CS6111: Foundations of Cryptography Sept 14, 2018

Homework 2

Instructor: Shweta Agrawal *Due: Sept 21, 2018*

Notation. We let $||$ denote the concatenation operator.

Problem 1: Pseudorandom Generators (6 pts)

a. Let $G : \{0,1\}^k \to \{0,1\}^{2k}$ be a PRG. Prove that it is also a OWF.

b. Let $G : \{0,1\}^k \to \{0,1\}^{k+1}$ be a PRG. Prove that $G'(x_1||x_2) = G(x_1)||G(x_2)$ is also a PRG.

c. Let $G : \{0,1\}^k \to \{0,1\}^{2k}$ be a PRG. Construct an exponential time adversary who can distinguish between $G(x)$ and R with probability greater than $1/2$.

Problem 2: Practice with the Hybrid Argument.(6 pts)

Let $\mathcal{X}_1, \mathcal{X}_2, \mathcal{Y}_1, \mathcal{Y}_2$ be efficiently samplable distributions for which $\mathcal{X}_1 \overset{c}{\approx} \mathcal{Y}_1$ and $\mathcal{X}_2 \overset{c}{\approx} \mathcal{Y}_2$. Here the notation $\overset{c}{\approx}$ denotes computational indistinguishability. Show that $(\mathcal{X}_1, \mathcal{X}_2) \overset{c}{\approx} (\mathcal{Y}_1, \mathcal{Y}_2)$.

Problem 3: PRFs. (4 pts)

Given a pseudorandom function $F_s : \{0,1\}^{k+\lceil \log k \rceil} \mapsto \{0,1\}$, construct a pseudorandom function $F'_s : \{0,1\}^k \mapsto \{0,1\}^k$ and prove that it is secure.

Problem 4: RSA. (6 pts.)

Given an RSA modulus n, an exponent e, and a value $y = x^e \bmod n$, it is hard to determine whether the least significant bit of a message ($LSB(x)$) is 0 or 1.

Now, consider the most significant bit of a message, defined as follows:

$$MSB(x) = \begin{cases} 0 & \text{if } x < n/2 \\ 1 & \text{if } x > n/2 \end{cases}$$

Prove that if it is hard to determine whether the LSB is 0 or 1, it is also hard to determine whether the MSB is 0 or 1. In other words, given an oracle $\mathcal{A}$ that on input $(n, e, y = x^e \bmod n)$ outputs $MSB(x)$ with non-negligible advantage, construct an oracle $\mathcal{B}$ to output $LSB(x)$ with non-negligible advantage.

Hint: Consider what happens to $LSB(2x \bmod n)$, given that we know n is odd.

Problem 5: Naor-Reingold PRF. (12 pts)

The Naor-Reingold PRF $F_{\mathbf{s}}$ $\{0,1\}^k \mapsto G$ takes as input k-bit binary strings and outputs elements of a group G of order q with generator g (for concreteness, let $G = \mathrm{QR}_p$ for a safe prime $p = 2q+1$). Its seed $\mathbf{s}$ is a vector of $k+1$ elements of $\mathbb{Z}_q$, $s_0, \ldots, s_k$. It is computed as follows: $F_{\mathbf{s}}(x) = g^{s_0 \prod_{\ell=1}^{k} s_\ell^{x_\ell}}$ where x_ℓ is the ℓ^{th} bit of x.

Here, we will use the hybrid argument to show that $F_{\mathbf{s}}(x)$ is a PRF under the decisional Diffie-Hellman assumption.

a. Note that, similarly to the GGM PRF, the Naor-Reingold PRF can be thought of as having a tree structure. Each node at depth ℓ is labelled with an ℓ-bit binary string that represents the path from the root to this node. The tree is of depth k, and the value $F_{\mathbf{s}}(x)$ is the value stored at the leaf labelled x. The value g^{s_0} is stored at the root of the tree (the root is labelled with the empty string ε). Given the definition of $F_{\mathbf{s}}(x)$ above, describe the procedure for computing the value stored at node $u \circ b$ (where $u \in \{0,1\}^\ell$ and b is a bit) from the value stored at node u.

b. Let H_i be the following hybrid oracle: at each node at depth i, this oracle defines and stores a random group element. It also picks $k-i$ random elements of $\mathbb{Z}_q$, $(s_{i+1}, \ldots, s_k)$. For each node at depth $\ell > i$, this oracle computes the value stored at this node from the value stored at its parent, as you described in part (a). Show that $H_0 = F_{\mathbf{s}}$, while H_k is a truly random function.

c. Let $H_{i,j}$ be the following hybrid oracle: It picks $k-i$ random elements of $\mathbb{Z}_q$, $(s_{i+1}, \ldots, s_k)$. Then for the first j queries, it behaves consistently with H_{i+1}: on input a query x, it defines and stores a random group element corresponding to the node with label $x_1 x_2 \ldots x_{i+1}$ (or retrieves the one that it has already stored for this node), and then it computes the value stored at leaf x from that of its ancestor at level $i+1$, using $(s_{i+2}, \ldots, s_k)$. Starting with query $j+1$, this oracle behaves consistently with H_i: on input a query x, it defines and stores a random group element corresponding to the node with label $x_1 x_2 \ldots x_i$ (or retrieves the one that it has already stored for this node), and then it computes the value stored at leaf x from that of its closest ancestor whose stored value has already been defined, using $(s_{i+1}, \ldots, s_k)$. Show that, from the point of view of an adversary $\mathcal{A}$ that makes at most $p(k)$ queries to its oracle, $H_{i,0} = H_i$, and $H_{i,p(k)} = H_{i+1}$.

d. Finally, we must show that, for all polynomials p, for all $0 \le i \le k$, $0 \le j \le p(k)$, under the decisional Diffie-Hellman (DDH) assumption, $H_{i,j} \approx H_{i,j+1}$. Recall the DDH assumption from HW7. Consider a reduction that uses the following idea: on input (g, g^a), H_i (and therefore, $H_{i,j}$) can pretend that $s_{i+1} = a$ without knowing a. In order to do so, H_i will pick the values at depth i as follows: instead of just sampling some h from G, it samples $r \leftarrow \mathbb{Z}_q$, and then sets $h = g^r$. The only difference between $H_{i,j}$ and $H_{i,j+1}$ is what happens in the j^{th} query.

   On input tuple (g, g^a, g^b, g^c), the reduction can use this idea to pretend that $s_{i+1} = a$. Let x be the contents of the j^{th} query. The reduction stores g^b at node labelled $x_1 \ldots x_i 0$ and g^c at node $x_1 \ldots x_i 1$, and responds to this and subsequent queries accordingly.

   Flesh out and analyze this reduction.

Problem 6: RSA SecurID card (16 pts)

The RSA SecurID card is a small device that displays 6 digits that change every minute. The idea is that when you log into your account remotely (say when you want to log into your UNIX account in IIT from an Internet Cafe) then you have to type the numbers that appear in the card in addition to your PIN or password.

Figure 1: RSA SecureID card

1. What is the security advantage of such a card over traditional password? That is, what sort of attack can this card resist which cannot be resisted using a standard password mechanism? (When making this comparison, assume that it's possible for users to remember a 6-digits PIN or a password with similar security.)

2. Describe how you would implement such a scheme using pseudorandom functions. Assume that the PRF family takes a seed of size n to map n bits to n bits, and that the number of possible devices is M (for $M < 2^n$). How many bits of storage does your implementation use at the server and each of the devices? (there is an implementation that uses at most $O(n)$ bits in each place).

3. Define what it means that such a scheme is secure. That is, write down a list of desired security properties that any such identification scheme should satisfy. Then, make a formal definition of security based on a game in which the scheme is secure if the probability that the adversary "wins" is small. Any scheme that satisfies the formal definition should have the desired security properties.

4. Prove that your construction above satisfies the definition you made. Say how the security depends on n - the number of bits that the device stores in memory (where its running time is polynomial in n) and on k - the number of digits that it displays to the user.