Notes for Lecture 24 (draft)

Summary

Today we introduce the notion of *zero knowledge proof* and design a zero knowledge protocol for the *graph isomorphism* problem.

1 Intuition

A *zero knowledge proof* is a proof between two parties, a *prover* and a *verifier*. Both parties have in input a "statement" that may or may not be true, for example, the description of a graph G and the statement that G is 3-colorable, or integers N, r and the statement that there is an integer x such that $x^2 \bmod N = r$. 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.

A related concept is that of a *zero knowledge proof of knowledge*, in which the two parties share an input to an NP-type problem, and the prover wants to convince the verifier that he (the prover) *knows a valid solution for the problem on that input*, while again making sure that no information leaks. For example, the common input may be a graph G, and the prover may want to prove that he knows a valid 3-coloring of G, or the common input may be N, r and the prover may want to prove that he knows an x such that $x^2 \bmod N = r$. An example showing the difference between the two definitions is that the common input is an integer N, and the prover wants to prove that he knows a non-trivial factor N. (Here the corresponding "statement" would be that N is composite, but this can easily be checked by the verifier offline, without the need for an interaction.)

2 The Graph Non-Isomorphism Protocol

We say that two graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ are *isomorphic* if there is a bijective relabeling $\pi : V \to V$ of the vertices such that the relabeling of G_1 is the same graph as G_2, that is, if

$$(u, v) \in E_1 \Leftrightarrow (\pi(u), \pi(v)) \in E_2$$

We call $\pi(G_1)$ the graph that as an edge $(\pi(u), \pi(v))$ for every edge (u, v) of E_1.

The graph isomorphism problem is, given two graphs, to check if they are isomorphic.

Here we describe an interactive protocol in which a prover can "convince" a verifier that two given graphs are not isomorphic, and in which the verifier only makes questions for which he already knows an answer, so that, intuitively, he gains no new knowledge from the interaction. (We will give a precise definition later, but we will not prove anything formal about this protocol, which is presented only for intuition.) For the prover, unfortunately, we only know how to provide an exponential time implementation. The verifier algorithm, however, is very efficient.

- Common input: two graphs $G_1 = (V, E_1)$, $G_2 = (V, E_2)$; the prover wants to convince the verifier that they are *not* isomorphic
- The verifier pick a random $b \in \{1, 2\}$ and a permutation $\pi : V \to V$; the verifier send $G = \pi(G_b)$ to the prover
- The prover finds the bit $a \in \{1, 2\}$ such that G_a and G are isomorphic; the prover sends a to the verifier
- The verifier checks that $a = b$, and, if so, accepts

Theorem 1 *Let P be the prover algorithm and V be the verifier algorithm in the above protocol. Then*

1. *If G_1, G_2 are not isomorphic, then the interaction $P(x) \leftrightarrow V(x)$ ends with the verifier accepting with probability 1*
2. *If G_1, G_2 are isomorphic, then for every alternative prover strategy P^*, of arbitrary complexity, the interaction $P^*(x) \leftrightarrow V(x)$ ends with the verifier accepting with probability $1/2$*

The probability in (2) can be reduced to 2^{-k} by repeating the protocol k times.

3 The Graph Isomorphism Protocol

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.

- Verifier's input: two graphs $G_1 = (V, E_1)$, $G_2 = (V, E_2)$;
- Prover's input: G_1, G_2 and permutation π^* such that $\pi^*(G_1) = G_2$; the prover wants to convince the verifier that the graphs are isomorphic
- The prover picks a random permutation $\pi_R : V \to V$ and sends the graph $G := \pi_R(G_2)$
- The verifier picks at random $b \in \{1, 2\}$ and sends b to the prover
- The prover sends back π_R if $b = 2$, and $\pi_R(\pi^*(\cdot))$ otherwise
- The verifier cheks that the permutation π received at the previous round is such that $\pi(G_b) = G$, and accepts if so

Theorem 2 *Let P be the prover algorithm and V be the verifier algorithm in the above protocol. Then*

1. *If G_1, G_2 are isomorphic, then the interaction $P(x) \leftrightarrow V(x)$ ends with the verifier accepting with probability 1*
2. *If G_1, G_2 are not isomorphic, then for every alternative prover strategy P^*, of arbitrary complexity, the interaction $P^*(x) \leftrightarrow V(x)$ ends with the verifier accepting with probability* $1/2$

4 Definition of Proof System and of Zero Knowledge

We conclude the lecture with the formal definition of *proof system* (which captures the fact that a prover successfully convinces a verifier of the truth of a statement) of *honest verifier* zero knowledge, and of (general) zero knowledge.