Homework 9

Due: November 22, 2024

CS 1510: Intro. to Cryptography and Computer Security

1 Zero Knowledge Proofs for Vertex Cover

In class, we saw a zero-knowledge protocol for proving 3-colorability. Let us design a proof for a different NP-complete problem.

Definition 1 (Vertex Cover) *Given a graph $G = (V, E)$, a vertex cover of size k is a subset $C \subseteq V$ such that $|C| = k$ and that for all edges $e \in E$ at least one endpoint is in C. Consider the language* VertexCover *defined as follows:*

$$\textsc{VertexCover} = \{(G, k) \ : \ G \text{ has a vertex cover of size } k\}$$

Suppose the prover Alice and the verifier Bob share a graph G and the description of a cryptographic commitment algorithm Com. Alice claims that G has a size k vertex cover. Alice will attempt to prove it as follows.

- We assume that each vertex of the original graph is labeled with a unique number from 1 to n (such that $n = |V|$) known to both the prover Alice and the verifier Bob. Alice permutes the set of vertices using a random permutation $\pi : [n] \to [n]$, where we denote the permuted set as $V' = \{\pi(1), \ldots, \pi(n)\}$. Alice computes a list of commitments to the permuted vertices $C_{V'} = \{\mathsf{Com}(\pi(1)), \mathsf{Com}(\pi(2)), \ldots, \mathsf{Com}(\pi(n))\}$.
- Define a new set of edges E' such that for each original edge $e = (u, v) \in E$ between two vertices u and v, there exists an edge $e' = (\pi(u), \pi(v))$ between the permuted vertices $\pi(u)$ and $\pi(v)$. Alice computes commitments to all the edges in E', namely $\{\mathsf{Com}(u', v') \mid e' = (u', v') \in E'\}$, and then randomly permutes these commitments to obtain a set $C_{E'}$.
- Finally, for all $v' \in V'$, Alice proceeds as follows: If $\pi^{-1}(v') \in C$, then set $b_{v'} := 1$; otherwise set $b_{v'} := 0$. Alice computes a list of commitments to the permuted indicators $C_{B'} = \{\mathsf{Com}(b_1), \mathsf{Com}(b_2), \ldots, \mathsf{Com}(b_n)\}$.

How does the rest of this protocol work? We're going to divide the solution into several small steps.

a. Suppose Bob only wanted to check that the graph is represented correctly (i.e. that the graph is the same one, G, that he and Alice have agreed on). How can he confirm this without learning any other information?

b. Now suppose Bob only wants to check that the cover has the appropriate size (the agreed upon k). How can he confirm this without learning any other information?

c. Finally, suppose Bob only wants to check that each edge is covered. How can Bob do this without learning any other information? Specifically, Bob is allowed to examine only one edge in each round. With what probability is he guaranteed to catch Alice in a given round if the graph does not have a k-cover?

d. What should Bob's overall strategy for verifying Alice's vertex cover proof be, and what is the guaranteed probability with which Alice will be caught if the graph does not have a k-cover? (Hint: the strategy will probably have to be randomized.)

e. Assuming Alice is willing to repeat this process as many times as necessary, how many times should Bob run this algorithm so that if Alice cheats she will be caught with probability $1 - \mathsf{negl}(n)$?

f. Give an informal sketch for how the zero-knowledge simulator for this proof system might work.

2 Zero-Knowledge Proofs for CLIQUE

An instance of the *CLIQUE* problem consists of an undirected graph $G = (V, E)$ and a positive integer k. A graph G is in this language for integer k if it has a complete subgraph with k verticies.

$$CLIQUE = \{(G, k) \ : \ G \text{ has a clique of size } k\}$$

That is, if there exists a set $S \subseteq V$, $|S| \geq k$ such that every pair of vertices in S is connected by an edge in E, then (G, k) is in the language. Construct a zero-knowledge proof for the clique problem, and prove your scheme to be sound and complete. You need only sketch the proof of zero-knowledge (i.e., construct a simulator). You may use commitment schemes, but may *not* reduce this problem to any other NP-complete problem.

3 Zero-Knowledge Proofs of Graph Isomorphism

Given two graphs G, H, we say the two graphs are *isomorphic*, denoted as $G \simeq H$, if there exists a permutation of the vertices of G, which, when applied to G, yields exactly H.

a. Give a zero-knowledge proof system (P, V) for graph isomorphism. Prove completeness, soundness, and zero-knowledge for the system you devise.

$$L = \{(G, H) \ : \ G \simeq H\}$$

b. Give a zero-knowledge proof system (P, V) for the following language. Prove completeness, soundness, and zero-knowledge for the system you devise.

$$L = \left\{\left((G_0, G_1), (G_0', G_1')\right) \ : \ G_0 \simeq G_1 \bigvee G_0' \simeq G_1'\right\}$$

Note: Make sure the verifier does not learn which of the two pairs of graphs is isomorphic.

Hint: You do not *need commitments in this question.*

4 Summary Question

Summarize the most important insights from this week's material, including from the lectures, notes, textbooks, homework problems, and other resources you find helpful, into a one-page resource. We expect that these summary pages will help you with the take-home midterm and final. Please note this question is graded based on completion—we will not be checking it for correctness. However, we will be checking it for relevance to the week's content and length.