Zero Knowledge Proofs

In the previous tutorial we saw signatures as a tool to prove some document's origin. This proof is done after the signing of the document, and one can prove his identity by signing the challenges sent to him by the one who seek proof to his identity (this couple will be known as Peggy and Vig), though this method of signing challenges suffers from several problems.

The main idea behind Zero-Knowledge Proofs (ZK Proofs), is that, Vig receives no information from Peggy's proof, but has to conclude that Peggy knows something that Vig (and no-one else besides Peggy) knows.

This makes ZK Proofs to a most powerful tool in proving one identity, while preventing from everybody else to gain information about the proof, so they will not be able to proof the statement themself.

Requirements

We require two things from ZK Proofs (actually from the protocols around ZK Proofs):

completeness

If x is the right answer to a (decision) problem, then Vic will accept Peggy's proof.

soundness

If x is the wrong answer to a (decision) problem, then Vic will reject Peggy's proof with high probability (close to 1).

We will speak about Interactive Proofs, where the computations of the verifier are done in a polynomial time. Note that the Prover Peggy is considered to have no limitations on her computational power (we of course will show cases where this requirement will not be necessary, otherwise it will not be that practical)

Graph Isomorphism

The problem of Graph Isomorphism (GI) is considered to be in NP, and if $P \neq NP$ then the GI is not NP-Complete. The problem is defined as follows:
Given $(G_1, G_2$, are the graphs are isomorphic, is there a permutation on G_1 vertexes which will make him to be G_2. If such permutation exist, we say that $(G_1, G_2) \in GI$.

The ZK Prof is based on the fact that it is hard to find the permutation (otherwise, the GI Language would not have been NP-Complete). The precomputation phase, is made by Peggy, which finds a graph G_1 and find a random permutation π and computes $G_2 = \pi(G_1)$, and publishes the pair (G_1, G_2) (and keeps π in secret).

When Peggy is required to prove her identity the following algorithm is carried out:

1. Peggy chooses at random permutation τ and sends $H = \tau(G_1)$ to Vic.
2. Vic chooses at random $i \in \{1, 2\}$ and sends i back

to Peggy.

3. Peggy sends $\rho = \tau$ if $i = 1$ and $\rho = \tau \circ \pi$ if $i = 2$.
4. Vic checks that $H = \rho(G_i)$

It is easy to see that only if Peggy knows the permutation π she can efficiently find ρ in all cases. Oscar who tries to pretend he is Peggy would have to guess which value Vic is going to choose as i, and compute $H' = \rho(G_i)$, as he will not be able to find a permutation from the other graph to H'.

Note that Peggy's side does not need to keep π, as we assume she has infinite computing power, but in order to implement the above algorithm in real-life, she must keep π in secret.

We shall now show that Vic gains no information from her conversation with Peggy. The usual method for doing so, is to prove, that Vic can talk to an oracle she has in her pocket, and receive the same information from it. The method of proving that Vic receives no information about the protocol is to look on the session between Vic and Peggy, named *transcript*. In our case the transcript T is of the form:
$T = ((G_1, G_2); (H_1, i_1, \rho_1); \ldots; (H_n, i_n, \rho_n))$
We can show an algorithm to create such transcripts on our own (without the help of Peggy - though the chance of computing randomly a given transcript is small), the algorithm is based on the fact, that we can choose i and then create ρ and H (like Oscar would have done if he

could know what Vic will choose).

- Get (G_1, G_2)
- for $j = 1$ to n do
  1. Choose $i_j = 1$ or 2 at random.
  2. Choose ρ_j at random.
  3. Compute $H_j = \rho_j(G_{i_j})$.
  4. Add (H_j, i_j, ρ_j) to the transcript.

One can see that the above algorithm will create a valid transcripts, and thus Vic receives no information from her conversation with Peggy (and thus remain unaware to π). Note that the full proof require to show that each transcript has the same probability both when the simulator creates it and when it is derived from a real conversation (See for more information Stinson page 394).