# Theoretical Cryptography, Lecture 6

Instructor: Manuel Blum
Scribe: George Skoptsov

February 6, 2006

## 1 Overview

Today's topics:

- Even More Zero-knowledge.
- Definition and zero-knowledge proof of Graph Isomorphism.
- Definition and zero-knowledge proof of Graph Nonisomorphism

Today's lecture is on perfect zero-knowledge protocols for both graph isomorphism and graph non-isomorphism. This is a fairly standard lecture, googling for the topics is a good idea.

## 2 Graph Isomorphism

### 2.1 Review

Let's review what graph isomorphism is as a decision problem.

*Instance*: $G = (V_G, E_G)$ and $H = (V_H, E_H)$ are two $n$-node graphs ($|V_G| = |V_H|$, presented as $n \times n$ incidence matrices. Since they're undirected graphs, matrices are symmetric.

*Question*: Is $G \cong H$? In other words, does there $\exists f : V_G \to V_H$, such that $\forall i, j,\ (v_i, v_j) \in E_G \Leftrightarrow (f(v_i), f(v_j)) \in E_H$ (does there exist a function $f$, which maps vertices of $G$ to vertices of $H$ so that iff there is an edge between vertices of $G$, there is an edge between their mappings in $H$)? Such a function $f$ is an isomorphism between $G$ and $H$.

### 2.2 Zero-knowledge protocol

How does Prover prove to Verifier that an isomorphism exists?

*Input*:
2 isomorphic graphs $G$, $H$ on $n$ nodes each.
Prover knows isomorphism $f$.
A security parameter $k$ (positive integer).

*Output*:
A zero-knowledge protocol that proves $P$ knows $f$. Prover gives no info to $\tilde{V}$ $\tilde{P}$ can cheat (successfully) with probability $\leq 1/2^n$.

*Protocol*: repeat $k$ times.
**Prover**: Privately take $G$ and randomly permute vertices to get a graph $F$.
**Prover**: Publically present $F$ to Verifier ($G$ and $H$ are public from the beginning).
**Verifier**: Toss a coin, and ask Prover to show that $G \cong F$ if heads, or $H \cong F$ if tails.

**Theorem 2.1** *Prover gives no information to* $\tilde{V}$.

**Proof** $\tilde{V}$ gets either $G \cong F$ (a random relabeling of $G$) or $H \cong F$ (a random relabeling of $H$). For either result, $\tilde{V}$ could have it generated herself. ∎

**Theorem 2.2** *Verifier gets proof that* $P$ *knows* $f$. *Probability that* $\tilde{P}$ *cheats (successfully) is* $\leq 1/2^k$.

**Proof** $P$ has to reveal the isomorphism mapping from either $G$ to $F$ or from $H$ to $F$. Prover knows the former simply by construction (it's the random permutation from above), and the latter also because it knows the isomorphism from $H$ to $G$. A crooked prover $\tilde{P}$ doesn't know the isomorphism from $H$ to $G$, so he wouldn't know at least one of two isomorphisms: the one between $G$ and $F$, or the one between $H$ and $F$. Hence in each round Verifier has probability of $1/2$ of catching him. ∎

Notice that this is the first zero-knowledge proof we have seen that has no gray paint: it's a perfect zero-knowledge proof.

## 3 Perfect zero-knowledge

*Question*: What is perfect zero-knowledge?

**Rudich**: Perfect zero-knowledge implies that the probability distribution of conversations between the Prover and Verifier is *identical* to the distribution of outcomes from the simulator. (Ryan agrees.)

**Blum**: Perfect zero-knowledge means there is no gray paint, so complexity and possibility of compromising bit-commitment are avoided. Note: even though zero-knowledge is perfect, and there is no way to leak information, it is still possible for Prover to cheat Verifier, and as we see later, vice versa.

## 4 Graph Non-isomorphism

We now look for a zero-knowledge protocol for the *complement* of Graph Isomorphism.

*Input*:
Two non-isomorphic graphs $G$, $H$ on $n$ nodes each.
A Prover $P$, who knows an efficient proof that $G \ncong H$, and the proof works for any relabelling $G'$ of $G$ and $H'$ of $H$.
A Verifier $V$ to be convinced that $G$ and $H$ are not isomorphic.
A security parameter $k$ (positive integer).

### 4.1 Bad protocol

Recall that Prover's goal is to convince Verifier that $G \ncong H$. Suppose Prover has an efficient way of telling these particular two graphs apart, in other words there is a known difference between them that is independent of labelling. Let's take a look at an example protocol and see if there's anything wrong with it.

The idea is to show that a random permutation of $G$ is not isomorphic to a random permutation of $H$.

**Verifier**: Take a disjoint graph $\langle G, H\rangle$ and relabel it as $\langle F_1, F_2\rangle$ in a random order. Keep the relabelling private, and pass the new graph to Prover.
**Prover**: Decide which of $F_1$ and $F_2$ is isomorphic to $G$ .
Repeat $k$ times.

What's wrong with this proof? Well, what if Verifier is crooked? Verifier may know some relabelled $F_1$ and $F_2$, which correspond to $H$ or $G$, but she may not know the exact correspondence. Then she may use this protocol to find out whether $F_1$ or $F_2$ is $G$, so the protocol is not at all zero-knowledge. The way to correct this proof is to force Verifier to prove that $\langle F_1, F_2\rangle$ has been constructed by her from $\langle G, H\rangle$. A way to do it is to ask Verifier to prove isomorphism between $\langle G, H\rangle$ and $\langle F_1, F_2\rangle$, which she should know by construction.

### 4.2 Correct protocol

Repeat the following $k$ times.
**Verifier**: Take a disjoint graph $\langle G, H\rangle$ and randomly relabel and permute it as $\langle F_1, F_2\rangle$. Keep the relabeling private, and pass the new graph to Prover.
**Verifier**: Give Prover a zero-knowledge proof that $\langle G, H\rangle \cong \langle F_1, F_2\rangle$.
**Prover**: Decide which of $F_1$ and $F_2$ is isomorphic to $G$ .

**Theorem 4.1** *The probability that a crooked Prover (who cannot tell the difference between $G$ and $H$) succeeds in cheating is $\leq 1/2^k$.*

**Proof** The crooked Prover can simply randomly guess which $F_i \cong G$. But he is wrong with probability $1/2$, and thus over $k$ rounds, the probability he was never wrong is at most $1/2^k$. ∎

**Theorem 4.2** *Honest Prover $P$ does not give $V$ any information that she does not already know (could not compute for herself).*

**Proof** The only information $P$ passes to $V$ is a a decision on whether $F_1$ or $F_2$ is isomorphic to $G$. Since $V$ must prove that it constructed $F_1$ and $F_2$, $P$ will not reveal any extra information about $G$ or $F_1$ and $F_2$– $V$ could have told whether $F_1$ or $F_2$ were isomorphic to $G$ herself. ∎

Two things to note:

- this is the first zero-knowledge protocol where Verifier goes first.
- this is the first zero-knowledge protocol where there is a possibility that Verifier can cheat Prover.

## 5 Final points

Here are five points to think about from this lecture:

1. Perfect zero-knowledge implies no gray paint, *i.e.* no bit-commitment (which could be broken) is required.

2. In all zero-knowledge proofs that we have seen until now, the proof system is explicit, and the Prover demonstrates that he knows a proof in that particular proof system. The proof is hidden, but the proof system itself is not hidden. For example, in the case of Hamilton cycle, the proof itself is a *Hamilton cycle*.

   For example, consider the related problem of Hamilton path in a tournament graph $G = (V, E)$. $G$ is a tournament graph iff $\forall v_j, v_j \in V \ i \neq j$, exactly one of $(v_i, v_j)$ and $(v_j, v_i)$ is in $E$. Such graphs always have a Hamilton path!

   For example, consider Subset Sum Variant 2 (we saw Variant 1 in Lecture 3).

   *Instance*: A set of $n$ distinct integers $A : \{a_1, a_2, ..., a_n\}$ such that for all $i$, $|a_i| \leq n^2$.

   *Question*: Do there exist disjoint subsets $B,\ C \subseteq A$ such that

   $$\sum_{a_i \in B} a_i = \sum_{a_i \in C} a_i?$$

   The answer must be yes, since there are $2^n$ distinct (but not necessarily disjoint) subsets, and the sum of all elements in $A$ is less than $2^n$. Hence there are two disjoint subsets whose sums are equal. This is a proof, but not in the standard proof system that exhibits the two subsets.

   As we have seen, the zero-knowledge proof of graph isomorphism is indeed in the standard proof system, which explicitly shows some isomorphism (though a random one). However, the protocol for graph non-isomorphism does not reveal the proof system used by the Prover.

3. In general, the definition of a zero-knowledge proof in general permits an exponentially small probability of error to both Prover and Verifier. (This is the two-sided error model that was discussed in Lecture 4.) In the graph isomorphism example, the prover gives away no information whatsoever about the specific proof (an actual isomorphism). No gray paint is required!

   In the case of graph non-isomorphism, Prover can be "cheated" even though there is no gray paint, since Verifier conducts a zero-knowledge proof of her own. Hence both Prover and Verifier have exponentially small probabilities of being cheated.

4. Note we've seen two general ways in which zero-knowledge protocols can be designed:

   - Verifier gets something from Prover, sampled from a uniform probability distribution.
   - Verifier constructs several instances and Prover chooses the correct one.

5. Suppose we are given two $n \times n$ matrices $A$ and $B$. Recall $A$ and $B$ are similar iff there exists an invertible $P$ such that $A = PBP^{-1}$. How could we prove that $A$ and $B$ are similar?