CS 442
Introduction to Cryptography

Lecture 10: Pseudorandom Functions - I

Instructor: Aarushi Goel

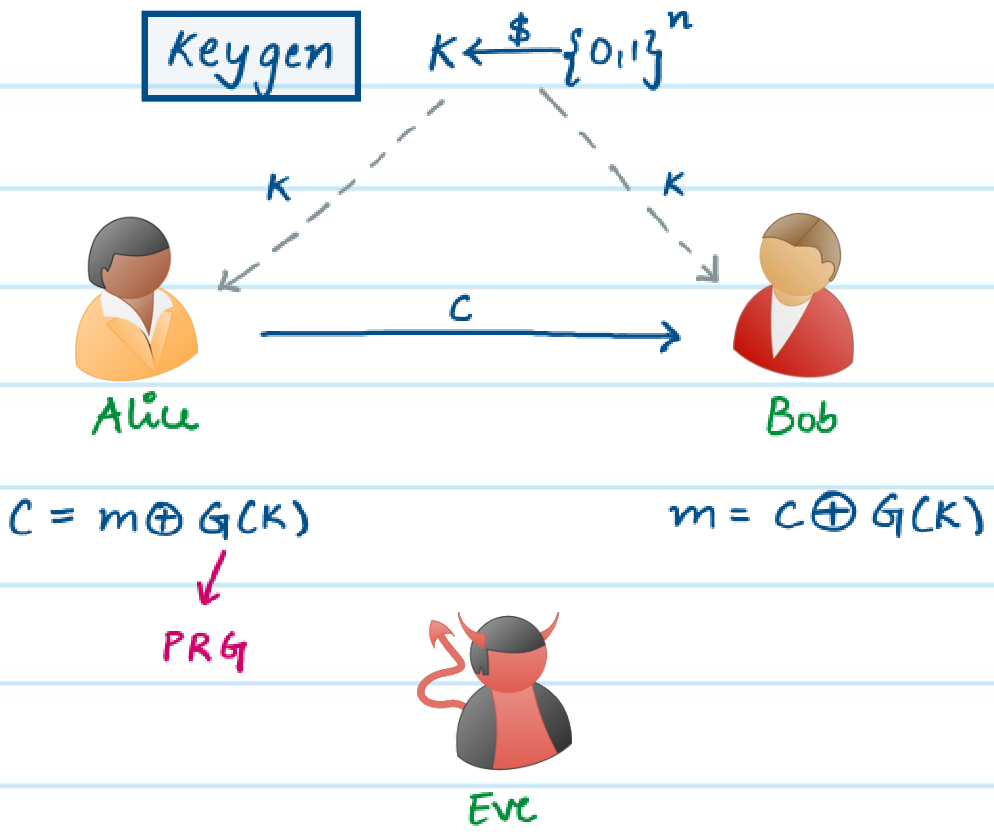
Spring 2026

Agenda

* Limitations of PRG-based Pseudorandom OTP encryption.
* Random functions
* Pseudorandom functions.

* HW2 is due this Sunday.

One-Time Pad Encryption



Nice Feature:

- * Perfectly secure, i.e., secure against all types of Eve, not just PPT eves

Limitations:

- * Key must be as long as the message
- * Each key can only be used once.

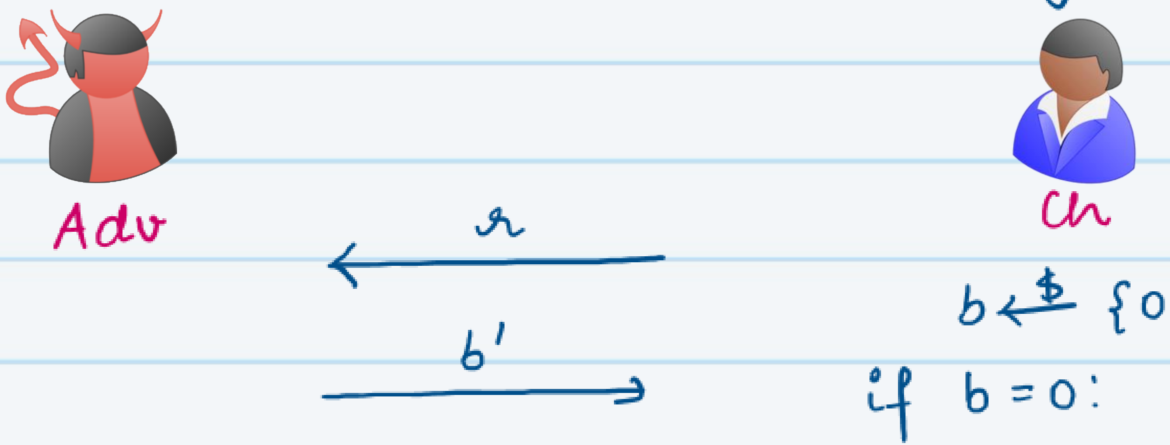
Pseudorandom Generators (PRG)

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

* G can be computed in poynomial time.

* $|G(x)| > |x|$

* For every PPT adversary, $\Pr[b = b'] = \frac{1}{2} + \mathrm{negl}(|x|)$ in the following game



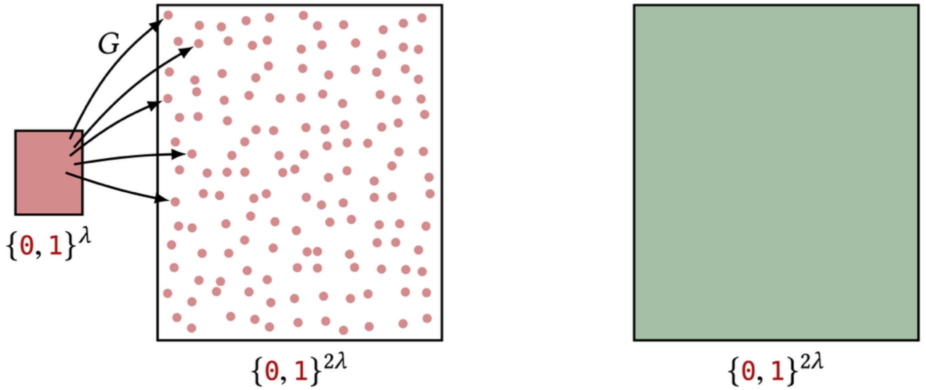
$b \xleftarrow{\$} \{0,1\}$

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

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

$r = G(s)$

Illustrating PRG $G : \{0,1\}^n \rightarrow \{0,1\}^{2n}$



pseudorandom distribution

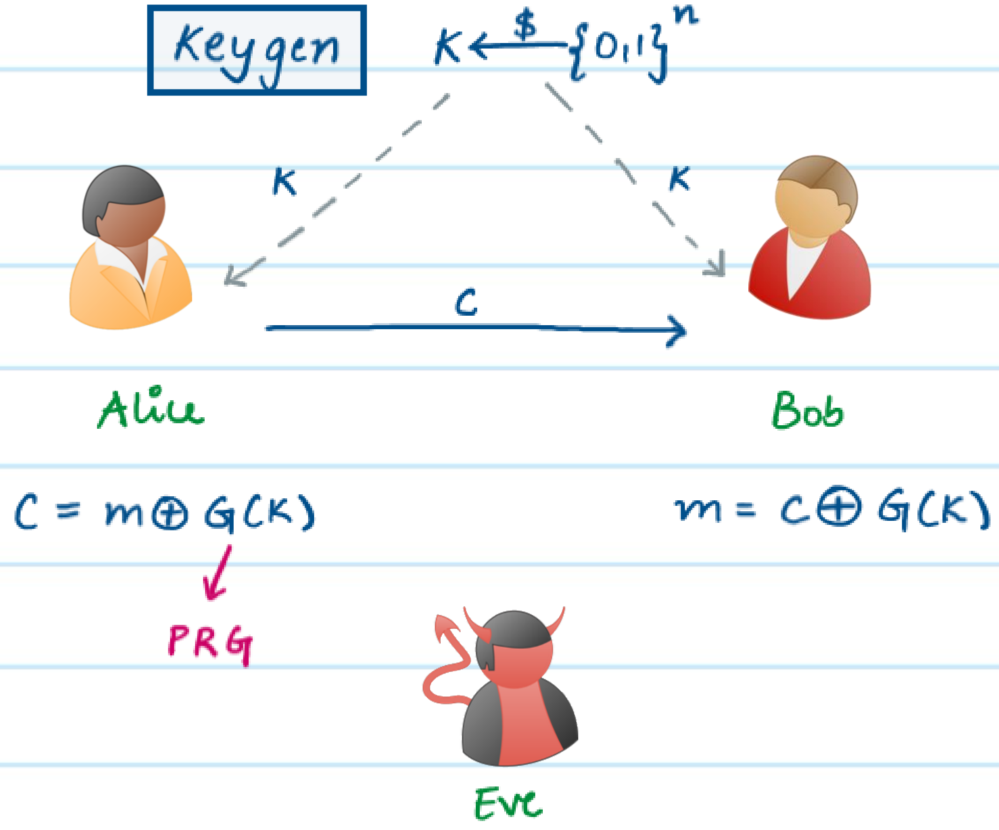
uniform distribution

* From a relative perspective, the PRG's output distribution is tiny. Out of the 2^{2n} possible strings in $\{0,1\}^{2n}$, only 2^n are possible outputs of G. These strings make up $2^n/2^{2n} = 1/2^n$ fraction of $\{0,1\}^{2n}$ — a negligible fraction.

* From an absolute perspective, the PRG's output distribution is huge. There are 2^n possible outputs of G, which is an exponential amount. This is large enough that a PPT adversary cannot distinguish it from the set $\{0,1\}^{2n}$.

Pseudorandom OTP Encryption Scheme

* Recall the computationally secure encryption scheme based on PRGs.



$C = m \oplus G(K)$

$m = C \oplus G(K)$

Nice Feature :

* Short Key

Limitations:

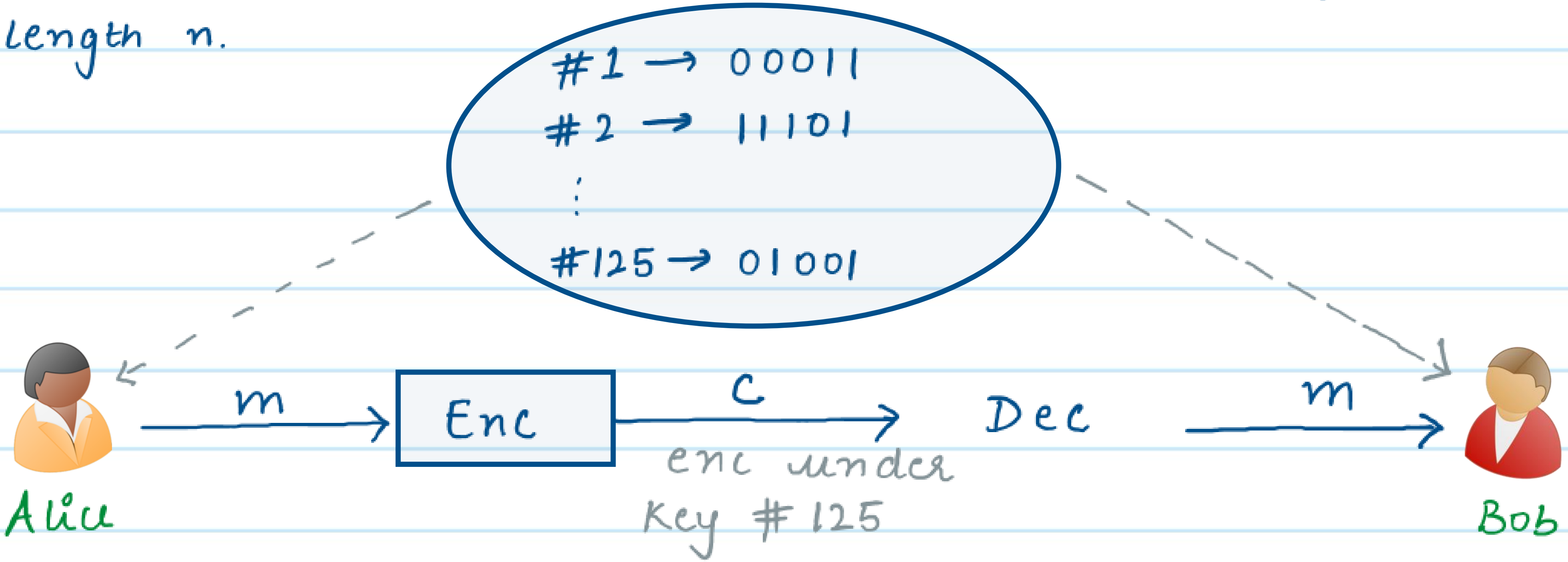
* Each key can only be used once.

Why?

Encrypting Multiple Messages

* Imagine if Alice and Bob had an exponential amount of shared randomness - not just a short key.

* They could split it up into n-bit chunks and use each one as a one-time pad whenever they want to send an encrypted message of length n.



* Although Alice publicly announces which location/chunk was used as each OTP Key, Eve doesn't Know the value at that location.

* Such an indexed list of exponentially many random values can be viewed as a *random function.*

Random Functions

* Consider a function $F: \{0,1\}^m \rightarrow \{0,1\}^n$.

* Let $\mathcal{F}_{m,n}$ be the set of all such functions that map m-length bit-strings to n-length bit strings.

* Each function $F \in \mathcal{F}_{m,n}$ can be uniquely represented by a list of length $\{0,1\}^m$, where the i-th entry in the list is the entry $F(i)$, $\forall\ i \in \{0,1\}^m$.

* This means, each entry in the list can be one on 2^n possible strings of length n. And there are total of 2^m such entries.

* So, the total number of distinct functions in the set $\mathcal{F}_{m,n}$ is

$$\overbrace{(2^n) \times (2^n) \times \ldots \times (2^n)}^{2^m\text{- times}} = (2^n)^{2^m}$$

* Each of these functions $F \in \mathcal{F}_{m,n}$ can be described using $n2^m$ bits

Randomly choosing one such function:

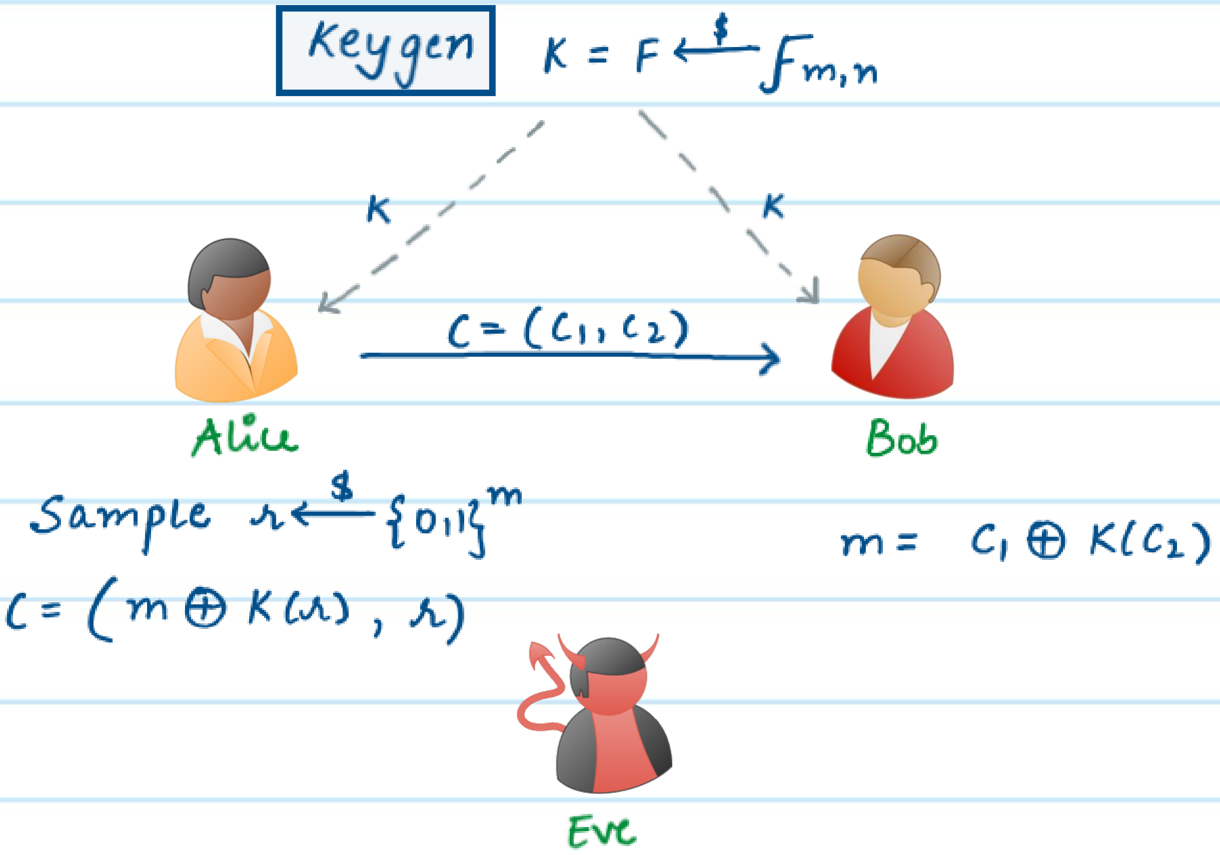
* Suppose we pick a random $F \xleftarrow{\$} \mathcal{F}_{m,n}$.
* Then, the evaluation of F at any input $x_1 \in \{0,1\}^m$ is uniformly random over $\{0,1\}^n$.
* Also, the evaluation of F at any other input $x_2 \in \{0,1\}^m$ given $F(x_1)$ is again uniformly random over $\{0,1\}^n$.
* In particular, the evaluation of F at an input x_t given $F(x_1), \ldots, F(x_{t-1})$ is uniformly random.
* Intuitively, the evaluation of a random F is completely unpredictable at any new input.

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

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

Encryption Using Random Functions

* Consider the following encryption scheme.



$K = F \xleftarrow{\$} \mathcal{F}_{m,n}$

Sample $r \xleftarrow{\$} \{0,1\}^m$

$c = (m \oplus K(r), r)$

$m = c_1 \oplus K(c_2)$

Nice Features of this Encryption:

Suppose the messages $m_1, \ldots, m_u$ are encrypted as ciphertexts $(c_1, r_1), \ldots, (c_u, r_u)$

* As long as $r_1, \ldots, r_u$ are all distinct, each one-time pad $F(r_1), \ldots, F(r_u)$ are uniform and independent of others.
  $\Rightarrow$ This scheme is perfectly secure.
* The probability that any two of the random values $r_1, \ldots, r_u$ are not distinct is very small.

Limitations:

* The secret key needs $n2^m$ bits to represent it, which is exponentially large.

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?

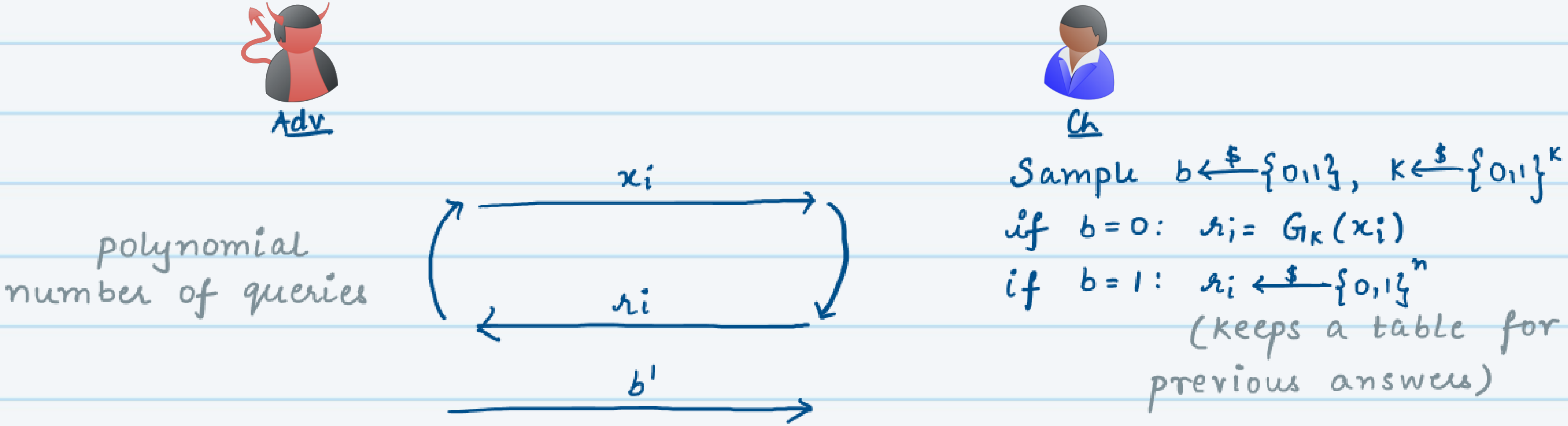
A No! Remember Kerckoff's Principle? Security by obscurity is never a good idea.

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

Defining Pseudorandom Functions

Definition: Let $\mathcal{G}_{m,n,k} = \{G_1, \dots, G_{2^k}\}$ be a set of functions such that each $G_i : \{0,1\}^m \to \{0,1\}^n$. This set of functions $\mathcal{G}_{m,n,k}$ 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:



Sample $b \xleftarrow{\$} \{0,1\}$, $k \xleftarrow{\$} \{0,1\}^k$

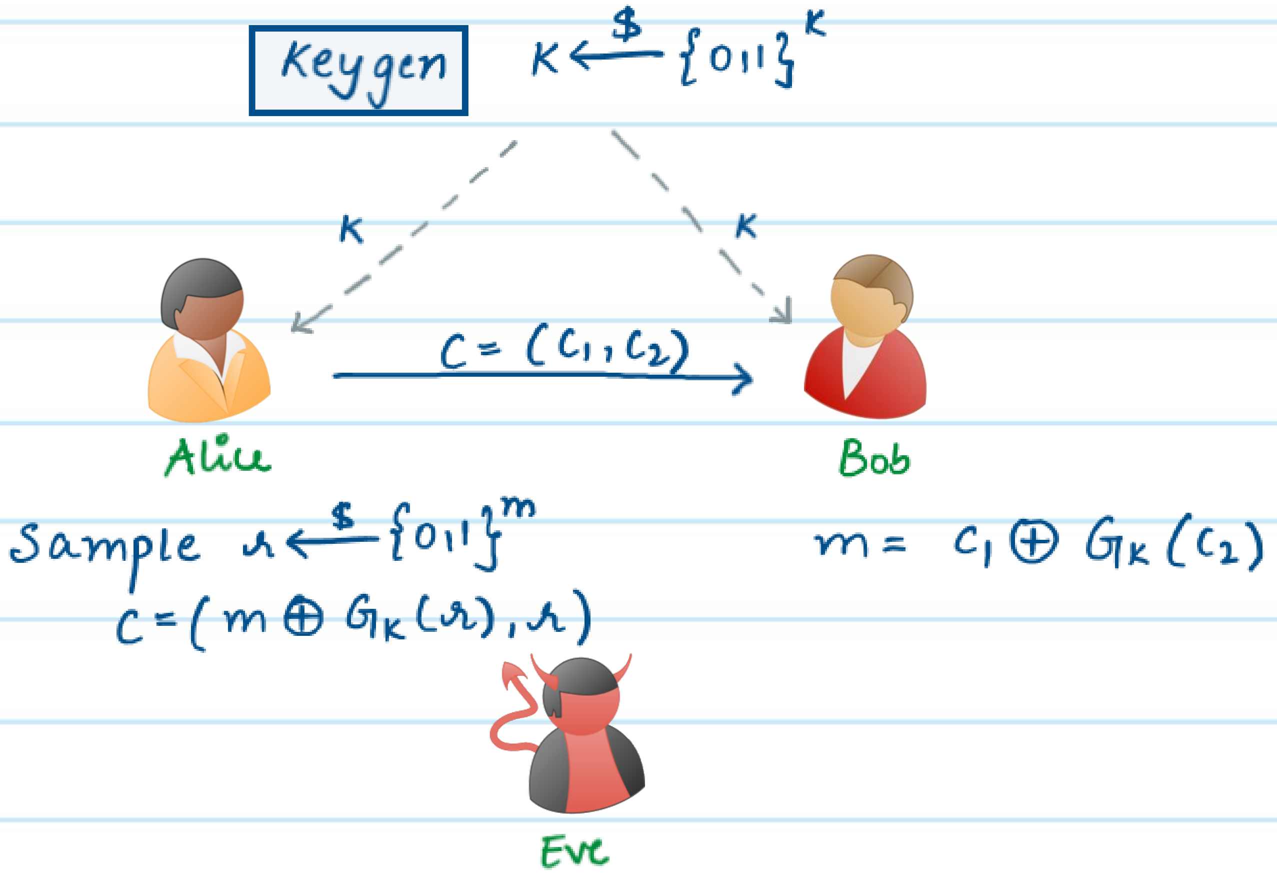
if $b = 0$: $r_i = G_k(x_i)$

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

(keeps a table for previous answers)

Encryption using PRFs

* Consider the following encryption scheme.



Keygen $K \xleftarrow{\$} \{0,1\}^K$

$c = (c_1, c_2)$

Sample $r \xleftarrow{\$} \{0,1\}^m$

$c = (m \oplus G_K(r), r)$

$m = c_1 \oplus G_K(c_2)$