Varun Sayal 2/13/2014

CSE 350, Theory of Computation.
LaTeX transcription of lecture 4.

Zero Knowledge Proofs:

Definition 1. *A zero knowledge proof is a proof that proves a given statement without revealing any further information. That is, one can be convinced, with reasonable certainty, about the veracity of a claim but has no ability to reproduce the proof. A ZK proof simply proves a statement is true and reveals absolutely nothing else.*

Example : How to prove that a planar graph is 3-colorable.

Definition 2. *A planar graph is a graph that can be drawn on a 2-dimensional plane without crossing lines.*

Definition 3. *A graph is 3-colorable if a graph's nodes can be colored with only 3 colors such that no to adjacent verticies are colored the same.*

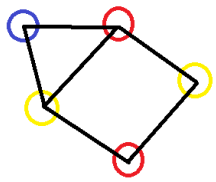

Figure 1: A planar, 3-colorable graph

$\star$ *Note:* It is important to notice that the problem of proving any graph in-general 3-colorable is NP-complete. That is, there is no polynomial time algorithm, yet, to find the coloring.

Normally, to prove a planar graph is 3-colorable, all one has to do is find the coloring. To verify the proof, one simply has to check that no two adjacent nodes have the same coloring. If there are no adjacent identically colored pairs, the coloring and proof are valid.

This however, as mentioned before, is NP-complete for general graphs and therefore makes computing the coloring difficult. Furthermore, the coloring reveals much more to the verifier than is necessary. All we are trying to prove is that a graph is 3-colorable, but to do this, we are exposing information about the relationships between every node on the graph. This seems like overkill.

Question : How do we, with high probability, prove a graph is 3-colorable without revealing anything else about the graph?

Let us define a procedure between the prover and the verifier that will accomplish this.

1. The prover, who knows the coloring, colors the nodes.

2. The prover then hides the colors. If there was a graph drawn on a blackboard, for example, imagine the prover covers all the colored chalk nodes with post-its.

3. The verifier then enters the room and gets to see the coloring of two randomly chosen nodes.[1]

4. The verifier then verifies that the colors are different.

5. Next, the verifier leaves the room.

6. Then, the prover randomly permutes the coloring, while maintaining the relationships[2] and we repeat.

7. If the verifier ever uncovers two adjacent nodes that are the same color, the verifier rejects the proof.

After each of the k iterations of this procedure. If the proof is valid, the verifier should accept it. If the proof is invalid, that is, there isn't a 3-coloring of that graph, the verifier should accept the proof with probability $< \frac{1}{2}^k$. After a large enough k, the probability of the prover passing off an invalid graph as valid will asymptotically approach 0.

That is, if we take the Probability of catching an error after one iteration to be C and the probability of not catching an error after k iterations to be N,

$$Pr\left(C\right) \geq \frac{1}{n^2} \tag{1}$$

$$Pr\left(N\right) \leq \left(1 - \frac{1}{n^2}\right)^k \tag{2}$$

Using the substitution $k = cn^2$ and the following deathbed formulas,

$$\left(1 + x\right)^x \approx e \tag{3}$$

$$\left(1 - \frac{1}{x}\right)^x \approx \frac{1}{e} \tag{4}$$

We get that,

$$Pr\left(N\right) \leq \left(1 - \frac{1}{n^2}\right)^k \approx \left(\frac{1}{e}\right)^c \tag{5}$$

$\star$ c is just a constant that depends on the k number of tries and n number of nodes.

Question : What makes this a ZK proof?

Basically, it is because we know absolutely nothing about the graph. Nothing about the relationship of the nodes is revealed to us because all the verifier ever sees is 2-randomly selected nodes.

Furthermore, this is ZK because a program can be written to simulate the graph being proven. If we inspect the procedure carefully, we note that the interaction that occurs between the verifier and prover boils down to two things.

[1] The nodes are chosen randomly so that the prover cannot trick the verifier by predicting which 2 nodes will be picked next.
[2] i.e Blue $\rightarrow$ Red, Yellow $\rightarrow$ Blue etc.

1. The verifier asks for 2 random connected nodes.
2. The prover responds with 2 random colors.

The simulator for this can be written as follows:

1. Get r, a random number s.t. $r \epsilon \{1, 2, 3....|E|\}$
2. Get two random colors, (C_1, C_2) s.t. $(C_1 \neq C_2) \wedge C_1, C_2 \epsilon \{red, yellow, blue\}$.
3. Print r
4. Print (C_1, C_2)

END

Notation Review:

Functions:

A function $f : A \to B$ is a relation $f :\subseteq A \times B$ s.t. $\forall\ a \in A\ \exists$ exactly one $b \in B$ s.t. $(a, b) \in R$.

Types of functions, F(x)

1. One-to-one (1:1), Injective. $\langle f(x_1) = y \wedge f(x_2) = y) \Rightarrow x_1 = x_2 \rangle$
2. Onto: Surjective. $(x_1 \neq x_2) \Rightarrow f(x_1) \neq f(x_2)$
3. Bijective = Surjective + Injective. $\Leftrightarrow$

Countably Infinite:

A set S is countably infinite if there exists a bijection between S and the natural numbers, $\mathbb{N}$.

Things that are countably infinite:

- Set of all possible finite binary strings.
- Set of all C computer programs.

Things that are not countably infinite:

- $\mathbb{R}$
- The powerset of $\mathbb{N}$, $P(\mathbb{N})$
- Set of all infinitely long strings.

Cardinality:

The cardinality of a set is the number of unique elements in the set.

$\star$ A set, S, is finite if $\exists\ n$ such that there is a bijection from $\{1, 2, 3....n\} \Leftrightarrow S$