CS 276: Homework 1

Due Date: Sunday, Feb 15, 2026 at 8:59pm via Gradescope

Usage of LLMs/Generative AI tools is prohibited. Other online resources (textbooks/lecture notes) are permissible.

1. Let f be a length-preserving one way function $f : \{0,1\}^n \to \{0,1\}^n$. Given $k > 0$, use f to build a one way function g such that g^k is a secure one-way function but g^{k+1} is insecure.

   Solution (There are multiple constructions that work, here is one)

   Let $f : \{0,1\}^n \to \{0,1\}^n$ be a length-preserving one-way function. Let the input of g be $x = (x_1, x_2, \ldots, x_{k+1})$, where each chunk $x_i \in \{0,1\}^n$. Given $k > 0$, we construct $g : \{0,1\}^{n(k+1)} \to \{0,1\}^{n(k+1)}$ as follows:

   $$g(x) = 0^n \| x_1 \| x_2 \| \ldots \| x_{k-1} \| f(x_k)$$

   1. Insecurity of g^{k+1}:

   We first observe that $g^{k+1}(x) = 0^{n(k+1)}$ for all $x \in \{0,1\}^{n(k+1)}$ – this function has a trivial adversary that succeeds in inverting with probability 1 and is hence not one-way.

   2. Security of g^k:

   Observe that for any x, $g^k(x) = 0^{nk} \| f(x_1)$. Suppose there exists a PPT adversary $\mathcal{A}$ that inverts g^k with non-negligible probability, i.e.,

   $$\mu_{A,g}(n(k+1)) = \Pr_{x \leftarrow \{0,1\}^{n(k+1)}} \left[A(1^{n(k+1)}, g(x)) \in g^{-1}(g(x)) \right] = \text{non-negl}$$

   We can construct an adversary $\mathcal{B}$ that uses $\mathcal{A}$ to invert f as follows:

   - $\mathcal{B}$ receives a challenge $y \in \{0,1\}^n$ from the OWF challenger for f.
   - $\mathcal{B}$ runs $\mathcal{A}(0^{nk} \| y)$ to obtain $x' = (x'_1, \ldots, x'_{k+1})$ and outputs x'_1.

   Note that by construction, if $\mathcal{A}$ succeeds, then $g^k(x') = (0^{nk} \| y) \iff f(x'_1) = y$. Hence, there is no loss in advantage between the two adversaries, and $\mu_{B,f}(n) = \text{non-negl}$ as well, contradicting the fact that f is a one-way function.

   ■

2. Suppose one way permutations exist. Does there exist a one-way permutation $f : \{0,1\}^n \to \{0,1\}^n$ with a fixed point, i.e. $f(0^n) = 0^n$?

   Solution Yes. Let $g : \{0,1\}^n \to \{0,1\}^n$ be a one-way permutation. Define $f : \{0,1\}^n \to \{0,1\}^n$ as $f(x) := g(x) \oplus g(0^n)$.
   First, observe that $f(0^n) = g(0^n) \oplus g(0^n) = 0^n$, so f has a fixed point. Second, since g is a permutation and XORing with a constant is a bijection, f is also a permutation.

To prove one-wayness, suppose there exists a PPT adversary $\mathcal{A}$ that inverts f with non-negligible probability:

$$\mu_{A,f}(n) = \Pr_{x \leftarrow \{0,1\}^n}\left[A(1^n, f(x)) \in f^{-1}(f(x))\right] = \text{non-negl}$$

We can construct an adversary $\mathcal{B}$ that uses $\mathcal{A}$ to invert g as follows:

- $\mathcal{B}$ receives a challenge $y \in \{0,1\}^n$ from the OWF challenger for g.
- $\mathcal{B}$ computes $y' = y \oplus g(0^n)$ and runs $x' \leftarrow \mathcal{A}(y')$.
- $\mathcal{B}$ outputs x'.

Note that if $\mathcal{A}$ succeeds, then $f(x') = y'$. By the definition of f, this implies:

$$g(x') \oplus g(0^n) = y \oplus g(0^n) \iff g(x') = y$$

Hence, $\mathcal{B}$ succeeds with the exact same probability as $\mathcal{A}$, contradicting the one-wayness of g. ■

3. Prove or disprove the following:

   (a) Let F be a pseudorandom generator. Then, $G(s) := F(s) \oplus F(\bar{s})$ is also a pseudorandom generator.

   (b) Let $F = \{F_k\}$ be a pseudorandom function family with key length equal to input length. Then, $G_k(x) := F_{F_k(x)}(x)$ is a also pseudorandom function.

   (c) Let $F = \{F_k\}$ be a pseudorandom function family with key length equal to input length. Then, $G_k(x) := F_{F_x(k)}(x)$ is also a pseudorandom function.

Solution

(a) No. If this statement were true for all PRG F, we obtain a contradiction.
Suppose the statement were true for G constructed from F for any F. Then, construct $H(s) := G(s) \oplus G(\bar{s})$, which should also be a PRG. However, $H(s) = G(s) \oplus G(\bar{s}) = F(s) \oplus F(\bar{s}) \oplus F(\bar{s}) \oplus F(s) = 0$, which is trivially not a PRG (as the output is always 0, unlike a random string)

(b) Yes. We use a hybrid argument. Let $\mathcal{A}$ be a PPT adversary distinguishing G_k from a random function.

- **Hybrid H_0:** This is identical to the original PRF, $H_0(x) := F_{F_k(x)}(x)$.
- **Hybrid H_1:** In this hybrid $H_1(x) := F_{R(x)}(x)$.
  We replace the inner PRF application with a true random function $R : \{0,1\}^n \to \{0,1\}^n$ (where n is the size of the key). The oracle computes $k' \leftarrow R(x)$ and outputs $F_{k'}(x)$ (Note that if the same input is queried again, the same output $R(x)$ is used).
  Indistinguishability ($H_0 \approx H_1$): Since F is a secure PRF, the function $x \mapsto F_k(x)$ is computationally indistinguishable from a random function $R_1(x)$. Any adversary capable of distinguishing H_0 from H_1 would break the security of the outer F (keyed by k).

- **Hybrid H_2:** Here $H_2(x) := R_{out}(x)$.
  We replace the entire construction with a true random function $R_{out} : \{0,1\}^n \to \{0,1\}^n$. The oracle simply returns $y \leftarrow R_{out}(x)$.
  Indistinguishability ($H_1 \approx H_2$): To show indistinguishability between H_1 and H_2 we define sub-hybrids for each query $i \in \{0, \ldots, q\}$ of the adversary as follows:
  - $H_{1,i}$: Sample a uniformly random k and the first i queries, $x_j \in \{x_1, \ldots, x_i\}$, of the adversary are answered as $F_k(x_j)$, while the rest, $x_j \in \{i+1, \ldots, q\}$ are answered as $F_{R(x_j)}(x_j)$.

  It is clear that $H_{1,0} = H_1$ and $H_{1,q} = H_2$.
  First we will use the following single-query property, that follows from the security of the PRF for any x (by transitivity – $F_k(x) \approx R(x) \approx F_{k'}(x)$):

$$\Pr[\mathcal{A}(F_{k'}(x)) = 1] \approx \Pr[\mathcal{A}(F_k(x)) = 1], \text{ for } k, k' \text{ uniformly random.}$$

  Now assume that an adversary $\mathcal{A}$ can distinguish between H_1 and H_2 with a non-negligible probability ϵ, then by the triangular inequality there is an $i^* \in \{0, \ldots, q-1\}$ such that the adversary distinguishes H_{1,i^*} and H_{1,i^*+1} with probability at least ϵ/q. By construction of the hybrids an adversary $\mathcal{B}$, using the H_{1,i^*}, H_{1,i^*+1} distinguishing adversary $\mathcal{A}$, can break the above single-query security property of the PRF.

Since $H_0 \approx H_1$ and $H_1 \approx H_2$, the construction G_k is indistinguishable from a random function.

(c) No. Consider a PRF F and modify it to create F':

$$F'_k(x) = \begin{cases} 0^n & \text{if } k = 0^n \\ F_k(x) & \text{otherwise} \end{cases}$$

F' remains a secure PRF because a random key k is equal to 0^n with probability 2^{-n} (negligible). However, consider the construction $G_k(x) := F'_{F'_x(k)}(x)$. An adversary can query the specific input $x = 0^n$. The inner function computes $k_{in} = F'_{0^n}(k) = 0^n$ (by definition of F'). The outer function computes $F'_{k_{in}}(0^n) = F'_{0^n}(0^n) = 0^n$. Thus, $G_k(0^n)$ always outputs 0^n, whereas a random function would output 0^n with probability 2^{-n}, and hence is trivially not a PRF.

■

4. Construct a *puncturable* PRF from a PRG $G : \{0,1\}^n \to \{0,1\}^{2n}$. (Write down the description of F in terms of G, describe the puncture and eval algorithms, and show that it satisfies the security definition below)
   A PRF $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ is called *puncturable* if

   - $\exists$ PPT algorithms
     $\mathsf{puncture}(k, x)$ that outputs a punctured key k_{-x},
     $\mathsf{eval}(k_{-x}, x)$ such that $\mathsf{eval}(k_{-x}, x') = F_k(x') \; \forall \, x' \neq x$.

- For all x, even given the punctured key, $F_K(x)$ is still (computationally) indistinguishable from random, i.e.,
  $\forall$ nu-PPT A $\exists \epsilon(n), n_A$ such that $\forall n > n_A$ and $\forall x \in \{0,1\}^n$, we have that

$$\left| \Pr_K[A(\mathsf{puncture}(K, x), F_K(x)) = 1] - \Pr_{K,r}[A(\mathsf{puncture}(K, x), r) = 1] \right| < \epsilon(n)$$

Solution Let $G : \{0,1\}^n \to \{0,1\}^{2n}$ be a PRG. We denote the output of $G(s)$ as $G(s) = G_0(s) \| G_1(s)$, where $G_0(s)$ and $G_1(s)$ are the first and last n bits respectively.

We identify the input space $\{0,1\}^n$ with paths in a binary tree of depth n. For an input $x = x_1 x_2 \dots x_n$, the value $F_k(x)$ is computed by traversing the tree from the root k according to the bits of x (this is just the GGM construction):

- Parse input x as $x_1 \dots x_n \in \{0,1\}^n$.
- Let $s_0 = k$. For $i = 1$ to n: compute $s_i = G_{x_i}(s_{i-1})$. Output $F_k(x) = s_n$.

$\mathsf{puncture}(k, x)$ computes the siblings of the nodes along the path from the root to the leaf x and outputs those:

- Let $s_0 = k$.
- For $i = 1$ to n:
  - Compute the sibling value $c_i = G_{1-x_i}(s_{i-1})$.
  - Compute the next node on path $s_i = G_{x_i}(s_{i-1})$.
- Output $k_{-x} = (x, c_1, c_2, \dots, c_n)$. (We include x to identify the path).

This punctured key is designed to allow all other evaluations x' to be computed except when $x' = x$. $\mathsf{eval}(k_{-x}, x')$ does the following:

- Parse $k_{-x} = (x, c_1, \dots, c_n)$.
- Since $x' \neq x$, find the first index i where they differ. Let i be the smallest integer such that $x'_i \neq x_i$. This implies $x'_j = x_j$ for all $j < i$, and $x'_i = 1 - x_i$.
- The value c_i in k_{-x} corresponds to the node at depth i on the path to x'. Set $s_i = c_i$. For $j = i + 1$ to n: compute $s_j = G_{x'_j}(s_{j-1})$.
- Output s_n.

We use a hybrid argument to prove that $F_k(x)$ is indistinguishable from random given k_{-x}. Define a sequence of hybrids $H_0, \dots, H_n$:

- **Hybrid H_0:** This is the real game. The adversary receives k_{-x} generated from a random key k, and the challenge value is $y = F_k(x)$.
- **Hybrid H_j $(1 \leq j \leq n)$:** In this experiment, we modify the generation of the values on the path to x. Instead of deriving s_j (the node at depth j on the path to x) and its sibling c_j from s_{j-1}, we replace them with truly random independent values. Specifically:
  - Choose $s_j \leftarrow \{0,1\}^n$ and $c_j \leftarrow \{0,1\}^n$ uniformly at random.

  - The values $c_1, \ldots, c_{j-1}$ are generated honestly (or are random if $j > 1$, carried over from previous hybrids).
  - The values $c_{j+1}, \ldots, c_n$ and the challenge y are derived from s_j using the standard GGM construction (applying G).

Indistinguishability ($H_{j-1} \approx H_j$): The difference between H_{j-1} and H_j lies in how s_j and c_j are generated.

- In H_{j-1}, s_{j-1} is already a random string (from the definition of H_{j-1}). Then (s_j, c_j) are the output of $G(s_{j-1})$. Note that s_j is the component on the path x, and c_j is the sibling.
- In H_j, (s_j, c_j) are replaced by truly random strings.
- The adversary only sees c_j (part of k_{-x}) and values derived from s_j (subsequent c's and y). Therefore, distinguishing H_{j-1} from H_j is exactly equivalent to distinguishing the output of the PRG $G(s_{j-1})$ from a random string of length $2n$. By the security of the PRG, $|\Pr[H_{j-1} = 1] - \Pr[H_j = 1]| \leq \mathsf{negl}(n)$.

By the triangle inequality, the distance between H_0 and H_n is at most $n \cdot \mathsf{negl}(n)$, which is negligible. In Hybrid H_n, the value s_n (which is the challenge y) is chosen uniformly at random, and it is independent of all values in k_{-x} (since k_{-x} consists of $c_1 \ldots c_n$ which were generated in previous steps or are independent random values). Thus, in H_n, the adversary sees a truly random y independent of k_{-x}.

This matches the security requirement:

$$\left| \Pr_K[A(\mathsf{puncture}(K, x), F_K(x)) = 1] - \Pr_{K,r}[A(\mathsf{puncture}(K, x), r) = 1] \right| \leq \mathsf{negl}(n)$$

■