Lecture 18 - Zero Knowledge (cont), identification protocols

Boaz Barak

April 7, 2010

Reading Boneh-Shoup chapter 18.

Review Protocol QR

Statement x is a quadratic residue mod n.

Public input x , n

Prover's (Alice) private input. w such that $x = w^2 \pmod n$.

P $\rightarrow$ **V** Alice chooses random $u \leftarrow_R \mathbb{Z}_n^*$ and sends $y = u^2$ to Bob.

P $\leftarrow$ **V** Bob chooses $b \leftarrow_R \{0,1\}$

P $\rightarrow$ **V** If $b = 0$, Alice sends u to Bob. If $b = 1$, Alice sends $w \cdot u \pmod n$.

Verification. Let z denote the number sent by Alice. Bob *accepts* the proof in the case $b = 0$, $z^2 = y \pmod n$. In the case $b = 1$, Bob accepts the proof if $z^2 = xy \pmod n$.

Simulator Uses same for $b = 0$, uses $y = z^2 x^{-1}$ for $b = 1$.

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 (in our example w is a square root of c) with probability at least $1/2$. Note this indeed implies normal soundness with soundness error δ.

Note that proof of knowledge condition makes sense even if there's no question if the statement is true, as we'll see below.

Parallel repetition. We can show that parallel repetition reduces the soundness to 2^{-k} by giving the even stronger statement: let $\vec{b} \in \{0,1\}^k$ denote the queries the verifier makes in k copies of the parallel repeated version. For every fixed k initial messages $y_1, ..., y_k$, and $\vec{b} \neq \vec{b'}$, one can obtain the square root of x from two accepting transcripts of the form $(y_1 \cdots y_k; \vec{b}; z_1 \cdots c_k)$ and $(y_1 \cdots y_k; \vec{b'}; z_1 \cdots c_k)$. This allows to show proof of knowledge using *rewinding*.

A three round protocol that is (perfect) honest verifier zero knowledge and has this knowledge soundness property is called a Σ protocol.

Honest verifier zero knowledge Parallel repetition does preserve honest verifier zero knowledge.

Schnorr's identification protocol Schnorr suggested the following proof of knowledge for the discrete logarithm:

Review Protocol Schnorr

Statement Alice knows discrete log of h w.r.t. g, where these are members of some group G of order q, and g is a generator.

Public input g, h

Prover's (Alice) private input. x such that $h = g^x$.

P $\rightarrow$ **V** Alice chooses random $r \in \mathbb{Z}_q$, and sends $a = g^r$.

P $\leftarrow$ **V** Bob chooses $b \leftarrow_{\mathrm{R}} \mathbb{Z}_q$ and sends b to Alice.

P $\rightarrow$ **V** Alice sends $c = r + xb \pmod q$ to Bob.

Verification. Bob verifies that $ah^b = g^c$.

Completeness obvious.

Proof of knowledge We have the Σ condition: if $b \neq b'$ then given a and $b \neq b'$ and $c \neq c'$ such that $ah^b = g^c$ and $ah^{b'} = g^{c'}$ we divide the two equations by each other to get $h^{b-b'} = g^{c-c'}$ but since we know b, b' we can take this to the power $(b - b')^{-1} \pmod q$ to get an equation of the form $h = g^x$.

Honest verifier zero knowledge The simulator does the following: choose $b, c \leftarrow_{\mathrm{R}} \mathbb{Z}_q$, choose a as $h^{-b}g^c$.

Application for identification w.r.t passive adversaries

Random oracle making Σ protocols non-interactive.

OR trick We now show how to transform a Σ protocol into an identification protocol that is secure w.r.t. active adversaries. We suppose we have a Σ protocol for proving knowledge of x given $f(x)$, where f is a one-way function. We show the following **OR Protocol** to prove, given y_1, y_2 that Alice knows *either* a preimage for y_1 *or* a preimage for y_2 (or possibly both). We assume there is a Σ protocol Π with messages (a, b, c) for proving knowledge of one preimage.

We first describe the protocol from the verifier's (Bob) point of view, without explaining how Alice implements her strategy.

Public input y_1, y_2

Alice (prover's) private input either x_1 such that $y_1 = f(x_1)$ or x_2 such that $y_2 = f(x_2)$.

First message Alice sends two messages a_1, a_2 for two instances of the Σ protocol: one to prove she knows x_1 and the other to prove she knows x_2. (Note that she doesn't know both.)

Second message Bob chooses b at random from the challenge space of the Σ protocol (say $\mathrm{GF}(2)^k$ or $\mathbb{Z}_q$) and sends b to Alice.

Third message Alice chooses b_1, b_2 such that $b_1 + b_2 = b$ and continues both interactions as if Bob sent b_1 in the first instance and b_2 in the second instance. Bob verifies that both instances accept.

Completeness Here is one of the rare examples that even completeness is not trivial— how does Alice manage to succeed in two instances where she only knows one witness? The idea is the following: suppose that she knows only x_2. She will choose b_1 at random herself, and use the *simulator* to compute her first message a_1 in the first instance. Then when Bob sends her b, she will just compute $b_2 = b - b_1$ and continue in both instances, in the first using the simulator and the second using the real prover strategy.

Soundness and knowledge soundness This is still a Σ protocol: suppose that we have fixed messages a_1, a_2 and $b \neq b'$ such that Alice can find b_1, b_2, b'_1, b'_2 and c_1, c_2, c'_1, c'_2 such that $b = b_1 + b_2$, $b' = b'_1 + b'_2$ and all the following four transcripts are accpeting:

$$(a_1, b_1, c_1)$$
$$(a_1, b'_1, c'_1)$$
$$(a_2, b_2, c_2)$$
$$(a_2, b'_2, c_2)$$

we claim that it must be the case that either $b'_1 \neq b_1$ or $b_2 \neq b'_2$. Indeed, otherwise we'd have $b'_1 = b_1$ and $b'_2 = b_2$ and hence $b_1 + b_2 = b'_1 + b'_2$.

Honest verifier zero knowledge This is also honest verifier ZK, but this is actually not the important property we need from this protocol.

Security against cheating verifiers. We show the following result:

Lemma 1. *Let Bob* be any (possibly) cheating polynomial-time verifier, and suppose that x_1, x_2 are chosen at random and we set $y_i = f(x_i)$ then for $j = 1, 2$ and $i = 1, 2$ the probability that Bob* outputs a preimage of y_i when interacting with Alice that gets as input x_j is negligible.*

Proof. Let's prove the lemma for $i = 2$ (the proof for $i = 1$ is symmetric). The main point is the following: no matter what Bob does, the interaction with Alice that gets as input x_1 and the interaction with Alice that gets as input x_2 is *identical*. In both cases, Bob gets exactly the same distribution as if Alice would have run the honest prover for both instances and chosen at random b_1, b_2 such that $b_1 + b_2 = b$. So it suffices to prove this for $i = 2$ and $j = 1$. But then neither Bob nor Alice get the preimage of y_2, and if Bob could obtain such a preimage with non-negligible probability then we can combine both Bob and Alice to get an invertor for the one-way function f. □

Zero knowledge protocol for Hamiltonian cycle