CS 120/CSCI E-177: Introduction to Cryptography

Problem Set 5

Assigned: Nov. 2, 2006 — Due: Nov. 7, 2006 (1:10 PM)

Justify all of your answers. See the syllabus for collaboration and lateness policies. You can submit by email to `ciocan@eecs` (please include source files) or by hardcopy Carol Harlow in MD 343.

Problem 1. (Quadratic residues and hardcore bits)

1. Read Section 7.5.1 of Katz–Lindell. Using the material there, prove that the least significant bit is *not* a hardcore bit for the modular exponentiation collection ($f_{p,g}(x) = g^x \bmod p$).
2. Show that the second least significant bit is also not a hardcore bit for the modular exponentiation collection. You may use the fact that a random n-bit prime will be of the form $4k+1$ for integer k with probability $\approx 1/2$. (Hint: try to generalize what Katz-Lindell describe for quadratic residues to quartic residues modulo primes of the form $4k+1$.)

Problem 2. (Length expansion for PRGs) In class, we saw one method for increasing the expansion function of a pseudorandom generator; here we give another. Let $G : \{0,1\}^* \to \{0,1\}^*$ be a pseudorandom generator with expansion function $\ell(n)$, and let $t(n)$ be a function. Consider the function

$$G'(x) = G^{(t(|x|))}(x) = \underbrace{G(G(G(\cdots G(}_{t(|x|)}x)))).$$

1. Show that when $\ell(n) = 2n$ and $t(n) = \log n$, G' is a pseudorandom generator. (Hint: use the hybrid technique.)
2. Does Part 1 also hold for $t(n) = n$? Identify necessary and sufficient conditions on the relationship between $\ell(n)$ and $t(n)$ for G' to be a pseudorandom generator.
3. What are the advantages and disadvantages of this method for length expansion as compared to the one given in class?

Problem 3. (Bit-commitment schemes) A *bit-commitment scheme* is a cryptographic primitive that involves two parties, a *sender* and a *receiver*. The sender *commits* to a value $b \in \{0,1\}$ by sending the receiver a string (called the *commitment*). Later, the sender can "reveal" the value b by sending the receiver another string (called the *opening*), which the receiver checks against the commitment. The commitment should be *binding*, meaning that it should be impossible for the sender to open it as both a 0 and 1. On the other hand, the commitment should be *hiding* in that the committed value should be completely hidden (to a polynomial-time receiver) prior to revelation.

1. Try to formally define the properties we want from a commitment scheme. (If you have trouble, then it may help to try Part 2 first and then formalize the properties of the scheme you construct.)

2. Construct a commitment scheme from any one-way permutation (and hardcore bit).

3. Extra Credit: Construct a commitment scheme from any pseudorandom generator. Actually, your scheme will probably require an extra step, where the receiver selects a random initialization string s which it sends to the sender, and the binding property will only hold with high probability over the receiver's choice of s. (Hint: Use a pseudorandom generator with a large expansion factor, and make use of $G_s(x) = G(x) \oplus s$ in addition to G itself.)