1 Zero-knowledge Proofs, Continued

The zero-knowledge (ZK) proof of graph Hamiltonicity from Lecture 16 had soundness error 1/2. We can reduce this soundness error through (sequential) repetition. Namely, we can repeat the zero-knowledge proof k times to give us a soundness error of 2^{-k}. This is in fact a consequence of a general theorem:

Theorem 1 *Let* Π *be a ZK proof with auxiliary inputs. Then sequential repetition of* Π *is also ZK.*

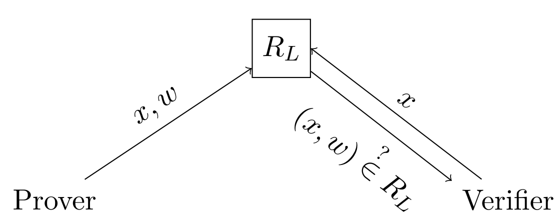
See [Gol01, §4.3.4] for the detailed proof. Note that the above only holds for *sequential* repetition. It is unknown whether *parallel* repetition holds for the graph Hamiltonicity protocol described in Lecture 16.

Thus, using sequential repetition we have a ZK proof for all of $\mathcal{NP}$ with negligible soundness error. However, this protocol is not constant-round. There *do* exist constant-round ZK proofs, but their constructions require assumptions stronger than one-way functions [Gol01, §4.9]. Note, however, that for the case of ZK *arguments*, there exist constant-round protocols from one-way functions [FS89]. Finally, note that it is unknown whether there exist *three-round* ZK proofs with negligible soundness error.

2 Proofs of Knowledge

A proof of knowledge (PoK) is similar to a ZK proof, except the simulator can "extract" a witness from any prover who can convince the verifier that some input x is in the language. Thus, proofs of knowledge demonstrate that the prover "knows" a witness w such that $(x, w) \in R_L$.

Consider the following functionality:



We can view ZK proofs as providing security in the above functionality against a malicious *verifier*, and we can view PoKs as providing security against a malicious *prover*. Thus, a ZKPoK realizes the above functionality.

2.1 Zero-knowledge Proofs of Knowledge Under Sequential Repetition

We now show that sequential repetition of the ZK proof Π for graph Hamiltonicity shown in Lecture 16 is a ZKPoK. We have already shown that the protocol is ZK, and by Theorem 1 we know that sequential repetition holds for the ZK property. Thus, to complete the proof we need to demonstrate a *knowledge extractor* K which runs in polynomial time and extracts a witness from an arbitrary prover in the case that the verifier accepts. Namely:

Claim 2 *If* $\Pr[\langle P^*, V\rangle = 1] = \varepsilon(k) > 2^{-k}$*, then there exists some knowledge extractor* K *which extracts a valid witness with probability* $\varepsilon(k)$*.*

Proof We construct knowledge extractor K as follows:

Knowledge Extractor K

1. Run execution with the prover P^*, behaving as an honest verifier V.
2. If V would reject then halt, outputting $\perp$.
3. Otherwise, extract a witness by doing the following:
   (a) Let $b_1, \ldots, b_k$ be the challenges sent by V in the k executions of Π. Then, repeat the following for $i \in [k]$, rewinding after each iteration: Use $b_1, \ldots, b_{i-1}, \overline{b_i}$ as the challenges to P^*. If an iteration succeeds, K knows both π, $\pi(G)$, and π(Hamiltonian cycle in G), and can thus extract a witness.

We claim that K extracts a witness with probability $\varepsilon(k)$. Note that, as $\varepsilon(k) > 2^{-k}$, for *any* challenge bitstring $b_1 \cdots b_k$ for which execution succeeds, there must exist some other challenge bitstring $b'_1 \cdots b'_k$ for which extraction would also succeed. Let $b_1 \cdots b_{i-1}$ denote the longest common prefix between these two strings. If K executes P^* for both b_i and $\overline{b_i}$, it learns both a permutation π of G as well as a permutation of a Hamiltonian cycle, and thus it can extract the desired witness. Thus, as long as the first set of challenges $b_1 \cdots b_k$ succeeds, K extracts a witness with probability 1. Noting that the probability of succeeding in this first step is $\varepsilon(k)$ completes the proof. ■

The simulator $\mathcal{S}$ for P^* (in the ZKPoK functionality) works as follows:

Simulator $\mathcal{S}(x)$

1. $\mathcal{S}$ runs the execution with P^*, acting as an honest verifier V.
2. If V would reject, $\mathcal{S}$ sends a dummy witness to R_L.
3. If V would accept, run the knowledge extractor K to extract a witness.

If $\Pr[\langle P^*, V\rangle = 1] \leq 2^{-k}$, then when execution succeeds $\mathcal{S}$ fails to extract, but this only happens with negligible probability. Now, if $\Pr[\langle P^*, V\rangle = 1] = \varepsilon(k) > 2^{-k}$, then the distributions in the real and ideal worlds are *identical*, since whenever P^* would have succeeded in the real world, K succeeds in extracting a witness in the ideal world.

2.2 Proofs of Knowledge Under Parallel Repetition

We now show that the same protocol run in parallel is a PoK. The simulator $\mathcal{S}$ for P^* works as follows:

Simulator $\mathcal{S}(x)$

1. $\mathcal{S}$ interacts with P^* just like an honest verifier V would.
2. If V would reject, $\mathcal{S}$ sends a dummy witness to R_L.
3. If V would accept, $\mathcal{S}$ does the following in parallel:
   (a) Rewind P^* and send another random challenge different from the original one until finding a successful execution.
   (b) If $\mathcal{S}$ fails to find a second accepting challenge after 2^k steps, $\mathcal{S}$ enumerates all possible challenges in parallel, trying random challenges. (This ensures that if at least one challenge is answered correctly, then two challenges will always be found.)

Claim 3 *$\mathcal{S}$ as defined above runs in expected polynomial time.*

Proof If P^* convinces V with some probability $\leq 2^{-k}$, then the expected running time is $\leq 2^{-k} \cdot 2^k \cdot \mathsf{poly}(k) = \mathsf{poly}(k)$. If P^* convinces V with some probability $N/2^k > 2^{-k}$, then the expected running time is $\leq N/2^k \cdot (2^k/(N-1)) \cdot \mathsf{poly}(k) < 2 \cdot \mathsf{poly}(k)$. ■

Claim 4 *If $\Pr[\langle P^*, V\rangle = 1] = \varepsilon(k) > 2^{-k}$, then extraction always succeeds.*

Proof Denote the two challenge bitstrings sent by $\mathcal{S}$ by $b_1 \dots b_k$ and $b'_1 \dots b'_k$, and let $i \in \{1, \dots, k\}$ be an index such that $b_i = \overline{b'_i}$. Thus, applying the same idea as in the proof of sequential repetition shows that we can extract a witness. ■

References

[FS89] Uriel Feige and Adi Shamir. Zero knowledge proofs of knowledge in two rounds. In Gilles Brassard, editor, *Advances in Cryptology – CRYPTO'89*, volume 435 of *Lecture Notes in Computer Science*, pages 526–544, Santa Barbara, CA, USA, August 20–24, 1989. Springer, Berlin, Germany.

[Gol01] Oded Goldreich. *Foundations of Cryptography: Volume 1, Basic Tools*. Cambridge University Press, 2001.