Notes for Lecture 25

Scribed by Alexandra Constantin, posted May 4, 2009

Summary

Today we show that the graph isomorphism protocol we defined last time is indeed a zero-knowledge protocol. Then we discuss the *quadratic residuosity problem* modulo a composite, and define a protocol for proving quadratic residuosity. (We shall prove that the protocol is zero knowledge next time.)

1 The Graph Isomorphism Protocol

Last time we considered the following protocol for the graph isomorphism problem.

- Verifier's input: two graphs $G_1 = (V, E_1)$, $G_2 = (V, E_2)$;
- Prover's input: G_1, G_2 and permutation π^* such that $\pi^*(G_1) = G_2$; the prover wants to convince the verifier that the graphs are isomorphic
- The prover picks a random permutation $\pi_R : V \to V$ and sends the graph $G := \pi_R(G_2)$
- The verifier picks at random $b \in \{1, 2\}$ and sends b to the prover
- The prover sends back π_R if $b = 2$, and $\pi_R(\pi^*(\cdot))$ otherwise
- The verifier cheks that the permutation π received at the previous round is such that $\pi(G_b) = G$, and accepts if so.

In order to prove that this protocol is zero knowledge, we have to show the existence of an efficient simulator.

Theorem 1 (Honest-Verifier Zero Knowledge) *There exists an efficient simulator algorithm S^* such that, for every two isomorphic graphs G_1, G_2, and for every isomorphism π between them, the distributions of transcripts*

$$P(\pi, G_1, G_2) \leftrightarrow Ver(G_1, G_2) \tag{1}$$

and

$$S(G_1, G_2) \tag{2}$$

are identical, where P is the prover algorithm and Ver is the verifier algorithm in the above protocol.

PROOF:

Algorithm S on input G_1, G_2 is described as follows:

- Input: graphs G_1, G_2
- pick uniformly at random $b \in \{1, 2\}$, $\pi_R : V \to V$
- output the transcript:
  1. prover sends $G = \pi_R(G_b)$
  2. verifier sends b
  3. prover sends π_R

At the first step, in the original protocol we have a random permutation of G_2, while in the simulation we have either a random permutation of G_1 or a random permutation of G_2; a random permutation of G_1, however, is distributed as $\pi_R(\pi^*(G_2))$, where π_R is uniformly distributed and π^* is fixed. This is the same as a random permutation of G_2, because composing a fixed permutation with a random permutation produces a random permutation.

The second step, both in the simulation and in the original protocol, is a random bit b, selected independently of the graph G sent in the first round. This is true in the simulation too, because the distribution of $G := \pi_R(G_b)$ conditioned on $b = 1$ is, by the above reasoning, identical to the distribution of G conditioned on $b = 0$.

Finally, the third step is, both in the protocol and in the simulation, a distribution uniformly distributed among those establishing an isomorphism between G and G_b. □

To establish that the protocol satisfies the general zero knowledge protocol, we need to be able to simulate cheating verifiers as well.

Theorem 2 (General Zero Knowledge) *For every verifier algorithm V^* of complexity t there is a simulator algorithm S^* of expected complexity $\leq 2t + O(n^2)$ such that, for every two isomorphic graphs G_1, G_2, and for every isomorphism π between them, the distributions of transcripts*

$$P(\pi, G_1, G_2) \leftrightarrow V^*(G_1, G_2) \tag{3}$$

and

$$S^*(G_1, G_2) \tag{4}$$

are identical.

PROOF:

Algorithm S^* on input G_1, G_2 is described as follows:

Input G_1, G_2

1. pick uniformly at random $b \in \{1, 2\}$, $\pi_R : V \to V$

- $G := \pi_R(G_b)$
- let b' be the second-round message of V^* given input G_1, G_2, first message G
- if $b \neq b'$, abort the simulation and go to 1.
- else output the transcript
  - prover sends G
  - verifier sends b
  - prover sends π_R

As in the proof of Theorem 1, G has the same distribution in the protocol and in the simulation.

The important observation is that b' depends only on G and on the input graphs, and hence is statistically independent of b. Hence, $\mathbb{P}[b = b'] = \frac{1}{2}$ and so, on average, we only need two attempts to generate a transcript (taking overall average time at most $2t + O(n^2)$). Finally, conditioned on outputting a transcript, G is distributed equally in the protocol and in the simulation, b is the answer of V^*, and π_R at the last round is uniformly distributed among permutations establishing an isomorphism between G and G_b. $\square$

2 The Quadratic Residuosity Problem

We review some basic facts about quadratic residuosity modulo a composite.

If $N = p \cdot q$ is the product of two distinct odd primes, and $\mathbb{Z}_N^*$ is the set of all numbers in $\{1, \ldots, N - 1\}$ having no common factor with N, then we have the following easy consequences of the Chinese remainder theorem:

- $\mathbb{Z}_N^*$ has $(p-1)\cdot(q-1)$ elements, and is a group with respect to multiplication;

  PROOF:

  Consider the mapping $x \to (x \bmod p, x \bmod q)$; it is a bijection because of the Chinese remainder theorem. (We will abuse notation and write $x = (x \bmod p, x \bmod q)$.) The elements of $\mathbb{Z}_N^*$ are precisely those which are mapped into pairs (a, b) such that $a \neq 0$ and $b \neq 0$, so there are precisely $(p-1)\cdot(q-1)$ elements in $\mathbb{Z}_N^*$.

  If $x = (x_p, x_q)$, $y = (y_p, y_q)$, and $z = (x_p \times y_p \bmod p, x_q \times y_q \bmod q)$, then $z = x \times y \bmod N$; note that if $x, y \in \mathbb{Z}_N^*$ then x_p, y_p, x_q, y_q are all non-zero, and so $z \bmod p$ and $z \bmod q$ are both non-zero and $z \in \mathbb{Z}_N^*$.

  If we consider any $x \in \mathbb{Z}_N^*$ and we denote $x' = (x_p^{-1} \bmod p, x_q^{-1} \bmod q)$, then $x \cdot x' \bmod N = (x_p x_p^{-1}, x_q x_q^{-1}) = (1, 1) = 1$.

  Therefore, $\mathbb{Z}_N^*$ is a group with respect to multiplication. $\square$

- If $r = x^2 \bmod N$ is a quadratic residue, and is an element of $\mathbb{Z}_N^*$, then it has exactly 4 square roots in $\mathbb{Z}_N^*$

  PROOF:

  If $r = x^2 \bmod N$ is a quadratic residue, and is an element of $\mathbb{Z}_N^*$, then:

  $r \equiv x^2 \bmod p$

  $r \equiv x^2 \bmod q$.

  Define $x_p = x \bmod p$ and $x_q = x \bmod q$ and consider the following four numbers:

  $x = x_1 = (x_p, x_q)$

  $x_2 = (-x_p, x_q)$

  $x_3 = (x_p, -x_q)$

  $x_4 = (-x_p, -x_q)$.

  $x^2 \equiv x_1^2 \equiv x_2^2 \equiv x_3^2 \equiv x_4^2 \equiv r \bmod N$.

  Therefore, x_1, x_2, x_3, x_4 are distinct square roots of r, so r has 4 square roots. $\square$

- Precisely $(p-1)\cdot(q-1)/4$ elements of $\mathbb{Z}_N^*$ are quadratic residues

  PROOF:

  According to the previous results, $\mathbb{Z}_N^*$ has $(p-1)\cdot(q-1)$ elements, and each quadratic residue in $\mathbb{Z}_N^*$ has exactly 4 square roots. Therefore, $(p-1)\cdot(q-1)/4$ elements of $\mathbb{Z}_N^*$ are quadratic residues. $\square$

- Knowing the factorization of N, there is an efficient algorithm to check if a given $y \in \mathbb{Z}_N^*$ is a quadratic residue and, if so, to find a square root.

It is, however, believed to be hard to find square roots and to check residuosity modulo N if the factorization of N is not known.

Indeed, we can show that from any algorithm that is able to find square roots efficiently mod N we can derive an algorithm that factors N efficiently.

Theorem 3 *If there exists an algorithm A of running time t that finds quadratic residues modulo $N = p \cdot q$ with probability $\geq \epsilon$, then there exists an algorithm A^* of running time $t + O(\log N)^{O(1)}$ that factors N with probability $\geq \frac{\epsilon}{2}$.*

Proof: Suppose that, for a quadratic residue $r \in \mathbb{Z}_N^*$, we can find two square roots x, y such that $x \neq \pm y \pmod N$. Then $x^2 \equiv y^2 \equiv r \bmod N$, then $x^2 - y^2 \equiv 0 \bmod N$. Therefore, $(x-y)(x+y) \equiv 0 \bmod N$. So either $(x-y)$ or $(x+y)$ contains p as a factor, the other contains q as a factor.

The algorithm A^* is described as follows:

Given $N = p \times q$

- pick $x \in \{0 \ldots N-1\}$
- if x has common factors with N, return $\gcd(N, x)$
- if $x \in \mathbb{Z}_N^*$
  - $r := x^2 \bmod N$
  - $y := A(N, r)$
  - if $y \neq \pm x \bmod N$ return $gcd(N, x + y)$

WIth probability ϵ over the choice of r, the algorithm finds a square root of r. Now the behavior of the algorithm is independent of how we selected r, that is which of the four square roots of r we selected as our x. Hence, there is probability 1/2 that, conditioned on the algorithm finding a square root of r, the square root y satisfies $x \neq \pm y \pmod N$, where x is the element we selected to generate r. □

3 The Quadratic Residuosity Protocol

We consider the following protocol for proving 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.

We show that:

- If r is a quadratic residue, the prover is given a square root x, and the parties follow the protocol, then the verifier accepts with probability 1;
- If r is not a quadratic residue, then for every cheating prover strategy P^*, the verifier rejects with probability $\geq 1/2$.

PROOF:

Suppose r is not a quadratic residue. Then it is not possible that both a and $a \times r$ are quadratic residues. If $a = y^2 \bmod N$ and $a \times r = w^2 \bmod N$, then $r = w^2(y^{-1})^2 \bmod N$, meaning that r is also a perfect square.

With probability 1/2, the verifier rejects no matter what the Prover's strategy is.

□

Next time we shall prove that the protocol is zero knowledge.