Concurrent/Resettable Zero-Knowledge Protocols for NP in the Public-Key Model

Silvio Micali and Leonid Reyzin[*]

August 18, 2000

Abstract

We propose a four-round protocol for concurrent and resettable zero-knowledge arguments for any language in NP, assuming the verifier has a pre-registered public-key. We also propose a three-round protocol with an additional timing assumption.

1 Four-round protocol in the public-key model

We propose the following 4-round protocol for resettable zero-knowledge arguments for NP-languages in the public-key model (see [CGGM00] for definitions).

Without loss of generality, we assume that the prover is trying to convince the verifier of 3-colorability of a graph G.

Prior to any interaction with the prover about G, the verifier has registered in the public file the public key, PK, of perfectly-committing encryption scheme E, for which the verifier knows the corresponding secret key SK.

We assume that there is a three-round proof-of-knowledge protocol for knowledge of SK. We do not need the protocol to be zero-knowledge, but do need it to be simulatable in time about 2^k if the knowledge-error is 2^{-k} (this is similar to the protocol used in [CGGM00]).

The protocol is as follows.

1. The verifier sends the prover the first message (commitment) of the three-round proof-of-knowledge for SK, as well as an encryption $E_{SK}(\sigma)$ of a random string σ.

2. The prover the sends the verifier a challenge for the three-round proof-of-knowledge for SK, together with a random string τ (to make the protocol resettable, both the challenge and the string τ should be computed as a pseudo-random function of the verifier's first message).

[*] Supported by the National Science Foundation Graduate Research Fellowship and by a grant from the NTT corporation.

3. The verifier sends the prover the response for the challenge (thus completing the proof of knowledge), together with τ and the random coins used to ecnrypt τ (thus decommitting the encryption of τ).

4. The prover checks the correctness of the response and the decommitment, computes $R = \sigma \oplus \tau$. Using the string R as the "shared random string," the prover then computes and sends to the verifier a non-interactive zero-knowledge proof [BFM88, BDMP91] that G is 3-colorable.[1]

4. The verifier computes $R = \sigma \oplus \tau$ and accepts if and only if the proof received from the prover is valid with respect to R.

2 Three-round protocol in the public-key model with timing

We propose the following 3-round protocol for resettable zero-knowledge arguments for NP-languages in the public-key model (see [CGGM00] for definitions). We have to assume that the prover and the verifier have timers maximum difference bounded by b.[2] The verifier also keeps a table of entries whose is limited as a function of b.

We assume familiarity with the notions of a pseudo-random function (PRF) [GGM86] and of a verifiable random function (VRF) [MRV99].

Without loss of generality, we assume that the prover is trying to convince the verifier of 3-colorability of a graph G.

Prior to any interaction with the prover about G, the verifier has registered in the public file the public key, PK, of a VRF, F, for which the verifier knows the corresponding secret key SK. (Thus $F(\cdot) = F(SK, \cdot)$.)

The protocol is as follows.

1. The prover looks up PK in the public file[3], randomly selects a secret seed s for a PRF f (i.e., $f(\cdot) = f_s(\cdot)$), where f produces suitably long outputs), and sends the verifier the string $\sigma = f(G, t, 3 - col)$ together with the prover's local time t.

2. The verifier checks that its own local time is between $t - b$ and $t + b$ (otherwise, it aborts). The verifier then checks its table to see if the entry (G, t) exists in it. If so, it aborts. If not, it adds (G, t) to the table, and removes any entries (G', t') from the table for which $t' + b$ is less than the verifier's current time. Then the verifier computes and sends to the prover

[1] Note that here we only need the simpler version of their protocols, in which the shared random string is used to prove only a single theorem.

[2] If the timers differ by more than b, then completeness (but not zero-knowledge nor soundness) is impaired

[3] If the verifier's identity is unknown to the prover, one can add an extra round where the verifier sends the public key to the prover, and the prover checks that it is indeed in the public file.

the strings, $\tau = F(G, t)$ (of the same length as σ) and π, the VRF's proof that indeed $\tau = F(G, t)$.

3. The prover checks the correctness of π relative to τ, PK, G, and t, and then computes $R = \sigma \oplus \tau$. Using the string R as the "shared random string," the prover then computes and sends to the verifier a non-interactive zero-knowledge proof [BFM88, BDMP91] that G is 3-colorable.[4]

4. The verifier computes $R = \sigma \oplus \tau$ and accepts if and only if the proof received from the prover is valid with respect to R.

Remark: The protocol can be improved if the verifier is allowed more storage.

References

[BDMP91] Manuel Blum, Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. Noninteractive zero-knowledge. *SIAM Journal on Computing*, 20(6):1084–1118, December 1991.

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, pages 103–112, Chicago, Illinois, 2–4 May 1988.

[CGGM00] Ran Canetti, Oded Goldreich, Shafi Goldwasser, and Silvio Micali. Resettable zero-knowledge. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, Portland, Oregon, 21–23 May 2000. Updated version available at the Cryptology ePrint Archive, record 1999/022, `http://eprint.iacr.org/`.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, October 1986.

[MRV99] Silvio Micali, Michael Rabin, and Salil Vadhan. Verifiable random functions. In *40th Annual Symposium on Foundations of Computer Science*, pages 120–130, New York, October 1999. IEEE.

[4] Note that here we only need the simpler version of their protocols, in which the shared random string is used to prove only a single theorem.