COS 433 — Cryptography — Homework 10.

Boaz Barak

Total of 140 points. Due April 21th, 2010.

Exercise 1 (100 points)**.** In this exercise you will prove that the protocol given in class is indeed a zero knowledge protocol. Note that that this question is worth 100 points, and correspondingly I expect a fully rigorous, and clearly written proof. You may find it useful to split the proof for some of the items into a few smaller claims. For the sake of completeness I provide below the definitions of homomorphic encryption and commitment schemes.

We consider the following protocol. Below (G, E, D) is a homomorphic encryption scheme with verifiable keys and ciphertexts, and Com is a commitment scheme.

Basic ZK:

Public input: A Boolean circuit $C : \{0,1\}^n \to \{0,1\}$. For concreteness, assume C has at most n^2 gates.

Prover's private input: $x \in \{0,1\}^n$ s.t. $C(x) = 1$.

Step 1 Prover runs $(e, d) \leftarrow G(1^n)$, sends e to verifier. Verifier checks e is in the range of G, and otherwise it rejects.

Step 2 Prover sends $\hat{x} = E_e(x_1) \cdots E_e(x_n)$ to verifier. Verifier checks each ciphertext $\hat{x}_i$ is in the range of E and otherwise it rejects.

Step 3 Verifier computes $\hat{c} = EVAL(C, \hat{x})$, sends $\hat{c}$ to prover.

Step 4 The prover computes $b' = D_d(c)$, sends $\hat{b'} = \mathsf{Com}(b')$ to verifier.

Step 5 Verifier sends all randomness it used in producing the ciphertext of Step 3. The prover verifies this applying $Eval$ with this randomness to C and $\hat{x}$ yields $\hat{c}$, and otherwise it aborts.

Step 6 The prover sends b' and also the randomness r used in producing the commitment. Verifier checks that indeed $\hat{b'} = \mathsf{Com}(b; r)$ and $b' = b$. If so, it accepts the proof. Otherwise it aborts.

1. Prove that the protocol satisfies the completeness property: if Prover and Verifier follow the protocol then the verifier accepts the proof with probability 1.

2. Prove that the protocol satisfies the soundness property: if Verifier follows the protocol and $C(x) = 0$ for every $x \in \{0,1\}^n$, then no matter what strategy the prover uses (even if it cannot be computed efficiently), the verifier will accept the proof with probability at most 0.6.

3. Prove that the protocol satisfies the honest verifier zero knowledge property: there is an algorithm SIM such that for every C, x s.t. $C(x) = 1$, if we let $SIM(C)$ be the random

variable denoting the output of SIM on C, then $SIM(C)$ is computationally indistinguishable from the view of the honest verifier in the protocol interaction with public input C and prover's private input x. The view of a party consists of its random coins, and all the messages it receives. Note that SIM only receives C and not x.

4. Prove that the protocol satisfies ϵ-zero knowledge: for every polynomial-time cheating verifier strategy V^*, there is an algorithm SIM^* such that for every C, x s.t. $C(x) = 1$ and $\epsilon > 0$, the random variable $SIM(C, \epsilon)$ is ϵ-computationally indistinguishable from the view of V^* in the protocol interaction with public input C and prover's private input x. You can assume ϵ is larger than $1/p(n)$ for some polynomial p (if it helps, just assume $\epsilon > 1/n$) and so in particular any negligible function μ satisfies $\mu(n) < \epsilon/2$ (for sufficiently large n). We say that two random variables X, Y parameterized by a security parameter n are *ϵ-computationally indistinguishable* if for every poly(n)-sized circuit T there is some negligible function μ such that
$$\left|\Pr[T(X) = 1] - \Pr[T(Y) = 1]\right| < \epsilon + \mu(n)$$
(As I mentioned in class, you may find it easier to prove this property by considering an "intermediate" simulator SIM' that does get x as input, and prove first that the outputs of SIM and SIM' are indistinguishable, and then that the output of SIM' and the real interaction are ϵ-indistinguishable.)

5. Suppose that we are not guaranteed that the scheme has the verifiable keys and ciphertext property. Change the protocol to obtain a complete, sound and ϵ-zero knowledge protocol in this case as well. You can relax the soundness condition so that the verifier is guaranteed to accept with probability at most $1 - 1/(10n)$ if the statement is false. See footnote for hint.[1]

Exercise 2 (40 points)**.** Let $(G, E, D, NAND)$ be a homomorphic encryption scheme. Prove the claim I sketched in class: there exists a polynomial-time algorithm $EVAL$ such that for every $(e, d) \leftarrow G(1^n)$, $x_1, \ldots, x_m \in \{0,1\}$, if $\hat{x}_i = E_e(x_i)$ and C is a Boolean circuit mapping $\{0,1\}^m$ to $\{0,1\}$, then
$$EVAL_e(C, \hat{x}_1, \ldots, \hat{x}_m) \approx_{|C|\mu(n)} E_e(C(x_1, \ldots, x_n))$$
where we say that $D \approx_\epsilon D'$ if their statistical distance is at most ϵ, μ is some negligible function, and $|C|$ denotes the number of gates of C.

Definitions of homomorphic encryption and commitment scheme:

Definition 1. We say that a tuple of probabilistic polynomial-time algorithm $(G, E, D, NAND)$ is a *fully homomorphic encryption scheme* if (G, E, D) is a CPA-secure public key encryption scheme for one bit messages and for every $(e, d) \in Support(G(1^n))$, $a, b \in \{0,1\}$, and $\hat{a} \in Support(E_e(a))$, $\hat{b} in Support(E_e(b))$,
$$NAND_e(\hat{a}, \hat{b}) \approx E_e(\overline{a \wedge b})$$
where for a distribution X, we denote by $Support(X)$ the set of elements that have positive probability of appearing in X, and $\approx$ denotes statistical indistinguishability (i.e., $n^{-\omega(1)}$ statistical distance).

[1] **Hint:** To test the public key was generated properly, you can add a Step 1.5 after step 1 where the verifier will toss a coin at random, and if it comes up heads then he will ask to see the randomness used in producing the public key, and if it comes up tails they will continue as before. To test a ciphertext the verifier can choose a random $\hat{x}_i$ and request to see the randomness used in producing it. You will need to pair this idea with a randomized encoding mapping x into a (possibly longer) string x' so that every bit of x' is uniform no matter what x was. (And hence it's safe to reveal any individual bit of x'.)

We say that (G, E, D) has the *verifiable keys and ciphertext* property if there is a polynomial-time algorithm $VER - KEY$ that on input $e, 1^n$ outputs 1 iff there exists some randomness for the key generator G that would cause it to produce a pair of the form (e, d) on input 1^n (i.e., if there is some d s.t. $(e, d) \in Support(G(1^n))$), and a polynomial-time $VER - CTEXT$ that on input $e, \hat{a}$ outputs 1 iff there is a bit $a \in \{0, 1\}$ and some randomness r such that $\hat{a}$ is the output of the encryption algorithm E on inputs e, a and randomness r (i.e., $\hat{a} \in Support(E_e(a))$). Needless to say, while these algorithms recognize valid keys and ciphertexts, they are not able to actually recover the randomness used to generate them (as otherwise they would break the cryptosystem).

Definition 2. We say that a function $\mathsf{Com} : \{0, 1\} \times \{0, 1\}^n \to \{0, 1\}^m$ is a *commitment scheme* if it satisfies: (Note that Com takes a one bit input and an n bit input, we call the second input the *randomness* of Com, and often use a semicolon to separate the two inputs.)

Perfect binding There do not exist $r, r' \in \{0, 1\}^n$ such that $\mathsf{Com}(0; r) = \mathsf{Com}(1; r')$.

Computational hiding The distributions $\mathsf{Com}(0; U_n)$ and $\mathsf{Com}(1; U_n)$ are computationally indistinguishable. (Recall that U_n is the uniform distribution over $\{0, 1\}^n$.)

For $a \in \{0, 1\}$, we denote by $\mathsf{Com}(a)$ the random variable $\mathsf{Com}(a; U_n)$.