Random Self-Reducibility and Zero Knowledge Interactive Proofs of Possession of Information

Martin Tompa
IBM Thomas J. Watson Research Center
P. O. Box 218
Yorktown Heights, New York 10598

Heather Woll
Department of Computer Science, FR35
University of Washington
Seattle, Washington 98195

Abstract

The notion of a zero knowledge interactive proof that one party "knows" some secret information is explored. It is shown that any "random self-reducible" problem has a zero knowledge interactive proof of this sort. The zero knowledge interactive proofs for graph isomorphism, quadratic residuosity, and "knowledge" of discrete logarithms all follow as special cases. Based on these results, new zero knowledge interactive proofs are exhibited for "knowledge" of the factorization of an integer, nonmembership in cyclic subgroups of $\mathcal{Z}_p^*$, and determining whether an element generates $\mathcal{Z}_p^*$. None of these proofs relies on any unproven assumptions.

1 Zero Knowledge Interactive Proofs

1.1 Informal Introductions

Goldwasser, Micali, and Rackoff [16] introduced the notions of "interactive proof" and "zero knowledge" in connection with distributed and cryptographic protocols. Informally, an interactive proof is a pair (P, V) of probabilistic coroutines executed by two parties, called the "prover" and the "verifier", whereby the prover attempts to convince the verifier of the validity of some proposition Π. Continuing informally, an interactive proof is "zero knowledge" if the verifier, even by deviating from its protocol, cannot gain any information from the prover (other than the validity of Π) that it could not have derived itself in polynomial expected time.

Until recently, the only type of proposition Π for which interactive proofs had been defined in the literature was that of language membership. In such a proof, the prover attempts to convince the verifier that the input is in some fixed language. There had been allusions to the existence of a second type, undefined until the recent paper of Feige, Fiat, and Shamir [13], in which the prover attempts to convince the verifier that it "knows" some secret, without giving away any information about the value of that secret.

The need for such an interactive proof typically arises when one party's protocol calls for it to compute some secret information, and the second party's interests would be compromised by proceeding without the assurance that the first party had done so. Examples from the literature are cited below:

1. The oblivious transfer [14,16] and the quadratic nonresiduosity protocol [16,5] each call for one party to prove that it "knows" a certain square root modulo N before proceeding.
2. The graph nonisomorphism protocol [15] calls for the verifier to prove that it "knows" an isomorphism between one of the two input graphs and a graph it has generated.
3. Chaum *et al.* [10,11] present protocols for "demonstrating possession of a discrete logarithm without revealing it".
4. The protocols for secret disclosure [8] and poker [12] call for "packet verification", which involves proving possession of certain information.

A new definition for this type of zero knowledge interactive proof is proposed in section 1.2, and used extensively in the sequel. The definition differs in some technical ways from that of Feige, Fiat, and Shamir [13], but the concept is the same. As can be surmised from the examples above, this concept is extremely useful in modularizing what would otherwise be quite complicated correctness proofs. For instance, using the new definition, corollary 5 proves that, given x and N, there is a zero knowledge interactive proof that the prover "knows" a square root of x modulo N. The construction and correctness of the oblivious transfer and quadratic nonresiduosity protocols can use this subprotocol as a black box.

Section 1.3 proves a general iteration lemma for zero knowledge interactive proofs. Namely, the probability of deceiving the verifier can be made exponentially small by iterating the proof linearly many times, without compromising the zero knowledge property.

Subsequent sections present new polynomial time zero knowledge interactive proofs for some number-theoretic problems that are not known (and not widely believed) to have polynomial time solutions. Among the problems considered are discrete logarithms, primitivity, and factorization.

This material is based upon work supported in part by the National Science Foundation under Grants DCR-8301212 and DCR-8352093. The work was performed while the second author was a visitor at the IBM Thomas J. Watson Research Center.

0272-5428/87/0000/0472$01.00 © 1987 IEEE

Goldreich, Micali, and Wigderson [15], Brassard and Crépeau [7], and Chaum [9] proved a very general result, namely that every language in $\mathcal{NP}$ has a polynomial time zero knowledge interactive proof, assuming the existence of secure encryption systems. The proofs in this paper do not depend upon any such unproven assumptions.

In section 2 the notion of "random self-reducibility" is formalized. It is shown that problems that are random self-reducible have polynomial time perfect zero knowledge interactive proofs. This single result unifies the zero knowledge interactive proofs for quadratic residuosity [16], discrete logarithms [10,11], and graph isomorphism [15], which all follow as immediate corollaries. The one for discrete logarithms is much simpler than that presented by Chaum *et al.* [10]. Chaum and van de Graaf [11] independently discovered this same simplification.

Sections 3-5 present new zero knowledge interactive proofs, based on results in section 2. Section 3 presents a polynomial time zero knowledge interactive proof that the prover "knows" the prime factorisation of a given integer. Section 4 presents a polynomial time zero knowledge interactive proof for membership in the set of generators of $\mathcal{Z}_p^*$, the multiplicative group of integers modulo some prime p. In section 5 it is shown that membership in $\mathcal{Z}_p^* - \langle a \rangle_p$ (the set of elements in $\mathcal{Z}_p^*$ not generated by some element a) has a polynomial time zero knowledge interactive proof. In a sense to be made precise, this is the complement of the discrete logarithm problem.

Beame [4] independently discovered zero knowledge interactive proofs for the problems of sections 4 and 5.

1.2 Definitions for Interactive Proofs

The formal model for the pair of coroutines is an *interactive pair of Turing machines* [16], slightly modified for convenience. This is a pair (A, B) of Turing machines operating on the following collection of tapes:

1. There is a single read-only *input tape* on which each of A and B has its own head.
2. Each of A and B has its own private one-way, read-only *random tape* containing an infinite string of uniform independent bits.
3. Each of A and B has its own private read/write *work tape.*
4. B has a private one-way, write-only *output tape.*
5. There is a pair of one-way *communication channel tapes.* One of these is write-only for A and read-only for B, and the other is write-only for B and read-only for A.

Let $T \in \{A, B\}$. $T(s)$ denotes T begun with s on its private work tape, its finite control in the start state, its input tape and work tape rewound, and its read-only communication channel tape empty; if s is omitted, it is assumed that the work tape is initially empty. $T_\rho(s)$ denotes $T(s)$ with random tape ρ; if ρ is omitted, $T(s)$ still begins with an infinite (but unspecified) random string.

$(A(s), B(s'))$ is said to *halt* if either A or B enters a halting state, to *accept* if B enters an accepting state, and to output z if z is the content of B's output tape when it accepts. $(A(s), B(s'))(x)$ denotes the output (if any), given that $(A(s), B(s'))$ begins with x on the input tape. Assume $(A(s), B(s'))$ halts on input x. $(\underline{A(s)}, B(s'))(x)$ denotes the quadruple $(x, s, \overline{\rho}, m)$, where $\overline{\rho}$ is the finite prefix of ρ that was read and m the final content of the communication channel tape on which B writes, given that $(A_\rho(s), B(s'))$ begins with x on the input tape. This quadruple will be called *A's history*, as it determines the actions of A. (Note that, although $\overline{\rho}$ is only a finite substring of ρ, it is always long enough to simulate A's behavior when $(A_\rho(s), B(s'))$ acts on input x.) Similarly, *B's history* $(A(s), \underline{B(s')})(x)$ denotes $(x, s', \overline{\rho'}, m')$, where $\overline{\rho'}$ is the finite portion of B's random tape that was read and m' is the final content of the communication channel tape on which A writes. Note that $(A(s), B(s'))(x)$, $(\underline{A(s)}, B(s'))(x)$, and $(A(s), \underline{B(s')})(x)$ are all random functions of x, s, and s'.

An interactive pair (P, V) of Turing machines is an *interactive proof for membership in the language L* if and only if, for some $\epsilon < 1$, (P, V) satisfies the following two conditions:

- *Completeness:* For any $x \in L$,
$$\Pr((P, V) \text{ accepts } x) = 1.$$
- *Soundness:* For any $x \notin L$ and for any prover's protocol P^*,
$$\Pr((P^*, V) \text{ accepts } x) \leq \epsilon.$$

The parameter ϵ is called the *error probability* of the interactive proof. Previous authors permitted a similar small probability of error in the definition of completeness. However, most interactive proofs do not exploit such error, and some of the subsequent results are simplified by prohibiting it. Lemma 2 in section 1.3 shows how to make the error probability arbitrarily small.

If M is any probabilistic algorithm with output, let $M(x)$ denote the random output produced by M on input x. If D is a circuit with one output gate and z is a random string, let $p_D(z) = \Pr(D \text{ outputs 1 on input } z)$. (P, V) is a *zero knowledge interactive proof* if and only if, in addition to completeness and soundness, it satisfies the following third condition:

- *Zero Knowledge:* For any polynomial expected time verifier's protocol V^*, there exists a probabilistic algorithm M_{V^*} such that, for any $x \in L$ and any s, M_{V^*} on input (x, s) runs in expected time $|x|^{O(1)}$, and
$$\text{for all polynomial size circuits } D, \quad |p_D((P, V^*(s))(x)) - p_D(M_{V^*}(x, s))| = \frac{1}{|x|^{\omega(1)}}. \tag{1}$$

$(P, V^*(s))(x)$ and $M_{V^*}(x, s)$ are said to be "polynomially indistinguishable" if condition 1 holds. The notion that V^* and M_{V^*} are each given some extra initial information s is a slight generalization of previous definitions of zero knowledge, seemingly required in the proof of lemma 2. Oren [17] independently made the same generalization in what he terms "auxiliary-input zero knowledge".

Finally, (P, V) is called a *perfect* zero knowledge interactive proof if and only if, in place of condition 1, it satisfies the more stringent condition

$$\text{for all } z, \Pr((P, V^*(s))(x) = z) = \Pr(M_{V^*}(x, s) = z). \quad (2)$$

The *time* used by an interactive proof (P, V) is the expected number of moves made by V before halting.

A second type of zero knowledge interactive proof, in which possession of information is demonstrated, is now introduced. In such proofs the prover, like the verifier, is time-bounded. The prover's only advantage over the verifier in this type will be that, in addition to the common input x, the prover's initial private work tape content s may contain some information about x unknown to the verifier. Let R be some binary relation. The goal of the prover in this second type of interactive proof will be to prove to the verifier that, from x and s, it can efficiently compute some y such that $(x, y) \in R$. A precise definition follows.

(P, V) is an *interactive proof that the prover can compute some y satisfying $(x, y) \in R$* if and only if, for some $\epsilon < 1$, it satisfies the following two conditions:

- *Completeness:* For any $(x, y) \in R$,
$$\Pr((P(y), V) \text{ accepts } x) = 1.$$
- *Soundness:* For any prover's protocol P^*, there exists a probabilistic algorithm C_{P^*} such that, for any x and s, C_{P^*} on input $(\underline{P^*_\rho(s)}, V_{\rho'})(x)$ runs in expected time polynomial in $|x|$ and the running time of $(P^*_\rho(s), V_{\rho'})$ on x, and
$$\Pr((P^*_\rho(s), V_{\rho'}) \text{ accepts } x \text{ and } (x, C_{P^*}((\underline{P^*_\rho(s)}, V_{\rho'})(x))) \notin R) \leq \epsilon. \quad (3)$$

Notice that the probability in condition 3 is over the random choices of ρ and ρ'. (A simpler definition of soundness that is also somewhat closer to that of Feige, Fiat, and Shamir [13] might replace condition 3 by something like

$$\Pr((P^*(s), V) \text{ accepts } x) > \epsilon \Rightarrow (x, C_{P^*}(x, s)) \in R.$$

This alternative appears inadequate for proving lemma 3, and so was not adopted in this paper.)

The definitions of zero knowledge and perfect zero knowledge for such an interactive proof are the same as the ones presented earlier for language membership, except that

1. "$(P, V^*(s))$" is replaced by "$(P(y), V^*(s))$", and
2. "$x \in L$" is replaced by "$(x, y) \in R$".

The definition of time for such an interactive proof (P, V) includes the expected number of moves made by P. The phrase "...even if the prover is no more powerful than the verifier" will be used as a reminder of this fact.

As an example of the idea, one of the new results presented in section 3 is that there is a zero knowledge interactive proof that the prover, given an integer x, can compute the prime factorization of x. The way such a situation might arise in practice is that the prover is the one supplying the common input x, which it has computed from some privately selected secret s. In the case of factorization, the prover might privately compute some large primes and multiply them together to produce x, which it then transmits to the verifier.

1.3 General Lemmas

Section 1 closes with 3 lemmas that are generally useful for zero knowledge interactive proofs.

The first lemma shows that it is sufficient for the simulator M_{V^*} in the definition of zero knowledge to be able to output verifier's histories of (P, V^*) accurately. This lemma is implicit in earlier papers [15,16].

Lemma 1 Zero knowledge, and perfect zero knowledge, can equivalently be defined by replacing $(P, V^*(s))(x)$ by $(P, \underline{V^*(s)})(x)$.

Proof: The original output-based notion implies the history-based notion proposed here, since any verifier's protocol with output can be modified to output instead its history. For the converse, notice that $(P, V^*(s))(x)$ can be computed in deterministic polynomial time, once V^* and $(P, V^*(s))(x)$ are fixed. Thus, if M_{V^*} can output $(P, \underline{V^*(s)})(x)$ accurately, it can be modified to output $(P, V^*(s))(x)$ just as accurately. □

The next two lemmas show that, like probabilistic algorithms, the error probability ϵ incurred by zero knowledge interactive proofs can be made to decay exponentially by iterating the protocol only linearly many times. Because of this, any ϵ bounded away from 1 suffices for the zero knowledge interactive proofs presented in subsequent sections. Previous papers had included this exponentially decaying error in the definitions.

Lemma 2 handles interactive proofs of language membership, and lemma 3 handles interactive proofs of information possession. The proofs of these lemmas may be skipped without loss of continuity.

Lemma 2 Let L be a language. For any $t = |x|^{O(1)}$, any interactive proof (zero knowledge interactive proof, perfect zero knowledge interactive proof, respectively) $\pi = (P, V)$ for membership in L with error probability ϵ can be transformed into one $\pi^t = (P^t, V^t)$ with error probability ϵ^t, at the expense of a factor of t in the time used.

Proof: π^t is the original proof π iterated t (probabilistically independent) times, with the following modification:

V^t accepts in π^t if and only if V accepts in each of the t iterations of π.

Completeness: Suppose $x \in L$, and prover and verifier both follow their protocols. By the completeness of π, $\Pr(\pi \text{ accepts } x) = 1$. Hence $\Pr(\pi^t \text{ accepts } x) = 1$.

Soundness: Suppose $x \notin L$, and the verifier follows its protocol. Let P^* be any prover's protocol for π^t. Notice that the t trials of π in (P^*, V^t) need not be independent, since the outcome of one trial may affect the subprotocol run by P^* in subsequent trials.

Consider P^* as a balanced computation tree of height t, where each node has some associated subprotocol $\tilde{P}$ that interacts with one iteration of V, and the branching is determined by the random tapes of P^* and V^t. At each node, prune the nonaccepting branches. By the soundness of π, for each $k \leq t$ the probability that π^t survives the k^{th} iteration, given that it survived the first $k-1$, is at most ϵ. Therefore, $\Pr((P^*, V^t) \text{ accepts } x) \leq \epsilon^t$.

Zero Knowledge: (This proof is due largely to Goldreich and Oren [personal communication], who discovered an error in an earlier version.) It is convenient for this proof to use the history-based notion of zero knowledge given in lemma 1. For any verifier's protocol $\tilde{V}$ for π, let $M_{\tilde{V}}$ be the simulator specified in lemma 1. Now let V^* be any verifier's protocol for π^t. Without loss of generality, for $i > 1$ assume V^* begins its i^{th} iteration with its finite control in the start state, its input tape and work tape rewound, its read-only communication channel tape empty to the right of its head, and its history during the first $i-1$ iterations written on its work tape. On input (x, s), the simulator $M^t_{V^*}$ of V^* does the following. Suppose that, after simulating $i-1$ iterations of π, it has produced the partial history s_i. Then $M^t_{V^*}$ simulates M_{V^*} on input (x, s_i) to extend the history s_i for another iteration.

If π is a perfect zero knowledge interactive proof, then π^t is as well. Suppose that π is not perfect, but that $(P, \underline{\tilde{V}(\tilde{s})})(x)$ and $M_{\tilde{V}}(x, \tilde{s})$ are polynomially indistinguishable, for all $\tilde{V}$ and $\tilde{s}$. Suppose that, for some polynomial size circuit C and some $0 \leq \delta \leq 1$,

$$|p_C((P^t, \underline{V^*(s)})(x)) - p_C(M^t_{V^*}(x, s))| \geq \delta. \qquad (4)$$

Any verifier's history $(P^t, \underline{V^*(s)})(x)$ can be decomposed into t single-iteration histories $((P, \underline{V^*(s_1)})(x), (P, \underline{V^*(s_2)})(x), \ldots, (P, \underline{V^*(s_t)})(x))$, where $s_1 = s$ and s_{i+1} is consistent with $(P, \underline{V^*(s_i)})(x)$. Similarly, $M^t_{V^*}(x, s)$ may be written $(M_{V^*}(x, s_1), M_{V^*}(x, s_2), \ldots, M_{V^*}(x, s_t))$. It follows from equation 4 that, for a slightly modified circuit D, there is an i satisfying

$$\begin{aligned} &|p_D((P, \underline{V^*(s_1)})(x), \ldots, (P, \underline{V^*(s_{i-1})})(x), (P, \underline{V^*(s_i)})(x), \\ &\quad M_{V^*}(x, s_{i+1}), \ldots, M_{V^*}(x, s_t)) \\ &- p_D((P, \underline{V^*(s_1)})(x), \ldots, (P, \underline{V^*(s_{i-1})})(x), M_{V^*}(x, s_i), \\ &\quad M_{V^*}(x, s_{i+1}), \ldots, M_{V^*}(x, s_t))| \geq \frac{\delta}{t}. \end{aligned}$$

Thus, there is a polynomial size circuit D' that, with probability at least $\frac{\delta}{t}$, distinguishes between the input distributions $(P, \underline{V^*(s_i)})(x)$ and $M_{V^*}(x, s_i)$: on input $h = (x, s_i, \rho_i, m_i)$, D' extracts $h_1 = (P, \underline{V^*(s_1)})(x), \ldots, h_{i-1} = (P, \underline{V^*(s_{i-1})})(x)$ from s_i, computes $h_{i+1} = M_{V^*}(x, s_{i+1}), \ldots, h_t = M_{V^*}(x, s_t)$ from h just as $M^t_{V^*}$ did, and finally simulates D on input $(h_1, \ldots, h_{i-1}, h, h_{i+1}, \ldots, h_t)$. By the polynomial indistinguishability of $(P, \underline{V^*(s_i)})(x)$ and $M_{V^*}(x, s_i)$, $\frac{\delta}{t} = \frac{1}{|x|^{\omega(1)}}$. But $t = |x|^{O(1)}$ by hypothesis, so $\delta = \frac{1}{|x|^{\omega(1)}}$. This implies that $(P^t, \underline{V^*(s)})(x)$ and $M^t_{V^*}(x, s)$ are polynomially indistinguishable. □

Lemma 3 Let R be a binary relation. Let $\pi = (P, V)$ be an interactive proof (zero knowledge interactive proof, perfect zero knowledge interactive proof, respectively) that the prover can compute some y satisfying $(x, y) \in R$, and suppose π has error probability ϵ. Suppose in addition that, given x and y, it can be determined in polynomial expected time whether or not $(x, y) \in R$. Then for any $t = |x|^{O(1)}$, there is an interactive proof (zero knowledge interactive proof, perfect zero knowledge interactive proof, respectively) $\pi^t = (P^t, V^t)$ that the prover can compute some y satisfying $(x, y) \in R$, where π^t has error probability ϵ^t and uses time a factor of t greater than that of π.

Proof: The construction, proof of completeness, and proof of zero knowledge are directly analogous to those of lemma 2, and are omitted here. Only the proof of soundness is presented.

For any prover's protocol $\tilde{P}$ for π, let $C_{\tilde{P}}$ be the algorithm specified by the definition of soundness. Now let P^* be any prover's protocol for π^t. Without loss of generality, for $i \geq 1$ assume P^* begins its i^{th} iteration with its finite control in the start state, its input tape and work tape rewound, and its read-only communication channel tape empty to the right of its head. The algorithm $C^t_{P^*}$ on input (x, s, ρ, m) does the following. Suppose that, during the i^{th} iteration of π, $P^*_\rho(s)$ begins with work tape content s_i and consumes substrings ρ_i of ρ and m_i of m. Note that s_i, ρ_i, and m_i can be computed deterministically from P^*, x, s, ρ, and m. Then $C^t_{P^*}$ simulates C_{P^*} on input (x, s_i, ρ_i, m_i). If this outputs y such that $(x, y) \in R$, then $C^t_{P^*}$ halts and outputs y. Otherwise, $C^t_{P^*}$ continues on to the $(i+1)^{th}$ iteration. If $C^t_{P^*}$ exhausts all t iterations without discovering such a y, it outputs arbitrarily.

By the soundness of π, for each i

$$\begin{aligned} &\Pr((P^*_{\rho_i}(s_i), V_{\rho'_i}) \text{ accepts } x \text{ and} \\ &\quad (x, C_{P^*}(x, s_i, \rho_i, m_i)) \notin R) \leq \epsilon. \end{aligned}$$

But $(P^*_\rho(s), V^t_{\rho'})$ accepts x if and only if $(P^*_{\rho_i}(s_i), V_{\rho'_i})$ accepts x, for all i, and $(x, C^t_{P^*}(x, s, \rho, m)) \notin R$ only if $(x, C_{P^*}(x, s_i, \rho_i, m_i)) \notin R$, for all i. Hence,

$$\begin{aligned} &\Pr((P^*_\rho(s), V^t_{\rho'}) \text{ accepts } x \text{ and} \\ &\quad (x, C^t_{P^*}(x, s, \rho, m)) \notin R) \leq \epsilon^t. \quad \square \end{aligned}$$

2 Random Self-Reducible Problems Have Zero Knowledge Proofs

Angluin and Lichtenstein [3] introduced the notion of "random self-reducibility" to describe functions possibly suitable as bases for secure encryption systems. Such functions have the property that they are as hard on average instances as they are on worst case instances. In this section, a generalization of random self-reducibility called "average reducibility" is introduced, and it is shown that average reducible problems have zero knowledge interactive proofs, in a sense to be made precise.

2.1 Average Reducibility

Let $\mathcal{N}$ be a countably infinite set. For any $N \in \mathcal{N}$, let $|N|$ denote the length of a suitable representation of N. For any $N \in \mathcal{N}$, let X_N, X'_N, Y_N, and Y'_N be finite sets, and $R_N \subseteq X_N \times Y_N$ and $R'_N \subseteq X'_N \times Y'_N$ be relations. Let

$$\text{dom } R_N = \{x \in X_N \mid (x,y) \in R_N \text{ for some } y \in Y_N\}$$

denote the "domain" of R_N, and

$$R_N(x) = \{y \mid (x,y) \in R_N\}$$

the "image" of $x \in X_N$. Let R be the relation

$$\{((N,x),y) \mid N \in \mathcal{N} \text{ and } (x,y) \in R_N\}$$

and R' the relation

$$\{((N,x),y) \mid N \in \mathcal{N} \text{ and } (x,y) \in R'_N\}.$$

R is *average reducible* to R' if and only if there is a $|N|^{O(1)}$ time algorithm A that, given any inputs $N \in \mathcal{N}$ and $x \in \text{dom } R_N$ and a source $r \in \{0,1\}^\omega$ of bits, outputs $x' = A(N,x,r) \in \text{dom } R'_N$ satisfying the following three properties:

R1. If the bits of r are random, uniform, and independent, then x' is uniformly distributed over dom R'_N.

R2. There is an $|N|^{O(1)}$ time algorithm that, given $N, x, \bar{r}$, and any $y' \in R'_N(x')$, outputs some $y \in R_N(x)$. Here, $\bar{r}$ is the finite prefix of r consumed in computing $A(N,x,r)$.

R3. There is an $|N|^{O(1)}$ time algorithm that, given N, x, r, and any $y \in R_N(x)$, outputs some $y' \in R'_N(x')$. If, in addition, the bits of r are random, uniform, and independent, then y' is uniformly distributed over $R'_N(x')$.

The relation R is said to be *random self-reducible* if and only if it is average reducible to itself. This is the case in the following 3 examples; that is, $X_N = X'_N, Y_N = Y'_N$, and $R_N = R'_N$. The more general notion of average reducibility will be exploited in section 5.

Example 1 *(square roots mod N):* Let $\mathcal{N}$ be the set of positive integers, $N \in \mathcal{N}$, and $R_N \subseteq \mathcal{Z}_N^* \times \mathcal{Z}_N^*$ be given by $(x,y) \in R_N$ if and only if $x \equiv y^2 \pmod N$. The domain of R_N is the set of quadratic residues mod N. Then R is random self-reducible, as follows. A random uniformly distributed value $\hat{r}$ in $\mathcal{Z}_N^*$ can be extracted from the random bit source r in a straightforward way. For any quadratic residue x, let A map (N,x,r) to $x' = \hat{r}^2 x \bmod N$. It is easy to show that x' is a uniformly distributed quadratic residue. The other properties follow from the fact that the equivalence $y' \equiv \hat{r}y \pmod N$ provides an efficient method of computing either y or y', given the other.

Example 2 *(discrete logarithms):* Let $\mathcal{N} = \{(p,a) \mid p \text{ is prime and } a \in \mathcal{Z}_p^*\}$. Let $R_{(p,a)} \subseteq \mathcal{Z}_p^* \times \mathcal{Z}_{p-1}$ be given by $(x,y) \in R_{(p,a)}$ if and only if $x \equiv a^y \pmod p$. The domain of $R_{(p,a)}$ is $\langle a \rangle_p$, the multiplicative subgroup of $\mathcal{Z}_p^*$ generated by a. Then R is random self-reducible, as follows. For $x \in \langle a \rangle_p$ and $\hat{r} \in \mathcal{Z}_{p-1}$, let A map (p,a,x,r) to $x' = a^{\hat{r}} x \bmod p$. It is easy to show that x' is uniformly distributed over $\langle a \rangle_p$. The other properties follow from the fact that the equivalence $y' \equiv \hat{r} + y \pmod{p-1}$ provides an efficient method of computing either y or y', given the other.

Example 3 *(graph isomorphism):* Let $\mathcal{N}$ be the set of undirected, vertex-labeled graphs. Let n be a positive integer, $\Gamma_n \subseteq \mathcal{N}$ be the set of labeled graphs on n vertices, S_n be the set of permutations on n elements, and $G = (V,E) \in \Gamma_n$. Let $R_G \subseteq \Gamma_n \times S_n$ be given by

$$R_G = \{(G',\pi) \mid G' = (V',E'), \text{ and } (u,v) \in E \text{ iff } (\pi(u),\pi(v)) \in E'\}.$$

The domain of R_G is the set of graphs isomorphic to G. Then R is random self-reducible, via the mapping A from G' to a random isomorphic copy of G'.

2.2 Zero Knowledge Proofs for Average Reducible Problems

This section assumes the formulation of interactive proofs for possession of information. Specifically, if R is average reducible to R', and N and $x \in \text{dom } R_N$ are input, the prover's extra information will be some y satisfying $(x,y) \in R_N$, and the prover's goal will be to convince the verifier that the prover knows such a y. Section 2.3 discusses applications to the original formulation of zero knowledge interactive proofs for language membership.

Theorem 4 Let R and R' satisfy the following conditions:

T0. R is average reducible to R'.

T1. There is a probabilistic $|N|^{O(1)}$ expected time algorithm that, given N, x', and y', determines whether $(x',y') \in R'_N$.

T2. There is a probabilistic $|N|^{O(1)}$ expected time algorithm that, on input N, outputs random pairs $(x', y') \in R'_N$ with x' uniformly distributed over dom R'_N and y' uniformly distributed over $R'_N(x')$.

Then, on input N and x, there is an $|N|^{O(1)}$ time perfect zero knowledge interactive proof that the prover can compute some y satisfying $((N, x), y) \in R$, even if the prover is otherwise no more powerful than the verifier. Furthermore, only the proof of zero knowledge relies on conditions R1 and T2.

(Note that the three examples of random self-reducible relations from section 2.1 each satisfy the additional conditions T1 and T2 of this theorem.)

Proof: CONSTRUCTION: On input N and x, the prover's and verifier's protocols are as follows:

Prover	*Comm. Channel*	*Verifier*
let y satisfy $(x, y) \in R_N$; let $r \in \{0,1\}^\omega$ be a source of random, uniform, independent bits; $x' \leftarrow A(N, x, r)$; let $\overline{r}$ be the finite prefix of r consumed in computing $A(N, x, r)$; compute $y' \in R'_N(x')$ from (N, x, r, y) via condition R3;	$\xrightarrow{x'}$	
		choose $\beta \in \{0,1\}$ randomly and uniformly;
	$\xleftarrow{\beta}$	
if $\beta \notin \{0,1\}$ **then halt**; **if** $\beta = 0$ **then** $z \leftarrow \overline{r}$ **else** $z \leftarrow y'$;		
	$\xrightarrow{z}$	
		if $(\beta = 0$ & $A(N, x, z) = x')$ **or** $(\beta = 1$ & $(x', z) \in R'_N)$ **then accept else reject.**

CORRECTNESS:

Completeness: Suppose the prover's initial work tape content is some y satisfying $(x, y) \in R_N$, and prover and verifier both follow the protocols given. By the definition of average reducibility and condition R3, the prover can compute $x' = A(N, x, r)$ and $y' \in R'_N(x')$. Therefore, no matter what value of β the verifier chooses, the prover will respond with a value of z that causes the verifier to accept. (If $\beta = 1$, condition T1 ensures that the verifier can determine that $(x', z) \in R'_N$.) Therefore the verifier accepts with probability 1.

Soundness: It will be shown that the error probability ϵ is $\frac{1}{2}$. Let P^* be any protocol for the prover, and let $P^*_\rho(s)(N, x, h)$ be the next message (if any) sent by $P^*_\rho(s)$ on inputs N and x, assuming h is the list of messages sent so far on the communication channel tapes. The algorithm C_{P^*}, on input (N, x, s, ρ, m), does the following:

$x' \leftarrow P^*_\rho(s)(N, x, \Lambda)$; **comment:** Λ = empty string;
$\overline{r} \leftarrow P^*_\rho(s)(N, x, (x', 0))$;
$y' \leftarrow P^*_\rho(s)(N, x, (x', 1))$;
if $A(N, x, \overline{r}) = x'$ & $(x', y') \in R'_N$
then output y computed from $(N, x, \overline{r}, y')$ via condition R2
else output arbitrarily.

If P^* fails to respond, C_{P^*} when simulating P^* will also fail to terminate. If $C_{P^*}(N, x, s, \rho, m)$ outputs y in the **then**-clause, condition R2 guarantees that $(x, y) \in R_N$. Note that only the first bit β of the verifier's random bit string ρ' affects acceptance in the interactive proof. Let

$$B = \{(\rho, \beta) \mid (P^*_\rho(s), V_\beta) \text{ accepts } (N, x) \text{ and } (x, C_{P^*}((P^*_\rho(s), V_\beta)(N, x))) \notin R_N\}.$$

For any ρ, the construction of C_{P^*} prevents the simultaneous occurrences of $(\rho, 0) \in B$ and $(\rho, 1) \in B$. Hence, $\Pr((\rho, \beta) \in B) \leq \frac{1}{2}$.

That C_{P^*} on input (N, x, s, ρ, m) runs in expected time $|N|^{O(1)}$ follows from conditions R2 and T1 and the running times of P^* and A.

Zero Knowledge: Let V^* be any polynomial expected time algorithm for the verifier, and s be any string. Let $V^*_\rho(s)(N, x, h)$ be the next message (if any) sent by $V^*_\rho(s)$ on inputs N and x, assuming h is the list of messages sent so far on the communication channel tapes. The simulator M_{V^*}, on input (N, x, s), does the following:

let $\rho \in \{0,1\}^\omega$ be a source of random, uniform, independent bits;
repeat forever
begin
choose $\beta' \in \{0,1\}$ randomly and uniformly;
comment: M_{V^*}'s guess of V^*'s β;
if $\beta' = 0$
then begin
let $r \in \{0,1\}^\omega$ be a source of random, uniform, independent bits;
$x' \leftarrow A(N, x, r)$;
let z be the finite prefix of r consumed in computing $A(N, x, r)$
end
else choose $(x', z) \in R'_N$ randomly with x' uniformly distributed over dom R'_N and z uniformly distributed over $R'_N(x')$, via condition T2;
$\beta \leftarrow V^*_\rho(s)(N, x, x')$;
if $\beta \notin \{0,1\}$ **then halt** and output $((N, x), s, \overline{\rho}, x')$;
if $\beta = \beta'$ **then halt** and output $((N, x), s, \overline{\rho}, (x', z))$;
end .

Let N, x, and y satisfy $(x,y) \in R_N$. It will be demonstrated that $(P(y), \underline{V^*(s)})(N,x)$ and $M_{V^*}(N,x,s)$ are identically distributed verifier's histories. The result then follows from lemma 1. Suppose that M_{V^*} outputs $((N,x),s,\overline{p},(x',z))$. No matter what the value of β', x' is uniformly distributed over dom R'_N (conditions R1 and T2), just as it is when the prover following its protocol sends it to the verifier (condition R1). The random variable β is computed by simulating $V^*_\rho(s)$, so its distribution is identical to what V^* sends to P. If $\beta = 0$ then z contains uniform independent bits in both $(P(y), \underline{V^*(s)})(N,x)$ and $M_{V^*}(N,x,s)$. If, on the other hand, $\beta = 1$, then z is uniformly distributed over $R'_N(x')$ in both $(P(y), \underline{V^*(s)})(N,x)$ (by condition R3) and $M_{V^*}(N,x,s)$ (by condition T2).

It must now be shown that M_{V^*} runs in expected time $|N|^{O(1)}$. That each iteration of the **repeat**-loop runs in polynomial time follows from the hypothesized running times of A and V^*, and condition T2. The analysis will be completed by showing that, in any iteration, $\beta = \beta'$ with probability $\frac{1}{2}$, and so the expected number of iterations is 2. Because x' is uniformly distributed over dom R'_N no matter what the value of β' is, β' and x' are independent random variables. Hence β' and $\beta = V^*_\rho(s)(N,x,x')$ are independent, so $\Pr(\beta' = 0 \mid \beta) = \Pr(\beta' = 0) = \frac{1}{2}$. From this it follows that $\Pr(\beta' = \beta) = \frac{1}{2}$.

ANALYSIS: That the interactive proof runs in $|N|^{O(1)}$ time follows from the hypothesized running time of A and conditions R3 and T1. □

Corollary 5 *(Goldwasser, Micali, Rackoff [16]):* There is a $(\log N)^{O(1)}$ time perfect zero knowledge interactive proof that the prover can compute a square root of x in $\mathcal{Z}^*_N$, even if the prover is otherwise no more powerful than the verifier.

Corollary 6 *(Chaum* et al. *[10,11]):* There is a $(\log p)^{O(1)}$ time perfect zero knowledge interactive proof that the prover can compute a discrete logarithm base a of x in $\mathcal{Z}_p$, where p is prime, even if the prover is otherwise no more powerful than the verifier.

The zero knowledge interactive proof of Chaum *et al.* [10] is considerably more complex than the one presented here. Chaum *et al.* also show how to extend corollary 6 to the case when p is composite, although abandoning the perfectness of the zero knowledge interactive proof.

Corollary 7 *(Goldreich, Micali, Wigderson [15]):* There is an $n^{O(1)}$ time perfect zero knowledge interactive proof that the prover can compute an isomorphism between two n vertex graphs G and G', even if the prover is otherwise no more powerful than the verifier.

2.3 Applications to Proofs of Language Membership

Theorem 8 below shows that an interactive proof of information possession always yields an interactive proof of membership in a language, namely the language dom R.

Theorem 8 Let R be a binary relation. Let $\pi = (P,V)$ be an interactive proof (zero knowledge interactive proof, perfect zero knowledge interactive proof, respectively) that the prover can compute some y satisfying $(x,y) \in R$. Then π can be converted into an interactive proof (zero knowledge interactive proof, perfect zero knowledge interactive proof, respectively) for membership in the language dom R.

Proof: Given $x \in$ dom R, the prover first computes some $y \in R(x)$ (possibly requiring its great computational power), and then simulates the prover $P(y)$ of π. The proofs of completeness and zero knowledge are straightforward. As for soundness, let P^* be any prover's protocol, and let $x \notin$ dom R. By the soundness of π, for any s

$$\Pr((P^*_\rho(s), V_{\rho'}) \text{ accepts } x \text{ and}$$
$$(x, C_{P^*}((\underline{P^*_\rho(s)}, V_{\rho'})(x))) \notin R) \leq \epsilon.$$

Since $x \notin$ dom R, any output y of C_{P^*} will satisfy $(x,y) \notin R$. Hence, $\Pr((P^*,V) \text{ accepts } x) \leq \epsilon$, as required. □

As a consequence of theorem 8 and corollaries 5-7, the following languages all have polynomial time perfect zero knowledge interactive proofs in the language membership formulation as described in section 1:

1. $\{(N,x) \mid x$ is a quadratic residue modulo $N\}$,
2. $\{(p,a,x) \mid x \in \langle a \rangle_p\}$, where $\langle a \rangle_p$ is the multiplicative subgroup of $\mathcal{Z}^*_p$ generated by a, and
3. $\{(G,G') \mid G \cong G'\}$.

3 A Zero Knowledge Proof for Factorization

The zero knowledge interactive proof that the prover can compute the prime factorization of N exploits the equivalence of the problems of factoring an integer N and computing square roots modulo N. This equivalence is presented in lemmas 9 and 10. Lemma 9 is used to prove the completeness and lemma 10 the soundness of the interactive proof presented in theorem 11.

Lemma 9 There is a probabilistic algorithm that, given the prime factorization of an integer N and any quadratic residue $x \in \mathcal{Z}^*_N$, computes a square root of x modulo N in polynomial expected time.

Proof: Algorithms are given by Adleman, Manders, and Miller [2], Berlekamp [6], and Rabin [18,19]. □

Lemma 10 Let N be odd and γ, δ be constants satisfying $0 < \gamma, \delta \leq 1$. Suppose there is a probabilistic algorithm SQUAREROOT(N,x) that, for a fraction δ of the quadratic residues x in $\mathcal{Z}^*_N$, outputs a single square root of x modulo N with probability γ in expected time $(\log N)^{O(1)}$. Then there is an algorithm that outputs the prime factorization of N in expected time $(\log N)^{O(1)}$.

Proof: This simply generalizes a result of Rabin [18], which assumed N had only 2 prime factors.

CONSTRUCTION: Let SQUAREROOT(N, x) return a square root of x modulo N or a special *don't know* answer in expected time polynomial in $\log N$. The following algorithm then factors N:

$S \leftarrow \{N\}$;
repeat forever
begin
  choose $r \in \mathcal{Z}_N^*$ randomly and uniformly;
  $y \leftarrow$ SQUAREROOT$(r^2 \bmod N)$;
  if $y \neq$ *don't know*
    then begin
      for all $m \in S$ **do**
        $S \leftarrow (S - \{m\}) \cup \{\gcd(m, y - r), \frac{m}{\gcd(m, y - r)}\}$;
      $S \leftarrow S - \{1\}$;
      if all $m \in S$ are prime powers
        then halt and output S
    end
end .

CORRECTNESS: The product of the elements of S is always N, as can be seen by induction on the size of S. Thus, if the algorithm ever halts, S will contain the prime factorization of N.

ANALYSIS: By assumption, the SQUAREROOT oracle returns an answer in expected time $(\log N)^{O(1)}$. The remaining operations in each iteration also run in this expected time [1]. The analysis will be completed by showing that the expected number of iterations until some $m \in S$ is replaced by two proper factors is $\frac{2}{\gamma\delta}$. Hence, the total expected number of iterations is at most $\frac{2}{\gamma\delta}\log_2 N$.

Suppose the prime factorization of N is $N = \Pi_{i=1}^{k} p_i^{e_i}$, where each $e_i \geq 1$, and some $m \in S$ satisfies $p_i p_j \mid m$, where $i \neq j$. Since $r^2 \bmod N$ is a uniformly distributed quadratic residue, the expected number of iterations until SQUAREROOT returns some $y \neq$ *don't know* is $\frac{1}{\gamma\delta}$. For each such y, it will now be shown that, with probability $\frac{1}{2}$, exactly one of p_i and p_j divides $y - r$.

By the Chinese remainder theorem, any square root x of $y^2 \bmod N$ is a solution to the k equivalences $x \equiv b_i y \pmod{p_i^{e_i}}$ for some choice of $(b_1, b_2, \ldots, b_k) \in \{-1, 1\}^k$; the 2^k square roots of $y^2 \bmod N$ are in one-to-one correspondence to the 2^k vectors $(b_1, b_2, \ldots, b_k)$. Since r is a uniformly distributed square root of y^2, $r \equiv b_i y \pmod{p_i^{e_i}}$, where each b_i is chosen uniformly and independently of the others. Hence, $\Pr(p_i^{e_i} \mid (y - r)) \geq \Pr(b_i = 1) = \frac{1}{2}$. But if $b_i = -1$, then $p_i^{e_i} \mid (y + r)$, so $p_i \not\mid (y - r)$, since $\gcd(r, N) = 1$. Hence, $\Pr(p_i \mid (y - r)) = \frac{1}{2}$. It follows that with probability $\frac{1}{2}$ exactly one of p_i and p_j divides $y - r$, so that $\gcd(m, y - r)$ is a proper factor of m. □

Theorem 11 Let N be an integer. There is a $(\log N)^{O(1)}$ time zero knowledge interactive proof that the prover can compute the prime factorization of N, even if the prover is otherwise no more powerful than the verifier.

Proof: CONSTRUCTION: Assume without loss of generality that N is odd (since both prover and verifier can remove factors of 2 from N). On input N, the protocols of the prover and the verifier are as follows:

Prover	*Comm. Channel*	*Verifier*
		choose $r \in \mathcal{Z}_N^*$ randomly and uniformly;
		$x \leftarrow r^2 \bmod N$;
	$\xleftarrow{x}$	

Perfect zero knowledge interactive proof of corollary 5 and lemma 8 that x is a quadratic residue in $\mathcal{Z}_N^*$, with prover's and verifier's roles reversed, and with error probability at most $\frac{1}{N}$. If the prover would reject, it halts the outer proof. If the verifier would halt, it rejects the outer proof. If the prover would accept, the proof continues as below.

Use the factorization of N to compute a square root of x modulo N via lemma 9;

Perfect zero knowledge interactive proof of corollary 5 that the prover can compute a square root of x in $\mathcal{Z}_N^*$, with error probability $\frac{1}{2}$.

CORRECTNESS:

Completeness: Assume that the prover possesses the prime factorization y of N and that both the prover and the verifier follow their protocols. Notice that $(P(y), V)$ accepts if and only if the prover accepts the verifier's proof that x is a quadratic residue and the verifier accepts the prover's proof that it can compute a square root of x. Since the verifier possesses the square root r of x, by the completeness of the first subprotocol the prover will accept with probability 1. Since the prover can compute a square root of x (lemma 9), by the completeness of the second subprotocol the verifier will accept with probability 1. Hence, the overall probability of acceptance is 1.

Soundness: It will be shown that the error probability ϵ is $\frac{7}{8}$. Let P^* be any protocol for the prover. Without loss of generality, assume that each of P^* and V begins each of the two subprotocols with its finite control in the start state, its input tape and work tape rewound, and its read-only communication channel tape empty to the right of its head. The algorithm C_{P^*}, on input (N, s, ρ, m), is constructed from a closely allied algorithm B_{P^*} on input (N, x, s), described now. From lemma 1 and the fact that the first subprotocol is perfect zero knowledge, there is a polynomial expected time

simulator $M^1_{P^*}$ whose output $M^1_{P^*}(N,x,s)$ is identically distributed to the first subprotocol's history $(V, \underline{P^*(s)})(N,x)$. B_{P^*} simulates $M^1_{P^*}$ on input (N,x,s) and, from the history produced, computes the work tape content s_2 of P^* at the beginning of the second subprotocol. It then simulates the second subprotocol $(P^*_\sigma(s_2), V_{\sigma'})$ on input (N,x) and from this simulates the algorithm $C^2_{P^*}$, whose existence is guaranteed by the soundness of the second subprotocol, on input $(\underline{P^*_\sigma(s_2)}, V_{\sigma'})(N,x)$, using random bit strings σ and σ'.

Now suppose that $\Pr((P^*_\rho(s), V_{\rho'})$ accepts $N) > \frac{7}{8}$, for if not the proof is completed. It must be the case that at least $\frac{1}{2}$ of the quadratic residues x transmitted by V each lead to acceptance with probability exceeding $\frac{3}{4}$, for otherwise

$$\Pr((P^*_\rho(s), V_{\rho'}) \text{ accepts } N) < \frac{1}{2}\cdot\frac{3}{4} + \frac{1}{2}\cdot 1 = \frac{7}{8}.$$

In particular, at least $\frac{1}{2}$ of the quadratic residues each cause the verifier to accept the second subprotocol with probability exceeding $\frac{3}{4}$. By the soundness of that subprotocol, at least $\frac{1}{2}$ of the quadratic residues each cause $C^2_{P^*}((\underline{P^*_\sigma(s_2)}, V_{\sigma'})(N,x))$ to be a square root of x modulo N with probability exceeding $\frac{1}{4}$. Thus, B_{P^*} runs in polynomial expected time and, with probability at least $\frac{1}{4}$, outputs a square root of x modulo N for $\frac{1}{2}$ of the quadratic residues x. C_{P^*} is now constructed from B_{P^*} by applying lemma 10.

Zero Knowledge: Let V^* be any polynomial time algorithm for the verifier. Without loss of generality, assume that each of P and V^* begins each of the two subprotocols with its finite control in the start state, its input tape and work tape rewound, and its read-only communication channel tape empty to the right of its head. The simulator M_{V^*} reproduces a verifier's history by directly simulating both verifier and prover up to the end of the first subprotocol. It can simulate V^* since it has access to V^*'s program. It can simulate the prover since, in the subprotocol, the roles are reversed and the prover follows the protocol of the polynomially time bounded verifier of corollary 5. From the verifier's history produced, M_{V^*} computes the work tape content s_2 of V^* at the beginning of the second subprotocol. M_{V^*} then produces the remainder of the verifier's history by simulating the simulator $M^2_{V^*}$, whose existence is guaranteed by the fact that the second subprotocol is zero knowledge, on input (N,x,s_2).

Suppose that the prover accepts the first subprotocol with probability exceeding $\frac{1}{N}$. By the soundness of this subprotocol, x is a quadratic residue modulo N, so that $M^2_{V^*}$ on input (N,x,s_2) runs in polynomial expected time and produces a verifier's history identically distributed to that produced by P and V^* during the second subprotocol. With probability at most $\frac{1}{N}$ the prover will accept the subprotocol even though x is a quadratic nonresidue. Therefore, for any s and any circuit D,

$$|p_D((P(y), \underline{V^*(s)})(N)) - p_D(M_{V^*}(N,s))| \le \frac{1}{N}.$$

The result then follows from lemma 1.

ANALYSIS: The subprotocols run in polynomial expected time, by corollary 5 and lemma 2. The prover's protocol runs in polynomial expected time, by lemma 9. □

4 A Zero Knowledge Proof for Primitivity

Theorem 12 There is a $(\log p)^{O(1)}$ time perfect zero knowledge interactive proof for membership in the language $\{(p,a) \mid p \text{ is prime and } \langle a\rangle_p = \mathcal{Z}^*_p\}$.

Proof: CONSTRUCTION: On inputs p and a, the protocols for the prover and verifier are as follows:

Prover	*Comm. Channel*	*Verifier*
		choose $r \in \mathcal{Z}^*_p$ randomly and uniformly;
	$\leftarrow r$	

Perfect zero knowledge interactive proof of corollary 6 and theorem 8 that $r \in \langle a\rangle_p$, with error probability $\frac{1}{2}$.

The proof of correctness appears in the full paper.

5 A Zero Knowledge Proof for $\mathcal{Z}^*_p - \langle a\rangle_p$

The complements of two of the three languages discussed in section 2.3 previously have been shown to have zero knowledge interactive proofs, namely quadratic nonresiduosity [16] and graph nonisomorphism [15]. The technique used in these two proof has been identified and named "cryptographic capsules" by Benaloh [5]. This section describes a polynomial time zero knowledge interactive proof for the remaining complement, $\mathcal{Z}^*_p - \langle a\rangle_p$. The technique used is the same, and theorem 4 proves useful in demonstrating the correctness of a subprotocol.

Lemma 13 Given a prime p and $a,b,x \in \mathcal{Z}^*_p$, there is a $(\log p)^{O(1)}$ time interactive proof that the prover can compute some $y_a \in \mathcal{Z}_{p-1}$ and $y_b \in \{0,1\}$ satisfying $x \equiv a^{y_a}b^{y_b} \pmod p$, even if the prover is otherwise no more powerful than the verifier. Furthermore, if $b \in \langle a\rangle_p$, it is a perfect zero knowledge interactive proof.

Proof: In the statement of theorem 4, let

$$\begin{aligned}
\mathcal{N} &= \{(p,a,b) \mid p \text{ is prime, and } a,b \in \mathcal{Z}^*_p\},\\
X &= \mathcal{Z}^*_p,\\
Y &= \mathcal{Z}_{p-1} \times \{0,1\},\\
R &= \{(x,(y_a,y_b)) \mid x \equiv a^{y_a}b^{y_b} \pmod p\},\\
X' &= \langle a\rangle_p\langle b\rangle_p \times \langle a\rangle_p\langle b\rangle_p,\\
Y' &= \mathcal{Z}_{p-1}, \text{ and}\\
R' &= \{((x'_1,x'_2),y') \mid x'_1 \equiv a^{y'} \pmod p \\
&\qquad \text{or } x'_2 \equiv a^{y'} \pmod p\}.
\end{aligned}$$

(For notational simplicity, the subscript (p,a,b) is dropped in this proof.) On inputs $(p,a,b) \in \mathcal{N}$, $x \in X$, and

$r \in \{0,1\}^\omega$, the algorithm A first extracts $\beta \in \{0,1\}$ and $s, t \in \mathcal{Z}_{p-1}$ from r in the straightforward way. It then outputs $x' = (xa^s(b^{-1})^\beta, xa^t(b^{-1})^{1-\beta}) \in X'$.

In order to apply theorem 4, it must be established that properties R2, R3, and T1 hold and, if $b \in \langle a \rangle_p$, R1 and T2 hold as well.

R2: If $x'_1 = xa^s(b^{-1})^\beta \equiv a^{y'} \pmod p$, then $y_a = y' - s$ and $y_b = \beta$ satisfy $x \equiv a^{y_a}b^{y_b} \pmod p$. If $x'_2 = xa^t(b^{-1})^{1-\beta} \equiv a^{y'} \pmod p$, then $y_a = y' - t$ and $y_b = 1 - \beta$ satisfy $x \equiv a^{y_a}b^{y_b} \pmod p$. Which of these two cases holds is simple to ascertain.

R3: Suppose $x \equiv a^{y_a}b^{y_b} \pmod p$. Then

$$y' = \begin{cases} y_a + s, & \text{if } y_b = \beta \\ y_a + t, & \text{if } y_b = 1 - \beta \end{cases}$$

satisfies $(x', y') \in R'$. Furthermore, when x' is fixed, the values of y' satisfying $(x', y') \in R'$ differ by multiples of the order of a. Since that order divides $p-1$, the uniform choice of s and t from $\mathcal{Z}_{p-1}$ guarantees that $y_a + s$ and $y_a + t$ are uniformly distributed over $R'(x')$.

T1: This is simply a matter of computing $a^{y'} \bmod p$.

Now suppose that $b \in \langle a \rangle_p$, so

$$X' = \text{dom } R' = \langle a \rangle_p \times \langle a \rangle_p.$$

R1: Since the order of a divides $p-1$, the uniform choice of s and t from $\mathcal{Z}_{p-1}$ guarantees that x' is uniformly distributed over dom R'.

T2: Choose $y', z \in \mathcal{Z}_{p-1}$ randomly, uniformly, and independently. Output one of $((x'_1, x'_2), y')$ or $((x'_2, x'_1), y')$ equiprobably, where $x'_1 = a^{y'} \bmod p$ and $x'_2 = a^z \bmod p$. □

Theorem 14 There is a $(\log p)^{O(1)}$ time zero knowledge interactive proof for membership in the language

$$\{(p, a, b) \mid b \in \mathcal{Z}_p^* - \langle a \rangle_p\}.$$

Proof: CONSTRUCTION: On inputs p, a, and b, the protocols are as follows:

Prover	*Comm. Channel*	*Verifier*
		choose $r \in \mathcal{Z}_{p-1}$ and $\beta \in \{0,1\}$ randomly, uniformly, and independently;
		$z \leftarrow a^r b^\beta \bmod p$;
	$\xleftarrow{z}$	

Interactive proof of lemma 13 that the *verifier* can compute r and β satisfying $z \equiv a^r b^\beta \pmod p$, with error probability at most $\frac{1}{p}$. If the verifier would halt, it rejects the outer proof. If the prover would reject, it halts the outer proof. If the prover would accept, the proof continues as below.

if $z \in \langle a \rangle_p$ **then** $\gamma \leftarrow 0$ **else** $\gamma \leftarrow 1$;		
	$\xrightarrow{\gamma}$	
		if $\beta = \gamma$ **then accept else reject.**

CORRECTNESS:

Completeness: Suppose $b \in \mathcal{Z}_p^* - \langle a \rangle_p$, and both prover and verifier follow their protocols. By the completeness of the interactive proof of lemma 13, the prover will accept the subprotocol with probability 1. Since $b \notin \langle a \rangle_p$, the prover will choose $\gamma = \beta$. Hence, the probability that the verifier accepts is also 1.

Soundness: Suppose $b \in \langle a \rangle_p$, and the verifier follows its protocol. By lemma 13, in this case the subprotocol is a perfect zero knowledge interactive proof. That is, if $\gamma = (V(r, \beta), P^*(s))(p, a, b, z)$ is the random output of the subprotocol on input z, there is a simulator M_{p*} whose output $\gamma = M_{p*}(p, a, b, z, s)$ is identically distributed.

Now z is uniformly distributed over $\langle a \rangle_p$ no matter what value β has, so z and β are independent random variables. Then $\gamma = M_{p*}(p, a, b, z, s)$ and β are also independent, since P^* (as reflected in s) and M_{p*} have no knowledge of β. Thus, $\Pr(\beta = \gamma) = \frac{1}{2}$. Therefore, the probability that the verifier accepts is at most $\frac{1}{2}$.

Zero Knowledge: Let V^* be any polynomial time algorithm for the verifier, s be any string, and $(P, \underline{V^*(s)})(p, a, b)$ be the random verifier's history of $(P, \underline{V^*(s)})$ on input $p, a \in \mathcal{Z}_p^*$ and $b \in \mathcal{Z}_p^* - \langle a \rangle_p$. Let the work tape content of V^* at the beginning of the subprotocol be s_1. The simulator M_{V*} reproduces a verifier's history by directly simulating both verifier and prover up to the end of the subprotocol. It can simulate $V^*(s)$ since it has access to V^*'s program. It can simulate the prover since, in the subprotocol, the roles are reversed and the prover follows the protocol of the polynomially time-bounded verifier of lemma 13.

If the simulated prover accepts the subprotocol, M_{V*} simulates the algorithm C^1_{V*}, whose existence is guaranteed by the soundness of the subprotocol, on input $(\underline{V^*_\rho(s_1)}, P_{\rho'})(p, a, b, z)$. If this outputs (r, β) satisfying $z \equiv a^r b^\beta \pmod p$, M_{V*} then appends β to the verifier's history. Since $b \notin \langle a \rangle_p$, $\beta = \gamma$, so this is identical to what the prover would transmit. By the soundness of the subprotocol, with probability at most $\frac{1}{p}$ the prover will accept the subprotocol, but C^1_{V*} will fail to output r and β satisfying $z \equiv a^r b^\beta \pmod p$. Therefore, for any circuit D,

$$|p_D((P, \underline{V^*(s)})(p, a, b)) - p_D(M_{V*}(p, a, b, s))| \leq \frac{1}{p}.$$

The result then follows from lemma 1.

ANALYSIS: The subprotocol runs in time $(\log p)^{O(1)}$, by lemmas 13 and 3. □

A weak complement of theorem 12 follows as a corollary to theorem 14: if, for each prime p, there is a publicly known generator b of $\mathcal{Z}_p^*$, then the prover can prove that a is not a generator of $\mathcal{Z}_p^*$.

Acknowledgements

We are indebted to Mike Fischer for his healthy skepticism of the underpinnings of this area, and many insights that arose from it. He graciously spent several lengthy sessions helping us secure numerous aspects of those underpinnings. It is also a pleasure to acknowledge the help of Ashok Chandra, Ron Fagin, Joan Feigenbaum, Oded Goldreich, Neil Immerman, Silvio Micali, Yair Oren, Charlie Rackoff, Prabhakar Raghavan, and Prasoon Tiwari.

References

[1] L. M. Adleman and M. A. Huang, "Recognizing Primes in Random Polynomial Time", *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, New York, N.Y., May 1987, 462-469.

[2] L. M. Adleman, K. Manders, and G. Miller, "On Taking Roots in Finite Fields", *18th Annual Symposium on Foundations of Computer Science*, Providence, Rhode Island, October-November 1977, 175-178.

[3] D. Angluin and D. Lichtenstein, "Provable Security of Cryptosystems: a Survey", Technical Report TR-288, Yale University, October 1983.

[4] P. Beame, "Minimum Knowledge Proofs in Elementary Group Theory", preprint, 1987.

[5] J. C. Benaloh, "Cryptographic Capsules: a Disjunctive Primitive for Interactive Protocols", *Crypto 86*, Abstract #21, Santa Barbara, California, August 1986.

[6] E. Berlekamp, "Factoring Polynomials over Large Finite Fields", *Mathematics of Computation*, vol. 24, 1970, 713-735.

[7] G. Brassard and C. Crépeau, "Non-Transitive Transfer of Confidence: A Perfect Zero-Knowledge Interactive Protocol for SAT and Beyond", *27th Annual Symposium on Foundations of Computer Science*, Toronto, Ontario, October 1986, 188-195.

[8] G. Brassard, C. Crépeau, and J.-M. Robert, "All-or-Nothing Disclosure of Secrets", *Crypto 86*, Abstract #23, Santa Barbara, California, August 1986.

[9] D. Chaum, "Demonstrating that a Public Predicate can be Satisfied Without Revealing Any Information About How", *Crypto 86*, Abstract #19, Santa Barbara, California, August 1986.

[10] D. Chaum, J.-H. Evertse, J. van de Graaf, and R. Peralta, "Demonstrating Possession of a Discrete Logarithm without Revealing It", *Crypto 86*, Abstract #20, Santa Barbara, California, August 1986.

[11] D. Chaum and J. van de Graaf, "An Improved Protocol for Demonstrating Possession of a Discrete Logarithm and Some Generalizations", *Eurocrypt 87*, Amsterdam, The Netherlands, April 1987, IV-15 to IV-21.

[12] C. Crépeau, "A Zero-Knowledge Poker Protocol that Achieves Confidentiality of the Players' Strategy *or* How to Achieve an Electronic Poker Face", *Crypto 86*, Abstract #24, Santa Barbara, California, August 1986.

[13] U. Feige, A. Fiat, and A. Shamir, "Zero Knowledge Proofs of Identity", *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, New York, N.Y., May 1987, 210-217.

[14] M. Fischer, S. Micali, and C. Rackoff, "A Secure Protocol for the Oblivious Transfer", *Eurocrypt 84*.

[15] O. Goldreich, S. Micali, and A. Wigderson, "Proofs that Yield Nothing But their Validity and a Methodology of Cryptographic Protocol Design", *27th Annual Symposium on Foundations of Computer Science*, Toronto, Ontario, October 1986, 174-187.

[16] S. Goldwasser, S. Micali, and C. Rackoff, "The Knowledge Complexity of Interactive Proof-Systems", *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, Providence, Rhode Island, May 1985, 291-304.

[17] Y. Oren, "On the Cunning Power of Cheating Verifiers: Some Observations about Zero Knowledge Proofs", these Proceedings.

[18] M. O. Rabin, "Digitalized Signatures and Public-Key Functions as Intractable as Factorization", Technical Report MIT/LCS/TR-212, M.I.T., January 1979.

[19] M. O. Rabin, "Probabilistic Algorithms in Finite Fields" *SIAM Journal on Computing*, vol. 9 (1980), 273-280.