Four Rounds Are Optimal for Zero Knowledge

Johan Wallén

October 10, 2001

Motivation

If one-way functions exists, all languages in $\mathbf{IP}$ ($= \mathbf{PSPACE}$) have computational zero-knowledge proofs.

Are one-way functions also **necessary**?

- Increases our understanding of the relationships between $\mathbf{CZK}$ and other cryptographic primitives.
- Shows that zero knowledge proofs build from any one-way function are **optimal**: they use the weakest assumptions possible.

The answer is positive: One-way functions are indeed essential.

Motivation continued

Zero knowledge proofs and arguments are fundamental building blocks for larger cryptographic protocols. In these applications, the **round complexity** is important.

What is the optimal number of rounds?

- Any three round protocol is trivial. If a language L has a three round $\mathbf{CZK}$ proof or argument, $L \in \mathbf{BPP}$.
- Four round $\mathbf{CZK}$ **arguments** built from any one-way function exist for all of $\mathbf{NP}$.

Interactive proofs and arguments

Recall that an interactive proof system (V, P) for L consists of

- A probabilistic polynomial time verifier V.
- An unrestricted prover P.

For common input x, P and V interacts by exchanging at most polynomially many messages of polynomial length. The protocol is ended by V either accepting or rejecting x.

The **transcript** $\langle V, P\rangle(x)$ consists of x and all exchanged messages.

The number of **rounds** is the total number of messages exchanged.

We require that the error probability is negligible: For some negligible function ϵ and all sufficiently large x,

Completeness: $x \in L \Longrightarrow \mathrm{Pr}(V \text{ accepts } x) > 1 - \epsilon(|x|)$.

Soundness: For all provers P', $x \notin L \Longrightarrow \mathrm{Pr}(V \text{ accepts } x) < \epsilon(|x|)$.

If P and all provers P' in the soundness condition are restricted to probabilistic polynomial time, we get an interactive **argument**.

Computational indistinguishability

Let $X = \{X_n\}_{n \in \mathbb{N}}$, and $Y = \{Y_n\}_{n \in \mathbb{N}}$ be probability ensembles, where each X_i, Y_i range over strings of length n.

The ensembles are said to be **computationally indistinguishable**, written as $X \overset{c}{\sim} Y$, if for every probabilistic polynomial time algorithm A,

$$d_A(n) = |\Pr(A(X_n) = \top) - \Pr(A(Y_n) = \top)|$$

is a negligible function of n.

Computational zero knowledge

An interactive proof (or argument) (V, P) is a **black-box computational zero knowledge proof (argument)** for L, if there is a generic oracle calling algorithm S (the **simulator**) running in expected polynomial time, such that for all verifiers V',

$$\{\langle V', P\rangle(x)\}_{x \in L} \overset{c}{\sim} \{S^{V'}(x)\}_{x \in L} .$$

Four rounds are needed

The main result covered in the survey is

Theorem 1 (Goldreich and Krawczyk, 1996). A language L has a three round black-box computational zero knowledge proof or argument if and only if $L \in \mathbf{BPP}$.

Proof sketch. Let (P, V) be a three round zero knowledge proof or argument for L, and let S be the black-box simulator. We will derive a $\mathbf{BPP}$ algorithm deciding L. The algorithm A will use S as a subroutine to distinguish between $x \in L$ and $x \notin L$.

Let x be the input, and $n = |x|$.

The protocol: First, P sends α to V. V responds with β. Finally, P sends γ to V, and the protocol is terminated by V accepting or rejecting.

β is a function of x, α, and V's random string r: $\beta = \beta_V(x, r, \alpha)$. Let ρ_V be the predicate used by V in order to accept or reject. For convenience, we assume that this is a deterministic function: $\rho_V(x, \alpha, \beta, \gamma)$.

The simulation: Let V^* be any verifier, and write $S = S^{V^*}$. The simulation process consist of several tries. In each try, the simulator feeds V^* with the arguments y (the input) and α (the prover's message). Let $\beta = V^*(y, \alpha)$ be the response of V^* in this try. We can assume the following

- S always outputs a transcript of the form $(x, \alpha, \beta, \gamma)$, where (x, α) are fed to V^* in some try, and $\beta = V^*(x, \alpha)$ is the response in this try.
- S runs in polynomial worst-case time. Let $t(n)$ be a polynomial bound on the number of tries by S for inputs of length n.

The $\mathbf{BPP}$ algorithm A: A crucial observation is that the simulation process is completely determined by x, the random string of S, and the responses of the verifier.

The algorithm A operates as follows. First, it picks a random string R_S for S and $t = t(n)$ random strings $r^1, \ldots, r^t$. The simulator S is used as a subroutine with x as input and R_S as random string.

Whenever S invokes the verifier with arguments y and α, we find the first unused r^i and respond with $\beta^i = \beta_V(x, r^i, \alpha)$. If S invokes the verifier with the same α more than once, we return the same β as the last time, regardless of the value of y.

Note that the conversation is uniquely specified by x, R_S, and $r^1, \ldots, r^t$. Thus, we can define the function $\mathrm{conv}_S(x, R_S, r^1, \ldots, r^t)$ to be the resulting transcript.

Let α^i be the ith different value of α output by S during the simulation. Clearly, α is a function of x, R_S, and $r^1, \ldots, r^{i-1}$, $\alpha^i = \alpha_S(x, R_S, r^1, \ldots, r^{i-1})$.

By our assumptions, if $\mathrm{conv}_S(x, R_S, r^1, \ldots, r^t) = (x, \alpha, \beta, \gamma)$, S tried (x, α) in some try, Thus, there is an i such that $\beta = \beta_V(x, r^i, \alpha)$.

Definition 2. A vector $(x, R_S, r^1, \ldots, r^t)$ is S**-good**, if $\mathrm{conv}_S(x, R_S, r^1, \ldots, r^t) = (x, \alpha, \beta, \gamma)$ and $\rho_V(x, \alpha, \beta, \gamma) = \top$.

The algorithm A accepts x if and only if $(x, R_S, r^1, \ldots, r^t)$ is S-good.

Since S and V runs in polynomial time, A runs in polynomial time. By the following two technical lemmas, A can distinguish between $x \in L$ and $x \notin L$ with small enough error probability. Thus, $L \in \mathbf{BPP}$. $\square$

Lemma 3. For $x \notin L$, only a negligible fraction of the vectors $(x, R_S, r^1, \ldots, r^t)$ are S-good.

Lemma 4. For $x \in L$, a non-negligible fraction of the vectors $(x, R_S, r^1, \ldots, r^t)$ are S-good.

One-way functions are essential

Another important result briefly discussed in the survey is

Theorem 5 (Ostrovsky and Wigderson, 1993). Under the assumption that auxiliary input one-way functions does not exist, if L has a $\mathbf{CZK}$ proof, then $L \in \mathbf{BPP}$.

An optimal zero knowledge argument

Given these result, it is natural to ask if there are four round computational zero knowledge proofs or arguments build from any one-way function for languages outside $\mathbf{BPP}$.

A partial answer is discussed in the survey: a four round $\mathbf{CZK}$ **argument** built from any one-way function is sketched for the $\mathbf{NP}$ complete SAT.

The protocol [Bellare, Jakobsson, and Yung, 1997] uses an atomic commitment based four round zero knowledge proof with error probability 0.5. The basic idea is to run several independent copies of the atomic protocol in parallel. With some care, this gives a four round $\mathbf{CZK}$ argument for SAT with negligible error probability.