# The Complexity of Zero Knowledge

Salil Vadhan*
School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138
salil@eecs.harvard.edu
http://eecs.harvard.edu/~salil

October 16, 2007

**Abstract**

We give an informal introduction to zero-knowledge proofs, and survey their role both in the interface between complexity theory and cryptography and as objects of complexity-theoretic study in their own right.

## 1 Introduction

*Zero-knowledge proofs* are interactive protocols whereby one party, the *prover*, can convince another, the *verifier*, that some assertion is true with the remarkable property that the verifier learns nothing other than the fact that the assertion being proven is true. In the quarter-century since they were introduced by Goldwasser, Micali, and Rackoff [GMR], zero-knowledge proofs have played a central role in the design and study of cryptographic protocols. In addition, they have provided one of the most fertile grounds for interaction between complexity theory and cryptography, leading to exciting developments in each area. It is the role of zero knowledge in this interaction that is the subject of the present survey.

We begin with an informal introduction to zero-knowledge proofs in Section 2, using two classic examples. In Section 3, we survey how zero-knowledge proofs have provided an avenue for ideas and techniques to flow in both directions between cryptography and complexity theory. In Section 4, we survey the way in which zero knowledge has turned out to be interesting as a complexity-theoretic object of study on its own. We conclude in Section 5 with some directions for further research.

## 2 Definitions and Examples

In this section, we provide an informal introduction to zero-knowledge proofs. For a more detailed treatment, we refer the reader to [Vad1, Gol].

*Written while visiting U.C. Berkeley, supported by the Miller Institute for Basic Research in Science, a Guggenheim Fellowship, and NSF grant CNS-0430336.

**Interactive Proofs and Arguments.** Before discussing what it means for a proof to be "zero knowledge," we need to reconsider what we mean by a "proof." The classical mathematical notion of proof is as a static object that can be written down once and for all, and then easily verified by anyone according to fixed rules. It turns out that the power of such classical proofs can be captured by the complexity class NP. To make this precise, we consider the assertions to be proven as strings over some fixed alphabet, and consider a language $L$ that identifies the assertions that are 'true'. For example, language SAT contains a string $x$ iff $x$ encodes a boolean formula $\phi$ such that the assertion "$\phi$ is satisfiable" is true. Then a *proof system* for a language $L$ is given by a verification algorithm $V$ with the following properties:

- (Completeness) True assertions have proofs. That is, if $x \in L$, then there exists $\pi$ such that $V(x, \pi) = \texttt{accept}$.
- (Soundness) False assertions have no proofs. That is, if $x \notin L$, then for all $\pi^*$, $V(x, \pi^*) = \texttt{reject}$.
- (Efficiency) $V(x, \pi)$ runs in time $\mathrm{poly}(|x|)$.

It is well-known that NP is exactly the class of languages having classical proof systems as defined above. (Indeed, NP is now often defined in this way, cf. [Sip].) Thus the P vs. NP question asks whether proofs actually help in deciding the validity of assertions, or whether deciding validity without a proof can always be done in time comparable to the time it takes to verify a proof.

Now zero-knowledge proofs are concerned with the question of how much one *learns* when verifying a proof. By definition, one learns that the assertion being proven is true. But we typically think of mathematical proofs as teaching us much more. Indeed, when given a classical NP proof, one also gains the ability to convince others that the same assertion is true, by copying the same proof. To get around this obstacle and make it possible to have proofs that leak "zero knowledge," Goldwasser, Micali, and Rackoff [GMR] added two ingredients to the classical notion of proof. The first is *randomization* — the verification of proofs can be probabilistic, and may err with a small but controllable error probability. The second ingredient is *interaction* — the static, written proof is replaced by a dynamic *prover* who exchanges messages with the verifier and tries to convince it to accept.

In more detail, we consider an interactive protocol $(P, V)$ between a "prover" algorithm $P$ and a "verifier" algorithm $V$. $P$ and $V$ are given a common input $x$, they each may privately toss coins, and then they exchange up to polynomially many messages (where the next message of each party is obtained by applying the appropriate algorithm $P$ or $V$ to the common input, the party's private coin tosses, and the transcript of messages exchanged so far). At the end of the interaction, the verifier accepts or rejects. We denote by $(P, V)(x)$ the interaction between $P$ and $V$ on input $x$. Analogous to classical proofs, we require the following properties:

- (Completeness) If $x \in L$, then $V$ accepts in $(P, V)(x)$ with probability at least $2/3$.
- (Soundness) If $x \notin L$, then for "all" $P^*$, $V$ accepts in $(P^*, V)(x)$ with probability at most $1/3$.
- (Efficiency) On common input $x$, $V$ always runs in time $\mathrm{poly}(|x|)$.

A consequence of the efficiency condition is that the total length of communication between the two parties is bounded by a polynomial in $|x|$. As with randomized algorithms, the constants of

2/3 and 1/3 in the completeness and soundness probabilities are arbitrary, and can be made be exponentially close to 1 and 0, respectively, by repeating the protocol many times and having the verifier rule by majority.

We think of the soundness condition as a "security" property because it protects the verifier from adversarial behavior by the prover. Like most security properties in cryptography, it has two commonly used versions:

- (Statistical Soundness) If $x \notin L$, then for all, even *computationally unbounded*, strategies $P^*$, $V$ accepts in $(P^*, V)(x)$ with probability at most 1/3. This gives rise to *interactive proof systems*, the original model of [GMR].
- (Computational Soundness) If $x \notin L$, then for all (nonuniform) *polynomial-time* strategies $P^*$, $V$ accepts in $(P^*, V)(x)$ with probability at most 1/3. This gives rise to *interactive argument systems*, a model proposed by Brassard, Chaum, and Crépeau [BCC].

Note that the honest prover $P$ must have some computational advantage over the verifier to be of any use. Otherwise, the verifier could simply simulate the prover on its own, implying that the language $L$ is decidable in probabilistic polynomial time (i.e. in the complexity class BPP). Thus, typically one either allows the honest prover $P$ to be computationally unbounded or requires $P$ to be polynomial time but provides it with an NP witness for the membership of $x$ in $L$. The former choice is mainly of complexity-theoretic interest, and is usually made only for interactive proof systems, since they also provide security against computationally unbounded cheating provers. The latter choice, where the prover is efficient given a witness, is the one most appropriate for cryptographic applications.

**Zero Knowledge.** While interactive proofs and arguments are already fascinating notions on their own (cf., [LFKN, Sha, Kil, Mic]), here we are interested in when such protocols possess a "zero knowledge" property — where the verifier learns nothing other than the fact that the assertion being proven is true. Before discussing how zero-knowledge can be defined precisely, we illustrate the notion with a classic example for Graph Nonisomorphism. Here an instance is a pair of graphs $(G_0, G_1)$, and it is a YES instance if $G_0$ and $G_1$ are non-isomorphic (written $G_0 \not\cong G_1$), and a NO instance if they are isomorphic (written $G_0 \cong G_1$).

The zero-knowledge proof is based on two observations. First, if two graphs are non-isomorphic, then their sets of isomorphic copies are disjoint. Second, if two graphs are isomorphic, then a random isomorphic copy of one graph is indistinguishable from a random isomorphic copy of the other (both have the same distribution). Thus, the proof system, given in Protocol 2.1, tests whether the (computationally unbounded) prover can distinguish between random isomorphic copies of the two graphs.

We first verify that this protocol meets the definition of an interactive proof system. If $G_0$ and $G_1$ are nonisomorphic, then $G_0 \cong H$ if and only if $b = 0$. So the prover strategy specified above will make the verifier accept with probability 1. Thus completeness is satisfied. On the other hand, if $G_0$ and $G_1$ are isomorphic, then $H$ has the same distribution when $b = 0$ as it does when $b = 1$ Thus, $b$ is independent of $H$ and the prover has at most probability at most 1/2 of guessing $b$ correctly *no matter what strategy it follows.* This shows that the protocol is sound.

For zero knowledge, observe that the only information sent from the prover to the verifier is the guess $c$ for the verifier's coin toss $b$. As argued above, when the statement being proven is true (i.e. $G_0 \not\cong H$), this guess is always correct. That is, the prover is sending the verifier a value

**Protocol 2.1: Interactive proof $(P, V)$ for** GRAPH NONISOMORPHISM

Common Input: Graphs $G_0$ and $G_1$ on vertex set $[n]$

1. $V$: Select a random bit $b \in \{0,1\}$. Select a uniformly random permutation $\pi$ on $[n]$. Let $H$ be the graph obtained by permuting the vertices of $G_b$ according to $\pi$. Send $H$ to $P$.

2. $P$: If $G_0 \cong H$, let $c = 0$. Else let $c = 1$. Send $c$ to $V$.

3. $V$: If $c = b$, accept. Otherwise, reject.

that it already knows. Intuitively, this means that the verifier learns nothing from the protocol. (Note that this intuition relies on the assumption that the verifier follows the specified protocol, and actually constructs the graph $H$ by permuting one of the two input graphs.)

The notion of zero knowledge is formalized by requiring that the verifier could have simulated everything it sees in the interaction on its own. That is, there should be a probabilistic polynomial-time, noninteractive algorithm $S$, called the *simulator*, that when given[1] "any" verifier strategy $V^*$ and any instance $x \in L$, produces an output that is "indistinguishable" from the verifier's view of its interaction with the prover on input $x$ (i.e. the transcript of the interaction together with the verifier's private coin tosses). Zero knowledge is a security property, protecting the prover from leaking unnecessary information to an adversarial verifier, and thus comes in both statistical and computational versions. With *statistical zero knowledge*, we require that the zero-knowledge condition hold for even computationally unbounded verifier strategies $V^*$, and require that the output of the simulator is statistically close (e.g. in variation distance) to the verifier's view. With *computational zero knowledge*, we only require the zero-knowledge condition to hold for (nonuniform) polynomial-time verifier strategies $V^*$ and require that the output of the simulator "computationally indistinguishable" from the verifier's view of the interaction, which means that no (nonuniform) polynomial-time algorithm can distinguish the two distributions except with negligible probability.

For the GRAPH NONISOMORPHISM protocol above, it is easy to illustrate a simulator that produces a distribution that is *identical* to the view of "honest" verifier $V$, but the protocol does not appear to be zero knowledge for verifier strategies $V^*$ that deviate from the specified protocol. Thus we refer to the protocol as being *honest-verifier* statistical zero knowledge (or even honest-verifier *perfect* zero knowledge, since the simulation produces exactly the correct distribution). Honest-verifier zero knowledge is already a very nontrivial and interesting notion, but cryptographic applications usually require the stronger and more standard notion of zero knowledge against cheating verifier strategies $V^*$. This stronger notion can be achieved for GRAPH NONISOMORPHISM using a more sophisticated protocol [GMW]. Thus we have:

**Theorem 2.2 ([GMW])** GRAPH NONISOMORPHISM *has a statistical zero-knowledge proof system*

[1] In this informal survey, we do not discuss the ways in which the simulator can be 'given' a verifier strategy. One possibility is that the simulator is given the code of the verifier, e.g. as a boolean circuit, which gives rise to the notion of *auxiliary-input zero knowledge* [GO]. Another is that the simulator is given the verifier strategy as an oracle, which gives rise to the notion of *black-box zero knowledge* [GO].

*(in fact a perfect zero-knowledge proof system).*

This provides an example of the power of zero-knowledge proofs (and also of interactive proofs, since Graph Nonisomorphism is not known to be in NP). An even more striking demonstration, however, is general construction of zero-knowledge proofs for all of NP, also due to [GMW].

**Zero Knowledge for NP.** To achieve this, Goldreich, Micali, and Wigderson [GMW] observed that it suffices to give a zero-knowledge proof for a single NP-complete problem, such as Graph 3-Coloring. A *3-coloring* of a graph $G = ([n], E)$ is an assignment $C : [n] \to \{\mathrm{R}, \mathrm{G}, \mathrm{B}\}$ (for "Red," "Green," and "Blue") such that no pair of adjacent vertices are assigned the same color. Graph 3-Coloring is the language consisting of graphs $G$ that are 3-colorable.

The zero-knowledge proof for Graph 3-Coloring is based on the observation that the classical NP proof can be broken into "pieces" and randomized in such a way that (a) the entire proof is valid if and only if every piece is valid, yet (b) each piece reveals nothing on its own. For Graph 3-Coloring, the classical proof is a three-coloring of the graph, and the pieces are the restriction of the coloring to the individual edges: (a) An assignment of colors to vertices of the graph is a proper 3-coloring if and only if the endpoints of every edge have distinct colors, yet (b) if the three colors are randomly permuted, then the colors assigned to the endpoints of any particular edge are merely a random pair of distinct colors and hence reveal nothing.

In Protocol 2.3, we show how to use the above observations to obtain a zero-knowledge proof for Graph 3-Coloring which makes use of "physical" implements — namely opaque, lockable boxes. The actual proof system will obtained by replacing these boxes with a suitable cryptographic primitive.

**Protocol 2.3: "Physical" Proof System $(P, V)$ for** Graph 3-Coloring

Common Input: A graph $G = ([n], E)$

1. $P$: Let $C$ be any 3-coloring of $G$ (either given as an auxiliary input to a polynomial-time $P$, or found by exhaustive search in case we allow $P$ to be computationally unbounded). Let $\pi$ be a permutation of $\{R, G, B\}$ selected uniformly at random. Let $C' = \pi \circ C$.
2. $P$: For every vertex $v \in [n]$, place $C'(v)$ inside a box $B_v$, lock the box using a key $K_v$, and send the box $B_v$ to $V$.
3. $V$: Select an edge $e = (x, y) \in E$ uniformly at random and send $e$ to $P$.
4. $P$: Receive edge $e = (x, y) \in E$, and send the keys $K_x$ and $K_y$ to $V$.
5. $V$: Unlock the boxes $B_x$ and $B_y$, and accept if the colors inside are different.

We now explain why this protocol works. For completeness, first observe that if $C$ is a proper 3-coloring of $G$ then so is $C'$. Thus, no matter which edge $(x, y) \in E$ the verifier selects, the colors

$C'(x)$ and $C'(y)$ inside boxes $B_x$ and $B_y$ will be different. Therefore, the verifier accepts with probability 1 when $G$ is 3-colorable.

For soundness, consider the colors inside the boxes sent by the prover in Step 2 as assigning a color to each vertex of $G$. If $G$ is not 3-colorable, then it must be the case that for some edge $(x, y) \in E$, $B_x$ and $B_y$ contain the same color. So the verifier will reject with probability at least $1/|E|$. By repeating the protocol $|E| + 1$ times, the probability that the verifier accepts on a non-3-colorable graph $G$ will be reduced to $(1 - 1/|E|)^{|E|+1} < 1/3$.

To argue that Protocol 2.3 is "zero knowledge," let's consider what a verifier "sees" in an execution of the protocol (when the graph is 3-colorable). The verifier sees $n$ boxes $\{B_v\}$, all of which are locked and opaque, except for a pair $B_x$, $B_y$ corresponding to an edge in $G$. For that pair, the keys $K_x$ and $K_y$ are given and the colors $C'(x)$ and $C'(y)$ are revealed. Of all this, only $C'(x)$ and $C'(y)$ can potentially leak knowledge to the verifier. However, since the coloring is randomly permuted by $\pi$, $C'(x)$ and $C'(y)$ are simply a (uniformly) random pair of distinct colors from $\{\mathrm{R}, \mathrm{G}, \mathrm{B}\}$, and clearly this is something the verifier can generate on its own.

In this intuitive argument, we have reasoned as if the verifier selects the edge $(x, y)$ in advance, or at least independently of the permutation $\pi$. This would of course be true if the verifier follows the specified protocol and selects the edge randomly, but the definition of zero knowledge requires that we also consider cheating verifier strategies whose edge selection may depend on the messages previously received from the prover (i.e., the collection of boxes). However, the perfect opaqueness of the boxes guarantees that the verifier has no information about their contents, so we can indeed view $(x, y)$ as being selected in advance by the verifier, prior to receiving any messages from the prover.

What is left is to describe how to implement the physical boxes algorithmically. This is done with a cryptographic primitive known as a *commitment scheme*. It is a two-stage interactive protocol between a pair of probabilistic polynomial-time parties, called the *sender* and the *receiver*. In the first stage, the sender "commits" to a string $m$, corresponding to locking an object in the box, as done in Step 2 of Protocol 2.3. In the second stage, the sender "reveals" $m$ to the receiver, corresponding to opening the box, as done in Steps 4 and 5 of Protocol 2.3.

Like zero-knowledge protocols, commitment schemes have two security properties. Informally, *hiding* says that a cheating receiver should not be able to learn anything about $m$ during the commit stage, and *binding* says that a cheating sender should not be able to reveal two different messages after the commit stage. Again, each of these properties can be statistical (holding against computationally unbounded cheating strategies, except with negligible probability) or computational (holding against polynomial-time cheating strategies, except with negligible probability). Thus we again get four flavors of commitment schemes, but it is easily seen to be impossible to simultaneously achieve statistical security for both hiding and binding. However, as long as we allow one of the security properties to be computational, it seems likely that commitment schemes exist. Indeed, commitment schemes with either statistical binding or statistical hiding can be constructed from any one-way function (a function that is easy to compute, but hard to invert even on random outputs) [HILL, Nao, NOV, HR], and the existence of one-way functions is the most basic assumption of complexity-based cryptography [DH, IL]. Thus, we conclude:

**Theorem 2.4** *If one-way functions exist, then every language in* NP *has both a computational zero-knowledge proof system and a statistical zero-knowledge argument system.*

We note that the first construction of statistical zero-knowledge argument systems was given by Brassard, Chaum, and Crépeau [BCC], in a work independent of [GMW], but was based on stronger cryptographic primitives than just statistically hiding commitment schemes.

## 3 Zero Knowledge as an Interface

In this section, we survey the way in which zero-knowledge proofs have provided an avenue for ideas and techniques to be transported between complexity theory and cryptography.

The concept of zero-knowledge proofs originated with efforts to formalize problems arising in the design of cryptographic protocols (such as [LMR]), where it is often the case that one party needs to convince another of some fact without revealing too much information. However, as evidenced even by the title of their paper "The Knowledge Complexity of Interactive Proof Systems," Goldwasser, Micali, and Rackoff [GMR] seemed to recognize the significance of the new notions for complexity theory as well. Indeed, interactive proof systems (as well as the Arthur–Merlin games independently introduced by Babai [Bab], which turned out to be equivalent in power [GS]), soon became a central concept in complexity theory. Their power was completely characterized in the remarkable works of Lund, Fortnow, Karloff, and Nisan [LFKN] and Shamir [Sha], which showed that IP, the class of languages having interactive proofs, equals PSPACE, the class of languages decidable in polynomial space. Since PSPACE is believed to be much larger than NP, this result shows that interactive proofs are much more powerful than classical written proofs.

In the other direction, we have already seen how a powerful concept from complexity theory, namely NP-completeness, was leveraged in the study zero-knowledge proofs, namely, Theorem 2.4. Traditionally, we think of NP-completeness as being used for negative purposes, to give evidence that a problem is hard, but here it has been used in a positive way — zero-knowledge proofs were obtained for an entire class by constructing them for a single complete problem. This discovery of zero-knowledge proofs for all of NP played a crucial role in striking general results of [Yao, GMW] about *secure computation*, where several parties engage in a protocol to jointly compute a function on their private inputs in such a way that no party learns anything other than the output of the protocol. These landmark results of [Yao, GMW] say that every polynomial-time computable function can be computed securely in this sense. Zero knowledge plays a crucial role, enabling the parties to convince each other that they are following the specified protocol, without revealing their private input.

In the study of secure computation, researchers realized that the use of complexity assumptions (e.g. the existence of one-way functions) could be removed by working in a model with private communication channels [CCD, BGW]. Similarly, Ben-Or, Goldwasser, Kilian, and Wigderson [BGKW] to introduced the *multiprover* model for interactive proofs, where two or more noncommunicating provers try to convince the verifier of an assertion, and the verifier can interrogate with each prover on a private channel that is inaccessible to the other prover(s) (similarly to how detectives interrogate suspects). The main motivation of [BGKW] was to find a model in which zero-knowledge protocols for all of NP could be obtained without any complexity assumption (in contrast to Theorem 2.4). However, multiprover interactive proofs turned out to be even more significant for complexity theory than interactive proofs were. Following the proof that IP = PSPACE mentioned above, Babai, Fortnow, and Lund [BFL] showed that the class MIP of languages having *multiprover* interactive proofs equals NEXP, nondeterministic exponential time, a class that is provably larger than NP (by diagonalization). Multiprover interactive proofs also turned out to be equivalent in power to *probabilistically checkable proofs* (PCPs) [FRS]. PCPs are static strings, like classical

NP proofs, but can be verified probabilistically by a verifier that reads only a small portion of the proof. Scaling down the proof that MIP = NEXP and incorporating a number of new ideas led to the celebrated PCP Theorem[BFLS, FGL$^+$, AS, ALM$^+$], showing that membership in any NP language can be proven using PCPs that can be verified by reading only a constant number of bits of the proof. The significance of the PCP Theorem was magnified by a surprising connection between PCP constructions for NP and showing that NP-complete optimization problems are hard to *approximate* [FGL$^+$, ALM$^+$], the latter being an open question from the early days of NP-completeness. A long line of subsequent work (beyond the scope of this survey) has optimized PCP constructions in order to get tight inapproximability results for a variety of NP-complete optimization problems.

The PCP Theorem provided returns to zero knowledge and cryptography through the work of Kilian [Kil], who used it to construct zero-knowledge *argument* systems for NP in which the verifier's computation time depends only polylogarithmically (rather than polynomially) on the length of the statement being proven. A generalization of Kilian's work, due to Micali [Mic], was used in [CGH] to obtain negative results about realizing the "random oracle model," which is an idealized model sometimes used in the design of cryptographic protocols. This technique of [CGH] was an inspiration for Barak's breakthrough work on "non-black-box simulation" zero knowledge [Bar1]. In this work, Barak showed how to exploit the actual code of the adversarial verifier's strategy to simulate a zero knowledge protocol (rather than merely treating the verifier as a black-box subroutine). Using this method, Barak obtained a zero-knowledge argument system with properties that were known to be impossible with black-box simulation [GK1]. Subsequently, non-black-box use of the adversary's code has proved to be useful in the solution of a number of other cryptographic problems, particularly ones concerned with maintaining security when several protocols are being executed concurrently [Bar2, PR1, Lin, Pas, PR2, BS].

# 4 Zero Knowledge as an Object of Study

We now turn zero knowledge as a complexity-theoretic object of study in itself. By this, we refer to the study of the complexity classes consisting of the languages that have zero-knowledge protocols of various types. We have already seen in the previous section that the classes IP and MIP arising from interactive proofs and their multiprover variant turned out to be very interesting and useful for complexity theory, and we might hope for the same to occur when we impose the zero knowledge constraint. From a philosophical point of view, it seems interesting to understand to what extent the requirement that we do not leak knowledge restricts the kinds of assertions we can prove. For cryptography, the complexity-theoretic study of zero knowledge can illuminate the limits of what can be achieved with zero-knowledge protocols, yield new techniques useful for other cryptographic problems, and help understand the relation of zero knowledge to other primitives in cryptography.

Recall that zero-knowledge protocols have two security conditions—soundness and zero knowledge—and these each come in both statistical and computational versions. Thus we obtain four main flavors of zero knowledge protocols, and thus four complexity classes consisting of the languages that zero-knowledge protocols of a particular type. We denote these classes SZKP, CZKP, SZKA, and CZKA, with the prefix of S or C indicating statistical or computational zero knowledge and the suffix of P or A denoting interactive proofs (statistical soundness) or arguments (computational soundness). The main goals are to characterize these classes, for example via complete problems or establishing relations with other, better-understand complexity classes; to establish properties of these classes (eg closure under various operations); and to obtain general results about zero-knowledge protocols. The first result along these lines was Theorem 2.4, which showed that the zero-knowledge classes involving computational security (namely, CZKP, SZKA, and CZKA) contain all of NP if one-way functions exist. Aside from this initial result and a follow-up that we will discuss later [IY, BGG$^+$], much of the complexity-theoretic study of zero knowledge was developed first for SZKP.

### 4.1 Statistical Security: SZKP

From a security point of view, statistical zero-knowledge proofs are of course the most attractive of the four types of zero-knowledge protocols we are discussing, since their security properties hold regardless of the computational power of the adversary. So the first question is whether this high level of security is achievable for nontrivial languages (i.e. ones that cannot be decided in probabilistic polynomial time). We have already seen one candidate, Graph Nonisomorphism, and in fact SZKP is known to contain a number of other specific problems believed to be hard, such as Graph Isomorphism [GMW], Quadratic Residuosity and Quadratic Nonresiduosity [GMR], a problem equivalent to the Discrete Log [GK2], approximate versions of the Shortest Vector Problem and Closest Vector Problem in high-dimensional lattices [GG], and various group-theoretic problems [AD]. On the other hand, recall that the general construction of zero-knowledge protocols for NP (Theorem 2.4) does not yield SZKP protocols, because (because there do not exist commitment schemes that are simultaneously statistically hiding and statistically binding). This phenomenon was explained in the work of Fortnow, Aiello, and Håstad [For, AH], who made the first progress towards a complexity-theoretic characterization of SZKP. Specifically, they showed that SZKP is contained in AM $\cap$ coAM, where the complexity class AM is a randomized analogue of NP, and consequently deduced that SZKP is unlikely to contain NP-hard problems. Indeed an NP-hard problem in SZKP $\subseteq$ AM $\cap$ coAM implies that AM = coAM, which seems unlikely for the same reason that NP = co-NP seems unlikely — there is no reason that a efficient provability of statements ($x \in L$) should imply efficient provability of their negations ($x \notin L$). (Like NP = co-NP, AM = coAM also implies the collapse of the polynomial-time hierarchy, which is commonly conjectured to be infinite.)

The next major steps in our understanding of SZKP came in the work of Okamoto [Oka], who proved that (a) SZKP is closed under complement, and (b) every language in SZKP has a statistical zero-knowledge proof system that is *public coin*, meaning that the verifier's messages consist only of random coin tosses (a property that holds for the Graph 3-Coloring protocol in the previous section, but not the Graph Nonisomorphism protocol).[2] The first result, closure under complement, was particularly surprising, because as mentioned above, there is no reason to believe that the existence of proofs for certain statements should imply anything about the negations of those statements. However, it was the second result that proved most useful in subsequent work, because public-coin protocols are much easier to analyze and manipulate than general, private-coin protocol. (Indeed, the equivalence of private coins and public coins for (non-zero-knowledge) interactive proofs [GS], found numerous applications, e.g. [BM, GS, BHZ, FGM$^+$].)

Using Okamoto's result as a starting point, SZKP was characterized exactly by two natural

[2] Okamoto's results were actually proven for *honest-verifier* statistical zero knowledge, but, as mentioned below, it was subsequently shown that every honest-verifier statistical zero-knowledge proof can be transformed into one that tolerates cheating verifiers [GSV1].

complete problems.[3] The first was STATISTICAL DIFFERENCE [SV], which amounts to the problem of approximating the statistical difference (i.e. variation distance) between two efficiently samplable distributions (specified by boolean circuits that sample from the distributions). The second problem, ENTROPY DIFFERENCE [GV], amounts to approximating the difference in the entropies of two efficiently samplable distributions (which is computationally equivalent to approximating the entropy of a single efficiently samplable distributions). In addition to providing a simple characterization of SZKP (as the class of problems that reduce to either of the complete problems), these complete problems show that the class SZKP is of interest beyond the study of zero-knowledge proofs. Indeed, estimating statistical properties of efficiently samplable distributions is a natural algorithmic task, and now we see that its complexity is captured by the class SZKP.

Using Okamoto's results and the complete problems, other general results about statistical zero knowledge were obtained, including more closure properties [DDPY, SV], an equivalence between honest-verifier SZKP and general, cheating-verifier SZKP [DGW, GSV1], an equivalence between efficient-prover SZKP and unbounded-prover SZKP for problems in NP [MV, NV], and relations between SZKP and other models of zero-knowledge protocols [GSV2, DSY, BG2]. There have also been studies of the relation between SZKP and quantum computation, including both the question of whether every problem in SZKP has a polynomial-time quantum algorithm [Aar, AT] and a complexity-theoretic study of the quantum analogue of SZKP [Wat]

## 4.2 Computational Security: CZKP, SZKA, and CZKA

Perhaps one reason that the complexity theory of SZKP developed more rapidly than that of the classes involving computational security is that early results seemed to indicate the latter were completely understood. Indeed, Theorem 2.4 says that under standard complexity assumptions, all of the classes CZKP, SZKA, and CZKA are very powerful, in that they contain all of NP. Soon afterwards, this result was strengthened was extended to give zero-knowledge proofs for all of IP [IY, BGG$^+$], again under the assumption that one-way functions exist. (This result allows for the honest prover to be computationally unbounded. For efficient honest provers, IP should be replaced by MA, which is a slight generalization of NP in which the verifier is a randomized algorithm.)

In cryptography, the assumption that one-way functions exist is standard; indeed, most of modern cryptography would not be able to get off the ground without it. However, from a complexity-theoretic perspective, there is a significant difference between results that make an unproven assumption and those that are unconditional. So a natural question is whether the assumption that one-way functions is really necessary to prove Theorem 2.4 and to characterize the power of zero knowledge with computational security.

Partial converses to Theorem 2.4, suggesting that one-way functions are necessary, were given by Ostrovsky and Wigderson [OW], building on an earlier work of Ostrovsky [Ost] about SZKP. Ostrovsky and Wigderson first proved that if there is a zero-knowledge protocol (even with both security properties computational) for a "hard-on-average" language, then one-way functions exist. Thus, we get a "gap theorem" for zero knowledge: either one-way functions exist and zero knowledge is very powerful, or one-way functions do not exist, and zero knowledge is relatively weak. They

[3]The complete problems for SZKP, as well as some of the other problems mentioned to be in SZKP are not actually languages, but rather *promise problems.* In a promise problem, some strings are YES instances, some strings are NO instances, and the rest are excluded (i.e. we are promised that the input is either a YES instance or a NO instance). Languages correspond to the special case where there are no excluded inputs.

also proved that if there is a zero-knowledge protocol for a language not in BPP (probabilistic polynomial time), then a "weak form" of one-way functions exist. (Note that we do not expect to deduce anything for languages in BPP, since every language in BPP has a trivial perfect zero knowledge proof, in which the prover sends nothing and the verifier decides membership on its own.)

While it was a major step in our understanding of zero knowledge, the Ostrovsky–Wigderson Theorems [OW] do not provide a complete characterization of the classes CZKA, CZKP, and SZKA. The reason is that for languages that are neither hard on average nor in BPP, we only get the "weak form" of one-way functions of their second result, which do not seem to suffice for constructing commitment schemes and hence zero-knowledge protocols. Exact characterizations were obtained more recently, using a variant of the Ostrovsky–Wigderson approach [Vad2, OV]. Instead of doing a case analysis based on whether a language is in BPP or not, we consider whether a language is in SZKP or not, and thus are able to replace the "weak form" of one-way functions with something much closer to the standard notion of one-way functions. Specifically, it was shown that every language $L$ in CZKA can be "decomposed" into a problem[4] in SZKP together with a set $I$ of instances from which (finite analogues of) one-way functions can be constructed. Conversely, every problem in NP having such a decomposition is in CZKA. A similar characterization is obtained for CZKP by additionally requiring that $I$ contains only strings in $L$, and for SZKA by requiring that $I$ contain only strings not in $L$. These results, referred to as the SZKP–OWF Characterizations, reduce the study of the computational forms of zero knowledge to the study of SZKP together with the consequences of one-way functions, both of which are well-understood. Indeed, using these characterizations, a variety of unconditional general results were proven about the classes CZKP, SZKA, and CZKA, such as closure properties, the equivalence of honest-verifier zero knowledge and general, cheating-verifier zero knowledge, and the equivalence of efficient-prover and unbounded-prover zero knowledge [Vad2, NV, OV]. Moreover, ideas developed in this line of work on unconditional results, such as [NV], turned out to be helpful also for conditional results, specifically the construction of statistically hiding commitments from arbitrary one-way functions [NOV, HR], which resolved a long-standing open problem in the foundations of cryptography (previously, statistically hiding commitments were only known from stronger complexity assumptions, such as the existence of one-way *permutations* [NOVY]).

# 5 Future Directions

Recall that our discussion of zero knowledge as an interface between complexity and cryptography in Section 3 ended with the non-black-box zero-knowledge protocol of Barak [Bar1], which found a variety of other applications in cryptography. It seems likely that the Barak's work will also have an impact on complexity theory as well. In particular, it points to the potential power of "non-black-box reductions" between computational problems. Typically, when we say that computational problem $A$ "reduces" to computational problem $B$, we mean that we can efficiently solve $A$ given access to a black box that solves problem $B$. We interpret such a reduction as saying that $A$ is no harder than $B$. In particular, if $B$ can be solved efficiently, so can $A$. However, it is possible to establish implications of the latter form without exhibiting a (black-box) reduction in the usual sense, because it may be possible to exploit an *efficient* algorithm for $B$ in ways that we cannot exploit a black-box for $B$ (e.g. by directly using the code of the algorithm in some way). While

[4]Again, the SZKP problems referred to by the SZKP–OWF Characterizations are actually promise problems.

we have had examples of "non-black-box reductions" in complexity theory for a long time (such as the collapse of the entire polynomial hierarchy to P if P = NP), Barak's work has begun to inspire complexity theorists to reexamine whether known limitations of black-box reductions (such as for worst-case/average-case connections [BT]) can be bypassed with various types of non-black-box reductions [GT].

In terms of the complexity-theoretic study of SZKP, one intriguing open problem is to find a combinatorial or number-theoretic complete problem. The known complete problems [SV, GV] can be argued to be "natural," but they still make an explicit reference to computation (since the input distributions are specified by boolean circuits). Finding a combinatorial or number-theoretic complete problem would likely further illuminate the class SZKP, and would also provide strong evidence that the particular problem is intractable. We are currently lacking in ways to provide evidence that problems are intractable short of showing them to be NP-hard. The recent sequence of results showing that Nash Equilibrium is complete for the class PPAD [DGP, CD] is one of the few exceptions. Approximate versions of lattice problems (see [GG, MV]) seem to be promising candidates for SZKP-completeness.

Another direction for further work is to carry out complexity-theoretic investigations, similar to those described in Section 4, for common variants of zero-knowledge protocols. These include noninteractive zero knowledge (for which there has been some progress [DDPY, GSV2, BG2, PS], mainly for the case of statistical security), proofs and arguments of knowledge (where the prover demonstrates that it "knows" a witness of membership), and witness-indistinguishable protocols (where the particular witness used by the prover remains hidden from the verifier, but other knowledge may be leaked). Also, we currently have a rather incomplete complexity-theoretic understanding of argument systems with sublinear communication, such as [Kil, Mic, BG1], not to mention their zero knowledge variants. The current constructions of such argument systems rely on collision-resistant hash functions, but we do not even know if one-way functions are necessary (cf., [Wee]).

# References


[Aar] Scott Aaronson. Quantum lower bound for the collision problem. In *Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing*, pages 635–642 (electronic), New York, 2002. ACM.

[AD] Vikraman Arvind and Bireswar Das. Szk proofs for black-box group problems. In Dima Grigoriev, John Harrison, and Edward A. Hirsch, editors, *CSR*, volume 3967 of *Lecture Notes in Computer Science*, pages 6–17. Springer, 2006.

[AH] William Aiello and Johan Håstad. Statistical zero-knowledge languages can be recognized in two rounds. *Journal of Computer and System Sciences*, 42(3):327–345, 1991. Preliminary version in *FOCS'87*.

[ALM$^+$] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998.

[AS] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: a new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998.

[AT] Dorit Aharonov and Amnon Ta-Shma. Adiabatic quantum state generation. *SIAM Journal on Computing*, 37(1):47–82 (electronic), 2007.

[Bab] László Babai. Trading group theory for randomness. In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing (STOC)*, pages 421–429, 1985.

[Bar1] Boaz Barak. How to go beyond the black-box simulation barrier. In *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 106–115. IEEE Computer Society, 2001.

[Bar2] Boaz Barak. Constant-round coin-tossing with a man in the middle or realizing the shared random string model. In *Proceedings of the 43rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 345–355, 2002.

[BCC] Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, 1988.

[BFL] László Babai, Lance Fortnow, and Carsten Lund. Nondeterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1(1):3–40, 1991.

[BFLS] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *STOC*, pages 21–31. ACM, 1991.

[BG1] Boaz Barak and Oded Goldreich. Universal arguments and their applications. In *IEEE Conference on Computational Complexity*, pages 194–203, 2002.

[BG2] Michael Ben-Or and Dan Gutfreund. Trading help for interaction in statistical zero-knowledge proofs. *Journal of Cryptology*, 16(2):95–116, 2003.

[BGG$^+$] Michael Ben-Or, Oded Goldreich, Shafi Goldwasser, Johan Håstad, Joe Kilian, Silvio Micali, and Phillip Rogaway. Everything provable is provable in zero-knowledge. In *Advances in Cryptology – CRYPTO '88*, volume 403 of *Lecture Notes in Computer Science*, pages 37–56. Springer, 1988.

[BGKW] Michael Ben-Or, Shafi Goldwasser, Joe Kilian, and Avi Wigderson. Multi-prover interactive proofs: how to remove intractability assumptions. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC)*, pages 113–131. ACM Press, 1988.

[BGW] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, pages 1–10, 1988.

[BHZ] Ravi B. Boppana, Johan Håstad, and Stathis Zachos. Does co-NP have short interactive proofs? *Information Processing Letters*, 25:127–132, 1987.

[BM] László Babai and Shlomo Moran. Arthur-Merlin games: A randomized proof system and a hierarchy of complexity classes. *Journal of Computer and System Sciences*, 36:254–276, 1988.

[BS] Boaz Barak and Amit Sahai. How to play almost any mental game over the net - concurrent composition via super-polynomial simulation. In *FOCS*, pages 543–552. IEEE Computer Society, 2005.

[BT] Andrej Bogdanov and Luca Trevisan. On worst-case to average-case reductions for NP problems. *SIAM Journal on Computing*, 36(4):1119–1159 (electronic), 2006.

[CCD] David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols (extended abstract). In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, pages 11–19, 1988.

[CD] Xi Chen and Xiaotie Deng. Settling the complexity of two-player nash equilibrium. In *FOCS*, pages 261–272. IEEE Computer Society, 2006.

[CGH] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *Journal of the ACM*, 51(4):557–594 (electronic), 2004.

[DDPY] Alfredo De Santis, Giovanni Di Crescenzo, Giuseppe Persiano, and Moti Yung. Image Density is complete for non-interactive-SZK. In *Automata, Languages and Programming, 25th International Colloquium, ICALP*, pages 784–795, 1998. See also preliminary draft of full version, May 1999.

[DGOW] Ivan Damgård, Oded Goldreich, Tatsuaki Okamoto, and Avi Wigderson. Honest verifier vs. dishonest verifier in public coin zero-knowledge proofs. In *Advances in Cryptology – CRYPTO '95*, volume 963 of *Lecture Notes in Computer Science*, pages 325–338. Springer, 1995.

[DGP] Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a Nash equilibrium. In *STOC'06: Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 71–78, New York, 2006. ACM.

[DGW] Ivan Damgård, Oded Goldreich, and Avi Wigderson. Hashing functions can simplify zero-knowledge protocol design (too). Technical Report RS-94–39, BRICS, November 1994. See Part 1 of [DGOW].

[DH] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[DSY] Giovanni Di Crescenzo, Kouichi Sakurai, and Moti Yung. On zero-knowledge proofs: "from membership to decision". In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 255–264. ACM Press, 2000.

[FGL+] Uriel Feige, Shafi Goldwasser, Laszlo Lovász, Shmuel Safra, and Mario Szegedy. Interactive proofs and the hardness of approximating cliques. *Journal of the ACM*, 43(2):268–292, 1996.

[FGM+] Martin Fürer, Oded Goldreich, Yishay Mansour, Michael Sipser, and Stathis Zachos. On completeness and soundness in interactive proof systems. *Advances in Computing Research*, 5:429–442, 1989. Preliminary version in *FOCS'87*.

[For] Lance Fortnow. The complexity of perfect zero-knowledge. *Advances in Computing Research: Randomness and Computation*, 5:327–343, 1989.

[FRS] Lance Fortnow, John Rompel, and Michael Sipser. On the power of multi-prover interactive protocols. *Theoretical Computer Science*, 134(2):545–557, 1994.

[GG] Oded Goldreich and Shafi Goldwasser. On the limits of non-approximability of lattice problems. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC)*, pages 1–9, 1998.

[GK1] Oded Goldreich and Hugo Krawczyk. On the composition of zero-knowledge proof systems. *SIAM Journal on Computing*, 25(1):169–192, 1996. Preliminary version in *ICALP'90*.

[GK2] Oded Goldreich and Eyal Kushilevitz. A perfect zero-knowledge proof system for a problem equivalent to the discrete logarithm. *Journal of Cryptology*, 6:97–116, 1993.

[GMR] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989. Preliminary version in *STOC'85*.

[GMW] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(1):691–729, 1991. Preliminary version in *FOCS'86*.

[GO] Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7(1):1–32, 1994.

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

[GS] Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. *Advances in Computing Research: Randomness and Computation*, 5:73–90, 1989.

[GSV1] Oded Goldreich, Amit Sahai, and Salil Vadhan. Honest verifier statistical zero-knowledge equals general statistical zero-knowledge. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC)*, pages 399–408, 1998.

[GSV2] Oded Goldreich, Amit Sahai, and Salil Vadhan. Can statistical zero-knowledge be made non-interactive?, or On the relationship of SZK and NISZK. In *Advances in Cryptology – CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 467–484. Springer, 1999.

[GT] Dan Gutfreund and Amnon Ta-Shma. Worst-case to average-case reductions revisited. In Moses Charikar, Klaus Jansen, Omer Reingold, and José D. P. Rolim, editors, *APPROX-RANDOM*, volume 4627 of *Lecture Notes in Computer Science*, pages 569–583. Springer, 2007.

[GV] Oded Goldreich and Salil P. Vadhan. Comparing entropies in statistical zero knowledge with applications to the structure of SZK. In *IEEE Conference on Computational Complexity*, pages 54–73. IEEE Computer Society, 1999.

[HILL] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999. Preliminary versions in *STOC'89* and *STOC'90*.

[HR] Iftach Haitner and Omer Reingold. Statistically-hiding commitment from any one-way function. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC)*. ACM Press, 2007.

[IL] Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 230–235, 1989.

[IY] Russell Impagliazzo and Moti Yung. Direct minimum-knowledge computations (extended abstract). In *Advances in Cryptology – CRYPTO '87*, volume 293 of *Lecture Notes in Computer Science*, pages 40–51. Springer, 1987.

[Kil] Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing (STOC)*, pages 723–732, 1992.

[LFKN] Carsten Lund, Lance Fortnow, Howard Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *Journal of the ACM*, 39(4):859–868, 1992.

[Lin] Yehuda Lindell. Protocols for bounded-concurrent secure two-party computation in the plain model. *Chicago Journal of Theoretical Computer Science*, pages Article 1, 50 pp. (electronic), 2006.

[LMR] Michael Luby, Silvio Micali, and Charles Rackoff. How to simultaneously exchange a secret bit by flipping a symmetrically-biased coin. In *FOCS*, pages 11–21. IEEE, 1983.

[Mic] Silvio Micali. Computationally sound proofs. *SIAM Journal on Computing*, 30(4):1253–1298, 2000. Preliminary version in *FOCS'94*.

[MV] Daniele Micciancio and Salil Vadhan. Statistical zero-knowledge proofs with efficient provers: lattice problems and more. In *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 282–298. Springer, 2003.

[Nao] Moni Naor. Bit commitment using pseudorandomness. *Journal of Cryptology*, 4(2):151–158, 1991. Preliminary version in *CRYPTO'89*.

[NOV] Minh-Huyen Nguyen, Shien Jin Ong, and Salil Vadhan. Statistical zero-knowledge arguments for NP from any one-way function. In *Proceedings of the 47th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 3–14. IEEE Computer Society, 2006.

[NOVY] Moni Naor, Rafail Ostrovsky, Ramarathnam Venkatesan, and Moti Yung. Perfect zero-knowledge arguments for NP using any one-way permutation. *Journal of Cryptology*, 11(2):87–108, 1998. Preliminary version in *CRYPTO'92*.

[NV] Minh-Huyen Nguyen and Salil Vadhan. Zero knowledge with efficient provers. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)*, pages 287–295. ACM Press, 2006.

[Oka] Tatsuaki Okamoto. On relationships between statistical zero-knowledge proofs. *Journal of Computer and System Sciences*, 60(1):47–108, 2000. Preliminary version in *STOC'96*.

[Ost] Rafail Ostrovsky. One-way functions, hard on average problems, and statistical zero-knowledge proofs. In *Proceedings of the 6th Annual Structure in Complexity Theory Conference*, pages 133–138. IEEE Computer Society, 1991.

[OV] Shien Jin Ong and Salil Vadhan. Zero knowledge and soundness are symmetric. In *Advances in Cryptology – EUROCRYPT 2007*, Lecture Notes in Computer Science. Springer, 2007.

[OW] Rafail Ostrovsky and Avi Wigderson. One-way functions are essential for non-trivial zero-knowledge. In *Proceedings of the 2nd Israel Symposium on Theory of Computing Systems*, pages 3–17. IEEE Computer Society, 1993.

[Pas] Rafael Pass. Bounded-concurrent secure multi-party computation with a dishonest majority. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 232–241 (electronic), New York, 2004. ACM.

[PR1] Rafael Pass and Alon Rosen. Bounded-concurrent secure two-party computation in a constant number of rounds. In *FOCS*, pages 404–. IEEE Computer Society, 2003.

[PR2] Rafael Pass and Alon Rosen. New and improved constructions of non-malleable cryptographic protocols. In *STOC'05: Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 533–542, New York, 2005. ACM.

[PS] Rafael Pass and Abhi Shelat. Unconditional characterizations of non-interactive zero-knowledge. In *Advances in Cryptology - CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 118–134. Springer, 2005.

[Sha] Adi Shamir. IP = PSPACE. *Journal of the ACM*, 39(4):869–877, 1992.

[Sip] Michael Sipser. *Introduction to the Theory of Computation*. Thomson Course Technology, Boston, MA, USA, second edition, 2005.

[SV] Amit Sahai and Salil Vadhan. A complete problem for statistical zero knowledge. *Journal of the ACM*, 50(2):196–249, 2003. Preliminary version in *FOCS'97*.

[Vad1] Salil Vadhan. Probabilistic proof systems, part I — interactive & zero-knowledge proofs. In S. Rudich and A. Wigderson, editors, *Computational Complexity Theory*, volume 10 of *IAS/Park City Mathematics Series*, pages 315–348. American Mathematical Society, 2004.

[Vad2] Salil P. Vadhan. An unconditional study of computational zero knowledge. *SIAM Journal on Computing*, 36(4):1160–1214, 2006. Preliminary version in *FOCS'04*.

[Wat] John Watrous. Limits on the power of quantum statistical zero-knowledge. In *Proceedings of the 43rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 459–, 2002.

[Wee] Hoeteck Wee. Finding Pessiland. In *Theory of cryptography*, volume 3876 of *Lecture Notes in Comput. Sci.*, pages 429–442. Springer, Berlin, 2006.

[Yao] Andrew Chi-Chih Yao. How to generate and exchange secrets. In *Proceedings of the 27th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 162–167. IEEE Computer Society, 1986.