Problem 4-1 (PRF Candidates) 22+6 Points

Suppose that \( \{ F_s : \{0,1\}^k \rightarrow \{0,1\}^k \mid s \in \{0,1\}^k \} \) 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 construct a new PRF family. Consider the following constructions, and for each of them show whether it is good or bad (namely whether the specified family is pseudo-random or not). Make the proof formal for one positive example (and convincing for the others). For negative examples, also try to differentiate whether the resulting function family is never pseudorandom (i.e., give a generic attack), or could be non-pseudorandom (i.e., given any PRF family \( F'_S \), construct specific PRF family \( F_S \) which is "bad" for the construction at hand).

Below, "\( \circ \)" denotes concatenation, and \( \overline{x} \) denotes the bitwise negation of \( x \).

(a) (3 points) \( F^1_S(x) = F_S(0^k) \circ F_S(x) \).

Solution: ******************* INSERT PROBLEM 1a SOLUTION HERE *******************

(b) (3 points) \( F^2_S(x) = F_S(x) \circ F_S(\overline{x}) \).

Solution: ******************* INSERT PROBLEM 1b SOLUTION HERE *******************

(c) (3 points) \( F^3_S(x) = F_{0^k}(x) \circ F_S(x) \).

Solution: ******************* INSERT PROBLEM 1c SOLUTION HERE *******************

(d) (5 points) \( F^4_S(x) = F_{S_1}(x) \circ F_{S_2}(x) \), where \( S_1 = F_S(0^k) \) and \( S_2 = F_S(1^k) \).

Solution: ******************* INSERT PROBLEM 1d SOLUTION HERE *******************

(e) (4 points) \( F^5_S(x) = F_x(S) \).
Solution: ****************** INSERT PROBLEM 1e SOLUTION HERE ******************

(f) (4 points) $F^6_S(x) = F_S(x) \oplus x$.

Solution: ****************** INSERT PROBLEM 1f SOLUTION HERE ******************

(g**) Extra Credit. (6 points) $F^7_S(x) = F_S(x) \oplus S$.

Solution: ****************** INSERT PROBLEM 1g SOLUTION HERE ******************

Problem 4-2 (Chosen Ciphertext Attacks) 12+8 Points

Recall the following stateless secret-key encryption scheme, which we called XOR. The secret key is the seed $s$ to a PRF $f_s : \{0,1\}^k \rightarrow \{0,1\}^k$. To encrypt $m \in \{0,1\}^k$, we had $E_s(m) = (r, f_s(r) \oplus m)$, where $r$ is chosen at random. To decrypt $(r,c)$, we had $D_s(r,c) = f_s(r) \oplus c$. We showed that this scheme is IND-secure against CPA-attack. For your reference, here is the formal definition of IND-security (of a general encryption scheme not just XOR) against CPA-attack.

Definition 1 SKE is IND-secure against CPA iff $\forall$ PPT $B = (B_1, B_2)$,

$$\Pr[b = \tilde{b} \mid s \leftarrow \text{Gen}(1^k); \quad (m_0, m_1, \beta) \leftarrow B_1^{E_s}(1^k); \quad b \leftarrow \{0,1\}; \quad \tilde{c} \leftarrow E_s(m_b); \quad \tilde{b} \leftarrow B_2^{E_s}(\tilde{c}, \beta);] \leq \frac{1}{2} + \text{negl}(k)$$

In this problem we consider the security of this scheme against stronger chosen ciphertext attacks. In these attacks the adversary also has access to the decryption oracle $D_s(\cdot)$ in addition to $E_s(\cdot)$. Two kinds of chosen ciphertext attack are possible. In the first one, called lunch-time attack, the adversary can call $D_s(\cdot)$ only in the “find” stage (when he tries to output $m_0$ and $m_1$, i.e., when running $B_1$ but not $B_2$). In the second, called CCA attack, he can call it in the “guess” stage as well (i.e. for both $B_1$ and $B_2$). Of course, to make sense of the CCA attack we must prohibit to submit the challenge ciphertext $\tilde{c}$ to the decryption oracle in the “guess” stage (but any other ciphertext $c \neq \tilde{c}$ is OK).

(a) (8 points) Give a proof that the XOR scheme is secure against the lunch-time attack when $f_s$ is a PRF, by turning the following (informal) argument into a formal proof. Intuitively, as long as the value $r$ selected to encrypt the challenge message $m_b$ is “fresh”, then the security of the PRF should make the scheme secure. Let $q_1$ be the total number of encryption queries, and $q_2$ be the number of decryption queries $B_1$ makes in the “find” stage. Tell in English what “fresh” above means, and then write an upper bound on the probability that the value $r$ is not fresh as a function of $q_1$, $q_2$ and $k$ (assuming here that $f_s$ is a truly random function). Conclude that this probability is negligible and use it to show that the XOR scheme is lunch-time secure.

PS4-2
Solution: ***************** INSERT PROBLEM 2a SOLUTION HERE ******************

(b) (4 points) Next, argue that XOR is completely insecure against the IND-CCA attacker. Namely, given a challenge ciphertext $\tilde{c} = (\tilde{r}, \tilde{z})$, make a single decryption query $(r, z) \neq (\tilde{r}, \tilde{z})$ (i.e., either $r \neq \tilde{r}$ or $z \neq \tilde{z}$), the answer to which allows you to completely decrypt the challenge ciphertext $\tilde{c}$.

Solution: ***************** INSERT PROBLEM 2b SOLUTION HERE ******************

(c∗) Extra Credit. (8 points) Let us modify the XOR scheme as follows. We assume the domain of $f$ is twice as large ($2k$ rather than $k$ bits), $f$ is a strong PRP rather than a PRF, we use concatenation $\circ$ in place of $\oplus$, and also drop the no longer required value $r$ from the ciphertext. Formally, $E_s(m) = f_s(m \circ r)$, where $m \in \{0,1\}^k$ is the message, $r \in \{0,1\}^k$ is chosen at random and $f_s : \{0,1\}^{2k} \rightarrow \{0,1\}^{2k}$ is a strong PRP. To decrypt $c$, compute $m \circ r = f_s^{-1}(c)$, and simply drop the value $r$. Argue that the resulting scheme is CCA-secure.

(Hint: Argue that the decryption oracle is “useless”, since it always returns random answers.)

Solution: ***************** INSERT PROBLEM 2c SOLUTION HERE ******************

Problem 4-3 (Necessity of Luby-Rackoff) 15 Points

Recall the Luby-Rackoff construction of PRP’s from PRF’s. Namely, let $F$ be a PRF family of functions from $k$ to $k$ bits. Given a function $f \in F$, define the Feistel function $H_f(L, R) = (R, L \oplus f(R))$. Finally, define $H^i_F(L, R) = H_{f_1}(\ldots H_{f_i}(L, R)\ldots)$, be the Feistel function evaluated $i$ times with $f_1 \ldots f_i$, each of which is drawn from $F$ at random and independently from others. We mentioned that functions $H^3_F : \{0,1\}^{2k} \rightarrow \{0,1\}^{2k}$ form a PRP family, while $H^4_F : \{0,1\}^{2k} \rightarrow \{0,1\}^{2k}$ form a strong PRP family. You are to show that these results cannot be improved.

(a) (3 points) Show that $H^1_F$ do not form a PRP family. In fact, one query to $H^1_F$ allows to distinguish it from a random permutation.

(Hint: What happens to the right part?)

Solution: ***************** INSERT PROBLEM 3a SOLUTION HERE ******************

(b) (5 points) Show that $H^2_F$ do not form a PRP family as well. In fact, two queries to $H^2_F$ allow to distinguish it from a random permutation.

(Hint: What happens if you “reuse” the right part?)

Solution: ***************** INSERT PROBLEM 3b SOLUTION HERE ******************
Problem 4-4 (Security against Related Key Attacks) 8+8 Points

In this problem you are going to consider a stronger security notion for pseudo random functions, called security against related key attacks. Intuitively, this notion says that a PRF should remain secure even if the adversary can see the evaluation of the PRF under a key “perturbed” by the adversary in some (almost) arbitrary way. However, for the sake of this exercise, we simply define this notion only for the case of the Naor-Reingold’s PRF that we have studied in class.

Let $R : (\mathbb{Z}_q^*)^{\ell+1} \times \{0, 1\}^\ell \rightarrow \mathbb{Z}_q$ be a truly random function, i.e., imagine that on every input $(\delta_0, \ldots, \delta_\ell, x)$, $R$ samples a fresh value $r \in \mathbb{Z}_q$ uniformly at random and outputs $r$ (but, when queried on the same input twice, $R$ returns the same output).

Define the Naor-Reingold (NR) PRF as follows:

$$NR_{p,g,a_0,a_1,\ldots,a_\ell}(x) = g^{a_0 \prod_{i=1}^{\ell} x_i + a_i}$$

where $(a_0, \ldots, a_\ell) \in \mathbb{Z}_q^{\ell+1}$ is the seed, and $x \in \{0, 1\}^\ell$ is the input.

For any $a = (a_0, \ldots, a_\ell)$, we define the function $F_a : (\mathbb{Z}_q^*)^{\ell+1} \times \{0, 1\}^\ell \rightarrow \mathbb{Z}_q$ as follows: on input $\delta = (\delta_0, \ldots, \delta_\ell) \in (\mathbb{Z}_q^*)^{\ell+1}$ and $x \in \{0, 1\}^\ell$:

$$F_a(\delta, x) = NR_{p,g,a_0,\delta_0,a_1,\ldots,a_\ell,\delta_\ell}(x) = g^{a_0 \delta_0 \prod_{i=1}^{\ell} x_i + a_\ell \delta_\ell}$$

i.e., $F$ evaluates $NR$ on input $x$ using a key which is obtained by multiplying $a$ by $\delta$ component-wise.

Then, the NR PRF is said secure against related key attacks if for any PPT adversary $A$:

$$\text{Adv}_{A}^{\text{rka}}(k) = |\Pr[A^{F_a(\cdot, \cdot)}(1^k) = 1 | a \leftarrow \mathbb{Z}_q^{\ell+1}] - \Pr[A^{R(\cdot, \cdot)}(1^k) = 1]| \leq \text{negl}(k)$$

where the probabilities are taken over the random choices of $a = (a_0, \ldots, a_\ell)$ and $R$ respectively.

(a) (3 points) In this part you are to show that given access to the standard PRF oracle $NR_{p,g,a_0,a_1,\ldots,a_\ell}(\cdot)$, it is possible to efficiently simulate the oracle $F_a(\cdot, \cdot)$ without knowing the secret key $a$. More precisely, show that for any $x \in \{0, 1\}^\ell$ and any $\delta = (\delta_0, \ldots, \delta_\ell) \in (\mathbb{Z}_q^*)^{\ell+1}$, given the output $y = NR_{p,g,a_0,a_1,\ldots,a_\ell}(x)$, it is possible to efficiently compute $F_a(\delta, x)$ (without knowing $a$).

Solution: ****************** INSERT PROBLEM 4a SOLUTION HERE ******************

(b) (5 points) Show that the NR PRF is not secure according to our new definition, i.e., show an efficient adversary $A$ that uses its oracle (either $F_a(\cdot, \cdot)$ or $R(\cdot, \cdot)$) to obtain advantage close to 1 in the related key attack experiment described above.

(Hint: Use part (a) and the information given in part (c) below as a hint for your attack.)

(c)** (7 points) Show that $H^3_\ell$ do not form a strong PRP family. Naturally, you should use queries to both $H^3_\ell$ and $(H^3_\ell)^{-1}$ (I believe two queries to the former and one the to latter suffice).

Solution: ****************** INSERT PROBLEM 3c SOLUTION HERE ******************
(c) **(Extra credit)** (8 points) Show that the NR PRF is secure according to our new definition if we restrict the adversary to query $A$ the oracle only on distinct inputs $x$. (**Hint**: Use part (a) to reduce the “RKA security” of NR PRF to its regular security. Where do you use the fact that all the inputs are distinct?)

Solution: ****************** INSERT PROBLEM 4c SOLUTION HERE ******************