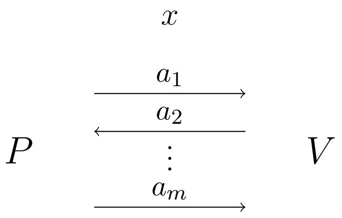
CS 6810: Theory of Computing **Fall 2021**

Lecture 19: Nov 2, 2021

Lecturer: Eshan Chattopadhyay *Scribe: Matthew Eichhorn*

1 Zero-Knowledge Proofs (ZKP)

Over the past few lectures, we have focused on the paradigm of *Interactive Proofs.* Here, there is a computationally-constrained verifier V and an unconstrained prover P. Upon receiving an input string x, P and V pass a series of message strings $a_1, \dots, a_m$, through which P attempts to convince V whether x is in the language L.



We've shown that without the verifier's access to randomness, this exchange of information provides no additional computational power than NP. As such, we've focused our attention on the case where the verifier has access to a private source of randomness. In this randomized setting, we want our verifier V to utilize a communication protocol that obeys the following properties:

Completeness: $x \in L \implies \exists$ a prover P for which $\Pr\Big[\text{out}_V\langle P, V\rangle(x) = 1\Big] = 1.$

Soundness: $x \notin L \implies \forall$ provers P, $\Pr\Big[\text{out}_V\langle P, V\rangle(x) = 1\Big] < \frac{1}{3}.$

Here, the probability is computed over the randomness of V. Completeness tells us that there is some prover, namely the honest prover, that is able to convince V that strings are in the language with high probability.

Soundness guards the verifier against a dishonest prover by ensuring that they will only be wrongly convinced that a string $x \notin L$ should be accepted with low probability.

In the setting of zero-knowledge proofs, we add a third property, the zero-knowledge property. By choosing a protocol that conforms to the zero-knowledge property, V provides a safeguard to P that they will not need to reveal any privileged information during the communication. In some domains, such as cryptography, this is a consequential guarantee: an agent with some private information may wish to demonstrate that they possess this knowledge without revealing it. A zero-knowledge proof allows the prover to provide such a certificate without compromising the information.

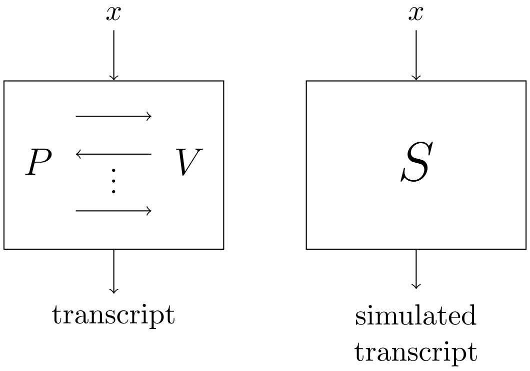
We consider how to formally define this zero-knowledge property. We do this by means of a simulator S. S is a probabilistic (expected) polynomial time algorithm. It can interact with V, but does not have access to its private randomness. We define the zero-knowledge property as follows.

Zero-Knowledge: Given any verifier V, and an honest prover P, there is some simulator S

if $x \in L$, then the distribution of $S(x)$ is indistinguishable from the interaction transcript $\langle a_1, a_2, \ldots, a_m \rangle$ between P and V.

There are different notions of indistinguishability. In today's class, we will insist that the distributions are close in statistical distance. Towards the end, we briefly mention about zero-knowledge proofs in the computational setting.

Pictorially, we can think about two ways for generating a transcript. On the left, we observe P and V as they are communicating about input x, recording the messages that they pass to each other. On the right, we run the simulator on input x.



If our protocol is zero-knowledge, then the two transcripts that are produced should be indistinguishable whenever the input $x \in L$. Said another way, for a given $x \in L$, since V has access to randomness, it can run the simulator on input x in expected polynomial time, as opposed to conversing with P, and still obtain the same information on which it bases its final decision. In this way, V does not gain any new information via the messages in the zero-knowledge protocol. Nonetheless, there are such protocols that still afford V the desired soundness and completeness guarantees. We describe an example of such a protocol below.

2 An Example Zero-Knowledge Proof

We consider the graph isomorphism problem $\mathsf{GI} = \Big\{ \langle G_0, G_1 \rangle \ : \ \exists\, \pi,\ G_1 = \pi(G_0) \Big\}$.

$\mathsf{GI} \in \mathsf{NP}$ since it admits a (deterministic) interactive proof that requires only one round of interaction. The prover simply sends the permutation π to the verifier, who can verify that π is a permutation of V that maps the edges of G_0 onto the edges of G_1. This protocol is not zero-knowledge, since no polynomial time simulator can deterministically generate the permutation π (unless we can find a polynomial time algorithm for GI or prove $\mathsf{P} = \mathsf{NP}$).

Alternatively, we consider the following protocol:

1. P uniformly samples a permutation π and a bit b and sends graph $H = \pi(G_b)$ to V.
2. V uniformly samples a random bit b' and sends to P.
3. P sends the permutation σ such that $\sigma(G_{b'}) = H$.
4. V checks σ and outputs 1 if $\sigma(G_{b'}) = H$.

We argue that this protocol satisfies the three properties of a zero-knowledge proof.

Completeness:

Suppose that $x = \langle G_0, G_1 \rangle \in \mathsf{GI}$. Then $G_1 = \rho(G_0)$ for some permutation ρ. We argue that the by following the protocol, the honest prover will always be able to compute a permutation σ to please the verifier.

- If $b = b'$, then $H = \pi(G_b) = \pi(G_{b'})$, so we can take $\sigma = \pi$.
- If $b = 0$ and $b' = 1$, then $H = \pi(G_0) = \pi(\rho^{-1}(G_1)) = (\pi \circ \rho^{-1})(G_1)$, so we can take $\sigma = \pi \circ \rho^{-1}$.
- If $b = 1$ and $b' = 0$, then $H = \pi(G_1) = \pi(\rho(G_0)) = (\pi \circ \rho)(G_0)$, so we can take $\sigma = \pi \circ \rho$.

In any case, the verifier's check in Step 4 will be successful, so $\text{out}_V \langle P, V \rangle(x) = 1$. We have, in fact, proven the stronger property of perfect completeness.

Soundness:

Suppose that $x = \langle G_0, G_1 \rangle \notin \mathsf{GI}$. Then, H is isomorphic to at most one of $\{G_0, G_1\}$. Since our verifier selects b' uniformly at random, we have $\Pr\big[G_{b'} \text{ isomorphic to } H\big] \leq \frac{1}{2}$. In this case, P will fail to find a suitable permutation σ, so V will reject the input. By repeating the protocol twice and only accepting when both permutations σ are valid, we can magnify the rejection probability to at least $\frac{3}{4}$ (without affecting perfect completeness).

Zero-Knowledge:

Suppose that $x = \langle G_0, G_1 \rangle \in \mathsf{GI}$. A transcript from the communication will have the form $\langle H, b', \sigma \rangle$. We must design a simulator that also produces transcripts of this form.

As an initial attempt, we can first uniformly sample a permutation σ and a bit b' and then use these to construct the permutation $H = \sigma(G_{b'})$. Since each possible σ in the completeness proof involves the uniform random permutation π, σ in the transcript should also a uniform random permutation. Additionally, since G_0 and G_1 are isomorphic, H will be uniform random permutation of this graph. However, the bit b' presents a problem. Since the simulator does not have access to the randomness determining b', it cannot guarantee that the simulator is actually sampling this bit uniformly. Therefore, we are not guaranteed that the simulated transcript has the same distribution as the actual transcript.

Instead, we consider the following simulator S. First, S uniformly samples π and b (just like the protocol) and calculates $H = \pi(G_b)$, sending this to the verifier. If $b = b'$, then the verifier can output $\langle H, b, \pi \rangle$, just as it would in the protocol. Otherwise, it repeats, sampling a new π and b. Conditioned on $b = b'$, the simulated transcript will be indistinguishable from the actual transcript. In each trial, this occurs with probability $\frac{1}{2}$, so the simulator runs for an expected 2 iterations, giving it an expected polynomial runtime.

We briefly mention that zero-knowledge proofs have been intensely studied in the computational setting as well. Assuming the existence of one-way functions, it is known that there exists (computational) zero-knowledge proofs for any language in IP.