One-Way Functions are Essential for Non-Trivial Zero-Knowledge

(EXTENDED ABSTRACT)

Rafail Ostrovsky[*] Avi Wigderson[†]

April 25, 2002

Abstract

It was known that if one-way functions exist, then there are zero-knowledge proofs for every language in $\mathcal{PSPACE}$. We prove that unless very *weak* one-way functions exist, Zero-Knowledge proofs can be given only for languages in $\mathcal{BPP}$. For average-case definitions of $\mathcal{BPP}$ we prove an analogue result under the assumption that *uniform* one-way functions do not exist.

[*] University of California at Berkeley and International Computer Science Institute at Berkeley. Part of this work was done at Bellcore and part at MIT.

[†] Princeton University and Hebrew University

1 Introduction

The complexity-theoretic approach to cryptography of the last several years has been to establish minimal complexity assumptions for basic cryptographic primitives and to establish connections among these primitives. In particular, at the very heart of cryptography is the notion of a one-way function [DH-76], which was shown to be necessary and sufficient for many cryptographic primitives. For example, pseudo-random generators [BM-82] and digital signatures [GMRi-88] were shown to be equivalent to the existence of one-way functions [ILL-89, Ha-90, R-90]. Moreover, many other cryptographic primitives, including identification, coin-flipping and secret key exchange were shown to imply the existence of a one-way function [ILu-89, BCG-89]. The subject of this paper is the relationship between one-way functions and zero-knowledge proofs.

1.1 Main notions and results

A function f is one-way if one efficient algorithm (encoder) can compute it, but no other efficient algorithm (inverter) can invert it too often. This notion has several flavors, depending on which of these two interacting algorithms is uniform. The standard one when both are uniform (i.e. Turing machines), f is called **uniform one-way**. (By f we actually mean a family of functions f_k, where k is a security parameter written in unary.)

In addition to uniform one-way functions, we consider **auxiliary input one-way** function which has both encoder and inverter uniform algorithms, with access to the same non-uniform input (e.g. the input to the proof system). That is, by auxiliary input one-way function f we denote a family of easy to compute functions $f_x(\cdot)$ (where x is a binary string) such that for infinitely many x, $f_x(\cdot)$ is almost always hard to invert. Note that if f is uniform one-way, then it is also auxiliary-input one-way, in which the auxiliary input is unary.

Interactive proofs, their knowledge complexity, and the associated complexity classes $\mathcal{IP}$ and $\mathcal{ZK}$ were introduced in the pioneering paper [GMR-85]. The class $\mathcal{IP}$ contains all languages L such that an infinitely powerful prover can convince a probabilistic efficient verifier to accept x for all $x \in L$, while no one can convince the same verifier to accept x when $x \notin L$. Note the two origins of non-uniformity in this interaction: the power of the prover and the externally given input x. The language L is in the class $\mathcal{ZK}$ if for $x \in L$ the proof above can be made to convey no knowledge (zero-knowledge) to any efficient verifier in a very strong sense: verifier could have generated the conversation with the prover by itself. Thus it is clear that trivial languages (in $\mathcal{BPP}$) trivially possess such proofs. We have:

FACT $\mathcal{BPP} \subseteq \mathcal{ZK} \subseteq \mathcal{IP}$.

In [GMW-86] it was shown that the existence of uniform one-way functions is **sufficient** for nontrivial zero-knowledge; they proved that this assumption implies $\mathcal{NP} \subseteq \mathcal{ZK}$. This result made zero-knowledge a central tool in secure protocol design and fault-tolerant distributed computing [GMW-86, GMW-87, Yao-86]. Extending this result, [IY-87, BGGHKMR-88] showed that in fact the existence of uniform one-way functions imply $\mathcal{ZK} = \mathcal{IP}$ (which we know to be equal to $\mathcal{PSPACE}$ [LFKN-90, S-90]). Thus, every provable language possesses a zero-knowledge proof, and $\mathcal{ZK}$ is as large as possible. Summarizing the sufficient condition we have

THEOREM [GMW-86,IY-87,LFKN-90,S-90] (Informal statement):
If uniform one-way functions exist then $\mathcal{ZK} = \mathcal{IP}$ $(= \mathcal{PSPACE})$.

We give two theorems which supply a weak converse to the above theorem. In particular, in our first theorem we show that auxiliary input one-way functions are **necessary** for nontrivial zero-knowledge. Namely, if they do not exist then $\mathcal{ZK}$ is as small as possible.

Theorem 1 *(Informal statement — Auxiliary-Input Version):*
If auxiliary-input one-way functions do not exist then $\mathcal{ZK}$= $\mathcal{BPP}$.

It is possible that auxiliary-input one-way functions exist, while uniform one-way functions do not. In this case it is possible to replace $\mathcal{BPP}$ in our theorem 1 with average $\mathcal{AVBPP}$, where the probability of correct acceptance/rejection is over a sampleable input distribution D and coin tosses of the algorithm.

Theorem 2 *(Informal statement — Average-Case Version):*
If uniform one-way functions do not exist then $\mathcal{ZK}$= $\mathcal{AVBPP}$.

REMARKS

- How strong a converse are our two theorems? It seems that to obtain a worst case complexity result it is impossible to avoid non-uniformity in the definition of one-way function, due to the (non-uniform) input to the proof system. On the other hand, we show that it is possible to obtain an average case complexity result assuming only the nonexistence of uniform one-way functions. Also, it seems that we can obtain an equivalence between the notions of zero-knowledge and one-way functions by defining both completely non-uniformly. Moreover, if we consider non-uniform case of $\mathcal{BPP}$ and $\mathcal{AVBPP}$ (i.e. family of circuits for each input length) then we can restate our results in terms of nonuniform (i.e. circuits) one-way and auxiliary one-way functions. We will pursue this further in the final version.

  Another point to remark on, is that in the sufficient condition, it is natural to assume that one-way functions exist for *all* input length, while in the necessary condition it is natural to assume that they do not exist for all input length. These are not complementary conditions, but like most other results of this type, both theorems have analogs where one-way functions exist for subsets of input length.

- Both our theorems are much stronger than stated: they hold even if we relax the definition of zero-knowledge condition to hold for *honest verifier* only. Since it is much easier to hide information from someone who does not cheat, constructing a zero-knowledge proof for this case is typically much easier.

1.2 Previous work and techniques

There are two previous results which deduce the existence of one-way functions from the existence of nontrivial zero-knowledge proofs of very special kinds. In [FS-89, D-89] it was shown that for an *AM* zero-knowledge protocol (i.e., the one restricted to a single round in which verifier can only send a single random string [B-85, GS-86]) of possession of information for a hard on the average [Le-86] problem does imply a bit-commitment scheme and, hence, a one-way function [ILu-89]. In [Ost-91] it was shown that any *statistical* zero-knowledge proof [F-87] for a hard on the average problem implies the existence of a one-way function. In this paper, we show that *any* nontrivial zero-knowledge proof (for the original, computational definition of zero-knowledge) implies the existence of a one-way function.

Our proof utilizes in essential ways many of the techniques (and their consequences) that were developed by [ILL-89, Ha-90] for showing the equivalence of one-way functions and pseudorandom generators. In particular, we use the notions of distributional one-way functions [ILu-89], efficient universal extrapolation [ILe-90], admissible random strings of the verifier [F-87], a note of [G-89] on computational versus statistical indistinguishability, and [Ost-91] algorithm.

The above developments have brought the field of cryptography to a point which allows fairly clean definitions of various primitives. This in turn enables to carry out formal, rigorous proofs in a clean, short form. We demonstrate this by isolating the ingredients needed in the proof in a few axioms (each a theorem of course, some under our assumption that there are no one-way functions), from which the proof is formally derived.

As part of our proof we establish a new connection between computational versus statistical indistinguishability in case when they satisfy certain additional constraints but one of the two distributions is not sampleable. We believe that this fact (axiom B3) is of independent interest and can be used elsewhere.

1.3 Corollaries

Under the assumption that one-way functions exist, many properties of (computational) zero-knowledge proofs were established. As alluded in the remarks above, theorem 2 shows that the existence of languages which can not be efficiently decided in $\mathcal{AVBPP}$ and have zero-knowledge proofs already implies the existence of one-way functions. Thus, theorems of the form "if one-way functions exist, then some property for languages outside $\mathcal{AVBPP}$ which have zero-knowledge proofs" can be now re-stated without assuming that one-way function exists, since it is already being implied.

Previously ([GS-86, GMS-87, LFKN-90, S-90]), it was known that *assuming that one-way functions exists*, $\mathcal{ZK}$ class is closed under complement; that any $\mathcal{ZK}$ proof can be turned into $\mathcal{ZK}$ proof in which verifier just tosses public coins; that the power of the prover can be bounded to be probabilistic PSPACE; that such proofs can be made one-sided (i.e. when x is in the language, prover convinces verifier with probability 1). As a corollary of our second theorem, we can now state the above results without the above (as we show redundant) assumption for the class of languages not in $\mathcal{AVBPP}$ which have $\mathcal{ZK}$ proofs (where we say that language L is not in $\mathcal{AVBPP}$ if there exists a sampleable distribution D such that for all sufficiently large input length there is no algorithm (or family of circuits in the nonuniform case) which can decide

L with probability (over D and coin flips of the inverting algorithm or circuit) bounded away from half.) In addition, in the full version of the paper, we show that the class of languages which has a $\mathcal{ZK}$ proof *for honest verifier only* is equivalent to $\mathcal{ZK}$ (i.e. class of languages which has $\mathcal{ZK}$ proof for any, even cheating verifier.)

Our results also extends to Non-Interactive Zero-Knowledge (NIZK) in the common random string model, introduced by Blum, Feldman and Micali [BFM-88], where we can show that for languages outside $\mathcal{AVBPP}$ which have NIZK proof implies the existence of a uniform one-way function. Moreover, we can restate our theorem for ZK arguments [BCC-88] as well. We explore this and other applications of our result (to ZK proofs of knowledge and knowledge complexity) in the full version of this paper.

1.4 Organization of the paper

The next section is devoted to an intuitive description of the important definitions, and of the basic results we shall use. It is paralleled by a similar section in the appendix which treats these formally. In section 3 we give an intuitive description of the proof of the main theorem, trying to point to the subtleties. Again, this is paralleled by a section in the appendix containing the full formal proof.

2 (Informal) Definitions and basic results

This section informally defines the necessary notions and describes the required results for proving the main theorem. As usual we will refer here to input, input length, distribution, probability, when we should really talk about an infinite sequence of such objects. The formal definitions and results appear in the appendix in a similar partitions to subsections, so the reader can conveniently look them up. Moreover, the appendix contains more elaborate discussions.

Probabilistic and Efficient Turing machines

Informally, $\mathcal{PTM}$ is the class of all probabilistic Turing machines, whose output length is polynomial in the input length. $\mathcal{PPT}$ is the class of all probabilistic Turing machines M that on input x halt in polynomial time. The distributions $M(x)$ generated by machines in $\mathcal{PPT}$ are called (efficiently) sampleable. For a distribution D (on strings), $M(D)$ will denote the the output distribution of M where the input x is chosen at random from D. The languages recognized by $\mathcal{PPT}$ machines form the class $\mathcal{BPP}$.

One-way functions

A one way function can be best described as a game between two $\mathcal{PPT}$ machines M and N. On common input x, M computes the distribution $M(x) = (y, z)$ (think of $y = f(z)$). On input (x, y), N is trying to "invert" f, i.e., to compute z' such that $y = f(z')$. The machine M wins on x if the probability that N succeeds is negligible, where probability is taken over coin tosses of M and N. If x is unary, the notion of a one-way is a standard one, where x corresponds to a security parameter of a one-way function. (In fact, for unary x, definitions of weak and strong one-way functions presented below correspond to standard definitions of weak and strong one-way functions, see for example [ILL-89, Ha-90, Lu-92]. If x is binary, this captures a notion of a one-way function with auxiliary input, given to both players. We explicitly specify whether x is binary on unary.

Negligible fractions and statistical closeness of distributions ($\overset{s}{=}$)

A negligible probability is a fraction smaller than the inverse of any polynomial (in the input length). Other fractions are non-negligible. For two probability distributions (random variables) on strings D, E, we say that $D \overset{s}{=} E$ if the L_1 norm of their difference, $||D - E||_1$ is negligible. Note that we can apply transitivity to the relation $\overset{s}{=}$ polynomially many times. Also, for any $M \in \mathcal{PTM}$, if $D \overset{s}{=} E$ then $(D, M(D)) \overset{s}{=} (E, M(E))$. That is, let D_T, E_T be two ensembles, then we say that $D_T \overset{s}{=} E_T$ if $\forall c > 0, x \in T$, we have
$||D^x - E^x||_1 \leq O(|x|^{-c})$, where $|| \; ||_1$ is the L_1 norm.

Universal extrapolation and approximation

While in the definition of one-way functions the machine N is requested "merely" to find any legal continuation of the given partial output of machine M, here it is required to produce essentially the same distribution. Consider again machines M with $M(x) = (y, z)$. Informally, by *universal extrapolation* we mean that for every $M \in \mathcal{PPT}$ there exists $N \in \mathcal{PPT}$ satisfying for all $x \in \Sigma^*$, $M(x) \overset{s}{=} (y, N(x, y))$.

In the definition if universal extrapolation, machine N is required to find a legal continuation of the given partial output of machine M essentially preserving the distribution. *Universal Approximation* requires to estimate the number of possible continuations of M with arbitrary accuracy. That is, consider again machines M with $M(x) = (y, z)$ and let $Z_{x,y} \overset{\triangle}{=} \{z | M(x) = (y, z)\}$. Informally, by *universal approximation* we mean that for every $M \in \mathcal{PPT}$ there exists $N \in \mathcal{PPT}$ satisfying for all $x \in \Sigma^*$, given y s.t. $M(x) = (y, z)$, $|Z_{x,y}|$ can be approximated within a constant fraction by $N(x, y)$ with high probability.

We note that assuming that there are no one-way functions, universal extrapolation and approximation is possible [ILL-89, Ha-90, ILu-89, ILe-90, F-87].

Computationally indistinguishable distributions ($\overset{c}{=}$)

Intuitively, D and E are computationally indistinguishable if no machine in $\mathcal{PPT}$ can tell them apart with non-negligible probability. Rephrasing, $D \overset{c}{=} E$ if for every boolean $N \in \mathcal{PPT}$ (i.e. that one which outputs either 0 or 1) $N(D) \overset{s}{=} N(E)$. We note that if there are no one-way functions, then computational equivalence of two sampleable distributions implies statistical equivalence [G-89].

Interactive machines and conversations

A conversation (of n rounds, on common input x) between machines $P, V \in \mathcal{PTM}$ is the sequences $C = C^{PV} = C^{PV(x)} = (m_1 \# m_2 \# \cdots \# m_n)$ (we shall omit as many superscripts as possible when there is no danger of confusion) of messages they alternatingly send each other (starting, say with V). The i'th prefix of the conversation, with $0 \leq i \leq n$, denoted $C_i = C_i^{PV}$ is the sequence of first i messages. It will be useful to make the random tape of V explicit. We call it $R = R^{PV}$, and assume w.l.o.g. it is initially chosen uniformly at random from $\{0,1\}^n$. The *transcript* of the conversation is Z^{PV} is a pair $(R^{PV} \# C^{PV})$. It is crucial to observe that messages of P *do not* depend on R, only on previous messages.

Let $M \in \mathcal{PTM}, N \in \mathcal{PPT}$ and $\hat{N}$ be the deterministic analog of N. The *transcript* of the conversation between M and N on $T \subseteq \{0,1\}^n$ is the ensemble $Z_T^{MN} = \{Z_{\{x \in T\}}^{MN(x)}\}$. It is defined by $Z^{MN(x)} = x \# R \# m_1 \# m_2 \# \cdots \# m_n$, with $n = |x|^{|N|}$, $|R| = |m_i| = n$ for all i inductively as follows:

- $R \in U^n$ (R is uniformly distributed over $\{0,1\}^n$).
- $C_0 = \emptyset$ (the empty string).
- $$m_{i+1} = \begin{cases} \hat{N}(x \# R \# C_i) & \text{for even } i \geq 0 \\ M(x \# C_i) & \text{for odd } i \geq 0 \end{cases}$$

  and $C_{i+1} = C_i \# m_{i+1}$ for all $i \geq 0$.

Interactive and Zero-Knowledge proofs

A language $L \subseteq \{0,1\}^*$ is in $\mathcal{IP}$ if there are $P \in \mathcal{PTM}$ (called the 'prover'), and $V \in \mathcal{PPT}$ (called the 'verifier', whose final message is 'accept' or 'reject') such that

1. $\Pr[m_n^{PV(x)} = 1^n] \geq \frac{2}{3}$ for every $x \in L$.
2. $\Pr[m_n^{\bar{P}V(x)} = 1^n] \leq \frac{1}{3}$ for every $\bar{P} \in \mathcal{PTM}$ and $x \notin L$

where in both cases the probability space is over the coin tosses of the machines. The pair (P, V) is called an interactive proof for L.

Let (P, V) be an interactive proof for L. Intuitively, this proof is zero-knowledge if for every $\bar{V}$, a distribution indistinguishable from the transcript $Z^{P\bar{V}(x)}$ (which is defined $R^{PV} \# C^{PV}$) can be generated in $\mathcal{PPT}$ for every $x \in L$. Formally, $L \in \mathcal{ZK}$ if for all $\bar{V} \in \mathcal{PPT}$ there exists $S^{\bar{V}} \in \mathcal{PPT}$ (called the "simulator") such that $Z_L^{P\bar{V}} \overset{c}{=} S_L^{\bar{V}}$, where $Z_L^{P\bar{V}} = \{Z^{P\bar{V}(x)}\}_{x \in L}$ and $S_L^{\bar{V}} = \{S^{\bar{V}(x)}\}_{x \in L}$. (We note that in the definition of statistical zero-knowledge ([F-87]) the 'only' difference is that the last two distributions are required to be statistically close). Now, we wish to stress several properties of zero-knowledge interactive proofs, which are going to be essential in our proof.

Remarks:

- For all $i \geq 0$ and all $c_i \in C_i^{PV}$ define *consistent*$(c_i) \overset{\triangle}{=} \{r | r \in (R^{PV} \# c_i)\}$, i.e. all the random strings of the verifier which are consistent with the prefix of the conversation c_i. We claim that for any $i \geq 0$ and any $c_i \in C_i^{PV}$ (i.e. prefix of the conversation between prover and honest verifier) the distribution $R^{PV} \# c_i$ is flat (i.e. uniformly distributed) over *consistent*(c_i). To see that this is so, notice, that for $i = 0$ this is clearly the case. For all $i > 0$ it follows by induction, as when prover "speaks", it does not have access to the verifiers random tape, so the set does not change, and when verifier "speaks" at round $i+1$ it restricts *consistent*(c_i) to a subset *consistent*$(c_i \# \hat{V}(x, (r \in$ *consistent*$(c_i)), c_i))$ which is again uniformly distributed, where $\hat{V}$ is a deterministic analog of V with its random bits fixed to be $r \in$ *consistent*(c_i).

- WLOG, we can assume that in $S^{V(x)}$ the messages of honest verifier V are "legal", that is, for every even $i \geq 0$ of transcript $R^S \# C^S_{i+1}$ it is the case that $m_{i+1} = \hat{V}(x \# R^S \# C_i)$. Notice that this can be assumed without loss of generality as otherwise whenever this is not the case we can trivially distinguish outputs of the simulator from the real conversations, where this *is* always the case.

An important corollary to universal extrapolation, is that for any $S \in \mathcal{PPT}$ there is another machine $S^{-1} \in \mathcal{PPT}$ that "inverts" the random bits of S used to produced a (partial) transcript of the simulated conversation. That is, for a fixed $x \in L$ and partial prefix of the conversation c_i, $S^{-1}(c_i)$ finds a random ω such that the prefix of $S(\omega) = r \# c_i$.

¿From now on, when clear from the context, we abuse the notation and for a fixed c_i, by $S^{-1}(c_i)$ denote either distribution on ω or the distribution of r's which $S(\omega)$ produces (by universal extrapolation). In particular, when we write $(S^{-1}(c_i), c_i)$, by $S^{-1}(c_i)$ we denote the distribution on r's. If for some c_i, $S^{-1}(c_i)$ fails to find ω, then it prints a special "reject" symbol.

Remark: An important superclass of $\mathcal{ZK}$, $\mathcal{ZKHV}$ (Zero-Knowledge for Honest Verifier), is when we demand from (P, V) only that the real transcript can be generated, i.e. $\exists S = S^V$ such that $Z^{PV}_L \overset{c}{=} S^V_L$. Clearly, $\mathcal{ZK} \subseteq \mathcal{ZKHV}$. The usual difficulty in constructing zero-knowledge proofs is when $\bar{V} \neq V$, as we know V, but $\bar{V}$ can be arbitrary. However, our proof of the main theorem only uses the zero-knowledge property for the honest verifier, and thus in the main theorem one can replace $\mathcal{ZK}$ with the (possibly larger) $\mathcal{ZKHV}$.

3 Proof sketch of Theorem 1

We start with a zero-knowledge proof system (P, V) for the language L, with the associated simulator S, and wish to derive an $\mathcal{BPP}$ algorithm for recognizing L. Such an algorithm was given in [Ost-91], for the case that (P, V) is statistical zero-knowledge proof of L. It will turn out that the same algorithm will work for us, but this will require some extra arguments as will become clear later. Let us describe this algorithm A.

Algorithm A: On input x, this algorithm will generate transcripts of conversations $Z^{A(x)}$ between the real verifier V with random tape $R = R^A$, and a 'fake' prover P^* which we next define. P^* will "extrapolate" the simulator S on prover's messages, i.e. will satisfy $C^S_i, P^*(C^S_i) \overset{s}{=} C^S_{i+1}$. By theorem 3 P^*, and hence A are in $\mathcal{PPT}$.

Note that P^* does not use the verifier's random tape R^A in this conversation, and hence by the definition of interactive proofs, if $x \notin L$, V (and hence A) will reject x with probability $\geq \frac{2}{3}$. This part of the argument does not use zero-knowledge, and will work for us as well. The hard part is showing that A will accept most of the time when $x \in L$. We first recall the way it was done in the statistical zero-knowledge, and then show our proof for the computational case.

Statistical zero-knowledge ($x \in L$)

The main claim is that when $x \in L$ then $C^{A(x)} \overset{s}{=} C^{PV(x)}$, which guarantees acceptance with probability $\geq \frac{2}{3}$. The proof will use only that $C^{S(x)} \overset{s}{=} C^{PV(x)}$, and thus does not require the simulator to generate the whole transcript (which includes the verifier's random tape). This is one crucial difference to the computational case, in which such weak simulation will not suffice!

It follows by induction on i, showing $C_i^A \overset{s}{=} C_i^S \overset{s}{=} C_i^{PV}$ (we shall omit the fixed $x \in L$ from now on). Note that the base $i = 0$ trivially holds, and that

$$(*) \text{ for all } i, C_i^S \overset{s}{=} C_i^{PV}$$

holds by the perfect zero knowledge property. The inductive step proceeds differently, according to whether $i + 1$ is a Prover's message (case P) or a verifier's message (case V). Note that (*) allows us to prove that C_i^A is $\overset{s}{=}$ to either one of C_i^S or C_i^{PV}.

Case V: Here we show $C_{i+1}^A \overset{s}{=} C_{i+1}^{PV}$. In both cases the same $\hat{V}$ (i.e. deterministic version of V) is applied to $R^A \# C_i^A$ and $R^{PV} \# C_i^{PV}$. Moreover, by inductive hypothesis, we assume that $R^A \# C_i^A \overset{s}{=} R^{PV} \# C_i^{PV}$. However, it is the case that when we apply the same deterministic poly-time algorithm to two statistically close distributions, we get a distribution which is statistically close.

Case P: First, notice that when it is provers turn to speak, the set R^{PV} does not change, as prover does not "see" verifiers random tape. Similarly, algorithm A is designed so that it does not use random tape of the verifier. Thus, what could break statistical equality? Only messages of the prover in algorithm A vs. the actual conversation. But how does algorithm A compute the next message of the prover? It first finds a uniformly distributed ω to that $S(\omega)$ outputs $(x, r, C_i \# m_{i+1} \# \ldots \# m_n)$ and outputs m_{i+1}. However, observe that $C_{i+1}^A \overset{s}{=} C_{i+1}^S$, which follows from the property of P^* above, namely that it extrapolates the S nearly perfectly on C_i. However, since we are in case of *statistical* $\mathcal{ZK}$, we know that S is statistically close to the real conversation and we are done.

Computational zero-knowledge ($x \in L$)

Our main lemma asserts that $Z^A \overset{c}{=} Z^{PV}$. The first difference to observe from the statistical case is that here we need to explicitly use the random tape in all transcripts.

We know that

$$(**) \text{ for all } i, R^S, C_i^S \overset{c}{=} R^{PV}, C_i^{PV}$$

To see intuitively where the action is, let i_0 be the last i for which this $\overset{c}{=}$ can be replaced by $\overset{s}{=}$ (clearly we can do it for $i = 0$). If $i_0 = n$, then the proof is statistical zero-knowledge and we are done. Otherwise, the distributions R^S, C_{i+1}^S and R^{PV}, C_i^{PV} are $\overset{c}{=}$ but not $\overset{s}{=}$. If they were both efficiently sampleable, Theorem 5 would rule out this possibility and we would be done. But the real conversation is not, and we must be more refined.

We will prove by induction that $R^A, C_i^A \overset{s}{=} R^S, C_i^S$, and at the same time $R^A, C_i^A \overset{c}{=} R^{PV}, C_i^{PV}$. Here we can use Theorem 5 and (**) to show that (like in the statistical case), it sufficient to prove either one of these two relations!

Case V This case is exactly the same as in statistical case. That is, here we prove the second relation for $i + 1$, which follows by induction since the same V is applied to R^A, C_i^A and R^{PV}, C_i^{PV}.

Case P This is the difficult case. Here we prove the first relation for $i + 1$. The problem is that when S simulates a prover's message, it may use R^S (while P^* in A is not allowed to use

R^A). Indeed, all known computational zero-knowledge proofs (which are not statistical) do this at some point (and when it first happens this is $i_0 + 1$ alluded to before). Moreover, in these proofs it happens exactly when P uses a one-way function! This is good news, as we assume there are no one-way functions. Thus we try to infer that in fact S does not "use" R^S at this step. Roughly speaking, if it did, since this does not happen in the real (PV) conversation, we would be able to distinguish the sizes of admissible R^S and R^{PV}.

More precisely, we note that given a prefix of the conversation, we can always "invert" it on a simulator, and estimate the number of random strings of the simulator which could produce this prefix of the conversation (using universal approximation). Then, we show that the number of random strings of the simulator helps us to estimate the number of admissible random strings in R^{PV}, since if we can not estimate it sufficiently close, then we would be able to construct a distinguisher. In the proof of the last fact, we use in essential way the fact that the distribution of admissible random strings produced in the real conversation is flat (i.e. uniformly distributed on some subset.) (We remark that to the best of our knowledge, this fact was not used in any prior work.)

More specifically, the main tool in proving the independence of the simulators behavior from the random string it finally produces for the verifier, is the following theorem, saying that the random tape of the verifier **in the real conversation**, may be obtained (statistically) from *the real conversation* (assuming there there are no one-way functions, of course) by inverting the simulator S on it, even though S is only *computationally* close to the real conversation:

THEOREM:

$$\forall i, R^{PV}\#C_i^{PV} \overset{s}{\doteq} S^{-1}(C_i^{PV})\#C_i^{PV}$$

Proof: Notice that from the definition of zero-knowledge and remarks that follow the definition of zero knowledge we know that:

(1) $R^S\#C_i^S \overset{c}{\doteq} R^{PV}\#C_i^{PV}$

(2) For any partial conversation $c_i \in C_i^{PV}$, $R^{PV}\#c_i$ is flat.

(3) For any partial conversation $c_i \in C_i^{PV}$, $\{r|r \in S^{-1}(c_i)\}$ is a subset of the support set of $R^{PV}\#c_i$.

Thus, to prove the above theorem, we must show that conditions (1), (2) (3) and our assumption that there is no one-way function imply the above theorem. This is exactly what the proof of "axiom" $B3$ achieves (but in a more general setting). The proof is given in the appendix. □

Acknowledgments

We would like to thank Oded Goldreich, Russell Impagliazzo, Mike Luby, Silvio Micali and Noam Nisan for helpful discussions. We are especially grateful to Russell Impagliazzo for discovering a bug in the initially fallacious proof of one of the important lemmas, which made us look harder for a correct one.

In addition, the first author wishes to thank Silvio Micali for introducing him to the field in 1989 and for numerous illuminating conversations since then, and the second author wishes to thank Oded Goldreich for a decade of conversations on these subtle topics, which were essential for really understanding them.

References

[B-85] L. Babai, "Trading Group Theory for Randomness", *Proc. 17th STOC, 1985, pp. 421–429.*

[BM-82] M. Blum, and S. Micali "How to Generate Cryptographically Strong Sequences Of Pseudo-Random Bits" *SIAM J. on Computing,* Vol 13, 1984, pp. 850-864, FOCS 82.

[BFM-88] M. Blum, P. Feldman and S. Micali "Non-interactive Zero-Knowledge Proof Systems" *Proc. 20th STOC, 1988, pp. 103-112.*

[BCC-88] G. Brassard, D. Chaum and C. Crépeau, *Minimum Disclosure Proofs of Knowledge*, JCSS, v. 37, pp 156-189.

[BP-92] M. Bellare, E. Petrank "Making Zero-Knowledge Provers Efficient" *Proc. 24th STOC, 1992, pp. 711-722.*

[BCG-89] M. Bellare, L. Cowen, and S. Goldwasser "Secret Key Exchange" *DIMACS-89 workshop on distributed computing and cryptography.*

[BGGHKMR-88] M. Ben-Or, O. Goldreich, S. Goldwasser, J. Hastad, J. Kilian, S. Micali and P. Rogaway "Everything Provable is Provable in Zero-Knowledge", *Crypto 88*

[GS-86] S. Goldwasser and M. Sipser "Private coins versus public coints in interactive proof systems" STOC 1986, also in *Advances in Computing Research 5: Randomness and Computation* S. Micali, ed., JAI Press, Greenwich, CT, 1989.

[DH-76] W. Diffie, M. Hellman, "New dirctions in cryptography", *IEEE Trans. on Inf. Theory,* IT-22, pp. 644–654, 1976.

[GMS-87] Goldreich, O., Y. Mansour, and M. Sipser, "Interactive Proof Systems: Provers that never Fail and Random Selection," FOCS 87. Journal version by Furer. M., Goldreich, O., Mansour, Y., Sipser, M., and Zachos, S., "On Completeness and soundness in interactive proof systems" in *Advances in Computing Research 5: Randomness and Computation*, Micali, S., ed., JAI Press, Greenwich, CT, 1989.

[LFKN-90] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. "Algebraic Methods for Interactive Proof Systems", *Proc. of the 31st FOCS* (St. Louis, MO; October, 1990), IEEE, 2–10.

[NW-88] N. Nissan, and A. Wigderson, "Hardness vs. Randomness" *FOCS 88.*

[D-89] I. Damgard "On the existence of bit commitment schemes and zero-knowledge proofs" *CTYPTO 89*

[F-87] L. Fortnow, “The Complexity of Perfect Zero-Knowledge” STOC 87; also in *Advances in Computing Research 5: Randomness and Computation*, Micali, S., ed., JAI Press, Greenwich, CT, 1989.

[FS-89] U. Feige, and A. Shamir, “Zero Knowledge Proofs of Knowledge in Two Rounds” *CRYPTO 89.*

[GMR-85] S. Goldwasser, S. Micali and C. Rackoff, “The Knowledge Complexity of Interactive Proof-Systems”, *SIAM J. Comput.* 18 (1989), pp. 186-208; (also in STOC 85, pp. 291-304.)

[GMRi-88] S. Goldwasser, S Micali and R. Rivest, “A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacs” *SIAM J. Comput.,* 17 (1988), pp.281-308.

[G-89] O. Goldreich “A Note On Computational Indistinguishability”, *Manuscript* August 10, 1989.

[GK-89] O. Goldreich, H. Krawczyk “Sparse Pseudorandom Distributions”, *Crypto-89*

[GL-89] O. Goldreich, and L. Levin “A Hard-Core Predicate for all One-Way Functions” *STOC, 89*, pp.25-32.

[GMW-86] O. Goldreich, S. Micali, and A. Wigderson, “Proofs that Yield Nothing but their Validity”, *FOCS 86*; also J. ACM (to appear)

[GMW-87] O. Goldreich, S. Micali and A. Wigderson, “A Completeness Theorem for Protocols with Honest Majority,” *STOC 87.*

[GS-86] S. Goldwasser and M. Sipser, “Private Coins versus Public Coins”, *STOC, 1986*; also in *Advances in Computing Research 5: Randomness and Computation*, Micali, S., ed., JAI Press, Greenwich, CT, 1989.

[Ha-90] J. Hastad, “Pseudo-Random Generators under Uniform Assumptions” *STOC 90*

[ILe-90] R. Impagliazzo and L. Levin “No Better Ways to Generate Hard NP Instances than Picking Uniformly at Random” *FOCS 90.*

[ILu-89] R. Impagliazzo and M. Luby, “ One-way Functions are Essential for Complexity-Based Cryptography” *FOCS 89.*

[ILL-89] R. Impagliazzo, R., L. Levin, and M. Luby “Pseudo-Random Generation from One-Way Functions,” *STOC 89.*

[IY-87] R. Impagliazzo, and M. Yung “Direct Minimum-Knowledge Computation” *Crypto 87.*

[IZ-89] R. Impagliazzo and D. Zukerman “How to Recycle Random Bits” *FOCS 89.*

[Lu-92] Luby, M., “Pseudo-randomness and Applications” monograph in progress.

[Le-86] L. Levin “Average Case Complete Problems”, *SIAM Journal of Computing* 15: 285-286, 1986.

[Ost-91] R. Ostrovsky "One-way functions, Hard-on-Average Problems, and Statistical Zero-Knowledge Proofs" *Structures in Complexity Theory* 91.

[R-90] J. Rompel "One-way functions are Necessary and Sufficient for Secure Signatures" *STOC* 90.

[S-90] A. Shamir. "IP = PSPACE", *Proc. of the 31st FOCS* (St. Louis, MO; October, 1990), IEEE, 11–15.

[Yao-86] A.C. Yao "How to Generate and Exchange Secrets" FOCS 86.

[Yao-82] A.C. Yao "Theory and Applications of Trapdoor Functions" *FOCS 82.*

4 APPENDIX 1: (Formal) Definitions and Basic Results

This section defines the necessary notions and describes the required results for proving our main results. As usual we will refer here to input, input length, distribution, probability, when we should really talk about an infinite sequence of such objects. We follow informal discussion by formal definitions.

Probabilistic and Efficient Turing machines

For a probabilistic Turing machine M, $x \in \Sigma^*$, $M(x)$ denotes the output distribution of M on input x (without l.o.g. we assume that $M(x)$ has fixed, $|x|^{|M|}$ length). Let $|x|$ denote the length of the string x, and $|M|$ denotes the length of the description of the Turing machine M. Thus every $M \in \mathcal{PTM}$ satisfies $|M(x)| \leq O(|x|^{|M|})$. $\mathcal{PPT}$ is the class of all probabilistic Turing machines M that on input x halt after $O(|x|^{3|M|})$ steps. The distributions generated by machines in $\mathcal{PPT}$ are called (efficiently) sampleable. It will be convenient (and of no loss of generality) to assume that M uses exactly $|x|^{|M|}$ random bits R, and to consider $\hat{M}$ the deterministic analog of M with $\hat{M}(R\#x) = M(x)$, where R is chosen uniformly from $\{0,1\}^{|x|^{|M|}}$.

One-way functions

Let Σ be a finite input alphabet. Let $M \in \mathcal{PTM}$ be such that $M(x) = M_1(x)\#M_2(x)$. ($M_2(x)$ may be thought of as an inverse of $M_1(x)$.)

Call x *hard* for $N \in \mathcal{PPT}$,

$$\Pr[M_1(x)\#N(x, M_1(x)) \text{ is an output of } M(x)] \leq O(|x|^{-|N|})$$

where probability is taken over coin tosses of M and N.

Let $H_N(M) \subseteq \Sigma^*$ be the set of all hard x for N. Now we can make explicit:

- ($\exists S1WF$) There exist strong one-way functions $\triangleq \exists M \in \mathcal{PPT}$ such that $\forall N \in \mathcal{PPT}$ $|\Sigma^* - H_N(M)| < \infty$.
- ($\exists 1WF$) There exist one-way functions $\triangleq \exists M \in \mathcal{PPT}$ such that $\forall N \in \mathcal{PPT}$, $|H_N(M)| = \infty$.
- ($\nexists 1WF$) There are no one-way functions $\triangleq \forall M \in \mathcal{PPT}$, $\exists N \in \mathcal{PPT}$ $|H_N(M)| < \infty$.

Remarks:

1. The definition of $\exists 1WF$ is the weakest possible. Stronger definitions are possible, we have chosen the one above for concreteness — any choice would give a result that will essentially say that non-trivial $\mathcal{ZK}$ proofs exist whenever $1WF$ exists.
2. Analogous non-uniform definitions (i.e. where both M and N are families of circuits) can be defined and our results can be transformed into non-uniform model as well.

3. There is a difference between choosing $\Sigma = \{0,1\}$ and $\Sigma = \{1\}$. The *unary* case $\Sigma = \{1\}$ is the standard one considered in most of the literature when only length is given.

   The *binary* case $\Sigma = \{0,1\}$ captures the case of an auxiliary input given to all participating machines (e.g. the input to a $\mathcal{ZK}$ proof system). We assume $\not\exists 1WF$ for binary Σ and obtain results on worst case complexity. We remark that this is the strongest assumption that can be made.

Ensembles of Probabilistic Distribution

For $T \subseteq \Sigma^*$, $D_T = \{D^x\}_{x\in T}$ will denote the collection of probability distributions (or random variables) on strings over $\{0,1,\#\}$, indexed by elements of T. We will always assume for simplicity that $D^x = D^x_1\#D^x_2\#\cdots\#D^x_k$, with $D^x_i \in \{0,1\}^*$, that $|D^x_i| = \Theta(|X|^c)$ for a fixed constant c. Of importance are ensembles generated by probabilistic Turing machines. For $M \in \mathcal{PTM}$, $T \subseteq \Sigma^*$ denotes all $M_T = \{M(x)\}_{x\in T}$. Denote by U^n the uniform distribution of $\{0,1\}^n$.

Universal extrapolation and approximation

Consider machines M with $M(x) = M_1(x)\#M_2(x)$.

(UE) Universal Extrapolation $\overset{\triangle}{=}$ For every $M \in \mathcal{PPT}$ there exists $N \in \mathcal{PPT}$ satisfying for all $x \in \Sigma^*$,

$$||M_1(x)\#M_2(x) \; - \; M_1(x)\#N(x, M_1(x))||_1 \leq O(|x|^{-|M|})$$

Note that the distribution $M(x)$ can be generated by an infinite sequence of machines M^i (by padding M). The associated machines N^i extrapolate $M(x)$ arbitrarily well. It will be useful to abuse notation and redefine *UE informally* by

$$\forall M \in \mathcal{PPT} \quad \exists N^\infty \in \mathcal{PPT} \text{ (the ``limit'' of } N^i\text{) such that}$$

$$M(x) \overset{s}{=} M_1(x)\#N^\infty(x, M_1(x))$$

It will be more convenient to work with the informal definition. We remark that this is similar to the way arbitrarily small or large numbers are defined in nonstandard models of analysis.

Here is the first major consequence of our assumption $\not\exists 1WF$; It follows from the referenced papers, though they prove it for one-way functions without auxiliary input (i.e. they consider $\Sigma = \{1\}$).

Theorem 3 ([ILL-89, Ha-90, ILu-89, ILe-90]:)

$$\not\exists 1WF \Longrightarrow UE$$

Remark: Again, this result can be interpreted under different definitions of one-way functions. As noted in [ILe], in general *UE* is possible whenever there are no one-way functions (e.g. for a sequence of input lengths).

In the definition if universal extrapolation, machine N is required to find a legal continuation of the given partial output of machine M essentially preserving the distribution. *Universal Approximation* requires to estimate the number of possible continuations of M with arbitrary accuracy. That is, consider again machines M with $M(x) = (y, z)$ and let $Z_{x,y} \stackrel{\triangle}{=} \{z|M(x) = (y, z)\}$. Informally, by *universal approximation* we mean that for every $M \in \mathcal{PPT}$ there exists $N \in \mathcal{PPT}$ satisfying for all $x \in \Sigma^*$, given y s.t. $M(x) = (y, z)$, $|Z_{x,y}|$ can be approximated within a constant fraction by $N(x, y)$ with high probability. More formally, consider again machines M with $M(x) = M_1(x)\#M_2(x)$. For any $x \in \Sigma^*$ define $|M_2|_{x,y} \stackrel{\triangle}{=} |\{M_2(x) : M_1(x) = y\}|$

(UA) Universal Approximation $\stackrel{\triangle}{=}$ For every $M \in \mathcal{PPT}$ and for every $0 < \alpha < 1$ there exists $N \in \mathcal{PPT}$ (which runs in time polynomial in x and $\frac{1}{\alpha}$) satisfying for all $x \in \Sigma^*$

$$Prob\Big((1-\alpha)|M_2|_{x,M_1(x)} < N(x, M_1(x), \frac{1}{\alpha}) < (1+\alpha)|M_2|_{x,M_1(x)}\Big) \geq 1 - O(|x|^{-|M|})$$

Again note that the distribution $M(x)$ can be generated by an infinite sequence of machines M^i (by padding M). Here is the second major consequence of our assumption $\not\exists 1WF$:

Theorem 4

$$\not\exists 1WF \Longrightarrow UA$$

Computationally indistinguishable distributions ($\stackrel{c}{=}$)

Intuitively, D and E are computationally indistinguishable if no machine in $\mathcal{PPT}$ can tell them apart with non-negligible probability. Rephrasing, $D \stackrel{c}{=} E$ if for every boolean $N \in \mathcal{PPT}$ (i.e. that one which outputs either 0 or 1) $N(D) \stackrel{s}{=} N(E)$. Formally, $D_T \stackrel{c}{=} E_T$ if for all $N \in \mathcal{PPT}, x \in T$,

$$||N(x, D^x) - N(x, E^x)||_1 \leq O(|x|^{-|N|})$$

As in the case of $\stackrel{s}{=}$, we can apply transitivity to $\stackrel{c}{=}$ polynomially many times. Also, if $D \stackrel{c}{=} E$ and $M \in PPT$, then $(D, M(D)) \stackrel{c}{=} (E, M(E))$.

We are now ready for the second major consequence of our assumption $\not\exists 1WF$. It is clear from the definitions that statistical closeness is a stronger condition than computational indistinguishability, i.e $(D \stackrel{s}{=} S) \Longrightarrow (D \stackrel{c}{=} E)$. Assuming $\not\exists 1WF$, the converse is true for all pairs D, E which can be generated efficiently! (We remark that both D and E are sampleable is necessary, as shown by [GK-89]).

Theorem 5 ([ILL-89, Ha-90, G-89]) Assuming $\not\exists 1WF$, if $M, N \in \mathcal{PPT}$, then

$$(M \stackrel{c}{=} N) \Longrightarrow (M \stackrel{s}{=} N)$$

A couple of other useful facts about computational indistinguishability are given below. For $M \in \mathcal{PTM}$ and ensemble $D = D_T$, we denote by $M(D_T)$ the ensemble $\{M(x, D^x)\}_{x \in T}$. It it the case that If $D \stackrel{c}{=} E$ and $M \in \mathcal{PPT}$, then $D\#M(D) \stackrel{c}{=} E\#M(E)$. Moreover, if $M \in \mathcal{PPT}$ and $D\#E \stackrel{c}{=} F\#M(F)$, then $D\#E \stackrel{c}{=} D\#M(D)$.

BPP

For convenience, we define *BPP* by $\{0,1\}^n \supseteq L \in \mathcal{BPP}$ iff $\exists M \in \mathcal{PPT}$ such that $\forall x \in \Sigma^*$, $\Pr[M(x) = L(x)] \geq \frac{3}{5}$. (Probability over coin tosses of M.)

4.1 Average Case Complexity

Let $D_{\mathcal{N}}$ be an ensemble on $\{0,1\}^*$ with D^n sampleable distributions over $\{0,1\}^n$. The pair $(L, D) \in \mathcal{AVBPP}$ if $\exists M \in \mathcal{PPT}$ s.t. $\forall n \in \mathcal{N}$, $\Pr[M(x) = L(x)] \geq \frac{1}{2} + |x|^{-|M|}$, where the probability is over $x \in D^n$ and coin tosses of M.

Interactive Machines and Structured Conversations

Let $M \in \mathcal{PTM}, N \in \mathcal{PPT}$ and $\hat{N}$ be the deterministic analog of N. The *transcript* of the conversation between M and N on $T \subseteq \{0,1\}^n$ is the ensemble $Z_T^{MN} = \{Z_{\{x \in T\}}^{MN(x)}\}$. It is defined by $Z^{MN(x)} = x\#R\#m_1\#m_2\# \cdots \#m_n$, with $n = |x|^{|N|}$, $|R| = |m_i| = n$ for all i inductively as follows:

- $R \in U^n$ (R is uniformly distributed over $\{0,1\}^n$).
- $C_0 = \emptyset$ (the empty string).
- $$m_{i+1} = \begin{cases} \hat{N}(x\#R\#C_i) & \text{for even } i \geq 0 \\ M(x\#C_i) & \text{for odd } i \geq 0 \end{cases}$$

  and $C_{i+1} = C_i\#m_{i+1}$ for all $i \geq 0$.

Interactive Proofs

A language $L \subseteq \{0,1\}^*$ is in $\mathcal{IP}$ if there are $P \in \mathcal{PTM}$ (called the 'prover'), and $V \in \mathcal{PPT}$ (called the 'verifier', whose final message is 'accept' or 'reject') such that

1. $\Pr[m_n^{PV(x)} = 1^n] \geq \frac{2}{3}$ for every $x \in L$.
2. $\Pr[m_n^{\bar{P}V(x)} = 1^n] \leq \frac{1}{3}$ for every $\bar{P} \in \mathcal{PTM}$ and $x \notin L$

where in both cases the probability space is over the coin tosses of the machines. The pair (P, V) is called an interactive proof for L.

Zero-knowledge proofs

Let (P, V) be an interactive proof for L. Intuitively, this proof is zero-knowledge if for every $\bar{V}$, a distribution indistinguishable from the transcript $Z^{P\bar{V}(x)}$ can be generated in $\mathcal{PPT}$ for every $x \in L$. Formally, $L \in \mathcal{ZK}$ if for all $\bar{V} \in \mathcal{PPT}$ there exists $S^{\bar{V}} \in \mathcal{PPT}$ (called the "simulator") such that $Z_L^{P\bar{V}} \overset{c}{=} S_L^{\bar{V}}$, where $Z_L^{P\bar{V}} = \{Z^{P\bar{V}(x)}\}_{x \in L}$ and $S_L^{\bar{V}} = \{S^{\bar{V}(x)}\}_{x \in L}$. (We note that in the definition of statistical zero-knowledge ([F-87]) the 'only' difference is that the last two distributions are required to be statistically close). Now, we wish to stress several preperties of zero-knowledge interactive proofs, which are going to be essential in our proof.

Remarks:

- For all $i \geq 0$ and all $c_i \in C_i^{PV}$ define *consistent*$(c_i) \overset{\triangle}{=} \{r | r \in (R^{PV}\#c_i)\}$, i.e. all the random strings of the verifier which are consistent with the prefix of the conversation c_i. We claim that for any $i \geq 0$ and any $c_i \in C_i^{PV}$ (i.e. prefix of the conversation between prover and honest verifier) the distribution $R^{PV}\#c_i$ is flat (i.e. uniformly distributed) over *consistent*(c_i). To see that this is so, notice, that for $i = 0$ this is clearly the case. For all $i > 0$ it follows by induction, as when prover "speaks", it does not have access to the verifiers random tape, so the set does not change, and when verifier "speaks" at round $i+1$ it restricts *consistent*(c_i) to a subset *consistent*$(c_i\#\hat{V}(x, (r \in \textit{consistent}(c_i)), c_i))$ which is again uniformly distributed, where $\hat{V}$ is a deterministic analog of V with its random bits fixed to be $r \in \textit{consistent}(c_i)$.

- WLOG, we can assume that in $S^{V(x)}$ the messages of honest verifier V are "legal", that is, for every even $i \geq 0$ of transcript $R^S\#C_{i+1}^S$ it is the case that $m_{i+1} = \hat{V}(x\#R^S\#C_i)$. Notice that this can be assumed without loss of generality as otherwise whenever this is not the case we can trivially distinguish outputs of the simulator from the ral conversations, where this *is* always the case.

An important corollary to universal extrapolation, is that for any $S \in \mathcal{PPT}$ there is another machine $S^{-1} \in \mathcal{PPT}$ that "inverts" the random bits of S used to produced a (partial) transcript of the simulated conversation. That is, for a fixed $x \in L$ and partial prefix of the conversation c_i, $S^{-1}(c_i)$ finds a random ω such that the prefix of $S(\omega) = r\#c_i$.

¿From now on, when clear from the context, we abuse the notation and for a fixed c_i, by $S^{-1}(c_i)$ denote either distribution on ω or the distribution of r's which $S(\omega)$ produces (by universal extrapolation). In particular, when we write $(S^{-1}(c_i), c_i)$, by $S^{-1}(c_i)$ we denote the distribution on r's. If for some c_i, $S^{-1}(c_i)$ fails to find ω, then it prints a special "reject" symbol.

Corollary 1

$$\not\exists 1WF \Longrightarrow \exists S^{-1} \in \mathcal{PPT} \text{ with } (S^{-1}(C_i^S), C_i^S) \overset{s}{=} (R^S, C_i^S)$$

An important superclass of $\mathcal{ZK}$, $\mathcal{ZKHV}$ (Zero-Knowledge for Honest Verifier), is when we demand from (P, V) only that the real transcript can be generated, i.e. $\exists S = S^V$ such that $Z_L^{PV} \overset{c}{=} S_L^V$. Clearly, $\mathcal{ZK} \subseteq \mathcal{ZKHV}$.

The usual difficulty in constructing zero-knowledge proofs is when $\bar{V} \neq V$, as we know V, but $\bar{V}$ can be arbitrary. However, our proof of the main theorem only uses the zero-knowledge property for the honest verifier, and thus in the main theorem one can replace $\mathcal{ZK}$ with the (possibly larger) $\mathcal{ZKHV}$.

However, one of the corollaries of our main theorem is:

Corollary 2

$$\mathcal{ZK} = \mathcal{ZKHV}$$

We can now re-state a stronger versions of our first theorem:

THEOREM 1 (strong form):
$\not\exists 1WF$ over $\Sigma = \{0,1\} \Longrightarrow (\mathcal{ZKHV} = \mathcal{BPP})$

It is possible that $\exists 1WF$ over $\Sigma = \{0,1\}$ but not over $\Sigma = \{1\}$. In this case it is possible to replace $\mathcal{BPP}$ in the main theorem with $\mathcal{AVBPP}$. However, this consequence will remain true even under a very liberal definition of zero-knowledge proofs, in which both the proof and the zero-knowledge conditions hold on average where inputs are chosen from a sampleable distribution. We defer a formal definition of $\mathcal{AVZK}$ (which is the standard one) to the final paper, and just mention the corresponding average case statement of the main theorem.

THEOREM 2 (strong form):

$$\not\exists 1WF \text{ over } \Sigma = \{1\} \Longrightarrow (\mathcal{AVZK} = \mathcal{AVBPP})$$

5 APPENDIX 2: Proof of The Main Theorem

5.1 Axioms Used in the Proof

While we continue to use $\mathcal{PPT}$ as our notion of efficient, this setup can be readily converted to agree about other complexity notions. All the axioms below are in fact theorems; the axioms *A1–A7* follow from the definitions, and axioms *B1, B2, B3* use in addition our assumption $\not\exists 1WF$.

Let D, E, F denote arbitrary ensembles. We have three relation symbols in ensembles: $=, \overset{s}{=}, \overset{c}{=}$ (where $=$ is standard equality).

A1: $(D = E) \Longrightarrow (D \overset{s}{=} E) \Longrightarrow (D \overset{c}{=} E)$

A2: $(D_1 \# D_2 \overset{c}{=} E_1 \# E_2) \Longrightarrow (D_1 \overset{c}{=} E_1)$.
More generally, if $D^x = D_1^x \# D_2^x \# \cdots \# D_k^x$, $E^x = E_1^x \# E_2^x \# \cdots \# E_k^x$, (with $k = O(|x|^c)$, $0 \le i = i(x) \le k$) then
$\{D_1^x \# \cdots \# D_i^x\}_{(x,i)} \overset{c}{=} \{E_i^x \# \cdots \# E_i^x\}_{(x,i)}$.

A3: All three relations are polynomially transitive; namely transitivity can be used for distributions indexed by $x \in \Sigma^*$ only $|x|^c$ times.

The next two axioms hold for every $M \in \mathcal{PTM}$ (recall that M's output length is polynomial in its input length):

A4: $(D \overset{s}{=} E) \Longrightarrow (D \# M(D) \overset{s}{=} E \# M(E))$

A5: $(D \# E \overset{s}{=} F \# M(F)) \Longrightarrow (D \# E \overset{s}{=} D \# M(D))$

The analogous two axioms for $\overset{c}{=}$ hold for every $M \in \mathcal{PPT}$:

A6: $(D \overset{c}{=} E) \Longrightarrow (D \# M(D)) \overset{c}{=} E \# M(E))$

A7: $(D \# E \overset{c}{=} F \# M(F)) \Longrightarrow (D \# E \overset{c}{=} D \# M(D))$

Finally, identify $N, M \in \mathcal{PPT}$ with the distributions they generate. For such samplable distributions we have (under our assumption $\not\exists 1WF$):

B1: $(M = M_1 \# M_2) \Longrightarrow \exists N \in \mathcal{PPT}$ such that $(M \overset{s}{=} M_1 \# N(M_1))$

B2: $(M \overset{c}{=} N) \Longrightarrow (M \overset{s}{=} N)$

B3: Let $D\#E$ be an arbitrary distribution (we stress that we do not assume that $D\#E$ is sampleable). For any e, let $D\#e$ be conditional distribution on D given that the second component of $D\#E$ is fixed to e (if $e \notin E$, we say that $D\#e$ is *empty*), and let $T_e \overset{\triangle}{=} \{d |\ d\#e$ is in the support set of $D\#e\}$. Let $D^N\#E^N$ be a distribution which is sampleable according to some $N \in \mathcal{PPT}$. For any e let $D^N\#e$ denote conditional distribution on D^N given that the second component of $D^N\#E^N$ is fixed to e (we say that $D^N\#e$ is *empty* if $e \notin E^N$) and define $T_e^N \overset{\triangle}{=} \{d |\ d\#e$ is in the support set of $D^N\#e\}$. Then, the following imlication holds:

$$\left.\begin{array}{ll} \text{(a)} & D\#E \overset{c}{=} D^N\#E^N \\ & \quad\textit{and} \\ \text{(b)} & \text{For any } e,\ T_e^N \subseteq T_e, \text{ and} \\ & \quad D\#e \text{ is uniform on } T_e \end{array}\right\} \Longrightarrow D\#E \overset{s}{=} N^{-1}(E)\#E$$

5.2 The proof of our main result

We first state our main theorem again:

$$\not\exists 1WF, L \in \mathcal{ZKHV} \Longrightarrow L \in \mathcal{BPP}$$

Let $L \in \mathcal{ZKHV}$, let (P, V) be a zero-knowledge (for honest verifier) proof system for $L \subseteq \{0,1\}^*$, and S the "simulation" associated with V (both $S, V \in \mathcal{PPT}$). We know that the transcript of (P, V) on input $x \in \{0,1\}^*$ has the structure $Z^{PV(x)} = R^{PV(x)}\#m_1^{PV(x)}\# \cdots \#m_n^{PV(x)} = R^{PV(x)}\#C_n^{PV(x)}$ (we remove x from the output for convenience).

We assume without loss of generality that the output of S on x, denoted here $Z^{S(x)}$ has the same structure, i.e. $Z^{S(x)} = R^{S(x)}\#m_1^{S(x)}\# \cdots \#m_n^{S(x)}$, and furthermore that $R^{S(x)}$ is uniform over $\{0,1\}^n$. Again we define $C_0^{S(x)} = \emptyset$, and $C_{i+1}^{S(x)} = C_i^{S(x)}\#m_{i+1}^{S(x)}$, so $Z^{S(x)} = R^{S(x)}\#C_n^{S(x)}$.

We mention again the three properties of the proof system (note we use only $L \in \mathcal{ZKHV}$):

1. for every $x \in L$, $\Pr[m_n^{PV(x)} = 1^n] \geq \frac{2}{3}$
2. for every $\bar{P} \in \mathcal{PTM}$, every $x \notin L$, $\Pr[m_n^{PV(x)} = 1^n] \leq \frac{1}{3}$
3. $Z_L^{PV} \overset{c}{=} Z_L^S$ (the two ensembles, indexed by elements $x \in L$, are computationally indistinguishable)

Our task is to present a $\mathcal{PPT}$ algorithm for recognizing L. This will be the same algorithm used in [Ost-91] in the case that L had statistical zero-knowledge proof (i.e. $\overset{c}{=}$ in condition (3) was replaced by $\overset{s}{=}$).

Let $\bar{S}$ be a machine just like S, but whose output on x is only $C_n^{S(x)}$ (rather than $R^{S(x)}\#C_n^{S(x)}$). Let P^* be a machine that "extrapolates" $\bar{S}$, i.e. on input $x, C_i^{S(x)}$ produces $P^*(C_i^{S(x)})$ (x is implicit in the input) such that the ensembles satisfy $C_{i+1}^S \overset{s}{=} C_i^S\#P^*(C_i^{S(x)})$.

By axiom *B1*, as $\bar{S} \in \mathcal{PPT}$, also $P^* \in \mathcal{PPT}$ (see comment after definition of universal extrapolation). Now define the algorithm $A \in \mathcal{PPT}$ which on input x generates the transcript $Z^{A(x)} = Z^{P^*V(x)}$. Our main lemma states that on $x \in L$ this distribution is indistinguishable from the real one.

Main Lemma: $Z_L^A \overset{c}{=} Z_L^{PV}$

The $\mathcal{BPP}$ algorithm B for L will simply compute $Z^{A(x)}$ and accept x iff $m_n^{A(x)} = 1^n$. The completeness condition (2) guarantees that B will reject every $x \notin L$ with probability $\geq \frac{2}{3}$, as P^* is a special case of $\bar{P}$. The fact that B will accept each $x \in L$ with probability $\geq \frac{3}{5}$ (say) follows immediately from the the main lemma, as the gap between $\frac{3}{5}$ and $\frac{2}{3}$ is easily distinguishable in $\mathcal{PPT}$.

5.3 Proof of the Main Lemma

We will prove by induction (see technical remark below) on i, ($i = 0, 1, \cdots n = n(x)$) that the ensembles below which are indexed by $x \in L$ satisfy:

(1_i) $R^A\#C_i^A \overset{s}{=} R^S\#C_i^S$, and

(2_i) $R^A\#C_i^A \overset{c}{=} R^{PV}\#C_i^{PV}$

Technical Remark: We need to explain the formal meaning of using induction in the context of ensembles, where n is not fixed but depends on the length of input x. Our notation shortcuts the need to do induction for *each* large enough x (and n). There the "errors" implicit in the $\overset{s}{=}$ and $\overset{c}{=}$ are explicitly bounded for every $i \leq n$, during the induction. Afterwards, these bounds are combined for all x to derive $\overset{s}{=}$ or $\overset{c}{=}$. The important thing to note is that we use the transitivity of $\overset{s}{=}$ and $\overset{c}{=}$ only $O(n_x)$ times for any x, which takes care of bounding the errors.

Lemma 1 (Base case $i = 0$): (1_0), (2_0) hold.

Proof: By definition $R^A = R^S = R^{PV}$ □

Lemma 2 For every i,

(3_i) $R^S\#C_i^S \overset{c}{=} R^{PV}\#C_i^{PV}$

Proof: Follows from the zero-knowledge property and axiom *A3*. □

Lemma 3 For every i, (1_i) and (2_i) are equivalent (and so it will suffice to prove only one of them).

Proof: As (3_i) holds, we have:
$(3_i),(1_i) \Longrightarrow (2_i)$ by transitivity (axiom *A3*)
$(3_i),(2_i) \Longrightarrow R^A\#C_i^A \overset{c}{=} R^S\#C_i^S$, but both distributions are sampleable, and by axiom *B2* $(R^A\#C_i^A \overset{c}{=} R^S\#C_i^S) \Longrightarrow (R^A\#C_i^A \overset{s}{=} R^S\#C_i^S) = (1_i)$. □

From now on we assume that $(1_i),(2_i),(3_i)$ hold and we use them to prove either (1_{i+1}) or (2_{i+1}). There are two cases, CASE V and CASE P, depending on whether the $i+1$ message is sent by the verifier or the prover, respectively.

CASE V: i is even, so $i+1$ is a "verifier's message".

Lemma 4 $(2_i) \Longrightarrow (2_{i+1})$

Proof: $R^A\#C_{i+1}^A = R^A\#C_i^A\#\hat{V}(R^A\#C_i^A) \overset{c}{=} R^{PV}\#C_i^{PV}\#\hat{V}(R^{PV}\#C_i^{PV}) = R^{PV}\#C_{i+1}^{PV}$, where the $\overset{c}{=}$ step follows from (2_i) and axiom *A5*, and the fact that $V \in \mathcal{PPT}$ (recall that $\hat{V}$ is the deterministic version of V). □

CASE P: i is odd, so $i+1$ is a "prover's message". First, we state the important properties of the machines P^* and S^{-1} (both in $\mathcal{PPT}$), which hold for all i:

Lemma 5

(4_i) $C_{i+1}^S \overset{s}{=} C_i^S\#P^*(C_i^S)$

(5_i) $R_i^{PV}\#C_i^{PV} \overset{s}{=} S^{-1}(C_i^{PV})\#C_i^{PV}$

Proof: Property (4_i) follows from the definition of P^*. To show (5_i) we show that conditions of axiom $B3$ apply. In particular, we let $D\#E$ denote $R_i^{PV}\#C_i^{PV}$, we let N denote S and we let $D^N\#E^N$ denote $R^S\#C_i^S$. Then, condition (a) holds by (3_i). Condition (b) follows from our remarks after the defitnion of Zero-knoweldge proofs. □

The purpose of the next lemma is to prove that the $i+1$ message of the simulation m_{i+1}^s cannot depend on the "random tape" R^S. Intuitively, it is so since when it is a prover's message m_{i+1}^{PV} is independent of R^{PV}. However, this statement is *false* for every $\mathcal{ZK}$ proof known, and clearly our proof relies heavily on the assumption $\not\exists 1WF$.

Lemma 6 $R^S\#C_{i+1}^S \overset{s}{=} S^{-1}(C_i^S)\#C_{i+1}^S$

Proof: ¿From (5_i), $C_{i+1}^{PV} = C_i^{PV}\#P(C_i^{PV})$ and axiom *A4*, we have:

(6_{i+1}) $S^{-1}(C_i^{PV})\#C_{i+1}^{PV} \overset{s}{=} R^{PV}\#C_{i+1}^{PV}$

From $(3_{i+1}),(6_{i+1})$ and axiom *A7* imply $S^{-1}(C_i^S)\#C_{i+1}^S \overset{c}{=} R^S\#C_{i+1}^S$ and since both distributions are sampleable, $\overset{c}{=}$ can be replaced by $\overset{s}{=}$ to obtain the lemma. □

By the previous lemma and axiom *A2*,

Corollary 3 $R^S\#C_i^S \overset{s}{=} S^{-1}(C_i^S)\#C_i^S$

Lemma 7 (1_{i+1}) holds, i.e. $R^S\#C_{i+1}^S \overset{s}{=} R^A\#C_{i+1}^A$

Proof: By (1_i) and axiom *A4*, $C_{i+1}^S \overset{s}{=} C_i^S\#P^*(C_i^S) \overset{s}{=} C_i^A\#P^*(C_i^A) \overset{s}{=} C_{i+1}^A$. Hence by previous lemma and axiom *A4*, $R^S\#C_{i+1}^S \overset{s}{=} S^{-1}(C_i^A)\#C_i^A\#P^*(C_i^A)$. Now by using corollary 6 and axiom *A4*, we have $S^{-1}(C_i^A)\#C_i^A\#P^*(C_i^A) \overset{s}{=} R^A\#C_i^A\#P^*(C_i^A) \overset{s}{=} R^A\#C_{i+1}^A$ and we are done. □

5.4 Proof of B3

First, let us restate B3: Let $D\#E$ be an arbitrary distribution (we stress that we do not assume that $D\#E$ is sampleable). For any e, let $D\#e$ be conditional distribution on D given that the second component of $D\#E$ is fixed to e (if $e \not\in E$, we say that $D\#e$ is *empty*), and let $T_e \stackrel{\triangle}{=} \{d|\ d\#e \text{ is in the support set of } D\#e\}$. Let $D^N\#E^N$ be a distribution which is sampleable according to some $N \in \mathcal{PPT}$. For any e let $D^N\#e$ denote conditional distribution on D^N given that the second component of $D^N\#E^N$ is fixed to e (we say that $D^N\#e$ is *empty* if $e \not\in E^N$) and define $T_e^N \stackrel{\triangle}{=} \{d|\ d\#e \text{ is in the support set of } D^N\#e\}$. Then, the following imlication holds:

$$\left.\begin{array}{ll} \textit{(a)} & D\#E \stackrel{c}{=} D^N\#E^N \\ & \qquad \textit{and} \\ \textit{(b)} & \text{For any } e,\ T_e^N \subseteq T_e, \text{ and} \\ & \quad D\#e \text{ is uniform on } T_e \end{array}\right\} \Longrightarrow D\#E \stackrel{s}{=} N^{-1}(E)\#E$$

The proof is by contradiction. That is, we assume *(b)* and $D\#E \stackrel{s}{\neq} N^{-1}(E)\#E$ hold, and show that *(a)* does not hold. By ω let us denote the coin flips of randomized machine N, that is on input randomly chosen ω, $N(\omega)$ *deterministically* outputs $d\#e \in D^N\#E^N$. Fix some $e \in E$. For a fixed d, let

$$W_e^d \stackrel{\triangle}{=} \{\omega|\ \text{such that } N(\omega) = d\#e\}$$

We note that it is possible for W_e^d to be an empty set. Let $W_e \stackrel{\triangle}{=} \bigcup_d W_e^d$. Let $U(e)$ denote a uniform distribution on T_e, (note that this is exactly the distribution of d in $D\#e$ according to *(b)*, and that this distribution *may not be* sampleable) and let $N^{-1}(e)$ be a (sampleable) distribution on T_e^N which is computed according to a randomly chosen ω so that $N(\omega) = d\#e$. Recall that by *(b)*, $T_e^N \subseteq T_e$ and hence $U(e)$ is also uniform on T_e^N. Let $r(e)$ be a probability of e according to E.

Lemma 8 If $D\#E \stackrel{s}{\neq} N^{-1}(E)\#E$ then, there $\exists A \subseteq \{E\}$ and $\exists \alpha$ of size $1/poly(\omega)$ such that:

$$\sum_{e\in A} r(e) \geq \alpha \qquad \textit{and} \qquad \forall e \in A\ ||U(e) - N^{-1}(e)||_1 \geq \alpha$$

Proof: trivial. □

For a fixed e, let p_d denote probability of d according to $N^{-1}(e)$ and q_d denote probability of d according to $U(e)$ and $t_e \stackrel{\triangle}{=} |T_e|$.

Lemma 9 If $\not\exists 1WF$ then there exists $M \in \mathcal{PPT}$ such that on input $d\#e$ and $0 < \beta < 1$, M outputs (in time polynomial in $|d\#e|$ and $1/\beta$) p'_d and t'_e such that:

$$\textit{Prob}\,((1-\beta)\cdot p_d < p'_d < (1+\beta)\cdot p_d) \geq 1 - O(|x|^{-|M|})$$

$$\textit{Prob}\,((1-\beta)\cdot t_e < t'_e < (1+\beta)\cdot t_e) \geq 1 - O(|x|^{-|M|})$$

where probability is taken over coin-toses of M.

Proof: By universal aproximation (theorem 4) we can approximate $|W_e^d|$ and $|W_e|$. We note that $p_d = \frac{|W_e^d|}{|W_e|}$. Thus, again by theorem 4 we can get an arbitrary close estimate to p_d. In order to estimate t_e let us define a random variable $g \overset{\triangle}{=} \frac{1}{p_d}$. Then, the expected value $E(g) = \sum_{d \in T_e^N} p_d \cdot g = |T_e^N|$. Thus, by universal approximation we can estimate $|T_e^N|$ as well. We claim that our approximation of $|T_e^N|$ is also an approximation to t_e, as otherwise with probablity greater then *poly*(β) there will be $d\#e \in D\#E$ for which $|W_e^d| = 0$, which would give us an efficient distinguisher since for $d\#e \in D^N\#E^N$, $|W_e^d| \geq 1$. □

Recall that we are proving $B3$ by contradiction. Thus, we assume *(b)* and $D\#E \overset{s}{\not\equiv} N^{-1}(E)\#E$ hold, and show how to distinguish $D\#E$ from $D^N\#E^N$ assuming that there are no one-way functions. Our distinguisher operates as follows:

```
1. On input $d\#e$ and $\alpha$ compute $p'_d$ and $t'_e$ within $(1 \pm \frac{\alpha^4}{8})$ of $p_d$ and $t_e$.
2. compute $b = p'_d \cdot t'_e$.
3. IF $(b \geq 1)$
   THEN output $1$
   ELSE output a coin flip which is biased toward $1$ with probability $b$.
```

How good is our distinguisher? We wish to measure the difference in the probability of head when $d\#e$ comes from $D^N\#E^N$ distriburion and when $d\#e$ comes from $D\#E$ distribution. First, let us ignore the fact that we are dealing with approximations to p_d and t_e and calculate how good a distinguisher we would get if we could calculate the values of p_d and t_e exactly. Thus, let us express this differense as Z defined as follows:

$$Z \overset{\triangle}{=} \sum_d \left(p_d \cdot \min\left(p_d t_e, 1\right) - q_d \cdot \min\left(p_d t_e, 1\right)\right)$$

We now wish to bound Z:

Lemma 10

$$||N^{-1}(e) - U(e)||_1 = 2\alpha \quad \Longrightarrow \quad Z \geq \alpha^2$$

Proof: Note that

$$Z = \sum_{d \in T_e} \left(\left(p_d - q_d\right) \min\left(p_d t_e, 1\right)\right)$$

Split T_e into two sets $T_e = T_e^0 \bigcup T_e^1$ such that for all $d \in T_e^0$, $p_d t_e > 1$ and for $d \in T_e^1$, $p_d t_e \leq 1$. Then Z becomes

$$Z = \sum_{d \in T_e^0} \left(\left(p_d - q_d\right) \cdot 1\right) + \sum_{d \in T_e^1} \left(\left(p_d - q_d\right) \cdot p_d t_e\right)$$

Since $||N^{-1}(e) - U(e)||_1 = 2\alpha$ the first summation is clearly α. In the second summation, by our defintion of T_e^1, $p_d t_e \leq 1$, hence,

$$Z = \alpha + \sum_{d \in T_e^1} \left(\left(p_d - q_d\right) \cdot p_d t_e\right)$$

In order to prove our lemma, we wish to show that that second summation is greater then $-\alpha + \alpha^2$. Towards this end, negating and switching sign, we are interested in bounding from above:

$$\sum_{d \in T_e^1} (q_d p_d - p_d^2) \cdot t_e$$

Notice that the above sum in maxmimized when p_d is uniform. Moreover, we know that $\sum_{d \in T_e^1} p_d = \alpha$ and hence $p_d = \frac{\alpha}{|T_e^1|}$. Moreover, since $q_d = \frac{1}{t_e}$, the above sum becomes:

$$\sum_{d \in T_e^1} (q_d p_d - p_d^2) \cdot t_e = \sum_{d \in T_e^1} \left(\frac{1}{t_e} \cdot \frac{\alpha}{|T_e^1|} - \frac{\alpha^2}{|T_e^1|^2} \right) \cdot t_e$$

Since $|T_e^1| \leq t_e$ setting $|T_e^1| = t_e$ and summing gives us an $\alpha - \alpha^2$ bound. Thus,
$Z \geq \alpha - (\alpha - \alpha^2) = \alpha^2$ □

By lemma 8, since $D \# E \overset{s}{\not\approx} N^{-1}(E) \# E$ there $\exists A \subseteq \{E\}$ and $\exists \alpha$ of size $1/poly(\omega)$ such that $\sum_{e \in A} r(e) \geq \alpha$ and $\forall e \in A$ $||U(e) - N^{-1}(e)||_1 \geq \alpha$ On this subset A, our idealized distigisher distinguishes within α^2. However, for $e \notin A$ our distinguisher can either not have any bias (if for $e \notin A$, $N^{-1}(e)$ is uniform) or can have a bias *in the same direction* as on A! Thus, in the worst case, our idealized algorithm distinguishes within α^3. However, since we are dealing with approximations p'_d and t'_e, by setting $\beta = \frac{\alpha^4}{8}$ from lemma 9, we can distinguish at least within $\frac{\alpha^3}{2}$ and we are done with $B3$. □