CMSC 858K — Advanced Topics in Cryptography	April 8, 2004
Lecture 19	
Lecturer: Jonathan Katz	*Scribe(s):* *Nikolai Yakovenko* *Jeffrey Blank*

1 Introduction and Preliminaries

In a previous lecture, we showed a zero-knowledge (ZK) proof system for the language of graph isomorphism. Our goal here is to show a ZK proof system for any language in $\mathcal{NP}$. To do so, it suffices to show a ZK proof system Π for any $\mathcal{NP}$-complete language L (note that graph isomorphism is not believed to be $\mathcal{NP}$-complete); given such a Π and any $L' \in \mathcal{NP}$, we then obtain a ZK proof for L' by (1) reducing the common input x' (which is supposedly in L') to a string x such that $x' \in L' \Leftrightarrow x \in L$; and then (2) running the original proof system Π on common input x. (Actually, if we want this to work for a poly-time prover then we need the reduction from L' to L to also preserve witnesses; i.e., there should be poly-time computable functions f_1, f_2 such that $x' \in L' \Leftrightarrow f_1(x') \in L$ and if w' is a witness for $x' \in L'$ then $f_2(w')$ should be a witness that $f_1(x') \in L$.)

In this lecture, we show a ZK proof system for the language of 3-colorability, which is $\mathcal{NP}$-complete. Before doing so, we will first define the notion of a *commitment scheme*.

1.1 Commitment Schemes

Informally, a commitment scheme provides a way for a *sender* to commit to a value without revealing it to a *receiver*. At some later point, however, the sender can reveal his committed value and the receiver will be convinced that the sender did not "change his mind". A good analogy is the following commitment scheme which works when the parties are sitting at a table together: the sender writes his value on a piece of paper, places it in an envelope, seals the envelope, and places the envelope on the table. Assuming normal paper and ink, the sender certainly cannot change the value inside the envelope (i.e., he is *committed* to that value) yet the receiver cannot learn the value while the envelope remains unopened. Unfortunately, this only works when the parties are in the room together, but does not lead to a protocol that can be run over the Internet!

We refer to the properties sketched above as *hiding* and *binding*: The *hiding property* refers to the receiver's inability to learn the value after the sender has committed, but before he has revealed his commitment. The *binding property* refers to the sender's inability to change the value after committing to it.

If we try to formally define these notions, it turns out that there are two different "flavors" of commitment schemes one can consider: the first ensures that the binding property holds even for an all-powerful sender, but the hiding property "only" holds for a computationally-bounded[1] receiver. (We may also say that binding holds information-

[1] While one choice might be to equate "computationally-bounded" with PPT, for the application to ZK proofs we will need hiding to hold with respect to polynomial-size *circuits* (i.e., we need a non-uniform hardness assumption).

theoretically, while hiding holds only computationally.) Commitment schemes of this sort are called *standard*. The second type of commitment satisfies the hiding property even for an all-powerful receiver, but now the binding property only holds for a computationally-bounded sender. (I.e., this scheme achieves information-theoretic hiding, but only computational binding.) Such commitment schemes are termed *perfect*.[2] The scheme one uses will depend on the application, as we will see.

For now, we define only a standard commitment scheme. In general, a commitment scheme may be interactive, but for simplicity we give a definition only for the case of non-interactive commitment. Here, the sender outputs a pair $(\mathsf{com}, \mathsf{dec})$ consisting of a commitment and a decommitment: sending com to the receiver constitutes the commitment phase and sending dec to the receiver constitutes the decommitment phase.

Definition 1 A *standard commitment scheme* consists of a pair of PPT algorithms $(\mathcal{S}, \mathcal{R})$ satisfying the following:

Correctness For all k and all $b \in \{0,1\}$:

$$\Pr[(\mathsf{com}, \mathsf{dec}) \leftarrow \mathcal{S}(1^k, b) : \mathcal{R}(1^k, \mathsf{com}, \mathsf{dec}) = b] = 1.$$

Binding The following is negligible even for an all-powerful $\mathcal{S}^*$:

$$\Pr[(\mathsf{com}, \mathsf{dec}, \mathsf{dec}') := \mathcal{S}^*(1^k) : \mathcal{R}(1^k, \mathsf{com}, \mathsf{dec}) = 0 \wedge \mathcal{R}(1^k, \mathsf{com}, \mathsf{dec}') = 1].$$

Hiding The following is negligible for any family $\{R_k^*\}$ of polynomial-size circuits (see footnote 1):

$$\left| \Pr[b \leftarrow \{0,1\}; (\mathsf{com}, \mathsf{dec}) \leftarrow \mathcal{S}(1^k, b) : \mathcal{R}_k^*(\mathsf{com}) = b] - \frac{1}{2} \right|.$$

$\diamondsuit$

We remark that it is easy to extend the above definition to *string commitment* rather than just *bit commitment*. Furthermore, it is easy to construct a string commitment scheme from any bit commitment scheme: just commit to the bits of the string one-by-one. Here, the hiding definition may be more easily thought of in terms of an "indistinguishability-type" game as in the case of encryption: an adversary submits two strings m_0, m_1 to a "commitment oracle" which returns a commitment of m_b for random b; the adversary succeeds if it guesses the value of b, and we say the commitment scheme is secure if every poly-size family of circuits succeeds with probability negligibly close to half.

We do not pursue constructions of commitment schemes here, and instead defer that to another lecture. Here, we will instead be more interested in using a commitment scheme (as a black box) to construct a ZK proof system.

[2] In case you are wondering: it is not too difficult to show the *impossibility* of simultaneously achieving information-theoretic binding and hiding. At the other extreme, schemes achieving both computational binding and hiding may be suitable for some applications, but since we can (for the most part) achieve the stronger notions of security anyway, this case is not so interesting.

2 A ZK Proof System for 3-Colorability

The language of 3-colorability is the set of graphs which can be "3-colored"; i.e., graphs for which the colors "read", "blue", and "green" can be assigned to its vertices such that no two adjacent vertices (vertices sharing an edge) have the same color. Formally, a coloring of a graph G can be viewed as function ϕ from the vertices of G to the set $\{r, b, g\}$ such that if (u, v) is an edge in G, then $\phi(u) \neq \phi(v)$. Deciding 3-colorability is known to be an $\mathcal{NP}$-complete problem.

We now show a ZK proof for 3-colorability: At the beginning of the protocol, both the prover and verifier know the same graph G with n vertices, and the prover also knows a 3-coloring ϕ for this graph (we let ϕ_i denote $\phi(i)$; i.e., the color assigned to vertex i).

- First, the prover chooses a random permutation φ over the set $\{r, b, g\}$. He then commits to the (permuted) coloring vertex-by-vertex, and sends the n commitments $\boxed{\varphi(\phi_1)} \cdots \boxed{\varphi(\phi_n)}$.
- The verifier chooses a random edge (i, j) in G, and sends (i, j) to the prover.
- The prover sends decommitments to the i^{th} and j^{th} commitments that it sent in the first round.
- The verifier recovers the decommitted values, denoted φ_i and φ_j. The verifier accepts iff $\varphi_i, \varphi_j \in \{r, b, g\}$ and $\varphi_i \neq \varphi_j$.

Note that the proof system satisfies completeness, since an honest prover using a valid coloring will always cause the verifier to accept. Furthermore, (weak) soundness holds if the commitment scheme is binding. To see this, assume for a moment that the commitments are "perfect" (i.e., sealed envelopes) and let P^* be a cheating prover with G a graph that is not 3-colorable. Then after the first round, P^* is committed to *some* assignment of vertices to colors (we may assume that if a particular commitment is invalid in any way, then we arbitrarily assign it the color "red"). Since the graph is not 3-colorable, there must then be at least one edge (u, v) for which u and v are assigned the same color. So if the verifier chooses this edge, he will reject the proof. The probability that the verifier chooses such an edge is (at least) $1/|E| \geq 1/n^2$, where $|E|$ is the number of edges in G (it is at least this probability because there might be more than one edge whose vertices are not colored correctly). So, the prover fails to convince the verifier with probability at least $1/n^2$, which is inverse polynomial (in the size of the graph). Of course, we need to also take into account the fact that these are not "perfect" commitments; however, the probability that the prover can open any of these commitments in more than one way is negligible (by the binding property) so this decreases the probability that the verifier will accept by only a negligible probability.

As usual, repeating the protocol sufficiently-many (but polynomially-many) times (sequentially if we want to preserve the ZK property[3]) yields a proof system with negligible soundness error.

In the next two sections, we show that this proof system is *zero knowledge*.

[3] We rely here on the fact that the above proof system satisfies the stronger definition of *auxiliary-input zero knowledge*; see Lecture 17 and [1].

2.1 Simulation for an Honest Verifier

First, we informally discuss why the above protocol is *honest-verifier* zero knowledge. (We do not give a formal proof, since one will follow anyway from the stronger result we show in the following section.) Imagine the following simulator, which receives only the graph G (but no coloring); as usual, the simulator "guesses in advance" the challenge of the verifier:

- Choose a random edge (i, j) in G.
- Choose φ_i at random from $\{r, b, g\}$ and φ_j at random from $\{r, b, g\} \setminus \{\varphi_i\}$. For all $k \in \{1, \ldots, n\}$, $k \neq i, j$, set $\varphi_k = r$. Generate commitments $\boxed{\varphi_1} \cdots \boxed{\varphi_n}$ to these n values.
- Output the transcript $\boxed{\varphi_1} \cdots \boxed{\varphi_n}$; (i, j); φ_i, φ_j. (Actually, the last round should include decommitments to φ_i, φ_j.)

Now, note that the "only" difference between the distribution on transcripts output by the simulator, and the distribution on transcripts resulting from a real execution of the protocol are that, in the former, all commitments other than φ_i, φ_j are to "r" while, in the latter, all the commitments are to some valid 3-coloring. However, by the hiding property of the commitment scheme, these two distributions are computationally indistinguishable.[4]

2.2 Simulation for a Dishonest Verifier

We now show, more formally, a simulator for an arbitrary PPT verifier V^*.

$\underline{\mathsf{Sim}(1^k, G)}$
fix random tape ω for V^*
for $i = 1$ to $|E|^2$:
  choose random edge (u, v)
  generate vector of commitments com as in previous section
  run $V^*(\mathsf{com}; \omega)$ to obtain challenge (u^*, v^*)
  if $(u^*, v^*) = (u, v)$ output transcript as in previous section
if all previous iterations have failed, output $\perp$

We now want to claim that, for all G, ϕ, the output distribution defined by Sim is computationally indistinguishable from the distribution over real executions of the protocol. The intuition is exactly as in the case of the ZK proof of graph isomorphism: we want to claim that Sim outputs $\perp$ with only negligible probability, and that conditioned on *not* outputting $\perp$ the transcripts looks "the same". However, two differences arise here which did not arise previously:

1. First, it is not immediately clear that Sim outputs $\perp$ with only negligible probability. To argue this, we would like to claim that in any iteration of the loop the probability that $(u^*, v^*) = (u, v)$ is $1/|E|$ (similar to the case of graph isomorphism). In the

[4]We remark that this is in contrast to the ZK proof system for graph isomorphism, where the simulated transcripts were *perfectly* indistinguishable from real transcripts in the case of HVZK, and *statistically* indistinguishable from real transcripts in the case of ZK.

case of graph isomorphism, however, the view of V^* was independent of the challenge guessed by the simulator; here, this is *no longer true* since the vector of commitments given to V^* *does* reveal the guess of Sim (in an information-theoretic sense). On the other hand, since V^* runs in polynomial-time and the commitments are hiding we can show that this does not make "too much difference"; this requires formal proof.

2. Second, in the case of graph isomorphism the transcripts were identically distributed (conditioned on not outputting $\perp$); here, though, the transcripts will (only) be computationally indistinguishable.

We now give a (sketch of a) formal proof which will (we hope!) provide the interested reader with all the necessary elements to construct a full proof. (The reader is also invited to see [1].) First, consider the following modified "simulation" which is not really a simulation at all since it will use the coloring ϕ used by the real prover.

$\underline{\mathsf{Sim}'(1^k, G, \phi)}$
fix random tape ω for V^*
for $i = 1$ to $|E|^2$:
  choose random edge (u, v)
  Using ϕ, generate a vector of commitments com exactly like the honest prover
  run $V^*(\mathsf{com}; \omega)$ to obtain challenge (u^*, v^*)
  if $(u^*, v^*) = (u, v)$ output transcript as in previous section
if all previous iterations have failed, output $\perp$

We claim that (for all G, ϕ) the output distribution generated by Sim' is *statistically-close* to the distribution of real executions of the protocol. The argument here is *exactly* as in the case of graph isomorphism: note that now the "guess" of the simulator is information-theoretically hidden from V^* (since the vector of commitments is for a *valid* 3-coloring, so there is no way to tell which edge was guessed by Sim') and so the probability of outputting $\perp$ is negligible; furthermore, conditioned on not outputting $\perp$ the distributions are identical. (We stress that the above does *not* constitute a valid simulation, however, since Sim' is given ϕ. Instead, it is just a "mental experiment".)

We next claim that (for all G, ϕ) the output distribution generated by Sim' is computationally indistinguishable from the output distribution generated by Sim. (By a hybrid argument, this shows that the distribution generated by Sim is computationally indistinguishable from the real distribution, and completes the proof.) To see this, assume the contrary. Then there is a poly-time distinguisher D^* that can distinguish between the two distributions with probability that is not negligible. But then we can create a poly-time distinguisher[5] D that violates the hiding property of the commitment scheme as follows (we use the "indistinguishability-based" characterization of the hiding property, as discussed

[5] In fact, the distinguisher we construct will be a poly-size circuit (and this is why we need to commitment scheme to satisfy a non-uniform definition of security) because we will have to incorporate the graph G and the coloring ϕ. Such subtleties are glossed over in this write-up.

earlier):

$\underline{D(1^k, G, \phi)}$
fix random tape ω for V^*
for $i = 1$ to $|E|^2$:
  1. choose random edge (u, v)
  2. Choose random, different colors φ_u for u and φ_v for v and commit to these
  3. for all other vertices, generate two vectors of length $n-2$:
    one in which every vertex (i.e., other than u, v) is colored red,
    and one in which the vertices are colored using a random permutation φ of ϕ
    subject to $\varphi(u) = \varphi_u$ and $\varphi(v) = \varphi_v$
  3. submit these messages to the commitment oracle and get back
    a vector of $n-2$ commitments
    let com represent these commitments along with
    the commitments to φ_u, φ_v (all in the correct order)
  4. run $V^*(\mathsf{com}; \omega)$ to obtain challenge (u^*, v^*)
  if $(u^*, v^*) = (u, v)$ output $\langle \mathsf{com}; (u, v); \varphi_u, \varphi_v \rangle$ as the transcript
if all previous iterations have failed, output $\perp$
run D^* on the resulting transcript, and output whatever D^* outputs

The proof concludes by making the following observations: (1) if the commitments returned by the "commitment oracle" are of the first type (where vertices other than u, v are colored red) then the transcript given to D^* is distributed exactly according to the transcripts output by Sim; (2) if the commitments returned by the "commitment oracle" are of the second type (where they form a commitment to a valid 3-coloring) then the transcript given to D^* is distributed exactly according to the transcripts output by Sim'. Thus, (3) if D^* can distinguish between these, then D can distinguish what kind of commitments are being given to it by its oracle. Since D runs in polynomial time, this is a contradiction.

References

[1] O. Goldreich. *Foundation of Cryptography, vol. 1: Basic Tools*, Cambridge University Press, 2001.