On the Existence of Three Round Zero-Knowledge Proofs

Nils Fleischhacker[1,2](✉), Vipul Goyal[1], and Abhishek Jain[2]

[1] Carnegie Mellon University, Pittsburgh, USA
mail@nilsfleischhacker.de
[2] Johns Hopkins University, Baltimore, USA

Abstract. We study the round complexity of zero-knowledge (ZK) *proof* systems. While five round ZK proofs for NP are known from standard assumptions [Goldreich-Kahan, J. Cryptology'96], Katz [TCC'08] proved that four rounds are insufficient for this task w.r.t. black-box simulation. In this work, we study the feasibility of ZK proofs using *non-black-box* simulation. Our main result is that *three round* private-coin ZK proofs for NP do not exist (even w.r.t. non-black-box simulation), under certain assumptions on program obfuscation. Our approach builds upon the recent work of Kalai et al. [Crypto'17] who ruled out constant round *public-coin* ZK proofs under the same assumptions as ours.

1 Introduction

The notion of zero-knowledge (ZK) proofs [32] is fundamental in cryptography. Intuitively, ZK proofs allow one to prove a statement without revealing anything beyond the validity of the statement.

An important measure of efficiency of ZK protocols is *round complexity*. Ever since the introduction of ZK proofs nearly three decades ago, an extensive amount of research has been dedicated towards minimizing their round-complexity. Protocols with smaller round complexity are more desirable so as to minimize the effect of network latency, which in turn decreases the time complexity of the protocol.

Round-Complexity of ZK. In this work, we study the exact round complexity of ZK *proofs* that achieve soundness even against computationally unbounded adversarial provers (as opposed to *arguments* that achieve soundness only against polynomial-time adversarial provers). While initial constructions of ZK proofs required a polynomial number of rounds, the seminal work of Goldreich and Kahan [29] constructed a five round ZK proof system for NP based on collision-resistant hash functions.

N. Fleischhacker and V. Goyal—We acknowledge the generous support of Northrop Grumman.

N. Fleischhacker and A. Jain—Supported in part by a DARPA/ARL Safeware Grant W911NF-15-C-0213 and a sub-award from NSF CNS-1414023.

J. B. Nielsen and V. Rijmen (Eds.): EUROCRYPT 2018, LNCS 10822, pp. 3–33, 2018.
https://doi.org/10.1007/978-3-319-78372-7_1

In the negative direction, two-round ZK arguments for NP were ruled out by Goldreich and Oren [31]. Later, Goldreich and Krawcyzk [30] ruled out three round ZK arguments for NP where the ZK property holds w.r.t. a black-box simulator. More recently, Katz [37] proved that four round ZK proofs with black-box simulation only exist for languages whose complement is in MA.

The above state of the art motivates the following intriguing question:

Does there exist a three or four round ZK proof system for NP *using non-black-box simulation?*

In this work, we investigate precisely this question.

Private-coin vs Public-coin. In the study of ZK proofs, whether or not the verifier makes its random coins public or keeps them private has a strong bearing on the round-complexity. Indeed, constructing *public-coin* ZK proofs is viewed as a harder task. Very recently, Kalai et al. [36] ruled out constant round public-coin ZK proof systems for NP, even w.r.t. non-black-box simulation, assuming the existence of certain kinds of program obfuscation [7]. However, their approach breaks down in the *private coin* setting, where a verifier may keep its random coins used during the protocol private from the prover. This is not surprising, since five round private-coin ZK proofs are already known [29].

In this work, we investigate the feasibility of constructing private-coin ZK proofs (via non-black-box techniques) in less than five rounds. We remark that a candidate construction of three-round (private-coin) ZK proof system was given by Lepinski [40] based on a highly non-standard "knowledge-type" assumption; we discuss the bearing of our results on Lepinski's protocol (and the underlying assumption) below.

1.1 Our Results

We revisit the round complexity of zero-knowledge proof systems. As our main result, we rule out the existence of *three round* private-coin ZK proofs for languages outside BPP, under certain strong assumptions.

Theorem 1 (Informal). *Three round ZK proofs against non-uniform verifiers and distinguishers only exist for languages in* BPP*, assuming the following:*

- *Sub-exponentially secure one-way functions.*
- *Sub-exponentially secure indistinguishability obfuscation for circuits* [7,25].
- *Exponentially secure input-hiding obfuscation for multi-bit point functions* [5,11].

Our result relies on the same assumptions as those used in the recent work of Kalai et al. [36]. In their work, Kalai et al. use these assumptions to instantiate the Fiat-Shamir heuristic [24] and then rely upon its connection with public-coin ZK proofs [22] to rule out constant round public-coin ZK proofs. Naturally, this approach does not extend to the private coin setting. Nevertheless, we are able to build upon their techniques to obtain our result in Theorem 1.

Further, we note that our result contradicts the work of Lepinski [40] and thus refutes the knowledge-type assumption underlying Lepinski's protocol. We further elaborate on this in Sect. 1.3.

On our assumptions. Starting with the work of [25], several candidate constructions of indistinguishability obfuscation (iO) have been proposed over the last few years (see, e.g., [2–4,15,26,27,41–45,47]). During this time, (sub-exponentially secure) iO has also led to numerous advances in theoretical cryptography (see, e.g., [13,21,25,46]). Nevertheless, no iO scheme whose security is based on standard cryptographic assumptions is presently known.

Our second assumption on program obfuscation concerns with the notion of input-hiding obfuscation [5] for the class of multi-bit point functions $\mathcal{I}_{\alpha,\beta}$, where $\mathcal{I}_{\alpha,\beta}(\alpha) = \beta$ and 0, otherwise. Roughly speaking, an input-hiding obfuscator for this family is said to be T-secure, if any PPT adversary can succeed in guessing α with probability at most T^{-1}. For our purposes, we require T to be exponential in the security parameter. Candidate constructions of such obfuscation based on a strong variant of the DDH assumption are known from the works of [11,19] (see Sect. 2 for a more detailed discussion.)

Pessimistic Interpretation. While it is natural to be somewhat skeptical about the obfuscation assumptions we make, we note that our result implies that constructing three-round zero-knowledge proofs would require overcoming significant technical barriers. In particular, it would require disproving the existence of sub-exponentially secure iO, or the existence of exponentially secure input-hiding obfuscation for multi-bit point functions (or, less likely, disproving the existence of sub-exponentially secure one-way functions).

What about four rounds? Our result in Theorem 1 also extends to a specific relaxation of ZK, referred to as ϵ-ZK [14]. In this relaxed notion, the simulator's running time may grow polynomially with the distinguishing gap, which is allowed to be an inverse polynomial (unlike standard ZK, where the distinguishing gap must be negligible).

In a recent work, Bitansky et al. [14] construct a four round private coin ϵ-ZK proof system for NP, assuming the existence of keyless multi-collision-resistant hash functions (MCRH) [9,14,39]. Multi-collision-resistant hash functions weaken the standard notion of collision-resistant hash functions by only guaranteeing that an adversary cannot find many (rather than two) inputs that map to the same image. Presently, no constructions of keyless MCRH based on standard assumptions are known; however, unlike collision-resistant hash functions that cannot be secure against non-uniform adversaries in the keyless setting, keyless MCRH are meaningful even in the non-uniform setting if the number of required collisions are larger than the non-uniform advice to the adversary.

Their result serves as evidence that our techniques are unlikely to extend to the four round case, since otherwise it would imply the non-existence of keyless MCRH. While this is not implausible based on current evidence, in our eyes, it would be a rather surprising outcome.

It is of course possible that while four round private-coin ϵ-ZK proofs exist, four round private-coin ZK proofs do not. However, in light of the above, it seems that ruling out four round private-coin ZK proofs (w.r.t. non-black-box simulation) would require substantially new techniques.

1.2 Technical Overview

In order to rule out the existence of three-round zero knowledge proofs, we need to show that for any imaginable three round proof system, there exists a non-uniform adversarial verifier whose view cannot be efficiently simulated by any non-black-box simulator. Since a non-black-box simulator has access to the adversary's code, an immediate challenge is to "hide" the random coins of the adversarial verifier from the simulator.

Our starting approach to address this issue is to use program obfuscation. Let Π be any three-round private-coin proof system. To prove that Π is not ZK, we construct a "dummy" adversarial verifier V^* who receives as auxiliary input aux, an obfuscation of the next-message function of the honest verifier algorithm of Π. More concretely, the auxiliary input aux consists of an obfuscated program that has a key k for a pseudorandom function (PRF) hardwired in its description:

1. Upon receiving a message α from the prover, the program computes a message β of the honest verifier (as per protocol Π) using randomness $r = \mathsf{PRF}_k(\alpha)$.[1]
2. Upon receiving a protocol transcript (α, β, γ), it recomputes the randomness r used to compute β. Using the randomness r and the transcript, it honestly computes the verifier's output (i.e., whether to accept or reject the proof).

The adversarial verifier's code does not do anything intelligent on its own, and simply uses its auxiliary input aux to compute its protocol message.

Ruling out Rewinding Simulators. The above strategy for hiding the random coins of the verifier runs into the following problem: a simulator may fix the first two messages (α, β) of the protocol, and then observe the verifier's output on many different third messages to learn non-trivial information about the private randomness of the verifier. Indeed, it was recently shown in the work of Jain et al. [35] that in certain protocols, a simulator can learn the verifier's random tape by observing whether the verifier accepts or rejects in multiple trials.

A naive approach to address this problem is to simply modify the adversary and remove the protocol output from adversary's view. This can be achieved by deleting the second instruction in the obfuscated program aux. This approach, however, immediately fails because now a simulator can simply simulate a "rejecting" transcript and succeed in fooling any distinguisher.

We address this problem by using non-uniform distinguishers, in a manner similar to Goldreich and Oren [31] and the recent work of [1]. Specifically, we modify the adversarial verifier to be such that it simply outputs the protocol

[1] One may notice that this is similar to how protocols secure against "reset attacks" are constructed [6,20].

transcript at the end of the protocol. The revised auxiliary input aux only contains the first instruction described above. The PRF key k used to compute the verifier's randomness inside aux is given as non-uniform advice to the distinguisher. Note that this information is not available to the simulator. Now, given k and the protocol transcript, the distinguisher can easily decide whether or not to accept the transcript. Therefore, a simulator can no longer fool the distinguisher via a rejecting transcript.

How to rule out any Simulator? Of course the main problem remains. While the above approach constitutes a meaningful first step, we still need to formally argue that there does not exist any efficient simulator for the aforementioned adversarial verifier.

In prior works such as [31], this is achieved by showing that any efficient simulator algorithm can be used by a cheating prover to break the soundness of candidate protocol, which leads to a contradiction. It is, however, not immediately clear how to implement this strategy in our setting since a cheating prover does not have access to the code of the verifier (which is required for running the simulator algorithm).

We instead show that the existence of an efficient simulator can be used to disprove the computational soundness of a *different* protocol that is provably sound, leading to a contradiction.

Contradiction via Round Compression. We implement a compiler for compressing any three round private coin proof system into a two round *argument* system. Our round compression strategy is in fact very similar to the one developed in the recent work of Kalai et al. [36] in the context of public-coin ZK proofs. We then show that a simulator for the three round proof w.r.t. the aforementioned non-uniform verifier can be used to construct a cheating prover for the two round argument system.

We now elaborate on the round compression strategy. Consider the prover and verifier of the three-round proof to be two-stage algorithms. That is, P_1 produces the prover's first message α, V_1 is the verifier's next message function that on input α outputs the verifier's message β, P_2 on input β produces the prover's second message γ and finally V_2 is the decision procedure which uses the random tape to decide whether (α, β, γ) is an accepting transcript. The compressed two-round *argument* works as follows:

1. In the first round, the verifier obfuscates the code of a slightly modified V_1 that upon input α, computes its message β using randomness $r = \mathsf{PRF}_k(\alpha)$ generated via a hardcoded PRF key k. The verifier then sends the obfuscated program to the prover.
2. The prover now runs P_1 to get α, evaluates the obfuscated program on α to receive β and finally runs P_2 on α, β to get γ. The prover then sends α, β, γ to the verifier.
3. Finally, the verifier can use k to recompute the random tape $\mathsf{PRF}_k(\alpha)$ and run V_2 to validate the transcript.

A minor variant of the above strategy was recently used by Kalai et al. [36] in the case of public-coin ZK proofs. In their case, the obfuscated program simply corresponds to a PRF algorithm since it suffices to implement the strategy of a public-coin verifier.[2]

Now, using the above round compression strategy, we can compress any three-round proof system Π into a two-round argument system Π'. Now suppose that there exists an efficient zero-knowledge simulator Sim for Π w.r.t. the adversarial verifier V^* with auxiliary input aux, as described earlier. It is easy to see that such a simulator Sim can be used to construct an efficient cheating prover P^* for Π'. Indeed, the view of Sim in Π against V^* with aux is the same as the view of P^* against an honest verifier in Π'.

Thus, the main challenge now is to prove that our round-compression strategy indeed yields two-round arguments.

How to prove Soundness? To prove computational soundness of the two-round protocol, we proceed in two main steps:

1. First, we establish that there exists only a very small set of "bad" first messages α for which the cheating prover can even hope to be successful.
2. Second, we prove that the obfuscation sufficiently hides this small set to ensure that the cheating prover cannot find such an α.

Below, we elaborate on each of these steps.

Step 1: Upper bounding Bad α's. Imagine for a moment, that the three-round proof system is public coin. Then, for any $x \notin \mathcal{L}$ and any α, there can only exist a negligible fraction of random tapes (and therefore β) for which an accepting γ even exists. This is true because otherwise the computationally unbounded prover could simply exhaustively search for this γ once they receive β. Now, if the random tape, as in the two-round argument, is chosen pseudorandomly as a function of α, then only a very small set of α's will lead to such *bad* random tapes. This is because a distinguisher against the pseudorandom function can test for bad α's by exhaustively enumerating γ's because the PRF is assumed to be 2^n-secure. This small set would then be the set of bad α's. Clearly any successful cheating prover *must* use a bad α, since those are the only ones for which an accepting γ even exists.

In a private coin protocol, however, this notion of bad α's does not work. In fact in a private coin protocol, for any α and any random tape, an accepting γ may always *exist!* Indeed, any three-round proof system can be transformed into another proof system that has this property: the verifier in the new protocol acts exactly as the original verifier, except that it also chooses a random γ^* that it keeps private. Now, once it receives γ from the prover in the third round, the verifier accepts if either the original verifier accepts or $\gamma = \gamma^*$. Clearly in this protocol, there always exists an accepting γ but the protocol nevertheless remains sound. To break soundness, a prover must either break soundness of the original protocol or guess γ^* which is only possible with negligible probability, because the entire transcript is independent of γ^*.

[2] In particular, in the public-coin case, the obfuscated program can be interpreted as an instantiation of the random oracle in the Fiat-Shamir heuristic.

This example does not only show that the notion of bad α's from the public coin case does not work in the private-coin case, it also helps to illustrate how we can try to fix it. While an accepting γ may always exist, the prover only learns β and cannot tell which random tape was used by the verifier, beyond the obvious fact that it must have been consistent with β. Therefore, the only γ a prover can hope to use to break the soundness of the protocol are those that, for a fixed β, are accepted by *many* consistent random tapes.

We use this key observation to derive our new notion of bad α's. For any α there exists only a negligible fraction of random tapes that are consistent with a β such that there exists a γ that is accepted with high probability over all the random tapes consistent with β. This is true, because otherwise an unbounded prover could choose a random α and after receiving β, exhaustively search for all consistent random tapes and then search for the γ accepted by many of them. And then again, if the random tape, as is done in the two-round argument, is chosen pseudorandomly as a function of α, then only a very small set of α's will lead to such *bad* random tapes.

However, must a cheating prover in the two-round protocol necessarily use such a bad α to convince a verifier? *While in the public coin case this was a trivial fact, this is not at all obvious in the more general private-coin case.* Since even for "good" random tapes accepting γ's may exist, it is necessary to show that these remain hidden and cannot be used to cheat.

Here indistinguishability obfuscation comes to the rescue. Using iO and puncturable PRFs, we can show that a cheating prover must remain oblivious about which consistent random tape was used to compute β. This allows us to argue that a cheating prover cannot make use of γ's that are only accepting for a small number of consistent random tapes. Therefore, with overwhelming probability, a successful cheating prover must use a bad α.

Step 2: Hiding Bad α's. Now, it remains to argue that this set of bad α is hidden by the obfuscation. Once we have established that a cheating prover must output a bad α, the most obvious idea would be to try and lead this to a contradiction with the soundness of the three-round proof. However, to translate this into an attack, we need to use the security of the PRF. And while using iO, that means we need to puncture. Since the puncturing must be done *before* we learn α used by the cheating prover, we would incur an exponential loss in the success probability of the hypothetical three-round cheating prover. We can therefore only bring this to a contradiction if the three-round proof is exponentially sound, which would severely weaken the result. Instead, we follow the same approach as Kalai et al. [36] and "transfer" the exponential loss to another cryptographic primitive.

The idea is to use the security of another primitive to argue that bad α's are hidden. Since the goal is to argue that bad inputs to a circuit remain hidden a natural candidate for this primitive is input-hiding obfuscation. And indeed, sufficiently strong input-hiding obfuscation for multibit point functions allows to lead the existence of a cheating prover to a contradiction. Some technical issues arise in this proof due to the distribution of bad α's not being uniform.

However, using a clever trick of a "relaxed" verifier it is possible to show that the distributions are sufficiently close. In this part of the proof, we are able to adapt the elegant strategy of Kalai et al. [36] with only minor modifications.

Extension to ϵ-ZK. To extend our result to also rule out three-round ϵ-ZK proofs, we mainly need to argue that the cheating prover we described above is still successful in breaking soundness of the two-round argument, even if our starting point is an ϵ-ZK simulator instead of a regular ZK simulator.

Towards this, we note that the ϵ-ZK simulator, for every noticeable function ϵ, is required to output a distribution that is ϵ-indistinguishable from the real distribution. Thus, we can choose any small noticeable function ϵ, and then this means that, while the cheating prover against the two-round argument is no longer successful with all but negligible probability, it is still successful with probability $1 - \epsilon$. This is sufficient to break soundness and our main theorem therefore extends to ϵ-ZK proofs.

1.3 Implications to Lepinski's Protocol

Lepinski's 3-round ZK proof protocol [40] is based on a clever combination of the three round honest-verifier ZK protocol of Blum [16] for Hamiltonian Graphs and a special kind of oblivious transfer. While Lepinski chose to give a more direct description of his protocol, a more modular high-level construction is implicit in his thesis. His construction makes use of two building blocks:

1. The three round honest-verifier ZK protocol of Blum for Hamiltonian Graphs.
2. A three round string OT protocol with the following properties:
   - The protocol is "delayed input" on the sender's side. I.e., the first round of the OT can be computed independently of the sender's inputs (m_0, m_1).
   - The protocol achieves indistinguishability based security against a *computationally unbounded* malicious sender.
   - The protocol achieves simulation based security against a malicious polynomial time receiver.

Based on these assumptions a three-round ZK proof can be constructed as described below. In the description we focus on soundness 1/2. For this specific protocol, smaller soundness error can be achieved by parallel repetition without affecting the ZK property.

1. In the first round, the prover sets up the OT by sending the first message.
2. In the second round, the verifier sends the OT receiver message corresponding to their random challenge for Blum's protocol. I.e. the Blum challenge is used as the selection bit b in the OT.
3. In the third round the prover sends the first message of Blum's protocol. Additionally he sends the sender message of the OT, corresponding to the two possible prover responses to the (as of yet unknown) challenge. I.e. the prover sets m_b in the OT to be the response to challenge b.
4. Finally the verifier receives the OT message, thus learning m_b and verifies that m_b is a valid response in Blum's protocol.

It is easy to verify that this protocol is indeed sound: since the OT is secure against an unbounded sender, the prover must choose his first message without knowledge of the challenge and if the graph does not contain a Hamiltonian cycle then it can only give a valid response to one of the challenges and is thus only successful with probability 1/2. The soundness of this protocol is uncontested by our result.

To prove that the above protocol is also zero-knowledge, one can leverage the simulation-based security of the OT against malicious receivers. In particular, the ZK simulator uses the OT simulator to learn the OT selection bit, and then uses it to invoke the honest-verifier ZK simulator for Blum's protocol. This part is disputed by our result. Since the security of Blum's protocol is not in question, this means that our result disputes the existence of an OT protocol with the properties described above.

However, Lepinski implicitly gives a number-theoretic construction of such an OT protocol using a very specific "knowledge-type" assumption that is referred to as the "proof of knowledge assumption (POKA)" in this thesis. This assumption essentially states that a specific three-round public-coin proof of knowledge protocol remains a proof of knowledge even if the verifier's challenge is computed using a fixed hash function. This assumption is necessary to facilitate extraction of the receiver's selection bit in his OT protocol, which is the key to proving simulation-based security against malicious receivers.

The question, of course, remains how this protocol and the underlying assumption exactly relate to our impossibility result. For that, we should first note that Lepinski does not explicitly prove his protocol to be zero-knowledge relative to non-uniform verifiers. Since our impossibility result only rules out three-round ZK with non-uniform verifiers, our result – taken literally – does not directly apply to the protocol as stated. However, it is easy to see that Lepinski's protocol does, in fact, achieve ZK against non-uniform verifiers if the POKA assumption is suitably augmented so that it holds even against provers with arbitrary auxiliary input. This augmented assumption is therefore what is specifically refuted by our result.

In a bit more detail, what does it mean exactly to apply our result to Lepinski's protocol? As mentioned earlier, the soundness of the protocol is not in question. Therefore, the round compression part of our proof works exactly as stated, i.e., we are able to compress Lepinski's three-round proof into a two round argument. It is the second part of our result, where we show that the soundness of the two round argument and the zero-knowledge property of the three-round proof contradict each other, where we get the refutation of the POKA assumption.

Essentially, in this part of the proof, we show that in the compressed two-round argument, a malicious prover is capable of using the ZK-simulator for the three-round proof to cheat and break soundness. Since the soundness of the protocol is not in question, this means that we are refuting the existence of the ZK-simulator and thus, that the 3-round protocol can be zero-knowledge. In the generalized terms in which we described Lepinski's protocol above the

ZK-simulator only requires the HVZK-simulator of Blum's protocol and the OT-simulator to work. This means that our work specifically refutes the simulation based receiver-security of the OT protocol. If we look at our result in a bit more detail, it is also clear why this is the case. Essentially, we are constructing a malicious prover who is capable of running the ZK-simulator for the 3-round proof. For Lepinski's construction to work, this simulator must be able to extract the selection bit in the OT from the verifier's message. This means that we are constructing an algorithm capable of extracting the selection bit of the receiver while acting as a malicious sender in the OT protocol. Clearly, this immediately implies that the OT is broken.

1.4 Related Work

There is a large body of work dedicated to the study of round complexity of zero-knowledge protocols. Below, we provide a brief (and incomplete) summary of some of the prior work in this area.

ZK Proofs. Five-round ZK proofs are known based on collision-resistant hash functions [29], and four-round ϵ-ZK proofs were recently constructed based on keyless multi-collision-resistant hash functions [14]. Both of these constructions require the verifier to use private coins. There also exists a candidate for a three-round ZK proof due to Lepinski [40], which ultimately clashes with our result. Lepinski's protocol is based on a highly non-standard knowledge-type assumption which our result refutes. We explain the exact relationship and implications in Sect. 1.3.

Dwork et al. [22] (and independently, Hada and Tanaka [33]) established an intimate connection between the Fiat-Shamir paradigm [24] and constant-round public-coin ZK proofs. Using their result, [36] recently ruled out the existence of constant-round public-coin ZK proofs, under the same assumptions as in our work. Previously, such protocols were only ruled out w.r.t. black-box simulation by [30]. We refer the reader to [36] for further discussion on public-coin ZK proofs.

ZK Arguments. Four-round ZK arguments are known based on one-way functions [8,23]. Goldreich and Krawcyzk [30] ruled out the existence of three-round ZK arguments for NP w.r.t. black-box simulation. While three-round ZK arguments with non-black-box simulators were unknown for a long time, some recent works have studied them w.r.t. weaker adversaries such as uniform provers [10], or uniform verifiers [12], while finally Bitansky et al. [14] were very recently able to construct general three round ZK arguments for non-uniform provers and verifiers based on keyless multi-collision-resistant hash functions.

2 Preliminaries

We denote by $n \in \mathbb{N}$ the security parameter that is implicitly given as input to all algorithms in unary representation 1^n. We denote by $\{0,1\}^\ell$ the set of all bit-strings of length ℓ. For a finite set S, we denote the action of sampling x uniformly at random from S by $x \leftarrow_{\$} S$, and we denote the cardinality of S by $|S|$. Al algorithms are assumed to be randomized, unless explicitly stated otherwise. An algorithm is efficient or PPT if it runs in time polynomial in the security parameter. If $\mathcal{A}$ is randomized then by $y \leftarrow \mathcal{A}(x; r)$ we denote that $\mathcal{A}$ is run on input x and with random coins r and produced output y. If no randomness is specified, then it is assumed that $\mathcal{A}$ is run with freshly sampled uniform random coins, and write this as $y \leftarrow_{\$} \mathcal{A}(x)$. For a circuit C we denote by $|C|$ the size of the circuit. A function $\mathsf{negl}(n)$ is negligible if for any positive polynomial $\mathsf{poly}(n)$, there exists an $N \in \mathbb{N}$, such that for all $n > N$, $\mathsf{negl}(n) \leq \frac{1}{\mathsf{poly}(n)}$.

2.1 Interactive Proofs and Arguments

An interactive proof for an NP language $\mathcal{L}$ is an interactive protocol between two parties, a computationally unbounded prover and a polynomial-time verifier. The two parties receive a common input x and the prover tries to convince the verifier that $x \in \mathcal{L}$. Intuitively the prover should (almost) always be successful if x is indeed in $\mathcal{L}$, but should be limited in its ability to convince the verifier if $x \notin \mathcal{L}$. An interactive proof, as formally introduced by Goldwasser et al. [32] is defined as follows.

Definition 1 (Interactive Proof). *An r-round 2-Party protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ between a polynomial-time verifier V and an unbounded prover P is an interactive proof with soundness error ϵ for language $\mathcal{L}$ if the following two conditions hold:*

1. Completeness: *For all $x \in \mathcal{L}$ it holds that $\Pr_{\mathsf{P},\mathsf{V}}[1 \leftarrow \langle \mathsf{P}(x), \mathsf{V}(x) \rangle] = 1 - \mathsf{negl}(n)$.*
2. Soundness: *For all $x^* \notin \mathcal{L}$ and all computationally unbounded malicious provers P^* it holds that $\Pr_{\mathsf{P}^*,\mathsf{V}}[1 \leftarrow \langle \mathsf{P}^*, \mathsf{V}(x^*) \rangle] \leq \epsilon$.*

An interactive argument is very similar to an interactive proof, except that soundness is only required to hold relative to polynomial time malicious provers. Since also the honest prover is required to run in polynomial time, it receives an NP witness for x as an additional input. Formally, this leads to the following definition.

Definition 2 (Interactive Argument). *An r-round 2-Party protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ between a polynomial-time verifier V and a polynomial-time prover P is an interactive argument with soundness error ϵ for language $\mathcal{L}$ with associated relation $\mathcal{R}$ if the following two conditions hold:*

1. Completeness: *For all $(x, w) \in \mathcal{R}$ it holds that $\Pr_{\mathsf{P},\mathsf{V}}[1 \leftarrow \langle \mathsf{P}(x, w), \mathsf{V}(x) \rangle] = 1 - \mathsf{negl}(n)$.*
2. Soundness: *For all $x^* \notin \mathcal{L}$ and all polynomial-time malicious provers P^* it holds that $\Pr_{\mathsf{P}^*,\mathsf{V}}[1 \leftarrow \langle \mathsf{P}^*, \mathsf{V}(x^*) \rangle] \leq \epsilon$.*

An especially powerful class of interactive proofs and arguments are those that are zero-knowledge. Intuitively a zero-knowledge proof or argument ensures that a malicious polynomial time verifier cannot learn anything from an execution of the protocol, except that $x \in \mathcal{L}$. This was first formalized in [32] by requiring the existence of a polynomial time simulator capable of – without knowledge of an NP witness for x – simulating any interaction a malicious verifier might have with the prover. This implies that anything the verifier learns from a protocol execution it could have also learned without interacting with the prover. To obtain a contradiction in the main proof in Sect. 3 we will use the notion of non-uniform zero-knowledge, where both the malicious verifier as well as the distinguisher may be non-uniform.

Definition 3 (Non-uniform Zero-Knowledge with Auxiliary Input). *Let $\langle \mathsf{P}, \mathsf{V} \rangle$ be a 2-Party protocol. $\langle \mathsf{P}, \mathsf{V} \rangle$ is said to be non-uniformly zero-knowledge with auxiliary input, if for all (possibly malicious)* PPT *algorithms* V^* *there exists a* PPT *simulator* Sim*, such that for all* PPT *distinguishers $\mathcal{D}$ and all auxiliary inputs* aux *and* aux'*, it holds that for all statements x*

$$\left| \begin{array}{l} \quad \Pr\left[\mathcal{D}(\langle \mathsf{P}(x,w), \mathsf{V}^*(x,\mathsf{aux})\rangle, \mathsf{aux}') = 1\right] \\ -\Pr\left[\mathcal{D}(\mathsf{Sim}(x,\mathsf{aux}), \mathsf{aux}') = 1\right] \end{array} \right| \leq \mathsf{negl}(n).$$

2.2 Puncturable Pseudorandom Functions

The notion of puncturable pseudorandom functions was independently introduced in [17,18,38]. A puncturable pseudorandom function allows to *puncture* a key k on some fixed input x. This punctured key should still allow to correctly evaluate the PRF on any input other than x. However, the value of the function on input x should be indistinguishable from a unform random value, *even given the punctured key*. We define a strong notion of puncturable pseudorandom functions in the following.

Definition 4 (T-Secure Puncturable Pseudorandom Functions). *A pair of probabilistic polynomial time algorithms* $(\mathsf{PRF}, \mathsf{Puncture})$ *is a T-secure puncturable pseudorandom function with key length $\kappa(n)$ input length $i(n)$ and output length $o(n)$ if the following conditions hold:*

1. Functionality Preserved Under Puncturing: *For every $n \in \mathbb{N}$, every key $k \leftarrow_{\$} \{0,1\}^{\kappa(n)}$, every input $x \in \{0,1\}^{i(n)}$, every punctured key $\mathsf{k}\{x\}$, and every input $x' \in \{0,1\}^{i(n)} \setminus \{x\}$ it holds that $\mathsf{PRF}_k(x') = \mathsf{PRF}_{\mathsf{k}\{x\}}(x')$.*
2. Pseudorandomness: *For any fixed $x \in \{0,1\}^{i(n)}$ it holds that for every distinguisher $\mathcal{D}$ that runs in time at most $\mathsf{poly}(T(n))$ it holds that*

$$\left| \begin{array}{l} \quad \Pr_{k,\mathsf{Puncture}}\left[\mathcal{D}(\mathsf{Puncture}(k,x), x, \mathsf{PRF}_k(x)) = 1\right] \\ - \Pr_{k,\mathsf{Puncture},y}\left[\mathcal{D}(\mathsf{Puncture}(k,x), x, y) = 1\right] \end{array} \right| \leq \mathsf{negl}(T(n))$$

Our impossibility result uses 2^{2n}-secure puncturable pseudorandom functions. Note that these can be constructed using the GGM construction from any subexponentially secure one-way function [28,34] by for example using keys length $\kappa(n) = n^2$.

2.3 Obfuscation

Our impossibility result uses two different kinds of obfuscation, indistinguishability obfuscation and input-hiding obfuscation for multi-input point functions. Indistinguishability obfuscation (iO) was first suggested as a notion by Barak et al. [7] as a weaker form of obfuscation. The security guarantee of iO is that the obfuscation of two functionally equivalent circuits should result in indistinguishable output distributions. That is, any polnomial-time reverse engineering cannot detect which of two equivalent implementations was the source of an obfuscated program. This security may seem rather weak at first glance. However, following the introduction of a first candidate construction by Garg et al. [25] it has been shown in several works that even this seemingly weak notion of obfuscation is a very powerful tool. We formally define indistinguishability obfuscation below.

Definition 5 (T-Secure Indistinguishability Obfuscation). *Let $\mathbb{C}$ be a family of polynomial size boolean circuits. Let iO be a probabilistic polynomial time algorithm, which takes as input a circuit $\mathbf{C} \in \mathbb{C}$ and a security parameter 1^n, and outputs a boolean circuit $\mathbf{B}$ (not necessarily in $\mathbb{C}$). iO is a T-secure indistinguishability obfuscator if the following two conditions hold:*

1. Correctness: *For every $n \in \mathbb{N}$, every circuit $\mathbf{C} \in \mathbb{C}$ with input length ℓ, every obfuscated circuit $\mathbf{B} \leftarrow \mathsf{iO}(\mathbf{C}, 1^n)$ and every $x \in \{0,1\}^\ell$ it holds that $\mathbf{B}(x) = \mathbf{C}(x)$.*
2. Indistinguishability: *For every $n \in \mathbb{N}$, every pair of circuit $\mathbf{C}_1, \mathbf{C}_2 \in \mathbb{C}$ with identical input length ℓ and $|\mathbf{C}_1| = |\mathbf{C}_2|$, and every $\mathsf{poly}(T(n))$-time distinguisher $\mathcal{D}$ it holds that*

$$\left| \Pr_{\mathsf{iO},\mathcal{D}}[\mathcal{D}(\mathsf{iO}(\mathbf{C}_1, 1^n)) = 1] - \Pr_{\mathsf{iO},\mathcal{D}}[\mathcal{D}(\mathsf{iO}(\mathbf{C}_2, 1^n)) = 1] \right| \leq \mathsf{negl}(T(n))$$

Our impossibility result uses a strong notion of 2^{2n}-secure indistinguishability obfuscation for general circuits. This notion is implied by any subexponentially secure indistinguishability obfuscator by instantiating the security parameter with $\kappa(n) = n^2$.

The second form of obfuscation used in our result is input-hiding obfuscation for multi-bit point functions. The notion of input-hiding obfuscation was first suggested by Barak et al. in [5]. An input-hiding obfuscator for a family of circuits $\mathbb{C}$ guarantees that, given an obfuscation of a circuit $\mathbf{C}$ drawn uniformly at random from $\mathbb{C}$ it is hard for an adversary to find any input on which the circuit doesn't output 0.

Definition 6 (T-Secure Input-Hiding Obfuscation). *Let $\mathbb{C} = \{\mathbb{C}_n\}_{n\in\mathbb{N}}$ be a family of polynomial size boolean circuits, where $\mathbb{C}_n$ is a set of circuits operating on inputs of length n. A polynomial time obfuscator* hideO *is a T-secure input hiding obfuscator for $\mathbb{C}$ if the following two conditions hold:*

1. Correctness: *For every $n \in \mathbb{N}$, every circuit $\mathbf{C} \in \mathbb{C}_n$, every obfuscated circuit $\mathbf{B} \leftarrow \mathsf{iO}(\mathbf{C}, 1^n)$ and every $x \in \{0,1\}^n$ it holds that $\mathbf{B}(x) = \mathbf{C}(x)$.*
2. Input Hiding: *For every $n \in \mathbb{N}$, and all probabilistic polynomial time adversary $\mathcal{A}$ it holds that*

$$\Pr_{\mathbf{C}\leftarrow\mathbb{C}_n,\mathsf{hideO},\mathcal{A}}[\mathbf{C}(\mathcal{A}(\mathsf{hideO}(\mathbf{C}, 1^n))) \neq 0] \leq T^{-1}(n).$$

Note that this security definition differs from previous definitions of T-security in so far as it requires the adversary to run in polynomial time (in n). Our result specifically uses input-hiding obfuscation for multi-bit point functions. A multi-bit point function is characterized by two values x and y and is defined as the function that on input x outputs y and outputs 0 on all other inputs.

Definition 7 (T-Secure Input-Hiding Obfuscation for Multi-bit Point Functions). *Let $I_{x,y}$ denote the multi-bit point function with $I_{x,y}(x) = y$ and $I_{x,y}(x') = 0$ for all $x' \neq x$ and let k be a function $k : \mathbb{N} \to \mathbb{N}$. A polynomial time obfuscator* hideO *is a T-secure input hiding obfuscator for (n, k)-multi-bit point functions if it is a T-secure input-hiding obfuscator for all circuit families $\mathbb{C}$ for which the following properties hold.*

1. *All circuits in $\mathbb{C} = \{\mathbb{C}_n\}_{n\in\mathbb{N}}$ describe point functions with n-bit input and $k(n)$-bit output. I.e., $\mathbb{C}_n \subseteq \left\{I_{x,y} \middle| x \in \{0,1\}^n \wedge y \in \{0,1\}^{k(n)}\right\}$.*
2. *The marginal distribution on x is uniform for a uniformly sampled circuit $I_{x,y} \leftarrow_{\$} \mathbb{C}_n$.*

This notion was first studied by Bitansky and Cannetti in [11]. They also showed that an earlier candidate construction by Cannetti and Dakdouk [19] can be proven secure in the generic group model based on a strong variant of the DDH assumption. Our impossibility result requires 2^n-secure input hiding obfuscator for multi-bit point functions. This may on first glance seem problematic, since DDH (and thereby the instantiation due to Cannetti and Dakdouk [19]) can be broken in time less than 2^n even in the generic group model. However, in Definition 6 we explicitly – and in contrast to the other definitions in this section – require that the adversary runs in polynomial time. And known subexponential time attacks do not imply a polynomial time attack that is successful with probability greater than $\mathsf{poly}/2^{(n)}$.

3 Impossibility of Three-Round Zero-Knowledge Proofs

In this section we will prove our main result, i.e., that under the stated assumptions, zero-knowledge 3-round interactive proof systems for non-trivial languages cannot exist. Our result is formally stated in Theorem 2.

Theorem 2. *Let $\hat{\Pi}$ be a 3-round interactive proof system for a language $\mathcal{L} \notin \mathsf{BPP}$ with negligible soundness error μ. Assume the existence of a 2^{2n}-secure puncturable pseudorandom function, a 2^{2n}-secure indistinguishability obfuscator, and a $\mu \cdot 2^n \mathsf{poly}(n)$-secure input-hiding obfuscator for multi-bit point functions. Then $\hat{\Pi}$ cannot be non-uniformly zero-knowledge with auxiliary input.*

Proof (Theorem 2). Let $\hat{\Pi} = \langle \hat{\mathsf{P}}, \hat{\mathsf{V}} \rangle$ be a 3-round interactive proof system as described in Theorem 2. We consider the prover and verifier as two-stage algorithms, $\hat{\mathsf{V}} = (\hat{\mathsf{V}}_1, \hat{\mathsf{V}}_2)$, $\hat{\mathsf{P}} = (\hat{\mathsf{P}}_1, \hat{\mathsf{P}}_2)$. The first stage of the prover $\alpha \leftarrow \hat{\mathsf{P}}_1(x, w; r)$ on input the statement x, witness w and random coins r outputs the prover's first message α. The first stage of the verifier $\beta \leftarrow \hat{\mathsf{V}}_1(x, \alpha; s)$ on input the statement x, the prover's first message α and random coins s outputs the verifier's message β. The second stage of the prover $\gamma \leftarrow \hat{\mathsf{P}}_2(x, w, \beta; r)$ on input the statement x, witness w, the verifier's message β and random coins r outputs the prover's second message γ. The second stage of the verifier $b \leftarrow \hat{\mathsf{V}}_2(x, \alpha, \gamma; s)$ on input the statement x, the prover's messages α, γ and random coins s outputs a bit b indicating whether the proof is accepted of not. Note that without loss of generality we assume that the second stages do not take their own messages as input and instead recompute them when necessary.

First we slightly modify the protocol $\hat{\Pi}$ into the protocol $\Pi = \langle \mathsf{P}, \mathsf{V} \rangle$. The protocol behaves exactly as $\hat{\Pi}$, except that V_1 takes as its random coins $s = \sigma \| \hat{s}$ with $|\sigma| = \lceil \log \mu^{-1} \rceil$ and after running $\hat{\beta} \leftarrow \mathsf{V}_1(x, \alpha; \hat{s})$ outputs $\beta := \sigma \| \hat{\beta}$. The prover's second stage P_2 then again behaves exactly as $\hat{\mathsf{P}}_2$, and ignores σ. The following claim is immediately apparent.

Claim 3. *If $\hat{\Pi}$ is a 3-round interactive proof system with negligible soundness error μ, then Π is also a 3-round interactive proof system for the same language with the same negligible soundness error μ.*

This modification is therefore without loss of generality and will allow us to cleanly define a relaxed version of the verifier later in the proof, leading to a much simpler proof.

Now, we use the pseudorandom function PRF the indistinguishability obfuscator iO to construct a two-round protocol $\bar{\Pi} = \langle \bar{\mathsf{P}}, \bar{\mathsf{V}} \rangle$ as depicted in Fig. 1. The circuit $\mathbf{C}_{\mathsf{V}_1}$ is defined as follows:

```
$\mathbf{C}_{\mathsf{V}_1}[k, x](\alpha)$
$s := \mathsf{PRF}_k(\alpha)$
$\beta := \mathsf{V}_1(x, \alpha; s)$
return $\beta$
```

To prove Theorem 2 we will now use the following two lemmas proven in Sects. 3.1 and 3.2 respectively.

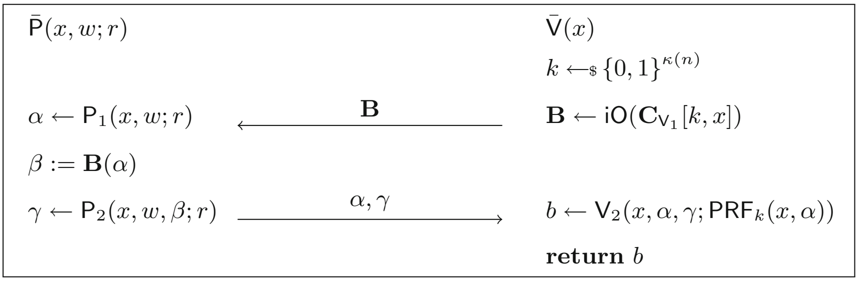
$\bar{\mathsf{P}}(x, w; r)$
$\bar{\mathsf{V}}(x)$
$k \leftarrow_{\$} \{0,1\}^{\kappa(n)}$
$\alpha \leftarrow \mathsf{P}_1(x, w; r)$
$\mathbf{B}$
$\mathbf{B} \leftarrow \mathsf{iO}(\mathbf{C}_{\mathsf{V}_1}[k, x])$
$\beta := \mathbf{B}(\alpha)$
$\gamma \leftarrow \mathsf{P}_2(x, w, \beta; r)$
α, γ
$b \leftarrow \mathsf{V}_2(x, \alpha, \gamma; \mathsf{PRF}_k(x, \alpha))$
return b

Fig. 1. The two-round argument system $\bar{\varPi} = \langle \bar{\mathsf{P}}, \bar{\mathsf{V}} \rangle$ resulting from compressing the three-round proof system $\varPi = \langle \mathsf{P}, \mathsf{V} \rangle$ into two rounds. The round compression is achieved by sending an obfuscated version of the verifier's own code to the prover as a first message. This allows the prover to compute the verifier's response to their first message without additional interaction. This construction is proven sound in Lemma 4.

Lemma 4. *Let $\hat{\varPi}$ be a 3-round interactive proof system with negligible soundness error μ as in Theorem 2. Let $\varPi$ be the modified 3-round interactive proof system as described above. Assume that PRF is a 2^{2n}-secure puncturable pseudorandom function, and iO is a 2^{2n}-secure indistinguishability obfuscator. Further assume that hideO is a 2^n-secure input-hiding obfuscator for multi-bit point functions. Then $\bar{\varPi}$, described in Fig. 1 is a 2-round interactive argument system with negligible soundness error $\bar{\mu}$.*

Lemma 5. *Let $\varPi$ be a 3-round interactive proof system for a language $\mathcal{L} \notin \mathsf{BPP}$. Let $\bar{\varPi}$ be the transformed 2-round argument system described in Fig. 1 with soundness error $\bar{\mu}$. If $\bar{\mu} \leq \mathsf{negl}(n)$ then $\varPi$ is not non-uniformly zero-knowledge with auxiliary input.*

Theorem 2 now follows as a simple corollary from combining Lemmas 4 and 5. By our assumption, $\varPi$ has a negligibly small soundness error μ, which by Lemma 4 also implies a negligible soundness error $\bar{\mu}$ for $\bar{\varPi}$. Since a negligible soundness error of $\bar{\varPi}$ implies that $\varPi$ is not non-uniformly zero-knowledge with auxiliary input, the theorem trivially follows. □

3.1 Proof of Lemma 4

Fix a modified 3-round interactive proof system $\varPi = \langle \mathsf{P}, \mathsf{V} \rangle$. Let $\mu \leq \mathsf{negl}(n)$ be the soundness error of $\varPi$. We assume without loss of generality, that all messages of the protocol have length n.

Assume towards contradiction that there exists a cheating PPT prover P^* breaking the soundness of $\bar{\varPi}$ for some $x^* \notin \mathcal{L}$ with probability $\nu = 1/\mathsf{poly}(n)$. I.e., we have that

$$\Pr_{k, \mathsf{iO}, \mathsf{P}^*}[\mathsf{V}_2(x^*, \alpha, \gamma; \mathsf{PRF}_k(\alpha)) = 1 : (\alpha, \gamma) \leftarrow \mathsf{P}^*(\mathsf{iO}(\mathbf{C}_{\mathsf{V}_1}[k, x]))] \geq \nu. \qquad (1)$$

To obtain a contradiction we analyze a variant of the protocol $\mathit{\Pi}$ that works with a relaxed verifier V'. The relaxed verifier V' works exactly as V, except that in addition to accepting whenever V does, it also accepts if $\beta = 0^{\lceil \log \nu/\mu \rceil} \| \beta'$ for some arbitrary β'. Remember, that $\beta = \sigma \| \hat{\beta}$ with $|\sigma| = \lceil \log \mu^{-1} \rceil$. I.e., V' also accepts if the first $\lceil \log \nu/\mu \rceil$ bits of σ are zero. In particular since V' always accepts if V accepts, it remains true that

$$\Pr_{k,\mathsf{iO},\mathsf{P}^*}[\mathsf{V}_2'(x^*, \alpha, \beta; \mathsf{PRF}_k(\alpha)) = 1 : \mathsf{P}^*(\mathsf{iO}(\mathbf{C}_{\mathsf{V}_1}[k, x])) = (\alpha, \gamma)] \geq \nu. \tag{2}$$

Further, using a union bound, we can bound the soundness error μ' of the relaxed 3-round protocol $\langle \mathsf{P}, \mathsf{V}' \rangle$ to be

$$\mu' \leq \mu + 2^{-\lceil \log \nu/\mu \rceil} \leq \mu + \frac{\mu}{\nu} \leq \frac{2\mu}{\nu}. \tag{3}$$

In particular, for any negligible μ, μ' remains negligible.

Let $S_{\alpha,\beta} = \{s | \mathsf{V}_1'(x^*, \alpha; s) = \beta\}$ denote the set of all random tapes that given α lead to the second message β. We define the following set of pairs (α, β), for which a malicious γ exists that will be accepted by the verifier for a large fraction of the random tapes that given α lead to β.

$$\mathsf{ACC} = \left\{ (\alpha, \beta) \middle| \exists \gamma : \Pr_{s' \twoheadleftarrow\$ S_{\alpha,\beta}}[\mathsf{V}_2'(x^*, \alpha, \gamma; s') = 1] \geq \frac{\nu}{2} \right\}.$$

Observe, that membership in ACC can be tested in time $2^{2n} \cdot \mathsf{poly}(n) = \mathcal{O}2^{2n}$ by enumerating all messages γ and all random tapes s, checking whether $\beta = \mathsf{V}_1'(x^*, \alpha; s)$ and $\mathsf{V}_2'(x^*, \alpha, \gamma; s) = 1$ and then computing the probability.[3] Given the cheating prover P^*, there exists an efficient algorithm $\mathcal{A}$ that outputs α, such that $(\alpha, \mathsf{V}_1'(\alpha; \mathsf{PRF}_k(\alpha))) \in \mathsf{ACC}$ with high probability. Formally this is stated in the following claim that is proven in Sect. 3.1.1.

Claim 6. *If there exists a malicious prover* P^* *as assumed above, then for the efficient algorithm* $\mathcal{A}$ *that on input* $\mathsf{iO}(\mathbf{C}_{\mathsf{V}_1}[k, x^*])|$ *runs* $(\alpha, \gamma) \leftarrow \mathsf{P}^*(\mathsf{iO}(\mathbf{C}_{\mathsf{V}_1}[k, x^*]))$*, discards* γ *and outputs* α *the following holds:*

$$\Pr_{k,\mathsf{iO},\mathcal{A}}[(\alpha, \mathsf{V}_1'(x^*, \alpha; \mathsf{PRF}_k(\alpha))) \in \mathsf{ACC} : \alpha \leftarrow \mathcal{A}(\mathsf{iO}(\mathbf{C}_{\mathsf{V}_1}[k, x^*]))] \geq \frac{\nu}{2} - 2^{-n}$$

Now consider the punctured version of the verifier circuit $\mathbf{C}_{\mathsf{pct}}$ defined follows:

```
$\mathbf{C}_{\mathsf{pct}}[k, \alpha^*, \beta^*](\alpha)$
if $\alpha \overset{?}{=} \alpha^*$
    $\beta := \beta^*$
else
    $s := \mathsf{PRF}_k(\alpha)$
    $\beta := \mathsf{V}_1'(x, \alpha; s)$
return $\beta$
```

[3] This assumes without loss of generality that $|\gamma| = |s| = n$.

We will use the following claim, which essentially states that, when given an obfuscation of the verifier's circuit punctured at α^*, the $\mathcal{A}$ from Claim 6 will output α^* with a probability slightly above random chance. The claim is proven in Sect. 3.1.2.

Claim 7. *If* PRF *is* 2^{2n}*-secure and* iO *is* 2^{2n}*-secure, then it must hold that*

$$\Pr_{k,\alpha^*,s^*,\mathsf{iO},\mathcal{A}}\Big[\mathcal{A}\Big(\mathsf{iO}\big(\mathbf{C}_{\mathsf{pct}}[\mathsf{k}\{\alpha^*\},\alpha^*,\mathsf{V}'_1(x^*,\alpha;s^*)]\big)\Big)=\alpha^* \,\Big|\, \Big(\alpha^*,\mathsf{V}'_1(x^*,\alpha;s^*)\Big)\in\mathsf{ACC}\Big]$$
$$\geq \frac{1}{8}\cdot 2^{-n}\cdot\frac{\nu^2}{\mu'}.$$

This property of $\mathcal{A}$ contradicts the security of the input hiding obfuscator hideO as shown in the following. We claim that

$$\Pr_{k,\alpha^*,s^*,\mathsf{hideO},\mathsf{iO},\mathcal{A}}\Big[\mathcal{A}\Big(\mathsf{iO}\big(\mathbf{C}_{\mathsf{hide}}[k,\mathsf{hideO}(\alpha^*,s^*)]\big)\Big)=\alpha^* \,\Big|\, \Big(\alpha^*,\mathsf{V}'_1(x^*,\alpha;s^*)\Big)\in\mathsf{ACC}\Big]$$

$$\geq \Pr_{k,\alpha^*,s^*,\mathsf{iO},\mathcal{A}}\Big[\mathcal{A}\Big(\mathsf{iO}\big(\mathbf{C}_{\mathsf{pct}}[\mathsf{k}\{\alpha^*\},\alpha^*,\mathsf{V}'_1(x^*,\alpha;s^*)]\big)\Big)=\alpha^* \,\Big|\, \Big(\alpha^*,\mathsf{V}'_1(x,\alpha;s^*)\Big)\in\mathsf{ACC}\Big] - \Pr_{\alpha^*,s^*}\Big[\Big(\alpha^*,\mathsf{V}'_1(x^*,\alpha;s^*)\Big)\in\mathsf{ACC}\Big]\cdot\mathsf{negl}[2^{2n}] \tag{4}$$

$$\geq \Pr_{k,\alpha^*,s^*,\mathsf{iO},\mathcal{A}}\Big[\mathcal{A}\Big(\mathsf{iO}\big(\mathbf{C}_{\mathsf{pct}}[\mathsf{k}\{\alpha^*\},\alpha^*,\mathsf{V}'_1(x^*,\alpha;s^*)]\big)\Big)=\alpha^* \,\Big|\, \Big(\alpha^*,\mathsf{V}'_1(x,\alpha;s^*)\Big)\in\mathsf{ACC}\Big] - 2^{-2n} \tag{5}$$

$$\geq \frac{1}{8}\cdot 2^{-n}\frac{\nu^2}{\mu'} - 2^{-2n} \geq \frac{1}{16}\cdot 2^{-n}\frac{\nu^2}{\mu'}, \tag{6}$$

where $\mathbf{C}_{\mathsf{hide}}[k,\mathbf{B}]$ is a circuit that defined as follows

```
$\mathbf{C}_{\mathsf{hide}}[k,\mathbf{B}](\alpha^*)$
$s^* := \mathbf{B}(\alpha^*)$
if $s^* = \bot$
    $s^* := \mathsf{PRF}_k(\alpha^*)$
$\beta^* := \mathsf{V}'_1(x^*,\alpha^*;s^*)$
return $\beta^*$
```

Equation 4 follows by reduction to the 2^{2n} security of the indistinguishability obfuscator as depicted in Fig. 2. Clearly, the two circuits are functionally equivalent. Further, if it holds that $(\alpha^*,\beta^*)\in\mathsf{ACC}$ then the two cases of the security definition of indistinguishability obfuscation directly correspond to the two cases of Eq. 4. The reduction $\mathcal{B}^{\mathsf{iO}}$ runs in time $\mathcal{O}2^{2n}$ and therefore, Eq. 4 follows. Equation 5 then follows simply by upper bounding the probability with 1 and the

```
$\mathcal{B}_1^{\mathsf{iO}}(1^n)$
$k \leftarrow_\$ \{0,1\}^{\kappa(n)} \alpha^* \leftarrow_\$ \{0,1\}^n$
$s^* \leftarrow_\$ \{0,1\}^n$
$\mathsf{k}\{\alpha^*\} := \mathsf{Puncture}(k, \alpha^*)$
$\beta^* := \mathsf{V}_1'(x^*, \alpha^*; s^*)$
$\mathbf{C}_0 = \mathbf{C}_{\mathsf{pct}}[\mathsf{k}\{\alpha^*\}, \alpha^*, \beta^*]$
$\mathbf{C}_1 = \mathbf{C}_{\mathsf{hide}}[k, \mathsf{hideO}(\alpha^*, s^*)]$
return $(\mathbf{C}_0, \mathbf{C}_1)$
```

```
$\mathcal{B}_2^{\mathsf{iO}}(\mathbf{B})$
if $(\alpha^*, \beta^*) \notin \mathsf{ACC}$
    $b \leftarrow_\$ \{0,1\}$
    return $b$
else if $\mathcal{A}(\mathbf{B}) = \alpha^*$
    return $0$
else
    return $1$
```

Fig. 2. The reduction from the claim of Eq. 4 to the 2^{2n} security of the indistinguishability obfuscator.

negligible function by 2^{-2n}. Finally Eq. 6 follows directly from Claim 7 and the last inequality follows by loosely upper bounding the negligible function 2^{-2n}.

Closely following [36], it remains to be shown that the distribution defined by uniformly sampling (α^*, β^*) from ACC is close to the distribution defined by uniformly sampling α^* and then sampling β^* conditioned on $(\alpha^*, \beta^*) \in \mathsf{ACC}$.

Formally, we define two distributions. Let D_0 be the distribution over pairs (α^*, β^*) defined by uniformly sampling $(\alpha^*, \beta^*) \leftarrow_\$ \mathsf{ACC}$. Let D_1 be the distribution over pairs (α^*, β^*) defined by uniformly sampling $\alpha^* \leftarrow_\$ \{0,1\}^n$ and then uniformly sampling $\beta^* \leftarrow_\$ \{\beta | (\alpha^*, \beta) \in \mathsf{ACC}\}$. We denote by $\mathsf{D}_b[\alpha^*, \beta^*]$ the probability of the pair (α^*, β^*) by distribution D_b.

Claim 8. *For any* $(\alpha^*, \beta^*) \in \{0,1\}^n \times \{0,1\}^{2n}$ *it holds that*

$$\mathsf{D}_1[\alpha^*, \beta^*] \geq \frac{\nu}{4} \mathsf{D}_0[\alpha^*, \beta^*]$$

It follows from Claim 8 that by drawing from D_1 instead of D_0, the probability of $\mathcal{A}$ outputting α^* can decrease at most by a multiplicative factor of $4/\nu$. Therefore, Claim 8 and Eq. 6 imply that there exists a PPT algorithm $\mathcal{A}$ such that

$$\Pr_{(\alpha^*,\beta^*,\mathsf{hideO},\mathcal{A}) \leftarrow_\$ \mathsf{D}_1,\mathsf{hideO},\mathcal{A}}[\mathcal{A}(\mathsf{hideO}(\mathbf{C}_{\mathsf{hide}}[\alpha^*, \beta^*])) = \alpha^*]$$

$$\geq \frac{\nu}{4} \cdot (\frac{1}{16} \cdot 2^{-n} \cdot \frac{\nu^2}{\mu'}) = \frac{1}{64} 2^{-n} \cdot \frac{\nu^3}{\mu'} \geq \mu^{-1} \cdot 2^{-n} \cdot \frac{\nu^3}{128}$$

Since the distribution of α^* drawn from D_1 is uniform, and ν is an inverse polynomial, this contradicts the $T = \mu \cdot 2^n \cdot \mathsf{poly}(n)$ security of the input hiding obfuscator and Lemma 4 follows. □

It remains to show that the various claims used in the above proof actually hold. The proofs for these claims are detailed in the following sections.

3.1.1 Proof of Claim 6

By definition of $\mathcal{A}$ we specifically need to show that

$$\Pr_{k,\mathsf{iO},\mathsf{P}^*}[(\alpha, \mathsf{V}_1'(x^*, \alpha; \mathsf{PRF}_k(\alpha))) \in \mathsf{ACC} : (\alpha, \gamma) \leftarrow \mathsf{P}^*(\mathsf{iO}(\mathbf{C}_{\mathsf{V}_1}[k, x^*]))] \geq \frac{\nu}{2} - 2^{-n}.$$

To do so, we will use the following claim, stating that if the cheating prover is successful in getting V_2' to accept using the random tape $\mathsf{PRF}_k(\alpha)$, then V_2' would accept with almost the same probability if the random tape were replaced with a randomly chosen $s \leftarrow_\$ S_{\alpha,\beta}$.

Claim 9. *If* PRF *is* 2^{2n}*-secure and* iO *is* 2^{2n}*-secure, then it must hold for any malicious prover* P^* *as assumed above, that*

$$\left| \begin{array}{l} \Pr\limits_{k,\mathsf{iO},\mathsf{P}^*}\left[\mathsf{V}_2'(x^*, \alpha^*, \gamma^*; \mathsf{PRF}_k(\alpha^*)) = 1 : \begin{array}{c} (\alpha^*, \gamma^*) \leftarrow \mathsf{P}^*(\mathsf{iO}(\mathbf{C}_{\mathsf{V}_1}[k, x^*])) \\ \beta^* \leftarrow \mathsf{V}_1'(x^*, \alpha^*; \mathsf{PRF}_k(\alpha^*)) \end{array}\right] \\ - \Pr\limits_{k,s,\mathsf{iO},\mathsf{P}^*}\left[\mathsf{V}_2'(x^*, \alpha^*, \gamma^*; s') = 1 : \begin{array}{c} (\alpha^*, \gamma^*) \leftarrow \mathsf{P}^*(\mathsf{iO}(\mathbf{C}_{\mathsf{V}_1}[k, x^*])) \\ \beta^* \leftarrow \mathsf{V}_1'(x^*, \alpha^*; \mathsf{PRF}_k(\alpha^*)) \\ s' \leftarrow_\$ S_{\alpha^*,\beta^*} \end{array}\right] \end{array} \right| \leq 2^{-n}.$$

We observe the following

$$\nu = \Pr_{k,\mathsf{iO},\mathsf{P}^*}[\mathsf{V}_2'(x^*, \alpha, \gamma; \mathsf{PRF}_k(\alpha)) = 1 : (\alpha, \gamma) \leftarrow \mathsf{P}^*(\mathsf{iO}(\mathbf{C}_{\mathsf{V}_1}[k, x^*]))] \tag{7}$$

$$\leq \Pr_{k,s',\mathsf{iO},\mathsf{P}^*}\left[\mathsf{V}_2'(x^*, \alpha, \gamma; s') = 1 : \begin{array}{c} (\alpha, \gamma) \leftarrow \mathsf{P}^*(\mathsf{iO}(\mathbf{C}_{\mathsf{V}_1}[k, x^*]) \\ \beta \leftarrow \mathsf{V}_1'(x^*, \alpha, \mathsf{PRF}_k(\alpha)) \\ s' \leftarrow_\$ S_{\alpha,\beta} \end{array}\right] + 2^{-n} \tag{8}$$

$$\begin{aligned} = & \underbrace{\Pr_{k,s',\mathsf{iO},\mathsf{P}^*}\left[\mathsf{V}_2'(x^*, \alpha, \gamma; s') = 1 : \begin{array}{c} (\alpha, \gamma) \leftarrow \mathsf{P}^*(\mathsf{iO}(\mathbf{C}_{\mathsf{V}_1}[k, x^*]) \\ \beta \leftarrow \mathsf{V}_1'(x^*, \alpha, \mathsf{PRF}_k(\alpha)) \\ s' \leftarrow_\$ S_{\alpha,\beta} \end{array} \middle| (\alpha, \beta) \in \mathsf{ACC}\right]}_{\leq 1} \\ & \cdot \Pr_{k,\mathsf{iO},\mathsf{P}^*}[(\alpha, \mathsf{V}_1'(x^*, \alpha; \mathsf{PRF}_k(\alpha))) \in \mathsf{ACC} : (\alpha, \gamma) \leftarrow \mathsf{P}^*(\mathsf{iO}(\mathbf{C}_{\mathsf{V}_1}[k, x^*]))] \\ & + \underbrace{\Pr_{k,s',\mathsf{iO},\mathsf{P}^*}\left[\mathsf{V}_2'(x^*, \alpha, \gamma; s') = 1 : \begin{array}{c} (\alpha, \gamma) \leftarrow \mathsf{P}^*(\mathsf{iO}(\mathbf{C}_{\mathsf{V}_1}[k, x^*]) \\ \beta \leftarrow \mathsf{V}_1'(x^*, \alpha, \mathsf{PRF}_k(\alpha)) \\ s' \leftarrow_\$ S_{\alpha,\beta} \end{array} \middle| (\alpha, \beta) \notin \mathsf{ACC}\right]}_{\leq \nu/2} \\ & \cdot \underbrace{\Pr_{k,\mathsf{iO},\mathsf{P}^*}[(\alpha, \mathsf{V}_1'(x^*, \alpha; \mathsf{PRF}_k(\alpha))) \notin \mathsf{ACC} : (\alpha, \gamma) \leftarrow \mathsf{P}^*(\mathsf{iO}(\mathbf{C}_{\mathsf{V}_1}[k, x^*]))]}_{=1-\Pr_{k,\mathsf{iO},\mathsf{P}^*}\left[(\alpha,\mathsf{V}_1'(x^*,\alpha;\mathsf{PRF}_k(\alpha)))\in\mathsf{ACC}:(\alpha,\gamma)\leftarrow\mathsf{P}^*(\mathsf{iO}(\mathbf{C}_{\mathsf{V}_1}[k,x^*]))\right]} \\ & + 2^{-n} \end{aligned} \tag{9}$$

$$\geq \left(1 - \frac{\nu}{2}\right) \Pr_{k,\mathsf{iO},\mathsf{P}^*}[(\alpha, \mathsf{V}'_1(x^*, \alpha; \mathsf{PRF}_k(\alpha))) \in \mathsf{ACC} : (\alpha, \gamma) \leftarrow \mathsf{P}^*(\mathsf{iO}(\mathbf{C}_{\mathsf{V}_1}[k, x^*]))] + \frac{\nu}{2} + 2^{-n} \tag{10}$$

where Eq. 7 follows from the definition of P^* and Eq. 8 follows directly from Claim 9. Equation 9 simply splits the probability into two cases and Eq. 10 upper bounds the probability of the verifier accepting in the two cases.

The above observation gives us

$$\Pr_{k,\mathsf{iO},\mathsf{P}^*}[(\alpha, \mathsf{V}'_1(x^*, \alpha; \mathsf{PRF}_k(\alpha))) \in \mathsf{ACC} : (\alpha, \gamma) \leftarrow \mathsf{P}^*(\mathsf{iO}(\mathbf{C}_{\mathsf{V}_1}[k, x^*]))]$$
$$\geq \frac{\nu - \frac{\nu}{2} - 2^{-n}}{1 - \frac{\nu}{2}} \geq \nu - \frac{\nu}{2} - 2^{-n} = \frac{\nu}{2} - 2^{-n}$$

as claimed. □

Proof of Claim 9. Let δ be any function such that

$$\left| \begin{array}{l} \Pr_{k,s',\mathsf{iO},\mathsf{P}^*}\left[\mathsf{V}'_2(x^*, \alpha^*, \gamma^*; \mathsf{PRF}_k(\alpha^*)) = 1 : \begin{array}{r} (\alpha^*, \gamma^*) \leftarrow \mathsf{P}^*(\mathsf{iO}(\mathbf{C}_{\mathsf{V}_1}[k, x^*])) \\ \beta^* \leftarrow \mathsf{V}'_1(x^*, \alpha^*; \mathsf{PRF}_k(\alpha^*)) \\ s' \leftarrow_{\$} S_{\alpha^*,\beta^*} \end{array}\right] \\ - \Pr_{k,s',\mathsf{iO},\mathsf{P}^*}\left[\mathsf{V}'_2(x^*, \alpha^*, \gamma^*; s') = 1 : \begin{array}{r} (\alpha^*, \gamma^*) \leftarrow \mathsf{P}^*(\mathsf{iO}(\mathbf{C}_{\mathsf{V}_1}[k, x^*])) \\ \mathit{beta}^* \leftarrow \mathsf{V}'_1(x^*, \alpha^*; \mathsf{PRF}_k(\alpha^*)) \\ s' \leftarrow_{\$} S_{\alpha^*,\beta^*} \end{array}\right] \end{array} \right| > \delta(n).$$

In this case, we also have that for a uniformly chosen value α,

$$\left| \begin{array}{l} \Pr_{k,s',\alpha,\mathsf{iO},\mathsf{P}^*}\left[\begin{array}{c} \mathsf{V}'_2(x^*, \alpha^*, \gamma^*; \mathsf{PRF}_k(\alpha^*)) = 1 \\ \wedge\ \alpha^* = \alpha \end{array} : \begin{array}{r} (\alpha^*, \gamma^*) \leftarrow \mathsf{P}^*(\mathsf{iO}(\mathbf{C}_{\mathsf{V}_1}[k, x^*])) \\ \beta^* \leftarrow \mathsf{V}'_1(x^*, \alpha^*; \mathsf{PRF}_k(\alpha^*)) \\ s' \leftarrow_{\$} S_{\alpha^*,\beta^*} \end{array}\right] \\ - \Pr_{k,s',\alpha,\mathsf{iO},\mathsf{P}^*}\left[\begin{array}{c} \mathsf{V}'_2(x^*, \alpha^*, \gamma^*; s') = 1 \\ \wedge\ \alpha^* = \alpha \end{array} : \begin{array}{r} (\alpha^*, \gamma^*) \leftarrow \mathsf{P}^*(\mathsf{iO}(\mathbf{C}_{\mathsf{V}_1}[k, x^*])) \\ \beta^* \leftarrow \mathsf{V}'_1(x^*, \alpha^*; \mathsf{PRF}_k(\alpha^*)) \\ s \leftarrow_{\$} S_{\alpha^*,\beta^*} \end{array}\right] \end{array} \right|$$
$$> 2^{-n} \cdot \delta(n).$$

Now consider the punctured version of the verifier circuit defined as before. By the 2^{2n} security of the obfuscator, the fact that the two circuits $\mathbf{C}_{\mathsf{V}_1}[k, x^*]$ and $\mathbf{C}_{\mathsf{pct}}[\mathsf{k}\{\alpha\}, \alpha, \mathsf{V}'_1(\alpha; \mathsf{PRF}_k(\alpha))]$ are functionally equivalent and the fact that $s \leftarrow_{\$} S_{\alpha^*,\beta^*}$ can be sampled in time $\mathcal{O}(2^n)$, it follows that

$$\left|\begin{array}{l}\displaystyle\Pr_{k,s',\alpha,\mathsf{iO},\mathsf{P}^*}\left[\begin{array}{c}\mathsf{V}'_2(x^*,\alpha^*,\gamma^*;\mathsf{PRF}_k(\alpha^*))=1\\ \wedge\ \alpha^*=\alpha\end{array} : \begin{array}{c}\beta\leftarrow\mathsf{V}'_1(x^*,\alpha;\mathsf{PRF}_k(\alpha))\\ (\alpha^*,\gamma^*)\leftarrow\mathsf{P}^*(\mathsf{iO}(\mathbf{C}_{\mathsf{pct}}[\mathsf{k}\{\alpha\},\alpha,\beta]))\\ s'\leftarrow_{\$}S_{\alpha^*,\beta}\end{array}\right]\\ \displaystyle-\quad\Pr_{k,s',\alpha,\mathsf{iO},\mathsf{P}^*}\left[\begin{array}{c}\mathsf{V}'_2(x^*,\alpha^*,\gamma^*;s')=1\\ \wedge\ \alpha^*=\alpha\end{array} : \begin{array}{c}\beta\leftarrow\mathsf{V}'_1(x^*,\alpha;\mathsf{PRF}_k(\alpha))\\ (\alpha^*,\gamma^*)\leftarrow\mathsf{P}^*(\mathsf{iO}(\mathbf{C}_{\mathsf{pct}}[\mathsf{k}\{\alpha\},\alpha,\beta]))\\ s'\leftarrow_{\$}S_{\alpha^*,\beta}\end{array}\right]\end{array}\right|$$

$$> 2^{-n}\cdot\delta(n)-\mathsf{negl}[2^{2n}].$$

Further, by the 2^{2n} security of the pseudorandom function and the fact that $s\leftarrow_{\$}S_{\alpha^*,\beta^*}$ can be sampled in time $\mathcal{O}(2^n)$, it follows that

$$\left|\begin{array}{l}\displaystyle\Pr_{k,s,s',\alpha,\mathsf{iO},\mathsf{P}^*}\left[\begin{array}{c}\mathsf{V}'_2(x^*,\alpha^*,\gamma^*;s)=1\\ \wedge\ \alpha^*=\alpha\end{array} : \begin{array}{c}\beta\leftarrow\mathsf{V}'_1(x^*,\alpha;s)\\ (\alpha^*,\gamma^*)\leftarrow\mathsf{P}^*(\mathsf{iO}(\mathbf{C}_{\mathsf{pct}}[\mathsf{k}\{\alpha\},\alpha,\beta]))\\ s'\leftarrow_{\$}S_{\alpha^*,\beta}\end{array}\right]\\ \displaystyle-\quad\Pr_{k,s,s',\alpha,\mathsf{iO},\mathsf{P}^*}\left[\begin{array}{c}\mathsf{V}'_2(x^*,\alpha^*,\gamma^*;s')=1\\ \wedge\ \alpha^*=\alpha\end{array} : \begin{array}{c}\beta\leftarrow\mathsf{V}'_1(x^*,\alpha;s)\\ (\alpha^*,\gamma^*)\leftarrow\mathsf{P}^*(\mathsf{iO}(\mathbf{C}_{\mathsf{pct}}[\mathsf{k}\{\alpha\},\alpha,\beta]))\\ s'\leftarrow_{\$}S_{\alpha^*,\beta}\end{array}\right]\end{array}\right|$$

$$\geq 2^{-n}\cdot\delta(n)-\mathsf{negl}(2^{2n})-\mathsf{negl}(2^{2n})\geq 2^{-n}\cdot\delta(n)-2^{-2n},$$

where the last inequality is obtained by loosely upper bounding the negligible functions. The circuit $\mathbf{C}_{\mathsf{pct}}[\mathsf{k}\{\alpha\},\alpha,\beta]$ no longer contains any information about s besides the fact that $s\in S_{\alpha^*,\beta^*}$. In the case where $\alpha^*=\alpha$, s and s' are, therefore, distributed identically and the two probabilities must in fact also be identical. Therefore, $2^{-n}\cdot\delta(n)-2^{-2n}\leq 0$, giving us $\delta(n)\leq 2^{-n}$. The claim thus follows. □

3.1.2 Proof of Claim 7

By definition of conditional probability, we have that

$$\Pr_{k,\alpha^*,s^*,\mathsf{iO},\mathcal{A}}\left[\mathcal{A}\Big(\mathsf{iO}\big(\mathbf{C}_{\mathsf{pct}}[\mathsf{k}\{\alpha^*\},\alpha^*,\mathsf{V}'_1(x^*,\alpha;s^*)]\big)\Big)=\alpha^*\;\Big|\;\Big(\alpha^*,\mathsf{V}'_1(x,\alpha;s^*)\Big)\in\mathsf{ACC}\right]$$

$$=\frac{\Pr_{k,\alpha^*,s^*,\mathsf{iO},\mathcal{A}}\left[\begin{array}{c}\mathcal{A}\Big(\mathsf{iO}\big(\mathbf{C}_{\mathsf{pct}}[\mathsf{k}\{\alpha^*\},\alpha^*,\mathsf{V}'_1(x^*,\alpha;s^*)]\big)\Big)=\alpha^*\\ \wedge\ \Big(\alpha^*,\mathsf{V}'_1(x,\alpha;s^*)\Big)\in\mathsf{ACC}\end{array}\right]}{\Pr_{\alpha^*,s^*}\left[\Big(\alpha^*,\mathsf{V}'_1(x^*,\alpha;s^*)\Big)\in\mathsf{ACC}\right]},$$

where we can easily bound $\Pr_{\alpha^*,s^*}\left[(\alpha^*,\mathsf{V}'_1(x^*,\alpha;s^*))\in\mathsf{ACC}\right]\leq 2\mu'/\nu$ using the soundness error μ' of $\langle\mathsf{P},\mathsf{V}'\rangle$. This is due to the fact that otherwise a (computationally unbounded) malicious prover could simply send a randomly sampled α^*. Upon receiving β^*, it would hold that $(\alpha^*,\beta^*)\in\mathsf{ACC}$ with probability greater than $2\mu'/\nu$. In this case, the prover could exhaustively search for a message γ^* that would lead many verifiers to accept. By definition of ACC, such a prover would win with probability greater than $(2\mu'/\nu) \cdot (\nu/2) = \mu'$, contradicting the soundness of the underlying protocol. It remains to bound the numerator, which we will do in two hops.

$$\Pr_{k,\alpha^*,s^*,\mathsf{iO},\mathcal{A}}\left[\begin{array}{c}\mathcal{A}(\mathsf{iO}(\mathbf{C}_{\mathsf{pct}}[\mathsf{k}\{\alpha^*\},\alpha^*,\mathsf{V}_1'(x^*,\alpha;s^*)])) = \alpha^* \\ \wedge\ (\alpha^*,\mathsf{V}_1'(x^*,\alpha;s^*)) \in \mathsf{ACC}\end{array}\right]$$

$$\geq \Pr_{k,\alpha^*,\mathsf{iO},\mathcal{A}}\left[\begin{array}{c}\mathcal{A}(\mathsf{iO}(\mathbf{C}_{\mathsf{pct}}[\mathsf{k}\{\alpha^*\},\alpha^*,\mathsf{V}_1'(x^*,\alpha^*;\mathsf{PRF}_k(\alpha^*))])) = \alpha^* \\ \wedge\ (\alpha^*,\mathsf{V}_1'(x^*,\alpha^*;\mathsf{PRF}_k(\alpha))) \in \mathsf{ACC}\end{array}\right] - \mathsf{negl}(2^{2n}) \quad (11)$$

$$\geq \Pr_{k,\alpha^*,\mathsf{iO},\mathcal{A}}\left[\begin{array}{c}\mathcal{A}(\mathsf{iO}(\mathbf{C}_{\mathsf{V}_1}[k,x^*])) = \alpha^* \\ \wedge\ (\alpha^*,\mathsf{V}_1'(x^*,\alpha^*;\mathsf{PRF}_k(\alpha^*))) \in \mathsf{ACC}\end{array}\right] - \mathsf{negl}(2^{2n}) - \mathsf{negl}(2^{2n}) \quad (12)$$

$$\geq \Pr_{k,\alpha^*,\mathsf{iO},\mathcal{A}}\left[\begin{array}{c}\mathcal{A}(\mathsf{iO}(\mathbf{C}_{\mathsf{V}_1}[k,x^*])) = \alpha^* \\ \wedge\ (\alpha^*,\mathsf{V}_1'(x^*,\alpha^*;\mathsf{PRF}_k(\alpha^*))) \in \mathsf{ACC}\end{array}\right] - 2^{-2n} \quad (13)$$

Equation 11 follows by reduction to the 2^{2n} security of the puncturable pseudorandom function as depicted in Fig. 3. Clearly, the two cases of the security definition for puncturable pseudorandom functions directly map to the two cases of Eq. 11. Further, the reduction $\mathcal{B}^{\mathsf{PRF}}$ runs in time $\mathcal{O}(2^{2n})$ and therefore, Eq. 11 follows.

```
$\mathcal{B}^{\mathsf{PRF}}(\mathsf{k}\{\alpha^*\}, s^*)$

$\beta^* := \mathsf{V}_1'(x^*, \alpha; s^*)$
$\mathbf{B} \leftarrow \mathsf{iO}(\mathbf{C}_{\mathsf{pct}}[\mathsf{k}\{\alpha^*\}, \alpha^*, \beta^*])$
if $\mathcal{A}(\mathbf{B}) = \alpha^* \wedge (\alpha^*, \beta^*) \in \mathsf{ACC}$
    return 1
else return 0
```

Fig. 3. The reduction from the claim of Eq. 11 to the 2^{2n} security of the puncturable pseudorandom function.

Equation 12 follows by reduction to the 2^{2n} security of the indistinguishability obfuscator as depicted in Fig. 4. Clearly, the two circuits are functionally equivalent and the two cases of the security definition for puncturable pseudorandom functions directly map to the two cases of Eq. 11. The reduction $\mathcal{B}^{\mathsf{iO}}$ runs in time $\mathcal{O}(2^{2n})$ and therefore, Eq. 12 follows. Finally, Eq. 13 then follows by the fact that the sum of two negligible functions is negligible and by loosely upper bounding the resulting negligible functions (note that 2^{-2n} is an inverse polynomial in 2^{2n}).

```
$\mathcal{B}_1^{\mathsf{iO}}(1^n)$
$k \leftarrow_\$ \{0,1\}^{\kappa(n)},\ \alpha^* \leftarrow_\$ \{0,1\}^n$
$\mathsf{k}\{\alpha^*\} := \mathsf{Puncture}(k, \alpha^*)$
$\beta^* := \mathsf{V}_1'(x^*, \alpha^*; \mathsf{PRF}_k(\alpha^*))$
$\mathbf{C}_0 = \mathbf{C}_{\mathsf{pct}}[\mathsf{k}\{\alpha^*\}, \alpha^*, \beta^*]$
$\mathbf{C}_1 = \mathbf{C}_{\mathsf{V}_1}[k, x^*]$
return $(\mathbf{C}_0, \mathbf{C}_1)$
```

```
$\mathcal{B}_2^{\mathsf{iO}}(\mathbf{B})$
if $\mathcal{A}(\mathbf{B}) = \alpha^* \wedge (\alpha^*, \beta^*) \in \mathsf{ACC}$
    return 1
else return 0
```

Fig. 4. The reduction from the claim of Eq. 12 to the 2^{2n} security of the indistinguishability obfuscator.

Using basic probability theory and Claim 6, we get

$$\begin{aligned}
&\Pr_{k,\alpha^*,\mathsf{iO},\mathcal{A}}[\mathcal{A}(\mathsf{iO}(\mathbf{C}_{\mathsf{V}_1}[k,x^*])) = \alpha^* \wedge (\alpha^*, \mathsf{V}_1'(x^*, \alpha^*; \mathsf{PRF}_k(\alpha^*))) \in \mathsf{ACC}] \\
= &\Pr_{k,\alpha^*,\mathsf{iO},\mathcal{A}}\left[\bigcup_\alpha \begin{pmatrix} \mathcal{A}(\mathsf{iO}(\mathbf{C}_{\mathsf{V}_1}[k,x])) = \alpha^* \\ \wedge (\alpha^*, \mathsf{V}_1'(x^*, \alpha^*; \mathsf{PRF}_k(\alpha^*))) \in \mathsf{ACC} \wedge \alpha^* = \alpha \end{pmatrix}\right] \\
= &\sum_\alpha \Pr_{k,\alpha^*,\mathsf{iO},\mathcal{A}}\begin{bmatrix} \mathcal{A}(\mathsf{iO}(\mathbf{C}_{\mathsf{V}_1}[k,x])) = \alpha^* \\ \wedge (\alpha^*, \mathsf{V}_1'(x^*, \alpha^*; \mathsf{PRF}_k(\alpha^*))) \in \mathsf{ACC} \wedge \alpha^* = \alpha \end{bmatrix} \\
= &2^{-n} \sum_\alpha \Pr_{k,\alpha^*,\mathsf{iO},\mathcal{A}}\begin{bmatrix} \mathcal{A}(\mathsf{iO}(\mathbf{C}_{\mathsf{V}_1}[k,x])) = \alpha^* \\ \wedge\ (\alpha^*, \mathsf{V}_1'(x^*, \alpha^*; \mathsf{PRF}_k(\alpha^*))) \in \mathsf{ACC} \end{bmatrix} \\
= &2^{-n} \Pr_{k,\mathsf{iO},\mathcal{A}}[(\alpha, \mathsf{V}_1'(x^*, \alpha; \mathsf{PRF}_k(\alpha))) \in \mathsf{ACC} : \alpha \leftarrow \mathcal{A}(\mathsf{iO}(\mathbf{C}_{\mathsf{V}_1}[k,x^*]))] \\
\geq &2^{-n} \cdot \left(\frac{\nu}{2} - 2^{-n}\right).
\end{aligned}$$

Combining this with Eq. 13, we get

$$\begin{aligned}
&\Pr_{k,\alpha^*,s^*,\mathsf{iO},\mathcal{A}}\begin{bmatrix} \mathcal{A}(\mathsf{iO}(\mathbf{C}_{\mathsf{pct}}[\mathsf{k}\{\alpha^*\}, \alpha^*, \mathsf{V}_1'(x^*, \alpha; s^*)])) = \alpha^* \\ \wedge\ (\alpha^*, \mathsf{V}_1'(x^*, \alpha; s^*)) \in \mathsf{ACC} \end{bmatrix} \\
\geq &2^{-n}\left(\frac{\nu}{2} - 2^{-n}\right) - 2^{-2n} = 2^{-n}\left(\frac{\nu}{2} - 2^{-n} - 2^{-n}\right) \geq 2^{-n} \cdot \frac{\nu}{4}
\end{aligned}$$

where the last inequality follows by loosely upper bounding the negligible function 2^{1-n} by the inverse polynomial $\nu/4$. Finally Claim 7 follows by

$$\begin{aligned}
&\Pr_{k,\alpha^*,s^*,\mathsf{iO},\mathcal{A}}\left[\mathcal{A}\Big(\mathsf{iO}\big(\mathbf{C}_{\mathsf{pct}}[\mathsf{k}\{\alpha^*\},\alpha^*,\mathsf{V}_1'(x^*,\alpha;s^*)]\big)\Big)=\alpha^*\;\Big|\;\Big(\alpha^*,\mathsf{V}_1'(x^*,\alpha;s^*)\Big)\in\mathsf{ACC}\right]\\
&=\frac{\Pr_{k,\alpha^*,s^*,\mathsf{iO},\mathcal{A}}\left[\begin{array}{c}\mathcal{A}\Big(\mathsf{iO}\big(\mathbf{C}_{\mathsf{pct}}[\mathsf{k}\{\alpha^*\},\alpha^*,\mathsf{V}_1'(x^*,\alpha;s^*)]\big)\Big)=\alpha^*\\ \wedge\;\Big(\alpha^*,\mathsf{V}_1'(x^*,\alpha;s^*)\Big)\in\mathsf{ACC}\end{array}\right]}{\Pr_{\alpha^*,s^*}\left[\Big(\alpha^*,\mathsf{V}_1'(x^*,\alpha;s^*)\Big)\in\mathsf{ACC}\right]}\\
&\geq\frac{2^{-n}\cdot\frac{\nu}{2}}{\mu'}=\frac{1}{2}\cdot 2^{-n}\frac{\nu}{\mu'}
\end{aligned}$$

□

3.1.3 Proof of Claim 8

For any α^* denote by $B_{\alpha^*} := \{\beta | (\alpha^*, \beta) \in \mathsf{ACC}\}$. By construction of the relaxed verifier we have that B_{α^*} contains at least a μ/ν fraction of all β. On the other hand, soundness of the protocol $\langle \mathsf{P}, \mathsf{V}' \rangle$ guarantees, that B_α does not contain more than a $2\mu'/\nu \leq 4\mu/\nu^2$ fraction of all β. Thus, we have

$$\frac{\mu}{\nu} \leq \frac{|B_{\alpha^*}|}{2^{2n}} \leq \frac{4\mu}{\nu^2}$$

In particular, for any α and α^*, we have that

$$|B_\alpha| \geq \frac{\nu}{4}|B_{\alpha^*}|$$

which gives us

$$\mathsf{D}_0[\alpha^*, \beta^*] = \frac{1}{\mathsf{ACC}} = \frac{1}{\sum_{\alpha\in\{0,1\}^n}|B_\alpha|} \leq \frac{2}{2^n \cdot |B_{\alpha^*}|} = \frac{4}{\nu}\mathsf{D}_1[\alpha^*, \beta^*].$$

□

3.2 Proof of Lemma 5

Consider the following malicious verifier $\mathsf{V}^* = (\mathsf{V}_1^*, \mathsf{V}_2^*)$. The first stage V_1^* on input the statement x, the prover's first message α and auxiliary input aux simply interprets the auxiliary input as a circuit, evaluates it on x, α, and outputs the result $\beta \leftarrow \mathsf{aux}(x, \alpha)$. The second stage V_2^* on input the statement x, the prover's messages α, γ and auxiliary input aux recomputes $\beta \leftarrow \mathsf{aux}(x, \alpha)$ and then simply outputs α, β, γ.

Now, assume towards contradiction, that Π is zero-knowledge, i.e., in particular for V^* as described above there exists a PPT simulator Sim such that for all PPT distinguishers $\mathcal{D}$, all auxiliary inputs aux and aux', and all statements x it holds that

$$\left|\begin{array}{l}\quad\Pr[\mathcal{D}(\langle\mathsf{P}(x,w),\mathsf{V}^*(x,\mathsf{aux})\rangle,\mathsf{aux}')=1]\\ -\Pr[\mathcal{D}(\mathsf{Sim}(x,\mathsf{aux}),\mathsf{aux}')=1]\end{array}\right| \leq \mathsf{negl}(n).$$

We will use said simulator to construct a malicious prover P^* against $\bar{\Pi}$ as follows: On input x and the verifier's message $\mathbf{B} = \mathsf{iO}(\mathbf{C}_{\mathsf{V}_1}[k])$, P^* invokes the simulator Sim on x and auxiliary input $\mathbf{B}$. The simulator will produce a transcript α, β, γ that P^* also outputs.

If $x \in \mathcal{L}$, then the zero-knowledge property and the completeness guarantee that $\bar{\mathsf{V}}_2$ will accept the proof with probability $1 - \mathsf{negl}(n)$, since otherwise we could easily construct a successful distinguisher against Sim as follows. The distinguisher $\mathcal{D}$ on input (α, β, γ) and auxiliary input aux' simply runs $\bar{\mathsf{V}}_2$ on (α, γ) and random coins aux' and outputs $b \leftarrow \bar{\mathsf{V}}_2((\alpha, \gamma); \mathsf{aux}')$. Further, even if $x \notin \mathcal{L}$, $\bar{\mathsf{V}}_2$ must still accept with all but negligible probability, since otherwise the combination of P^* and $\bar{\mathsf{V}}$ could be used to decide $\mathcal{L}$, implying that $\mathcal{L} \in \mathsf{BPP}$.

Therefore, P^* succeeds in convincing $\bar{\mathsf{V}}$ of false statements with all but negligible probability. Since this contradicts the premise that $\bar{\mu} \leq \mathsf{negl}(n)$, Sim cannot exist and therefore Π is not zero-knowledge. □

4 Extending the Lower Bound to ϵ-Zero Knowledge

In [14] Bitansky et al. introduced a weaker notion of zero-knowledge they called ϵ-zero-knowledge. In this weaker notion, the outputs of the simulator may be distinguishable with non-negligible probability, but the distinguishing advantage is upper bounded by any inverse monomial in the length of the statement. In this section we prove that our lower bound extends to this weaker notion of zero-knowledge. This is particularly interesting because Bitansky et al. [14] are able to construct a 4-round ϵ-zero-knowledge proof protocol from keyless multi-collision-resistant hash functions (MCRH). This provides evidence that our technique is unlikely to be extend to the case of 4-round proofs, since that would rule out MCRHs.

We start by defining ϵ-zero-knowledge. The definition is almost identical to regular zero-knowledge, except that the advantage of the distinguisher is not bounded by a negligible function.

Definition 8 (Non-uniform ϵ-Zero-Knowledge with Auxiliary Input). *Let $\langle \mathsf{P}, \mathsf{V} \rangle$ be a 2-Party protocol. $\langle \mathsf{P}, \mathsf{V} \rangle$ is said to be non-uniformly ϵ-zero-knowledge with auxiliary input, if for all (possibly malicious)* PPT *algorithms* V^* *there exists a* PPT *simulator* Sim*, such that for all* PPT *distinguishers $\mathcal{D}$ and all auxiliary inputs* aux *and* aux'*, it holds that for all statements x with $|x| = \lambda$ and every noticeable function $\epsilon(\lambda) = \lambda^{-\mathcal{O}(1)}$*

$$\left| \begin{array}{l} \quad \Pr\left[\mathcal{D}(\langle \mathsf{P}(x, w), \mathsf{V}^*(x, \mathsf{aux}) \rangle, \mathsf{aux}') = 1\right] \\ - \Pr\left[\mathcal{D}(\mathsf{Sim}(1^{1/\epsilon(\lambda)}, x, \mathsf{aux}), \mathsf{aux}') = 1\right] \end{array} \right| \leq \epsilon(\lambda).$$

Next, we state our generalized lemma about 3-round ϵ-zero-knowledge proofs. This lemma is a straightforward adaption of Lemma 5 to the ϵ-zero-knowledge case.

Lemma 10. *Let Π be a 3-round interactive proof system for a language $\mathcal{L} \notin$ BPP. Let $\bar{\Pi}$ be the transformed 2-round argument system described in Fig. 1 with soundness error $\bar{\mu}$. If $\bar{\mu} \leq \mathsf{negl}(n)$ then Π is not non-uniformly ϵ-zero-knowledge with auxiliary input.*

From combining Lemmas 4 and 10 a statement equivalent to Theorem 2 for ϵ-zero-knowledge follows as a simple corollary.

4.1 Proof of Lemma 10

Just like the lemma itself, the proof is a straightforward adaption of the proof for Lemma 5. We only need to make sure that the weaker requirement on the simulator does not cause the success probability of the cheating prover to deteriorate too much.

Consider the following malicious verifier $\mathsf{V}^* = (\mathsf{V}_1^*, \mathsf{V}_2^*)$. The first stage V_1^* on input the statement x, the prover's first message α and auxiliary input aux simply interprets the auxiliary input as a circuit, evaluates it on x, α, and outputs the result $\beta \leftarrow \mathsf{aux}(x, \alpha)$. The second stage V_2^* on input the statement x, the prover's messages α, γ and auxiliary input aux recomputes $\beta \leftarrow \mathsf{aux}(x, \alpha)$ and then simply outputs α, β, γ.

Now, assume towards contradiction, that Π is ϵ-zero-knowledge, i.e., in particular for V^* as described above there exists a PPT simulator Sim such that for all PPT distinguishers $\mathcal{D}$, all auxiliary inputs aux and aux', all statements x and all noticeable function $\epsilon(\lambda) = \lambda^{-\mathcal{O}(1)}$ it holds that

$$\left| \begin{array}{l} \Pr[\mathcal{D}(\langle \mathsf{P}(x, w), \mathsf{V}^*(x, \mathsf{aux}) \rangle, \mathsf{aux}') = 1] \\ - \Pr\left[\mathcal{D}(\mathsf{Sim}(1^{1/\epsilon(|x|)}, x, \mathsf{aux}), \mathsf{aux}') = 1\right] \end{array} \right| \leq \epsilon(|x|).$$

We will use said simulator to construct a malicious prover P^* against $\bar{\Pi}$ as follows: On input x and the verifier's message $\mathbf{B} = \mathsf{iO}(\mathbf{C}_{\mathsf{V}_1}[k])$, P^* invokes the simulator Sim on $1/\epsilon(\lambda)$, x and auxiliary input $\mathbf{B}$. The simulator will produce a transcript α, β, γ that P^* also outputs.

If $x \in \mathcal{L}$, then the zero-knowledge property guarantees that $\bar{\mathsf{V}}_2$ will accept the proof with probability greater than $1 - |x|^{-c}$ for any constant $c \in \mathbb{N}$, since otherwise we could easily construct a successful distinguisher against Sim as follows. The distinguisher $\mathcal{D}$ on input (α, β, γ) and auxiliary input aux' simply runs $\bar{\mathsf{V}}_2$ on (α, γ) and random coins aux' and outputs $b \leftarrow \bar{\mathsf{V}}_2((\alpha, \gamma); \mathsf{aux}')$. This distinguisher would therefore be able to distinguish between a real transcript and a simulated transcript with probability greater than $|x|^{-c}$ for some constant $c \in \mathbb{N}$, thus clearly clearly contradicting the fact that Sim is a valid simulator. Further, even if $x \notin \mathcal{L}$, $\bar{\mathsf{V}}_2$ must still accept with probability at least $1 - |x|^{-c} - \mathsf{negl}(n)$, since otherwise the combination of P^* and $\bar{\mathsf{V}}$ could be used to decide $\mathcal{L}$, implying that $\mathcal{L} \in \mathsf{BPP}$.

Therefore, P^* succeeds in convincing $\bar{\mathsf{V}}$ of false statements with probability greater than $1 - |x|^{-c} - \mathsf{negl}(n)$ for any constant $c \in \mathbb{N}$, which is clearly non-negligible. Since this contradicts the premise that $\bar{\mu} \leq \mathsf{negl}(n)$, Sim cannot exist and therefore Π is not zero-knowledge. □

References

1. Ananth, P., Jain, A.: On secure two-party computation in three rounds. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017. LNCS, vol. 10677, pp. 612–644. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70500-2_21
2. Ananth, P., Jain, A., Sahai, A.: Indistinguishability obfuscation from functional encryption for simple functions. Cryptology ePrint Archive, Report 2015/730 (2015). http://eprint.iacr.org/2015/730
3. Ananth, P., Sahai, A.: Projective arithmetic functional encryption and indistinguishability obfuscation from degree-5 multilinear maps. In: Coron, J.-S., Nielsen, J.B. (eds.) EUROCRYPT 2017. LNCS, vol. 10210, pp. 152–181. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56620-7_6
4. Applebaum, B., Brakerski, Z.: Obfuscating circuits via composite-order graded encoding. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015. LNCS, vol. 9015, pp. 528–556. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46497-7_21
5. Barak, B., Bitansky, N., Canetti, R., Kalai, Y.T., Paneth, O., Sahai, A.: Obfuscation for evasive functions. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 26–51. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54242-8_2
6. Barak, B., Goldreich, O., Goldwasser, S., Lindell, Y.: Resettably-sound zero-knowledge and its applications. In: 42nd Annual Symposium on Foundations of Computer Science, pp. 116–125. IEEE Computer Society Press, Las Vegas, 14–17 October 2001
7. Barak, B., Goldreich, O., Impagliazzo, R., Rudich, S., Sahai, A., Vadhan, S., Yang, K.: On the (im)possibility of obfuscating programs. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 1–18. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_1
8. Bellare, M., Jakobsson, M., Yung, M.: Round-optimal zero-knowledge arguments based on any one-way function. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 280–305. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-69053-0_20
9. Berman, I., Degwekar, A., Rothblum, R.D., Vasudevan, P.N.: Multi collision resistant hash functions and their applications. Cryptology ePrint Archive, Report 2017/489 (2017). http://eprint.iacr.org/2017/489
10. Bitansky, N., Brakerski, Z., Kalai, Y., Paneth, O., Vaikuntanathan, V.: 3-message zero knowledge against human ignorance. In: Hirt, M., Smith, A. (eds.) TCC 2016. LNCS, vol. 9985, pp. 57–83. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53641-4_3
11. Bitansky, N., Canetti, R.: On strong simulation and composable point obfuscation. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 520–537. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14623-7_28
12. Bitansky, N., Canetti, R., Paneth, O., Rosen, A.: On the existence of extractable one-way functions. In: Shmoys, D.B. (ed.) 46th Annual ACM Symposium on Theory of Computing, pp. 505–514. ACM Press, New York, 31 May–3 June 2014
13. Bitansky, N., Goldwasser, S., Jain, A., Paneth, O., Vaikuntanathan, V., Waters, B.: Time-lock puzzles from randomized encodings. In: Sudan, M. (ed.) ITCS 2016: 7th Innovations in Theoretical Computer Science, pp. 345–356. Association for Computing Machinery, Cambridge, 14–16 January 2016
14. Bitansky, N., Kalai, Y.T., Paneth, O.: Multi-collision resistance: a paradigm for keyless hash functions. Cryptology ePrint Archive, Report 2017/488 (2017). http://eprint.iacr.org/2017/488

15. Bitansky, N., Vaikuntanathan, V.: Indistinguishability obfuscation from functional encryption. In: Guruswami, V. (ed.) 56th Annual Symposium on Foundations of Computer Science, pp. 171–190. IEEE Computer Society Press, Berkeley, 17–20 October 2015
16. Blum, M.: How to prove a theorem so no one else can claim it. In: Proceedings of the International Congress of Mathematicians, vol. 1, p. 2 (1986)
17. Boneh, D., Waters, B.: Constrained pseudorandom functions and their applications. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013. LNCS, vol. 8270, pp. 280–300. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-42045-0_15
18. Boyle, E., Goldwasser, S., Ivan, I.: Functional signatures and pseudorandom functions. In: Krawczyk, H. (ed.) PKC 2014. LNCS, vol. 8383, pp. 501–519. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54631-0_29
19. Canetti, R., Dakdouk, R.R.: Obfuscating point functions with multibit output. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 489–508. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78967-3_28
20. Canetti, R., Goldreich, O., Goldwasser, S., Micali, S.: Resettable zero-knowledge (extended abstract). In: 32nd Annual ACM Symposium on Theory of Computing, pp. 235–244. ACM Press, Portland, 21–23 May 2000
21. Cohen, A., Holmgren, J., Nishimaki, R., Vaikuntanathan, V., Wichs, D.: Watermarking cryptographic capabilities. In: Wichs, D., Mansour, Y. (eds.) 48th Annual ACM Symposium on Theory of Computing, pp. 1115–1127. ACM Press, Cambridge, 18–21 June 2016
22. Dwork, C., Naor, M., Reingold, O., Stockmeyer, L.J.: Magic functions. In: 40th Annual Symposium on Foundations of Computer Science, pp. 523–534. IEEE Computer Society Press, New York, 17–19 October 1999
23. Feige, U., Shamir, A.: Witness indistinguishable and witness hiding protocols. In: 22nd Annual ACM Symposium on Theory of Computing, pp. 416–426. ACM Press, Baltimore, 14–16 May 1990
24. Fiat, A., Shamir, A.: How to prove yourself: practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987). https://doi.org/10.1007/3-540-47721-7_12
25. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. In: 54th Annual Symposium on Foundations of Computer Science, pp. 40–49. IEEE Computer Society Press, Berkeley, 26–29 October 2013
26. Garg, S., Miles, E., Mukherjee, P., Sahai, A., Srinivasan, A., Zhandry, M.: Secure obfuscation in a weak multilinear map model. In: Hirt, M., Smith, A. (eds.) TCC 2016. LNCS, vol. 9986, pp. 241–268. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53644-5_10
27. Gentry, C., Lewko, A.B., Sahai, A., Waters, B.: Indistinguishability obfuscation from the multilinear subgroup elimination assumption. In: Guruswami, V. (ed.) 56th Annual Symposium on Foundations of Computer Science, pp. 151–170. IEEE Computer Society Press, Berkeley, 17–20 October 2015
28. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions. J. ACM **33**(4), 792–807 (1986)
29. Goldreich, O., Kahan, A.: How to construct constant-round zero-knowledge proof systems for NP. J. Cryptol. **9**(3), 167–190 (1996)
30. Goldreich, O., Krawczyk, H.: On the composition of zero-knowledge proof systems. SIAM J. Comput. **25**(1), 169–192 (1996)

31. Goldreich, O., Oren, Y.: Definitions and properties of zero-knowledge proof systems. J. Cryptol. **7**(1), 1–32 (1994)
32. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems (extended abstract). In: 17th Annual ACM Symposium on Theory of Computing, pp. 291–304. ACM Press, Providence, 6–8 May 1985
33. Hada, S., Tanaka, T.: On the existence of 3-round zero-knowledge protocols. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 408–423. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0055744
34. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM J. Comput. **28**(4), 1364–1396 (1999)
35. Jain, A., Kalai, Y.T., Khurana, D., Rothblum, R.: Distinguisher-dependent simulation in two rounds and its applications. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part II. LNCS, vol. 10402, pp. 158–189. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63715-0_6
36. Kalai, Y.T., Rothblum, G.N., Rothblum, R.D.: From obfuscation to the security of Fiat-Shamir for proofs. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part II. LNCS, vol. 10402, pp. 224–251. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63715-0_8
37. Katz, J.: Which languages have 4-round zero-knowledge proofs? In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 73–88. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78524-8_5
38. Kiayias, A., Papadopoulos, S., Triandopoulos, N., Zacharias, T.: Delegatable pseudorandom functions and applications. In: Sadeghi, A.R., Gligor, V.D., Yung, M. (eds.) ACM CCS 2013: 20th Conference on Computer and Communications Security, pp. 669–684. ACM Press, Berlin, 4–8 November 2013
39. Komargodski, I., Naor, M., Yogev, E.: Collision resistant hashing for paranoids: dealing with multiple collisions. Cryptology ePrint Archive, Report 2017/486 (2017). http://eprint.iacr.org/2017/486
40. Lepinski, M.: On the existence of 3-round zero-knowledge proofs. Ph.D. thesis, Massachusetts Institute of Technology (2002)
41. Lin, H.: Indistinguishability obfuscation from constant-degree graded encoding schemes. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016, Part I. LNCS, vol. 9665, pp. 28–57. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49890-3_2
42. Lin, H.: Indistinguishability obfuscation from SXDH on 5-linear maps and locality-5 PRGs. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10401, pp. 599–629. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63688-7_20
43. Lin, H., Tessaro, S.: Indistinguishability obfuscation from trilinear maps and block-wise local PRGs. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part I. LNCS, vol. 10401, pp. 630–660. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63688-7_21
44. Lin, H., Vaikuntanathan, V.: Indistinguishability obfuscation from DDH-like assumptions on constant-degree graded encodings. In: Dinur, I. (ed.) 57th Annual Symposium on Foundations of Computer Science, pp. 11–20. IEEE Computer Society Press, New Brunswick, 9–11 October 2016
45. Pass, R., Seth, K., Telang, S.: Indistinguishability obfuscation from semantically-secure multilinear encodings. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 500–517. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44371-2_28

46. Sahai, A., Waters, B.: How to use indistinguishability obfuscation: deniable encryption, and more. In: Shmoys, D.B. (ed.) 46th Annual ACM Symposium on Theory of Computing, pp. 475–484. ACM Press, New York, 31 May–3 June 2014
47. Zimmerman, J.: How to obfuscate programs directly. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 439–467. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46803-6_15