CS 171: Discussion Section 11 (April 15)

1 Zero-Knowledge Protocol for Graph Isomorphism

Two graphs are **isomorphic** if it is possible to permute the vertices of one graph to obtain the other graph.

Let $G = (V, E)$ be a graph with n vertices: $V = \{1, \ldots, n\} = [n]$. Let $\pi : [n] \to [n]$ be a permutation of the vertices. We can define a permutation of the graph as follows[1]: $\pi(G) = (V', E')$ is a graph with vertex set $V' = V$ and edge set

$$E' = \{(u, v) \in V \times V : \left(\pi^{-1}(u), \pi^{-1}(v)\right) \in E\}$$

In other words, applying π to the vertices of G produces the graph $\pi(G)$.

Definition 1.1 (Isomorphic Graphs)**.** *Two graphs G_0 and G_1 are **isomorphic** (notated as $G_0 \simeq G_1$) if they have the same number of vertices n, and there exists a permutation $\pi^* : [n] \to [n]$ such that*

$$G_0 = \pi^*(G_1)$$

Question: Give a zero-knowledge proof system for the language of isomorphic graphs $\mathcal{L} = \{(G_0, G_1) : G_0 \simeq G_1\}$. Prove that the scheme satisfies completeness, soundness, and zero-knowledge.

1.1 Definitions

Let (P, V) be the honest prover and honest verifier, respectively. They follow the protocol. Let P^* and V^* be a dishonest prover and verifier, respectively, who may deviate from the protocol. Also, let $\lambda \in \mathbb{N}$ be the security parameter.

Completeness says that a valid proof will be accepted with overwhelming probability.

Definition 1.2 (Completeness)**.** *The protocol satisfies **completeness** if when P and V interact and their inputs satisfy $G_0 = \pi^*(G_1)$, then the verifier will accept the proof with probability $\geq 1 - \mathsf{negl}(\lambda)$.*

Soundness says that if $G_0 \not\simeq G_1$, then no adversarial prover will be able to "trick" the verifier into accepting the proof with greater than negligible probability.

Definition 1.3 (Soundness)**.** *The protocol satisfies **soundness** if for any adversarial prover P^*, when P^* and V interact and their inputs satisfy $G_0 \not\simeq G_1$, then the verifier will accept the proof with probability $\leq \mathsf{negl}(\lambda)$.*

[1] It's technically an abuse of notation to write $\pi(G)$ since π was defined to take a vertex as input, not a graph, but we'll do it anyways.

Zero-Knowledge

Zero-knowledge says that an adversarial verifier cannot learn anything about π^* during the protocol because the information available to the verifier (their view) can be simulated without knowledge of π^*.

To make this definition more formal, let's establish some notation.

- Let V^* be an adversarial verifier for the proof system that may deviate from the protocol in order to try to learn something about π^*. V^* runs in polynomial time.
- Let the verifier's **view**, $\mathsf{view}(V^*; 1^\lambda, G_0, G_1, \pi^*)$, be a list of the verifier's inputs $(1^\lambda, G_0, G_1)$ and any messages sent to or from the verifier during the protocol, when the protocol has inputs $(1^\lambda, G_0, G_1, \pi^*)$.
- Let the simulator Sim be an algorithm that tries to simulate the verifier's view given only $(1^\lambda, G_0, G_1)$. Note that Sim is not given π^*.

  Next, Sim is given black-box access to V^* (notated as Sim^{V^*}). This means Sim can run V^* on any inputs of its choice and rewind V^* to any step, but it cannot modify the internal workings of V^*.

  Finally, the expected value of Sim's runtime should be polynomial in the size of Sim's inputs.
- Let the distinguisher D be an algorithm that outputs a bit and tries to distinguish the verifier's real view from the one produced by the simulator.

Informally, the protocol satisfies **zero-knowledge** if whenever $G_0 = \pi^*(G_1)$, the distinguisher cannot distinguish the real view from the simulated view.

Here is a more-formal definition:

Definition 1.4 (Black-Box Zero-Knowledge). *The protocol satisfies (black-box)* ***zero-knowledge*** *if there exists a simulator* Sim *such that for any* V^* *and any inputs* $(1^\lambda, G_0, G_1, \pi^*)$ *that satisfy* $G_0 = \pi^*(G_1)$ *and any distinguisher* D*:*

$$\left| \Pr\left[D\big(\mathsf{view}(V^*; 1^\lambda, G_0, G_1, \pi^*)\big) \to 1 \right] - \Pr\left[D\big(\mathsf{Sim}^{V^*}(1^\lambda, G_0, G_1)\big) \to 1 \right] \right| \leq \mathsf{negl}(\lambda)$$

2 Polynomial Commitments

Question: Prove that the KZG commitment scheme is not hiding.

2.1 The KZG Commitment Scheme

1. $\mathsf{Setup}(1^n)$:
   (a) Set up a bilinear map by sampling
   $$\mathsf{pp} = (\mathbb{G}, \mathbb{G}_T, q, g, e) \leftarrow \mathcal{G}(1^n)$$
   (b) Sample $\tau \leftarrow \mathbb{Z}_q^*$.
   (c) Finally, output
   $$\mathsf{srs} = \left(\mathsf{pp}, g^{\tau}, g^{(\tau^2)}, \ldots, g^{(\tau^{d-1})}\right)$$
2. $\mathsf{Commit}(f, \mathsf{srs})$:
   (a) Let f be a polynomial $\in \mathbb{Z}_q[X]$ of degree $\leq d-1$:
   $$f(X) = \sum_{i=0}^{d-1} c_i \cdot X^i$$
   where every $c_i \in \mathbb{Z}_q$.
   (b) Compute and output the commitment:
   $$\begin{aligned} F &= \prod_{i=0}^{d-1} \left(g^{(\tau^i)}\right)^{c_i} \\ &= g^{f(\tau)} \end{aligned}$$
3. Open:
   (a) Let $z \in \mathbb{Z}_q$ be an input on which to open the commitment, and let $s = f(z)$. Now the sender will prove that $s = f(z)$.
   (b) The sender computes the polynomial:
   $$t(X) := \frac{f(X) - s}{X - z}$$
   and a commitment $T = \mathsf{Commit}(t, \mathsf{srs})$. Then they send (z, s, T) to the receiver.
   (c) The receiver accepts the opening if and only if:
   $$e(F \cdot g^{-s}, g) = e(T, g^{\tau} \cdot g^{-z}) \tag{2.1}$$

Note that equation 2.1 is satisfied if and only if:

$$\begin{aligned} e(g^{f(\tau)-s}, g) &= e(g^{(f(\tau)-s)/(\tau-z)}, g^{\tau-z}) \\ f(\tau) - s &= t(\tau) \cdot (\tau - z) \end{aligned}$$