CS 442
Introduction to Cryptography

Lecture 7: Computational Indistinguishability and Pseudorandom Generators

Instructor: Aarushi Goel

Spring 2026

Agenda

* Negligible Functions.
* Pseudorandom Generators
* Computational Indistinguishability
* Hybrid Lemma.

Computationally Secure Encryption.

An encryption scheme (KeyGen, Enc, Dec) with message space $\mathcal{M}$ is computationally secure if it satisfies correctness (as defined previously) and if for every PPT Eve, the following holds in the game below.

$$\Pr[b = b'] = \frac{1}{2} + \varepsilon$$

What is ε?
How do we define it?

Eve

Challenger

$\xrightarrow{\; m_1, m_2 \in \mathcal{M} \;}$

$\xleftarrow{\; c = \mathrm{Enc}(K, m_b) \;}$

$\xrightarrow{\; b' \;}$

$\mathrm{KeyGen} \rightarrow K$

$b \xleftarrow{\$} \{1, 2\}$

Negligible Functions

* Even the best PPT Eve should have an *extremely small* advantage
* One option is to consider exponentially small. But that is an overkill.
* We capture this using negligible functions.

Definition: A function $\nu(\cdot)$ is negligible, if for every polynomial $p(\cdot)$, we have $\lim_{n\to\infty} p(n)\cdot\nu(n) = 0$

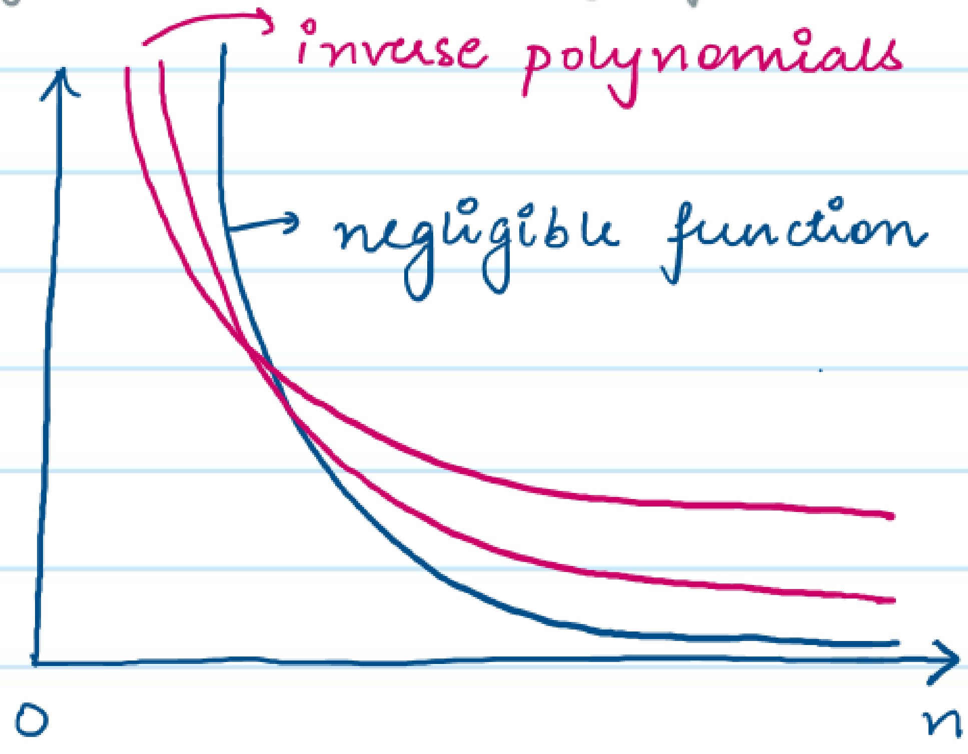
$\Rightarrow$ A negligible function decays faster than all inverse polynomial functions.

Definition: A function $\nu(n)$ is negligible if $\forall c \geq 0, \exists N$, s.t. $\forall n > N$, $\nu(n) \leq \frac{1}{n^c}$

order of quantifiers is important here (see Lecture 2)

Negligible Functions

A negligible function decays faster than all inverse polynomial functions.



Events that happen with negligible probability look to poly-time (& PPT) algorithms like they never occur

Computationally Secure Encryption.

An encryption scheme (KeyGen, Enc, Dec) with message space $\mathcal{M}$ is computationally secure if it satisfies correctness (as defined previously) and if for every PPT Eve, the following holds in the game below.

$$\Pr[b = b'] = 1/2 + \varepsilon(\lambda)$$

negligible function in the security parameter

Eve

$m_1, m_2 \in \mathcal{M}$ $\longrightarrow$

$\longleftarrow$ $c = \mathrm{Enc}(K, m_b)$

b' $\longrightarrow$

Challenger

$\mathrm{KeyGen} \rightarrow K$

$b \xleftarrow{\$} \{1, 2\}$

Examples of Negligible Functions

* **Ex 1:** $\frac{1}{2^n}$ This is negligible since for any polynomial $p(n) = n^c$, there always exists N, such that $\forall\, n > N$, $\frac{1}{2^n} \leq \frac{1}{n^c}$. This is because $\frac{1}{2^n}$ is exponential, so it is asymptotically smaller than any inverse polynomial $1/n^c$.

* **Ex 2:** $2^{-\omega(\log n)}$. Recall that ω is defined as follows:

$f(n) = \omega(g(n))$ if $\forall c > 0$, $\exists\, n_0 > 0$, s.t. $\forall\, n > n_0$, it holds that

$$f(n) > c \cdot g(n)$$

$$\omega(\log n) > c.\log n \Rightarrow -\omega(\log n) < -c \cdot \log n$$

$$\begin{aligned} \Rightarrow 2^{-\omega(\log n)} &< 2^{-c.\log n} \\ &< 2^{-\log n^c} \\ &< \frac{1}{n^c} \end{aligned}$$

Examples of Functions that are Not Negligible

* **Ex 1:** $\frac{1}{n^2}$ This is not negligible since for polynomial n^3, & any $n \geq 1$,

$$\frac{1}{n^2} \not< \frac{1}{n^3}$$

* **Ex 2:** Let $f(n)$ & $g(n)$ be negligible functions.

  Then $\frac{f(n)}{g(n)}$ may or may not be negligible.

  - Let $f(n) = \frac{1}{2^n}$ & $g(n) = \frac{1}{4^n}$

    $f(n)/g(n) = \frac{4^n}{2^n} = 2^n$ which is clearly not negligible

  - Let $f(n) = \frac{1}{4^n}$ & $g(n) = \frac{1}{2^n}$

    $f(n)/g(n) = \frac{1}{2^n}$ which is negligible.

Non-Negligible Functions.

Definition: A function $\nu(n)$ is non-negligible if $\exists c$, such that $\forall N$, $\exists n > N$, it holds that

$$\nu(n) \geq \frac{1}{n^c}$$

Candidate Construction for computationally Secure Encryption.

* Recall the construction of one-time pad encryption

  $K \oplus m = C$ → but this key must be as long as the message.

* Potential Idea: $K \xrightarrow{G} G(K)$

  small key — some expansion function.

  $G(K) \oplus m = C$

* What is G? Can it be something like $K \xrightarrow{G} K \| K \| K \| \ldots$?

  No! Remember Vignère cipher.

* G should be a *pseudorandom generator*!

Pseudorandom Generators (PRG)

* $G : \{0,1\}^n \rightarrow \{0,1\}^{\ell(n)}$, $\ell(n) > n$. PRGs are length expanding.
* PRGs are deterministic functions
* The output of a PRG is pseudorandom, i.e., it looks like a randomly sampled string to a computationally bounded adversary.

Adversary

Challenger

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

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

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

$r = G(s)$

Adversary $\xleftarrow{\quad r \quad}$ Challenger

Adversary $\xrightarrow{\quad b' \quad}$ Challenger

Adv wins if $b = b'$.

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

Adv

Ch

$$\text{Adv} \xleftarrow{\quad r \quad} \text{Ch}$$

$$\text{Adv} \xrightarrow{\quad b' \quad} \text{Ch}$$

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

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

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

$r = G(s)$

Computational Indistinguishability

* These type of game based definitions can be generalized.
* Let $\{A_n\}, \{B_n\}$ be distribution ensembles parameterized by n
* $\{A_n\}, \{B_n\}$ are computationally indistinguishable, if $\forall n \in \mathbb{N}$

PPT Adv

Ch

$\xleftarrow{\quad x \quad}$

$\xrightarrow{\quad b' \quad}$

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

if $b=0$: $x \xleftarrow{\$} A_n$

if $b=1$: $y \xleftarrow{\$} B_n$

$$\Pr[b' = b] = \frac{1}{2} + \nu(n)$$

↳ negligible function.

Computational Indistinguishability

An equivalent definition.

Definition: Distribution ensembles $\{A_n\}$, $\{B_n\}$ are computationally indistinguishable if $\forall$ PPT distinguishing tests T, $\exists$ negligible function $\nu(.)$, such that $\forall n \in \mathbb{N}$,

$$\left| \Pr_{x \leftarrow A_n}[T_n(x) = 0] - \Pr_{x \leftarrow B_n}[T_n(x) = 0] \right| \leq \nu(n)$$

$$\{A_n\} \approx_c \{B_n\}$$

Why are these definitions equivalent?

$$\Pr[b' = b] = ?$$

$$= \Pr[b' = b = 0] + \Pr[b' = b = 1]$$

$$= \frac{1}{2} \cdot \Pr[b' = 0 \mid b = 0] + \frac{1}{2} \cdot \Pr[b' = 1 \mid b = 1]$$

$$= \frac{1}{2}\left(\Pr[b' = 0 \mid b = 0] + \left(1 - \Pr[b' = 0 \mid b = 1]\right)\right)$$

$$= \frac{1}{2} + \frac{1}{2}\left(\Pr[b' = 0 \mid b = 0] - \Pr[b' = 0 \mid b = 1]\right)$$

$$= \frac{1}{2} + \frac{1}{2}\left(\Pr_{x \leftarrow A_n}[T(x) = 0] - \Pr_{x \leftarrow B_n}[T(x) = 0]\right)$$

$$= \frac{1}{2} + \frac{\Delta(A_n, B_n)}{2}$$

$$\Pr[b' = b] \leq \frac{1}{2} + \frac{\Delta(A, B)}{2}$$

→ distinguishing advantage

→ should be negl(n)

Definition: Distribution ensembles $\{A_n\}$, $\{B_n\}$ are computationally indistinguishable if $\forall$ PPT distinguishing tests T, $\exists$ negligible function $\nu(.)$, such that $\forall n \in \mathbb{N}$,

$$\text{Advantage}(n) = \Pr[b' = b] - \frac{1}{2} \leq \nu(n)$$

Properties of Computational Indistinguishability

* **Closure:** If we apply a polytime operation (i.e., an efficient operation) on computationally indistinguishable ensembles $\{A_n\}, \{B_n\}$, they remain computationally indistinguishable. That is, $\forall$ PPT M,

$$\{A_n\} \approx_c \{B_n\} \Rightarrow \{M(A_n)\} \approx_c \{M(A_n)\}$$

why?

* **Transitivity:** If $\{A_n\}, \{B_n\}$ are computationally indistinguishable and $\{B_n\}, \{C_n\}$ are computationally indistinguishable, then $\{A_n\}, \{C_n\}$ are also computationally indistinguishable.

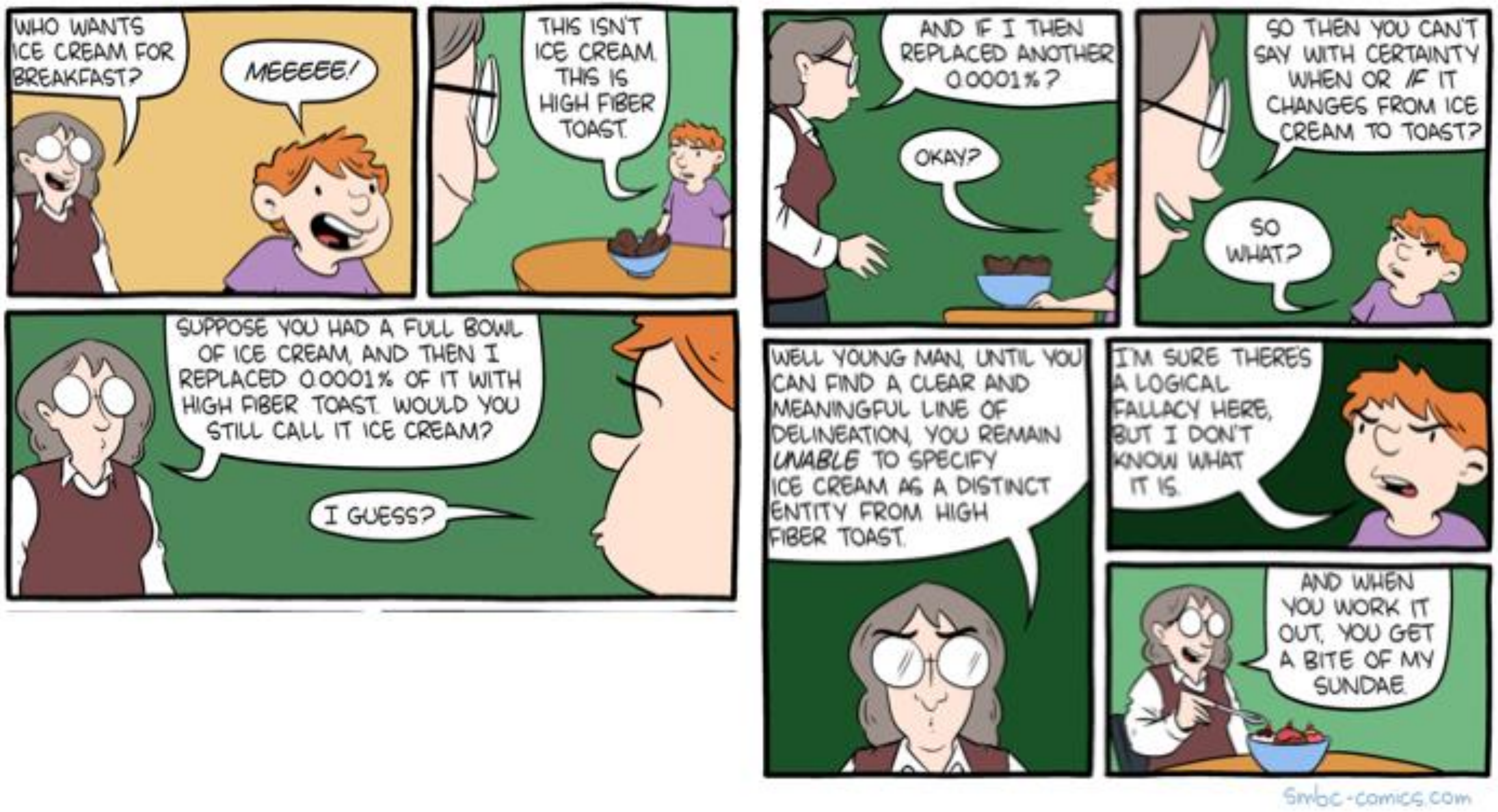
$$\{A_n\} \approx_c \{B_n\} \ \& \ \{B_n\} \approx_c \{C_n\} \Rightarrow \{A_n\} \approx_c \{C_n\}.$$

Generalizing Transitivity : Hybrid Lemma

Lemma: Let $\{A_n^1\}, \ldots, \{A_n^m\}$ be distribution ensembles, where $m = poly(n)$. If $\forall i \in [m-1]$, $\{A_n^i\}, \{A_n^{i+1}\}$ are computationally indistinguishable, then $\{A_n^1\}, \{A_n^m\}$ are computationally indistinguishable.

This lemma is used in most crypto proofs.

Hybrid Lemma



Contrapositive View of the Hybrid Lemma

Here is an alternate way to tate the hybrid lemma.

Lemma: Let $\{A_n^1\}, ..., \{A_n^m\}$ be distribution ensembles, where $m = poly(n)$. Suppose there exists a PPT adversary A, who can distinguish between $\{A_n^1\}, \{A_n^m\}$ with probability μ. Then there must exist $i \in [m-1]$, such that A can distinguish between $\{A_n^i\}$ and $\{A_n^{i+1}\}$ with probability at least μ/m.

$\Rightarrow$ if $\{A_n^1\}, \{A_n^m\}$ are computationally indistinguishable, then there cannot exist any $i \in [m-1]$ for which there exists a PPT adv who can distinguish between $\{A_n^i\}, \{A_n^{i+1}\}$ with non-negligible probability.