Notes for Lecture 26 (draft)

Summary

After showing that last week's protocol for quadratic residuosity is indeed zero-knowledge, we move on to the formal definition of *proof of knowledge*, and we show that the quadratic residuosity protocol is also a proof of knowledge.

1 The Quadratic Residuosity Protocol

Last time we considered the following protocol for quadratic residuosity:

- Verifier's input: an integer N (product of two unknown odd primes) and a integer $r \in \mathbb{Z}_N^*$;
- Prover's input: N, r and a square root $x \in Z_N^*$ such that $x^2 \bmod N = r$.
- The prover picks a random $y \in Z_N^*$ and sends $a := y^2 \bmod N$ to the verifier
- The verifier picks at random $b \in \{0, 1\}$ and sends b to the prover
- The prover sends back $c := y$ if $b = 0$ or $c := y \cdot x \bmod N$ if $b = 1$
- The verifier cheks that $c^2 \bmod N = a$ if $b = 0$ or that $c^2 \equiv a \cdot r \pmod N$ if $b = 1$, and accepts if so.

Today we show that it is zero knowledge, that is,

Theorem 1 *For every verifier algorithm V^* of complexity $\leq t$ there is a simulator algorithm of average complexity $\leq 2t + (\log N)^{O(1)}$ such that for every odd composite N, every r which is a quadratic residue $\pmod N$ and every square root of x of r, the distributions*

$$S^*(N, r) \tag{1}$$

and

$$P(N, r, x) \leftrightarrow V^*(N, r) \tag{2}$$

are identical.

2 Proofs of Knowledge

Suppose that L is a language in NP; then there is an NP relation $R_L(\cdot,\cdot)$ computable in polynomial time and polynomial $p(\cdot)$ such that $x \in L$ if and only if there exists a witness w such that $|w| \leq p(|x|)$ (where we use $|z|$ to denote the length of a bit-string z) and $R(x,w) = 1$.

Recall the definition of *soundness* of a proof system (P,V) for L: we say that the proof system has soundness error at most ϵ if for every $x \notin L$ and for every cheating prover strategy P^* the probability that $P^*(x) \leftrightarrow V(x)$ accepts is at most ϵ. Equivalently, if there is a prover strategy P^* such that the probability that $P^*(x) \leftrightarrow V(x)$ accepts is bigger than ϵ, then it must be the case that $x \in L$. This captures the fact that if the verifier accepts then it has high confidence that indeed $x \in L$.

In a proof-of-knowledge, the prover is trying to do more than convince the verifier that a witness exists proving $x \in L$; he wants to convince the verifier that he (the prover) *knows* a witness w such that $R(x,w) = 1$. How can we capture the notion that an algorithm "knows" something?

Definition 2 (Proof of Knowledge) *A proof system (P,V) for an NP relation R_L is a proof of knowledge with knowledge error at most ϵ and extractor slowdown es if there is an algorithm K (called a knowledge extractor) such that, for every verifier strategy P^* of complexity $\leq t$ and every input x, if*

$$\mathbb{P}[P^*(x) \leftrightarrow V(x) \text{ accepts}] \geq \epsilon + \delta$$

then $K(P^*,x)$ outputs a w such that $R(x,w) = 1$ in average time at most

$$es \cdot (n^{O(1)} + t) \cdot \delta^{-1}$$

In the definition, giving P^* as an input to K means to give the code of P^* to K. A stronger definition, which is satisfied by all the proof systems we shall see, is to let K be an oracle algorithm of complexity $\delta^{-1} \cdot es \cdot poly(n)$, and allow K to have oracle access to P^*. In such a case, "oracle access to a verifier strategy" means that K is allowed to select the randomness used by P^*, to fix an initial part of the interaction, and then obtain as an answer what the next response from P^* would be given the randomness and the initial interaction.

Theorem 3 *The protocol for quadratic residuosity of the previous section is a proof of knowledge with knowledge error $1/2$ and extractor slowdown 2.*