Lecture 15 - Zero Knowledge (continued)

Boaz Barak

November 10, 2005

Note about HO7 Ex3 Clarified security definition.

From prev class. Finish proof that Protocol QR is zero knowledge.

Reducing error To reduce soundness error from $1/2$ to 2^{-k}, we repeat the protocol k times one after the other. The verifier accepts the combined proof only if all repetitions are valid. It is immediate that the combined proof is still complete. It is not immediate but not hard to prove that the combined proof has soundness error 2^{-k}: we can without loss of generality think of a cheating prover P^* as deterministic. Now, if $x \notin L$ then the verifier has in each iteration probability $1/2$ of rejecting and this probability is over the verifier's coins. Now, because the verifier uses independent coins in each execution these are independent events and so the overall probability that the verifier accepts k times is at most 2^{-k}. Zero knowledge is also preserved, albeit the proof here is a bit more complicated and uses the hybrid argument, see Goldreich's book for the proof.[1]

Parallel repetition. It is tempting to try to get a combined proof that uses much fewer rounds of interaction by *parallel repetition*. That is, the prover and verifier run k independent copies of the basic protocol in parallel, and the verifier accepts only if all of the copies accept. Again, completeness is immediate. Soundness is not so immediate but it turns out that indeed also in this case the soundness error is 2^{-k}. The zero knowledge property is a different story. It is known that for some protocol, parallel repetition ruins the zero knowledge property. However, it is not known whether or not the parallel version of Protocol QR is zero knowledge. This is in my opinion one of the fascinating open questions in crypto today, because it is intimately connected with the question of the security of a very popular heuristic called the "random oracle model" which we'll learn about later in the course. Note that it is known that if the parallel version of Protocol QR is zero knowledge then the simulator for it must be more sophisticated than the simulator for Protocol QR in the sense that it can not use the strategy V^* as a "black box" but must make essential use of the actual circuit V^* uses.

Zero knowledge for 3-coloring. We gave a ZK proof for the language QR of (x, n) such that $x \in QR_n$. We'll now give a ZK proof (due to Goldreich, Micali and Wigderson) for a different language - the set of 2 colorable graphs. That is, we say that a graph $G(V, E)$ on n vertices is in $3COL$ if there is a function $c : V \to \{R, G, B\}$ such that for every edges $(u, v) \in E$, $c(u) \neq c(v)$.

[1] Note that because our computational model is non-uniform circuits, this is equivalent to Goldreich's definition of auxiliary-input zero knowledge.

Why is this interesting. Intuitively, it seems that the language of quadratic residues is more interesting to crypto than $3COL$ and in some sense it is. Then, why are we interested in a protocol for $3COL$?

The reason is that a protocol for $3COL$ actually implies a protocol for QR and for almost any other language we are interested in, because $3COL$ is **NP**-complete. For example, we'll show why it implies a ZK protocol for QR:

The fact that $3COL$ is **NP**-complete means that we have a function red that on input (x, n) gives a graph G such that $x \in QR_n$ iff G is 3 colorable. Thus, if we want to prove in ZK that $(x \in QR_n$ we can use that reduction to obtain a graph G and prove that G is 3 colorable.

An important point is that, although this is not usually stressed, the standard NP-completeness reductions also reduce the solution or witness from one problem to the other. That is, along with the function $red(\cdot)$ we also have a function red' that maps a number w such that $w^2 = x$ to a 3-coloring $c : V \to \{R, G, B\}$ of the graph $G = red(x, n)$. This can be used for the prover to convert their private input into an input appropriate for the $3COL$ protocol.

Other interesting NP statements. Once we can prove any language in **NP** we can have protocols like this:

- Alice sends Bob a number n and proves in ZK that it $n = pq$ for two primes p, q with $p \pmod 4 = q \pmod 4 = 3$.
- Suppose that the encryption of Alice's tax return data is available on the web, and Alice wants to persuade Bob to give her a grant without opening all of the encryption. She can prove in zero knowledge that the bottom line is that she earned less than $10K$.
- Alice can send a string y to Bob and prove that this string is a commitment one of the two following strings "Eva" or "Fantasia" without Bob knowing which one it is.

Commitment schemes Before describing the protocol, let's remind ourselves what is a *commitment scheme*. A commitment scheme is a function $\mathsf{Com} : \{0,1\}^\ell \times \{0,1\}^n \to \{0,1\}^{kn'}$ satisfying the following properties

Hiding / Secrecy / Indistinguishability For every $x, x' \in \{0,1\}^\ell$, $\mathsf{Com}(x, U_n)$ is computationally indistinguishable from $\mathsf{Com}(x', U_n)$. (Note this is the same as the indistinguishability property for encryption scheme, and implies that given $y = \mathsf{Com}(x, U_n)$ an adversary can't learn any new information about x.)

Binding For every y there exists at most a *single* x such that $y = \mathsf{Com}(x, r)$ for some $r \in \{0,1\}^n$. (This implies that it is not possible to come up with two different pairs x, r and x', r' with $x \neq x'$ that yield y.)

We'll use a commitment scheme for messages of length 2, which we'll think of as numbers between 0 and 3. We'll use n bits of randomness for the commitment. If x is some message, we denote by $\mathsf{Com}(x)$ the random variable $\mathsf{Com}(x, U_n)$.

A ZK protocol for $3COL$ We now describe Protocol 3COL. The public input is a graph $G(V, E)$ of n vertices and m edges (with $m \leq n^2$). The prover also gets as a private input a function $c : V \to \{R, G, B\}$ such that for every $(u, v) \in E$, $c(u) \neq c(v)$.

P $\rightarrow$ **V** Prover chooses a random 1-to-1 function $\psi : \{R, G, B\} \rightarrow \{1, 2, 3\}$. It defines $c' : V \rightarrow \{1, 2, 3\}$ to be the $\psi \circ c$ (i.e., for every $v \in V$, $c'(v) = \psi(c(v))$). It computes $y_1, \ldots, y_n$ in the following way y_i is a commitment to $c'(v_i)$ where v_i is the i^{th} vertex. Prover then sends $y_1, \ldots, y_n$ to the verifier.

P $\leftarrow$ **V** Verifier chooses a random edge $(v_i, v_j) \leftarrow_{\mathrm{R}} E$ and sends (v_i, v_j) to the prover.

P $\rightarrow$ **V** Prover opens the commitments y_i and y_j and sends this information to the verifier. That is, it sends $r_i, r_j \in \{0, 1\}^n$ and $x_i, x_j \in \{1, 2, 3\}$ such that **(*)** $y_i = \mathsf{Com}(x_i, r_i)$, $y_j = \mathsf{Com}(x_j, r_j)$.

Verification Verifier accepts if and only if the openings are valid (i.e., satisfy **(*)** above), $x_i, x_j \in \{1, 2, 3\}$ and $x_i \neq x_j$.

Completeness. Completeness is again pretty immediate.

Soundness. We're going to show very low soundness error for this protocol: that if G is not 3-colorable, then the verifier will reject with probability at least $1 - 1/m$ where m is the number of edges. However, this is enough since if we repeat the protocol mk times we'll get soundness error $(1 - 1/m)^{mk} \sim 2^{-k}$.

Lemma 1. *Suppose that G is* not *3-colorable. Then, the verifier will reject with probability at least $1 - 1/m$ where m is the number of edges in the graph.*

Proof. By the binding property of the commitment scheme, for every y there's at most a single value $x \in \{0, 1, 2, 3\}$ such that there exists r with $\mathsf{Com}(x, r) = y$. Let's define this value x as $\mathsf{Com}^{-1}(y)$ (if there's no such x, define $\mathsf{Com}^{-1}(y) = \perp$. Note that the function Com^{-1} is not efficiently computable, but it is still mathematically well defined.

Let G be a non-3-colorable graph, and let P^* be a possibly cheating prover strategy for G and let $y_1, \ldots, y_n$ be its output on the empty string (i.e., it's first message). We define a coloring function $c : V \rightarrow \{R, G, B\}$ in the following way: for every vertex v_i, we consider $x_i = \mathsf{Com}^{-1}(y)$ if $x_i = 1$ we let $c(v_i) = R$, if $x_i = 2$ we let $c(v_i) = G$ and if $x_i = 3$ we let $c(v_i) = B$. If $x_i = \perp$ or $x_i = 0$ then we pick $c(v_i)$ arbitrarily (say $c(v_i) = R$).

Now the graph is not 3-colorable and hence there exists an edge (v_i, v_j) such that $c(v_i) = c(v_j)$. With probability at least $1/m$, the verifier will choose this edge. We claim that in this case the verifier will surely reject. Indeed, the prover P^* can either not open the commitments (in which case the verifier rejects) or (if $\mathsf{Com}^{-1}(y_i)$ and $\mathsf{Com}^{-1}(y_j)$ are not $\perp$) send x_i and x_j. Now if one of the x_i or x_j equals 0 then the verifier will reject. However, if $x_i, x_j \in \{1, 2, 3\}$, then since $c(v_i) = c(v_j)$ we know that $x_i = x_j$ and hence the verifier will reject. □

Zero Knowledge The simulator for our protocol will be in some sense similar to the simulator of Protocol QR, although in this case we'll have only computational indistinguishability and not statistical indistinguishability. The simulator will do the following:

Algorithm S

1. Input: G a graph on n vertices and m edges.
2. Guess a random edge $(i', j') \leftarrow_{\mathrm{R}} E$.
3. Choose c_1 at random from $\{1, 2, 3\}$ and choose c_2 at random from $\{1, 2, 3\} \setminus \{c_1\}$.

4. For every $1 \leq i \leq n$, compute y_i as follows: if $i \notin \{i', j'\}$ then $y_i = \mathsf{Com}(0)$ (i.e., commitment to 0 with fresh independent coins). If $i = i'$ then $y_i = \mathsf{Com}(c_1)$ and if $i = j'$ then $y_i = \mathsf{Com}(c_2)$.
5. Compute $(i, j) = V^*(y_1, \ldots, y_n)$ (i.e., feed the message $y_1, \ldots, y_n$ to V^* to obtain its response which we can always interpret as an edge $(i, j) \in E$.
6. If $(i, j) \neq (i', j')$ then go back to Step 2.
7. Otherwise, compute z to be the openings of y_i and y_j and output the transcript $\langle y_1 \cdots y_n, (i, j), z \rangle$.

Proof that simulator works To prove that S is a valid simulator we'll construct a *hybrid simulator* HS will get as extra input the witness a valid coloring $c : V \to \{1, 2, 3\}$ (this is fine since HS is just a tool for the proof). We will prove that **(1)** The output of HS is indistinguishable from the output of S and **(2)** The output of HS is indistinguishable from a transcript in which V^* interacts with the honest prover.

Algorithm HS

1. Input: G a graph on n vertices and m edges. S' also gets $c : V \to \{R, G, B\}$ such that $c(u) \neq c(v)$ for all $(u, v) \in E$.
2. Guess a random edge $(i', j') \leftarrow_{\mathrm{R}} E$.
3. Choose c_1 at random from $\{1, 2, 3\}$ and choose c_2 at random from $\{1, 2, 3\} \setminus \{c_1\}$.
4. Let $\psi : \{R, G, B\} \to \{1, 2, 3\}$ be the unique one-to-one function such that $\psi(c(v_{i'})) = c_1$ and $\psi(c(v_{j'})) = c_2$.
5. For every $1 \leq i \leq n$, compute y_i as follows: $y_i \mathsf{Com}(\psi(c(v_i)))$, Note that $y_{i'} = \mathsf{Com}(c_1)$, $y_{j'} = \mathsf{Com}(c_2)$.
6. Compute $(i, j) = V^*(y_1, \ldots, y_n)$ (i.e., feed the message $y_1, \ldots, y_n$ to V^* to obtain its response which we can always interpret as an edge $(i, j) \in E$.
7. If $(i, j) \neq (i', j')$ then go back to Step 2.
8. Otherwise, compute z to be the openings of y_i and y_j and output the transcript $\langle y_1 \cdots y_n, (i, j), z \rangle$.

It's not hard to see that ψ is a random one-to-one mapping from $\{R, G, B\}$ to $\{1, 2, 3\}$ and hence **(1)** the sequence $(y_1, \ldots, y_n)$ is independent from the choice of (i', j') and hence $(i', j') = (i, j)$ with probability at least $1/m$ and **(2)** the output of HS is identical to the transcript of an interaction between V^* and the honest prover.

We'll show that any difference in behavior (whether it is running time or output distribution) between HS and S will contradict the security of the commitment scheme. We show this in the following way:

For i between 1 and n, define $S_i(G, c)$ as follows: act exactly like $S(G)$ except that when computing the commitments $y_1, \ldots, y_n$, the first i commitments that are not $y_{i'}, y_{j'}$ will be computed to be commitments to the same values as HS does (and not to zero). Clearly, the output $S_0(G, c)$ is identical to the output of $S(G)$ and the output of $S_{n-2}(G, c)$ is identical to the output of $HS(G, c)$. Thus, it is enough to prove that for any i, $S_i(G, c)$ is indistinguishable from $S_{i-1}(G, c)$. However, this follows immediately from the hiding property commitment scheme: assume otherwise, and define the following distinguisher: given an input y that is either $\mathsf{Com}(0)$ or $\mathsf{Com}(\psi(c(v_i)))$, define $\hat{S}(G, c, y)$ to be the following algorithm: use the first $i - 1$ commitments as HS does, for the i^{th} commitment use y, and for the rest use commitments to zero. We see that if we can distinguish between $S_i(G,c)$ and $S_{i-1}(G,c)$ then we can use $\hat{S}(G,c,y)$ as a distinguisher between $\mathsf{Com}(0)$ or $\mathsf{Com}(\psi(c(v_i)))$.

Summary We have the following definition for a ZK proof:

Definition 1. Let L be a language in **NP** and let R be its corresponding witness relation (that is, $x \in L$ if and only if there's some w such that $(x,w) \in R$). A proof system (P,V) is *a zero knowledge proof* for L with (T,ϵ)-zero knowledge and soundness error δ if it satisfies the following:

Completeness If $(x,w) \in R$ and the public input is x and the prover P is given w as private input then the verifier will accept with probability one.

Soundness with error δ If $x \notin L$ then for every possibly cheating prover P^*, the probability that $\mathsf{out}_V\langle P^*, V_{x,r}\rangle = \mathsf{accept}$ is at most δ, where this probability is taken over the random choices r of the verifier.

(T,ϵ)**-Zero knowledge** For every T-time cheating strategy V^* there exists a poly(T)-time non-interactive algorithm S such that for every $(x,w) \in R$ the following two random variables are (T,ϵ)-computationally indistinguishable:

- $view_{V^*}\langle P_{U_m,x,w}, V^*\rangle$. (Where m is the number of random coins P uses
- $S(x)$. (Note that S is probabilistic and so this is a random variable).

Proofs of knowledge A notion we did not talk a lot about in class is proof of knowledge. Basically this is a stronger form of soundness that says that if P^* convinces the verifier with noticeable probability (i.e., more than the soundness error), then not only this means that the statement x is in L but it actually means that P^* "knows" a witness in the sense that it could obtain a witness by running some algorithm. This is often useful for proving security of identification protocol where simple soundness falls short of what we need to make the proof work.

We say that (P,V) is a *proof of knowledge* if soundness is replaced by the following stronger requirement:

Knowledge soundness with error δ For every possibly cheating prover P^*, and every x, if P^* satisfies that

$$\Pr[\mathsf{out}_V\langle P^*, V_{x,r}\rangle = \mathsf{accept}] > \delta + \rho$$

(where this probability is taken over the random choices r of the verifier)

then there's a algorithm E (called a knowledge extractor) with running time polynomial in $1/\rho$ and the running time of P^*, that on input x outputs a witness w for x (i.e. w such that $(x,w) \in R$) with probability at least 1/2. Note this indeed implies normal soundness with soundness error δ.

Main Theorem Using the $3COL$ protocol, we have the following theorem:

Theorem 2. *Assume that (T,ϵ)-secure commitment schemes exist. Let L be any language in* **NP**. *Then, there exists a zero knowledge protocol for proving that $x \in L$ where the prover and verifier run in* poly$(|x|)$ *time, the soundness error is $2^{-|x|}$ and it is (T',ϵ')-zero knowledge for T',ϵ' polynomially related to T,ϵ. Furthermore this protocol satisfies the stronger condition of proof of knowledge.*