Milder Definitions of Computational Approximability: The Case of Zero-Knowledge Protocols

Mohammad Sadeq Dousti and Rasool Jalili

Data and Network Security Lab, Department of Computer Engineering,
Sharif University of Technology, Tehran, Iran.
{dousti@ce.,jalili@}sharif.edu

Abstract

Many cryptographic primitives—such as pseudorandom generators, encryption schemes, and zero-knowledge proofs—center around the notion of *approximability*. For instance, a pseudorandom generator is an expanding function which on a random seed, *approximates* the uniform distribution. In this paper, we classify different notions of computational approximability in the literature, and provide several new types of approximability. More specifically, we identify two hierarchies of computational approximability: The first hierarchy ranges from *strong* approximability—which is the most common type in the cryptography—to the *weak* approximability—as defined by Dwork *et al.* (FOCS 1999). We define semi-strong, mild, and semi-weak types as well. The second hierarchy, termed K-approximability, is inspired by the ε-approximability of Dwork *et al.* (STOC 1998). K-approximability has the same levels as the first hierarchy, ranging from strong K-approximability to weak K-approximability. While both hierarchies are general and can be used to define various cryptographic constructs with different levels of security, they are best illustrated in the context of zero-knowledge protocols.

Assuming the existence of (trapdoor) one-way permutations, and exploiting the random oracle model, we present a separation between two definitions of zero knowledge: one based on strong K-approximability, and the other based on semi-strong K-approximability. Especially, we present a protocol which is zero knowledge only in the latter sense. The protocol is interesting in its own right, and can be used for efficient identification. Next, we show that our model for zero knowledge was *not* closed under sequential composition, and change the model to resolve this issue. After proving a composition theorem, we finally provide a version of the identification protocol which satisfies the requirements of the new model. Some techniques provided in this paper are of independent interest, such as proving a composition theorem in the presence of both simulator and knowledge extractor.

Keywords: Approximability, Indistinguishability, Zero Knowledge, Random Oracle, Trapdoor One-Way Permutation, Sequential Composition.

1 Introduction

The notion of *computational approximability* can be tracked down to works such as [GM82, Yao82, BM82, GM84, BM84, GMR85], but it was probably the work of Goldwasser, Micali, and Rackoff on zero-knowledge proofs [GMR89, Section 3.2] which explicitly defined the notion. Informally, the output of a probabilistic machine is said to **approximate** a random variable if no polynomial-size circuit can tell them apart. All works which use the notion of *indistinguishability* can be reformulated to use the notion of *approximability* instead. For instance, pseudorandom generators, encryption schemes, and witness-hiding proofs can all be defined in terms of approximability. However, approximability is best illustrated in the context of zero-knowledge protocols. In fact, our research on approximability was initiated while we were exploring less strict models of zero knowledge.

Let us explain the motivation behind the need for looser models of zero knowledge: Over time, some authors proved inherent limitations to the accepted notions of zero-knowledge proofs, most of which were imposed on the round complexity of the proof. For instance, Goldreich, Oren, and Krawczyk [GO94, GK96] proved lower bounds on the round complexity of auxiliary-input and black-box zero-knowledge proofs (with negligible soundness error). To overcome these and similar limitations, less strict models of zero knowledge was suggested. To name just a few examples, Brassard *et al.* [BCC88] put forward the notion of arguments, Barak [Bar01] advocated a non–black-box model of zero knowledge, Dwork and Stockmeyer [DS02] proposed a model where the prover's resources were limited, Pass [Pas03b] suggested permitting the simulator to run in quasi-polynomial time, and Birrell and Vadhan [BV10] modeled the verifier as circuits with bounded non-uniformity.

Most relevant to the present work, several researchers argued that the current formulation of approximability is too strong for some purposes, and consequently proposed weaker notions of approximability. For instance, Dwork, Naor, Reingold, and Stockmeyer [DNRS99, DNRS99] proposed the notions of weak and ultra-weak approximabilities.[1] Let us intuitively compare the current formulation of approximability (which we termed "strong approximability") with the weak variant defined by Dwork *et al.*: Suppose M is a probabilistic polynomial-time machine which is going to approximate a distribution ensemble $\{U(x)\}$ indexed by some set L (i.e. $x \in L$).

1. In *strong* approximability, M should approximate $\{U(x)\}$, such that the output of $M(x)$ is indistinguishable from $U(x)$ by all polynomial-size tests D.

2. In *weak* approximability, the code of M may depend on both x and D, as if we say: Disclose the index and the test, and we will exhibit an approximator which beats the test.

The security guarantees provided by the weak approximability is way too low, as M can *arbitrarily* depend on the code of the adversary. As a matter of fact, weak approximability was not introduced to serve security purposes at all. Therefore, we sought *milder* notions of approximability, which provide better security guarantees than the weak approximability, yet are not as strict as the strong approximability.

More specifically, we consider a hierarchy of successive weakenings of approximability, and put forward three notions of semi-strong, mild, and semi-weak approximabilities. Intuitively, the semi-strong variant assumes that the approximator has black-box access to the distinguisher. Mild approximability requires a universal approximator which may receive the description of the distinguisher as an auxiliary input. Finally, the semi-weak approximability allows the approximator to depend arbitrarily on the distinguisher, but not the index (x).

The security guarantees of the semi-strong approximation is still very high: It does not seem that the one-bit output of the distinguisher is of much help to the approximator.[2] The same holds for the mild approximability: Unless the approximator can "reverse engineer" the description of the distinguisher, it cannot gain insight significantly better than an approximator which merely has black-box access to the distinguisher. That said, we successfully exhibit a separation between the strong and semi-strong approximations in the random-oracle model, assuming the existence of (trapdoor) one-way permutations.

Another hierarchy of approximability is inspired by the work of Dwork, Naor, and Sahai [DNS98, DNS04]. Failing to demonstrate a concurrent zero-knowledge proof with low round complexity (due to some inherent limitations), they promoted a new definition which we term ε-approximability. In this definition, the running time of the approximator can be a polynomial in the running time of the distinguisher, as well as the inverse of the distinguishing gap (ε^{-1}). Applying the same ideas, we provide another hierarchy termed K-approximability, whose levels range from strong K-approximability to weak K-approximability. This hierarchy combines approximators with knowledge extractors, and is somehow weaker than the previous hierarchy. The aforementioned separation actually separates strong and semi-strong K-approximabilities.

The ideas and techniques offered by this paper might be of independent interest, among them is an efficient identification protocol used to separate two notions of approximability, and a new technique for proving a composition theorem in the presence of both simulator and knowledge extractor. However, due to space limitations, most proofs are omitted from this abstract (though they appear in the appendices).

1.1 Motivation

A natural question that may arise is why we weaken the existing definitions. There are several answers to this question:

1. The new definitions are not weaker than all the existing ones; rather, they are stronger than definitions like the *weak* and *ultra weak zero knowledge* defined by Dwork *et al.* [DNRS03]. The weak definitions never found their way into the practice, because the community felt that they are inadequate for everyday protocols and applications. However, such weak definitions are important to the theorists, as they are related to *selective commitment* and *magic functions* (see [DNRS03] for more information).

   This paper provides definitions which are *milder* than the existing ones, i.e. they are stronger than some existing definitions, and weaker than other ones. They might bridge the gap, and provide models which are of interest to both theorists and practitioners.

[1] In fact, they proposed models for weak and ultra-weak zero-knowledge, from which we extracted the corresponding definitions for weak and ultra-weak approximability.

[2] It must be noted, though, that the importance of a single bit should not be underestimated. For instance, a single bit of advice can help compute some uncomputable functions [Gol08, Theorem 1.13]. That said, it is hard to conceive of a *natural* problem in which the one-bit output of the distinguisher is of much help to the approximator (see [Dou11]). In fact, even this paper does not use the one-bit output of the distinguisher; rather, it uses the random-oracle model to force the distinguisher make some queries, and then monitors them.

2. Before the introduction of the notion of *witness-hiding proofs* [FS90], no formal proof for the security of the parallel version of Feige-Fiat-Shamir identification protocol [FFS88] was available. However, the security guarantee of being "witness hiding" is much weaker than being "zero knowledge." Consequently, it is desirable to have definitions based on which a tighter security guarantee is possible. Therefore, weakening strong definitions of zero knowledge is desirable in some cases.

   As a matter of fact, we were able to prove the security of efficient identification protocols (see Protocols 1 and 3) in a rather tight manner. According to previous definitions, these protocols were either deemed insecure, or their security were proven in a loose manner.

3. In models such as the UC Framework [Can01], no useful protocol can be zero knowledge without trusted parties or setup assumptions [CF01]. Weakening the strong definitions of zero knowledge can be a workaround.

4. As discussed in Section 6, the new definitions shed light on an alternative way of replacing a random oracle with a new, suitable cryptographic assumption.

1.2 Organization

The rest of this paper is organized as follows: Section 2 provides abbreviations, conventions, and definitions used in throughout this paper. Section 3 defines two hierarchies of computational approximation. Section 4 exhibits a separation between strong and semi-strong K-approximations in the context of zero-knowledge protocols. Section 5 shows that the definition of zero knowledge provided in section 4 is not closed under sequential compositions. It then resolves the issue by providing a new definition, and concludes by illustrating a protocol which satisfies the new definition. Section 6 provides insights into the future line of research.

2 Preliminaries

2.1 Notions and Abbreviations.

Let $\mathbb{N} = \{0, 1, 2, \ldots\}$ denote the set of natural numbers. For a language L and a number $n \in \mathbb{N}$, define $L_n \overset{\text{def}}{=} L \cap \{0, 1\}^n$. The expected value of a random variable X is denoted by $\mathbb{E}[X]$. We use the quantifier $\forall^\infty$ as a shorthand for "for all but finitely many." For instance, $(\forall^\infty n \in \mathbb{N})[\varphi(n)]$ means "for all but finitely many natural numbers n, the predicate $\varphi(n)$ holds." Formally, $(\exists n_0 \in \mathbb{N})(\forall n \in \mathbb{N})[n \geq n_0 \Rightarrow \varphi(n)]$.

Throughout the paper, we use the following abbreviations: RO and ROM are stand for "random oracle" and "random-oracle model." TM stands for Turing machine, PPT for probabilistic polynomial-time, PPTM for a PPT TM, ITM for interactive Turing machine, and OM for oracle machine. These terms might be combined together; for instance, PPT-OM means a "probabilistic polynomial-time oracle machine." We also use "ZK" for zero knowledge. To denote the type of ZK (see Section 2.2), we use prefixes such as AI (auxiliary input) and BB (black box). Therefore, AIZK means "auxiliary-input zero knowledge."

A family of circuits $C = \{C_n\}$ is called *polynomial-size* if there exist polynomials $p(\cdot)$ and $q(\cdot)$ such that for all $n \in \mathbb{N}$, the size and the number of inputs of C_n are bounded by $p(n)$ and $q(n)$, respectively. We assume that all circuits are probabilistic.

Convention. When we say a machine is polynomial-time, we mean polynomial in the length of its ***first input***. We may "pair" two or more inputs and feed them as the first input to a machine. For instance, the first input to $M(\langle x, y, z\rangle, w, t)$ is $\langle x, y, z\rangle$. The same convention holds for polynomial-size circuits.

2.2 Definitions

Definition 1 (Trapdoor One-way Permutation)**.** A family of permutations $\mathcal{F} = \{f_n\}$ is called a *collection of trapdoor one-way permutations* if there exist four PPT algorithms GEN, SAMP, EVAL, and INV, such that the following conditions hold:

1. **Easy to generate:** On input 1^n, algorithm GEN generates a description of f_n denoted $\mathsf{desc}(f_n)$, as well as the associated trapdoor t_n. Denote the first (i.e. $\mathsf{desc}(f_n)$) and second (i.e. t_n) components in the output of GEN by GEN_1 and GEN_2, respectively. In order to avoid mentioning 1^n explicitly in the inputs of SAMP, EVAL, and INV, we assume that $|\mathsf{desc}(f_n)| \geq n$.

2. **Easy to sample the domain:** On input $\mathsf{desc}(f_n)$, algorithm $\textsc{Samp}$ chooses an element from $\mathsf{dom}(f_n)$.

3. **Easy to evaluate:** On inputs $\mathsf{desc}(f_n)$ and $x \in \mathsf{dom}(f_n)$, the output of the algorithm $\textsc{Eval}$ is $f_n(x)$.

4. **Easy to invert with trapdoor:** On inputs $\mathsf{desc}(f_n)$, t_n, and $y \in \mathsf{dom}(f_n)$, the output of the algorithm $\textsc{Inv}$ is ${f_n}^{-1}(x)$. (Note that since f_n is a permutation, its range is identical to its domain.)

5. **Hard to invert without trapdoor:** For every family of polynomial-size circuits $\mathcal{A} = \{\mathcal{A}_n\}$, for every $c \in \mathbb{N}$, and for all sufficiently large n:

$$\Pr\Big[\mathsf{desc}\left(f_n\right) \leftarrow \textsc{Gen}_1\left(1^n\right), \quad x \leftarrow \textsc{Samp}\left(\mathsf{desc}\left(f_n\right)\right), \quad y \leftarrow \textsc{Eval}\left(\mathsf{desc}\left(f_n\right), x\right) : \\ \mathcal{A}_n(y, \mathsf{desc}(f_n)) = x\Big] < n^{-c} \quad , \qquad (1)$$

where the probability is taken over the random coins of $\textsc{Gen}$, $\textsc{Samp}$, $\textsc{Eval}$, and $\mathcal{A}$.

Definition 2 (Strong Approximability)**.** A polynomially-bounded distribution ensemble[3] $\{U(x,z)\}_{x\in L, z\in\{0,1\}^*}$ is said to be *strongly approximable* on the language L, if there exists a PPTM $M(\cdot,\cdot)$ such that for every family of polynomial-size circuits $D = \{D_n\}$, the following holds:

$$(\forall c \in \mathbb{N})(\forall^\infty n \in \mathbb{N})(\forall x \in L_n)(\forall z \in \{0,1\}^*) \\ \left|\Pr[D_n(x,z,M(x,z)) = 1] - \Pr[D_n(x,z,U(x,z)) = 1]\right| < n^{-c} \quad , \qquad (2)$$

where the first probability is taken over the random coins of M and D_n, and the second probability is taken over the (implicit) random coins of U and D_n.

Let $\langle P \leftrightarrow V^*(z)\rangle\,(x)$ be a protocol between ITM P and PPT-ITM V^*, where the common input is x and V^* has an auxiliary input z. Define $\mathsf{view}_{V^*} \overset{\text{def}}{=} \mathsf{view}_{V^*}(x,z) \overset{\text{def}}{=} \mathsf{view}_{V^*}\,\langle P \leftrightarrow V^*(z)\rangle\,(x)$ as whatever V^* sees during the interaction with P; that is, the common input (x), the auxiliary input (z), its random tape (r), and the messages it sent and received $\overline{m} = (m_1, m_2, \dots)$.

Definition 3 (Auxiliary-Input Zero Knowledge (AIZK))**.** The protocol $\langle P \leftrightarrow V^*(z)\rangle\,(x)$ is *AIZK* for P on L if for all PPTM V^*, there exists a PPTM simulator S_{V^*} which **strongly approximates** the view of V^*. That is, for every family of polynomial-size circuits $D = \{D_n\}$, the following holds:

$$(\forall c \in \mathbb{N})(\forall^\infty n \in \mathbb{N})(\forall x \in L_n)(\forall z \in \{0,1\}^*) \\ \left|\Pr[D_n(x,z,S_{V^*}(x,z)) = 1] - \Pr[D_n(x,z,\mathsf{view}_{V^*}(x,z)) = 1]\right| < n^{-c} \quad , \qquad (3)$$

where the first probability is taken over the random coins of S_{V^*} and D_n, and the second probability is taken over the random coins of P, V^* and D_n.

Throughout the paper, we are mostly concerned with the notion of "zero knowledge" rather than the notion of the "proof." Therefore, we hardly mention properties such as completeness and soundness, even if the zero-knowledge protocols provide them. Moreover, while the definitions resemble proofs of language membership [GMR89], they can be applied to proofs of knowledge [BG93] or proofs of computational ability [BG92] as well.

3 Two Hierarchies of Approximability

3.1 The First Hierarchy

Let us first present the notion of *weak approximability*, inspired by the *weak zero-knowledge* definition of Dwork *et al.* [DNRS99, DNRS03]:[4]

[3] That is, the output length of the distribution ensemble is bounded by a fixed polynomial in the length of its first input. Note that here x belongs to some language L, while z is arbitrary string, playing the role of an auxiliary input. The order of quantifiers in this definition does not allow x or z to be hard-coded into M's code.

[4] It must be noted, though, that their definition is based on the uniform zero knowledge [Gol93].

Definition 4 (Weak Approximability)**.** A poly-bounded distribution ensemble $U = \{U(x,z)\}_{x \in L, z \in \{0,1\}^*}$ is said to be *weakly approximable* on the language L, if for every family of polynomial-size circuits $D = \{D_n\}$, the following holds:

$$(\forall c \in \mathbb{N})(\forall^\infty n \in \mathbb{N})(\forall x \in L_n)(\forall z \in \{0,1\}^*)(\exists M \in \text{PPTM})$$
$$|\Pr[D_n(x,z,M(x,z)) = 1] - \Pr[D_n(x,z,U(x,z)) = 1]| < n^{-c} \ , \qquad (4)$$

where the first probability is taken over the random coins of M and D_n, and the second probability is taken over the (implicit) random coins of U and D_n.

Note that in Definition 4, the machine M can depend on x, z, and D_n, as well as U. It can be symbolized as $(\forall D)(\forall^\infty x \in L)(\forall z)(\exists M)[D(x,z,M(x,z)) \approx D(x,z,U(x,z))]$, which is interpreted: "You name the test (D_n) and the parameters (x, z), and I will present a machine (M) which approximates U in such a way that the test fails."

Remark 1. A natural but misleading question is the following: "In the *real world*, how can the approximator access the distinguisher?" The important point is that neither approximator nor the distinguisher are *real-world* entities; they are just parts of a thought experiment. The right interpretation is the following: Consider a real-world entity who uses some (internal) procedure to distinguish the distributions. This entity can modify the procedure to produce the right distribution.

The weak approximability seems to be *extremely* loose, and it is natural to think of a tighter notion. One such attempt is made in Definition 5.

Definition 5 (Semi-Weak Approximability)**.** A poly-bounded distribution ensemble $U = \{U(x,z)\}_{x \in L, z \in \{0,1\}^*}$ is said to be *semi-weakly approximable* on the language L, if for every family of polynomial-size circuits $D = \{D_n\}$, the following holds:

$$(\forall c \in \mathbb{N})(\forall^\infty n \in \mathbb{N})(\exists M \in \text{PPTM})(\forall x \in L_n)(\forall z \in \{0,1\}^*)$$
$$|\Pr[D_n(x,z,M(x,z)) = 1] - \Pr[D_n(x,z,U(x,z)) = 1]| < n^{-c} \ , \qquad (5)$$

where the first probability is taken over the random coins of M and D_n, and the second probability is taken over the (implicit) random coins of U and D_n.

Definition 5 allows M to depend on D_n, but not on x or z. It can be interpreted as "You name the test (D_n), and I will present a machine (M) which approximates U in such a way that the test fails." This is still too loose: M is effectively a circuit, not a PPTM. This is because M can depend on the non-uniformity of D_n, and access the same (or even longer) prefix of z that D_n does.

One way to restrain this power is *not* to allow M to depend on D_n *arbitrarily*. To this end, we require that there exists some universal PPTM M which can approximate U on L, but we let M to have *code access* or *black-box access* to D_n. Definitions 6 and 7 capture the new notions:

Definition 6 (Mild Approximability)**.** A poly-bounded distribution ensemble $U = \{U(x,z)\}_{x \in L, z \in \{0,1\}^*}$ is said to be *mildly approximable* on the language L, if there exists a PPTM M, such that for every family of polynomial-size circuits $D = \{D_n\}$, the following holds:

$$(\forall c \in \mathbb{N})(\forall^\infty n \in \mathbb{N})(\forall x \in L_n)(\forall z \in \{0,1\}^*)$$
$$|\Pr[D_n(x,z,M(\langle x, \mathsf{desc}(D_n)\rangle, z)) = 1] - \Pr[D_n(x,z,U(x,z)) = 1]| < n^{-c} \ , \qquad (6)$$

where the first probability is taken over the random coins of M and D_n, and the second probability is taken over the (implicit) random coins of U and D_n. Here, $\mathsf{desc}(D_n)$ means the description of the circuit D_n in some canonical encoding.

Note that since the pair $\langle x, \mathsf{desc}(D_n)\rangle$ is provided as the first input to M, the machine M has enough time to simulate the code of D_n.

Definition 7 (Semi-Strong Approximability)**.** A poly-bounded distribution ensemble $U = \{U(x,z)\}_{x \in L, z \in \{0,1\}^*}$ is said to be *semi-strongly approximable* on the language L, if there exists a PPT-OM M, such that for every family of polynomial-size circuits $D = \{D_n\}$, the following holds:

$$(\forall c \in \mathbb{N})(\forall^\infty n \in \mathbb{N})(\forall x \in L_n)(\forall z \in \{0,1\}^*)$$
$$|\Pr[D_n(x,z,M^{D_n}(x,z)) = 1] - \Pr[D_n(x,z,U(x,z)) = 1]| < n^{-c} \ , \qquad (7)$$

where the first probability is taken over the random coins of M and D_n, and the second probability is taken over the (implicit) random coins of U and D_n.

The interpretation of Definition 6 is: "I will provide an approximator M that, given the code of the test, will approximate U on L such that the test fails." Definition 7 is interpreted similarly; yet instead of accessing the description of the test, the machine M is only allowed black-box access to the test.

Theorem 1. Let us denote the "implication" by $\Rightarrow$. Then:

$$\begin{aligned}&\text{Strong approximability} \Rightarrow \text{Semi-strong approximability} \Rightarrow \text{Mild approximability} \Rightarrow \\ &\text{Semi-weak approximability} \Rightarrow \text{Weak approximability}\ . \end{aligned} \tag{8}$$

Proof. Strong approximability implies semi-strong approximability since, in the latter, the approximator (M) can have black-box access to the distinguisher. If the approximation is possible without such access (as is the case with strong approximability), it is *a fortiori* possible with black-box access.

The same reasoning holds while comparing semi-strong and mild approximabilities: If the approximation is possible with only black-box access (as is the case with semi-strong approximability), it is *a fortiori* possible with code access (as is the case with mild approximability).

Mild approximability implies semi-weak approximability since the order of quantifiers in the definition of the latter allows the approximator to depend *arbitrarily* on the distinguisher.

Semi-weak approximability implies weak approximability because the latter allows the approximator to depend not only on the distinguisher, but also on the common and auxiliary inputs. ■

It is easy to take cryptographic primitives which incorporate "strong approximability" and redefine them based on the new notions of approximability. For instance, consider the definition of "pseudorandom generators": Let $\ell(n) > n$ be a polynomially-bounded function. The function $G\colon \{0,1\}^n \to \{0,1\}^{\ell(n)}$ is called a pseudorandom generator if it is polynomial-time computable, and for a randomly selected $x \in \{0,1\}^n$, the uniform distribution over $\{0,1\}^{\ell(n)}$ is strongly approximated by $G(x)$.[5] It is now easy to define, say, a "mildly pseudorandom generator": The function G is called a *mildly pseudorandom generator* if it is polynomial-time computable, and for a randomly selected $x \in \{0,1\}^n$, the uniform distribution over $\{0,1\}^{\ell(n)}$ is **mildly approximated** by $G(x)$.

Separating Definitions 4–7 from each other and from Definition 2, as well as determining the security implications each provides, seems to be an interesting task. Specifically, it seems hard to separate the semi-strong approximability (Definition 7) from the strong approximability (Definition 2). On the one hand, the approximator M of Definition 7 can perform tests based on the non-uniformity of D_n, something that a PPTM cannot do by itself. On the other hand, the one-bit output of D_n does not seem to offer machine M of Definition 7 any competitive advantage over the machine M of Definition 2. However, in Section 4 we will see a separation between the strong approximability and the semi-strong approximability in the random-oracle model. (In fact, we provide such separation in the context of the second hierarchy; see below.)

3.2 The Second Hierarchy

The second hierarchy we present has more "semantics" attached to it. This hierarchy concerns the *knowledge* which might be encoded into the description of a distinguisher, or given as an external advice to it. The models in this hierarchy allow the approximator to extract the knowledge associated with a particular distinguisher, and then try to approximate the distribution ensemble.

Informally, a TM/circuit P is said to *know* something with probability q if there exists a probabilistic TM K (called the *knowledge extractor*), which runs in expected time bounded by to $1/q$ (up to a polynomial factor), and extracts the knowledge of D. The machine K may have black-box [FFS88, FS90, BG93] or code access [BGGL01, BL02] to P. Depending on how well K extracts the knowledge, one can define **strong** proofs of knowledge [Gol01, Definition 4.7.13], (ordinary) proofs of knowledge [Gol01, Definition 4.7.2], and **weak** proofs of knowledge (an adaption of *weak proofs of ability* defined in [BG92]). The combination of {black-box, code} access, and {strong, ordinary, weak} models provide us with 6 possible ways of defining a hierarchy. Below we will present some natural combination; but let us provide an example first.

Consider a cryptographic protocol, such as an identification scheme. In this protocol, the prover P must prove his knowledge of some secret s (related to his identity) to a verifier V^*. Assume the protocol is defined in such way that it has a special behavior:[6]

[5] The technicality here is that the selection of x from L is not *universally quantified*; instead, x is *randomly* selected from $\{0,1\}^n$. One can easily change the definitions of approximability to cover this case as well.

[6] In Sections 4–5, we will show that there exist protocols with such behavior.

1. Unless V^* "knows" s, she cannot distinguish the real execution from a simulated one.

2. If V^* gets to "know" s, she might be able to distinguish the real and simulated executions.

The question is: "does the second case harm the security of the identification scheme?" After all, if the adversary knows the secret, she can simulate the protocol all by herself, without having to resort to the simulator. We may therefore present an informal definition of *simulatable identification schemes*, as below:

> An identification scheme is *simulatable* if for every PPTM adversary V^*, there exists a simulator S_{V^*} which simulates the view of V^*, in such a way that if the adversary can distinguish the real and simulated views with probability q, we can conclude that she knows the secret of the prover with probability roughly q.

This notion of simulatability is closely related to what Dwork, Naor, and Sahai [DNS98, DNS04] called ε-knowledge, based on which one can define ε-approximability. However, in order to be consistent with the naming convention of the previous hierarchy, we call it "**strong** ε-approximability."

Definition 8 (Strong ε-Approximability)**.** A poly-bounded distribution ensemble $U = \{U(x,z)\}_{x\in L, z\in\{0,1\}^*}$ is said to be *strongly ε-approximable* on the language L, if for all functions $0 < \varepsilon(\cdot) = o(1)$, there exists a PPTM M, such that for every family of polynomial-size circuits $D = \{D_n\}$, the following holds:

$$\begin{aligned}&(\forall c \in \mathbb{N})(\forall^\infty n \in \mathbb{N})(\forall x \in L_n)(\forall z \in \{0,1\}^*)\\ &\qquad |\Pr[D_n(x,z,M(\langle x, 1^{1/\varepsilon(n)}, |D_n|\rangle, z)) = 1] - \Pr[D_n(x,z,U(x,z)) = 1]| < n^{-c} + \varepsilon(n) \enspace , \qquad (9)\end{aligned}$$

where the first probability is taken over the random coins of M and D_n, and the second probability is taken over the (implicit) random coins of U and D_n.

Note that the running time of M can be a polynomial in the size of the distinguisher, as well as the inverse of the distinguishing gap (ε^{-1}). For several years, it was unknown whether zero-knowledge is a stricter concept than ε-knowledge. Barak and Lindell [BL02] showed a separation between the two concepts: While there exist constant-round strict polynomial-time black-box simulator ε-knowledge proofs for NP (with negligible soundness error), such constant-round strict polynomial-time black-box simulator ZK proofs (with negligible soundness error) exist only for BPP languages.[7]

As in the previous section, it is possible to define the hierarchy of *weak ε-approximability*, *semi-weak ε-approximability*, *mild ε-approximability*, and *semi-strong ε-approximability*. However, the goal of this section is to provide a different hierarchy based on the notion of knowledge extraction.

Recall the example about the identification scheme: It was designed such that if the adversary could distinguish the real and simulated executions with probability q, then it would *know* the secret of the prover with probability q. The word "know" is italicized because it is informal. One can formalize this definition by requiring the existence of a knowledge extractor K, such that K accesses the adversary, extracts her knowledge, and *tries to simulate the protocol*. Definitions 9 and 16 formalize this concept.

Definition 9 (Mild K-Approximability)**.** A poly-bounded distribution ensemble $U = \{U(x,z)\}_{x\in L, z\in\{0,1\}^*}$ is said to be *mildly K-approximable* on the language L, if there exists a PPTM M and an expected PPTM K, such that for every family of polynomial-size circuits $D = \{D_n\}$, for all $n \in \mathbb{N}$, for all $x \in L_n$, and for all $z \in \{0,1\}^*$, if the *advantage*

$$\Psi = \mathbf{Adv}_{D_n}^{M,U}(x,z) \stackrel{\text{def}}{=} |\Pr[D_n(x,z,M(x,z)) = 1] - \Pr[D_n(x,z,U(x,z)) = 1]| \qquad (10)$$

is nonzero, then on input (x,z), the following holds:

$$|\Pr[D_n(x,z,K(\langle x, 1^{1/\Psi}, \mathsf{desc}(D_n)\rangle, z)) = 1] - \Pr[D_n(x,z,U(x,z)) = 1]| < n^{-c} \enspace , \qquad (11)$$

where the first probability is taken over the random coins of K and D_n, and the second probability is taken over the (implicit) random coins of U and D_n.

[7] A very recent treatment of this subject can be found in [Gol10].

Definition 10 (Semi-Strong K-Approximability)**.** A poly-bounded distribution ensemble $U = \{U(x,z)\}_{x \in L, z \in \{0,1\}^*}$ is said to be *semi-strongly K-approximable* on the language L, if there exists a PPTM M and an expected PPT-OM K, such that for every family of polynomial-size circuits $D = \{D_n\}$, for all $n \in \mathbb{N}$, for all $x \in L_n$, and for all $z \in \{0,1\}^*$, if $\Psi = \mathbf{Adv}_{D_n}^{M,U}(x,z)$ (defined in (10)) is nonzero, the following holds:

$$|\Pr[D_n(x,z,K^{D_n}(\langle x, 1^{1/\Psi}\rangle, z)) = 1] - \Pr[D_n(x,z,U(x,z)) = 1]| < n^{-c} \ , \tag{12}$$

where the first probability is taken over the random coins of K and D_n, and the second probability is taken over the (implicit) random coins of U and D_n.

As in the first hierarchy, other levels of the K-approximability (strong, semi-weak, and weak K-approximabilities) are conceivable as well. See Appendix A

Theorem 2. Each definition of approximability entails the corresponding variant of K-approximability. Moreover,

$$\begin{aligned}&\text{Strong } K\text{-approximability} \Rightarrow \text{Semi-strong } K\text{-approximability} \Rightarrow \text{Mild } K\text{-approximability} \Rightarrow \\ &\text{Semi-weak } K\text{-approximability} \Rightarrow \text{Weak } K\text{-approximability} \ .\end{aligned} \tag{13}$$

The proof appears in Appendix B.

4 Separating Semi-Strong Approximability from Strong Approximability

In this section, we present a separation between the semi-strong and strong notions of approximability (and K-approximability). In particular, we construct a protocol in the random-oracle model (ROM) [BR93], which is not ZK based on the strong approximability, but is ZK based on the semi-strong approximability. The separation assumes the existence of (*trapdoor*) *one-way permutations* (Definition 1).

Let us recall three definitions of ZK in the ROM (see [Wee09] for a full discussion). These definitions differ in the ability of the simulator to *program* the random oracle at specific points, before the distinguisher can query the random oracle.

Definition 11 (NPRO ZK, EPRO ZK, and FPRO ZK)**.** An interactive protocol $\langle P \leftrightarrow V^*(z)\rangle(x)$, where P is an OM and V^* is a PPT-OM, is ZK for P on L in the ROM, if for every PPT-OM V^*, there exists a PPT-OM S_{V^*}, such that for all polynomial-size family of (oracle) circuits $D = \{D_n\}$, the following holds:

$$\begin{aligned}&(\forall c \in \mathbb{N})(\forall^\infty n \in \mathbb{N})(\forall x \in L_n)(\forall z \in \{0,1\}^*)\\ &\quad\Big|\Pr_{\text{RO}}\left[D_n^{O_1}\left(x,z,\mathsf{view}_{V^*}\left\langle P^{RO} \leftrightarrow V^{*RO}(z)\right\rangle(x)\right) = 1\right] - \\ &\qquad\qquad\Pr_{\text{RO}}\left[D_n^{O_2}\left(x,z,S_{V^*}^{RO}(x,z)\right) = 1\right]\Big| < n^{-c} \ ,\end{aligned} \tag{14}$$

where the first probability is taken over the random coins of S_{V^*}, D_n, and the random selection of RO, and the second probability is taken over the random coins of P, V^*, D_n, and the random selection of RO. The oracles O_1 and O_2 are determined based on the type of ZK in question:

- Non-programmable RO (NPRO) ZK model: $O_1 = O_2 = \text{RO}$.
- Explicitly-programmable RO (EPRO) ZK model: $O_1 = \text{RO}$ and $O_2 = \text{RO}[\ell]$.[8]
- Fully-programmable RO (FPRO) ZK model: $O_1 = O_2 = \varnothing$.

Remark 2. Definition 11 resembles the auxiliary-input ZK (AIZK) in the Standard Model. However, as shown by Wee [Wee09], neither NPRO ZK nor EPRO ZK is closed under the sequential compositions (and the status of FPRO ZK is unknown). This is because in a real interaction, V^* can learn auxiliary information which not only depends on x, but also depends on the RO. As a remedy, Wee suggests using oracle-dependent auxiliary inputs (as defined by Unruh [Unr07]). We will return to this issue in Section 5; until then, we assume nothing about whether our definitions are closed under any type of composition. □

[8] RO[ℓ] behaves much like RO, except that it is programmed according to the list ℓ.

Remark 3. It might be tempting to define other variants of ZK, such as black-box ZK (BBZK) in the ROM. In fact, there exist such definitions in the literature [YZ06, Gag08]. However, it should be noted that in the BBZK, the verifier V^* is chosen after the simulator S is fixed, and therefore V^* can run much longer than S. In particular, let $p(\cdot)$ be a polynomial upper-bounding the running time of S. Then, a cheating verifier V' can start by asking $p(n)+1$ queries from the RO, and then act as the cheating verifier V^*. This will exhaust the simulator, since S has to monitor all queries asked by V'. Another drawback is pointed to the authors by Boaz Barak [Bar10]:

> While a random-oracle model simulator may be efficient, it's obviously not black-box, because it is supposed to somehow look at the execution of V^* and understand from it when V^* is evaluating a hash function. For this reason, it does not make sense to say that a random-oracle model simulator is "black-box."

However, if we neglect this subtle *conceptual* point, there is still one way to *syntactically* define BBZK in the ROM. The point is to define the running time of S as polynomial *not only in $|x|$, but also in the number of queries to RO made by V'*. (This definition was proposed to the authors by David Cash [Cas10]). □

Remark 3 discusses the technicalities one faces when trying to define *semi-strong approximability* in the ROM: Since the approximator should have BB access to the distinguisher, the issue mentioned in Remark 3 arises. Fortunately, there is a *conceptual* work-around (in addition to the syntactical one), if we consider the *semi-strong **K-approximability***. Recall idea behind defining *semi-strong K-approximability*: If D_n "knows" something, the adversary can use a knowledge extractor to extract this knowledge, and then use it to simulate the protocol by itself. Informally, we say that the knowledge of D_n does not *decrease* if it asks more queries from the RO. Therefore, for and D_n, one can construct another circuit D'_n, which:

- If it deems a query made by D_n as dummy, it will answer the query without passing it to the RO.
- Otherwise, it passes the query to the RO and returns the answer.

The statistical independence of $\mathrm{RO}(q_1)$ and $\mathrm{RO}(q_2)$ (whenever $q_1 \neq q_2$) allows D'_n to decide upon the status of a query (dummy or not) independently. Obviously, the number of queries D'_n makes to the RO must be determined before fixing the approximator. Below, we will present a protocol in which D'_n makes no more than a single query. This point is clarified in the proof of Lemma 2.

Having seen many pitfalls along the path, we are now ready to present the definition of RO ZK based on *semi-strong approximability*. Here, we use *semi-strong K-approximability*, because as discussed, it does not suffer from the issues in Remark 3. To simplify the exposition, we only define the **NPRO semi-strong *K*-ZK**. Definitions for EPRO and FPRO follow easily.

Definition 12 (NPRO Semi-Strong K-ZK)**.** An interactive protocol $\langle P \leftrightarrow V^*(z)\rangle(x)$, where P is an OM and V^* is a PPT-OM, is ZK for P on L in the ROM, if for every PPT-OM V^*, there exists a PPT-OM S_{V^*} and an expected PPT-OM K, such that for all polynomial-size family of (oracle) circuits $D = \{D_n\}$, for all $n \in \mathbb{N}$, for all $x \in L_n$, and for all $z \in \{0,1\}^*$, if the advantage

$$\Psi = \mathbf{Adv}_{D_n}^{S,\mathsf{view}_{V^*}}(x,z) \overset{\text{def}}{=} \left| \Pr_{\text{RO}}\left[D_n^{\text{RO}}\left(x,z,\mathsf{view}_{V^*}\left\langle P^{RO} \leftrightarrow V^{*RO}(z)\right\rangle(x)\right)=1\right] - \Pr_{\text{RO}}\left[D_n^{\text{RO}}\left(x,z,S_{V^*}^{\text{RO}}(x,z)\right)=1\right] \right| \quad (15)$$

is nonzero, then the following holds:

$$\left| \Pr_{\text{RO}}\left[D_n^{\text{RO}}\left(x,z,\mathsf{view}_{V^*}\left\langle P^{RO} \leftrightarrow V^{*RO}(z)\right\rangle(x)\right)=1\right] - \Pr_{\text{RO}}\left[D_n^{\text{RO}}\left(x,z,K^{D_n,\text{RO}}(\langle x,1^{1/\Psi}\rangle,z)\right)=1\right] \right| < n^{-c} \ , \quad (16)$$

where the first probability is taken over the random coins of P, V^*, D_n and the random selection of RO, and the second probability is taken over the random coins of K, D_n, and the random selection of RO.

Theorem 3. Assuming the existence of *trapdoor one-way permutations*, there exists an *efficient-prover* protocol in the ROM, which is not ZK even in the EPRO-ZK sense, but is NPRO semi-strong K-ZK.

For the lack of space, we prove a simpler version of Theorem 3: "Assuming the existence of trapdoor one-way permutations, there exists an efficient-prover protocol in the ROM, which is not **NPRO-ZK**, but is NPRO semi-strong K-ZK." This simpler form provides the required separation (between strong and semi-strong approximabilities), and its ideas can be easily extended to prove Theorem 3.

Proof. Let $\mathcal{F} = \{f_n\}$ be a collection of trapdoor one-way permutations, and $t = \{t_n\}$ be the corresponding trapdoor set. Consider Protocol 1:

PROTOCOL 1:

- **Common Input:** Description of f_n.
- **Prover's Auxiliary Input:** t_n.
- **Protocol Description:**
  1. V computes $x \leftarrow \textsc{Samp}(\mathsf{desc}(f_n))$ and $y \leftarrow \textsc{Eval}(\mathsf{desc}(f_n), x)$, and sends y to P.
  2. The *efficient* prover P computes $x \leftarrow \textsc{Inv}(\mathsf{desc}(f_n), t_n, y)$, and $w \leftarrow \mathrm{RO}(x)$, and sends w to V.
- **Verification:** V accepts if $w = \mathrm{RO}(x)$, and rejects otherwise.

Remark 4. Protocol 1 is a proof of computational ability [BG92], and can be used as an *efficient* identification scheme if the RO is instantiated properly (where the prover demonstrates his ability of inverting a one-way permutation to the verifier.) However, as pointed out in Section 5, the ZK property of this protocol is not preserved under sequential composition. For this reason, we suggest using Protocol 3, which is as efficient as Protocol 1. □

We next prove that Protocol 1 is not NPRO ZK, but is NPRO semi-strong K-ZK.

Lemma 1. Assuming $\mathcal{F} = \{f_n\}$ is a collection of trapdoor one-way permutations, Protocol 1 is not NPRO ZK.

The proof appears in Appendix C. The proof can be easily modified to prove that Protocol 1 is not EPRO ZK. That is, even the ability of S_{V^*} to program RO at polynomially many points does not help it to strongly approximate the view of V^*. This is mainly due to the fact that the PPT-OM S_{V^*} cannot compute the right value (i.e. ${f_n}^{-1}(y)$) at which RO should be programmed.

Lemma 2. Protocol 1 is NPRO semi-strong K-ZK (as per Definition 12).

The proof appears in Appendix D. Together, Lemmas 1 and 2 prove Theorem 3. ■

5 Sequential Composition

Recent works on composition, such as [Wee09, BV10], showed that proving composition theorems is a subtle task. In this section, we first prove that a variant of Protocol 1 is not closed under sequential compositions, and therefore rule out the closeness of NPRO semi-strong ZK under such compositions.[9] We then provide a model called NPRO semi-strong ZK with **oracle-dependent auxiliary-input**, and prove that it is closed under sequential compositions. Finally, we present Protocol 3—a modification of Protocol 1—which is ZK in this model but not in the EPRO ZK model.

5.1 Insecurity under Sequential Compositions

Consider Protocol 2, which is a variant of Protocol 1. Note that we made two reasonable assumptions about the underlying collection of trapdoor one-way permutations ($\mathcal{F} = \{f_n\}$): For the given security parameter,

(1) The range of the random oracle coincides with $\mathsf{dom}({f_n}^{-1}) = \mathsf{dom}(f_n)$.

(2) The distribution which $\textsc{Samp}(\mathsf{desc}(f_n))$ induces on $\mathsf{dom}(f_n)$ is computationally indistinguishable from the uniform distribution (since by assumption (1), the random oracle induces a uniform distribution on $\mathsf{dom}(f_n)$).

Interestingly, these assumptions are those required for the validity of full-domain hash [BR93, BR96]. As pointed out in [BR93, Section 4], while standard trapdoor permutations (such as RSA) do not possess these properties, the scheme can be patched nonetheless to provide them as well.

PROTOCOL 2:

[9] In fact, this is totally expected, because NPRO semi-strong ZK is more general than NPRO ZK, and as proved in [Wee09], NPRO ZK is not closed under sequential compositions.

- **Common Input:** Description of f_n.
- **Prover's Auxiliary Input:** t_n.
- **Protocol Description:**
  1. V sends some string α to P.
  2. Using t_n, the efficient prover computes $\beta = \mathrm{RO}\left({f_n}^{-1}(\mathrm{RO}(0^n))\right)$. If $\alpha = \beta$, the prover sends t_n to V. Otherwise, the prover sends β.
- **Verification:** V always accepts (i.e. the soundness holds vacuously).

Theorem 4. Assuming that $\mathcal{F} = \{f_n\}$ is defined as above, Protocol 2 possesses the following properties:

(i) It is EPRO-ZK but not NPRO-ZK.

(ii) It is NPRO semi-strong K-ZK.

(iii) If composed twice (sequentially), it is no longer zero knowledge.

The proof appears in Appendix E.

Remark 5. The reason why Protocol 2 is not ZK under compositions is that that auxiliary input to V^* in the second execution (i.e. β) depends on RO, while the traditional auxiliary input z cannot depend on RO (see Definition 11, where z is selected before RO is determined). We will resolve this issue in the next section.

5.2 Models with Oracle-Dependent Auxiliary-Input

To devise a sequentially composable model of ZK in the ROM, we have to make *compromises*. Specifically, if z is allowed to depend arbitrarily on RO, we will stuck at the proof of the composition theorem, for we cannot use the averaging argument as in the standard model. (The issue is discussed more clearly during the course of the proof of Theorem 5; see also [Wee09, footnote 12]).

The *compromise* is to consider a model where all parties are modeled as PPTMs, and the auxiliary input to V^* is generated by a nonuniform PPT-OM which has access to RO. Let us exemplify this model in the definition of NPRO semi-strong K-ZK with oracle-dependent auxiliary input (cf. Definition 12):

Definition 13 (NPRO Semi-Strong K-ZK with Oracle-Dependent Auxiliary Input)**.** An interactive protocol $\langle P(y) \leftrightarrow V^*(z)\rangle(x)$, where P and V^* are a PPT-OMs, is ZK for P on $L \subseteq \mathsf{NP}$ in the ROM, if for every PPT-OM V^*, there exists a PPT-OM S_{V^*} and an expected PPT-OM K, such that for all polynomial-size family of (oracle) circuits $D = \{D_n\}$ and $Z = \{Z_n\}$, for all $n \in \mathbb{N}$, for all $(x, y) \in R_{L_n}$, and for all $\zeta \in \{0,1\}^*$, if

$$\Psi = \mathbf{Adv}^{S,\mathsf{view}_{V^*}}_{D_n,Z_n}(x,y,\zeta) \overset{\text{def}}{=} \underset{\mathrm{RO}}{\mathbb{E}}\left[z \leftarrow Z_n^{\mathrm{RO}}(\zeta) \quad : \quad \left|\Pr\left[D_n^{\mathrm{RO}}\left(x,z,\mathsf{view}_{V^*}\langle P^{\mathrm{RO}}(y) \leftrightarrow V^{*\mathrm{RO}}(z)\rangle(x)\right)=1\right] - \Pr\left[D_n^{\mathrm{RO}}\left(x,z,S_{V^*}^{\mathrm{RO}}(x,z)\right)=1\right]\right|\right] \tag{17}$$

is nonzero, then the following holds:

$$\left|\Pr\left[D_n^{\mathrm{RO}}\left(x,z,\mathsf{view}_{V^*}\langle P^{\mathrm{RO}}(y) \leftrightarrow V^{*\mathrm{RO}}(z)\rangle(x)\right)=1\right] - \Pr\left[D_n^{\mathrm{RO}}\left(x,z,K^{D_n,\mathrm{RO}}(\langle x, 1^{1/\Psi}\rangle, z)\right)=1\right]\right| < n^{-c} \ . \tag{18}$$

Theorem 5. Definition 13 is closed under sequential compositions.

The proof appears in Appendix F.

5.3 A New Protocol

It is easy to show that Protocol 1 is not ZK under Definition 13. An informal proof follows: Let Z_n compute $x \leftarrow \textsc{Samp}(\mathsf{desc}\, f_n)$ and $y \leftarrow \textsc{Eval}(\mathsf{desc}\, f_n, x)$, and output $z = y \mid\mid \mathrm{RO}(\mathrm{RO}(x))$. A cheating verifier V^* forwards y to the prover (or simulator), instead of computing it as prescribed. On receiving the response from the honest prover (which should be $w = \mathrm{RO}(x)$ by definition), V^* just queries RO at w, and accepts if $\mathrm{RO}(w) = \mathrm{RO}(\mathrm{RO}(x))$. (The right-hand side is extracted from z). On the other hand, to compute x, the simulator must either invert y or invert RO; yet both tasks are infeasible for it. Let w be the output of the simulator. To check the output, the distinguisher D_n simply queries RO at w, and compares the answer to the second component in z (i.e. RO(RO(x)). Note that in this case, D_n does not *know* the value of x, so its queries does not help any extractor K.

To fix this problem, we propose Protocol 3, which exploits Pass' commitments [Pas03a]: To commit to a string x in the ROM, choose a random s, and send $(s, \mathrm{RO}(x \mid\mid s))$. Note that there is no decommitment phase, since the prescribed verifier already knows x.

PROTOCOL 3:

- **Common Input:** Description of f_n.
- **Prover's Auxiliary Input:** t_n.
- **Protocol Description:**
  1. V computes $x \leftarrow \textsc{Samp}(\mathsf{desc}(f_n))$ and $y \leftarrow \textsc{Eval}(\mathsf{desc}(f_n), x)$, and sends y to P.
  2. The *efficient* prover P computes $x \leftarrow \textsc{Inv}(\mathsf{desc}(f_n), t_n, y)$, chooses $s \leftarrow_R \{0,1\}^{|x|}$, computes $w \leftarrow \mathrm{RO}(x \mid\mid s)$, and sends (s, w) to V.
- **Verification:** V accepts if $w = \mathrm{RO}(x \mid\mid s)$, and rejects otherwise.

Theorem 6. Assuming that $\mathcal{F} = \{f_n\}$ is defined as above, Protocol 3 possesses the following properties:

(i) It is ZK under Definition 13.

(ii) It is not EPRO ZK.

The proof appears in Appendix G.

6 Future Work

We believe that the most important task is to remove the need for the RO, and replace it with some suitable assumption. One possible solution is to extract the required properties which the RO satisfies, and try to find a cryptographic primitive which satisfies these constraints (similar to [Can97, CMR98, CD08]). Specifically, we believe that a suitable assumption, similar to the knowledge-of-exponent assumption (KEA) [Dam92, HT98, BP04, CD08] may prove useful. We are currently studying the plausibility of the following assumption (stated intuitively): For any PPTM D which distinguishes with non-negligible advantage between $(f_n(x), g^r, g^{rx}, z)$ and $(f_n(x), g^r, g^s, z)$, where p and $q = (p-1)/2$ are primes, $g \in \mathbb{Z}_p^*$ has order q, and r and s are uniformly selected from $\mathbb{Z}_q$, there exists another PPTM S which outputs x. This assumption can be seen as a *decisional* version of the KEA, and can be used to provide a **weak uniform ZK** protocol based on Protocol 3, in the standard model.

It is also interesting to study the closedness of the new definitions under other types of compositions. Moreover, separating various levels of the two hierarchies from each other is desirable.

References

[Bar01] Boaz Barak. How to Go Beyond the Black-Box Simulation Barrier. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science* (*FOCS '01*), pages 106–115, Las Vegas, Nevada, USA, 2001. IEEE Computer Society.

[Bar10] Boaz Barak. Personal communication, 2010. The transcript is available at `http://cstheory.stackexchange.com/questions/1454/1568#1568`.

[BCC88] Gilles Brassard, David Chaum, and Claude Crépeau. Minimum Disclosure Proofs of Knowledge. *Journal of Computer and System Sciences* (*JCSS*), 37(2):156–189, 1988.

[BG92] Mihir Bellare and Oded Goldreich. Proving Computational Ability. Available from: `http://cseweb.ucsd.edu/~mihir/papers/poa.ps` or `http://www.wisdom.weizmann.ac.il/~oded/PS/poa.ps`. Published recently in [Gol11], 1992.

[BG93] Mihir Bellare and Oded Goldreich. On Defining Proofs of Knowledge. In *Advances in Cryptology—CRYPTO '92*, pages 390–420, London, UK, 1993. Springer-Verlag.

[BGGL01] Boaz Barak, Oded Goldreich, Shafi Goldwasser, and Yehuda Lindell. Resettably-Sound Zero-Knowledge and Its Applications. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science* (*FOCS '01*), pages 116–125. IEEE, 2001.

[BL02] Boaz Barak and Yehuda Lindell. Strict Polynomial-Time in Simulation and Extraction. In *Proceedings of the 34th Annual ACM Symposium on Theory of Csomputing* (*STOC '02*), pages 484–493, New York, NY, USA, 2002.

[BM82] Manuel Blum and Silvio Micali. How to Generate Cryptographically Strong Sequences of Pseudo-Random Bits. In *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science* (*FOCS '82*), pages 112–117, Washington, DC, USA, 1982. IEEE Computer Society. See [BM84] for the journal version.

[BM84] Manuel Blum and Silvio Micali. How to Generate Cryptographically Strong Sequences of Pseudo-Random Bits. *SIAM Journal on Computing*, 13(4):850–864, November 1984. See [BM82] for the conference version.

[BP04] Mihir Bellare and Adriana Palacio. The Knowledge-of-Exponent Assumptions and 3-Round Zero-Knowledge Protocols. In *Advances in Cryptology—CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 227–232. Springer Berlin / Heidelberg, 2004.

[BR93] Mihir Bellare and Phillip Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In *Proceedings of the 1st Annual ACM Conference on Computer and Communications Security*, pages 62–73. ACM, 1993.

[BR96] Mihir Bellare and Phillip Rogaway. The Exact Security of Digital Signatures—How to Sign with RSA and Rabin. In *Advances in Cryptology—EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 399–416. Springer Berlin / Heidelberg, 1996.

[BV10] Eleanor Birrell and Salil Vadhan. Composition of Zero-Knowledge Proofs with Efficient Provers. In *Theory of Cryptography—TCC '10*, volume 5978 of *Lecture Notes in Computer Science*, pages 572–587. Springer Berlin / Heidelberg, 2010. Full version is available at `http://eprint.iacr.org/2009/604`.

[Can97] Ran Canetti. Towards Realizing Random Oracles: Hash Functions That Hide All Partial Information. In *Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology—Crypto '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 455–469, Santa Barbara, California, USA, 1997. Springer-Verlag. See [Can00] for the full version.

[Can00] Ran Canetti. Towards Realizing Random Oracles: Hash Functions that Hide All Partial Information, 2000. Unpublished Manuscript. Available from `http://www.research.ibm.com/security/pof-long.ps`. See [Can97] for the conference version.

[Can01] Ran Canetti. Universally Composable Security: A New Paradigm for Cryptographic Protocols (Extended Abstract). In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science* (*FOCS '01*), page 136, Washington, DC, USA, 2001. IEEE Computer Society. See [Can05] for the full version.

[Can05] Ran Canetti. Universally Composable Security: A New Paradigm for Cryptographic Protocols. Cryptology ePrint Archive, Report 2000/067, 2005. Available from `http://eprint.iacr.org/2000/067`. See [Can01] for the conference version.

[Cas10] David Cash. Personal communication, 2010. The transcript is available at `http://cstheory.stackexchange.com/questions/1454/1509#1509`.

[CD08] Ran Canetti and Ronny Ramzi Dakdouk. Extractable Perfectly One-Way Functions. In *International Colloquium on Automata, Languages and Programming—ICALP '08*, volume 5126 of *Lecture Notes in Computer Science*, pages 449–460. Springer Berlin / Heidelberg, 2008. See [Dak09] for the full version.

[CF01] Ran Canetti and Marc Fischlin. Universally Composable Commitments. In *Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology—CRYPTO '01*, volume 2045 of *Lecture Notes in Computer Science*, pages 19–40, Santa Barbara, California, USA, 2001. Springer-Verlag.

[CMR98] Ran Canetti, Daniele Micciancio, and Omer Reingold. Perfectly One-Way Probabilistic Hash Functions (Preliminary Version). In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing* (*STOC '98*), pages 131–140, New York, NY, USA, 1998. ACM.

[Dak09] Ronny Ramzi Dakdouk. *Theory and Application of Extractable Functions*. PhD thesis, Yale University, New Haven, Connecticut, USA, 2009. Available from `http://www.cs.yale.edu/~jf/Ronny-thesis.pdf`.

[Dam92] Ivan Damgård. Towards Practical Public Key Systems Secure Against Chosen Ciphertext Attacks. In *Advances in Cryptology—CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*, pages 445–456. Springer Berlin / Heidelberg, 1992.

[DNRS99] Cynthia Dwork, Moni Naor, Omer Reingold, and Larry J. Stockmeyer. Magic Functions. In *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science* (*FOCS '99*), pages 523–534, New York, NY, USA, 1999. IEEE Computer Society. See [DNRS03] for the full version.

[DNRS03] Cynthia Dwork, Moni Naor, Omer Reingold, and Larry J. Stockmeyer. Magic Functions. *Journal of the ACM* (*JACM*), 50(6):852–921, November 2003. See [DNRS99] for the conference version.

[DNS98] Cynthia Dwork, Moni Naor, and Amit Sahai. Concurrent zero-knowledge. In *Proceedings of the 30th Annual ACM Symposium on Theory of Csomputing* (*STOC '98*), pages 409–418, New York, NY, USA, 1998. See [DNS04] for the conference version.

[DNS04] Cynthia Dwork, Moni Naor, and Amit Sahai. Concurrent Zero-Knowledge. *Journal of the ACM* (*JACM*), 51(6):851–898, November 2004. See [DNS98] for the conference version.

[Dou11] Mohammad Sadeq Dousti. Beating Nonuniformity by Oracle Access, 2011. The transcript is available at `http://cstheory.stackexchange.com/q/4796/873`.

[DS02] Cynthia Dwork and Larry Stockmeyer. 2-Round Zero Knowledge and Proof Auditors. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing* (*STOC '02*), pages 322–331, Montréal, Quebec, Canada, 2002. ACM.

[FFS88] Uriel Feige, Amos Fiat, and Adi Shamir. Zero-Knowledge Proofs of Identity. *Journal of Cryptology*, 1(2):77–94, 1988.

[FS90] Uriel Feige and Adi Shamir. Witness Indistinguishable and Witness Hiding Protocols. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing* (*STOC '90*), pages 416–426, New York, NY, USA, 1990. ACM.

[Gag08] Martin Gagné. *A Study of the Random Oracle Model*. PhD thesis, University of California at Davis, CA, USA, 2008. Available from `http://wwwlib.umi.com/dissertations/fullcit/3336254`.

[GK90] Oded Goldreich and Hugo Krawczyk. On the Composition of Zero-Knowledge Proof Systems. In Mike Paterson, editor, *Proceedings of the 17th International Colloquium on Automata, Languages and Programming* (*ICALP '90*), volume 443 of *Lecture Notes in Computer Science*, pages 268–282, Warwick University, England, 1990. Springer. See [GK96] for the journal version.

[GK96] Oded Goldreich and Hugo Krawczyk. On the Composition of Zero-Knowledge Proof Systems. *SIAM Journal on Computing*, 25(1):169–192, 1996. See [GK90] for the conference version.

[GM82] Shafi Goldwasser and Silvio Micali. Probabilistic Encryption & How to Play Mental Poker Keeping Secret All Partial Information. In *Proceedings of the 14th Annual ACM Symposium on Theory of Csomputing* (*STOC '82*), pages 365–377, New York, NY, USA, 1982. See [GM84] for the journal version.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic Encryption. *Journal of Computer and System Sciences* (*JCSS*), 28(2):270–299, 1984. See [GM82] for the conference version.

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The Knowledge Complexity of Interactive Proof Systems. In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing*, pages 291–304, 1985.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The Knowledge Complexity of Interactive Proof Systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[GO94] Oded Goldreich and Yair Oren. Definitions and Properties of Zero-Knowledge Proof Systems. *Journal of Cryptology*, 7:1–32, 1994. See [Ore87] for the conference version.

[Gol93] Oded Goldreich. A Uniform-Complexity Treatment of Encryption and Zero-Knowledge. *Journal of Cryptology*, 6(1):21–53, 1993.

[Gol01] Oded Goldreich. *Foundations of Cryptography: Volume 1, Basic Tools*. Cambridge University Press, 2001.

[Gol07] Oded Goldreich. On Expected Probabilistic Polynomial-Time Adversaries: A Suggestion for Restricted Definitions and Their Benefits. In *Theory of Cryptography—TCC '07*, volume 4392 of *Lecture Notes in Computer Science*, pages 174–193. Springer Berlin / Heidelberg, 2007. See [Gol10] for the journal version.

[Gol08] Oded Goldreich. *Computational Complexity: A Conceptual Perspective*. Cambridge University Press, 1st edition, 2008.

[Gol10] Oded Goldreich. On Expected Probabilistic Polynomial-Time Adversaries: A Suggestion for Restricted Definitions and Their Benefits. *Journal of Cryptology*, 23(1):1–36, 2010. See [Gol07] for the conference version.

[Gol11] Oded Goldreich. *Studies in Complexity and Cryptography: Miscellanea on the Interplay between Randomness and Computation*, volume 6650 of *LNCS*, chapter Proving Computational Ability, pages 6–12. Springer, 2011.

[HT98] Satoshi Hada and Toshiaki Tanaka. On the Existence of 3-Round Zero-Knowledge Protocols. In *Advances in Cryptology—CRYPTO '98*, volume 1462 of *Lecture Notes in Computer Science*, pages 197–202. Springer Berlin / Heidelberg, 1998. See `http://eprint.iacr.org/1999/009` for the full and corrected version.

[Ore87] Yair Oren. On the Cunning Power of Cheating Verifiers: Some Observations about Zero Knowledge Proofs (Extended Abstract). In Ashok K. Chandra, editor, *Proceedings of the 28th Annual IEEE Symposium on Foundations of Computer Science* (*FOCS '87*), pages 462–471, Los Angeles, California, USA, 1987. IEEE Computer Society Press. See [GO94] for the journal version.

[Pas03a] Rafael Pass. On Deniability in the Common Reference String and Random Oracle Model. In *Advances in Cryptology—CRYPTO 2003*, pages 316–337. Springer-Verlag, 2003.

[Pas03b] Rafael Pass. Simulation in Quasi-Polynomial Time, and Its Application to Protocol Composition. In *Advances in Cryptology—EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 642–643. Springer Berlin / Heidelberg, 2003.

[Unr07] Dominique Unruh. Random Oracles and Auxiliary Input. In *Advances in Cryptology—CRYPTO 2007*, pages 205–223. Springer-Verlag, 2007.

[Wee09] Hoeteck Wee. Zero Knowledge in the Random Oracle Model, Revisited. In *Advances in Cryptology—ASIACRYPT 2009*, pages 417–434. Springer-Verlag, 2009.

[Yao82] Andrew Chi-Chih Yao. Theory and Applications of Trapdoor Functions (extended abstract). In *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science* (*FOCS '82*), pages 80–91. IEEE, 1982.

[YZ06] Moti Yung and Yunlei Zhao. Interactive Zero-Knowledge with Restricted Random Oracles. In *Theory of Cryptography—TCC '06*, volume 3876 of *Lecture Notes in Computer Science*, pages 21–40. Springer Berlin / Heidelberg, 2006.

Appendices

A Some Omitted Definitions

Definition 14 (Strong K-Approximability)**.** A poly-bounded distribution ensemble $U = \{U(x,z)\}_{x\in L, z\in\{0,1\}^*}$ is said to be *strongly K-approximable* on the language L, if there exists a PPTM M and an expected PPTM K, such that for every family of polynomial-size circuits $D = \{D_n\}$, for all $n \in \mathbb{N}$, for all $x \in L_n$, and for all $z \in \{0,1\}^*$, if $\Psi = \mathbf{Adv}_{D_n}^{M,U}(x,z)$ (defined in (10)) is nonzero, then the following holds:

$$|\Pr[D_n(x,z,K(\langle x, 1^{1/\Psi}\rangle, z)) = 1] - \Pr[D_n(x,z,U(x,z)) = 1]| < n^{-c} \ , \tag{19}$$

where the first probability is taken over the random coins of K and D_n, and the second probability is taken over the (implicit) random coins of U and D_n.

Definition 15 (Semi-Weak K-Approximability)**.** A poly-bounded distribution ensemble $U = \{U(x,z)\}_{x\in L, z\in\{0,1\}^*}$ is said to be *semi-weakly K-approximable* on the language L, if for every family of polynomial-size circuits $D = \{D_n\}$, for all $n \in \mathbb{N}$, there exists a PPTM M and an expected PPTM K, such that for all $x \in L_n$, and for all $z \in \{0,1\}^*$, if $\Psi = \mathbf{Adv}_{D_n}^{M,U}(x,z)$ (defined in (10)) is nonzero, the following holds:

$$|\Pr[D_n(x,z,K(\langle x, 1^{1/\Psi}\rangle, z)) = 1] - \Pr[D_n(x,z,U(x,z)) = 1]| < n^{-c} \ , \tag{20}$$

where the first probability is taken over the random coins of K and D_n, and the second probability is taken over the (implicit) random coins of U and D_n.

Definition 16 (Weak K-Approximability)**.** A poly-bounded distribution ensemble $U = \{U(x,z)\}_{x\in L, z\in\{0,1\}^*}$ is said to be *weakly K-approximable* on the language L, if for every family of polynomial-size circuits $D = \{D_n\}$, for all $n \in \mathbb{N}$, for every $x \in L_n$, and for all $z \in \{0,1\}^*$, there exists a PPTM M and an expected PPTM K, such that if $\Psi = \mathbf{Adv}_{D_n}^{M,U}(x,z)$ (defined in (10)) is nonzero, the following holds:

$$|\Pr[D_n(x,z,K(\langle x, 1^{1/\Psi}\rangle, z)) = 1] - \Pr[D_n(x,z,U(x,z)) = 1]| < n^{-c} \ , \tag{21}$$

where the first probability is taken over the random coins of K and D_n, and the second probability is taken over the (implicit) random coins of U and D_n.

B Proof of Theorem 2

In the definitions of K-approximability, the knowledge extractor has more freedom over the approximator, since its running time may depend on the distinguishing advantage. Therefore, if some distribution ensemble is approximable, it is *a fortiori* K-approximable.

The entailments in (13) can be proven similar to the proof of Theorem 1:

Strong K-approximability implies semi-strong K-approximability since, in the latter, the knowledge extractor (K) can have black-box access to the distinguisher. If the approximation is possible without such access (as is the case with strong approximability), it is *a fortiori* possible with black-box access.

The same reasoning holds while comparing semi-strong and mild approximabilities: If the approximation is possible with only black-box access (as is the case with semi-strong approximability), it is *a fortiori* possible with code access (as is the case with mild approximability).

Mild approximability implies semi-weak approximability since the order of quantifiers in the definition of the latter allows the knowledge extractor to depend *arbitrarily* on the distinguisher.

Semi-weak approximability implies weak approximability because the latter allows the knowledge extractor to depend not only on the distinguisher, but also on the common and auxiliary inputs.

C Proof of Lemma 1

Assume, towards contradiction, that there exists a PPT-OM simulator S_{V^*} for Protocol 1, which satisfies the NPRO-ZK requirement of Definition 11. Assume that the running time of S_{V^*} is bounded by a polynomial $m(\cdot)$.

With no loss of generality, we assume that $m(\cdot)$ dominates the running time of V^* (this is due to the order of quantifiers in Definition 11, which allows S_{V^*} to depend on V^*).

For the common input $\mathsf{desc}(f_n)$, define the auxiliary input of V^* as $z = \langle y \mid\mid 1^{m(n)} \mid\mid t_n \rangle$, where $\mid\mid$ denotes concatenation, and $y \leftarrow \textsc{Eval}(\mathsf{desc}(f_n), \textsc{Samp}(\mathsf{desc}(f_n)))$. The cheating V^* reads the prefix y of z, and forwards it to S_{V^*} (instead of computing it via $\textsc{Samp}$ and $\textsc{Eval}$). This way, we are assured (with overwhelming probability) that V^* does not *know* $x = f_n^{-1}(y)$.[10] When V^* receives the answer, it **halts** the protocol and tries to process its view to increase its knowledge. In other words, the cheating verifier **does not make any queries** to the RO, and does not produce any outputs.

Claim 1. Assuming $\mathcal{F} = \{f_n\}$ is a collection of trapdoor one-way permutations, the probability that S_{V^*} queries RO at $x = f_n^{-1}(y)$ is negligible.

Proof. Obviously, S_{V^*} cannot read the suffix t_n of z, since its running time is limited to $m(n)$. Assume towards contradiction that the probability that S_{V^*} queries RO at $x = f_n^{-1}(y)$ is not negligible. We present a PPTM $\mathcal{A}$ which uses S_{V^*} as a subroutine to invert $\mathcal{F}$ on infinitely many n's with non-negligible probability.

By hypothesis, the probability that S_{V^*} queries RO at $x = f_n^{-1}(y)$ is not negligible. In other words, that there exist infinitely many n's, for which on common input $\mathsf{desc}(f_n)$ and auxiliary input z defined above, the simulator queries RO at $x = f_n^{-1}(y)$ with non-negligible.

$\mathcal{A}$ does the following: On input y and $\mathsf{desc}(f_n)$, it simply runs S_{V^*} on common input $\mathsf{desc}(f_n)$ and auxiliary input $z = \langle y \mid\mid 1^{m(n)} \mid\mid t_n \rangle$, monitors its queries to RO. Every time S_{V^*} queries RO at some point $\hat{x}$, $\mathcal{A}$ checks the condition $y = f_n(x)$, and halts and outputs $x = \hat{x}$ if the condition holds. Otherwise, $\mathcal{A}$ answers the query *consistently at random*[11]

It is evident that the running time of $\mathcal{A}$ is polynomial, and the probability that $\mathcal{A}$ outputs $x = f_n^{-1}(y)$ equals the probability that S_{V^*} queries RO at $x = f_n^{-1}(y)$, which is not negligible by hypothesis. Therefore, if y is chosen according to Definition 1, i.e.

$$x \leftarrow \textsc{Samp}\left(\mathsf{desc}\left(f_n\right)\right), \quad y \leftarrow \textsc{Eval}\left(\mathsf{desc}\left(f_n\right), x\right) \quad ,$$

$\mathcal{A}$ manages to invert f_n with probability that is not negligible, contradicting the assumption that $\mathcal{F} = \{f_n\}$ is a collection of trapdoor one-way permutations. ■

Claim 2. If S_{V^*} does not query RO at $x = f_n^{-1}(y)$, its output is *distinguishable* from the view of V^* with *overwhelming* probability.

Proof. Assume that S_{V^*} makes several queries to RO, none of which occurs at $x = f_n^{-1}(y)$. It then outputs the (simulated) view of V^*, including the transcript (y, w).

There exists a family of polynomial-size circuits whose members are sufficiently large to read the suffix t_n out of z. Let $D = \{D_n\}$ be one such family, in which the circuit D_n just computes $x \leftarrow \textsc{Inv}(\mathsf{desc}(f_n), t_n, y)$, and compares $\mathrm{RO}(x)$ with w. It outputs 1 if and only if the equality holds.

If S_{V^*} has not queried RO at $x = f_n^{-1}(y)$, the probability that D_n outputs 1 on the simulated view is $\Pr[w = \mathrm{RO}(x)] = 2^{-n}$. This is an information-theoretic result, and does not rely on any complexity-theoretic assumption.

On the other hand, the probability that D_n outputs 1 on the real view is 1. Therefore, in this case, the simulated and real views are distinguishable with probability $1 - 2^{-n}$, which is overwhelming. ■

Define E_1 and E_2 as the following events:

- E_1: The event that S_{V^*} queries RO at $x = f_n^{-1}(y)$.
- E_2: The event that the real view is *distinguishable* from the simulated view.

[10] This is just a conceptual observation, and we do not need it for the rest of the proof. It is proven by showing that V^* cannot compute x, as is shown next for S_{V^*}.

[11] The term *consistently at random* requires elaboration. It means that $\mathcal{A}$ keeps a table of all previous queries and answers. If a query has already been asked, the table is looked-up, and the same answer is returned (consistency). Otherwise, a random answer is picked and returned, and the table is updated.

By Claims 1, we have $\Pr[E_1] \leq \epsilon_1(n)$, for some negligible function $\epsilon_1(\cdot)$. By Claims 2, we have $\Pr[E_2 \mid \neg E_1] \geq \epsilon_2(n)$, for some negligible function $\epsilon_2(\cdot)$. Moreover, we can assume that S_{V^*} is intelligible enough to output the right distribution if it somehow manages to query RO at $x = f_n^{-1}(y)$. Hence $\Pr[E_2 \mid E_1] = 0$.

Now we can prove the following:

$$\begin{aligned}\Pr[E_2] &= \Pr[E_2 \mid E_1] \cdot \Pr[E_1] + \Pr[E_2 \mid \neg E_1] \cdot \Pr[\neg E_1] \\ &\geq 0 \cdot \epsilon_1(n) + (1 - \epsilon_2(n)) \cdot (1 - \epsilon_1(n)) \\ &\geq 1 - \epsilon_1(n) - \epsilon_2(n) + \epsilon_1(n)\epsilon_2(n) \ \ ,\end{aligned}$$

which is an overwhelming quantity. This shows that no simulator can output the right distribution, and Lemma 1 follows.

D Proof of Lemma 2

The verification stage of Protocol 1 requires only a single query. Therefore, we may assume that $D = \{D_n\}$ is a family of single-query circuits. (See Remark 3 for more information.) Otherwise, we construct a family of single-query circuits $D' = \{D'_n\}$ from D, which performs as D, but passes the query to the RO only if the query y satisfies $y = f_n(x)$, and this is the first time the query y is asked. Otherwise, D' answers the query *consistently at random* (see Footnote 11). Due to the independence of $\mathrm{RO}(\alpha)$ and $\mathrm{RO}(\alpha')$ for any $\alpha \neq \alpha'$, the output distribution of D' is identical to that of D, and therefore single-query circuits perform as well as multi-query circuits in this experiment, and there is no loss of generality in assuming that the distinguishers are single-query circuits.

Consider a simulator S_{V^*} which receives the input $(\mathsf{desc}(f_n), z)$. Let y be computed in the same way as V^* computes y. The simulator simply computes a consistently random value w, and outputs $(\mathsf{desc}(f_n), y, w, r, z)$. Here, r is the random tape of V^*.

Now consider any family of single-query circuits $D' = \{D'_n\}$, and perform the following experiment:

```
1. Let $b \leftarrow_R \{0,1\}$.
2. IF $b = 0$ THEN
3.         Let $\tau \leftarrow S_{V^*}^{\mathrm{RO}}(\mathsf{desc}(f_n), z)$.
4. ELSE
5.         Let $\tau \leftarrow \mathsf{view}_{V^{*\mathrm{RO}}}(\mathsf{desc}(f_n), z)$.
6. Let $b' \leftarrow {D'_n}^{\mathrm{RO}}(\tau)$.
7. IF $b = b'$ THEN output 1; ELSE output 0.
```

Let E_1 be the event that D'_n queries RO at $x = {f_n}^{-1}(y)$, and E_2 be the event that the output of the experiment is 1. Assume that $\Pr[E_2] \geq \frac{1}{2}$; otherwise, negate the verdict of D'_n, and this inequality holds. Note that if D'_n does not query RO at x, the probability that it announces the correct verdict is $\frac{1}{2}$; in other words, $\Pr[E_2 \mid \neg E_1] = \frac{1}{2}$. This is because D'_n cannot distinguish a consistently random w from $\mathrm{RO}(x)$ without first querying RO at x.

Now, by the "law of total probability":

$$\begin{aligned}\Pr[E_2] &= \Pr[E_1] \cdot \Pr[E_2 \mid E_1] + \Pr[\neg E_1] \cdot \Pr[E_2 \mid \neg E_1] \\ &\leq \Pr[E_1] + \Pr[E_2 \mid \neg E_1] = \Pr[E_1] + \frac{1}{2} \ \ . \end{aligned} \tag{22}$$

We deduce that $\Pr[E_1] \geq \Pr[E_2] - \frac{1}{2}$. Next, it is shown that $\Pr[E_2] - \frac{1}{2} = \Psi/2$, where Ψ is the *advantage* of ${D'_n}^{\mathrm{RO}}$ in distinguishing the real and simulated views (see (15)). For $i, j \in \{0,1\}$, define

$$P_{ij} \stackrel{\text{def}}{=} \Pr\left[{D'_n}^{\mathrm{RO}}(\tau) = i \,\middle|\, b = j\right] \ \ . \tag{23}$$

We therefore have $P_{1j} = 1 - P_{0j}$, and:

$$\begin{aligned}
\Pr[E_2] - \frac{1}{2} &= \left(\frac{1}{2} \cdot P_{00} + \frac{1}{2} \cdot P_{11}\right) - \frac{1}{2} \\
&= \frac{1}{2} \cdot (-P_{10} + P_{11}) \\
&= \frac{1}{2} \cdot | - P_{10} + P_{11}| \\
&= \Psi/2 \ . \qquad (24)
\end{aligned}$$

The third equality follows from the fact that we assumed the left-hand side is positive. Combining (22) and (24), we infer that $\Pr[E_1] \geq \Psi/2$. That is, ${D'_n}^{\mathrm{RO}}$ queries RO at x with probability at least half of its distinguishing advantage.

We are now ready to present the algorithm of the knowledge extractor K required by the Definition 12. Note that K has black-box access to both D'_n and RO, and it is run on input $\big(\langle \mathsf{desc}(f_n), 1^{1/\Psi}\rangle, z\big)$.

```
1. REPEAT $\frac{2n}{\Psi}$ times:
2.        Let $\tau \leftarrow S_{V^*}^{\mathrm{RO}}(\mathsf{desc}(f_n), z)$.
3.        Let $q$ be the (single) query $D'_n(\tau)$ makes to RO (if any).
4.        IF $y = f_n(q)$ output $(\mathsf{desc}(f_n), y, \mathrm{RO}(q), r, z)$ and HALT.
5. Find $x = {f_n}^{-1}(y)$ by exhaustive search.
6. Output $(\mathsf{desc}(f_n), y, \mathrm{RO}(x), r, z)$.
```

The probability of HALT at each iteration is $\Pr[E_1] \geq \Psi/2$. Therefore, the probability of running exhaustive search is less than $(1 - \Psi/2)^{2n/\Psi} < e^{-n}$. The cost of exhaustive search is 2^n. Therefore, the contribution of Step 5 to the expected running time of K is bounded by $e^{-n} \cdot 2^n$, which is negligible in n.

We showed that K runs in expected polynomial time, and can successfully simulate the protocol by finding x.

E Proof of Theorem 4

(i) Protocol 2 is EPRO-ZK because the simulator can compute $x \leftarrow \textsc{Samp}(\mathsf{desc}(f_n))$ and $y \leftarrow \textsc{Eval}(\mathsf{desc}(f_n), x)$. It then programs RO so that $\mathrm{RO}(0^n) = y$. This way, $\beta = \mathrm{RO}(f_n^{-1}(\mathrm{RO}(0^n)))$ equals $\mathrm{RO}(f_n^{-1}(y)) = \mathrm{RO}(x)$.

In the unlikely event that $\alpha = \beta$, the simulator just starts over; since as opposed to the real prover, it cannot output t_n. If even after n retries the simulator fails, it outputs the special failure symbol $\perp$. This happens if after sampling n points $x_1, \ldots, x_n$ (not necessarily distinct) from the domain of f_n, we have $\mathrm{RO}(x_1) = \cdots = \mathrm{RO}(x_n)$. This happens with probability 2^{-n^2}.

On the other hand, if the simulator finds some x_i for which $\beta_i = \mathrm{RO}(x_i) \neq \alpha$, it outputs $(\mathsf{desc}(f_n), \alpha, \beta_i, r, z)$, where α is chosen by V^* and r and z are the V^*'s random tape and auxiliary input, respectively. Note that since S_{V^*} has programmed RO, the output is *perfectly* indistinguishable from the real view, unless the output is $\perp$. The probability of outputting $\perp$ is negligible and independent of the computing power of V^*. We conclude that the output of the simulator is *statistically* indistinguishable from the view of V^*, and therefore the protocol is EPRO-ZK.

Quite contrary, Protocol 2 is not NPRO-ZK. The proof is similar to the proof of Lemma 1. Let $m(\cdot)$ be a polynomial which upper-bounds the running time of S_{V^*}, and assume $z = \langle 0^{m(n)} \mid\mid t_n \rangle$. The simulator cannot read t_n, while there exists a family of poly-size circuits $D = \{D_n\}$ sufficiently large to read z in its entirety. Therefore, in order for S_{V^*} to approximate the real view, it must be able to produce either t_n or β (whichever applies). A reducibility argument can show that in both cases, S_{V^*} can be used (as a black-box) to invert f_n, contradicting its one-wayness.

(ii) The proof is similar to that of Lemma 2. Specifically, instead of β, the simulator outputs some value β^* chosen consistently at random. Let us confine ourselves to single-query distinguishers $D' = \{D'_n\}$, as in the proof of Lemma 2. Let E_1 be the event that D'_n queries RO at $x = {f_n}^{-1}(\mathrm{RO}(0^n))$.

In a completely similar way to the proof of Lemma 2, one can demonstrate that $\Pr[E_1] \geq \Psi/2$, where Ψ is the distinguishing advantage of D'_n. Consequently, a knowledge extractor can compute β in expected time $\mathrm{poly}(n, \Psi^{-1})$, and generate a valid simulation thereafter.

(iii) A cheating verifier V^* can send some junk as α^* in the first step, and get $\beta = \mathrm{RO}\left({f_n}^{-1}(\mathrm{RO}(0^n))\right)$ from the honest prover. In the next execution of the protocol, the verifier sets $\alpha \leftarrow \beta$, sends α to the honest prover, and receives t_n. Since V^* could not compute t_n, we conclude that the sequential composition of Protocol 2 is not zero knowledge.

Remark 6. It is instrumental to construct a protocol which satisfies the conditions of Theorem 4, except that it is not EPRO-ZK. To this end, we must replace $\mathrm{RO}(0^n)$ in Protocol 2 with some value which S_{V^*} cannot program. One possible solution is to let the verifier choose a random r from $\mathsf{dom}({f_n}^{-1})$ (possibly using algorithms SAMP and EVAL), compute α, and send (α, r) to the prover. The prover then uses the value $\hat{\beta} = \mathrm{RO}\left({f_n}^{-1}\left(\mathrm{RO}\left({f_n}^{-1}(r)\right)\right)\right)$ instead of the β used in Protocol 2. The reason for using a random r instead of 0^n is to prevent S_{V^*} from guessing the point at which RO should be programmed. The reason of incorporating two layers of ${f_n}^{-1}$ and two layers of RO is to prevent a cheating V^* from choosing r in a special way so that she can compute β.

It can be proven that the new protocol satisfies all of the conditions of Theorem 4, except that it is not EPRO-ZK. The proof is omitted. □

F Proof of Theorem 5

For simplicity, we only prove the case of sequential *repetition*, where a *single* protocol $\langle P(y) \leftrightarrow V^*(z)\rangle(x)$ is repeated $Q \stackrel{\text{def}}{=} Q(|x|)$ times (Q is a polynomial): In each run, $(x, y) \in R_{L_n}$ is fixed, P uses independent random coins, and the auxiliary input to the cheating verifier includes the history of all previous runs.

Define $Q+1$ hybrids $H_0, H_1, \ldots, H_Q$: The ith hybrid is defined as the output of the following Gedanken- (thought-) experiment:

- Let $z \leftarrow Z_n^{\mathrm{RO}}(\zeta)$ and $h_0 \leftarrow z$.
- Allow the cheating verifier and the honest prover interact i times; for $j \in \{1, 2, \ldots, i\}$ define $h_j \leftarrow \mathsf{view}_{V^*}\langle P^{\mathrm{RO}}(y) \leftrightarrow V^{*\mathrm{RO}}(h_{j-1})\rangle(x)$.
- Run the simulator $Q - i$ times, and let $h_j \leftarrow S_{V^*}^{\mathrm{RO}}(x, h_{j-1})$ for $j \in \{i+1, i+2, \ldots, Q\}$.
- Output (x, z, h_Q).

Note that the extreme hybrids H_0 and H_Q denote the simulated and the real views, respectively. Now assume, contrary to the theorem, that D_n can distinguish the extreme hybrids with non-negligible advantage Ψ. Then, by a hybrid argument, there exists some $i \in \{0, 1, \ldots, Q-1\}$ such that D_n distinguishes H_i from H_{i+1} with advantage at least Ψ/Q, which is non-negligible. Let $Z_{n,i}$ be a circuit which computes a prefix of the above experiment up to the ith execution; i.e. $Z_{n,i}$ is defined as below:

- Let $z \leftarrow Z_n^{\mathrm{RO}}(\zeta)$ and $h_0 \leftarrow z$.
- Allow the cheating verifier and the honest prover interact i times; for $j \in \{1, 2, \ldots, i\}$ define $h_j \leftarrow \mathsf{view}_{V^*}\langle P^{\mathrm{RO}}(y) \leftrightarrow V^{*\mathrm{RO}}(h_{j-1})\rangle(x)$.
- Output (x, z, h_i).

Note that $Z_{n,i}$ can be realized by a poly-size circuit, since Z_n is a poly-size circuit, and the order of quantifiers in Definition 13 allows ζ to include y as well as the code of P (since the prover is assumed to be polynomial). Looking ahead, this is the reason we made the *compromise* discussed at the beginning of Section 5.2: In the standard model, a simple averaging argument can be used to fix the auxiliary input; however, the auxiliary input of our model depends on the RO and cannot be fixed before RO is chosen. Therefore, some variation of Z_n must be incorporated into the code of the distinguisher (see below).

Using $Z_{n,i}$, rewrite the H_i and H_{i+1} as follows:

- $(x, z, h_i) \leftarrow Z^{\mathrm{RO}}_{n,i}(\zeta)$.
- $h_{i+1} \leftarrow S^{\mathrm{RO}}_{V^*}(x, h_i)$.
- For $j \in \{i+2, \ldots, Q\}$, let $h_k \leftarrow S^{\mathrm{RO}}_{V^*}(x, h_{k-1})$.
- Output (x, z, h_Q).

- $(x, z, h_i) \leftarrow Z^{\mathrm{RO}}_{n,i}(\zeta)$.
- $h'_{i+1} \leftarrow \mathsf{view}_{V^*}\langle P^{\mathrm{RO}}(y) \leftrightarrow V^{*\mathrm{RO}}(h_i)\rangle(x)$.
- For $j \in \{i+2, \ldots, Q\}$, let $h'_k \leftarrow S^{\mathrm{RO}}_{V^*}(x, h'_{k-1})$.
- Output (x, z, h'_Q).

Since we assumed that D_n can distinguish the hybrids H_i and H_{i+1} with non-negligible advantage Ψ/Q, there exists an advice ζ, a poly-size circuit $Z_{n,i}$ and a poly-size distinguisher D'_n which—using the oracle-dependent auxiliary input generated by $Z_{n,i}$ on ζ (i.e. h_i)— distinguishes between h_{i+1} and h'_{i+1} with the same advantage: Just simulate S_{V^*} for $Q - i - 1$ rounds (as above) to obtain either h_Q or h'_Q, and then output as D_n does.

Now we exploit the K whose existence is guaranteed by Definition 13: $K^{D'_n,\mathrm{RO}}$ runs in (expected) time poly$(Q(n), \Psi^{-1})$, and generates an output so that D'_n can merely distinguish between hybrids H_i and H_{i+1} with negligible probability. Along the same line of reasoning, if a poly-size circuit D''_n distinguishes between any two adjacent hybrids, $K^{D''_n,\mathrm{RO}}$ can fill the distinguishing gap. Therefore, all hybrids H'_i and H'_{i+1} (generated by K instead of S_{V^*}) are computationally indistinguishable. We conclude that the extreme hybrids H'_0 and H'_n are computationally indistinguishable, and the theorem follows.

G Proof of Theorem 6

(i) Let $Z = \{Z_n\}$ be as in Definition 13. On common and auxiliary inputs $(\mathsf{desc}(f_n), \zeta)$, let Z^{RO}_n output the string z. This string might include, among other things, a list $\ell = \{(q_i, a_i)\}$ of queries q_i to the RO, along with the corresponding answer $a_i = \mathrm{RO}(q_i)$.[12] We call a query *fresh* if it does not belong to ℓ. Note that z (and in particular, ℓ) might be encoded in such a way that it can be understood only by Z, V^*, and D, but it is incomprehensible by S_{V^*} or K.

On input $(\mathsf{desc}(f_n), z)$, the simulator first obtains y from V^*. It then computes $x' \leftarrow \textsc{Samp}(\mathsf{desc}(f_n))$, and $s' \leftarrow_R \{0,1\}^{|x'|}$. It then computes $w' \leftarrow \mathrm{RO}(x' \mid\mid s')$, and outputs $(\mathsf{desc}(f_n), y, s', w', r, z)$. (Here, r denotes the random tape of V^*.)

If the list ℓ contains T queries (which is a polynomial in n since Z_n is a poly-size circuit), the probability that $x'||s'$ is a fresh query is $1 - \frac{T}{2^{2|x|}}$, which is an overwhelming quantity assuming that $|x|$ is super-logarithmic in n. This is indeed the case, because otherwise it would be easy to invert f_n for all n.

Now, if the query $x' \mid\mid s'$ is actually fresh, the oracle-dependent auxiliary input does not help D_n to distinguish the real and simulated view without first making query to RO. In this case, we can prove—similar to the proof of Lemma 2—that if D_n distinguishes the two distributions, there exists a knowledge extractor K which can output x by monitoring the queries of D_n.

However, if the query $x' \mid\mid s'$ is not fresh, D_n can distinguish the two distributions without making any queries to RO. In this case, K may resort to exhaustive search, which is justifiable because the probability of S_{V^*} query not being is negligible. Alternatively, K might test any $T + 1$ new queries, among which one will be certainly fresh (the size of z can be used to obtain an upper bound for the value of T).

(ii) To be EPRO ZK, the simulator should output a list $\mathcal{L}$ at which RO is programmed. It must also output a pair $(r, \mathrm{RO}[\mathcal{L}](x \mid\mid r))$, where $\mathrm{RO}[\mathcal{L}]$ denotes RO programmed according to the pairs in $\mathcal{L}$.

There are two possible ways for the simulated view to be accepted:

(a) The list $\mathcal{L}$ includes the query $x \mid\mid r$. The probability of this event happening for infinitely many n's is negligible, because it means S_{V^*} managed to invert $f_n(y)$ and obtain x.

[12] An a_i might be a function applied to several other answers. However, as such functions can later be computed by both V^* and D, there is no loss of generality in ignoring them.

(b) $\mathcal{L}$ does not include the query $x \mid\mid r$, but the simulator manages to query RO at point $x \mid\mid r$. Again, this happens with negligible probability for infinitely many n's, since otherwise we could exhibit an inverter for $\mathcal{F}$ (see Appendix C for a similar proof).

Therefore, Protocol 3 is not EPRO ZK.