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

(EXTENDED ABSTRACT*)

Rafail Ostrovsky[†] Avi Wigderson[‡]

TR-93-073

Novemeber, 1993

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 analogous result under the assumption that *uniform* one-way functions do not exist.

Thus, very loosely speaking, zero–knowledge is either *useless* (exists only for "easy" languages), or *universal* (exists for every provable language).

[*] Preliminary Version Appeared in Prooceedings of the second Israel Symposium on Theory of Computing and Systems (ISTCS93) Netanya, Israel, June 7th-9th, 1993.

[†] University of California at Berkeley Computer Science Division, and International Computer Science Institute at Berkeley. E-mail: rafail@melody.berkeley.edu. Supported by NSF postdoctoral fellowship and ICSI.

[‡] Princeton University and Hebrew University. This research was partially supported by the Wolfson Research Awards, administered by the Israel Academy of Sciences and Humanities.

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. 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, NY-89, 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].

A central notion to modern cryptography is the notion of zero-knowledge proof system, pioneered by [GMR-85]. The subject of this paper is the relationship between one-way functions and zero-knowledge proofs. While one-way functions were shown to be sufficient for zero-knowledge proofs [GMR-85, GMW-86, IY-87, ILL-89, Ha-90, N-89], the question of necessity was open.

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 $f_k(\cdot)$, 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 some proof system). That is, by auxiliary input one-way function f we denote a family of always 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. However, auxiliary-input one-way functions may exist even if uniform one-way functions do not (for example, the set of x for which $f_x(\cdot)$ is one-way may not be sampleable by any polynomial-time algorithm.)

Interactive proofs, their knowledge complexity, and the associated complexity classes $\mathcal{IP}$ and $\mathcal{ZK}$ were introduced and formalized in [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: the verifier could have generated by itself a conversation, indistinguishable from the one it had with the prover. Thus it is clear that trivial languages (in $\mathcal{BPP}$) 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 $\mathcal{NP} \subseteq \mathcal{ZK}$ (in [GMW-86] they used one-way permutations, from [ILL-89, Ha-90, N-89] the result can be restated with arbitrary one-way functions.) This results made zero-knowledge a central tool in secure protocol design and fault-tolerant distributed computing [GMW-86, Yao-86, GMW-87]. Extending this result, [IY-87] (see also [B+ 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 (follows from references above) *(Informal statement):* If uniform one-way functions exist then $\mathcal{ZK} = \mathcal{IP}$ $(= \mathcal{PSPACE})$.

We give two theorems which supply a converse to the above theorem. Both show essentially that if one–way functions do not exist, then $\mathcal{ZK}$ is as small as possible, namely zero–knowledge proofs exist only for "trivial" languages. Put differently, one-way functions are essential for "non–trivial" zero–knowledge.

The theorems differ in the meaning of "trivial" and the type of one–way functions we assume do not exist. Under the strong assumption that even auxiliary–input one–way functions do not exist we get the strongest implication.

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

Under the weaker assumption that (the commonly used) uniform one–way functions do not exist, we get an average–case result. Specifically, our notion of "trivial" extends to contain all languages that have an efficient probabilistic algorithm on average, when the input is generated by some efficiently sampleable distribution. Call this complexity class $\mathcal{AVBPP}$. Then

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? We show that it is possible to obtain an average case complexity result assuming only the nonexistence of uniform one-way functions. On the other hand, 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. We consider the case when all definitions are made completely non-uniformly in the full version of the paper.

- An important point that has to be addressed in such theorems is for which input length the assumption and conclusion hold. In the sufficient condition, it is natural to assume that one-way functions exist for *every* input length, while in the necessary condition it is natural to assume that they do not exist for *every* input length. These are not complementary conditions, but like most other results of this type, both theorems have analogs showing that zero–knowledge is possible exactly for those (sets of) input lengths for which one-way functions exist.

- 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 Relation to previous work and techniques

Necessity of one-way functions for various restricted classes of zero-knowledge proofs was previously considered in [FS-89, D-89, Ost-91]. The main such result is by [Ost-91]. He proves it for the very special case of *statistical* zero–knowledge proofs [GMR-85]. In these proofs, the simulator is required to produce a distribution of prover-verifier conversations that is statistically close to the real distribution. Note that statistical zero-knowledge proofs are known only for a handful of languages, such as graph isomorphism, graph non-isomorphism, and quadratic residuosity. In fact, only languages in $AM\bigcap co\text{-}AM$ can have such proofs [AH-87, F-87]. Nevertheless, the ideas developed in [Ost-91] are utilized in an essential way in our construction.

Our proofs also utilize many of the techniques (and their consequences) that were developed by [ILL-89, Ha-90] for showing the equivalence of one-way functions and pseudo-random generators. In particular, we use the notions of distributional one-way functions [ILu-89], efficient universal extrapolation [ILe-90], and the note of [G-89] on computational versus statistical indistinguishability.

The main technical contribution of our paper is an extension of this last mentioned result of [G-89]. In [G-89] it is shown that if two efficiently sampleable distributions are computationally indistinguishable but statistically different, then this implies the existence of a one-way function. We show that the same holds in *some* cases in which these distributions are not sampleable! These cases contain the interesting situation that one of the distributions is the transcript of a conversation in a zero–knowledge proof. This result is the key to our theorems, and we expect both it and its proof to be useful elsewhere.

From this result we conclude that (assuming there are no one–way functions), the simulator in every zero–knowledge proof has the following property: the messages of the prover in the conversations it generates are statistically independent from the random tape it assigns to the verifier. This property evidently holds in the real conversations, and so automatically holds for simulators in statistical zero–knowledge proofs. It is not clear why such statistical independence hold in computational zero–knowledge proofs. Indeed, this property fails for every computational zero–knowledge proof known. However, all these proofs use one–way functions! Thus, loosely speaking, what we show is that this is the only possible approach.

The developments in computational cryptography of the last few years have brought this field 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.

1.3 Corollaries

Caution: This subsection is written informally. The technical statements of the corollaries below is quite cumbersome, due to the fact that our notions of existence and non–existence of one–way functions are not complementary. We defer formal statements and further applications of our results to the final paper.

Under the assumption that one-way functions exist, many properties of (computational) zero-knowledge proofs were established. 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 holds 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. Moreover, as these properties trivially hold for languages in $\mathcal{AVBPP}$, these theorems become unconditional statements about the class $\mathcal{ZK}$.

For example, it was shown ([GS-86, GMS-87, LFKN-90, S-90]) 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 $\mathcal{PSPACE}$; 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 assumption.

Another important corollary follows from the fact that both theorems 1 and 2 use only the zero knowledge property with respect to the "honest" verifier. Thus, we can conclude that the class of languages which has a $\mathcal{ZK}$ proof *for honest verifier only* is **equivalent** to $\mathcal{ZK}$! This reduces the task of a zero-knowledge proof-system designer to establish the zero-knowledge property only for the honest verifier, a task that that is typically much easier.

1.4 Organization of the paper

The next section is devoted to the description of the important definitions, and of the basic results we shall use. The last subsection of section 2 formally states our two main theorems. In section 3, we give a set of theorems concerning statistical and computational indistinguishability and provide the proof of theorem $B3$ which is central in our construction. In section 4 we give an intuitive description of the proof of our main theorems, trying to point to the subtleties. Finally, in section 5 we give the proof of our main result.

2 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.

2.1 Probabilistic and Efficient Turing machines

2.1.1 The class $\mathcal{PTM}$ of Probabilistic Turing Machines

Informally, $\mathcal{PTM}$ is the class of all probabilistic Turing machines, whose output length is polynomial in the input length. For a probabilistic Turing machine M, $x \in \Sigma^*$, $M(x)$ denotes the output distribution of M on input x. Let $|x|$ denote the length of the string x, and $|M|$ denotes the length of the description of the Turing machine M. Every machine $M \in \mathcal{PTM}$ satisfies $|M(x)| \leq O(|x|^{|M|})$.

2.1.2 The class $\mathcal{PPT}$ of Probabilistic Polynomial Time Turing Machines

$\mathcal{PPT}$ is the class of all probabilistic Turing machines M that on input x halt in polynomial time (say $|x|^{3|M|}$). 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 output distribution of M where the input x is chosen at random from D. We assume w.l.o.g that M uses exactly $|x|^{|M|}$ random bits R. It will be convenient to explicitly define $\hat{M}$ to be the deterministic "analog" of M, with $\hat{M}(R\sharp x) = M(x)$, where R is chosen uniformly from $\{0,1\}^{|x|^{|M|}}$ and $\sharp$ is a special delimiter used for string concatenation.

The languages recognized by $\mathcal{PPT}$ machines form the class $\mathcal{BPP}$. The languages recognized by $\mathcal{PPT}$ machines on average, over some sampleable distribution form the class $\mathcal{AVBPP}$.

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

Let $D_{\mathcal{N}}$ be a sampleable ensemble on $\{0,1\}^*$ with D^n distributed over $\{0,1\}^n$. The pair $(L, D) \in \mathcal{AVBPP}$ if $\exists M \in \mathcal{PPT}$ s.t. $\forall n \in \mathcal{N}$, $\Pr[M(x) = L(x)] \geq \frac{3}{5}$, where the probability is over $x \in D^n$ and coin tosses of M.

2.2 One-way Functions

A one-way function $f_x(\cdot)$ can be best described using a game between two $\mathcal{PPT}$ machines M and N on common input x, given to both players. First, M computes the distribution $M(x) = (y, z)$, where $y = f(z)$ and gives y to N. Then, 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. We explicitly specify whether the common input x is binary or unary string, as these two possibilities determine the two notions of one–way functions we use in this paper.

If x is *unary*, this is the standard notion of a *uniform* one-way function. Here x is sometimes called the security parameter. If x is *binary*, this captures a notion of a one-way function with *auxiliary input*, given to both players.

We proceed more formally. Let Σ be an input alphabet (either $\{0\}$ in the unary case or $\{0,1\}$ in the binary case. Let $M \in \mathcal{PTM}$ be such that for every $x \in \Sigma$, $M(x) = M_1(x)\sharp M_2(x)$, with $M_1(x) = F(x, M_2(x))$ for some function F computed in deterministic polynomial time.

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

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

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 the possible assumptions:

- ($\exists S1WF$) There exist strong one-way functions $\stackrel{\triangle}{=} \exists M \in \mathcal{PPT}$ such that $\forall N \in \mathcal{PPT}$ $|\Sigma^* - H_N(M)| < \infty$.
- ($\exists 1WF$) There exist one-way functions $\stackrel{\triangle}{=} \exists M \in \mathcal{PPT}$ such that $\forall N \in \mathcal{PPT}$, $|H_N(M)| = \infty$.
- ($\not\exists 1WF$) There are no one-way functions $\stackrel{\triangle}{=} \forall M \in \mathcal{PPT}, \exists N \in \mathcal{PPT}, |H_N(M)| < \infty$.

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 F, M and N are families of circuits) can be given and our results can be transformed into non-uniform model as well.

2.3 Ensembles of Probability Distributions

For $T \subseteq \Sigma^*$, $D_T = \{D^x\}_{x \in T}$ will denote the collection of probability distributions (or random variables) on strings over $\{0,1,\sharp\}$, indexed by elements of T. Of importance are ensembles generated by probabilistic Turing machines. For $M \in \mathcal{PTM}$, $T \subseteq \Sigma^*$ denotes all $M_T = \{M(x)\}_{x \in T}$.

By the notation $M(D)$ with $M \in \mathcal{PTM}$ and D and ensemble we mean the ensemble whose elements are the distributions $M(x\sharp D^x)$, i.e. the distribution generated by M on input x and a random element of D^x. Denote by U^n the uniform distribution of $\{0,1\}^n$, and by $U_{\mathcal{N}}$ the ensemble of uniform distributions for every $n \in \mathcal{N}$.

2.4 Negligible Fractions and Statistical Closeness of Distributions ($\stackrel{s}{=}$)

A negligible probability is a fraction smaller than the inverse of any polynomial (in the input/index length). Other fractions are non-negligible. For two probability ensembles (of random variables) on strings D, E, we say that $D \stackrel{s}{=} E$, if for every constant c and for every index x, $||D^x - E^x||_1 \leq |x|^{-c}$ (where $|| \cdot ||_1$ stands for the L_1 norm.

Note that we can apply transitivity to the relation $\stackrel{s}{=}$ polynomially (in $|x|$) many times. Also, for any $M \in \mathcal{PTM}$, if $D \stackrel{s}{=} E$ then $(D, M(D)) \stackrel{s}{=} (E, M(E))$.

2.5 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 $x \in \Sigma^*$, $M(x) \stackrel{s}{=} (y, N(x, y))$. A related notion is *universal approximation*, where the machine N is requested to approximate (on input (x, y)) the number of extensions z for which $M(x) = (y, z)$.

Formally, consider machines M with $M(x) = M_1(x)\sharp M_2(x)$. Denote the number of extensions of $M_1(x)$ by $|M_2|_{x,y} \stackrel{\triangle}{=} |\{M_2(x) : M_1(x) = y\}|$. In both notions the "inverting" machine N is given an additional accuracy input α in unary.

(UE) *Universal Extrapolation* $\stackrel{\triangle}{=}$ For every $M \in \mathcal{PPT}$ there exists $N \in \mathcal{PPT}$ (which runs in time polynomial in $|x|$ and the accuracy parameter $1/\alpha$) satisfying for all $x \in \Sigma^*$,

$$||M_1(x)\sharp M_2(x) \; - \; M_1(x)\sharp N(x, M_1(x), \alpha)||_1 \leq \alpha$$

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

$$\mathit{Prob}\left[(1-\alpha)|M_2|_{x,M_1(x)} < N(x, M_1(x), \alpha) < (1+\alpha)|M_2|_{x,M_1(x)}\right] \geq 1 - \alpha$$

Here are the first two major consequences of our assumption $\not\exists 1WF$; both follow from [JVV-86, ILL-89, Ha-90, ILu-89, ILe-90, BP-92] though they show it for uniform one-way functions (i.e. they consider $\Sigma = \{0\}$). The extension to auxiliary–input one–way functions is straightforward.

Theorem 3 [Universal Extrapolation] (follows from references above)*:*

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

Theorem 4 [Universal Approximation]*)* (follows from references above)*:*

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

Remark: These are typical results where the proofs show in fact that the assumption and conclusion coexist for the same sets of input lengths (as mentioned for UE in [ILu-89]).

2.6 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) $N(D) \overset{s}{=} N(E)$. Formally, $D_T \overset{c}{=} E_T$ 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 $\overset{s}{=}$, we can apply transitivity to $\overset{c}{=}$ polynomially many times. Also, if $D \overset{c}{=} E$ and $M \in PPT$, then $(D, M(D)) \overset{c}{=} (E, M(E))$.

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 $(D \overset{s}{=} S) \Longrightarrow (D \overset{c}{=} E)$. Assuming $\not\exists 1WF$, the converse is true for all pairs D, E which can be generated efficiently!

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

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

One cannot replace the sampleable ensembles M, N in this theorem by arbitrary ensembles, as shown by [GK-89]).

2.7 Interactive Machines and Structured Conversations

A conversation (of n rounds, on common input x) between machines $P, V \in \mathcal{PTM}$ is the sequences $C = C^{PV} = C^{PV(x)} = (m_1 \sharp m_2 \sharp \cdots \sharp m_n)$ (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 $(R^{PV} \sharp C^{PV})$. 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 $Z^{MN(x)} = x \sharp R \sharp m_1 \sharp m_2 \sharp \cdots \sharp m_n$, 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).

- $m_{i+1} = \begin{cases} \hat{N}(x\sharp R\sharp C_i) & \text{for even } i \geq 0 \\ M(x\sharp C_i) & \text{for odd } i \geq 0 \end{cases}$ and

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

2.8 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. $\Pr[m_n^{PV(x)} = ``accept''] \geq \frac{2}{3}$ for every $x \in L$.
2. $\Pr[m_n^{\bar{P}V(x)} = ``accept''] \leq \frac{1}{3}$ 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.

The main result about interactive proofs is:

Theorem 6 ([LFKN-90, S-90]:)

$$\mathcal{IP} = \mathcal{PSPACE}$$

2.9 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 $Z^{P\bar{V}(x)}$ (which is defined $R^{P\bar{V}}\sharp C^{P\bar{V}}$) 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 $Z_L^{P\bar{V}} \stackrel{c}{=} S_L^{\bar{V}}$ holds[1], where $Z_L^{P\bar{V}} = \{Z^{P\bar{V}(x)}\}_{x\in L}$ and $S_L^{\bar{V}} = \{S^{\bar{V}(x)}\}_{x\in L}$.

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 $Z_L^{PV} \stackrel{c}{=} S_L^V$. (We put no restriction on conversations generated by P and some arbitrary $\bar{V}$.) From now on we regard S as the simulator S^V of the "honest" prover in a proof system (P, V).

Given $D_{\mathcal{N}}$, a sampleable distribution from which inputs of each input length are chosen, one can define the average case analogs $\mathcal{AVZK}$ and $\mathcal{AVZKHV}$. We simply require that (P, V) is a proof system for L, and that zero–knowledge holds on average. Namely, that $Z^{PV(D|L)} = Z^{S(D|L)}$, where $(D|L)$ is the ensemble of distributions where for each n, x is chosen from $\{0,1\}^n$ according to D^n conditional on $x \in L$ (this ensemble may not be sampleable efficiently).

Two important structural properties regarding the transcripts $Z^{PV(x)}$ and $Z^{S(x)}$ for $x \in L$ follow from the definitions (we shall omit the x when clear). Fix a round number i

[1] We note that in the definition of statistical zero-knowledge the 'only' difference is that the last two ensembles are required to be statistically close.

and a perfix of the conversation c_i of C_i^{PV} (the facts below will hold for every such choice). Let $T_{c_i}^{PV}$ (resp. $T_{c_i}^{S}$) be the set of all ("consistent") random tapes r of the verifier V for which $r\sharp c_i$ has positive probability in the conditional distribution $R^{PV}\sharp c_i$ (resp. $R^{S}\sharp c_i$).

Fact 1: The conditional distribution $R^{PV}\sharp c_i$ is uniform on $T^{PV}(c_i)$ (such distributions are called *flat*).

This easily follows by induction on i, and the fact that the prover P cannot access R.

Fact 2: Without loss of generality, $T_{c_i}^{S} \subseteq T_{c_i}^{PV}$.

If this were not the case with some nonnegligible probability, it would lead to an easy distinguisher between Z^{PV} and Z^{S}, contradicting zero–knowledge. Thus it will fail only with negligible probability.

2.10 Main Results

We can now state formally our main results, in the strongest form.

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

THEOREM 2 (strong form): $\not\exists 1WF$ over $\Sigma = \{0\} \Longrightarrow (\mathcal{AVZKHV} = \mathcal{AVBPP})$

3 Facts about indistinguishability

In this section, we list basic theorems about computational and statistical indistiguishablility which we are going to use (as axioms) in our proof. We then present proofs, first dispensing with realitvely simple cases and them focusing our attention on theorem $B3$, which establishes a new connection between statistical and computational indistinguishability.

While we continue to use $\mathcal{PPT}$ as our notion of efficient, this setup can be readily converted to agree about other complexity notions. The theorems *A1–A7* follow from the definitions. The theorems *B1, B2, B3* use in addition our assumption $\not\exists 1WF$. (We remark that theorems 3 and 4 on universal apporximation and extrapolation are also relevant here, and for the sake of completness, should be inlcuded in this section as well.)

3.1 Basic Theorems

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

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

A2: $(D_1 \sharp D_2 \stackrel{c}{=} E_1 \sharp E_2) \Longrightarrow (D_1 \stackrel{c}{=} E_1)$
$(D_1 \sharp D_2 \stackrel{s}{=} E_1 \sharp E_2) \Longrightarrow (D_1 \stackrel{s}{=} E_1)$

A3: All three relations are polynomially transitive (with additive bounds on distinguishing probablity; namely transitivity can be used for distributions indexed by $x \in \Sigma^*$ only $|x|^{O(1)}$ times[2].)

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

A4: $(D \stackrel{s}{=} E) \Longrightarrow (D \sharp M(D) \stackrel{s}{=} E \sharp M(E))$

A5: $(D \sharp E \stackrel{s}{=} F \sharp M(F)) \Longrightarrow (D \sharp E \stackrel{s}{=} D \sharp M(D))$

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

A6: $(D \stackrel{c}{=} E) \Longrightarrow (D \sharp M(D)) \stackrel{c}{=} E \sharp M(E)$

A7: $(D \sharp E \stackrel{c}{=} F \sharp M(F)) \Longrightarrow (D \sharp E \stackrel{c}{=} D \sharp M(D))$

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

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

B3: Let $D \sharp E$ be an arbitrary distribution (we stress that we do not assume that $D \sharp E$ is sampleable). For any e, let $D \sharp e$ be conditional distribution on D given that the second component of $D \sharp E$ is fixed to e, and let $T_e \stackrel{\triangle}{=} \{d |\ d \sharp e$ is in the support set of $D \sharp e\}$. For a fixed machine $N \in \mathcal{PPT}$ consider the distribution $N(E) \sharp E$ (generated by picking en element e according to E for the second component and applying $N(\cdot)$ to it to

[2] Caution: this theorem must be used with care, and misusing it can lead to errors, as was pointed to us by R. Impagliazzo. The thing to beware of is the effect of transitivity on the exponent of $|x|$ implicit in the definition of $\stackrel{c}{=}$ and $\stackrel{s}{=}$.

obtain the first component). $T^N_e \overset{\triangle}{=} \{d|\ d\sharp e$ is in the support set of $N(e)\sharp e\}$. Then, the following imlication holds:

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

3.2 Preliminaries

Before we proceed to the proof of $B3$ let us indicate how other theorems are proven. $A1$ and $A3$ follows from definitions. $A2$ is by contradiction: assume that D_1 and E_1 can be distinguished, then we can distinguish $D_1\sharp D_2$ from $E_1\sharp E_2$ by applying our distinguisher to the first component. Similarly, $A4$ and $A6$ are by contradiction: assume that we can distingush $D\sharp M(D)$ from $E\sharp M(E)$, but then we can distinguish D from E, by applying M to the string in question and using the assumed distinguisher.

$A5$ (and $A7$) are derived using $A2$, $A3$ and $A4$: from our assumption that $D\sharp E \overset{s}{=} F\sharp M(F)$ and $A2$ it follows that $D \overset{s}{=} F$. This combined with $A4$ implies that $D\sharp M(D) \overset{s}{=} F\sharp M(F)$, but $F\sharp M(F) \overset{s}{=} D\sharp E$, hence using $A3$ we conlude that $D\sharp M(D) \overset{s}{=} D\sharp E$.

Theorem $B1$ follows from universal extrapolation (theorem 3) and our assumption that $\not\exists 1WF$. Theorem $B2$ follows from [G-89] (see theorem 5) and again our assumption that $\not\exists 1WF$. Now, we turn $B3$.

3.3 Proof of B3

The proof is by contradiction. That is, we assume *(b)* and $D\sharp E \overset{s}{\not=} N(E)\sharp E$ 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, let $\hat{N}(\omega, e)$ be a deterministic analog of $N(e)$. Fix some $e \in E$. For every d, let

$$W^d_e \overset{\triangle}{=} \{\omega|\ \text{such that } \hat{N}(\omega, e) = d\sharp e\}$$

We note that it is possible for W^d_e to be an empty set.

The general idea of the proof is that assuming *(b)* and $D\sharp E \overset{s}{\not=} N(E)\sharp E$, there is a noticable difference between the expected size $|W^d_e|$ when $d\sharp e$ comes from $D\sharp E$ versus $N(E)\sharp E$. Estimating this size using universal approximation can be used to construct an efficient distinguisher between $D\sharp E$ and $N(E)\sharp E$, contradicting *(a)*. We now give more details.

Let $W_e \overset{\triangle}{=} \bigcup_d W^d_e$. Let $U(e)$ denote a uniform distribution on T_e, (note that this is exactly the distribution of d in $D\sharp e$ according to *(b)*, and that this distribution *may not be* sampleable) and let $N(e)$ be a (sampleable) distribution on T^N_e which is computed according to a randomly chosen ω so that $\hat{N}(\omega, e) = d\sharp e$. Recall that by *(b)*, $T^N_e \subseteq T_e$ and hence we can consider from now $N(e)$ to be a distribution on T_e as well. Let $r(e)$ be a probability of e according to E.

Lemma 1 If $D\sharp E \stackrel{s}{\not\approx} N(E)\sharp E$ then, there $\exists A \subseteq \{E\}$ and $\exists \alpha$ of size $1/poly(\omega)$ such that:

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

Proof: trivial. ■

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

We first show that using universal approximation we can estimate, for every fixed e, the relevant quantities defined above to arbitrary accuracy.

Lemma 2 If $\not\exists 1WF$ then there exists $M \in \mathcal{PPT}$ such that on input $d\sharp e$ and $0 < \beta < 1$, M outputs (in time polynomial in $|d\sharp e|$ and $1/\beta$) p'_d and t' such that:

$$Prob\left((1-\beta)\cdot p_d < p'_d < (1+\beta)\cdot p_d\right) \geq 1-\beta$$

and

$$Prob\left((1-\beta)\cdot t < t' < (1+\beta)\cdot t\right) \geq 1-\beta$$

where probability is taken over coin-toses of M.

Proof: By universal approximation (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 let us define a random variable $g \stackrel{\triangle}{=} \frac{1}{p_d}$. 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 $t = |T_e|$, as otherwise with probability greater then $1/poly(\beta)$ there will be $d\sharp e \in N(E)\sharp E$ for which $|W_e^d| = 0$. This would give us an efficient distinguisher since for $d\sharp e \in N(E)\sharp E$, $|W_e^d| \geq 1$. ■

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

```
1.  On input $d\sharp e$ and $\alpha$ compute $p'_d$ and $t'$
    within $(1 \pm \frac{\alpha^4}{8})$ of $p_d$ and $t$.
2.  compute $b = p'_d \cdot t'$.
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 $d\sharp e$ comes from $N(E)\sharp E$ distriburion and when $d\sharp e$ comes from $D\sharp E$ distribution. First, let us ignore the fact that we are dealing with approximations to p_d and t and calculate how good a distinguisher we would get if we could calculate the values of p_d and t exactly. Thus, let us express this difference as δ_e defined as follows:

$$\delta_e \stackrel{\triangle}{=} \sum_d (p_d \cdot \min(p_d t, 1) - q_d \cdot \min(p_d t, 1))$$

We now wish to bound δ_e:

Lemma 3

$$||N(e) - U(e)||_1 = 2\gamma_e \quad \Longrightarrow \quad \delta_e \geq \gamma_e^2$$

Proof: Note that

$$\delta_e = \sum_{d \in T_e} ((p_d - q_d) \min(p_d t, 1))$$

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

$$\delta_e = \sum_{d \in T^0} ((p_d - q_d) \cdot 1) + \sum_{d \in T^1} ((p_d - q_d) \cdot p_d t)$$

Since $||N(e) - U(e)||_1 = 2\gamma_e$ by our assumption, the first summation is clearly γ_e. In the second summation, by our definition of T^1, $p_d t \leq 1$, hence,

$$\delta_e = \gamma_e + \sum_{d \in T^1} ((p_d - q_d) \cdot p_d t)$$

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

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

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

$$\sum_{d \in T^1} (q_d p_d - p_d^2) \cdot t = \sum_{d \in T^1} \left(\frac{1}{t} \cdot \frac{\gamma_e}{|T^1|} - \frac{\gamma_e^2}{|T^1|^2} \right) \cdot t$$

Since $|T^1| \leq t$ setting $|T^1| = t$ and summing gives us an $\gamma_e - \gamma_e^2$ bound. Thus, $\delta_e \geq \gamma_e - (\gamma_e - \gamma_e^2) = \gamma_e^2$. ■

Note that the above lemma holds for every value of γ_e. Thus in particular $\delta_e \geq 0$ for every e. Furthermore, by lemma 1, there $\exists A \subseteq \{E\}$ and $\exists \alpha$ of size $1/poly(\omega)$ such that $\sum_{e \in A} r(e) \geq \alpha$ and $\forall e \in A$ $||U(e) - N^{-1}(e)||_1 \geq \alpha$.

Thus, if δ is the gap between the probability our distinguisher outputs "heads" when the input $d\sharp e$ is from $N(E)\sharp E$ and when it is from $D\sharp E$, then (still assuming no approximation errors) we can bound it to be

$$\begin{aligned}\delta &= \left(Pr\left[e \in A\right] \cdot \sum_{e \in A} \delta_e\right) + \left(Pr\left[e \notin A\right] \cdot \sum_{e \notin A} \delta_e\right) \\ &\geq \alpha\alpha^2 + 0 = \alpha^3\end{aligned}$$

where probabilities are taken over E. Since we are dealing with approximations p'_d and t', by setting $\beta = \frac{\alpha^4}{8}$ from lemma 2, we can distinguish at least within $\frac{\alpha^3}{2}$ and we are done with $B3$. ■

4 Proof outline 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 his 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 $C_i^S \sharp P^*(C_i^S) \overset{s}{=} C_{i+1}^S$.
>
> On input x, algorithm A operates as follows: it picks a uniformly chosen random tape R^A for V. Then it simulates moves of prover P^* and and the original verifier V with random tape R^A. If it is V's turn, A simulates the original V to produce V's next message. When it is P's turn, A takes the history so far, and extrapolates the behavior of the simulator S on this history to compute S's prediction of P's next message. A accepts iff V accepts in this run.

Assuming that there are no one-way functions implies that P^*, and hence A are in $\mathcal{PPT}$. Also 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 $x \in L$. This is in fact what we show. In particular, for the statistical case [Ost-91] show that $C^{A(x \in L)} \overset{s}{=} C^{PV(x \in L)}$, which guarantees that for all $x \in L$ algorithm A will except with probability close to $\frac{2}{3}$. We show that in the computational case $Z^{A(x \in L)} \overset{c}{=} Z^{PV(x \in L)}$, which again guarantees that A will except all $x \in L$ with probability which is at most $1/\mathit{poly}$ away from $\frac{2}{3}$, which again guarantees the correctness of A. First, we 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 trivially guarantees acceptance with probability $\geq \frac{2}{3}$ (as it is trivial to distinguish rejecting and accepting transcripts of the conversation). The proof uses only the fact 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!

The proof 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 $R^A\sharp C_i^A$ and $R^{PV}\sharp C_i^{PV}$. Moreover, by inductive hypothesis, we assume that $R^A\sharp C_i^A \overset{s}{=} R^{PV}\sharp C_i^{PV}$. 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 the prover's 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 so that $S(\omega)$ outputs $(x, R, C_i\sharp m_{i+1}\sharp \ldots \sharp m_n)$ 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 ($x \in L$)

We wish to show that $C^A \overset{c}{=} C^{PV}$. What we show is actually stronger. In our main lemma, we show that for all rounds i,

$$Z_i^A \overset{c}{=} Z_i^{PV}$$

(recall that Z_i^{PV} is a short-hand for $R^{PV(x)}\sharp C_i^{PV(x)}$; Z_i^S is a shorthand for $R^{S(x)}\sharp C_i^{S(x)}$ and Z_i^A is for $R^{A(x)}\sharp C_i^{A(x)}$). We note that the first difference to observe from the statistical case is that we explicitly use the random tape in all the transcripts. The high-level idea of the proof of our main lemma is as follows. We know that

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

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 since the simulator is statistically close to the real conversation. Otherwise, the distributions $R^S \sharp C^S_{i+1}$ and $R^{PV} \sharp C^{PV}_i$ are $\overset{c}{=}$ but not $\overset{s}{=}$. If they were both efficiently sampleable, then we would have two distributions which are statistically different but computationally indistinguishable, which implies the existence of a one-way function by [G-89] note (in other words, theorem 5 would rule out this possibility and we would be done). But the real conversation is *not* sampleable. Thus, we have two distributions, (which are statistically different but computationally indistinguishable) such that one of this two distributions is not sampleable. Thus, we must be more refined. In order to prove our main lemma, we prove by induction that

(a) $R^A \sharp C^A_i \overset{s}{=} R^S \sharp C^S_i$ (i.e. our main lemma) *and at the same time*

(b) $R^A \sharp C^A_i \overset{c}{=} R^{PV} \sharp C^{PV}_i$

Actually, we first use theorem 5 and $(**)$ to show that (a) and (b) are equivalent, and hence it sufficient in the induction to prove either one of these two relations! The proof is by induction on rounds. For $i = 0$ both relations trivially hold. Now we assume that both (a) and (b) hold for round i and show that one of them holds for round $i + 1$ (and by their equivalence the other one holds as well). We choose whether to prove (a) or (b) depending on whether it is message of the prover or of the verifier as follows.

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

Case P This is the difficult case. Here we prove the first relation (a) 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 for $x \in L$, we would be able to distinguish the sizes of admissible R^S and R^{PV}, and thus construct a distinguisher between the actual conversation and the simulation.

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** for $x \in L$, 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 7

$$\forall i, \;\; R^{PV} \sharp C^{PV}_i \overset{s}{=} S^{-1}(C^{PV}_i) \sharp C^{PV}_i$$

Proof: Notice that from the definition of zero-knowledge and facts $1, 2$ that follow the definition of zero knowledge we know that for all i:

(1) $R^S \sharp C_i^S \stackrel{c}{=} R^{PV} \sharp C_i^{PV}$

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

We show that conditions (1), (2) (3) and our assumption that there is no one-way function imply the above theorem. By universal extrapolation (theorem 3), we know that:

(4) $R^S \sharp C_i^S \stackrel{s}{=} S^{-1}(C_i^S) \sharp C_i^S$

By theorems $A1$, $A3$, (1) and (4) we then have:

(5) $S^{-1}(C_i^S) \sharp C_i^S \stackrel{c}{=} R^{PV} \sharp C_i^{PV}$

Now applying theorem $A7$ to (5) we get:

(6) $S^{-1}(C_i^{PV}) \sharp C_i^{PV} \stackrel{c}{=} R^{PV} \sharp C_i^{PV}$

Finally, we note that (6), (2) and (3) and our assumption that there is no one-way functions are exactly conditions that are required in theorem $B3$. Thus, applying $B3$ we get:

(7) $S^{-1}(C_i^{PV}) \sharp C_i^{PV} \stackrel{s}{=} R^{PV} \sharp C_i^{PV}$

and we are done. $\blacksquare$

REMARK: The proof of theorem 2 is essentially equivalent to the proof of theorem 1, except that instead of x given non-uniformly, we run the sampling algorithm D to find x. The algorithm which is used to decide if $x \in L$ from theorem 1 works just the same, contradicting the fact that on distribution D, to decide if $x \in L$ is hard.

5 Proof of Theorem 1

5.1 Reduction (to our Main Lemma)

We first state theorem 1 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 $Z^{PV(x)} = R^{PV(x)} \sharp m_1^{PV(x)} \sharp \cdots \sharp m_n^{PV(x)} = R^{PV(x)} \sharp C_n^{PV(x)}$ (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. $Z^{S(x)} = R^{S(x)}\sharp m_1^{S(x)}\sharp\cdots\sharp m_n^{S(x)}$, and furthermore that $R^{S(x)}$ is uniform over $\{0,1\}^n$. Again we define $C_0^{S(x)} = \emptyset$, and $C_{i+1}^{S(x)} = C_i^{S(x)}\sharp m_{i+1}^{S(x)}$, so $Z^{S(x)} = R^{S(x)}\sharp C_n^{S(x)}$.

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 $R^{S(x)}\sharp C_n^{S(x)}$). 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 $C_{i+1}^S \overset{s}{=} C_i^S\sharp P^*(C_i^{S(x)})$.

By axiom *B1*, as $\bar{S} \in \mathcal{PPT}$, also $P^* \in \mathcal{PPT}$. 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: $Z_L^A \overset{c}{=} Z_L^{PV}$

Using our main lemma, we cam easily complete the proof. The $\mathcal{BPP}$ algorithm B for L will simply compute $Z^{A(x)}$ and accept x iff $m_n^{A(x)} = 1^n$ (i.e. if the verifier accepts in this conversation). 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 $\mathcal{PPT}$.

5.2 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) $R^A\sharp C_i^A \overset{s}{=} R^S\sharp C_i^S$, and

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

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

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

Lemmas $5, 6$ show that (1_i) and (2_i) are equivalent.

Lemma 5 For every i,

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

Proof: Follows from the zero-knowledge property and axiom *A2*. ■

Lemma 6 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*)
$(3_i), (2_i) \Longrightarrow R^A \sharp C_i^A \overset{c}{=} R^S \sharp C_i^S$, but both distributions are sampleable, and by axiom *B2* $(R^A \sharp C_i^A \overset{c}{=} R^S \sharp C_i^S) \Longrightarrow (R^A \sharp C_i^A \overset{s}{=} R^S \sharp C_i^S) = (1_i)$. ■

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. In lemma 7 we take care of the easy case – when it is verifier's move, by proving that (2_i) implies (2_{i+1}). In lemmas $8 - 12$ we show the difficult case – when it is prover's move, by proving that (1_i) implies (1_{i+1}).

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

Lemma 7 $(2_i) \Longrightarrow (2_{i+1})$

Proof: $R^A \sharp C_{i+1}^A = R^A \sharp C_i^A \sharp \hat{V}(R^A \sharp C_i^A) \overset{c}{=} R^{PV} \sharp C_i^{PV} \sharp \hat{V}(R^{PV} \sharp C_i^{PV}) = R^{PV} \sharp C_{i+1}^{PV}$, 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 8

(4_i) $C^S_{i+1} \stackrel{s}{=} C^S_i \sharp P^*(C^S_i)$

(5_i) $R^{PV}_i \sharp C^{PV}_i \stackrel{s}{=} S^{-1}(C^{PV}_i) \sharp C^{PV}_i$

Proof: Property (4_i) follows from the definition of P^*. (5_i) follows from theorem 7. ■

We remark that (5_i) is technically the most challenging, and the proof of (5_i) (inside theorem 7) is precisely where we use $B3$.

The purpose of the next lemma is to prove that the $i+1$ message of the simulation m^S_{i+1} cannot depend on the "random tape" R^S. Intuitively, it is so since when it is a prover's message m^{PV}_{i+1} 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 9 $R^S \sharp C^S_{i+1} \stackrel{s}{=} S^{-1}(C^S_i) \sharp C^S_{i+1}$

Proof: From (5_i), $C^{PV}_{i+1} = C^{PV}_i \sharp P(C^{PV}_i)$ and axiom $A4$, we have:

(6_{i+1}) $S^{-1}(C^{PV}_i) \sharp C^{PV}_{i+1} \stackrel{s}{=} R^{PV} \sharp C^{PV}_{i+1}$

From $(3_{i+1}), (6_{i+1})$ and axiom $A7$ imply $S^{-1}(C^S_i) \sharp C^S_{i+1} \stackrel{c}{=} R^S \sharp C^S_{i+1}$ and since both distributions are sampleable, $\stackrel{c}{=}$ can be replaced by $\stackrel{s}{=}$ to obtain the lemma. ■

Lemma 10 $S^{-1}(C^S_i) \sharp C^S_i \sharp P^*(C^S_i) \stackrel{s}{=} R^S \sharp C^S_{i+1}$

Proof: By definition of P^* and universal extrapolation, $C^S_i \sharp P^*(C^S_i) \stackrel{s}{=} C^S_{i+1}$. Thus, applying $A4$, $S^{-1}(C^S_i) \sharp C^S_i \sharp P^*(C^S_i) \stackrel{s}{=} S^{-1}(C^S_i) \sharp C^S_{i+1}$. However, from lemma 9 and $A3$ this equation can be extended to: $S^{-1}(C^S_i) \sharp C^S_i \sharp P^*(C^S_i) \stackrel{s}{=} S^{-1}(C^S_i) \sharp C^S_{i+1} \stackrel{s}{=} R^S \sharp C^S_{i+1}$, from which the lemma follows. ■

Lemma 11 $R^S \sharp C^S_i \stackrel{s}{=} S^{-1}(C^S_i) \sharp C^S_i$

Proof: By lemma 9 and axiom $A2$. ■

Lemma 12 (1_{i+1}) holds, i.e.
$R^S \sharp C^S_{i+1} \stackrel{s}{=} R^A \sharp C^A_{i+1}$

Proof: From lemma 11, (1_i) and axiom $A5$ we get:

(7_i) $R^S \sharp C^S_i \stackrel{s}{=} R^A \sharp C^A_i \stackrel{s}{=}$
$\stackrel{s}{=} S^{-1}(C^S_i) \sharp C^S_i \stackrel{s}{=} S^{-1}(C^A_i) \sharp C^A_i$

Applying $A4$ to second and fourth expression in (7_i) we get:

(8_i) $R^A \sharp C^A_i \sharp P^*(C^A_i) \stackrel{s}{=} S^{-1}(C^A_i) \sharp C^A_i \sharp P^*(C^A_i)$

Notice that $R^A \sharp C^A_i \sharp P^*(C^A_i) \stackrel{s}{=} R^A \sharp C^A_{i+1}$ by specification of algorithm A. Thus, by $A3$

(9_i) $R^A \sharp C^A_{i+1} \stackrel{s}{=} S^{-1}(C^A_i) \sharp C^A_i \sharp P^*(C^A_i)$

Finally, from (1_i) and $A2$ we know that $C_i^S \overset{s}{=} C_i^A$. Applying $A4$ twice (with $S^{-1}(\cdot)$ and $P^*(\cdot)$) to both sides, we get:

$$(10_i) \quad S^{-1}(C_i^S)\sharp C_i^S \sharp P^*(C_i^S) \overset{s}{=} S^{-1}(C_i^A)\sharp C_i^A \sharp P^*(C_i^A)$$

By $A3$, combining (10_i), (9_i) and lemma 10 yields the result. ■

Acknowledgments

We would like to thank Oded Goldreich, Russell Impagliazzo, Mike Luby, Silvio Micali, Noam Nisan and Martin Tompa for helpful discussions. We are especially grateful to Russell Impagliazzo for pointing out an error in an earlier draft in one of the important lemmas, which made us look harder for a correct proof.

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

[AH-87] W. Aiello, and J. Hastad Perfect Zero-Knowledge can be Recognized in Two Rounds *FOCS 87.*

[B-85] L. Babai, Trading Group Theory for Randomness, *Proc. 17th STOC, 1985, pp. 421–429.* See also L. Babai and S. Moran, Arthur-Merlin Games: a Randomized Proof System and a Hierarchy of Complexity Classes, J. Comput. Syst. Sci. 36 (1988) 254-276

[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.

[BP-92] M. Bellare, E. Petrank Making Zero-Knowledge Provers Efficient. *STOC-92.*

[B+ 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*

[DH-76] W. Diffie, M. Hellman, New directions 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. See also 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.

[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 Attacks *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, 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.*

[JVV-86] M. Jerrum, L. Valian and V. Vazirani Random Generation of Combinatorial Structures from a Uniform Distribution *Theoretical Computer Science* 43, pp. 169-188, 1986.

[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.*

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

[N-89] M. Naor “Bit Commitment Using Pseudo-Randomness”, Crypto-89 pp.123-132. (Also: J. of Cryptology).

[NY-89] M. Naor and M. Yung, “Universal One-Way Hash Functions and their Cryptographic Applications”, STOC 89.

[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.*