Lecture 9: Zero-Knowledge Proofs

Lecturer: Kurt Mehlhorn & He Sun

A zero-knowledge proof is an interactive protocol (game) between two parties, a prover and a verifier. Both parties have a statement as input that may or may not be true, for example, the description of a graph G and the statement that G is 3-colorable. The goal of the prover is to convince the verifier that the statement is true, and, at the same time, make sure that no information other than the truth of the statement is leaked through the protocol.

1 Brief History and Reference

The concept of zero-knowledge was introduced by Goldwasser, Micali and Rackoff [4]. The early version of their paper has existed as early as in 1982, and were rejected three times from major conferences (FOCS '83, STOC '84, FOCS '84) before appearing in STOC '85. Goldreich, Micali and Wigderson [3] showed how to construct zero-knowledge systems from any **NP**-set.

Goldreich's article *A short Tutorial of Zero-Knowledge* is an excellent reference for studying zero-knowledge. The earliest version of the article appeared in 2002 titled *Zero-Knowledge twenty years after its invention* [2]. In addition, Aaronson [1] discusses zero-knowledge proofs from a Philosophical point of view.

2 A Very Simple ZKP

The following example is from Wikipedia. Assume that Peggy has uncovered the secret word used to open a magic door in a cave. The cave is shaped like a circle, with the entrance on one side and the magic door blocking the opposite side. Victor wants to know whether Peggy knows the secret word; but Peggy, being a very private person, does not want to reveal the fact of her knowledge to the world in general.

They label the left and right paths from the entrance A and B. First, Victor waits outside the cave as Peggy goes in. Peggy takes either path A or B; Victor is not allowed to see which path she takes. Then, Victor enters the cave and shouts the name of the path he wants her to use to return, either A or B, chosen at random. Providing she really does know the magic word, this is easy: she opens the door, if necessary, and returns along the desired path.

However, suppose she did not know the word. Then, she would only be able to return by the named path if Victor were to give the name of the same path that she had entered by. Since Victor would choose A or B at random, she would have a 50% chance of guessing correctly. If they were to repeat this trick many times, say 20 times in a row, her chance of successfully anticipating all of Victor's requests would become vanishingly small (about one in 1.05 million).

Thus, if Peggy repeatedly appears at the exit Victor names, he can conclude that it is very probable that Peggy does in fact know the secret word.

However, even if Victor is wearing a hidden camera that records the whole transaction, the only thing the camera will record is Victor shouting "A!" and Peggy appearing at A; Victor shouting "B!" and Peggy appearing at B. A recording of this type would be trivial for any two people to fake (requiring only that Peggy and Victor agree beforehand on the sequence of A's and B's that Victor will shout). Such a recording will certainly never be convincing to anyone but the original participants. In fact, even a person who was present as an observer at the original experiment would be unconvinced, since Victor and Peggy might have orchestrated the whole "experiment" from start to finish.

3 A ZKP for Graph Non-Isomorphism

Definition 1 (Interactive Proof Systems and **IP**)**.** *An interactive proof system for a set S is a two-party game, between a verifier executing a probabilistic polynomial-time strategy (denoted by V) and a prover which executes a computational unbounded strategy (denoted by P) satisfying*

- *(Completeness) for every input $x \in S$ for which the correct answer is YES, there is a witness w such that $P(x, w) \leftrightarrows V(x)$ interaction ends with V accepting with probability one.*
- *(Soundness) for every input $x \notin S$ for which answer is NO, for algorithm $P^\star$ of arbitrary complexity $P^\star(x, w) \leftrightarrows V(x)$ interaction ends with V rejecting with probability at least half (or at least $1 - \frac{1}{2^k}$ if protocol repeated k times).*

The class of problems having interactive proof systems is denoted by **IP**.

Definition 2 (Graph Isomorphism)**.** *Two graphs $G = (V_1, E_1)$ and $H = (V_2, E_2)$ with $|V_1| = |V_2| = n$ are called* isomorphic *if and only if there exists a permutation $\pi \in S_n$ such that $\{u, v\} \in E_1$ iff $\{\pi(u), \pi(v)\} \in E_2$*

It is believed that this problem is not **NP**-complete. However any algorithm that runs faster than $O(2^{\sqrt{n}})$ is not known.

We first look at a zero-knowledge proof for graph non-isomorphism. We have two graphs G_1 and G_2 as an input of a game, and there are two players in the game exchanging information. The goal of one player, the prover, is to convince the other player, the verifier, that these two graphs G_1 and G_2 are not isomorphic. The zero-knowledge protocol is as follows:

A ZKP for Graph Non-Isomorphism

- Common Input: Two graphs $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$. The prover wants to convince the verifier that they are not isomorphic.
- The verifier picks a random $b \in \{1, 2\}$ and a permutation $\pi : V \mapsto V$ and sends $G = \pi(G_b)$ to the prover.
- The prover finds the bit $a \in \{1, 2\}$ such that G_a and G are isomorphic and sends a to the verifier.
- The verifier checks that $a = b$, and, if so, accepts.

Remark:

- The prover is computational unbounded, and the goal of the prover is to convince the verifier of the validity of some assertion.
- The verifier is computational bounded, and is a probabilistic polynomial-time algorithm.

Theorem 3. *Let P be the prover algorithm and V be the verifier algorithm in the above protocol. Then*

1. *Completeness: If G_1 and G_2 are not isomorphic, then the interaction ends with the verifier accepting with probability 1.*
2. *Soundness: If G_1 and G_2 are isomorphic, then for every alternative prover strategy P^*, of arbitrary complexity, the interaction ends with the verifier accepting with probability $1/2$.*

The first part of the theorem is true as for every permutation $\pi(G_1)$ is not isomorphic to G_2 and similarly for every permutation $\pi(G_2)$ is not isomorphic to G_1, therefore if G_1 and G_2 are not isomorphic no relabeling of G_1 can make it isomorphic to G_2. Since the prover is computational unbounded, he can always find out which graph the verifier has started from and therefore the prover always gives the right answer.

The second part of the theorem is true as there exists a permutation $\pi^\star$ such that $\pi^\star(G_2) = G_1$. Then if verifier picks a random permutation π_R then the distribution we obtain by $\pi_R(\pi^\star(G_2))$ and the distribution $\pi_R(G_1)$ are exactly the same as both are just random relabelling of, say, G_1. Therefore here the answer of the prover is independent on b and the prover succeeds with probability half. This probability of $1/2$ can be reduced to 2^{-k} by repeating the protocol k times. The reason why the verifier is convinced is because the prover would need to do something that is information theoretically impossible if the graphs are isomorphic. Therefore, it is not the answers themselves that convince the verifier but the fact that prover can give those answers without knowing the isomorphism.

4 A ZKP for Graph Isomorphism

Suppose now that the prover wants to prove that two given graphs G_1, G_2 are isomorphic, and that he, in fact, knows an isomorphism. We shall present a protocol for this problem in which both the prover and the verifier are efficient.

A ZKP for Graph Isomorphism

- Common input: two graphs $G_1 = (V, E_1)$, $G_2 = (V, E_2)$. The prover knows the permutation $\pi^\star$ such that $\pi^\star(G_1) = G_2$. The prover wants to convince the verifier that the graphs are isomorphic without showing the permutation $\pi^\star$.
- The prover picks a random permutation $\pi_R : V \to V$ and sends the graph $G \triangleq \pi_R(G_1)$
- The verifier picks at random $b \in \{1, 2\}$ and sends b to the prover.
- The prover sends back π_R if $b = 1$, and $\pi_R \circ (\pi^\star)^{-1}$ otherwise.
- The verifier checks that the permutation π received at the previous round is such that $\pi(G_b) = G$, and accepts if so.

Theorem 4. *Let P be the prover algorithm and V be the verifier algorithm in the above protocol. Then*

1. *Completeness: If G_1, G_2 are isomorphic, then the interaction ends with the verifier accepting with probability* 1.
2. *Soundness: If G_1, G_2 are not isomorphic, then for every alternative prover strategy P^* of arbitrary complexity, the interaction ends with the verifier accepting with probability* 1/2.

By construction, the first statement of Theorem 4 holds. Now we see what happens if G_1 and G_2 are not isomorphic and the prover is not following the protocol and is trying to cheat a verifier? Since in the first round the prover sends a graph G, and G_1 and G_2 are not isomorphic, then G can not be isomorphic to both G_1 and G_2. So in second round with probability at least half the verifier is going to pick G_b that is not isomorphic to G. When this happens there is nothing that the prover can send in the third round to make the verifier accept, since the verifier accepts only if what prover sends in the third round is the isomorphism between G and G_b. Hence the prover will fail with probability a half at each round and if we do the same protocol for several rounds the prover will be able to cheat only with exponentially small probability.

5 ZKPs for NP-Complete Problems

We introduce the notion of *commitment schemes*. Commitment schemes are digital analogies of sealed envelopes (or, better, locked boxes). Sending a commitment means sending a string that binds the sender to a unique value without revealing this value to the receiver (as when getting a locked box). De-committing to the value means sending some auxiliary information that allows to read the uniquely committed value. By assuming the existence of commitment schemes, there exists zero-knowledge proofs of membership in any **NP** set. Let us look at the following problem for example.

Problem 5 (3-coloring problem)**.** *Let $G = (V, E)$ be a graph with vertex set $V = \{1, \ldots, n\}$. The 3-coloring problem asks if there is a coloring function $\phi : V \mapsto \{1, 2, 3\}$ such that $\phi(u) \neq \phi(v)$ for any edge $\{u, v\} \in E$.*

A ZKP for 3-Coloring Problem

- The prover selects uniformly a permutation π over $\{1, 2, 3\}$. For $i = 1$ to n, send the verifier a commitment to the value $\pi(\phi(i))$.
- The verifier select uniformly an edge $e \in E$ and send it to the prover.
- Upon receiving $e = (i, j) \in E$, the prover decommits to the ith and jth values sent in the first step.
- The verifier checks whether or not the decommitted values are different elements of $\{1, 2, 3\}$ and whether or not they match the commitments received in the first step.

We can repeat the protocol $t \cdot |E|$ times so that the soundness error probability is bounded by $\exp(-t)$.

By using the standard Karp-reduction to 3-Colorability, the protocol above can be used for constructing zero-knowledge proofs for any problem in **NP**.

References

[1] Scott Aaronson. Why philosophers should care about computational complexity. *CoRR*, abs/1108.1791, 2011.

[2] Oded Goldreich. Zero-knowledge twenty years after its invention. *Electronic Colloquium on Computational Complexity (ECCC)*, (063), 2002.

[3] 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 1991.

[4] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *17th Annual ACM Symposium on Theory of Computing (STOC'85)*, pages 291–304, 1985.