CS 442
Introduction to Cryptography

Lecture 12: Pseudorandom Functions - III

Instructor: Aarushi Goel

Spring 2026

Agenda

* Construction of PRFs from PRGs.
* Example Problems on PRFs
* Review

* Midterm on March 5 (in-class)

Defining Pseudorandom Functions

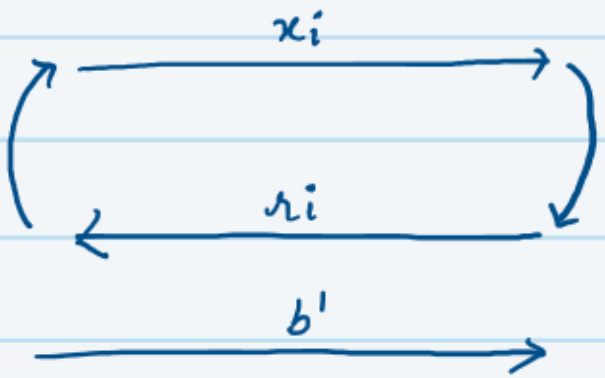
Definition: Let $G_{m,n,\kappa} = \{G_1, \ldots, G_{2^\kappa}\}$ be a set of functions such that each $G_i : \{0,1\}^m \to \{0,1\}^n$. This set of functions $G_{m,n,\kappa}$ is called a pseudorandom function if:

* each G_i can be computed in polynomial time.
* for every non-uniform PPT adversary, there exists a negligible function μ, such that $\forall n \in \mathbb{N}$, $Pr[b = b'] \leq \frac{1}{2} + \mu(n)$ in the following game:

Adv

Ch

polynomial number of queries



Sample $b \xleftarrow{\$} \{0,1\}$, $K \xleftarrow{\$} \{0,1\}^\kappa$

if $b = 0$: $r_i = G_K(x_i)$

if $b = 1$: $r_i \xleftarrow{\$} \{0,1\}^n$

(keeps a table for previous answers)

Constructing a PRF (From PRG to PRF with n-bit input)

Goldreich - Goldwasser - Micali Construction.

* Let $G: \{0,1\}^n \rightarrow \{0,1\}^{2n}$ be a length-doubling PRG.
* Let's define G_0 and G_1 as

$$G(s) = G_0(s) \| G_1(s)$$

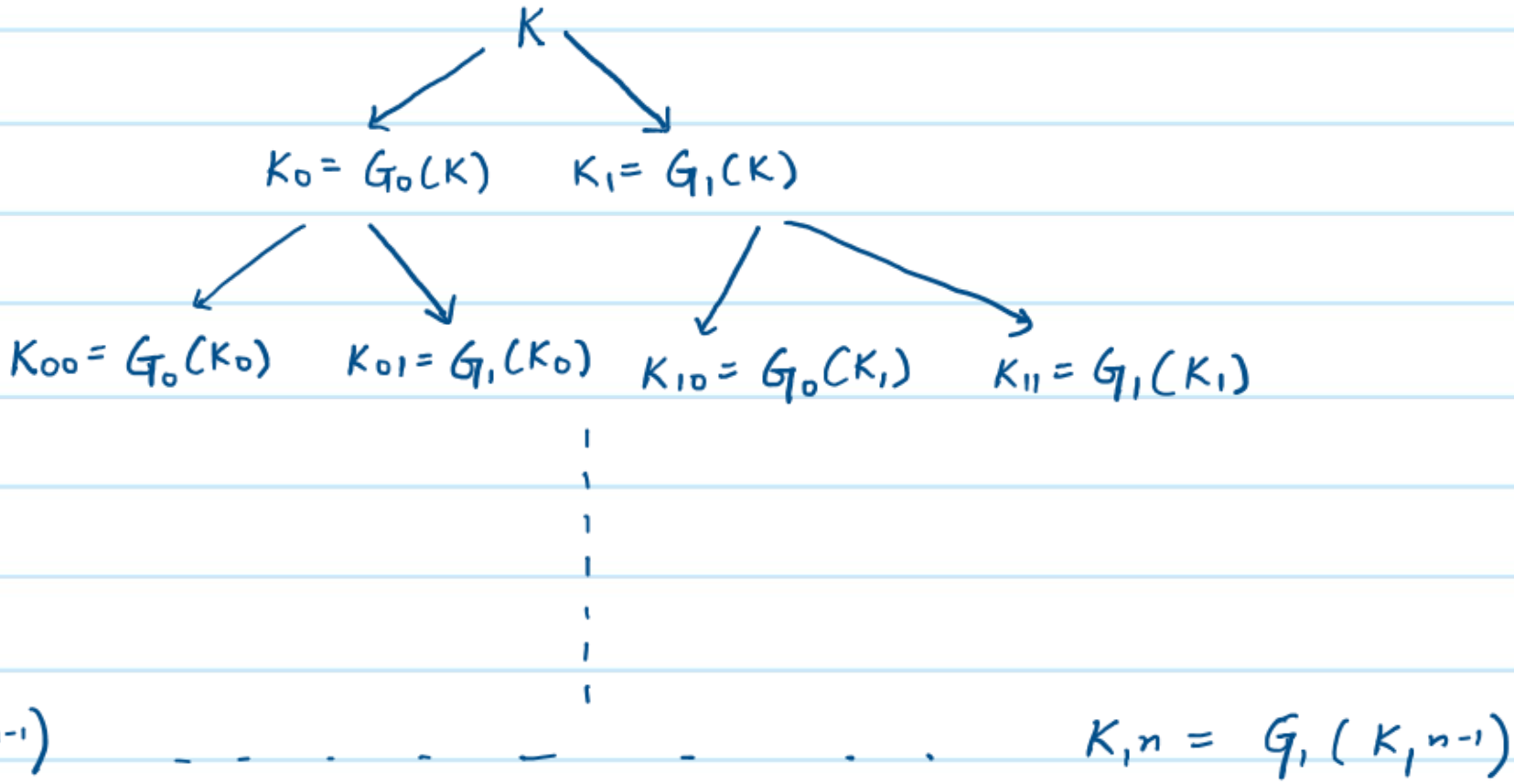
i.e., G_0 chooses left half of G and G_1 chooses the right half.

* PRF construction: Set $K = S$. On n-bit input $x \in \{0,1\}^n$, parse $x = x_1, x_2, \ldots, x_n$

$$F_K(x) = G_{x_n}\left(G_{x_{n-1}}\left(\ldots\left(G_{x_1}(K)\right)\ldots\right)\right)$$

$$F_K(x) = G_{x_n}(G_{x_{n-1}}(\dots(G_{x_1}(K))\dots))$$

* We can represent F_K as a binary tree of size 2^n.



Why is this a PRF? (Proof Strategy)

Let's try to prove that this a PRF. We consider the following hybrids:

H_1: $F_K(x_1 \dots x_n) = G_{x_n}(\dots(G_{x_2}(G_{x_1}(K)))\dots)$

H_2: $F_K(x_1 \dots x_n) = G_{x_n}(\dots(G_{x_2}(s_1))\dots)$, where $s_1 \xleftarrow{\$} \{0,1\}^n$

H_3: $F_K(x_1 \dots x_n) = G_{x_n}(\dots(s_2)\dots)$, where $s_2 \xleftarrow{\$} \{0,1\}^n$

$\vdots$

H_n: $F_K(x_1 \dots x_n) = s_n$, where $s_n \xleftarrow{\$} \{0,1\}^n$

* The function defined in H_n is a random function. All we need to show is that the outputs of function defined in H_1 are computationally indistinguishable from the outputs of the function defined in H_n.
By hybrid lemma, this boils down to showing that $\forall i \in [n-1]$, the outputs of the function defined in H_i are computationally indistinguishable from the outputs of the function defined in H_{i+1}.
* Let us consider H_1 and H_2 (all other H_i, H_{i+1} can be argued similarly)
We know that $G_{x_1}(K)$ is computationally indistinguishable from $s_1 \xleftarrow{\$} \{0,1\}^n$. Intuitively speaking, as a result, outputs of the form $G_{x_n}(\ldots(G_{x_2}(G_{x_1}(K)))\ldots)$ and $G_{x_n}(\ldots(G_{x_2}(s_1))\ldots)$ must also be computationally indistinguishable.
* Note that this is not a formal proof. We are ignoring several subtleties here. The full proof is out of scope for now.

Example # 1

Q Let $\{f_K\}_K$ be a family of PRFs. Is $\{g_K\}_K$ also a family of PRFs, where $g_K(x) = f_K(x) \| f_K(\bar{x})$?

No! Consider an adversary who queries the challenger on input 0^n and receive $y_1 \| y_2$. Next the adversary queries the challenger on input 1^n and receive $y_1' \| y_2'$. If $y_1 = y_2'$ and $y_1' = y_2$, then the adversary knows with a very high probability that the challenger must be using g_K to answer his queries as opposed to a random function. Hence, this is not a PRF because the outputs of this function are easy to distinguish from those of a random function.

Example #2

Q Let $\{f_K\}_K$ be a family of PRFs. Is $\{g_K\}_K$ also a family of PRFs, where $g_K(x) = f_K(0\|x) \| f_K(1\|x)$?

YES! Intuitively speaking, a single output of $g_K(\cdot)$ is equivalent to obtaining two different outputs of $f_K(\cdot)$ on related inputs. Since f_K is a PRF, we know that the outputs of $f_K(\cdot)$ are computationally indistinguishable from the outputs of a random function. Therefore, the outputs of $g_K(\cdot)$ must also be computationally indistinguishable from those of a random function.

Think! Why can't we use a similar argument in the previous example?

Review

* Kerckhoff's Principle
* Traditional Ciphers – No provable guarantees
* Perfectly Secure Encryption
    * Definition
    * One-time Pad encryption
    * Shannon's Theorem
    * Limitations of Perfectly secure encryption.
* Computational Security
    * Computational assumptions
    * Negligible functions
    * Computationally secure encryption

* Computational Indistinguishability
* Hybrid Lemma
* Pseudorandom generators
* Pseudorandom OTP encryption
* Proofs by reduction
* Pseudorandom functions