6.875: Cryptography and Cryptanalysis **September 8, 2021**

Problem Set 1 Solutions

Instructor: Vinod Vaikuntanathan **TAs:** Lali Devadas and Sacha Servan-Schreiber

Instructions.

- **When:** This problem set is due on **September 24, 2021** before **5pm ET**.
- **How:** You should use LaTeX to type up your solutions (you can use our LaTeX template from the course webpage). Solutions should be uploaded on Gradescope as a single pdf file.
- **Acknowledge your collaborators:** Collaboration is permitted and encouraged in small groups of at most three. You must write up your solutions *entirely on your own* and *acknowledge your collaborators*.
- **Reference your sources:** If you use material from outside the lectures, you must reference your sources (papers, websites, wikipedia, ...).
- **When in doubt, ask questions:** Use Piazza or the TA office hours for questions about the problem set. See the course webpage for the timings.

Problem 1. Perfect vs. Statistical Secrecy

Shannon's Perfect Secrecy. Recall the definition of an encryption scheme satisfying *prefect correctness* and *perfect secrecy*. An encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$ and ciphertext space $\mathcal{C}$ is said to be:

- **perfectly correct** if for all messages $m \in \mathcal{M}$,

$$\Pr\left[\ k \leftarrow \mathsf{Gen}(1^\lambda) : \mathsf{Dec}(k, \mathsf{Enc}(k, m)) = m\ \right] = 1$$

  That is, the encryption scheme always correctly decrypts an encrypted message under every secret key k.

- **perfectly secret** if for all messages $m_0, m_1 \in \mathcal{M}$, and for all $c \in \mathcal{C}$

$$\Pr\left[\ k \leftarrow \mathsf{Gen}(1^\lambda) : \mathsf{Enc}(k, m_0) = c\ \right] = \Pr\left[\ k \leftarrow \mathsf{Gen}(1^\lambda) : \mathsf{Enc}(k, m_1) = c\ \right],$$

  where the probability is over the randomness of Gen and Enc.

Note that Gen and Enc could be randomized, and Dec is deterministic. It was shown in lecture that, in order to achieve these requirements, the size of the key space has to be at least as large as the message space. In this problem, we will consider variations and relaxations of this definition. Before diving in, we will define the notions of *statistical distance* and *statistical secrecy*.

Statistical Distance. Let X and Y be two random variables over a finite set S with distributions D_X and D_Y, respectively. The **statistical distance** (also called the total variation distance) between X and Y, denoted $\Delta(X, Y)$ is defined as

$$\Delta(X, Y) := \frac{1}{2} \sum_{z \in S} \Big| \Pr[X = z] - \Pr[Y = z] \Big|.$$

Note that $0 \leq \Delta(X, Y) \leq 1$. $\Delta(X, Y) = 1$ precisely when the supports of the distributions D_X and D_Y are disjoint, and $\Delta(X, Y) = 0$ precisely when the two distributions are identical.

Statistical Secrecy. Let λ be a security parameter. An encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$ and ciphertext space $\mathcal{C}$. For all messages $m \in \mathcal{M}$, let $\mathcal{C}_m$ be the distribution of ciphertexts sampled according to the following process:

$$k \leftarrow \mathsf{Gen}(1^\lambda) \text{ and } c \leftarrow \mathsf{Enc}(k, m_b).$$

$(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is said to be **statistically secret** if for all messages $m_0, m_1 \in \mathcal{M}$, there exists a negligible function $\epsilon = \epsilon(\lambda)$ such that

$$\Delta(\mathcal{C}_{m_0}, \mathcal{C}_{m_1}) \leq \epsilon.$$

(a) Suppose Alice and Bob want to communicate securely without an evil three-letter agency E.V.E. decrypting their communication. However, Alice and Bob live in a world where the MOD operator has been banned by E.V.E., who does not want people using Shannon's one-time pad. That is, using $\text{XOR}(b_1, b_2) = b_1 + b_2 \pmod 2$ is no longer possible, even though computing $\text{ADD}(b_1, b_2) = b_1 + b_2$ *is* possible.

Alice proposes using the following encryption scheme, which she hopes will be secure against even the most powerful computers that E.V.E. has at their disposal.

Alice's One-time Pad without MOD:
Let $\mathbb{Z}^+$ be the set of ~~non-negative~~ positive integers (i.e., ~~$\{0, 1, 2, \dots\}$~~ $\{1, 2, \dots\}$) and let λ be a security parameter. Let $\mathcal{M} = \mathcal{K} = \{i \in \mathbb{Z}^+ \mid i \leq 2^\lambda\}$, and define $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ as follows.

- $k \leftarrow \mathsf{Gen}(1^\lambda)$: sample a uniformly random k from $\mathcal{K}$.
- $c \leftarrow \mathsf{Enc}(k, m)$: Output $k + m$.
- $m \leftarrow \mathsf{Dec}(k, c)$: Output $c - k$.

Prove that Alice's scheme satisfies perfect correctness but does not satisfy perfect secrecy. Compute the statistical distance ϵ between encryptions of (any given) m_0 and m_1 under Alice's scheme. Does Alice's scheme satisfy *statistical secrecy*? If not, modify her scheme to make it statistically secret. (Hint: you do not need to modify Enc and Dec.)

Solution

(Proving perfect correctness) Suppose, for contradiction, that the scheme does not satisfy perfect correctness. Then, we have that there exists an $m \in \mathcal{M}$ such that

$$\Pr\left[\ k \leftarrow \mathsf{Gen}(1^\lambda) : \mathsf{Dec}(k, \mathsf{Enc}(k, m)) = m\ \right] < 1.$$

This implies that there exist two messages $m_0 \neq m_1$ such that

$$\mathsf{Dec}(k, \mathsf{Enc}(k, m_0)) = m_1.$$

Substituting the instantiation of Enc and Dec we have that

$$\underbrace{\underbrace{(k + m_0)}_{\mathsf{Enc}} - k}_{\mathsf{Dec}} = m_1,$$

and so we have that $m_0 = m_1$, which contradicts the starting assumption.

(Disproving statistical and perfect secrecy) To prove that the scheme does not satisfy even statistical secrecy, let alone perfect secrecy, we compute the statistical distance between two encryptions of messages $m_0, m_1 \in \mathcal{M}$ under a key k.

$$\begin{aligned}
\Delta(\mathcal{C}_{m_0}, \mathcal{C}_{m_1}) &= \frac{1}{2} \sum_{c \in \mathbb{Z}^+, c \leq 2^\lambda + 1} \left|\Pr[\mathsf{Enc}(k, m_0) = c] - \Pr[\mathsf{Enc}(k, m_1) = c]\right| \\
&= \frac{1}{2} \sum_{i=2}^{2^\lambda+1} \left|\Pr[k_0 \leftarrow \mathsf{Gen}(1^\lambda) : k_0 + m_0 = i] - \Pr[k_1 \leftarrow \mathsf{Gen}(1^\lambda) : k_1 + m_1 = i]\right| \\
&= \frac{1}{2} \left(\frac{2|m_0 - m_1|}{2^\lambda}\right) \\
&= \frac{|m_0 - m_1|}{2^\lambda}.
\end{aligned}$$

Plugging in $m_0 = 1$ and $m_1 = 2^\lambda$ we get that $\Delta(\mathcal{C}_1, \mathcal{C}_{2^\lambda}) = \frac{(2^\lambda - 1)}{2^\lambda} \approx 1$ and hence does not satisfy the statistical secrecy definition. Because statistical secrecy is strictly weaker compared to perfect secrecy, we get that the scheme doesn't satisfy perfect secrecy either.

(Achieving statistical secrecy) It suffices[a] to set $\mathcal{M} = \{0, 1\}$.
We then only have to consider $\Delta(\mathcal{C}_{m_0}, \mathcal{C}_{m_1})$ for the case where $m_0 = 0$ and $m_1 = 1$. $\mathcal{C}_0$ ranges over 0 to 2^λ, while $\mathcal{C}_1$ ranges over 1 to $2^\lambda + 1$. Note that our only problematic ciphertexts are 1, which could never be an encryption of 1, and $2^\lambda + 1$, which could never be an encryption of 0.

$$\begin{aligned}
\Delta(\mathcal{C}_0, \mathcal{C}_1) &= \frac{1}{2} \sum_{c \in \mathbb{Z}^+, c \leq 2^\lambda + 1} \left|\Pr[k \leftarrow \mathsf{Gen}(1^\lambda) : \mathsf{Enc}(k, 0) = c] - \Pr[k \leftarrow \mathsf{Gen}(1^\lambda) : \mathsf{Enc}(k, 1) = c]\right| \\
&= \frac{1}{2} \left(\left|\frac{1}{2^\lambda} - 0\right| + \sum_{1 < c \leq 2^\lambda} \left|\frac{1}{2^\lambda} - \frac{1}{2^\lambda}\right| + \left|0 - \frac{1}{2^\lambda}\right|\right) \\
&= \frac{1}{2^\lambda} \leq \mathsf{negl}(\lambda)
\end{aligned}$$

[a] An alternative solution is to increase the key space.

(b) Alice and Bob find a classified document claiming that E.V.E. can only read *partially-corrupted* ciphertexts sent over the internet (even though Alice and Bob receive the *uncorrupted* ciphertext). Both Alice and Bob grow tired of having to trek across the MIT campus to generate a new random key for each message they encrypt, and they wonder if they can use E.V.E.'s lack of error-correction to their advantage.

Formally, E.V.E.'s lack of error-correction can be modeled by her receiving ciphertexts $\tilde{c}$ where the i^{th} bit of $\tilde{c}$ is the i^{th} bit of c with probability $1-p$ and is the opposite bit with probability p. Observe that if $p = 0$, E.V.E. receives the uncorrupted ciphertext, i.e. $\tilde{c} = c$.

~~**Let** $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ **be Alice's OTP-without-MOD from part (a) (without your modifications). What is the smallest value of** p **that would make** $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ ***statistically secret*****? How about** ***perfectly secret*****?**~~

Alice comes up with the following encryption scheme $(\mathsf{Gen}^*, \mathsf{Enc}^*, \mathsf{Dec}^*)$ **that doesn't make use of the key! Specifically,** $\mathsf{Enc}(k, m) = m$ **and** $\mathsf{Dec}(k, m) = m$**. What value of** p **makes** $(\mathsf{Gen}^*, \mathsf{Enc}^*, \mathsf{Dec}^*)$ ***statistically secret*** **against E.V.E who only observes corrupted ciphertexts** $\tilde{c}$**? How about** ***perfectly secret*****? You may assume that** $\mathcal{C} = \mathcal{M} = \{0,1\}^\lambda$**.**

Solution

First we will show that for $p < \frac{1}{2} - \frac{1}{\mathsf{poly}(\lambda)}$ (or symmetrically $p > \frac{1}{2} + \frac{1}{\mathsf{poly}(\lambda)}$), we can never have statistical secrecy. Set $m_0 = 0^\lambda$ (the all-zero bit string) and $m_1 = 1^\lambda$ (the all-one bit string). Suppose $\lambda = 1$, then

$$\Pr[m = m_0 : \tilde{c} = 0] = \Pr[\text{not flipped}] = 1 - p = \frac{1}{2} + \frac{1}{\mathsf{poly}(\lambda)}$$

and

$$\Pr[m = m_1 : \tilde{c} = 0] = \Pr[\text{flipped}] = p = \frac{1}{2} - \frac{1}{\mathsf{poly}(\lambda)},$$

where the probabilities are over the noise of the ciphertext (remember that the key isn't used, so the randomness of its generation is irrelevant). The distinguisher, if it sees 0, outputs 0, and if it sees 1, outputs 1. The advantage is $\frac{1}{\mathsf{poly}(\lambda)}$, as can be easily seen.

Solution (continued...)

Now, we will show that we do get statistical secrecy for $p = \frac{1}{2} - \mathsf{negl}(\lambda)$. Fix arbitrary messages m_0, m_1. For every bit string $\tilde{c} \in \{0,1\}^\lambda$, we can compute the probability that E.V.E. sees $\tilde{c}$ given that the message being encrypted was m_0 as follows:

$$\Pr[\tilde{c} \mid m_0] = \left(\frac{1}{2} - \mathsf{negl}(\lambda)\right)^h \left(\frac{1}{2} + \mathsf{negl}(\lambda)\right)^{\lambda - h}$$

where h is the number of locations that m_0 and s differ (also known as the Hamming distance between m_0 and s). We will now lower-bound and upper-bound the probability as follows. Consider one extreme where $h = 0$ ($\tilde{c} = m_0$). Then, we have that

$$\Pr[\tilde{c} \mid m_0] = \left(\frac{1}{2} + \mathsf{negl}(\lambda)\right)^\lambda = \frac{1}{2^\lambda} \cdot (1 + 2 \cdot \mathsf{negl}(\lambda))^\lambda \leq \frac{1}{2^\lambda} \cdot (1 + \mathsf{negl}'(\lambda)).$$

At the other extreme, we have $h = \lambda$ ($\tilde{c} = \overline{m_0}$) and we get:

$$\Pr[\tilde{c} \mid m_0] = \left(\frac{1}{2} - \mathsf{negl}(\lambda)\right)^\lambda = \frac{1}{2^\lambda} \cdot (1 - 2 \cdot \mathsf{negl}(\lambda))^\lambda \geq \frac{1}{2^\lambda} \cdot (1 - \mathsf{negl}''(\lambda)).$$

Therefore, we get that

$$\frac{1}{2^\lambda} \cdot (1 - \mathsf{negl}''(\lambda)) \;\leq\; \Pr[\tilde{c} \mid m_0] \;\leq\; \frac{1}{2^\lambda} \cdot (1 + \mathsf{negl}'(\lambda)).$$

The same argument holds symmetrically for m_1. Then

$$\begin{aligned}\Delta(\mathcal{C}_{m_0}, \mathcal{C}_{m_1}) &= \frac{1}{2} \sum_{\tilde{c} \in \{0,1\}^\lambda} |\Pr[\tilde{c} \mid m_0] - \Pr[\tilde{c} \mid m_1]| \\ &\leq \frac{1}{2} \sum_{\tilde{c} \in \{0,1\}^\lambda} \left(\frac{1}{2^\lambda} \cdot \mathsf{negl}'''(\lambda)\right) \leq \frac{1}{2} \cdot \mathsf{negl}'''(\lambda),\end{aligned}$$

so we have statistical secrecy as desired.

Problem 2. Pseudorandom Generators (PRG)

Notation. In this problem, $||$ denotes concatenation of strings, and $\bar{x}$ denotes the bit-wise complement of x, e.g., if $x = 01011$ then $\bar{x} = 10100$. A substring of x is denoted $x_{[a:b]}$, where indices a and b are inclusive. We will write $x_{[i]}$ to denote just the i^{th} bit of x.

Let $G_1 : \{0,1\}^\lambda \to \{0,1\}^{\lambda+1}$ and $G_2 : \{0,1\}^\lambda \to \{0,1\}^{\lambda+1}$ be PRGs stretching a λ-bit input into a $(\lambda+1)$-bit output.

1. $G'(x) = G_1(x) \mid\mid G_2(x)$

Solution

Not always a PRG. Counter-example: Set $G_1(x) = G_2(x)$, then all outputs are of the form $y||y$, which is distinguishable in polynomial time. Moreover, the probability that the distinguisher succeeds is $1 - \frac{1}{2^\lambda}$ (the only time the distinguisher fails is when the a truly random string is of the form $r||r$, which happens with probability $\frac{1}{2^\lambda}$). Thus the distinguishing success probability is non-negligible which shows that G' is not always PRG.

2. $G'(x) = G_2(G_1(x)_{[1:\lambda]}) \mid\mid G_1(x)_{[\lambda+1]}$

Solution

Is a PRG. Suppose, for contradiction that G' is not a PRG. Then there exists a PPT distinguisher $\mathcal{A}$ distinguishing between outputs of G' and unformly random strings with non-negligible probability. We construct a PPT distinguisher $\mathcal{D}$ for G where G is either G_1 or G_2 (in both cases arising in a contradiction). We do so via a hybrid argument.

Consider the following hybrid distributions.
$\mathcal{H}_0$: $G_2(G_1(x)_{[1:\lambda]}) \mid\mid G_1(x)_{[\lambda+1]}$.
$\mathcal{H}_1$: $G_2(R) \mid\mid b$ where R is a random λ-bit string and b is a random bit.
$\mathcal{H}_2$: $R \mid\mid b$ where R is a random $\lambda + 1$-bit string and b is a random bit.

Claim: $\mathcal{H}_0 \approx_c \mathcal{H}_1$**. Proof:** Suppose, towards contradiction, that $\mathcal{H}_0$ and $\mathcal{H}_1$ are not computationally indistinguishable. Then there exists a distinguisher $\mathcal{A}$ distinguishing between hybrids $\mathcal{H}_0$ and $\mathcal{H}_1$ with non-negligible probability $\delta(\lambda)$. We construct a distinguisher $\mathcal{D}$ for G_1 for as follows.

1: On input $y \in \{0,1\}^{\lambda+1}$ where y is either the output of G_1 or a uniformly random string, construct $y' = G_2(y_{[1:\lambda]})||y_{[\lambda+1]}$.

2: Run $\mathcal{A}(y')$ and output as $\mathcal{A}$ does.

Analysis: If $\mathcal{A}$ exists, then we can construct a PPT distinguisher $\mathcal{D}$ for G_1 contradicting the assumption that G_1 is a PRGs. The distribution given to $\mathcal{A}$ matches the distribution expected by $\mathcal{A}$: if y is random, then b is random and $\mathcal{A}$ gets as input an instance of $\mathcal{H}_1$. Otherwise, if y is the output of G_1 then $\mathcal{A}$ gets as input an instance of $\mathcal{H}_0$. As such, $\mathcal{D}$ succeeds with the same probability as $\mathcal{A}$, that is, with non-negligible probability $\delta(\lambda)$, which contradicts the assumption that G_1 is a PRG. □

Claim: $\mathcal{H}_1 \approx_c \mathcal{H}_2$**. Proof:** Suppose, towards contradiction, that $\mathcal{H}_1$ and $\mathcal{H}_2$ are not computationally indistinguishable. Then there exists a distinguisher $\mathcal{A}$ distinguishing between hybrids $\mathcal{H}_1$ and $\mathcal{H}_2$ with non-negligible probability $\delta(\lambda)$. We construct a distinguisher $\mathcal{D}$ for G_2 for as follows.

1: On input $y \in \{0,1\}^{\lambda+1}$ where y is either the output of G_2 or a uniformly random string, construct $y' = y||b$ where b is sampled at random.

2: Run $\mathcal{A}(y')$ and output as $\mathcal{A}$ does.

Analysis: If $\mathcal{A}$ exists, then we can construct a PPT distinguisher $\mathcal{D}$ for G_2 contradicting the assumption that G_2 is a PRGs. The distribution given to $\mathcal{A}$ matches the distribution expected by $\mathcal{A}$: if y is random, $\mathcal{A}$ gets as input an instance of $\mathcal{H}_2$. Otherwise, if y is the output of G_2 then $\mathcal{A}$ gets as input an instance of $\mathcal{H}_1$. As such, $\mathcal{D}$ succeeds with the same probability as $\mathcal{A}$, that is, with non-negligible probability $\delta(\lambda)$, which contradicts the assumption that G_2 is a PRG. □

By the above two claims, we get that $\mathcal{H}_0 \approx_c \mathcal{H}_1 \approx_c \mathcal{H}_2$. As a result, if G_1 and G_2 are PRGs then so is G'.

3. $G'(x) = G_1(x) \oplus G_2(\bar{x})$

Solution

Not always a PRG. Counter-example: Set $G_1(x) = G_2(\bar{x})$, then all outputs are of the form $0^{\lambda+1}$, which is distinguishable in polynomial time. The distinguisher succeeds with probability $1 - \frac{1}{2^{\lambda+1}}$ given that the probability that a uniformly random string is of the form $0^{\lambda+1}$ is $\frac{1}{2^{\lambda+1}}$. Thus the distinguishing success probability is non-negligible which shows that G' is not always PRG.

4. $G'(x||y) = G_1(x||0^{\lambda/2}) \;||\; y$ where $|x| = |y| = \lambda/2$.

Solution

Not always a PRG. Counter-example: Set $G_1(x_1||x_2) = x_1||G''(x_2)$ where $G'' : \{0,1\}^{\lambda/2} \to \{0,1\}^{\lambda/2+1}$ is a PRG. G' constructed with the G_1 above will not be a PRG. The proof involves two steps. First we must show that G_1 as constructed is indeed a PRG. Second we must show that G' instantiated with G_1 is not a PRG (proving the that the counter-example is correct).

Claim: G_1 (as constructed above) is PRG. Suppose, towards contradiction, that G_1 is not a PRG. Then there exists a PPT distinguisher $\mathcal{A}$ distinguishing between outputs of G_1 and unformly random strings with non-negligible probability $\delta(\lambda)$. We construct a PPT distinguisher $\mathcal{D}$ for G'' as follows.

1: On input $y \in \{0,1\}^{\lambda/2+1}$ where y is either the output of G'' or a uniformly random string, construct $y' = r||y$ where r is a uniformly random $\lambda/2$ bit string.

2: Run $\mathcal{A}(y')$ and output as $\mathcal{A}$ does.

Analysis: If $\mathcal{A}$ exists, then we can construct a distinguisher $\mathcal{D}$ for G'', which contradicts our assumption that G'' is a PRG. The reduction is correct because:

- $\mathcal{D}$ runs in polynomial time given that sampling r running $\mathcal{A}$ takes polynomial time relative to the input.
- The distribution of y' is exactly the same distribution expected by $\mathcal{A}$ as input.
- $\mathcal{A}$ outputs 0/1 depending on its guess and succeeds in distinguishing between G' and random with non-negligible probability. This success probability translates directly to the success probability of $\mathcal{D}$ by construction.

As a result, we get that G_1 (as constructed for this proof) is a PRG if G'' is a PRG.

Claim: G' is not PRG. Using the G_1 we constructed, we turn to examining the output of

$$\begin{aligned} G'(x||y) &= G_1(x||0^{\lambda/2})||y \\ &= x||G''(0^{\lambda/2})||y \end{aligned}$$

This output of G' is therefore trivially distinguishable in polynomial time given that the distinguisher need only evaluate $G''(0^{\lambda/2})$ and see if it is a substring of the output. Because G'' takes polynomial time to evaluate, the distinguisher is efficient and succeeds with probability $1 - \frac{1}{2^{\lambda/2+1}}$ (where $\frac{1}{2^{\lambda/2+1}}$ is the probability that the middle $\lambda/2+1$ bits of a random string are the output of $G''(0^{\lambda/2})$). Because this success probability is non-negligible, G' is not a PRG.

For each of the above constructions of G', prove that G' is a PRG OR provide a counterexample and proof showing why it is not a PRG.

Problem 3. Pseudorandom Permutations (PRP)

After the first few lectures of 6.875, Bob feels confident that he can build pseudorandom functions. He puts together a pseudorandom function family $\mathcal{F} = \{f_k : \{0,1\}^\ell \to \{0,1\}^\ell\}_{k \in \{0,1\}^q}$ for $q = q(\lambda), \ell = \ell(\lambda)$.

He brags about this to Alice, who complains that even given the key k, Bob may not be able to *invert* his pseudorandom function; that is, given k and $f_k(x)$, it is unclear how to recover x. She claims that she can one-up him by transforming his pseudorandom function into one that is also invertible (i.e. a permutation) given the key. She defines a **pseudorandom permutation family** as a family of functions $\mathcal{P} = \{p_s : \{0,1\}^n \to \{0,1\}^n\}_{s \in \{0,1\}^m}$ for $m = m(\lambda), n = n(\lambda)$ such that

- **correctness:** for all $s \in \{0,1\}^m$, $p_s : \{0,1\}^n \to \{0,1\}^n$ is an efficiently computable bijection.
- **security:** for all probabilistic polynomial-time (PPT) distinguishers $\mathcal{A}$, there exists some negligible function $\epsilon = \epsilon(\lambda)$ such that

$$\left| \Pr\left[s \xleftarrow{R} \{0,1\}^m : \mathcal{A}^{p_s}(1^\lambda) = 1 \right] - \Pr\left[p \xleftarrow{R} \mathbb{P}_n : \mathcal{A}^{p}(1^\lambda) = 1 \right] \right| \leq \epsilon$$

  where $\mathbb{P}_n$ is the set of all permutations (i.e., bijective functions) on $\{0,1\}^n$.

(a) Alice's first idea is to set $n = 3\ell$ and split the input $x \in \{0,1\}^n$ into three equal parts, which we will call $L_0, M_0, R_0 \in \{0,1\}^\ell$. That is,

$$L_0 := x_{[1:\ell]},\ M_0 := x_{[\ell+1:2\ell]},\ R_0 := x_{[2\ell+1:n]}.$$

She defines $p_s(\cdot)$ to be:

$p_s(x)$

1 : **parse** $x = L_0 || M_0 || R_0$ `// as above`
2 : **parse** $s = k$
3 : $L_1 := M_0$
4 : $M_1 := R_0 \oplus f_k(M_0)$
5 : $R_1 := L_0 \oplus f_k(R_0)$
6 : **return** $L_1 || M_1 || R_1$

Prove that this is a permutation.

Solution

We will prove that this function is a permutation by showing the inverse of the function exists and is unique.

Given output $y = L_1||M_1||R_1$, observe that:

1. $L_1 = M_0$,
2. $M_1 \oplus f_k(M_0) = (R_0 \oplus f_k(M_0)) \oplus f_k(M_0) = R_0$,
3. $R_1 \oplus f_k(R_0) = (L_0 \oplus f_k(R_0)) \oplus f_k(R_0) = L_0$.

We can then reconstruct the *unique* input as

$$x = L_0||M_0||R_0.$$

It is easy to check that:

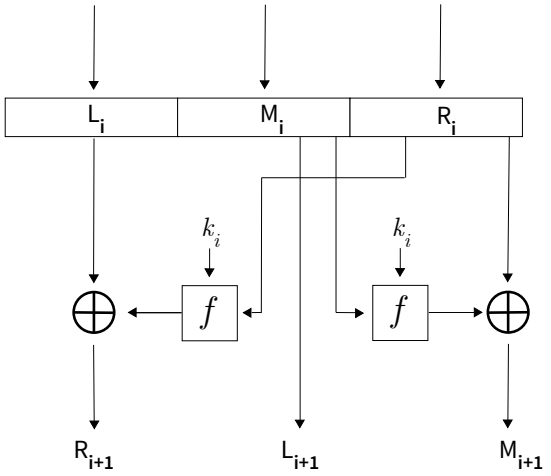
$$p_s(x) = y,$$

as required.

(b) Since the above is obviously not pseudorandom (convince yourself!), Alice's next idea is to repeat the procedure iteratively multiple times. She picks a number of "rounds" r and sets $m = rq$ so that each round can have a different key $k_i \in \{0,1\}^q$ for the pseudorandom function (i.e. $s = k_0||\cdots||k_{r-1} \in \{0,1\}^m$). This time she defines $p_s(\cdot)$ to be:

$p_s(x)$

1 : **parse** $x = L_0||M_0||R_0$
2 : **parse** $s = k_0||\cdots||k_{r-1}$
3 : $L_1 := M_0$
4 : $M_1 := R_0 \oplus f_{k_0}(M_0)$
5 : $R_1 := L_0 \oplus f_{k_0}(R_0)$
6 : **foreach** $i = 1, \ldots, r-1$ **do**
7 : $\quad L_{i+1} := M_i$
8 : $\quad M_{i+1} := R_i \oplus f_{k_i}(M_i)$
9 : $\quad R_{i+1} := L_i \oplus f_{k_i}(R_i)$
10 : **endfor**
11 : **return** $L_r||M_r||R_r$



Prove that p_s is still a permutation for any $r \geq 1$.

Solution

We prove this by induction.

Base case: The base case is exactly the argument of 3(a).

Inductive step: For any $r > 1$, given $y_r = L_r||M_r||R_r$, we can compute $L_{r-1}||M_{r-1}||R_{r-1}$, as follows:

1. $L_r = M_{r-1}$,
2. $M_r \oplus f_k(M_{r-1}) = (R_{r-1} \oplus f_k(M_{r-1})) \oplus f_k(M_0) = R_{r-1}$,
3. $R_r \oplus f_k(R_{r-1}) = (L_{r-1} \oplus f_k(R_{r-1})) \oplus f_k(R_{r-1}) = L_{r-1}$.

By the inductive hypothesis, we can compute $L_{r-2}||M_{r-2}||R_{r-2}$, and so on so forth until we reconstruct the unique input $x = L_0||M_0||R_0$, which maps to y_0.

The following lemma is also sufficient.

Lemma 1 *The composition $P : X \to X$ of permutations $P_i : X \to X$ from $i = 1$ to $i = r$ (i.e. $P(x) = P_r(\cdots(P_1(x)))$) is also a permutation.*

Proof: The P_i are permutations, so let $P_i^{-1} : X \to X$ be an inverse for each P_i. Define P^{-1} as the composition of all the P_i^{-1} from $i = r$ to $i = 1$. Then we have $\forall\, x \in X$,

$$P^{-1}(P(x)) = P_1^{-1}(\cdots(P_r^{-1}(P_r(\cdots(P_1(x)))))) = P_1^{-1}(P_1(x)) = x$$

$$P(P^{-1}(x)) = P_r(\cdots(P_1(P_1^{-1}(\cdots(P_r^{-1}(x)))))) = P_r(P_r^{-1}(x)) = x$$

and thus P is a permutation. □

(c) Alice gets excited and claims that the above is a pseudorandom permutation for $r = 3$, but Bob thinks it still isn't pseudorandom.

Who is correct? Prove your answer. (If you take Bob's side, you need to show an attack on the permutation family; if you take Alice's side, you need to show a proof of security that it satisfies the definition above.)

Solution

We will prove that Bob is correct by showing an attack which distinguishes this from a truly random permutation (regardless of the choice of pseudorandom function f). First let's compute $p_s(L_0||M_0||R_0)$ for $s = k_0||k_1||k_2$:

$$
\begin{aligned}
&L_0||M_0||R_0 \\
&\to M_0 \;||\; R_0 \oplus f_{k_0}(M_0) \;||\; L_0 \oplus f_{k_0}(R_0) \\
&\to R_0 \oplus f_{k_0}(M_0) \;||\; L_0 \oplus f_{k_0}(R_0) \oplus f_{k_1}\big(R_0 \oplus f_{k_0}(M_0)\big) \;||\; M_0 \oplus f_{k_1}\big(L_0 \oplus f_{k_0}(R_0)\big) \\
&\to L_0 \oplus f_{k_0}(R_0) \oplus f_{k_1}\big(R_0 \oplus f_{k_0}(M_0)\big) \\
&\qquad ||\; M_0 \oplus f_{k_1}\big(L_0 \oplus f_{k_0}(R_0)\big) \oplus f_{k_2}\Big(L_0 \oplus f_{k_0}(R_0) \oplus f_{k_1}\big(R_0 \oplus f_{k_0}(M_0)\big)\Big) \\
&\qquad ||\; R_0 \oplus f_{k_0}(M_0) \oplus f_{k_2}\Big(M_0 \oplus f_{k_1}\big(L_0 \oplus f_{k_0}(R_0)\big)\Big)
\end{aligned}
$$

The distinguisher $\mathcal{A}^{\mathcal{O}}$ does the following:

$\mathcal{A}^{\mathcal{O}}$

1 : $L||M||R \leftarrow \mathcal{O}\left(0^\ell||0^\ell||0^\ell\right)$

2 : $L'||M'||R' \leftarrow \mathcal{O}\left(1^\ell||0^\ell||0^\ell\right)$

3 : **if** $L \oplus L' \stackrel{?}{=} 1^\ell$: **return** 1

4 : **else** : **return** 0

If $\mathcal{O}$ is a truly random permutation, then L and L' are both random and independent, so $L \oplus L' = 1^\ell$ with probability $\frac{1}{2^\ell}$. By the computation above, if $\mathcal{O} = p_s$, then $L = f_{k_0}(0^\ell) \oplus f_{k_1}\big(f_{k_0}(0^\ell)\big)$ and $L' = 1^\ell \oplus f_{k_0}(0^\ell) \oplus f_{k_1}\big(f_{k_0}(0^\ell)\big)$, so $L \oplus L' = 1^\ell$ with probability 1. Thus we have that

$$\left|\Pr\left[\; s \stackrel{R}{\leftarrow} \{0,1\}^m : \mathcal{A}^{p_s}(1^\lambda) = 1 \;\right] - \Pr\left[\; p \stackrel{R}{\leftarrow} \mathbb{P}_n : \mathcal{A}^{p}(1^\lambda) = 1 \;\right]\right| = 1 - \frac{1}{2^\ell} \geq \mathsf{nonnegl}(\lambda)$$

Problem 4. Pseudorandom Functions (PRF)

Alice and Bob have a different problem. They want to use a NIST-approved pseudorandom function family defined as

$$\mathcal{F} = \{f_k : \{0,1\}^\ell \to \{0,1\}^\ell\}_{k \in \{0,1\}^q}$$

where $q = q(\lambda)$, just as before. But they discover that what they need is slightly different, namely, a pseudorandom function that operates on inputs of length 3ℓ.

(a) Bob has a bright idea: he suggests to XOR the function outputs! That is, he defines a function family $\mathcal{G} = \{g_k : \{0,1\}^{3\ell} \to \{0,1\}^\ell\}_{k \in \{0,1\}^{3q}}$ by the functions

$$g_{k_0||k_1||k_2}(x_0||x_1||x_2) = f_{k_0}(x_0) \oplus f_{k_1}(x_1) \oplus f_{k_2}(x_2)\ .$$

Is $\mathcal{G}$ a pseudorandom function family? Provide an attack or a proof of that it satisfies the definition of a PRF.

Solution

(Throughout the solution for problem 4 we will use $\mathbb{F}_{a,b}$ to denote the set of all functions $\{0,1\}^a \to \{0,1\}^b$.) We construct a PPT distinguisher $\mathcal{A}^{\mathcal{O}}$ as follows:

$\mathcal{A}^{\mathcal{O}}$

1 : $a \leftarrow \mathcal{O}(0^\ell||0^\ell||0^\ell)$
2 : $b \leftarrow \mathcal{O}(1^\ell||0^\ell||0^\ell)$
3 : $c \leftarrow \mathcal{O}(0^\ell||1^\ell||1^\ell)$
4 : $d \leftarrow \mathcal{O}(1^\ell||1^\ell||1^\ell)$
5 : **if** $a \oplus b \oplus c \oplus d \stackrel{?}{=} 0^\ell$: **return** 1
6 : **else** : **return** 0

Analysis: First, note that $\mathcal{A}$ requires only four queries to the oracle, and thus runs in polynomial time. Second, note that when $\mathcal{O} = g_{k_0||k_1||k_2}$, then

$$\begin{aligned} a \oplus b \oplus c \oplus d &= \Big(f_{k_0}(0^\ell) \oplus f_{k_1}(0^\ell) \oplus f_{k_2}(0^\ell)\Big) \oplus \Big(f_{k_0}(1^\ell) \oplus f_{k_1}(0^\ell) \oplus f_{k_2}(0^\ell)\Big) \\ &\quad + \Big(f_{k_0}(0^\ell) \oplus f_{k_1}(1^\ell) \oplus f_{k_2}(1^\ell)\Big) \oplus \Big(f_{k_0}(1^\ell) \oplus f_{k_1}(1^\ell) \oplus f_{k_2}(1^\ell)\Big) \\ &= f_{k_0}(0^\ell) \oplus f_{k_0}(1^\ell) \oplus f_{k_0}(0^\ell) \oplus f_{k_0}(1^\ell) = 0^\ell \end{aligned}$$

with probability 1. On the other hand, if $\mathcal{O}$ is a truly random function R , then $a \oplus b \oplus c \oplus d$ is also truly random and is 0^ℓ with probability $\frac{1}{2^\ell}$. Thus we have that

$$\begin{aligned} &\Big|\Pr\Big[\ k \xleftarrow{R} \{0,1\}^{3q} : \mathcal{A}^{g_k}(1^\lambda) = 1\ \Big] - \Pr\Big[\ R \xleftarrow{R} \mathbb{F}_{3\ell,\ell} : \mathcal{A}^{R}(1^\lambda) = 1\ \Big]\Big| \\ &\qquad = 1 - \frac{1}{2^\ell} \geq \mathsf{nonnegl}(\lambda). \end{aligned}$$

(b) It is now Alice's turn. She proceeds to define the function family $\mathcal{H} = \{h_k : \{0,1\}^{3\ell} \to \{0,1\}^\ell\}_{k \in \{0,1\}^\ell}$ by the following iterative process:

$$\begin{aligned} y_0 &= f_k(x_0); \\ y_1 &= f_{y_0}(x_1); \\ y_2 &= f_{y_1}(x_2). \end{aligned}$$

Finally, she sets $h_k(x_0||x_1||x_2) = y_2$.

Is $\mathcal{H}$ a pseudorandom function family? Provide an attack or a proof of that it satisfies the definition of a PRF.

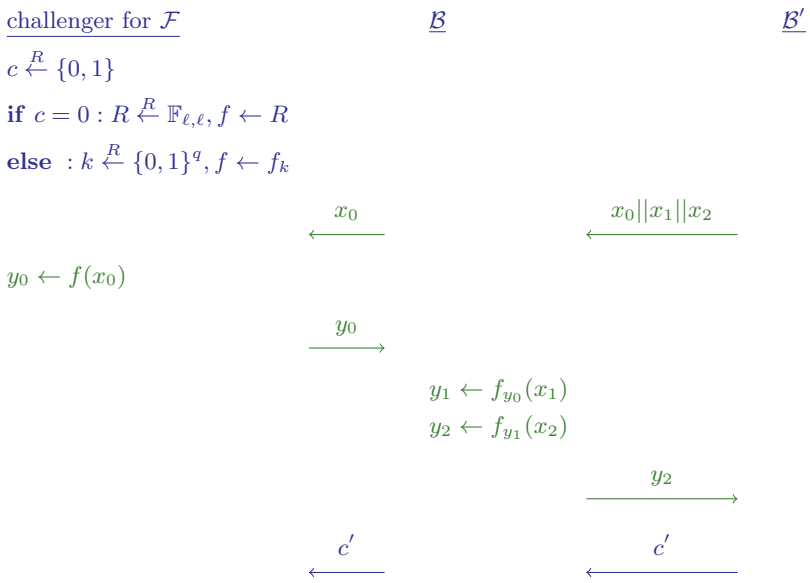
Solution

$\mathcal{H}$ is a pseudorandom function family. We will show this via a hybrid argument. Suppose there is some PPT adversary $\mathcal{A}^{\mathcal{O}}$ which can distinguish $\mathcal{H}$ from random by making at most $Q = \mathsf{poly}(\lambda)$ queries. Consider the following sequence of hybrids:

- $\mathcal{H}_0$ is exactly Alice's function family. It draws a key $k \xleftarrow{R} \{0,1\}^\ell$. On input $x = x_0||x_1||x_2$, it computes $y_0 = f_k(x_0)$, $y_1 = f_{y_0}(x_1)$, $y_2 = f_{y_1}(x_2)$ and sets $h_k(x) = y_2$.

- $\mathcal{H}_1$ replaces f_k in the first iteration with a truly random function. It draws $R \xleftarrow{R} \mathbb{F}_{\ell,\ell}$ and sets $k = R$. On input $x = x_0||x_1||x_2$ it computes $y_0 = R(x_0)$, $y_1 = f_{y_0}(x_1)$, $y_2 = f_{y_1}(x_2)$ and sets $h_k(x) = y_2$. We show that $\mathcal{H}_0 \approx_c \mathcal{H}_1$ via a reduction.

  Suppose towards contradiction that there exists some PPT adversary $\mathcal{B}'^{\mathcal{O}}$ which can distinguish $\mathcal{H}_0$ and $\mathcal{H}_1$. We construct a PPT adversary $\mathcal{B}^{\mathcal{O}}$ which uses $\mathcal{B}'$ to break the pseudorandomness of $\mathcal{F}$ (the section in green can be repeated polynomially many times, to answer all of $\mathcal{B}'$'s queries):



  Analysis: Note that when the challenger for $\mathcal{F}$ is acting like f_k, then $\mathcal{B}'$ is interacting with $\mathcal{H}_0$, and when the challenger for $\mathcal{F}$ is acting like a truly random function, then $\mathcal{B}'$ is interacting with $\mathcal{H}_1$. Since $\mathcal{B}'$ can distinguish between $\mathcal{H}_0$ and $\mathcal{H}_1$ with non-negligible probability in the security parameter λ, $\mathcal{B}$ is a distinguisher for $\mathcal{F}$.

- $\mathcal{H}_{2,0}$ partitions the input space based on x_0. It draws Q keys $\{k_1, \ldots, k_Q \xleftarrow{R} \{0,1\}^\ell\}$ which will be used for the different values of x_0 the adversary queries (recall that the adversary makes at most Q queries, so its queries can have at most Q different values of x_0) and sets $k = k_1||\cdots||k_Q$. On input $x = x_0||x_1||x_2$, if x_0 is unassigned (i.e. hasn't shown up in previous queries), it assigns x_0 to the next available key, then it computes $y_1 = f_{k_{x_0}}(x_1)$, $y_2 = f_{y_1}(x_2)$, and sets $h_k(x) = y_2$. This is only a semantic change from $\mathcal{H}_1$, with the sampled key k_{x_0} corresponding to the value $R(x_0)$.

Solution (continued...)

- $\mathcal{H}_{2,1}$ through $\mathcal{H}_{2,Q}$ follow a similar style as hybrid arguments we've seen in class to replace each keyed pseudorandom function with a truly random function. For $1 \leq i \leq Q$, $\mathcal{H}_{2,i}$ draws $Q-i$ keys $\{k_1, \ldots, k_{Q-i} \xleftarrow{R} \{0,1\}^\ell\}$, which will be used for the first $Q-i$ values of x_0 which the adversary queries, and i random functions $\{R^1, \ldots, R^i \xleftarrow{R} \mathbb{F}_{\ell,\ell}\}$, which will be used for the remaining values of x_0 the adversary queries (recall that the adversary makes at most Q queries, so its queries can have at most Q different values of x_0), and sets $k = k_1||\cdots||k_{Q-i}||R^1||\cdots||R^i$. On input $x = x_0||x_1||x_2$,
  - if x_0 is unassigned and there are keys left, x_0 is assigned to the next available key k; if x_0 is unassigned and there are no keys left, x_0 is assigned to the next available random function R; then
  - if x_0 is assigned to a key k_{x_0}, it computes $y_1 = f_{k_{x_0}}(x_1)$, $y_2 = f_{y_1}(x_2)$ and sets $h_k(x) = y_2$;
  - if x_0 is assigned to a random function R^{x_0}, it computes $y_1 = R^{x_0}(x_1)$, $y_2 = f_{y_1}(x_2)$ and sets $h_k(x) = y_2$.

  We show that $\mathcal{H}_{2,i-1} \approx_c \mathcal{H}_{2,i}$ for $1 \leq i \leq Q$ via a reduction.

  Suppose towards contradiction that there exists some PPT adversary $\mathcal{B}'^{\mathcal{O}}$ which can distinguish $\mathcal{H}_{2,i-1}$ and $\mathcal{H}_{2,i}$. We construct a PPT adversary $\mathcal{B}^{\mathcal{O}}$ which uses $\mathcal{B}'$ to break the pseudorandomness of $\mathcal{F}$:

  $\mathcal{B}^{\mathcal{O}}$ samples keys and random functions like $\mathcal{H}_{2,i}$, except it only samples $i-1$ random functions $\{R^2, \ldots, R^i \xleftarrow{R} \mathbb{F}_{\ell,\ell}\}$. It runs $\mathcal{B}'^{\mathcal{O}}$, and on query $x_0||x_1||x_2$, responds like $\mathcal{H}_{2,i}$, unless $x_0 = x_0^*$, the prefix which would have been assigned to R^1. On query $x_0^*||x_1||x_2$, it sends x_1 as a query to its challenger for $\mathcal{F}$, and uses its response in place of $R^1(x_1)$ in its calculation.

  > *Analysis:* Note that when the challenger for $\mathcal{F}$ is acting like f_k, then $\mathcal{B}'$ is interacting with $\mathcal{H}_{2,i-1}$, and when the challenger for $\mathcal{F}$ is acting like a truly random function, then $\mathcal{B}'$ is interacting with $\mathcal{H}_{2,i}$. Since $\mathcal{B}'$ can distinguish between $\mathcal{H}_{2,i-1}$ and $\mathcal{H}_{2,i}$ with non-negligible probability in the security parameter λ, $\mathcal{B}$ is a distinguisher for $\mathcal{F}$.

- $\mathcal{H}_{3,0}$ partitions the input space based on $x_0||x_1$. It draws Q keys $\{k_1, \ldots, k_Q \xleftarrow{R} \{0,1\}^\ell\}$ and sets $k = k_1||\cdots||k_Q$. On input $x = x_0||x_1||x_2$, if $x_0||x_1$ is unassigned (i.e. hasn't shown up in previous queries), it assigns $x_0||x_1$ to the next available key, then it computes $y_2 = f_{k_{x_0||x_1}}(x_2)$, and sets $h_k(x) = y_2$. This is only a semantic change from $\mathcal{H}_{2,Q}$, with the sampled key $k_{x_0||x_1}$ corresponding to the value $R^{x_0}(x_1)$.

Solution (continued...)

- $\mathcal{H}_{3,1}$ through $\mathcal{H}_{3,Q}$ follow a similar style as hybrid arguments we've seen in class to again replace each keyed pseudorandom function with a truly random function. For $1 \leq i \leq Q$, $\mathcal{H}_{3,i}$ draws $Q - i$ keys $\{k_1, \ldots, k_{Q-i} \xleftarrow{R} \{0,1\}^\ell\}$ and i random functions $\{R^1, \ldots, R^i \xleftarrow{R} \mathbb{F}_{\ell,\ell}\}$, and sets $k = k_1 || \cdots k_{Q-i} || R^1 || \cdots || R^i$. On input $x = x_0 || x_1 || x_2$,
  - if $x_0||x_1$ is unassigned and there are keys left, $x_0||x_1$ is assigned to the next available key k; if $x_0||x_1$ is unassigned and there are no keys left, $x_0||x_1$ is assigned to the next available random function R; then
  - if $x_0||x_1$ is assigned to a key $k_{x_0||x_1}$, it computes $y_2 = f_{k_{x_0||x_1}}(x_2)$ and sets $h_k(x) = y_2$,
  - if x_0 is assigned to a random function $R^{x_0||x_1}$, it computes $y_2 = R^{x_0||x_1}(x_2)$ and sets $h_k(x) = y_2$,

  We show that $\mathcal{H}_{3,i-1} \approx_c \mathcal{H}_{3,i}$ for $1 \leq i \leq Q$ via a reduction.

  Suppose towards contradiction that there exists some PPT adversary $\mathcal{B}'^{\mathcal{O}}$ which can distinguish $\mathcal{H}_{3,i-1}$ and $\mathcal{H}_{3,i}$. We construct a PPT adversary $\mathcal{B}^{\mathcal{O}}$ which uses $\mathcal{B}'$ to break the pseudorandomness of $\mathcal{F}$:

  $\mathcal{B}^{\mathcal{O}}$ samples keys and random functions like $\mathcal{H}_{3,i}$, except it only samples $i-1$ random functions $\{R^2, \ldots, R^i \xleftarrow{R} \mathbb{F}_{\ell,\ell}\}$. It runs $\mathcal{B}'^{\mathcal{O}}$, and on query $x_0||x_1||x_2$, responds like $\mathcal{H}_{2,i}$, unless $x_0||x_1 = x_0^*||x_1^*$, the prefix which would have been assigned to R^1. On query $x_0^*||x_1^*||x_2$, it sends x_2 as a query to its challenger for $\mathcal{F}$, and uses its response in place of $R^1(x_2)$ in its calculation.

  > *Analysis:* Note that when the challenger for $\mathcal{F}$ is acting like f_k, then $\mathcal{B}'$ is interacting with $\mathcal{H}_{3,i-1}$, and when the challenger for $\mathcal{F}$ is acting like a truly random function, then $\mathcal{B}'$ is interacting with $\mathcal{H}_{3,i}$. Since $\mathcal{B}'$ can distinguish between $\mathcal{H}_{3,i-1}$ and $\mathcal{H}_{3,i}$ with non-negligible probability in the security parameter λ, $\mathcal{B}$ is a distinguisher for $\mathcal{F}$.

- $\mathcal{H}_4$ is a truly random function $R \xleftarrow{R} \mathbb{F}_{3\ell,\ell}$. This is only a semantic change from $\mathcal{H}_{3,Q}$, which computes $R^{x_0||x_1}(x_2)$.

(c) Bob thinks Alice's construction is *too iterative*. He optimizes her construction and defines the function family $\mathcal{H}' = \{h'_k : \{0,1\}^{3\ell} \to \{0,1\}^\ell\}_{k \in \{0,1\}^\ell}$ as follows:

$$y_0 = f_k(x_0); \;\; y_1 = f_k(x_1);$$
$$y_2 = f_{y_0 \oplus y_1}(x_2).$$

He sets $h'_k(x_0||x_1||x_2) = y_2$.

Is $\mathcal{H}'$ a pseudorandom function family? Provide an attack or a proof of that it satisfies the definition of a PRF.

Solution

We construct a PPT distinguisher $\mathcal{A}^{\mathcal{O}}$ as follows:

$\mathcal{A}^{\mathcal{O}}$

1 : $y \leftarrow \mathcal{O}\left(0^\ell || 0^\ell || 0^\ell\right)$

2 : **if** $y \stackrel{?}{=} f_{0^\ell}(0^\ell)$: **return** 1

3 : **return** 0

Analysis: First, note that $\mathcal{A}$ requires only one query to the oracle, and thus runs in polynomial time. Second, note that when $\mathcal{O} = h'_k$, then

$$y = y_2 = f_{y_0 \oplus y_1}(x_2) = f_{f_k(x_0) \oplus f_k(x_1)}(x_2) = f_{f_k(0^\ell) \oplus f_k(0^\ell)}(0^\ell) = f_{0^\ell}(0^\ell)$$

with probability 1. On the other hand, if $\mathcal{O}$ is a truly random function R, then y is also truly random and is $f_{0^\ell}(0^\ell)$ with probability $\frac{1}{2^\ell}$. Thus we have that

$$\left|\Pr\left[\ k \xleftarrow{R} \{0,1\}^\ell : \mathcal{A}^{h'_k}(1^\lambda) = 1\ \right] - \Pr\left[\ R \xleftarrow{R} \mathbb{F}_{3\ell,\ell} : \mathcal{A}^{R}(1^\lambda) = 1\ \right]\right|$$
$$= 1 - \frac{1}{2^\ell} \geq \mathsf{nonnegl}(\lambda).$$