Homework 5 for 6.875

Teacher: Silvio Micali, TA: Yevgeniy Dodis

Due Date: **Monday, November 16**

Problem 1: from PRG to PRF

Recall the "tree" construction of a pseudo-random function from a pseudo-random generator. Let $G : \{0,1\}^k \mapsto \{0,1\}^{2k}$ be a length-doubling PRG. Denote by $G_0(s)$ the first half of $G(s)$, by $G_1(s)$ – the second half. In other words, $G(s) = G_1(s) \circ G_2(s)$, where $|G_1(s)| = |G_2(s)| = k$ and $\circ$ is the concatenation operator. For any string $x = (x_1, \ldots x_k)$ we let $G_x(s) = G_{x_k}(\ldots G_{x_2}(G_{x_1}(s))\ldots)$ and $f_s(x) = G_x(s)$. We showed that the family $\mathcal{F}_k = \{f_s : \{0,1\}^k \mapsto \{0,1\}^k \mid s \in \{0,1\}^k\}$ is a pseudo-random function family. In other words, a function is chosen by randomly filling the root of the tree and, given x, we follow the path specified by x to find $f_s(x)$.

What about the family $\mathcal{G}_k = \{G_s : \{0,1\}^k \mapsto \{0,1\}^k \mid s \in \{0,1\}^k\}$? In other words, a function is chosen by choosing a random path down the tree and, given x, this path is followed with x placed at the root. Either prove that the construction works or provide a counter-example.

Problem 2: PRF Candidates

Suppose that $\{F_S : \{0,1\}^k \mapsto \{0,1\}^k \mid s \in \{0,1\}^k\}$ is a pseudo-random family of functions from k-bit input to k-bit output, indexed by a k-bit key ("seed"). We would like to get a new pseudo-random function family in which each function maps k bits to $2k$ bits. Consider the following construction, and for each show whether it is good or bad (namely whether the specified family is pseudo-random or not). Below, "$\circ$" denotes concatenation, and $\bar{x}$ denotes the bitwise negation of x.

1. $F^1_S(x) = F_S(0^k) \circ F_S(x)$.
2. $F^2_S(x) = F_S(x) \circ F_S(\bar{x})$.
3. $F^3_S(x) = F_{0^k}(x) \circ F_S(x)$.
4. $F^4_S(x) = F_{S_1}(x) \circ F_{S_2}(x)$, where $S_1 = F_S(0^k)$ and $S_2 = F_S(1^k)$.
5. $F^5_S(x) = F_x(S) \circ F_S(x)$.

Problem 3: "Coin Flipping in the Well"

Write a *formal* definition of a protocol that cryptographically implements coin flipping in the well.

Problem 4: Good or Bad?

For each of the following protocols and the corresponding hardness assumption, EITHER *prove* that the protocol implements coin flipping in the well (according to your formal definition and based on the specified assumption), OR show that it does not, suggest a simple fix and *prove* that the fix works.

1. Assumption: Factoring Blum integers is hard. Protocol:
   Stage 1:

   - Bob: Choose random k-bit primes p_1, p_2 s.t. $p_1 \equiv p_2 \equiv 3 \bmod 4$. Let $n = p_1 p_2$. Send n to Alice.
   - Alice: Choose random $x \in \mathbb{Z}_n^*$. Let $y = x^2 \bmod n$. Send y to Bob.
   - Bob: Pick random $b \in \{0, 1\}$. Send b to Alice.
   - Alice: Compute $c = b \oplus \left(\frac{x}{n}\right)$ (where $\left(\frac{x}{n}\right)$ is the Jacobi symbol of x modulo n).

   Stage 2:

   - Alice: Reveal x to Bob.
   - Bob: Check that $y = x^2 \bmod n$. Compute $c = b \oplus \left(\frac{x}{n}\right)$.

2. Assumption: Quadratic Residuosity is hard. Protocol:
   Stage 1:

   - Alice: Choose random k-bit primes p_1, p_2, set $n = p_1 p_2$. Choose random $x \in \mathbb{Z}_n^*$ with Jacobi symbol 1, $\left(\frac{x}{n}\right) = 1$. Send n and x to Bob.
   - Bob: Pick random $b \in \{0, 1\}$. Send b to Alice.
   - Alice: Compute $c = b \oplus \chi(x)$ (where $\chi(x)$ is the quadratic character of x modulo n).

   Stage 2:

   - Alice: Reveal p_1, p_2 to Bob.
   - Bob: Check that $n = p_1 p_2$. Let $\chi(x) = 1$ iff $\left(\frac{x}{p_1}\right) = \left(\frac{x}{p_2}\right) = 1$. Compute $c = b \oplus \chi(x)$.

3. Assumption: Discrete Log Problem is hard. Protocol:
   Stage 1:

   - Bob: Choose random k-bit prime p, a generator g of $\mathbb{Z}_p^*$ and a random $R \in \mathbb{Z}_p^*$ s.t. $R \neq 1$. Send p, g, R to Alice.
   - Alice: Check that p is prime. Choose random $a \in \{0, 1\}$ and random $z \in \mathbb{Z}_p^*$. If $a = 0$, let $y = g^z \bmod p$, else $y = g^z R \bmod p$. Send y to Bob.
   - Bob: Pick random $b \in \{0, 1\}$. Send b to Alice.
   - Alice: Compute $c = a \oplus b$.

   Stage 2:

   - Alice: Reveal z to Bob.
   - Bob: Check that either $y = g^z \bmod p$ (then set $a = 0$) or $y = g^z R \bmod p$ (then set $a = 1$). Compute $c = a \oplus b$.