CAS CS 538. Problem Set 7

Due in class Tuesday, October 30, 2012, *before* the start of lecture

Problem 1. (40 points) In this problem you will demonstrate how much easier life is in the random oracle model. Let $\{f_i : D_i \to D_i\}$ be a trapdoor permutation family (which, of course, comes with the following probabilistic polynomial-time algorithms: the algorithm GenT to generate i and trapdoor t, an algorithm to compute $f_i(x)$ given i and $x \in D_i$, and an algorhtm to compute $f_i^{-1}(y)$ given t and y). Let H denote the random oracle. Let $(\text{Gen}, \text{Enc}, \text{Dec})$ be the following encryption scheme:

- Gen picks a trapdoor permutation: runs GenT to generate $\text{PK} = i$ and $\text{SK} = t$
- $\text{Enc}(\text{PK}, m)$ picks a random $r \in D_i$, computes $y = f_i(r)$, $p = H(r)$, $d = p \oplus m$, and outputs $c = (y, d)$.
- $\text{Dec}(\text{SK}, c)$ computes $r = f_i^{-1}(y)$, $p = H(r)$ and outputs $m = d \oplus p$.

Show that this encryption scheme is polynomially secure in the random oracle model, by reduction to the one-wayness of $\{f_i\}$. I will help you with the reduction by providing an intermediate step. If you can't prove the intermediate step in part (a), you can anyway use it in part (b).

(a) (20 points) Let D be the distinguisher. Show that D queries H on $r = f^{-1}(y)$ with probability at least ϵ, where the probability is taken over random choice of f, random choices of D, random choice of whether to encrypt m_0 or m_1, and random choices made during the encryption process.

Advice: Recall (Notes 5, Definition 3) that D produces (m_0, m_1), then m_b (for $b = 0$ or $b = 1$) gets encrypted to get (y, d), and then D has to guess b given an encryption (y, d) of m_b. The way we phrased Definition 3, we considered D's advantage ϵ to be the difference between probability that D outputs 1 when $b = 0$ and probability that D outputs 1 when $b = 1$. For this part, it is easier to use the following, equivalent formulation: b is chosen at random, and the advantage of D is the probability that D outputs b minus $1/2$. You don't need to prove the equivalence of the two formulations.

(b) (20 points) Now complete the reduction.

Here is some algebra review that may be useful in the next problem. If G is a group, then *order of* G is simply another way of saying "the number of elements in G," or "$|G|$." If $g \in G$, the the *order of* g is the smallest integer $r > 0$ such that $g^r = 1$ in G. It is a theorem from algebra that the order of g always divides the order of G. If the order of g is equal to the order of G, then one can show easily that the powers of g cover the entire group G, and hence g is a generator of G. If a group has a generator, it is called *cyclic*.

Note that if G has prime order q, and $g \in G$ has order r, the $r = 1$ or $r = q$ (because $r|q$ and q is prime). If $r = 1$, then $g = g^1 = g^r = 1$. Else, the order of g is the same as the order of G, and so g is a generator of G. Thus, in a group of prime order every element but the element 1 is a generator, so it is always cyclic. In particular, if $p = 2q + 1$, and p, q are primes, then $|QR_p| = (p-1)/2 = q$ is prime, and hence QR_p is cyclic, with every element but 1 as its generator.

Problem 2. (60 points) Let $p_1 = 2q_1 + 1$ and $p_2 = 2q_2 + 1$ be two distinct safe primes, and $n = p_1 p_2$ be of length k bits.

(a) (10 points) Show by CRT that QR_n is cyclic (the easiest way to do this is to try to construct a generator of QR_n and show that it is indeed a generator by showing that its order is $q_1 q_2$).

(b) (10 points) Show how to efficiently factor n given $q_1 q_2$. In fact, you can factor n given any multiple of $q_1 q_2$, but we won't show it here.

(c) (20 points) Let g be a generator of QR_n, and let $M_{n,g} = R_{n,g} = \{1, 2, \ldots, q_1 q_2\}$. Let $H_{n,g}(m, r) = g^{m2^k + r} \bmod n$ (note that $m2^k + r$ is just concatenation of m and r as bit strings). Show that this is a collision-resistant hash family under the assumption that such n are hard to factor. Spefically, show how, given (m_1, r_1) and (m_2, r_2) that collide, one can factor n. You may use without proof the fact that n can be efficiently factored given any non-zero multiple of $q_1 q_2$.

(d) (20 points) Show also that this is a trapdoor hash family (like in problem 3 on the last problem set, find what trapdoor information Gen should output and what the algorithm T will do). Analyze the running time of T to notice how it makes the process described in the paragraph above problem 3 on Problem Set 6 very efficient.