impossibility Result

Equipped with a formal definition of the class of constructions that we consider in this paper (recall Definition 6), in this section we prove the following theorem:

▶ **Theorem 7.** *Let* $((V, \overline{V}), \ell_r, k, M, T_M, \epsilon_{M,1}, \epsilon_{M,2})$ *be a fully black-box construction of a worst-case hard* $(\mathsf{MIP}, \mathsf{coMIP})$ *protocol pair from an injective one-way function* f *and an indistinguishability obfuscator for all oracle-aided circuits* C^f*. Then, it holds that*

$$\epsilon_{M,1}(n) \cdot \epsilon_{M,2}(n) \leq 2^{-\Omega(n)}.$$

That is, at least one out of the adversary-dependent security loss $\epsilon_{M,1}(\cdot)$ *and the adversary-independent security loss* $\epsilon_{M,2}(\cdot)$ *is exponential.*

Theorem 7 rules out, in particular, standard "polynomial-time polynomial-loss" reductions. More generally, the theorem implies that if the adversary-dependent security loss $\epsilon_{M,1}(\cdot)$ is polynomial (as is typically the case in cryptographic reductions), then the adversary-independent security loss $\epsilon_{M,2}(\cdot)$ must be exponential. Thus, this also rules out constructions that are based on indistinguishability obfuscation with sub-exponential security (e.g., [11, 12]).

In what follows we first introduce our generalized decision oracle, and capture its main property on which we rely in our proof, as discussed in Section 1.2. Then, in Section 4.2 we introduce the additional oracles on which we rely, and in Sections 4.3 and 4.4 we prove that relative to these oracles and to our decision oracle there exist an injective one-way function and an indistinguishability obfuscator, respectively. Finally, in Section 4.5 we derive the proof of Theorem 7.

4.1 Our Generalized Decision Oracle

For a family of oracles $\mathfrak{S}$ and for any specific oracle $\Psi \in \mathfrak{S}$, we define the oracle $\mathsf{Decide}_{\mathfrak{S}}^{\Psi}$ as the following function: Given as input tuple $(C_0, C_1, 1^{\ell_r}, 1^k)$, where C_0 and C_1 are oracle-aided circuits, and ℓ_r and k are non-negative integers, for every $\Phi \in \mathfrak{S}$ the oracle checks if exactly one of the following two cases holds:

- There exists a function $P_1 : \{0,1\}^* \to \{0,1\}$ such that
$$\Pr_{r \leftarrow \{0,1\}^{\ell_r}} \left[\left\langle C_1^{\Phi}(r), P_1 \right\rangle_k = 1 \right] \geq 2/3 ,$$
and for every function $P_0 : \{0,1\}^* \to \{0,1\}$ it holds that
$$\Pr_{r \leftarrow \{0,1\}^{\ell_r}} \left[\left\langle C_0^{\Phi}(r), P_0 \right\rangle_k = 1 \right] \leq 1/3 .$$
In this case, we say that $(C_0^{\Phi}, C_1^{\Phi}, 1^{\ell_r}, 1^k)$ is a *yes-instance*.
- There exists a function $P_0 : \{0,1\}^* \to \{0,1\}$ such that
$$\Pr_{r \leftarrow \{0,1\}^{\ell_r}} \left[\left\langle C_0^{\Phi}(r), P_0 \right\rangle_k = 1 \right] \geq 2/3 ,$$
and for every function $P_1 : \{0,1\}^* \to \{0,1\}$ it holds that
$$\Pr_{r \leftarrow \{0,1\}^{\ell_r}} \left[\left\langle C_1^{\Phi}(r), P_1 \right\rangle_k = 1 \right] \leq 1/3 .$$
In this case, we say that $(C_0^{\Phi}, C_1^{\Phi}, 1^{\ell_r}, 1^k)$ is a *no-instance*.

If there exists an oracle $\Phi \in \mathfrak{S}$ such that not exactly one of the above cases hold, then we say that the input $(C_0, C_1, 1^{\ell_r}, 1^k)$ is *invalid* and set $\mathsf{Decide}_{\mathfrak{S}}^{\Psi}$ to output $\perp$. Otherwise, $\mathsf{Decide}_{\mathfrak{S}}^{\Psi}$ outputs 1 or 0 according to whether $(C_0^{\Psi}, C_1^{\Psi}, 1^{\ell_r}, 1^k)$ is a yes-instance or a no-instance.[5]

The following simple lemma shows that the oracle $\mathsf{Decide}_{\mathfrak{S}}^{\Psi}$ can be easily used in order to decide any language that is defined via a $(\mathsf{MIP}, \mathsf{coMIP})$ protocol pair:

▶ **Lemma 8.** *Let $\mathfrak{S}$ be a family of oracles, and let $(V, \overline{V})$ be a pair of oracle-aided polynomial-time algorithms that is an $(\mathsf{MIP}, \mathsf{coMIP})$ protocol pair, with respect to polynomials $\ell_r(\cdot)$ and $k(\cdot)$, relative to every oracle $\Psi \in \mathfrak{S}$. Then, there exists a polynomial-time single-query algorithm $\mathcal{A}$ such that for every $\Psi \in \mathfrak{S}$, given oracle access to $\mathsf{Decide}_{\mathfrak{S}}^{\Psi}$ the algorithm $\mathcal{A}$ decides the language $L^{\Psi} \subseteq \{0,1\}^*$ defined by $(V, \overline{V}, \ell_r, k)$ relative to Ψ. That is, for every $\Psi \in \mathfrak{S}$ and $x \in \{0,1\}^*$ the algorithm $\mathcal{A}^{\mathsf{Decide}_{\mathfrak{S}}^{\Psi}}(x)$ outputs 1 if and only if $x \in L^{\Psi}$.*

Proof. Since V and $\overline{V}$ are polynomial time, there exists a polynomial $p(n)$ such that on input of size n their output is of size at most $p(n)$. Given $x \in \{0,1\}^*$ as input and oracle access to $\mathsf{Decide}_{\mathfrak{S}}^{\Psi}$, the algorithm $\mathcal{A}$ queries $\mathsf{Decide}_{\mathfrak{S}}^{\Psi}$ on $(C_0, C_1, 1^{\ell_r(|x|)}, 1^{k(|x|)})$, where C_0 and C_1 are the hardwired oracle-aided circuits $\overline{V}(x, \cdot)$ and $V(x, \cdot)$ respectively, the input size of both circuits is $\lceil \log(k(|x|) + 1) \rceil + k(|x|)$ (where $\lceil \log(k(|x|) + 1) \rceil$ bits are for the index of the communication round and $k(|x|)$ bits are for the messages of the prover) and the output size is $p(|x| + \lceil \log(k(|x|) + 1) \rceil + k(|x|))$. Finally, the algorithm $\mathcal{A}$ outputs 1 if and only if the oracle's response to the query is 1. ◀

[5] Note that for an input $(C_0, C_1, 1^{\ell_r}, 1^k)$, either it is invalid and then $\mathsf{Decide}_{\mathfrak{S}}^{\Psi}$ outputs $\perp$ for every $\Psi \in \mathfrak{S}$, or it is valid and then $\mathsf{Decide}_{\mathfrak{S}}^{\Psi}$ outputs 0 or 1 depending on Ψ.

The following lemma captures the key property of our oracle, as discussed in Section 1.2:

▶ **Lemma 9.** *Let $\mathfrak{S}$ be a family of oracles, let $\mathcal{Q}$ be the set of all possible queries for every oracle in the family, let* $\mathsf{lab} : \mathcal{Q} \to \mathcal{X}$ *be a "labeling" of the possible queries, and let $d \in \mathbb{N}$ be a parameter.*

For any $\Psi \in \mathfrak{S}$ and for any $\mathsf{Decide}^{\Psi}_{\mathfrak{S}}$*-query $(C_0, C_1, 1^{\ell_r}, 1^k)$ such that each of the circuits C_0 and C_1 contains at most q oracle gates, there exists a set of labels $\mathbf{I} = \mathbf{I}(\mathfrak{S},\Psi,C_0,C_1,\ell_r,k,\mathsf{lab},d) \subseteq \mathcal{X}$, which we call the* influential labels*, satisfying the following two properties:*

1. *The set is small: $|\mathbf{I}| \leq 3 \cdot q \cdot k \cdot d$.*
2. *Any changes to Ψ involving at most d labels outside of $\mathbf{I}$ do not change the answer of the query: Let $\Phi \in \mathfrak{S}$ be another oracle, such that there exists a set $\mathcal{D} \subseteq \mathcal{X} \setminus \mathbf{I}$ of labels with cardinality at most d such that if $\Phi(q) \neq \Psi(q)$ then $\mathsf{lab}(q) \in \mathcal{D}$. Then, it holds that*

$$\mathsf{Decide}^{\Phi}_{\mathfrak{S}}(C_0, C_1, 1^{\ell_r}, 1^k) = \mathsf{Decide}^{\Psi}_{\mathfrak{S}}(C_0, C_1, 1^{\ell_r}, 1^k) \ .$$

Proof. If $\mathsf{Decide}^{\Psi}_{\mathfrak{S}}(C_0, C_1, 1^{\ell_r}, 1^k) = \perp$ this means that the input $(C_0, C_1, 1^{\ell_r}, 1^k)$ is invalid, and then $\mathsf{Decide}^{\Phi}_{\mathfrak{S}}(C_0, C_1, 1^{\ell_r}, 1^k) = \perp$ holds for every $\Phi \in \mathfrak{S}$ and the claim follows for $\mathbf{I} = \emptyset$. Otherwise, suppose without loss of generality that $\mathsf{Decide}^{\Psi}_{\mathfrak{S}}(C_0, C_1, 1^{\ell_r}, 1^k) = 1$ and let $P_1 : \{0,1\}^* \to \{0,1\}$ such that

$$\Pr_{r \leftarrow \{0,1\}^{\ell_r}} \left[\left\langle C_1^{\Psi}(r), P_1 \right\rangle_k = 1 \right] \geq 2/3 \ .$$

Roughly speaking, we define $\mathbf{I} \subseteq \mathcal{X}$ to be the set of all labels for which a query with that label is performed during the execution of the protocol $\left\langle C_1^{\Psi}(\cdot), P_1 \right\rangle_k$ with high probability over the choice of r. More formally, we define

$$\mathbf{I} = \left\{ \mathsf{label} \in \mathcal{X} \middle| \Pr_{r \leftarrow \{0,1\}^{\ell_r}} \left[\begin{array}{c} \text{A query } q \in \mathcal{Q} \text{ such that } \mathsf{lab}(q) = \mathsf{label} \text{ is performed} \\ \text{during the execution of } \left\langle C_1^{\Psi}(r), P_1 \right\rangle_k \end{array} \right] \geq \frac{1}{3 \cdot d} \right\} \ .$$

First, for every $r \in \{0,1\}^{\ell_r}$ at most $q \cdot k$ queries are performed during the execution of $\left\langle C_1^{\Psi}(r), P_1 \right\rangle_k$. Therefore, for any $0 < \epsilon \leq 1$ there are at most $q \cdot k / \epsilon$ labels such that

$$\Pr_{r \leftarrow \{0,1\}^{\ell_r}} \left[\begin{array}{c} \text{A query } q \in \mathcal{Q} \text{ such that } \mathsf{lab}(q) = \mathsf{label} \text{ is performed} \\ \text{during the execution of } \left\langle C_1^{\Psi}(r), P_1 \right\rangle_k \end{array} \right] \geq \epsilon \ .$$

In our case, this means that $\mathbf{I} \leq q \cdot k \cdot 3 \cdot d$ as claimed.

Next, let $\Phi \in \mathfrak{S}$ such that there exists a set $\mathcal{D} \subseteq \mathcal{X} \setminus \mathbf{I}$ of labels with cardinality at most d such that if $\Phi(q) \neq \Psi(q)$ then $\mathsf{lab}(q) \in \mathcal{D}$. By a union bound it holds that

$$\Pr_{r \leftarrow \{0,1\}^{\ell_r}} \left[\begin{array}{c} \text{A query } q \in \mathcal{Q} \text{ such that } \mathsf{lab}(q) \in \mathcal{D} \text{ is performed} \\ \text{during the execution of } \left\langle C_1^{\Psi}(r), P_1 \right\rangle_k \end{array} \right] < \frac{|\mathcal{D}|}{3 \cdot d} \leq \frac{1}{3} \ .$$

If the above event does not occur then $\left\langle C_1^{\Phi}(r), P_1 \right\rangle_k = \left\langle C_1^{\Psi}(r), P_1 \right\rangle_k$. Hence,

$$\begin{aligned} &\Pr_{r \leftarrow \{0,1\}^{\ell_r}} \left[\left\langle C_1^{\Phi}(r), P_1 \right\rangle_k = 1 \right] \\ &\quad \geq \Pr_{r \leftarrow \{0,1\}^{\ell_r}} \left[\left\langle C_1^{\Psi}(r), P_1 \right\rangle_k = 1 \right] - \Pr_{r \leftarrow \{0,1\}^{\ell_r}} \left[\left\langle C_1^{\Phi}(r), P_1 \right\rangle_k \neq \left\langle C_1^{\Psi}(r), P_1 \right\rangle_k \right] \\ &\quad > \frac{2}{3} - \frac{1}{3} = \frac{1}{3} \ , \end{aligned}$$

so $(C_0^{\Phi}, C_1^{\Phi}, 1^{\ell_r}, 1^k)$ is not a no-instance. Since $\mathsf{Decide}^{\Psi}_{\mathfrak{S}}(C_0, C_1, 1^{\ell_r}, 1^k) \neq \perp$, $(C_0^{\Phi}, C_1^{\Phi}, 1^{\ell_r}, 1^k)$ must be a yes-instance and therefore $\mathsf{Decide}^{\Phi}_{\mathfrak{S}}(C_0, C_1, 1^{\ell_r}, 1^k) = 1 = \mathsf{Decide}^{\Psi}_{\mathfrak{S}}(C_0, C_1, 1^{\ell_r}, 1^k)$ as claimed. ◀

4.2 Our Indistinguishability Obfuscation Oracle

In what follows we define the family $\mathfrak{S}$ of oracles that realize injective functions and strongly-unambiguous obfuscations relative to our decision oracle, and define a distribution $\mathcal{D}(\mathfrak{S})$ over that family. The family $\mathfrak{S}$ consists of all triplets $(f, \mathcal{O}, E) = (\{f_n\}_{n\in\mathbb{N}}, \{\mathcal{O}_n\}_{n\in\mathbb{N}}, \{E_n\}_{n\in\mathbb{N}})$, satisfying the following three conditions for every $n \in \mathbb{N}$:

1. The function $f_n : \{0,1\}^n \to \{0,1\}^{n+1}$ is injective. Looking ahead, f will serve as an injective one-way function.
2. The function $\mathcal{O}_n : \{0,1\}^{2n} \to \{0,1\}^{10n}$ is injective. Looking ahead, for an oracle-aided circuit $C \in \{0,1\}^n$ with f-gates and randomness $r \in \{0,1\}^n$, the output $\mathcal{O}_n(C,r)$ will serve as an obfuscation of C, and the restriction that $\mathcal{O}_n$ is injective means that the obfuscation is *strongly-unambiguous* in the sense that any obfuscation $\widehat{C} \in \mathrm{Image}(\mathcal{O}_n)$ only comes from a single circuit with a single randomness string.
3. The function $E_n : \{0,1\}^{11n} \to \{0,1\}^n$ satisfies the following condition: For every oracle-aided circuit $C \in \{0,1\}^n$ with f-gates, every randomness $r \in \{0,1\}^n$ and every input $\alpha \in \{0,1\}^n$, it holds that $E_n(\mathcal{O}_n(C,r), \alpha) = C^f(x)$. Namely, given an obfuscation $\widehat{C} = \mathcal{O}_n(C,r)$ and an input α, the function E_n evaluates C on input α with respect to the oracle f.
   We emphasize that for any $\widehat{C} \in \{0,1\}^{10n} \setminus \mathrm{Image}(\mathcal{O}_n)$, there is no restriction on $E_n(\widehat{C}, \cdot)$, so there is no clear way to verify whether some $\widehat{C} \in \{0,1\}^{10n}$ is a valid obfuscation. As noted by Bitansky et al. [10], it is necessary for the obfuscation to not be verifiable since an unambiguous and verifiable indistinguishability obfuscator does imply hardness in $\mathsf{NP} \cap \mathsf{coNP}$.

Now, we define a distribution $\mathcal{D}(\mathfrak{S})$ over $\mathfrak{S}$, relative to which we prove that an oracle $\Psi \leftarrow \mathcal{D}(\mathfrak{S})$ realizes an injective one-way function and an indistinguishability obfuscator. The distribution $\mathcal{D}(\mathfrak{S})$ is obtained by sampling a triplet $(f, \mathcal{O}, \mathsf{Eval}^{f,\mathcal{O}})$ from $\mathfrak{S}$ as follows:

1. For every $n \in \mathbb{N}$ the function f_n is uniformly chosen from the set $\mathsf{InjFunc}_n^{n+1}$ of all injective functions $f_n : \{0,1\}^n \to \{0,1\}^{n+1}$.
2. For every $n \in \mathbb{N}$ the function $\mathcal{O}_n : \{0,1\}^{2n} \to \{0,1\}^{10n}$ is sampled as follows: A function h is uniformly chosen from the set $\mathsf{InjFunc}_n^{5n}$, and for every $r \in \{0,1\}^n$ a function g_r is uniformly chosen from the set $\mathsf{InjFunc}_n^{5n}$. Then, for every input $(C,r) \in \{0,1\}^n \times \{0,1\}^n$ we define $\mathcal{O}_n(C,r) = (h(r), g_r(C))$. Note that $\mathcal{O}_n$ is injective as required, and that this distribution of the function $\mathcal{O}$ differs from that of Asharov and Segev [2] and Bitansky et al. [10], where $\mathcal{O}_n$ was a uniformly chosen injective function.
3. For every $n \in \mathbb{N}$, the function $\mathsf{Eval}^{f,\mathcal{O}}$ on input $(\widehat{C}, \alpha) \in \{0,1\}^{10n} \times \{0,1\}^n$ is defined as follows: If there exists a pair $(C,r) \in \{0,1\}^n \times \{0,1\}^n$ such that $\widehat{C} = \mathcal{O}_n(C,r)$ then it outputs $C^f(\alpha)$, and otherwise it outputs $\perp$. Note that $\mathsf{Eval}^{f,\mathcal{O}}$ satisfies the above third condition for membership in $\mathfrak{S}$.

4.3 The Existence of an Injective One-Way Function

In this section we prove that the injective function f is one way relative to $(\Psi, \mathsf{Decide}_{\mathfrak{S}}^{\Psi})$, where $\Psi = (f, \mathcal{O}, \mathsf{Eval}^{f,\mathcal{O}})$ is sampled from the distribution $\mathcal{D}(\mathfrak{S})$ over $\mathfrak{S}$ (see Section 4.2 for the description of this distribution). Our proof follows the structure of that of Bitansky, Degwekar and Vaikuntanathan [10], while strengthened to deal with our generalized decision oracle as explained in Section 1.2.

In what follows we call an oracle-aided algorithm $\mathcal{A}$ a *q-query algorithm*, for a function $q = q(n)$, if when given any input $x \in \{0,1\}^n$ it issues at most $q(n)$ queries to the oracle

Γ, each of its queries to Eval and Decide consists of circuits with at most $q(n)$ oracle gates, and the number of communication rounds in the proof systems corresponding to each of its queries to Decide is at most $q(n)$.

▶ **Theorem 10.** *For any oracle-aided $2^{n/12}$-query algorithm $\mathcal{A}$ it holds that*

$$\Pr_{\substack{\Psi \leftarrow \mathcal{D}(\mathfrak{S}) \\ x \leftarrow \{0,1\}^n}} \left[\mathcal{A}^{\Psi, \mathsf{Decide}^{\Psi}_{\mathfrak{S}}}(f(x)) = x \right] \leq O(2^{-n/2})$$

for all sufficiently large $n \in \mathbb{N}$.

In what follows, we let $\mathfrak{F}$ denote the family of ensembles $f = \{f_n\}_{n \in \mathbb{N}}$ where $f_n \in \mathsf{InjFunc}_n^{n+1}$ for all $n \in \mathbb{N}$. As our first step, we prove that $f \leftarrow \mathfrak{F}$ is one way relative to the oracle $(f, \mathsf{Decide}^{f}_{\mathfrak{F}})$.

▶ **Lemma 11.** *For any oracle-aided $2^{n/6}$-query algorithm $\mathcal{A}$ it holds that*

$$\Pr_{\substack{f \leftarrow \mathfrak{F} \\ x \leftarrow \{0,1\}^n}} \left[\mathcal{A}^{f, \mathsf{Decide}^{f}_{\mathfrak{F}}}(f(x)) = x \right] \leq O(2^{-n/2}) \ .$$

Proof. We prove that the lemma holds when even fixing the oracles $f_{-n} = \{f_k\}_{k \neq n}$ and only sampling f_n. We introduce a sequence of three hybrid experiments such that the first hybrid experiment $\mathcal{H}_1$ is the real one-wayness experiment and the last hybrid experiment $\mathcal{H}_3$ is an experiment in which the probability of the adversary is of winning is $1/2^n$. Then, by upper bounding the difference in the winning probability between each pair of consecutive hybrid experiments we deduce our claim.

The hybrid $\mathcal{H}_1$. This is the real experiment in which we sample $x \leftarrow \{0,1\}^n$, give $f_n(x) \in \{0,1\}^{n+1}$ to $\mathcal{A}$ as input, and give $\mathcal{A}$ oracle access to $\Gamma = (f, \mathsf{Decide}^{f}_{\mathfrak{F}})$.

The hybrid $\mathcal{H}_2$. In this experiment, we sample $y \leftarrow \{0,1\}^{n+1} \setminus \mathrm{Image}(f_n)$, give y to $\mathcal{A}$ as input, and give $\mathcal{A}$ oracle access to $\Gamma' = (f_{x \mapsto y}, \mathsf{Decide}^{f_{x \mapsto y}}_{\mathfrak{F}})$, where $f_{x \mapsto y}$ is defined as

$$f_{x \mapsto y}(z) = \begin{cases} y & \text{if } z = x \\ f(z) & \text{otherwise} \end{cases} .$$

That is, we "plant" y as the challenge and as the image of x.

The hybrid $\mathcal{H}_3$. This experiment is obtained from $\mathcal{H}_2$ by giving $\mathcal{A}$ oracle access to the original oracle Γ instead of the oracle Γ' with the planted y, while still giving $\mathcal{A}$ the planted y as input.

The following table summarizes our hybrid experiments:

Hybrid	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{H}_3$
Challenger Randomness	$x \leftarrow \{0,1\}^n$		
Injective Function	$f_n \leftarrow \mathsf{InjFunc}_n^{n+1}$		
Challenge	$f_n(x)$	$y \leftarrow \{0,1\}^{n+1} \setminus \mathrm{Image}(f_n)$	
Oracle	$\Gamma = (f, \mathsf{Decide}^{f}_{\mathfrak{F}})$	$\Gamma' = (f_{x \mapsto y}, \mathsf{Decide}^{f_{x \mapsto y}}_{\mathfrak{F}})$	$\Gamma = (f, \mathsf{Decide}^{f}_{\mathfrak{F}})$
Winning Condition	$\mathcal{A}$ outputs x		

▷ Claim 12. $\Pr[\mathcal{A} \textit{ wins in } \mathcal{H}_1] = \Pr[\mathcal{A} \textit{ wins in } \mathcal{H}_2]$.

Proof. We couple the experiments $\mathcal{H}_1$ and $\mathcal{H}_2$ as follows.[6] First, we sample the same $x \leftarrow \{0,1\}^n$ for both experiments. Then, we a uniformly sample a random injective function $\widehat{f} : \{0,1\}^n \setminus \{x\} \to \{0,1\}^{n+1}$. Next, we sample two distinct $y, y' \leftarrow \{0,1\} \setminus \text{Image}(\widehat{f})$. Now, in $\mathcal{H}_1$ we let the injective function be

$$f_n(z) = \begin{cases} y & \text{if } z = x \\ \widehat{f}(z) & \text{otherwise} \end{cases} ,$$

whereas in $\mathcal{H}_2$ we let the injective function be

$$f'_n(z) = \begin{cases} y' & \text{if } z = x \\ \widehat{f}(z) & \text{otherwise} \end{cases} ,$$

and let y be the planted challenge. It is easy to see that the marginal distribution in both experiments is correct, and that both experiments are identical. That is, $\mathcal{A}$ gets the same challenge as input and gets access to the same oracle, thus the claim follows. ◁

▷ Claim 13. $|\Pr[\mathcal{A} \textit{ wins in } \mathcal{H}_2] - \Pr[\mathcal{A} \textit{ wins in } \mathcal{H}_3]| \leq 3 \cdot q^3/2^n$.

Proof. We observe that the view of $\mathcal{A}$ in $\mathcal{H}_3$ is independent of the choice of x. Therefore, if a query to f_n is made, then the probability of it to be x is at most $1/2^n$. In any other case, the answer to this query is the same in $\mathcal{H}_2$ and $\mathcal{H}_3$, and both executions proceed the same way.

Now, if a query $(C_0, C_1, 1^{\ell_r}, 1^k)$ to $\mathsf{Decide}^f_{\mathfrak{F}}$ is made, then we apply Lemma 9. We take the label function $\mathsf{lab} : \mathcal{Q} \to \mathcal{X}$ to be the identity function. The set $\mathbf{I} = \mathbf{I}(\mathfrak{F}, f, C_0, C_1, \ell_r, k, \mathsf{lab}, 1) \subseteq \mathcal{X}$ of influential labels is independent of the choice of x. Therefore, the probably that $\mathbf{I}$ contains x is at most $|I|/2^n \leq 3q^2/2^n$. In any other case, the oracle $f_{x \mapsto y}$ of $\mathcal{H}_2$ is obtained from f of $\mathcal{H}_3$ by changes involving one label outside of $\mathbf{I}$, and therefore by Lemma 9 it holds that

$$\mathsf{Decide}^{f_{x \mapsto y}}_{\mathfrak{F}}(C_0, C_1, 1^{\ell_k}, 1^k) = \mathsf{Decide}^f_{\mathfrak{F}}(C_0, C_1, 1^{\ell_k}, 1^k) ,$$

and both executions proceed the same way. Applying a union bound we deduce that

$$\begin{aligned} &|\Pr[\mathcal{A} \text{ wins in } \mathcal{H}_2] - \Pr[\mathcal{A} \text{ wins in } \mathcal{H}_3]| \\ &\qquad \leq \Pr\left[\text{A query was answered differently in } \mathcal{A}^{\Gamma}(y) \text{ and } \mathcal{A}^{\Gamma'}(y)\right] \\ &\qquad \leq \frac{3q^3}{2^n} , \end{aligned}$$

and the claim follows. ◁

▷ Claim 14. $\Pr[\mathcal{A} \textit{ wins in } \mathcal{H}_3] = 1/2^n$.

Proof. In this experiment the view of $\mathcal{A}$ is independent of x. ◁

[6] To couple two probability distributions means to define a joint distribution whose marginals are exactly those two distributions.

Now we turn back to proving Lemma 11. It holds that

$$\begin{aligned}\Pr[\mathcal{A}\text{ wins in }\mathcal{H}_1] &\leq |\Pr[\mathcal{A}\text{ wins in }\mathcal{H}_1]-\Pr[\mathcal{A}\text{ wins in }\mathcal{H}_2]| \\ &\quad+|\Pr[\mathcal{A}\text{ wins in }\mathcal{H}_2]-\Pr[\mathcal{A}\text{ wins in }\mathcal{H}_3]|+\Pr[\mathcal{A}\text{ wins in }\mathcal{H}_3] \\ &\leq 0+\frac{3q^3}{2^n}+\frac{1}{2^n}=O\left(\frac{q^3}{2^n}\right),\end{aligned}$$

and by plugging $q=2^{n/6}$ we obtain Lemma 11. ◀

In the full version of this paper [35], we show how to deduce Theorem 10 from Lemma 11.

4.4 The Existence of an Indistinguishability Obfuscator

In this section we prove that relative to the oracle $\Gamma=(\Psi,\mathsf{Decide}^{\Psi}_{\mathfrak{S}})$, where $\Psi=(f,\mathcal{O},\mathsf{Eval}^{f,\mathcal{O}})$ is sampled from the distribution $\mathcal{D}(\mathfrak{S})$ defined in Section 4.2, there exists an indistinguishability obfuscator $i\mathcal{O}$ for all circuits with f-gates.

Our obfuscator is based on those of Asharov and Segev [2] and Bitansky et al. [10] but has a somewhat different structure. Similarly to their obfuscator, for every $n\in\mathbb{N}$, given an oracle-aided circuit $C\in\{0,1\}^n$, the obfuscator $i\mathcal{O}$ samples $r\leftarrow\{0,1\}^n$ and outputs the obfuscated circuit $\widehat{C}=\mathcal{O}_n(C,r)\in\{0,1\}^{10n}$. In turn, the oracle $\mathsf{Eval}^{f,\mathcal{O}}$ can be used for evaluating such an obfuscated circuit at any given point α: If there exists a pair $(C,r)\in\{0,1\}^n\times\{0,1\}^n$ such that $\widehat{C}=\mathcal{O}_n(C,r)$ then $\mathsf{Eval}^{f,\mathcal{O}}$ outputs $C^f(\alpha)$ and otherwise it outputs $\perp$.

However, unlike their obfuscator of Asharov and Segev [2] and Bitansky et al. [10], which was sampled uniformly at random among all injective functions (of the appropriate input and output lengths), recall that according to our definition of the distribution $\mathcal{D}(\mathfrak{S})$ it holds that $\mathcal{O}_n(C,r)=(h(r),g_r(C))$, where the function h is uniformly-chosen from the set $\mathsf{InjFunc}^{5n}_n$, and for every $r\in\{0,1\}^n$ a function g_r is uniformly-chosen from the set $\mathsf{InjFunc}^{5n}_n$.

Recall that we call an oracle-aided algorithm $\mathcal{A}$ a *q-query algorithm*, for a function $q=q(n)$, if when given any input $x\in\{0,1\}^n$ it issues at most $q(n)$ queries to the oracle Γ, each of its queries to Eval and Decide consists of circuits with at most $q(n)$ oracle gates, and the number of communication rounds in the proof systems corresponding to each of its queries to Decide is at most $q(n)$. Letting $\mathcal{C}$ denote the class of all oracle-aided circuit with f-gates, we prove the following theorem:

▶ **Theorem 15.** *For any oracle-aided $2^{n/6}$-query algorithm $\mathcal{A}$ it holds that*

$$\mathop{\mathbb{E}}_{\Gamma}\left|\Pr\left[\mathsf{Exp}^{i\mathcal{O}}_{\Gamma,i\mathcal{O},\mathcal{A},\mathcal{C}}(n)=1\right]-\frac{1}{2}\right|\leq O(2^{-n/4})$$

where the expectation is taken over the choice of $\Gamma=(\Psi,\mathsf{Decide}^{\Psi}_{\mathfrak{S}})$ where $\Psi\leftarrow\mathcal{D}(\mathfrak{S})$, and the inner probability is taken over the randomness of the experiment $\mathsf{Exp}^{i\mathcal{O}}_{\Gamma,i\mathcal{O},\mathcal{A},\mathcal{C}}(n)$.

Toward proving Theorem 15, we first prove the following lemma.

▶ **Lemma 16.** *For any oracle-aided $4\cdot2^{n/6}$-query algorithm $\mathcal{A}$ it holds that*

$$\left|\Pr\left[\mathsf{Exp}^{i\mathcal{O}}_{\Gamma,i\mathcal{O},\mathcal{A},\mathcal{C}}(n)=1\right]-\frac{1}{2}\right|\leq O(2^{-n/2})$$

where the probability is taken both over the choice of $\Gamma=(\Psi,\mathsf{Decide}^{\Psi}_{\mathfrak{S}})$ where $\Psi\leftarrow\mathcal{D}(\mathfrak{S})$, and over the randomness of the experiment $\mathsf{Exp}^{i\mathcal{O}}_{\Gamma,i\mathcal{O},\mathcal{A},\mathcal{C}}(n)$.

Proof. We prove that the lemma holds when even fixing the oracle f and $\mathcal{O}_{-n} = \{\mathcal{O}_k\}_{k \neq n}$, and only sampling $\mathcal{O}_n$. We introduce a sequence of 5 hybrid experiments such that the first hybrid experiment $\mathcal{H}_1$ is the real indistinguishability-obfuscation experiment $\mathsf{Exp}^{\mathsf{iO}}_{\Gamma,i\mathcal{O},\mathcal{A},\mathcal{C}}(n)$ and the last hybrid experiment $\mathcal{H}_5$ is an experiment in which the advantage of the adversary is 0. Then, by upper bounding the difference in the advantage between each pair of consecutive hybrid experiments we deduce our lemma.

In what follows we first describe the hybrid experiments (see also the table below for a summary – where we omit the function f since it has been fixed), and then present a sequence of claims for bounding the differences in the advantages.

The hybrid $\mathcal{H}_1$. This is the real experiment in which we sample $\mathcal{O}_n$ by sampling $h \leftarrow \mathsf{InjFunc}^{5n}_n$, sampling $g_r \leftarrow \mathsf{InjFunc}^{5n}_n$ for every $r \in \{0,1\}^n$, and setting $\mathcal{O}_n(C, r) = (h(r), g_r(C))$.

The hybrid $\mathcal{H}_2$. In this experiment, instead of giving the pre-challenge adversary $\mathcal{A}_0$ access to the oracle $\Gamma = (\Psi, \mathsf{Decide}^{\Psi}_{\mathfrak{G}})$ where $\Psi = (f, \mathcal{O}, \mathsf{Eval}^{\mathcal{O}})$, we sample a string $\widehat{h} \leftarrow \{0,1\}^{5n} \setminus \mathrm{Image}(h)$ and a function $\widehat{g} \leftarrow \mathsf{InjFunc}^{5n}_n$, then we give $\mathcal{A}_0$ access to the oracle $\Gamma' = (\Psi', \mathsf{Decide}^{\Psi'}_{\mathfrak{G}})$ where $\Psi' = (f, \mathcal{O}_{(\cdot,r^*)\to(\widehat{h},\widehat{g}(\cdot))}, \mathsf{Eval}^{\mathcal{O}})$ and for every $C, r \in \{0,1\}^n$ we define

$$\mathcal{O}_{(\cdot,r^*)\to(\widehat{h},\widehat{g}(\cdot))}(C, r) = \begin{cases} (\widehat{h}, \widehat{g}(C)) & \text{if } r = r^* \\ \mathcal{O}(C, r) & \text{otherwise} \end{cases}.$$

That is, for the challenge randomness r^*, instead of obfuscating using $h(r^*)$ and $g_{r^*}(\cdot)$ we use our "planted obfuscation" $\widehat{h}$ and $\widehat{g}(\cdot)$. The rest of the experiment proceeds as before.

The hybrid $\mathcal{H}_3$. In this experiment, we return to giving the pre-challenge adversary $\mathcal{A}_0$ access to the real oracle Γ. However, we now give the post-challenge adversary $\mathcal{A}_1$ a "planted challenge" $(\widehat{h}, \widehat{g}(C_b))$, and we give $\mathcal{A}_1$ access to the oracle $\Gamma' = (\Psi', \mathsf{Decide}^{\Psi'}_{\mathfrak{G}})$ where $\Psi' = (f, \mathcal{O}_{(\cdot,r^*)\to(\widehat{h},\widehat{g}(\cdot))}, \mathsf{Eval}^{\mathcal{O}})$.

The hybrid $\mathcal{H}_4$. For an obfuscator function of the form $\mathcal{O}(C, r) = (h(r), g_r(C))$, $\widehat{h} \in \{0,1\}^{5n} \setminus \mathrm{Image}(h)$ and $\widehat{g} \in \mathsf{InjFunc}^{5n}_n$, we define the planted evaluation function $\mathsf{PEval}^{\mathcal{O}}_{(\widehat{h},\widehat{g})}$ as

$$\mathsf{PEval}^{\mathcal{O}}_{(\widehat{h},\widehat{g})}(\widetilde{C}, \alpha) = \begin{cases} C^f(\alpha) & \text{if } \widetilde{C} = (\widehat{h}, \widehat{g}(C)) \text{ for a circuit } C \in \{0,1\}^n \\ \mathsf{Eval}^{\mathcal{O}}(\widetilde{C}, \alpha) & \text{otherwise} \end{cases}.$$

Note that since $\widehat{h} \notin \mathrm{Image}(h)$, it holds that $\mathsf{PEval}^{\mathcal{O}}_{(\widehat{h},\widehat{g})}$ is a valid evaluation function and therefore $(f, \mathcal{O}, \mathsf{PEval}^{\mathcal{O}}_{(\widehat{h},\widehat{g})}) \in \mathfrak{G}$. Now, the experiment $\mathcal{H}_4$ is obtained from $\mathcal{H}_3$ by replacing the post-challenge oracle Γ' with the oracle $\Gamma'' = (\Psi'', \mathsf{Decide}^{\Psi''}_{\mathfrak{G}})$ where $\Psi'' = (f, \mathcal{O}, \mathsf{PEval}^{\mathcal{O}}_{(\widehat{h},\widehat{g})})$. Note that in this experiment, the randomness r^* has no role.

The hybrid $\mathcal{H}_5$. This experiment is obtained from $\mathcal{H}_4$ by replacing the challenge obfuscation $(\widehat{h}, \widehat{g}(C_b))$ with $(\widehat{h}, \widehat{g}(C_0))$. Note that in this experiment, the bit b has no role except for the winning condition, namely, $\mathcal{A}$ wins if $\mathcal{A}_1$ outputs b.

<table>
<tr><th>Hybrid</th><th>$\mathcal{H}_1$</th><th>$\mathcal{H}_2$</th><th>$\mathcal{H}_3$</th><th>$\mathcal{H}_4$</th><th>$\mathcal{H}_5$</th></tr>
<tr><td>Challenger Randomness</td><td colspan="3">$b \leftarrow \{0,1\}$, $r^* \leftarrow \{0,1\}^n$</td><td colspan="2">$b \leftarrow \{0,1\}$</td></tr>
<tr><td>Obfuscator Fuction</td><td colspan="5">$\mathcal{O}_n(C,r) = (h(r), g_r(C))$, where $h \leftarrow \mathsf{InjFunc}_n^{5n}$ and $g_r \leftarrow \mathsf{InjFunc}_n^{5n}$ for every $r \in \{0,1\}^n$</td></tr>
<tr><td>Planted Obfuscation</td><td>N/A</td><td colspan="4">$\widehat{h} \leftarrow \{0,1\}^{5n} \setminus \text{Image}(h)$, $\widehat{g} \leftarrow \mathsf{InjFunc}_n^{5n}$</td></tr>
<tr><td>Pre-challenge Oracle</td><td>$\Psi = (\mathcal{O}, \mathsf{Eval}^{\mathcal{O}})$ $\mathsf{Decide}_{\mathfrak{S}}^{\Psi}$</td><td>$\Psi' = (\mathcal{O}' = \mathcal{O}_{(\cdot,r^*)\to(\widehat{h},\widehat{g}(\cdot))},$ $\mathsf{Eval}^{\mathcal{O}'}), \mathsf{Decide}_{\mathfrak{S}}^{\Psi'}$</td><td colspan="3">$\Psi = (\mathcal{O}, \mathsf{Eval}^{\mathcal{O}})$ $\mathsf{Decide}_{\mathfrak{S}}^{\Psi}$</td></tr>
<tr><td>Challenge Obfuscation</td><td colspan="2">$\mathcal{O}(C_b, r^*) = (h(r^*), g_{r^*}(C_b))$</td><td colspan="2">$(\widehat{h}, \widehat{g}(C_b))$</td><td>$(\widehat{h}, \widehat{g}(C_0))$</td></tr>
<tr><td>Post-challenge Oracle</td><td colspan="2">$\Psi = (\mathcal{O}, \mathsf{Eval}^{\mathcal{O}})$ $\mathsf{Decide}_{\mathfrak{S}}^{\Psi}$</td><td>$\Psi' = (\mathcal{O}' = \mathcal{O}_{(\cdot,r^*)\to(\widehat{h},\widehat{g}(\cdot))},$ $\mathsf{Eval}^{\mathcal{O}'}), \mathsf{Decide}_{\mathfrak{S}}^{\Psi'}$</td><td colspan="2">$\Psi'' = (\mathcal{O}, \mathsf{PEval}^{\mathcal{O}}_{(\widehat{h},\widehat{g})})$ $\mathsf{Decide}_{\mathfrak{S}}^{\Psi''}$</td></tr>
<tr><td>Winning Condition</td><td colspan="5">$\mathcal{A}_1$ outputs b</td></tr>
</table>

In the full version of this paper [35], we prove the following claims.

▷ Claim 17. $|\Pr[\mathcal{A} \textit{ wins in } \mathcal{H}_1] - \Pr[\mathcal{A} \textit{ wins in } \mathcal{H}_2]| \leq 27q^3/2^n$.

▷ Claim 18. $\Pr[\mathcal{A} \textit{ wins in } \mathcal{H}_2] = \Pr[\mathcal{A} \textit{ wins in } \mathcal{H}_3]$.

▷ Claim 19. $|\Pr[\mathcal{A} \textit{ wins in } \mathcal{H}_3] - \Pr[\mathcal{A} \textit{ wins in } \mathcal{H}_4]| \leq 12q^3/2^n$.

▷ Claim 20. $\Pr[\mathcal{A} \textit{ wins in } \mathcal{H}_4] = \Pr[\mathcal{A} \textit{ wins in } \mathcal{H}_5]$.

▷ Claim 21. $\Pr[\mathcal{A} \textit{ wins in } \mathcal{H}_5] = 1/2$.

Using the above claims, we can prove Lemma 16. It holds that

$$\begin{aligned}\left|\Pr[\mathcal{A} \text{ wins in } \mathcal{H}_1] - \tfrac{1}{2}\right| &= |\Pr[\mathcal{A} \text{ wins in } \mathcal{H}_1] - \Pr[\mathcal{A} \text{ wins in } \mathcal{H}_5]| \\ &\leq \sum_{i=1}^{4} |\Pr[\mathcal{A} \text{ wins in } \mathcal{H}_i] - \Pr[\mathcal{A} \text{ wins in } \mathcal{H}_{i+1}]| \leq \frac{40q^3}{2^n} ,\end{aligned}$$

and by plugging $q = 4 \cdot 2^{n/6}$ we obtain Lemma 16. ◀

In the full version of this paper [35], we show how to deduce Theorem 15 from Lemma 16.

4.5 Putting it All Together

Given Theorems 10 and 15 we can now derive Theorem 7.

Proof. Let $((V, \overline{V}), \ell_r, k, M, T_M, \epsilon_{M,1}, \epsilon_{M,2})$ be a fully black-box construction of a worst-case hard $(\mathsf{MIP}, \mathsf{coMIP})$ protocol pair from an injective one-way function f and an indistinguishability obfuscator for all oracle-aided circuits C^f. Lemma 8 guarantees the existence of a polynomial-time single-query algorithm $\mathcal{A}$ such that for every $\Psi = (f, \mathcal{O}, \mathsf{Eval}^{f,\mathcal{O}})$ in the support of the distribution $\mathcal{D}(\mathfrak{S})$, the algorithm $\mathcal{A}$ with oracle access to $\Gamma = (\Psi, \mathsf{Decide}_{\mathfrak{S}}^{\Psi})$ decides the language $L^{\Psi} \subseteq \{0,1\}^*$ defined by $(V, \overline{V}, \ell_r, k)$ relative to Ψ. That is, for every $x \in \{0,1\}^*$ it holds that $\mathcal{A}^{\Gamma}(x) = \chi_{L^{\Psi}}(x)$. For every $n \in \mathbb{N}$, denote by $T_{\mathcal{A}}(n)$ the polynomial running time of $\mathcal{A}$ on inputs of length n.

Definition 6 then states that there are two possible cases to consider: $\mathcal{A}$ can be used either for inverting the injective one-way function f, or for breaking the security of the indistinguishability obfuscator $i\mathcal{O}$. Specifically, in the first case we obtain from Definition 6 that for every $\Psi = (f, \mathcal{O}, \mathsf{Eval}^{f,\mathcal{O}})$ in the support of the distribution $\mathcal{D}(\mathfrak{S})$ it holds that

$$\Pr\left[M^{\Gamma,\mathcal{A}}\left(f(x)\right) = x\right] \geq \epsilon_{M,1}\left(T_{\mathcal{A}}(n)\right) \cdot \epsilon_{M,2}(n)$$

for infinitely many values of $n \in \mathbb{N}$, where $\Gamma = (\Psi, \mathsf{Decide}_{\mathfrak{S}}^{\Psi})$ and the probability is taken over the choice of $x \leftarrow \{0,1\}^n$ and over the internal randomness of M. The algorithm M may invoke $\mathcal{A}$ on various input lengths (i.e., in general M is not restricted to invoking $\mathcal{A}$ only on input length n), and we denote by $\ell(n)$ the maximal input length on which M invokes $\mathcal{A}$ (when M itself is invoked on input $f(x)$ for $x \in \{0,1\}^n$). Thus, viewing $M^{\mathcal{A}}$ as a single oracle-aided algorithm that has access to Γ, its running time $T_{M^{\mathcal{A}}}(n)$ satisfies $T_{M^{\mathcal{A}}}(n) \leq T_M(n) \cdot T_{\mathcal{A}}(\ell(n))$ (this follows since M may invoke $\mathcal{A}$ at most $T_M(n)$ times, and the running time of $\mathcal{A}$ on each such invocation is at most $T_{\mathcal{A}}(\ell(n))$). In particular, viewing $M' = M^{\mathcal{A}}$ as a single oracle-aided algorithm that has oracle access to Γ, implies that M' is a q-query algorithm where $q(n) = T_{M^{\mathcal{A}}}(n)$. This holds for any Ψ in the support of the distribution $\mathcal{D}(\mathfrak{S})$, and given that $q(n)$ is polynomial in n then Theorem 10 guarantees that $\epsilon_{M,1}(T_{\mathcal{A}}(n)) \cdot \epsilon_{M,2}(n) \leq O(2^{-n/2})$.

In the second case we obtain from Definition 6 that for every $\Psi = (f, \mathcal{O}, \mathsf{Eval}^{f,\mathcal{O}})$ in the support of the distribution $\mathcal{D}(\mathfrak{S})$ it holds that

$$\left| \Pr\left[\mathsf{Exp}^{\mathsf{iO}}_{\Gamma, i\mathcal{O}, M^{\mathcal{A}}, \mathcal{C}}(n) = 1 \right] - \frac{1}{2} \right| \geq \epsilon_{M,1}(T_{\mathcal{A}}(n)) \cdot \epsilon_{M,2}(n)$$

for infinitely many values of $n \in \mathbb{N}$, where the probability is taken over the randomness of the experiment $\mathsf{Exp}^{\mathsf{iO}}_{\Gamma, i\mathcal{O}, M^{\mathcal{A}}, \mathcal{C}}(n)$. The same reasoning applied to the first case, together with Theorem 15 guarantee that $\epsilon_{M,1}(T_{\mathcal{A}}(n)) \cdot \epsilon_{M,2}(n) \leq O(2^{-n/4})$.

We conclude the proof noting that the algorithm $\mathcal{A}$ provided by Lemma 8 runs in fact in linear time. That is, $T_{\mathcal{A}}(n) = O(n)$, and thus from the above two cases we obtain $\epsilon_{M,1}(n) \cdot \epsilon_{M,2}(n) \leq 2^{-\Omega(n)}$. ◀

References

1 William Aiello and Johan Håstad. Statistical zero-knowledge languages can be recognized in two rounds. *Journal of Computer and System Sciences*, 42(3):327–345, 1991.

2 Gilad Asharov and Gil Segev. Limits on the power of indistinguishability obfuscation and functional encryption. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science*, pages 191–209, 2015.

3 Gilad Asharov and Gil Segev. On constructing one-way permutations from indistinguishability obfuscation. In *Proceedings of the 13th Theory of Cryptography Conference*, pages 512–541, 2016.

4 László Babai, Lance Fortnow, and Carsten Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1:3–40, 1991.

5 Boaz Barak. Structure vs. combinatorics in computational complexity. Windows on Theory (available at `https://windowsontheory.org/2013/10/07/structure-vs-combinatorics-in-computational-complexity/`), 2013.

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

7 Paul Beame, Stephen A. Cook, Jeff Edmonds, Russell Impagliazzo, and Toniann Pitassi. The relative complexity of NP search problems. In *Proceedings of the 27th Annual ACM Symposium on Theory of Computing*, pages 303–314, 1995.

8 Michael Ben-Or, Shafi Goldwasser, Joe Kilian, and Avi Wigderson. Multi-prover interactive proofs: How to remove intractability assumptions. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pages 113–131, 1988.

9 Nir Bitansky and Akshay Degwekar. On the complexity of collision resistant hash functions: New and old black-box separations. In *Proceedings of the 17th Theory of Cryptography Conference*, pages 422–450, 2019.

10 Nir Bitansky, Akshay Degwekar, and Vinod Vaikuntanathan. Structure vs. hardness through the obfuscation lens. In *Advances in Cryptology – CRYPTO '17*, pages 696–723, 2017.

11 Nir Bitansky, Omer Paneth, and Alon Rosen. On the cryptographic hardness of finding a Nash equilibrium. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science*, pages 1480–1498, 2015.

12 Nir Bitansky, Omer Paneth, and Daniel Wichs. Perfect structure on the edge of chaos – trapdoor permutations from indistinguishability obfuscation. In *Proceedings of the 13th Theory of Cryptography Conference*, pages 474–502, 2016.

13 Manuel Blum and Russell Impagliazzo. Generic oracles and oracle classes. In *Proceedings of the 28th Annual IEEE Symposium on Foundations of Computer Science*, pages 118–126, 1987.

14 Andrej Bogdanov and Chin Ho Lee. Limits of provable security for homomorphic encryption. In *Advances in Cryptology – CRYPTO '13*, pages 111–128, 2013.

15 Gilles Brassard. Relativized cryptography. In *Proceedings of the 20th Annual IEEE Symposium on Foundations of Computer Science*, pages 383–391, 1979.

16 Richard Chang, Benny Chor, Oded Goldreich, Juris Hartmanis, Johan Håstad, Desh Ranjan, and Pankaj Rohatgi. The random oracle hypothesis is false. *Journal of Computer and System Sciences*, 49(1):24–39, 1994.

17 Stephen A. Cook, Russell Impagliazzo, and Tomoyuki Yamakami. A tight relationship between generic oracles and type-2 complexity theory. *Information and Computation*, 137(2):159–170, 1997.

18 Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

19 Lance Jeremy Fortnow. Complexity-theoretic aspects of interactive proof systems. Ph.D. Thesis, MIT, 1989.

20 Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science*, pages 40–49, 2013.

21 Sanjam Garg, Omkant Pandey, and Akshayaram Srinivasan. Revisiting the cryptographic hardness of finding a Nash equilibrium. In *Advances in Cryptology – CRYPTO '16*, pages 579–604, 2016.

22 Oded Goldreich. On security preserving reductions – revised terminology. Cryptology ePrint Archive, Report 2000/001, 2000.

23 Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

24 Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

25 Juris Hartmanis and Lane A. Hemachandra. One-way functions, robustness, and the non-isomorphism of NP-complete sets. In *Proceedings of the 2nd Annual Conference on Structure in Complexity Theory*, 1987.

26 Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, pages 44–61, 1989.

27 Ilan Komargodski, Tal Moran, Moni Naor, Rafael Pass, Alon Rosen, and Eylon Yogev. One-way functions and (im)perfect obfuscation. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science*, pages 374–383, 2014.

28 Tianren Liu and Vinod Vaikuntanathan. On basing private information retrieval on NP-hardness. In *Proceedings on the 13th Theory of Cryptography Conference*, pages 372–386, 2016.

29 Michael Luby. *Pseudorandomness and Cryptographic Applications.* Princeton University Press, 1996.

30 Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *Journal of the ACM*, 39(4):859–868, 1992.

31 Omer Reingold, Luca Trevisan, and Salil P. Vadhan. Notions of reducibility between cryptographic primitives. In *Proceedings of the 1st Theory of Cryptography Conference, TCC 2004*, pages 1–20, 2004.

32 Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communication of the ACM*, 21(2):120–126, 1978.

33 Steven Rudich. *Limits on the Provable Consequences of One-way Functions*. PhD thesis, EECS Department, University of California, Berkeley, 1988.

34 Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: Deniable encryption, and more. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 475–484, 2014.

35 Gil Segev and Ido Shahaf. Hardness vs. (very little) structure in cryptography: A multi-prover interactive proofs perspective. Cryptology ePrint Archive, Report 2020/330, 2020. URL: `https://eprint.iacr.org/2020/330`.

36 Adi Shamir. IP=PSPACE. In *Proceedings of the 31st Annual IEEE Symposium on Foundations of Computer Science*, pages 11–15, 1990.

37 Brent Waters. A punctured programming approach to adaptively secure functional encryption. In *Advances in Cryptology – CRYPTO '15*, pages 678–697, 2015.