6.5620 (6.875), Fall 2022

Homework # 1 **Due: 21 September 2022, 11:59:59pm ET**

- **Typsetting:** You are encouraged to use LaTeX to typeset your solutions. You can use the following template.
- **Submissions:** Solutions should be submitted to Gradescope.
- **Reference your sources:** If you use material outside the class, please reference your sources (including papers, websites, wikipedia).
- **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.

Problems:

1. *(5 points)* **Working with negligible functions.** Recall that a non-negative function $\nu : \mathbb{N} \to \mathbb{R}^+$ is *negligible* if it decreases faster than the inverse of any polynomial (otherwise, we say that ν is *non-negligible*). More precisely, for all $c > 0$, there exists a constant $N > 0$ such that for all $n \geq N$, $\nu(n) < n^{-c}$.

   State whether each of the following functions is negligible or non-negligible, and prove your assertion. For all of the problems below, we take the base of the logarithm to be 2.

   (a) $\nu(n) = 1/2^{100 \log n}$.

   Solution
   It is not negligible.
   Note that $\nu(n) = 1/2^{100 \log n} = 1/n^{100}$. Therefore, picking $c = 100$, we see that $\nu(n)$ is never strictly smaller than n^{-100} for any n. Therefore, $\nu(n)$ is not negligible.

   (b) $\nu(n) = 1/2^{100 \log n \cdot \log \log \log n}$.

   Solution
   It is negligible.
   Note that we can rewrite $\nu(n) = n^{-100 \log \log \log n}$. Consider some $c > 0$. Let N an integer such that $100 \log \log \log N > c$. Then, for all $n \geq N$, we have $\nu(n) = n^{-100 \log \log \log n} < n^{-c}$. Hence, $\nu(n)$ is negligible.

(c) $\nu(n) = p(n) \cdot \mu(n)$, where $p(n) = O(n^k)$ for some constant k, and $\mu(n)$ is a negligible function. Either prove that ν is always negligible, or come up with a counter-example.

Solution
It is always negligible.
Fix some $c > 0$. Since $\mu(n)$ is negligible, note that there exists some $N' > 0$ such that $\mu(n) < n^{-(c+k+1)}$ for all $n \geq N'$. Let $N'' > 0$ be such that $p(n) < n^{k+1}$ for all $n > N''$. Let $N = \max(N', N'')$. Then, $\mu(n) \cdot p(n) < n^{-(c+k+1)} \cdot n^{k+1} = n^{-c}$.

(d) $\nu(n) = (\mu(n))^{\frac{1}{p(n)}}$, where $p(n)$ and $\mu(n)$ are as defined in (c). Either prove that ν is always negligible, or come up with a counter-example.

Solution
It is not necessarily negligible. For example, let $\mu(n) = 2^{-n}$, and $p(n) = n$. Then, $\mu(n)^{\frac{1}{p(n)}} = (2^{-n})^{1/n} = 1/2$, which is not negligible.

(e) $\nu(n) = 1/2^{2^{\log^* n}}$, where $\log^* n$ is the number of times the logarithm function must be iteratively applied to n before the result is less than or equal to 1. More concretely,

$$\log^* n := \begin{cases} 0 & \text{if } n \leq 1 \\ 1 + \log^*(\log n) & \text{if } n > 1. \end{cases}$$

You may use the fact that $\log^* n \leq n$ without proof. (Hint: How does $\log^* n$ compare to $\log \log n$?)

Solution
It is not negligible. Note that $\log^* n \leq 1 + \log^*(\log n) \leq 2 + \log^*(\log \log n) \leq 2 + \log \log n$. Therefore, we have that

$$\nu(n) = 1/2^{2^{\log^* n}} \geq 1/2^{2^{2+\log \log n}} = 1/2^{4 \log n} = 1/n^4.$$

Therefore, $\nu(n)$ is not negligible.

2. *(11 points)* **Statistical and computational indistinguishability.** We think of distributions X, Y on a (finite) set Ω as functions $X, Y : \Omega \to [0, 1]$ such that for $\sum_{\omega \in \Omega} X(w) = \sum_{\omega \in \Omega} Y(\omega) = 1$. The statistical distance (also known as variational or L_1 distance) between X and Y is defined as

$$\Delta(X, Y) := \frac{1}{2} \sum_{\omega \in \Omega} |X(\omega) - Y(\omega)| .$$

(a) *(3 points)* Show that the following is an equivalent definition:

$$\Delta(X, Y) := \sup_{A \subseteq \Omega} |X(A) - Y(A)| \,,$$

where $X(A)$ is shorthand for $\sum_{\omega \in A} X(\omega)$.

Solution

Note that

$$\sum_{\omega \in \Omega} X(\omega) = \sum_{\omega \in \Omega} Y(\omega) = 1.$$

Let $B = \Omega \setminus A$. Hence, for any $A \subseteq \Omega$, we have that

$$\begin{aligned} X(A) + X(B) &= Y(A) + Y(B) \\ \Rightarrow X(A) - Y(A) &= Y(B) - X(B) \\ \Rightarrow |X(A) - Y(A)| &= |X(B) - Y(B)| \end{aligned}$$

Now, we pick $A \subseteq \Omega$ containing all $\omega \in \Omega$ such that we have that $X(\omega) > Y(\omega)$. Note that this set A maximises the quantity $X(A) - Y(A)$ because any other element will have $X(\omega) - Y(\omega) < 0$, and removing any other ω reduces the quantity.

Then, we have that

$$\begin{aligned} \sum_{\omega \in \Omega} |X(\omega) - Y(\omega)| &= \sum_{\omega \in A} |X(\omega) - Y(\omega)| + \sum_{\omega \in B} |X(\omega) - Y(\omega)| \\ &= \sum_{\omega \in A} (X(\omega) - Y(\omega)) + \sum_{\omega \in B} (Y(\omega) - X(\omega)) \\ &= (X(A) - Y(A)) + (Y(B) - X(B)) \\ &= 2(X(A) - Y(A)) = 2 \sup_{A \subseteq \Omega} |X(A) - Y(A)| \end{aligned}$$

Therefore, we have

$$\Delta(X, Y) = \frac{1}{2} \sum_{\omega \in \Omega} |X(\omega) - Y(\omega)| = \sup_{A \subseteq \Omega} |X(A) - Y(A)|$$

as defined.

(b) *(3 points)* Let D_0 and D_1 be two distributions over Ω. Suppose that we play the following game with an algorithm $\mathcal{A}$. First, we pick at random a bit $b \leftarrow \{0,1\}$ and then we pick $x \leftarrow D_b$ and we give x to $\mathcal{A}$. Finally, $\mathcal{A}$ returns a bit. It wins if the bit returned is equal to b. Show that the highest success probability in this game is exactly $\frac{1}{2} + \frac{1}{2}\Delta(D_0, D_1)$.

Solution

First, we give an algorithm $\mathcal{A}$ that gives us a probability of success of $\frac{1}{2} + \frac{1}{2}\Delta(D_0, D_1)$. The adversary, let's say Alice, plays the game as follows. Upon receiving $\omega \in \Omega$,

- If $D_0(\omega) > D_1(\omega)$, Alice guesses $b = 0$.
- If $D_1(\omega) \geq D_0(\omega)$, Alice guesses $b = 1$.

In other words, Alice's strategy is to guess the more likely distribution. We define $A \subseteq \Omega$ as the set of all $\omega \in \Omega$ such that $D_0(\omega) > D_1(\omega)$. Then, if $b = 0$, Alice wins with a probability $\sum_{\omega \in A} D_0(\omega) = D_0(A)$. On the other hand, if $b = 1$, Alice wins with a probability $\sum_{\omega \in \Omega \setminus A} D_1(\omega) = D_1(\Omega \setminus A)$. Since the chance of picking either distribution is $\frac{1}{2}$, this gives us

$$\begin{aligned}
\mathbf{Pr}(\text{Success}) &= \frac{1}{2}D_0(A) + \frac{1}{2}D_1(\Omega \setminus A) \\
&= \frac{1}{2}D_0(A) + \frac{1}{2}\left(1 - D_1(A)\right) \\
&= \frac{1}{2} + \frac{1}{2}(D_0(A) - D_1(A)) = \frac{1}{2} + \Delta(D_0, D_1)
\end{aligned}$$

Suppose there exists an algorithm $\mathcal{A}'$ that has a greater success rate. In other words, without loss of generality, we can assume that for some $\omega \in A$, there exists a better strategy than always reporting $b = 0$. Suppose $\mathcal{A}'$ reports $b = 0$ with probability p and $b = 1$ with probability $1 - p$. Let E_1 be the event that D_0 was chosen as the distibution, E_2 be the event that the output was ω. It is clear that $\mathbf{Pr}(E_1) = \frac{1}{2}$ and $\mathbf{Pr}(E_2) = \frac{1}{2}\left(D_0(\omega) + D_1(\omega)\right)$. Hence, we have that

$$\mathbf{Pr}(E_1|E_2) = \frac{\mathbf{Pr}(E_2|E_1)P(E_1)}{P(E_2)} = \frac{D_0(\omega)}{D_0(\omega) + D_1(\omega)}.$$

Solution
Hence, the probability of success for this algorithm is

$$\begin{aligned}
&\mathbf{Pr}(\text{Success of } \mathcal{A}' \text{ given that the input is } \omega) \\
&= p \cdot Pr(E_1|E_2) + (1-p) \cdot (1 - \mathbf{Pr}(E_1|E_2)) \\
&= \frac{p \cdot D_0(\omega) + (1-p) D_1(\omega)}{D_0(\omega) + D_1(\omega)} \\
&\leq \frac{p \cdot D_0(\omega) + (1-p) \cdot D_0(\omega)}{D_0(\omega) + D_1(\omega)} \\
&= \frac{D_0(\omega)}{D_0(\omega) + D_1(\omega)} = \mathbf{Pr}(\text{Success of } \mathcal{A} \text{ given that the input is } \omega)
\end{aligned}$$

Hence, we see that it is not possible for an algorithm to do better than algorithm $\mathcal{A}$ on any ω.

(c) *(1 point)* Give the definition of computational indistinguishability using similar language as in the previous question. (This part should not take more than 3-5 sentences.)

Solution
Suppose two distributions D_0 and D_1 are distributions over Ω. If D_0 and D_1 are computationally indistinguishable, it means that if the game described in (b) is played by a probabilistic polynomial time algorithm $\mathcal{A}$, the highest probability of success in this game is $\frac{1}{2} + \text{negl}(n)$.

(d) *(4 points)* For a probability distribution D over Ω and positive integer m, let D^m denote the *product distribution* over Ω^m, obtained by drawing a tuple of m independent samples from D. Let $\mathcal{X} = \{ X_n \}_n$ and $\mathcal{Y} = \{ Y_n \}_n$ be ensembles of distributions that are efficiently sampleable (in PPT), and let $m(n) = \text{poly}(n)$ be some fixed polynomial. Prove that if $\mathcal{X}$ and $\mathcal{Y}$ are computationally indistinguishable, or, in symbols, $\mathcal{X} \stackrel{c}{\approx} \mathcal{Y}$, then $\{ X_n^{m(n)} \} \stackrel{c}{\approx} \{ Y_n^{m(n)} \}$. (Where do you use that X_n, Y_n are efficiently sampleable?) This shows that if two efficiently sampleable distributions are computationally indistinguishable given one sample, then they are also computationally indistinguishable given polynomially many samples.

Solution

Suppose that an adversary $\mathcal{A}$ distinguishes $\mathcal{X}$ from $\mathcal{Y}$ with advantage $\epsilon(n)$. We now construct a distinguisher for X_n and Y_n with advantage $\epsilon(n)/m(n)$. Consider a distinguisher $\mathcal{A}'$ that works as follows:

- Let w be the sample from the challenger. If $b = 0$, then $w \leftarrow X$. Otherwise, $w \leftarrow Y$.
- Sample $i \leftarrow \{1, 2, \ldots, m(n)\}$, i.e. a random number from $1, 2, \ldots, m(n)$.
- Construct a sample $\mathbf{v} = (x_1, x_2, \ldots, x_{i-1}, w, y_{i+1}, \ldots, y_m)$, where x_j are independent samples from X_n, and y_k are independent samples from Y_n. This can be done efficiently since the distributions are efficiently sampleable.
- Send the $\mathbf{v}$ to $\mathcal{A}$, and suppose $\mathcal{A}$ returns b'. Then, $\mathcal{A}'$ also returns b'.

For notational convenience, let $Z_n^{(i)}$ denote the distribution over $\Omega^{m(n)}$ obtained by taking i independent samples from X_n and $m(n) - i$ independent samples from Y_n, in that order. In other words, $Z_n^{(i)} = X \times X \times \cdots \times X \times Y \times \cdots \times Y$. In particular, note that when $\mathcal{A}'$ samples i, $\mathbf{v} \leftarrow Z_n^{(i)}$ if $b = 0$, and $\mathbf{v} \leftarrow Z_n^{i-1}$ if $b = 1$. Now, note that

$$\begin{aligned}\mathbf{Pr}[b' = 0 | \mathbf{v} \leftarrow X] &= \sum_{i=1}^{m(n)} \mathbf{Pr}[b' = 0 | \mathcal{A}' \text{ selects index } i \text{ to construct } \mathbf{v}] \cdot \mathbf{Pr}[\mathcal{A}' \text{ chooses } i] \\ &= \frac{1}{m(n)} \sum_{i=1}^{n} \mathbf{Pr}[b' = 0 | \mathbf{v} \leftarrow Z_n^{(i)}].\end{aligned}$$

A similar argument gives us

$$\mathbf{Pr}[b' = 0 | \mathbf{v} \leftarrow Y] = \frac{1}{m(n)} \sum_{i=1}^{n} \mathbf{Pr}[b' = 0 | \mathbf{v} \leftarrow Z_n^{(i-1)}].$$

Therefore, we have that

$$\begin{aligned}&|\mathbf{Pr}[b' = 0 | \mathbf{v} \leftarrow X] - \mathbf{Pr}[b' = 0 | \mathbf{v} \leftarrow Y]| \\ &= \frac{1}{m(n)} \left| \sum_{i=1}^{n} \mathbf{Pr}[b' = 0 | \mathbf{v} \leftarrow Z_n^{(i)}] - \sum_{i=1}^{n} \mathbf{Pr}[b' = 0 | \mathbf{v} \leftarrow Z_n^{(i-1)}] \right| \\ &= \frac{1}{m(n)} \left| \mathbf{Pr}[b' = 0 | \mathbf{v} \leftarrow Z_n^{(m(n))}] - \mathbf{Pr}[b' = 0 | \mathbf{v} \leftarrow Z_n^{(0)}] \right| \\ &= \frac{1}{m(n)} \left| \mathbf{Pr}[b' = 0 | \mathbf{v} \leftarrow Y^{m(n)}] - \mathbf{Pr}[b' = 0 | \mathbf{v} \leftarrow X^{m(n)}] \right| = \frac{\epsilon(n)}{m(n)}.\end{aligned}$$

Therefore, this gives a non-negligible advantage in distinguishing X_n and Y_n, which is a contradiction.

3. *(8 points)* **PRG or not?** Let $G : \{0,1\}^{2n} \to \{0,1\}^{\ell}$ be a pseudorandom generator, where $\ell \geq 2n+1$. In each of the following, say whether G_c is necessarily a pseudorandom generator. If yes, give a proof. Otherwise, show a counterexample. Your counterexamples must rely only on the existence of pseudorandom generators.

   (a) *(2 points)* Consider $G_0 : \{0,1\}^{2n} \to \{0,1\}^{\ell}$, where $G_0(s) := G(\overline{s})$. Here, $\overline{s}$ is the bit-wise complement of s.

   Solution

   G_0 is a PRG. For notational convenience, let U_k be the uniform distribution on k bits. Suppose there were an adversary $\mathcal{A}$ which could distinguish $G_0(x)$, with $x \leftarrow U_{2n}$, from $y \leftarrow U_\ell$. Note that the random variable $G_0(x)$ (with $x \leftarrow U_{2n}$) is in fact distributed identically to the random variable $G(x)$ (with $x \leftarrow U_{2n}$), because $\overline{U_{2n}} = U_{2n}$. Therefore, an adversary $\mathcal{B}$ playing the distinguishing game for G can win with the same advantage that $\mathcal{A}$ does in the distinguishing game for G_0 by giving its ($\mathcal{B}$'s) challenge sample to $\mathcal{A}$ and outputting whatever $\mathcal{A}$ outputs. In the case where $\mathcal{B}$ receives a sample from $G(x)$ with $x \leftarrow U_{2n}$, $\mathcal{A}$ receives a sample from $G_0(x)$ with $x \leftarrow U_{2n}$; and, in the case where $\mathcal{B}$ receives a sample from U_ℓ, $\mathcal{A}$ also receives a sample from U_ℓ. It follows that $\mathcal{B}$ has the same advantage as $\mathcal{A}$.

   (b) *(3 points)* Consider $G_1 : \{0,1\}^{n} \to \{0,1\}^{\ell}$, where $G_1(s) := G(0^n || s)$.

   Solution

   G_1 is not always a PRG. Let G' be any PRG taking $2n$ bits to ℓ bits, and let G be the function

   $$G = \begin{cases} 0^\ell & s = 0^n \| s' \text{ for some } s' \\ G'(s) & \text{else} \end{cases}$$

   We firstly prove that G is a PRG. Suppose for contradiction that there is a distinguisher $\mathcal{D}$ such that

   $$\left| \Pr_{s \leftarrow_R \{0,1\}^{2n}}[\mathcal{D}(G(s)) = 1] - \Pr_{y \leftarrow_R \{0,1\}^{\ell}}[\mathcal{D}(y) = 1] \right| \geq \varepsilon(n)$$

   for some non-negligible function $\varepsilon(n)$.

Solution
(Continued.)
Since G and G' output different values on only 2^n bitstrings out of the 2^{2n} bitstrings in their common domain,

$$\begin{aligned}
&\left| \Pr_{s \leftarrow_R \{0,1\}^{2n}}[\mathcal{D}(G(s)) = 1] - \Pr_{s \leftarrow_R \{0,1\}^{2n}}[\mathcal{D}(G'(s)) = 1] \right| \\
&= \left| \sum_{s \in \{0,1\}^{2n}} \frac{1}{2^{2n}} \cdot (\mathcal{D}(G(s)) - \mathcal{D}(G'(s))) \right| \\
&= \left| \sum_{s \,:\, G(s) \neq G'(s)} \frac{1}{2^{2n}} \cdot (\mathcal{D}(G(s)) - \mathcal{D}(G'(s))) \right| \leq 2^{-n},
\end{aligned}$$

where we have used $\mathcal{D}(G(s))$ as a shorthand for $\Pr_{\text{internal coins of } \mathcal{D}}[\mathcal{D}(G(s)) = 1]$. Because G' is by assumption a secure PRG, we have

$$\left| \Pr_{s \leftarrow_R \{0,1\}^{2n}}[\mathcal{D}(G'(s)) = 1] - \Pr_{y \leftarrow_R \{0,1\}^{\ell}}[\mathcal{D}(y) = 1] \right| \leq \mu(n)$$

for some negligible function $\mu(n)$. Assuming wlog that $\Pr_{s \leftarrow_R \{0,1\}^{2n}}[\mathcal{D}(G(s)) = 1] \geq \Pr_{y \leftarrow_R \{0,1\}^{\ell}}[\mathcal{D}(y) = 1]$, and combining the last two block equations, we get

$$\begin{aligned}
&\Pr_{s \leftarrow_R \{0,1\}^{2n}}[\mathcal{D}(G(s)) = 1] - \Pr_{y \leftarrow_R \{0,1\}^{\ell}}[\mathcal{D}(y) = 1] \\
&\leq \Pr_{s \leftarrow_R \{0,1\}^{2n}}[\mathcal{D}(G'(s)) = 1] + 2 \cdot 2^{-n} - \Pr_{y \leftarrow_R \{0,1\}^{\ell}}[\mathcal{D}(y) = 1] \\
&\leq \mu(n) + 2 \cdot 2^{-n},
\end{aligned}$$

which is negligible. This is a contradiction, so $\mathcal{D}$ does not exist.
The output of $G_1 := G(0^n \| s)$ on $s \leftarrow U_n$ is always 0^ℓ, due to the way we have defined G, which is clearly distinguishable from uniformly random.

(c) *(3 points)* Consider $G_2 : \{0,1\}^{2n} \to \{0,1\}^{2\ell}$, where $G_2(s) := G(s)||G(s+1 \mod 2^{2n})$.

Solution

G_2 is not always a PRG. Let G' be any PRG taking $2n-1$ bits to ℓ bits. Define $G(s) := G'(\lfloor \frac{s}{2} \rfloor)$.

We firstly prove that G is a PRG. For notational convenience, let U_k be the uniform distribution on k bits. That the distribution of $G(s)$ (with $s \leftarrow U_{2n}$) is identical to the distribution of $G'(s')$ (with $s' \leftarrow U_{2n-1}$) can be verified by direct calculation: the probability that $\lfloor \frac{s}{2} \rfloor = s'$ for any given s' is exactly

$$\Pr[s = 2s'] + \Pr[s = 2s' + 1] = \frac{1}{2^{2n}} + \frac{1}{2^{2n}} = \frac{1}{2^{2n-1}}.$$

Therefore, for any adversary $\mathcal{A}$ that wins with non-negligible advantage in the distinguishing game for G', we can construct an adversary $\mathcal{B}$ playing the distinguishing game for G that wins with the same advantage that $\mathcal{A}$ does by giving its ($\mathcal{B}$'s) challenge sample to $\mathcal{A}$ and outputting whatever $\mathcal{A}$ outputs. In the case where $\mathcal{B}$ receives a sample from $G(s)$ with $s \leftarrow U_{2n}$, $\mathcal{A}$ receives a sample from $G'(s')$ with $s' \leftarrow U_{2n-1}$; and, in the case where $\mathcal{B}$ receives a sample from U_ℓ, $\mathcal{A}$ also receives a sample from U_ℓ. It follows that $\mathcal{B}$ has the same advantage as $\mathcal{A}$.

Now we prove that G_2 is not always a PRG. Observe that, in the case where s is even, $G(s)\|G(s+1 \mod 2^{2n})$ is of the form $y\|y$ for some $y \in \{0,1\}^\ell$. In the case where s is odd, $G(s)\|G(s+1 \mod 2^{2n})$ is of the form $G'(s')\|G'(s'+1 \mod 2^{2n-1})$ for uniformly random $s' \in \{0,1\}^{2n-1}$. Consider the following two cases:

i. $G'(s')\|G'(s'+1 \mod 2^{2n-1})$, with $s' \leftarrow U_{2n-1}$, is computationally distinguishable from a uniformly random 2ℓ-bit string. In this case, setting $G := G'$ in the definition of G_2, it follows that G_2 is not a PRG for all possible definitions of G.

ii. $G'(s')\|G'(s'+1 \mod 2^{2n-1})$, with $s' \leftarrow U_{2n-1}$, is computationally indistinguishable from a uniformly random 2ℓ-bit string. In this case, we can more precisely characterise the distribution of $G(s)\|G(s+1 \mod 2^{2n})$ (with $s \leftarrow U_{2n}$): it will be of the form $y\|y$ half the time (whenever s is even), and computationally indistinguishable from uniformly random half the time (whenever s is odd). The latter means that, conditioned on s being odd, any computationally efficient predicate $P : \{0,1\}^{2\ell} \to \{0,1\}$ must be such that

$$\left| \Pr_{s \in \{0,1\}^{2n},\ s \text{ odd}} \left[P\big(\ G(s)\|G(s+1 \mod 2^{2n}) \ \big) = 1 \right] - \Pr_{x \leftarrow U_{2\ell}} [P(x) = 1] \right| = \text{negl}(n). \tag{1}$$

Solution
(Continued.)
In particular, this is so for the predicate that outputs 1 iff the input 2ℓ-bit string is of the form $y\|y$ for $y \in \{0,1\}^\ell$. Therefore, an adversary for G_2 (with $G_2(s) := G(s)\|G(s+1 \mod 2^{2n})$) which outputs 'PRG' whenever it encounters a string of the form $y\|y$, and outputs 'uniform' otherwise, will:

i. output 'PRG' with probability 1 when its input is a PRG sample and s is even;

ii. output 'PRG' with probability $\text{negl}(n)$ when its input is a PRG sample and s is odd (by equation (1), and since a coincidence of the form $x = y\|y$, $x \leftarrow U_{2\ell}$ occurs with $\text{negl}(n)$ probability);

iii. output 'PRG' with probability $\text{negl}(n)$ when its input is a uniformly random string in $\{0,1\}^{2\ell}$ (since a coincidence of the form $x = y\|y$, $x \leftarrow U_{2\ell}$ occurs with $\text{negl}(n)$ probability).

The advantage of this adversary is therefore at least $\frac{1}{2} - \text{negl}(n)$. It follows that G_2 is not always a PRG.

4. *(3 points)* **A PRG from a PRF.** Prove that, if F is a length preserving pseudorandom function, then $G(s) \stackrel{\text{def}}{=} F_s(\langle 1\rangle)\|F_s(\langle 2\rangle)\|\ldots\|F_s(\langle \ell\rangle)$, where $\langle i\rangle$ is the n-bit binary representation of i, is a pseudorandom generator with expands ℓ bits to $\ell \cdot n$ bits.

Solution
Let k be the key length of F. We prove that G is a secure PRG using the next-bit unpredictability definition of PRG security. That is, for all $i \in [\ell n]$ and all computationally bounded $\mathcal{A}$,

$$\Pr_{y=G(s),s\leftarrow U_k}[\mathcal{A}(y_1,\ldots,y_{i-1}) = y_i] \leq \frac{1}{2} + \text{negl}(n).$$

Suppose, for contradiction, that there exists an efficient $\mathcal{A}$ and an index $i \leq \ell n$ for which

$$\Pr_{y=G(s),s\leftarrow U_k}[\mathcal{A}(y_1,\ldots,y_{i-1}) = y_i] = \frac{1}{2} + \varepsilon(n)$$

for some non-negligible function $\varepsilon(n)$.
Consider the following adversary $\mathcal{B}$ which uses the adversary $\mathcal{A}$ to break the PRF security of F. $\mathcal{B}$ queries its function oracle $\mathcal{F}$ on inputs $\langle 1\rangle,\ldots,\langle \ell\rangle$, and gives the first $i-1$ bits of the string $y := \mathcal{F}(\langle 1\rangle)\|\ldots\|\mathcal{F}(\langle \ell\rangle)$ to $\mathcal{A}$. $\mathcal{A}$ outputs a guess b for the ith bit of y. $\mathcal{B}$ then outputs 'PRF' iff b agrees with the ith bit of y.

Solution
(Continued.) In the case where $\mathcal{F}$ is a random oracle, b agrees with the ith bit of y with probability exactly $\frac{1}{2}$, because the inputs that $\mathcal{B}$ queries $\mathcal{F}$ on are all different, so the random oracle will return ℓ independent and uniformly sampled n-bit strings. If $\mathcal{F}$ is a PRF, then by definition the string which $\mathcal{B}$ gives as input to $\mathcal{A}$ is equal to the first $i-1$ bits of $G(s)$ for uniformly chosen s. Therefore, b (by hypothesis) agrees with the ith bit of y with probability $\frac{1}{2}+\varepsilon(n)$ for some non-negligible function $\varepsilon(n)$. $\mathcal{B}$ therefore has non-negligible advantage in the PRF security game of F, which is a contradiction.

5. *(8 points)* **A Locking Scheme.**

Veronica claims to Lucy, her classmate in 6.5620, that she can read minds, but Lucy does not believe it one bit, and is willing to bet \$100 that it's all baloney. They decide to play a game where Veronica can prove to Lucy that she is a bonafide mind-reader. If Veronica is lying, she should not be able to win in the game; and if Veronica indeed possesses this supernatural power, Lucy should grant victory (and \$100) to Veronica at the end of the game (no matter how much she hates to lose). They decide to play the following game.

(a) Veronica (we'll call her V) sends Lucy a string v (of a certain length that they decided on), chosen at random.

(b) Lucy (we'll call her L) chooses a random bit $\sigma \in \{0,1\}$, and Veronica has to guess what this bit is. To do this, Lucy sends

$$\ell \leftarrow L(\sigma, v; r)$$

to Veronica, where r is Lucy's private random coins that only she knows.

(c) Now, Veronica reads Lucy's mind and guesses what σ is.

(d) Finally, Lucy "unlocks" her bit σ by sending σ and r to Veronica. Veronica verifies that ℓ is indeed $L(\sigma, v; r)$.

Veronica wins if her guess of σ is correct.

(a) *(1 point)* Veronica should not gain any information about Lucy's bit from viewing the lock (i.e. after step 2 of the game). In other words, a malicious (but computationally *bounded*) Veronica V^* should not be able to learn anything about the honest L's choice bit σ, no matter what initial message v^* she sent.

Using the notion of indistinguishability, give a formal definition of this *concealing property* of L.

Solution
Since we do not want Veronica to determine σ given $\ell = L(\sigma, v; r)$, this means that for every message v^*, we want that for a ppt algorithms V^*,

$$\mathbf{Pr}[\sigma \leftarrow \{0,1\}, \ell \leftarrow L(\sigma, v^*; r), V^*(\ell) = \sigma] = \frac{1}{2} + \mu(n)$$

where μ is a negligible function.

(b) *(1 point)* Lucy should not be able to unlock her bit both ways, otherwise she can always get away with not paying Veronica \$100 even if Veronica is a mind-reader (do you see why?) To ensure this, the locking algorithm L has to be "unmodifiable". Give a formal definition of this property.

Solution
For all but a negligible fraction of strings v generated by Veronica, there does not exist strings r and r' such that $L(0, v; r) = L(1, v; r')$, i.e. there does not exist a "locking" message ℓ that can be opened to both bit 0 or bit 1. In other words,

$$\mathop{\mathbf{Pr}}_{v}[\exists r, r' \text{ such that } L(0, v; r) = L(1, v; r')] = \epsilon(n)$$

for some negligible function ϵ.

(c) *(2 points)* Let G be any length-tripling function, i.e., one for which $|G(x)| = 3|x|$ for every $x \in \{0,1\}^*$. Give an upper bound on the probability, over the choice of a random $3n$-bit string v, that there exist two inputs $x_1, x_2 \in \{0,1\}^n$ such that $G(x_1) \oplus G(x_2) = v$.

Solution
Let $\mathcal{G} = \mathrm{Im}(G)$, and let $F : \mathcal{G} \times \mathcal{G} \to \{0,1\}^{3n}$ such that $F(x_1, x_2) = G(x_1) \oplus G(x_2)$. Now, it is easy to see that $|\mathrm{Im}(F)| \leq |\mathcal{G}| \times |\mathcal{G}| \leq 2^{2n}$. Hence, the probability is at most $\frac{2^{2n}}{2^{3n}} = \frac{1}{2^n}$.

(d) *(4 points)* Let G be a length-tripling PRG (which we have seen can be obtained from any PRG). Use G to construct a secure locking scheme (i.e. define the algorithms V and L), and prove that it is both concealing and unmodifiable according to your definitions.

Solution

We denote by U_k the uniform distribution on k bits. We define the algorithms in the locking phase as follows:

- The verifier $V(1^n)$ draws two random strings $v_0, v_1 \leftarrow U_{3n}$, and outputs $v := v_0||v_1$.
- The locker computes $L(\sigma, v; r) = G(r) \oplus v_\sigma$, where G is a length-tripling PRG as defined in (c).

In the unlocking phase, the locker simply reveals σ and r to the verifier.

Concealing property. First, we show that the locking scheme is "concealing", as defined in (a). Suppose a malicious verifier picks a string $v^* := v_0^*||v_1^*$. Suppose the locker picks σ. Then, we argue that the distribution from $\ell \leftarrow \{G(r) \oplus v_\sigma^*\}_{r \leftarrow U_n}$ is computationally indistinguishable from a random string drawn from U_{3n}. Suppose otherwise, and there exists an adversary $\mathcal{A}$ which distinguishes the two distributions. Then, consider an adversary $\mathcal{A}'$ for the PRG game which simply takes either an input $x \leftarrow \{G(r)\}_{r \leftarrow U_n}$ or $x \leftarrow U_{3n}$. Then, the adversary sends $\ell = x \oplus v_\sigma^*$ to adversary $\mathcal{A}$. If x was drawn from the PRG, then ℓ is distributed as $\{G(r) \oplus v_\sigma^*\}_{r \leftarrow U_n}$. Otherwise, it is distributed as $\{x \oplus v_\sigma^*\}_{x \leftarrow U_{3n}} = U_{3n}$. Therefore, $\mathcal{A}'$ can distinguish the two distributions, contradicting the security of the PRG. Therefore,

$$\{L(\sigma, v; r)\}_{r \leftarrow U_n} = \{G(r) \oplus v_\sigma^*\}_{r \leftarrow U_n} \overset{c}{\approx} U_{3n}.$$

By applying this equivalence twice, we have that

$$\{L(0, v^*; r)\}_{r \leftarrow U_n} \overset{c}{\approx} U_{3n} \overset{c}{\approx} \{L(1, v; r)\}_{r \leftarrow U_n},$$

giving us the concealing property.

Unmodifiable property. For the unmodifiable property, it suffices to show that overall all strings $v = v_0||v_1$, for all but a negligible fraction of v, there does not exist r and r such that

$$G(r) \oplus v_0 = G(r') \oplus v_1 \iff G(r) \oplus G(r') = v_0 \oplus v_1.$$

Solution

(Continued.) As seen in (c), over the choice of all $R \leftarrow U_{3n}$, we know that for at most a $1/2^n$ fraction of them, there exist r and r' such that $G(r) \oplus G(r') = R$. Since we choose v_0 and v_1 uniformly from U_{3n}, we have that $v_0 \oplus v_1$ is distributed uniformly over U_{3n}. Therefore, for all but at most $1/2^n$ fraction of v, we have that the message ℓ can only be opened to either $\sigma = 0$ or $\sigma = 1$. This gives us the "unmodifiable" property.