CS 442
Introduction to Cryptography

Lecture 11: Pseudorandom Functions - II

Instructor: Aarushi Goel

Spring 2026

Agenda

* Defining pseudorandom functions.
* Non-examples of pseudorandom functions.
* Constructing pseudorandom functions.

* Midterm on March 5 (in-class)
  You can bring 1 hand-written cheat sheet.

Random Functions

* Intuitively speaking, a random function outputs random values on all inputs.

The set of all functions of the form $F: \{0,1\}^m \to \{0,1\}^n$

Definition: For a randomly sampled $F \xleftarrow{\$} \mathcal{F}_{m,n}$, any distinct inputs $x_1, \ldots, x_t \in \{0,1\}^m$ and any outputs $y_1, \ldots, y_t \in \{0,1\}^n$, the following holds

$$\Pr\left[F(x_t) = y_t \,\middle|\, F(x_1) = y_1, \ldots, F(x_{t-1}) = y_{t-1}\right] = \frac{1}{2^n}$$

Pseudorandom Functions

* Pseudorandom functions (PRFs) *look like* a random function and can be described using polynomial number of bits.
* What does *look like* mean?
  → Look like random functions to a computationally bounded (i.e., PPT) adversary.
  → In other words, they are computationally indistinguishable from random functions.
* How do we define this formally using a game-based definition?
  → Can we ask the challenger to either send a description of a random function or a PRF to the adversary, and ask them to distinguish?

NO!!

* Recall that a random function requires exponentially many bits to describe, whereas a PRF can be described using polynomial number of bits.
* The adversary can easily tell the two apart by looking at the size of description.
* A bigger issue with this idea is that our adversary is PPT, that is, it is computationally bounded. This means it does not even have enough computational resources to read an exponential size description.

Defining Pseudorandom Functions

* We will only let the adversary query the function on inputs of its choice, and see the output. Based on these outputs, it must tell whether these are outputs of a random function or a PRF.

Q Should we keep the description of PRF hidden from the adversary?

A No! Remember Kerckoff's Principle? Security by obscurity is never a good idea.

* PRF will be keyed function. Only the key will be secret. PRF evaluation algorithm will be public.

Defining Pseudorandom Functions

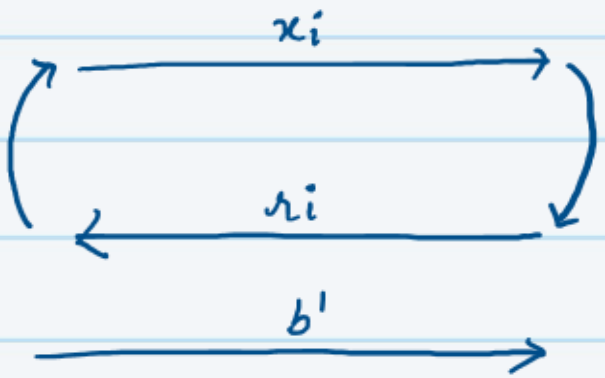
Definition: Let $G_{m,n,\kappa} = \{G_1, \ldots, G_{2^\kappa}\}$ be a set of functions such that each $G_i : \{0,1\}^m \to \{0,1\}^n$. This set of functions $G_{m,n,\kappa}$ is called a pseudorandom function if:

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

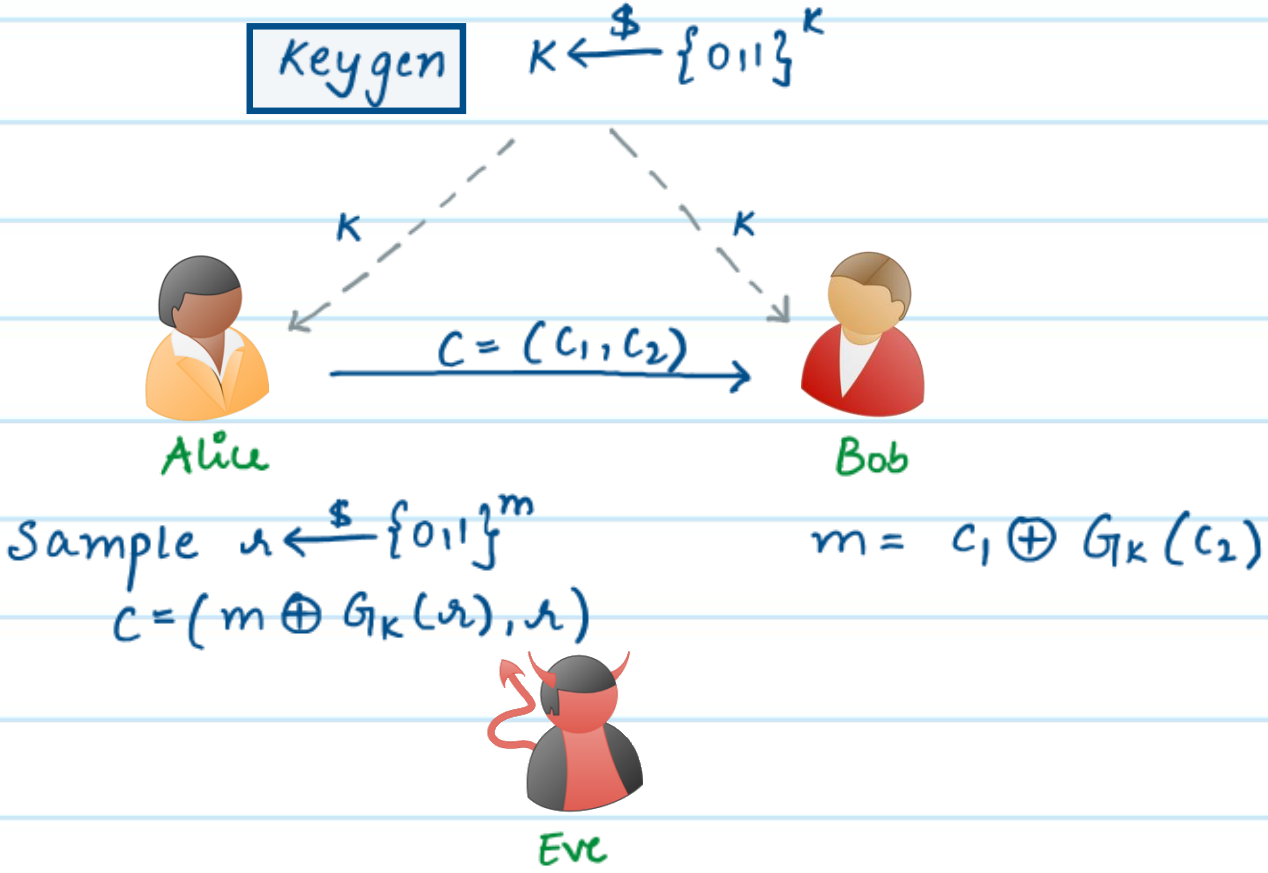
if $b = 0$: $r_i = G_K(x_i)$

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

(keeps a table for previous answers)

Encryption using PRFs

* Consider the following encryption scheme.



PRF or Not ??

Let $F^{(1)}$, $F^{(2)}$ be PRFs. Are the following PRFs or not?

#1 $F_K(x) = K \oplus x$

No! An adversary can simply query the challenger on inputs x_1, x_2. Let y_1, y_2 respectively be the response from the challenger. The adversary checks if $y_1 \oplus y_2 = x_1 \oplus x_2$, then the adversary knows with a high probability that the challenger chose F_K instead of a random function.

#2 $F_K(x_1 \| x_2) = F_K^{(1)}(x_1) \| F_K^{(2)}(x_2)$

No! An adversary can query the challenger on inputs $x_1 \| x_2$ and $x_1 \| x_3$. Let y_1, y_2 respectively be the response from the challenger. The adversary checks if the first half of y_1 is equal to the first half of y_2. If so, the adversary knows with a high probability that the the challenger chose F_K instead of a random function.

Constructing a PRF

* Let $G: \{0,1\}^n \to \{0,1\}^{2n}$ be a PRG.

Definition: A deterministic algorithm G is called a pseudorandom generator if:

* G can be computed in polynomial time.
* $|G(x)| > |x|$
* For every PPT adversary, $Pr[b = b'] = \frac{1}{2} + negl(|x|)$ in the following game

Adv

Ch

r (Adv → Ch)

b' (Adv ← Ch)

$b \xleftarrow{\$} \{0,1\}$

if $b = 0$: $r \xleftarrow{\$} \{0,1\}^{\ell(n)}$

if $b = 1$: $s \xleftarrow{\$} \{0,1\}^n$, $r = G(s)$

Constructing a PRF (From PRG to PRF with 1-bit input)

* As a warm-up, lets try to design a PRF of the form $F_K : \{0,1\} \to \{0,1\}^n$ where $K \in \{0,1\}^n$

* Since G is a length-doubling PRG, let $G(s) = y_0 \| y_1$, for $s \in \{0,1\}^n$. here $|y_0| = |y_1| = n$.

* PRF: Set $K = s$ and

$$F_K(0) = y_0 \quad , \quad F_K(1) = y_1$$

* What about PRFs with n-bit inputs?

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

Goldreich - Goldwasser - Micali Construction.

Oded Goldreich

Shafi Goldwasser

Silvio Micali

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

Goldreich - Goldwasser - Micali Construction.

* Let $G: \{0,1\}^n \to \{0,1\}^{2n}$ 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.