600.641 Special Topics in Theoretical Cryptography February 5, 2007

Lecture 5: Zero Knowledge for all of NP

Instructor: Susan Hohenberger *Scribe: Lori Kraus*

1 Administrative

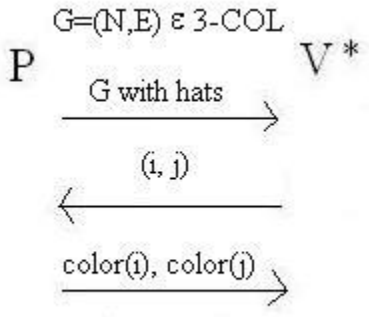
The first problem set goes out today. We can pick up a paper copy after class or find it online. We must email Susan a PDF of the completed problem set by midnight on Friday, February 16th. Note, the first problem is True/False/Unknown and there is no justification needed. Unknown refers to the fact that the statement is unknown within the community at large, not that we personally do not know the answer. To help with the problem set, there is a LaTeX template on the website.

The class mailing list is cs641@jhu.edu. If you didn't already get a test email, please send an email requesting to be on the list to susan@cs.jhu.edu. Note, Susan is not on the class email list, so we can discuss concepts with just our classmates. However, if you want to include her on the email, don't forget to add her address.

Scribed lectures of 1 and 3 will be posted soon.

2 Zero Knowledge Proofs from Yesterday

Last class we saw a method involving hats to prove in zero-knowledge that a graph is 3 colorable. The verifier is sent out of the room and the prover colors the graph and places hats on every node. The verifier returns and chooses an edge. The prover must then remove the hats from the two nodes connected by that edge and the verifier checks to make sure they are different colors. Then the protocol is repeated $|E|$ times. Each round, the prover can cheat with a $1 - \frac{1}{|E|}$ probability. Each round is independent.



Fact: $\forall t, n \in \mathbb{R},\ n \geq 1,\ |t| \leq n,\ (1 + \frac{t}{n})^n \leq e^t$.

Suppose we run the protocol $n = |E|$ times and $t = -1$ (since we want to analyze $(1-\frac{1}{|E|})^{|E|}$). Then $(1-\frac{1}{|E|})^{|E|} \leq e^{-1} \approx 0.36$ is the probability that a prover will be able to cheat. This is less than $1/2$, so this satisfies our soundness definition. If we run the protocol, $|E|^2$ times, then the soundness error reduces to $\leq (0.37)^{|E|}$, which is negligible.

3 Bit Commitment Schemes

Today we will look at a digital ZK proof for graph 3 coloring. In the first step of the protocol, we need a digital way for the prover to send a graph with "the nodes colored, but covered by hats" to the verifier. In the second step, the verifier will pick a random edge, just as before. In the final step, there has to be some way for the verifier to check that the two color revealed by the prover (as the colors of the nodes connected by the chosen edge) are truly the ones "under the hats" in the first step. How do we know that this is even possible?

We now need a new cryptographic tool called a *bit commitment scheme*. In a bit commitment scheme, there are two parties: a sender S and a receiver R.

Commit Phase: The sender "commits" to a bit.

1. S chooses a bit $\in \{0, 1\}$.
2. S runs Commit$(b, r) = C$ where r is some randomness chosen by S.
3. S sends C to R.

Open/Reveal Phase: The sender "opens" the commitment and the receiver verifies this opening.

1. S reveals the values (b, r), where b is the bit in the commitment and r is the randomness used to form the commitment.
2. R checks that Decommit(C, b, r) =accept; that is, that C is really a valid commitment to bit b using randomness r.

Properties that we want:

Hiding: "R cannot tell a commitment of 0 from a commitment of 1."

Binding: "A cheating S cannot open a commitment two ways." ie. A sender is bound to the value it commits to during the commit phase.

Now we will begin an example bit commitment scheme based on the Quadratic Residuosity Assumption (QRA) [GM84]. First, we will define the QRA.

Definition 3.1 (Quadratic Residuosity Assumption (QRA)) *Let p and q be large k-bit primes, where k is the security parameter. Set $n = pq$. Randomly select a value $a \stackrel{\$}{\leftarrow} Z_n^*$ with Jacobi symbol 1. Then for all PPT adversaries given (n, a) it is hard to decide if a is a* quadratic residue *or not; that is, if there exists a value $b \in \mathbb{Z}_n^*$ such that $b^2 \equiv a \pmod{n}$.*

Recall that Z_n^* is the multiplicative group consisting of the integers mod n (and excluding 0).

We will not define Jacobi symbols today. (This information can be found in many standard textbooks.) We will instead satisfy ourselves with the following two important facts:

1. There exists an efficient algorithm for computing Jacobi symbols.
2. Roughly half of Z_n^* have Jacobi symbol 1.

Under the QRA, it is hard to decide whether a is a quadratic residue or not. (The question is whether or not $\exists b \in Z_n^*$ such that $b^2 \equiv a \; mod \; n$). Out of this hard problem, we will now continue to give the details of the bit commitment scheme.

Public parameters: (n, x) such that $x \in Z_n^*$ is *not* a quadratic residue.

To Commit(0):

1. S picks a random $r \in Z_n^*$
2. S outputs $C = r^2 \; mod \; n$

To Commit(1):

1. S picks a random $r \in Z_n^*$
2. S outputs $C = r^2 \cdot x \; mod \; n$ (Note, if x is not a square, then $r^2 \cdot x$ is not a square either.)

To Open:

1. S sends (b, r) where $b \in \{0, 1\}$
2. R checks that $r^2 \cdot x^b \; mod \; n \equiv C$. If this checks, he accepts, otherwise, he rejects.

As far as hiding, if R can distinguish commitments of 0 from commitments of 1, then R is essentially distinguishing quadratic residues from non-residues, which is hard under the QRA. (Just like RSA, there is an assumed hardness of the underlying problem.)

The two categories of hiding:

Perfectly Hiding: Even if an all-powerful adversary is trying to find some difference between commitments of 0 and commitments of 1, he will not be able to do so. The distributions are exactly the same.

Computationally Hiding: For any PPT adversary, it is not possible to determine a difference between commitments of 0 and commitments of 1. The distributions may be far apart, but they are computationally indistinguishable.

The two categories of binding:

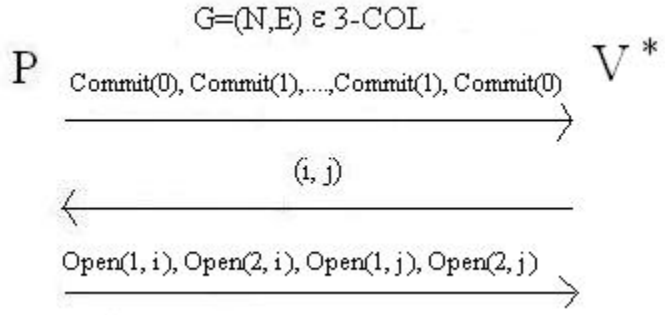
Perfectly Binding: Even an all-powerful prover cannot open a commitment to 0 as a commitment to 1 (or vice versa). (Technically, we might allow a very small probability of error based on the coins, if any, used by the receiver to check the opening, but this can be ignored for right now.)

Computationally Binding: For any PPT adversary, it is not possible to open a commitment in two different ways.

Our specific quadratic residue example is computationally hiding (the distributions are also disjoint since there is no overlap between squares and non-squares) and perfectly binding (no way to make a square look like a non-square or the other way around. It is not possible to give a square root of a non-square).

4 A Protocol for 3COL using Bit Commitments

Below is an example bit commitment scheme for the graph 3 coloring. Assume code 00=red, 01=blue, 10=green, and 11 is not used.



Note, each two "Commit"s represents the color of one node. Also, "Open" includes the committed bit and the random r.

If determining whether something is a quadratic residue is in fact a hard problem, then we can do zero knowledge proofs for everything in NP and if not, we can use hats. Instead of relying on the QRA, any one way function can be used. A one way function is a function that is easy to compute in one direction but very hard in the other.

We must prove to ourselves that the commitment scheme we just described is really a zero knowledge proof. We can see that it is complete. It is also sound with a negligible

probability of error when run $|E|^2$ times. (This is the same analysis as when using the hats, because even the unbounded prover cannot change his commitments after step one, since the bit commitment scheme is perfectly binding.) Lastly, does it have the property of zero knowledge? Consider the simulator (from before) that randomly colors each node of the graph and commitments to these colors. Once the verifier asks for the opening of an edge, the simulator opens the two nodes if their colors are different; otherwise, the simulator rewinds and starts over. Since the nodes are colored randomly, the simulator has a 2/3 chance of being able to answer the challenge. Thus, the expected running time is less than $2k$ rewinds to obtain k valid runs. So, creating *similar* conversations does not seem to be a problem, but is this simulated VIEW *exactly* the same as the real view?

No! The real and simulated VIEWs are not perfectly indistinguishable. The difference lies with an honest prover having a bunch of communications which are ALL valid, and a simulator producing many invalid first lines of communications. That is, the real prover always commits to a valid 3-coloring and the simulator may never commit to a valid 3-coloring. Therefore, this protocol would not satisfy our definition of ZKness stated below. We do not have *perfect* zero knowledge.

$$\forall \mathsf{V}^*_{PPT}\ \exists \mathsf{S}_{PPT}\ \forall x \in L\ \forall a \in \{0,1\}^{poly(|x|)}\ VIEW_{\mathsf{P},\mathsf{V}^*(a)}(x) = \mathsf{S}(a,x)$$

The obvious question is then, if a PPT distinguisher cannot tell the difference, should it matter that the distributions are not exactly the same? Maybe we ought to relax our definition of zero knowledge. Instead of the two distributions being exactly the same, should they just need to be computationally indistinguishable? ie. For a PPT verifier, $VIEW_{\mathsf{P},\mathsf{V}^*(a)}(x) \approx \mathsf{S}(a,x)$. We will talk about this more next class.

References

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.