One-Way Functions, Hard on Average Problems, and Statistical Zero-Knowledge Proofs

Extended Abstract*

Rafail Ostrovsky†
MIT Laboratory for Computer Science
545 Technology Square, Cambridge, MA 02139

Abstract

In this paper, we study connections among one-way functions, hard on the average problems, and statistical zero-knowledge proofs. In particular, we show how these three notions are related and how the third notion can be better characterized, assuming the first one.

1 Introduction

One-way functions, hard on the average problems, and Statistical Zero-Knowledge proofs have received a lot of attention in both complexity theory and cryptography (see, for example, [12, 15, 16, 26].) We start with informal explanations of all three notions and provide formal definitions in the subsequent sections.

Informally, a poly-time computable function f is one-way if when we pick x uniformly at random and compute $y \leftarrow f(x)$, it is infeasible for any polynomial time machine to find x' in $\{f^{-1}(y)\}$ for a non-negligible fraction of the instances.

Again informally, a poly-time computable function f is hard on the average if when we pick y uniformly at random, it is infeasible for any polynomial time machine to find x' in $\{f^{-1}(y)\}$ for a non-negligible fraction of the instances. We note that we can formulate this as a decision problem: a *language* L is hard on the average if for a non-negligible fraction of the instances, chosen uniformly at random, it is infeasible for any probabilistic polynomial-time algorithm to decide if $x \in L$ with probability bounded away from $\frac{1}{2}$.

Also informally, a language L possesses a statistical zero-knowledge interactive proof [8] if an infinitely-powerful prover can convince a probabilistic polynomial-time verifier that $x \in L$ without releasing to the verifier any additional information. The definition of statistical zero-knowledge allows a negligible probability of error and requires that no additional information is released to the verifier in a very strong, information-theoretic sense.

1.1 Implications

Either the existence of a one-way function or the existence of a hard on the average problem separates P from NP. In the opposite direction, hard on the average problems and one-way functions may not exist even if P$\neq$NP. That is, it could be the case that under sampleable distributions everything is easy, but in the worst (very rare) cases it is hard [13, 19, 20]. If this turns out to be the case, it would be bad news for cryptography since most cryptographic primitives, including pseudo-random generators [2], digital signatures [9], identification schemes and private-key encryption were shown to imply the existence of a one-way function [14, 16, 17, 24].

There are cases, however, when only the existence of a hard on the average problem is required. For example, an NP machine can commit bits to a polynomially-bounded machine, given any hard on the average problem [23]. Thus, it is natural to ask what is the relationship between hard on the average problems and one-way functions. The existence of a one-way function implies the existence of a hard on the average problem, but the reverse implication is not known. In this paper, we show that the reverse implication holds when-

* Preliminary versoion appeared in the Proceedings of the Structure In Complexity Theory 1991 Conference, June 30-July 3, Chicago

† Supported by IBM Graduate Fellowship. Part of this work was done at IBM T.J. Watson Research Center and AT&T Bell Labs. Author's email address: raf@theory.lcs.mit.edu

ever the language in question also possesses a statistical zero-knowledge proof:

THEOREM 1: *If any hard on the average language possesses a statistical zero-knowledge proof, then one-way functions exist.*

Actually, our result is stronger: the theorem holds even if we relax the definition of statistical zero-knowledge and do not require the simulator to output private coin tosses of the verifier.

In addition, many researchers were concerned about providing a better characterization of various properties of statistical zero-knowledge [1, 3, 6, 22]. Its relationship to one-way functions, however, remained unknown. Hence, we consider the following question: does statistical zero knowledge imply a one-way function? For trivial languages, which do not require any interaction (i.e. languages in BPP) the answer is **no**. (That is, even if P=NP, languages in BPP are vacuously zero-knowledge — the prover does not have to talk.) In this paper, we show that for any hard on the average language, the implication does hold.

1.2 Bounding the power of the prover

An interactive proof-system, introduced by [8] involves two players, an infinitely-powerful prover P and a polynomially bounded verifier V. (P, V) is a proof system for L if for all $x \in L$ the prover can convince the verifier that this is so, with probability $1 - \frac{1}{2^{|x|}}$. Moreover, if $x \notin L$ then any infinitely-powerful and malicious prover should not be able to convince the verifier that $x \in L$ with probability greater then $\frac{1}{2^{|x|}}$. Thus, by an "infinitely-powerful" prover, we denote an upper bound that shows that no matter how hard the cheating prover may try, and no matter how powerful he is, he can not fool the poly-bounded verifier. When it comes to the protocol specification, however, one can ask how much power an honest prover requires just to follow the protocol. In this paper, we consider this question in connection to statistical zero-knowledge interactive proofs. That is, we consider the question of how powerful the prover should be, in order to give a statistical zero-knowledge proof.

Joe Kilian originally posed this question, which was answered in [3], where they showed that a randomized PSPACE prover is sufficient to give any statistical zero-knowledge proof, assuming that the Discrete Log problem is hard. Notice that the power of the prover should not be confused with the complexity of the language L for which a statistical zero-knowledge proof-system (P, V) is designed. In fact, only languages in $\Sigma_2^p \bigcap \Pi_2^p$ (or, more specifically, in $AM \bigcap co\text{-}AM$) can be proven in statistical zero-knowledge [1, 6].

Why should the prover be in PSPACE, if the languages in question are very low in the polynomial-time hierarchy? The same (unsatisfactory) state of affairs exists for interactive proofs (i.e., without zero-knowledge constraints) as well. For example, to prove a co-NP statement, the current lower-bound on the power of the prover is at least $\#P$ [21] (and, by results of Toda, contains the entire polynomial-time hierarchy [25].) It turns out, however, that the information-theoretic zero-knowledge property can be utilized to substantially reduce the power of the prover. In fact, we show that the prover need not be more powerful then a randomized NP machine, under a general cryptographic assumption:

THEOREM 2: *If there exists any one-way permutation, then for all statistical zero-knowledge proofs, the prover need not be more powerful then a randomized NP machine.*

We note that it is *necessary* for our prover to be randomized (i.e., to be able to flip coins) since it was shown [22] that only languages in BPP have Statistical Zero-Knowledge proofs with deterministic provers. In section 3 we present a stronger version of theorem 2 as well.

2 Preliminaries

Most of the notations and definitions are standard, and appeared before in the literature (for example, see [2, 3, 8, 16, 17].)

By $|x|$ we denote a length of string x, by $x \uparrow i$ we denote the first i bits of x, where $i \leq |x|$. We use "$\circ$" for string concatenation. We emphasize the number of inputs received by an algorithm as follows. If algorithm A receives only one input we write "$A(\cdot)$"; if it receives two we write "$A(\cdot, \cdot)$", and so on. If A is a probabilistic algorithm then, for any input i the notation $A(i)$ refers to the probability space which to the string σ assigns the probability that A, on input i, outputs σ. If S is a probability space we denote by $\mathbf{P}_S(A)$ the probability that S associates to the set A. If A consists of the single element e we write $\mathbf{P}_S(e)$ rather than $\mathbf{P}_S(\{e\})$. We denote by $[S]$ the set of elements to which S assigns positive probability. If $f(\cdot)$ and $g(\cdot, \cdots)$ are probabilistic algorithms then $f(g(\cdot, \cdots))$ is the probabilistic algorithm obtained by composing f and g (i.e.

running f on g's output). For any inputs $x, y, \ldots$ the associated probability space is denoted $f(g(x, y, \cdots))$. If S is a probability space then $x \leftarrow S$ denotes the algorithm which assigns to x an element randomly selected according to S (that is, x is assigned the value e with probability $\mathbf{P}_S(e)$) (in the case that $[S]$ consists of only one element e we write $x \leftarrow e$ rather than $x \leftarrow \{e\}$). By S_n we denote the probability space which assigns positive probability only to strings of length n. For probability spaces $S, T, \ldots$, the notation $\mathbf{P}(p(x, y, \cdots) : x \leftarrow S; y \leftarrow T; \cdots)$ denotes the probability that the predicate $p(x, y, \cdots)$ is true after the (ordered) execution of the algorithms $x \leftarrow S$, $y \leftarrow T$, etc. The notation $\{f(x, y, \cdots) : x \leftarrow S; y \leftarrow T; \cdots\}$ denotes the probability space which to the string σ assigns the probability $\mathbf{P}(\sigma = f(x, y, \cdots) : x \leftarrow S; y \leftarrow T; \cdots)$, f being some function. If S is a finite set we will identify it with the probability space which assigns to each element of S the uniform probability $\frac{1}{|S|}$. (Then $x \leftarrow S$ denotes the operation of selecting an element of S uniformly at random). We let PPT denote the set of probabilistic (expected) polynomial time algorithms. We call S *samplable* if there exists a polynomial-time TM M with input length $k(n)$ and output length $m(n)$ such that, for each $n \in N$, $M(x_n) = S_n$ when x_n is a string of length $k(n)$ chosen uniformly. We call a function μ: $\mathbf{N} \mapsto \mathbf{N}$ *negligible* if for every constant $c > 0$ there exists a N_c such that for all $n > N_c$, $\mu(n) < \frac{1}{n^c}$.

Definition 1 *The probability spaces E_1 and E_2 are* statistically indistinguishable within ϵ *if $|\mathbf{P}_{E_1}(T) - \mathbf{P}_{E_2}(T)| < \epsilon$ for all $T \subseteq [E_1] \cup [E_2]$.*

Definition 2 *Let $L \subseteq \{0,1\}^*$. An ensemble with index set L is a collection $\{E(x)\}_{x \in L}$ of probability spaces, one for each $x \in L$.*

Definition 3 *The ensembles $\{E_1(x)\}_{x \in L}$ and $\{E_2(x)\}_{x \in L}$ are* statistically indistinguishable *(written $\{E_1(x)\}_{x \in L} \cong \{E_2(x)\}_{x \in L}$) if for any polynomial p there exists an n such that for all $x \in L$ of length at least n, the probability spaces $E_1(x)$ and $E_2(x)$ are statistically indistinguishable within $\frac{1}{p(|x|)}$.*

Next, we define Interactive Turing machines and protocols. The probability that (A, B) accepts the common input x is denoted $\mathbf{P}(\,(A, B)$ accepts $x\,)$, and the probability space of all conversations between A and B on input x is denoted $(A \leftrightarrow B)(x)$ (the probability in both cases is taken over the random tapes of both A and B). Sometimes we want to make the coin tosses of B explicit. For any $R \in \{0,1\}^*$ we write $B(R)$ for the (deterministic) machine B with R as its random tape. Then $(A \leftrightarrow B(R))(x)$ denotes the probability space of conversations between A and $B(R)$ on input x (the probability is over the random tapes of A). We let $B(R; x, \alpha_1\beta_1 \ldots \alpha_{i-1}\beta_{i-1})$ denote the next message that $B(R)$ sends when the conversation up to this point was $\alpha_1\beta_1 \ldots \alpha_{i-1}\beta_{i-1}$.

Definition 4 *An interactive protocol (P, V) is an* interactive proof system *for the language L if the following conditions hold:*

- Completeness*: For every $x \in L$,* $\mathbf{P}(\,(P, V)$ *accepts* $x) \geq 1 - 2^{-|x|}$ *.*
- Soundness*: For every ITM $\widehat{P}$ and every $x \notin L$,* $\mathbf{P}(\,(\widehat{P}, V)$ *accepts* $x) \leq 2^{-|x|}$ *.*

P and V are referred to as the prover *and the* verifier *respectively.*

The view of the verifier during an interaction with the prover is everything he sees: that is, his own coin tosses and the conversation between himself and the prover. Accordingly we define the *view* if the verifier to be:

Definition 5 *Let $(P, \widehat{V})$ be an interactive protocol and let $x \in \{0,1\}^*$. The* view *of $\widehat{V}$ on input x is the probability space*

$$\mathrm{View}_{(P,\widehat{V})}(x) = \{\,(R, C) \,:\, R \leftarrow \{0,1\}^{p(|x|)}\,;$$
$$C \leftarrow (P \leftrightarrow \widehat{V}(R))(x)\,\}\,,$$

where p is a polynomial bounding the running time of $\widehat{V}$.

Next, we define *restricted-view*:

Definition 6 *Let $(P, \widehat{V})$ be an interactive protocol and let $x \in \{0,1\}^*$. The* restricted-view *of $\widehat{V}$ on input x is the probability space*

$$\text{Restricted-View}_{(P,\widehat{V})}(x) = \{\,(C) \,:\, R \leftarrow \{0,1\}^{p(|x|)}\,;$$
$$C \leftarrow (P \leftrightarrow \widehat{V}(R))(x)\,\}\,,$$

where p is a polynomial bounding the running time of $\widehat{V}$.

Definition 7 *An interactive protocol (P, V) is a* statistical zero knowledge *protocol (SZK protocol) for L if for every polynomial time ITM $\widehat{V}$ there exists a PPT algorithm $S_{\widehat{V}}(\cdot)$ such that $\{S_{\widehat{V}}(x)\}_{x \in L} \cong \{\mathrm{View}_{(P,\widehat{V})}(x)\}_{x \in L}$ (this $S_{\widehat{V}}$ is called the* simulator*).*

If we substitute *view* by *restricted-view* in the above definition, then we call such protocol a *restricted statistical zero-knowledge*. When we wish to make the coin-tosses of the simulator $S_{\widehat{V}}$ explicit, we write $S_{\widehat{V}}(\cdot)(R)$.

Definition 8 *An interactive protocol* $(\overline{P}, \overline{V})$ *for* L *is an* honest verifier statistical zero knowledge *protocol (honest verifier SZK protocol) for* L *if there exists a PPT algorithm* $S(\cdot)$ *such that* $\{S(x)\}_{x\in L} \cong \{\mathrm{View}_{(\overline{P},\overline{V})}(x)\}_{x\in L}$ *(this* S *is called the* honest simulator*).*

Definition 9 *We say that* f *is* weak one-way *on sampleable* D *if, for some constant* $c > 0$*, for every PPT algorithm* $M(\cdot)$*,*

$$\mathbf{P}(f(x) = f(M(f(x))) : x \leftarrow D) \leq 1 - \frac{1}{n^c}$$

Definition 10 *We say that* f *is* strongly one-way *on distribution* D *if for every PPT algorithm* $M(\cdot)$*,* $\mathbf{P}(f(x) = f(M(f(x))) : x \leftarrow D)$ *is negligible.*

We note that if D is not mentioned in the above two definitions, the uniform probability is assumed. Finally, we need the definition of

Definition 11 distributionally one-way function [IL]: *We say that* f *is* distributionally one-way *if, for some constant* $c > 0$*, for every probabilistic polynomial-time algorithm* A*, the distribution defined by* $x \circ f(x)$ *and the distribution defined by* $A(f(x)) \circ f(x)$ *are statistically distinguishable by at least* n^{-c} *when* $x \in \{0,1\}^n$ *is drawn with uniform distribution.*

Intuitively, if f is distributionally one-way, then it is computationally infeasible to randomly generate preimages of $f(x)$.

By hard on the average, we mean that for a non-negligible fraction of the instances, chosen under a sampleable distribution, it is infeasible for any probabilistic polynomial-time algorithm, for every positive constant c, and for every large enough n, to decide if $x \in L$ with probability greater then $\frac{1}{2} + \frac{1}{n^c}$.

Finally, we define information-theoretic *bit commitment* protocol for two parties, Alice and Bob. The protocol consists of two stages:

- The *commit* stage: Alice has a bit b on her input tape, which she wishes to commit to Bob. She and Bob exchange messages. At the end of this stage Bob has some information that represents b written on its output tape.
- The *reveal* stage: Alice and Bob exchange messages (where their output tapes from the commit stage are serving as input tapes for this stage). At the end of the exchange, Bob writes b on its output tape.

To be *perfectly-secure* the protocol must obey the following: for *all* (even infinitely-powerful) Turing machines Bob, for all probabilistic polynomial time Alice, for all polynomials p and for large enough security parameter k,

1. (Security property:) After the commit stage, when Alice follows the protocol, Bob cannot guess b with probability greater than $\frac{1}{2} + \frac{1}{p(k)}$.
2. (Binding property:) After the commit stage in which Bob follows the protocol, with probability at least $1 - \frac{1}{p(k)}$ the polynomial-time Alice can reveal only one possible value.

Note that the security property does not rely on Bob being polynomial time. If in addition, Bob's algorithm can be performed in polynomial-time, we say that the bit commitment is "efficient". Analogous to interactive proofs, the notion of simulation can be extended to bit-commitment proofs. Thus, we say that the bit commitment protocol is "simulatable" if view of Alice can be approximated (statistically) by a polynomial-time simulator.

3 Construction

First, we prove our main lemma, based on which both results will follow.

Let L be Statistical ZK language. By definition, there exists a simulator $S_{\widehat{V}}(\cdot)(R)$ that on the input $x \in L$ and a random string ω, outputs a possible view $S_{\widehat{V}}(x)(\omega)$ of the verifier, which is statistically close (over ω) to the actual view. Define

$$F_{S_{\widehat{V}}}(x, i, \omega) \quad = \quad x \circ i \circ S_{\widehat{V}}(x)(\omega) \uparrow i$$

Lemma 1 *If* L *is a hard on the average, then* $F_{S_{\widehat{V}}}$ *is distributionally one-way.*

Proof: Suppose $F_{S_{\widehat{V}}}$ is not distributionally one-way. Then, there exists a PPT algorithm A which contradicts "distributional one-wayness" of $F_{S_{\widehat{V}}}$. We use A to efficiently decide L most of the time, contradicting its hardness on the average.

Suppose (towards the contradiction) that there exists a PPT algorithm $A(\cdot,\cdot,\cdot)$ which, given $(x \circ i \circ S_{\widehat{V}}(x)(\omega') \uparrow i)$ finds ω (almost) uniformly distributed among all ω which satisfy $F_{S_{\widehat{V}}}(x,i,\omega) = (x \circ i \circ S_{\widehat{V}}(x)(\omega') \uparrow i)$. Using A as a subroutine, we construct A' which can be used in place of the prover for every round. That is, if $x \in L$, we will be able to compute answers of the "virtual prover":

Let α be the conversation between the "virtual prover" and the *honest* verifier V so far. A' calls $A(x,|\alpha|,\alpha)$ – in order to (almost) uniformly sample ω which satisfies $F_{S_{\widehat{V}}}(x,|\alpha|,\omega) = x,|\alpha|,\alpha$. Then A' outputs $|\alpha|+j$ bits of $S_{\widehat{V}}(x)(\omega)$ as the next message of the "virtual prover", where we assume (without loss of generality) the all messages of the prover are of length j. The key point to notice is that A' outputs the next message of the "virtual prover", without the knowledge of R that our verifier $V(x,R)$ uses. After computing the next message from the "virtual prover" we compute (this time using R, and the "history-so-far") the next message of the honest verifier V and repeat.

From the fact that $S_{\widehat{V}}$ produces conversations which are statistically close to the actual ones, that A finds (statistically close to uniform) preimages ω of $F_{S_{\widehat{V}}}$, and that the actual prover can convince verifier V with probability $1-\frac{1}{2^{poly(g(|x|))}}$, (where $g(|x|)$ is the bound on the number of rounds of the original protocol) it follows that if $x \in L$ then A' generates responses of the "virtual prover" so that the honest verifier V excepts with probability at least $\frac{2}{3}$. Since the "virtual prover" algorithm does not look at the coin-tosses R of the verifier, the protocol remains an interactive proof-system and we conclude that if $x \notin L$ then verifier will reject with probability $1-\frac{1}{2^{|x|}}$. Thus, we have an efficient decision procedure for L, a contradiction. □

Theorem 1 *If there exists a hard on the average language L under any polynomially-sampleable distribution, and membership in L can be proven in statistical zero-knowledge, then there is a one-way function.*

Proof: Combining Lemma 1 with the result of [16] that distributionally one-way function implies the existence of a general one-way function we get the above theorem for a hard on the average L under a uniform distribution. The result extends to any polynomially-sampleable distribution using [15]. □

We note that the above theorem holds even if we substitute *statistical zero-knowledge* by *restricted statistical zero-knowledge.*

Next, we explore the power of the prover which is needed in order to give a Statistical Zero-Knowledge proof:

Theorem 2 *For any language L which possesses a Statistical Zero-Knowledge proof-system (P,V), there exists a SZK proof-system (P',V), where P' is bounded to be a probabilistic NP machine, given any one-way permutation.*

Proof outline: Randomized NP machine can (close to) uniformly sample F^{-1} from lemma 1. This, however, only gives us a proof-system which works for *honest verifier* and with probability of success at least $\frac{2}{3}$ for $x \in L$ and $\frac{1}{3}$ for $x \notin L$. The main theorem in [3] which converts a Statistical Zero-Knowledge proof system which works for *honest verifier only* into the one which works for any verifier while maintaining Statistical Zero-Knowledge property can be generalized to work given any simulatable information-theoretic bit commitment. Finally, the simulatable information-theoretic bit commitment can be implemented, given any one-way permutation [23] and we are done. □

In fact, we have proven a more general result:

Theorem 3 *For any language L which possesses an* **honest verifier** *statistical zero-knowledge proof-system (P,V), there exists a SZK proof-system (P',V), where P' is bounded to be a probabilistic NP machine, given any simulatable information-theoretic bit commitment.*

4 Open questions

- It is an open question if hard on the average problem implies the existence of a one-way function. Our results propose a way to establish this, if one can find a hard on the average (but not NP-complete [6]) problem and show a statistical zero-knowledge proof for it. (Actually, the task is somewhat simpler since the zero-knowledge simulator does not need to output coin tosses of the verifier.)

- It is not known, if a *computational* zero-knowledge proof for a hard on the average problem implies the existence of a one-way function. We *conjecture* that the answer to this question is **yes**. However, so far, it is only known that a computational MA-type (single round) zero-knowledge proof of possession of information for a hard on the average problem does imply a bit-commitment scheme and, hence, a one-way function [5, 7]. In general, however, the question is open.

5 Announcement of a New Result

Jointly with Avi wigderson, the second question posed above have been resolved. That is, we consider a very general definition of Zero-Knowledge Proofs, where the Zero-Knowledge property may hold only computationally and only on a sampleable distribution. We say that a Zero-Knowledge Proof is *boring* if the prover does not have to speak (i.e., the language is in BPP) — otherwise the proof is *interesting*. We show the equivalence of *interesting* Zero-Knowledge Proofs and one-way functions.

Acknowledgments

I am very grateful to Silvio Micali for uncountable number of discussions and insightful suggestions concerning this work. I would like to thank Manuel Blum and Oded Goldreich for helpful suggestions.

References

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

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

[3] Bellare, M., S. Micali and R. Ostrovsky, "The (True) Complexity of Statistical Zero Knowledge" STOC 90.

[4] Diffie,W., and Hellman, M., "New Directions in Cryptography" IEEE Trans. on Info. Theory, vol. IT-22, 6, (1976), pp. 644-654.

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

[6] Fortnow, L., "The Complexity of Perfect Zero-Knowledge" STOC 87.

[7] Feige, U., and A. Shamir, "Zero Knowledge Proofs of Knowledge in Two Rounds" CRYPTO 89.

[8] S. Goldwasser, S. Micali and C. Rackoff, "The Knowledge Complexity of Interactive Proof-Systems", STOC 85, pp. 291-304.

[9] S. Goldwasser, S Micali and R. Rivest, "A Secure Digital Signature Scheme" SIAM J. Comp., 17:2, 1988, pp. 281-308.

[10] Goldreich, O., and L. Levin "A Hard-Core Predicate for all One-Way Functions" Proc. 21st STOC, 1989, pp.25-32.

[11] Goldreich, O., S. Micali, and A. Wigderson, "Proofs that Yield Nothing but their Validity", FOCS 86.

[12] Goldreich, O., S. Micali and A. Wigderson, "A Completeness Theorem for Protocols with Honest Majority," STOC 87.

[13] Y. Gurevich "The Challenger-Solver Game" Bull. of Europ. Assoc. for Theor. Comp. Sci., October 89.

[14] J. Hastad, "Pseudo-Random Generators under Uniform Assumptions" STOC 90

[15] R. Impagliazzo and L. Levin "No Better Ways to Generate Hard NP Instances than Picking Uniformly at Random" FOCS 90.

[16] R. Impagliazzo and M. Luby, "One-way Functions are Essential for Complexity-Based Cryptography" FOCS 89.

[17] R. Impagliazzo, R., L. Levin, and M. Luby "Pseudo-Random Generation from One-Way Functions," STOC 89.

[18] L. Levin "Average Case Complete Problems", SIAM Journal of Computing 15: 285-286, 1986.

[19] D. Johnson "The NP-Completeness Column - an Ongoing Guide. Journal of Algorithms, 5:284-299, 1984.

[20] R. Karp "The Probabilistic Analysis of Some Combinatorial Search Algorithms: Algorithms and Complexity, J.G. Traub ed., Academic Press, NY 1976, pp.1-19.

[21] Lund, C., L. Fortnow, H. Karloff, and N. Nisan, "Algebraic Methods for Interactive Proof Systems" FOCS 90.

[22] Oren Y., "On The Cunning Power of Cheating Verifiers: Some Observations About Zero Knowledge Proofs", FOCS 87.

[23] R. Ostrovsky, R. Venkatesan, M. Yung, in preparation.

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

[25] Toda, S., "On the computational power of PP and Parity-P" FOCS 89.

[26] Venkatesan R., and L. Levin "Random Instances of a Graph Coloring Problem are Hard" STOC 88.

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