COS 433 — Cryptography — Homework 9.

Boaz Barak

Total of 120 points. Due April 14th, 2007.

In the following two questions, we consider a zero knowledge proof system for proving statements of the form "$x \in L$" where L is a subset (also called "language") of $\{0,1\}^*$. (We're only concerned here with standard soundness and not knowledge soundness.) We want to show that unless it's easy to verify statements like that just from the public input x (in which case there's a trivial zero knowledge protocol where the prover doesn't say anything), then both interaction and randomization are necessary.

Exercise 1 (Interaction is necessary, 15 points)**.** Let L be a language that is not decidable in polynomial time (that is, there is no efficient (possibly probabilistic) algorithm that on input x outputs 1 if $x \in L$ and 0 otherwise). Show that there is no *non-interactive* zero knowledge proof system for L. That is, show that if a language L has a proof system that consists of a single message from the prover to the verifier then L is decidable by a polynomial-time algorithm.

Exercise 2 (Randomness is necessary, 15 points)**.** Let L be a language that is not decidable in polynomial-time. Show that there is no *deterministic* zero knowledge proof system for L. That is, show that if a language L has a proof system where the verifier is deterministic then L is decidable by a polynomial-sized algorithm.

In the following exercises we'll use the **Quadratic Residuosity Axiom**: the following two distributions (X, N) and (Y, N) are computationally indistinguishable where N is a random Blum integer (obtained by setting $N = PQ$ where P, Q are two random n bit primes satisfying $P, Q = 3 \pmod 4$), X is a random quadratic residue modulo N, and Y is a random quadratic non-residue modulo N of Jacobi symbol $+1$.

The Jacobi symbol of X modulo a prime P (also known as the Legendre symbol for this case), denoted by $\left(\frac{X}{P}\right)$, is $+1$ is X is a quadratic residue and -1 if X is not a quadratic residue. The Jacobi symbol of X modulo $N = PQ$, is $\left(\frac{X}{N}\right) = \left(\frac{X}{P}\right)\left(\frac{X}{Q}\right)$. There is a known polynomial-time algorithm to compute the Jacobi symbol $\left(\frac{X}{N}\right)$.

It can be easily verified that the set of $X \in \mathbb{Z}_N^*$ with $\left(\frac{X}{N}\right) = +1$ is a subgroup of $\mathbb{Z}_N^*$ of size $|\mathbb{Z}_N^*|/2$. The Chinese remaindering theorem implies that if X is a quadratic residue modulo N than $\left(\frac{X}{N}\right) = +1$, although the quadratic residues account for only $|\mathbb{Z}*_N|/4$ of the X's with $\left(\frac{X}{N}\right) = +1$.

Exercise 3 (15 points)**.** Prove that if N is a Blum integer, then -1 is a non-quadratic residue modulo N and $\left(\frac{-1}{N}\right) = +1$.

Exercise 4 (25 points)**.**

1. Prove that the following public key cryptosystem (G, E, D) is CPA secure under the Quadratic Residuosity axiom:

   Key generation Given security parameter n, let P, Q two n-bit prime random primes and let $N = PQ$. The public key is N and the secret key is P, Q.

Encrypt To encrypt a bit $b \in \{0,1\}$, choose $X \leftarrow_{\mathrm{R}} \mathbb{Z}_N^*$ and output $X^2(-1)^b \pmod{N}$.

Decrypt To decrypt $Y \in \mathbb{Z}_N^*$, output 0 if Y is a quadratic residue and 1 otherwise. (Knowing the factorization, quadratic residuosity can be tested using Chinese remaindering.)

2. Prove that there is an algorithm that given the public key N and two ciphertexts $Y, Y' \in \mathbb{Z}_N^*$ that decrypt to b, b', outputs Z such that Z is identically distributed to an encryption of $b \oplus b'$ (where $\oplus$ denotes XOR). This property is called being *homomorphic* with respect to XOR. Does this property contradict CPA security? how about CCA security?

This cryptosystem is due to Goldwasser and Micali.

Exercise 5 (50 points)**.** In the "cloud computing problem" we think of a user Alice that wishes to store a large database on the cloud of the server Bob, but doesn't wish Bob to learn anything about Alice's data. Specifically, we think of the database as just a very long string $A \in \{0,1\}^M$. We'll think of M as being much larger than the key size/security parameter n we use for our cryptosystems etc.. A *cloud computing protocol* consists of the following:

- (Uploading phase) Alice uploads the database to Bob by sending him a string $\hat{A}$. She may keep a small state of $\mathrm{poly}(n)$ bits to herself where n is the security parameter, but she does not have memory to store the entire M bit long database on her own.

- (Recovery phase) Later, if Alice wants to recover A_i for some $i \in [M]$, she sends a message $\hat{i}$ to Bob, and gets back a message $\hat{b}$ from Bob. She should be able to obtain x_i from $\hat{b}$ (and her own state) by some efficient procedure.

The security notion for this protocol is that for every $A, A' \in \{0,1\}^M$ and $i, i' \in [M]$, the messages that Alice sends when uploading A and querying i are indistinguishable from the messages she sends when uploading A' and querying i'. (This simplified security notion is just protecting against passive/eavesdropping attacks by Bob— in real life we'd want to protect against active attacks as well.) We require Bob, Alice to run in polynomial time in n, M.

1. Show that there exists a secure cloud computing protocol.

2. Consider the following variant of cloud computing, where we think of the database A as a $\sqrt{M}$ by $\sqrt{M}$ matrix over $\mathsf{GF}(2)$ and i as a vector in $\mathsf{GF}(2)^{\sqrt{M}}$, and Alice wished to recover the vector Ai. Show that there is a secure cloud computing protocol for this variant, where the length of the messages exchanged between Alice and Bob in the recovery phase is at most $\sqrt{M}\mathrm{poly}(n)$.

3. Show that there exists a secure cloud computing protocol (in the standard sense) where the length of the messages exchanged between Alice and Bob in the recovery phase is at most $\sqrt{M}\mathrm{poly}(n)$.