COS 433 — Cryptography — Homework 10.

Boaz Barak

Total of 120 points. Due **Monday, December 17th, 2007 1:30pm**.
This exercise will have double weight (counts for two exercises).

Exercise 1 (30 points)**.** The following questions rely on the definition of two-party secure function evaluation given in class (using simulation), you can find a definition also in Oded Goldreich's book (or the online version) or in Yehuda Lindell's thesis. Recall that we think of computing a function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n \times \{0,1\}^n$, where if Alice has input x and Bob has input y, at the end of the protocol Alice gets w and Bob gets z, where $(w, z) = f(x, y)$, and we also use the notation $w = f_1(x, y)$, $z = f_2(x, y)$. Neither Alice nor Bob should get anything else.

1. Suppose we have a protocol to evaluate every polynomial-time computable $f()$ whose two outputs are the same: $f_1(x, y) = f_2(x, y)$ for every x, y. Use that to construct a protocol to evaluate every polynomial-time computable function.

2. Suppose we have a protocol to evaluate every poly-time function $f()$, use that to construct a protocol to evaluate every *poly-time probabilistic process*, where a probabilistic process is a function $g : \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n \times \{0,1\}^n$ on *three* inputs, and if Alice has input x and Bob has input y, then Alice gets w and Bob gets z where $(w, z) = f(x, y, r)$ for a uniformly random string r (note that neither Alice nor Bob can control or know r).

Exercise 2 (30 points)**.** The following two questions concerns the heuristic we saw used in some electronic voting protocols to transform an interactive public coin proof system into a non-interactive system (this is known as the *Fiat-Shamir heuristic*).

1. Let L be some language and Π be a 3-round (first prover message, verifier message, last prover message) interactive proof system for L with soundness 2^{-n} on inputs of length n that is *public coins* (i.e., verifier's message is a random string and his decision is obtained by running a polynomial-time algorithm on the transcript).[1] Suppose that in the random oracle model, the verifier's message is computed by applying the oracle on the input and first message, to obtain a *non-interactive* proof system for L (the prover computes a proof by computing his first message, applying the oracle to obtain the verifier's message, and then responding to that message). Prove that this proof system has still negligible ($n^{-\omega(1)}$) soundness error with respect to polynomial-time provers. See footnote for hint.[2] **Extra 15 points:** Prove this remains true even if Π has a constant (possibly larger than 3) number of rounds. That is, we take such a proof system Π and make it non-interactive by changing every verifier message to be obtained by applying the oracle to the concatenation of the input and all previous messages sent by the prover.

[1] We don't care if Π is or is not zero knowledge.

[2] **Hint:** This is true as long as a cheating prover can make at most a polynomial number of queries to the oracle, regardless of its running time.

2. Show that the previous statement is false if we put no condition on the number of rounds in Π. That is, show a language L and a public-coin proof system Π for L with soundness error 2^{-n} (possibly using polynomially many rounds), such that if we replace every message of the verifier by an application of the random oracle on all previous messages, then a polynomial-time cheating prover can come up with an accepting transcript for an input $x \notin L$.

Exercise 3 (45 points)**.** For each of the following statements, say whether it is *true* or *false*, and prove your assertion. You can consider as true all the assumption given in class as mentioned in the instructions on the first page. For example, if the statement is that there exists an object with some properties, then either prove that such an object exists by giving a construction based on the assumptions together with a proof that the construction satisfies the properties, or prove that such an object cannot exist.

1. There exists a pseudorandom generator $G = \{G_n\}$ where for every n, $G_n : \{0,1\}^n \to \{0,1\}^{2n}$ such that for every $x \in \{0,1\}^n$, the first $n/3$ bits of $G_n(x)$ are zero.

2. There exists a pseudorandom generator $G = \{G_n\}$ where for every n, $G_n : \{0,1\}^n \to \{0,1\}^{2n}$ such that for every $x \in \{0,1\}^n$, there exist $n/3$ bits of $G_n(x)$ that are zero.

3. There exists a pseudorandom generator $G = \{G_n\}$ where for every n, $G_n : \{0,1\}^n \to \{0,1\}^{2n}$ such that for every $x \in \{0,1\}^n$, if the first $n/3$ bits of x are zero then all the bits of $G_n(x)$ are zero (i.e., $G_n(x) = 0^{2n}$).

4. There exists a pseudorandom function collection $\{f_s\}_{s\in\{0,1\}^*}$ where, letting $n = |s|$, $f_s : \{0,1\}^n \to \{0,1\}^n$ that satisfies the following: for every $x \in \{0,1\}^n$, $f_{0^n}(x) = 0^n$ (i.e., $f_{0^n}(\cdot)$ is the constant zero function).

5. There exists a pseudorandom function collection $\{f_s\}_{s\in\{0,1\}^*}$ where, letting $n = |s|$, $f_s : \{0,1\}^n \to \{0,1\}^n$ that satisfies the following: for every $s \in \{0,1\}^n$, $f_s(0^n) = 0^n$.

6. For $\ell \geq 2$ and a string $x \in \{0,1\}^\ell$, let *cnot* : $\{0,1\}^\ell \to \{0,1\}^\ell$ be the following function: $\mathit{cnot}(x_1, \ldots, x_\ell) = x_1, x_2 \oplus x_1, x_3, \ldots, x_\ell$. (That is, *cnot* flips the second bit of x according to whether or not the first bit is one.) There exists a CPA-secure public key encryption scheme $(\mathsf{Gen}, \mathsf{E}, \mathsf{D})$ and a polynomial time algorithm A, such that for every n, if $(e,d) = \mathsf{Gen}(1^n)$ then for every $x \in \{0,1\}^\ell$ (where ℓ is the message size of the encryption scheme for security parameter n) it holds that
$$\mathsf{D}_d\Big(A\Big(e, \mathsf{E}_e(x)\Big)\Big) = \mathit{cnot}(x)$$

7. Same statement as Item 6 but with CPA-secure replaced with CCA-secure.

8. There exists a *commitment scheme* $C : \{0,1\} \times \{0,1\}^n \to \{0,1\}^m$ (i.e., a commitment for messages of length one bit, where to commit to the bit b, one chooses $r \leftarrow_{\mathrm{R}} \{0,1\}^n$ and sends $C(b,r)$) and a polynomial time algorithm $A : \{0,1\}^m \to \{0,1\}$, such that for both $b = 0$ and $b = 1$,
$$\Pr_{U_n}[A\Big(C(b, U_n)\Big) \oplus b = 1] \geq 2/3$$
where U_n denotes the uniform distribution over $\{0,1\}^n$.

9. Same statement as Item 8 but with $2/3$ replaced with $1/2$.