Lecture 6: Pseudorandomness - II

Recall: PRG from OWF

- Three steps:
  - Step 1: OWF (OWP) $\implies$ Hardcore Predicate for OWF (OWP)
  - Step 2: Hardcore Predicate for OWF (OWP) $\implies$ One-bit stretch PRG
  - Step 3: One-bit stretch PRG $\implies$ Poly-stretch PRG
- Last time: Step 2 for OWP and Step 3

Recall: PRG from OWF

- Three steps:
  - Step 1: OWF (OWP) $\implies$ Hardcore Predicate for OWF (OWP)
  - Step 2: Hardcore Predicate for OWF (OWP) $\implies$ One-bit stretch PRG
  - Step 3: One-bit stretch PRG $\implies$ Poly-stretch PRG
- Last time: Step 2 for OWP and Step 3
- **Today**: Step 1

One-way Function $\implies$ Hardcore Predicate

Theorem (Hardcore Predicate [Goldreich-Levin])

If $f\colon \{0,1\}^n \to \{0,1\}^n$ is a OWF, then:

One-way Function $\implies$ Hardcore Predicate

Theorem (Hardcore Predicate [Goldreich-Levin])

If $f\colon \{0,1\}^n \rightarrow \{0,1\}^n$ is a OWF, then:

- $g\colon \{0,1\}^{2n} \rightarrow \{0,1\}^{2n}$, *where* $g(x,r) := (\ f(x), r\ )$, *is also a OWF*

One-way Function $\implies$ Hardcore Predicate

Theorem (Hardcore Predicate [Goldreich-Levin])

If $f\colon \{0,1\}^n \to \{0,1\}^n$ is a OWF, then:

- $g\colon \{0,1\}^{2n} \to \{0,1\}^{2n}$, *where* $g(x,r) := (\ f(x), r\ )$, *is also a OWF*
- $h(x,r) := \langle x, r \rangle$ *is a hardcore predicate for* $g(x,r)$

One-way Function $\implies$ Hardcore Predicate

Theorem (Hardcore Predicate [Goldreich-Levin])

If $f\colon \{0,1\}^n \to \{0,1\}^n$ *is a OWF, then:*

- $g\colon \{0,1\}^{2n} \to \{0,1\}^{2n}$, *where* $g(x,r) := (\ f(x), r\ )$, *is also a OWF*
- $h(x,r) := \langle x, r\rangle$ *is a hardcore predicate for* $g(x,r)$

- Think: Reduction?

One-way Function $\implies$ Hardcore Predicate

Theorem (Hardcore Predicate [Goldreich-Levin])

If $f\colon \{0,1\}^n \to \{0,1\}^n$ is a OWF, then:

- $g\colon \{0,1\}^{2n} \to \{0,1\}^{2n}$, *where* $g(x,r) := (\ f(x), r\ )$, *is also a OWF*
- $h(x,r) := \langle x, r \rangle$ *is a hardcore predicate for* $g(x,r)$

- Think: Reduction?
- **Main challenge**: Adversary $\mathcal{A}$ for h only outputs 1 bit. Need to build an inverter $\mathcal{B}$ for f that outputs n bits.

Warmup Proof (1)

- Assumption: Given $g(x, r) = (f(x), r)$, adversary $\mathcal{A}$ *always* outputs $h(x, r)$ correctly

Warmup Proof (1)

- Assumption: Given $g(x, r) = (f(x), r)$, adversary $\mathcal{A}$ *always* outputs $h(x, r)$ correctly
- Inverter $\mathcal{B}$:

Warmup Proof (1)

- Assumption: Given $g(x,r) = (f(x), r)$, adversary $\mathcal{A}$ *always* outputs $h(x,r)$ correctly
- Inverter $\mathcal{B}$:
  - Compute $x_i^* \leftarrow \mathcal{A}(f(x), e_i)$ for every $i \in [n]$ where:

$$e_i = (\underbrace{0, \ldots, 0}_{(i-1)\text{-times}}, 1, \ldots, 0)$$

Warmup Proof (1)

- Assumption: Given $g(x,r) = (f(x), r)$, adversary $\mathcal{A}$ *always* outputs $h(x,r)$ correctly
- Inverter $\mathcal{B}$:
  - Compute $x_i^* \leftarrow \mathcal{A}(f(x), e_i)$ for every $i \in [n]$ where:

$$e_i = (\underbrace{0, \ldots, 0}_{(i-1)\text{-times}}, 1, \ldots, 0)$$

  - Output $x^* = x_1^* \ldots x_n^*$

Warmup Proof (2)

- Assumption: Given $g(x, r) = (f(x), r)$, adversary $\mathcal{A}$ outputs $h(x, r)$ with probability $3/4 + \varepsilon(n)$ (over choices of (x, r))

Warmup Proof (2)

- Assumption: Given $g(x,r) = (f(x), r)$, adversary $\mathcal{A}$ outputs $h(x,r)$ with probability $3/4 + \varepsilon(n)$ (over choices of (x,r))
- Define set S:

$$S := \left\{ x \colon \Pr[r \xleftarrow{\$} \{0,1\}^n : \mathcal{A}(f(x), r) = h(x,r)] \geqslant \frac{3}{4} + \frac{\varepsilon(n)}{2} \right\}$$

Warmup Proof (2)

- Assumption: Given $g(x,r) = (f(x), r)$, adversary $\mathcal{A}$ outputs $h(x,r)$ with probability $3/4 + \varepsilon(n)$ (over choices of (x,r))
- Define set S:

$$S := \left\{ x\colon \Pr[r \xleftarrow{\$} \{0,1\}^n : \mathcal{A}(f(x), r) = h(x,r)] \geqslant \frac{3}{4} + \frac{\varepsilon(n)}{2} \right\}$$

- $\Pr[x \in S] \geqslant \varepsilon(n)/2$

Warmup Proof (2)

- Assumption: Given $g(x,r) = (f(x), r)$, adversary $\mathcal{A}$ outputs $h(x,r)$ with probability $3/4 + \varepsilon(n)$ (over choices of (x,r))
- Define set S:

$$S := \left\{ x\colon \Pr[r \xleftarrow{\$} \{0,1\}^n : \mathcal{A}(f(x), r) = h(x,r)] \geqslant \frac{3}{4} + \frac{\varepsilon(n)}{2} \right\}$$

- $\Pr[x \in S] \geqslant \varepsilon(n)/2$
- Inverter $\mathcal{B}$:

Warmup Proof (2)

- Assumption: Given $g(x,r) = (f(x), r)$, adversary $\mathcal{A}$ outputs $h(x,r)$ with probability $3/4 + \varepsilon(n)$ (over choices of (x,r))
- Define set S:

$$S := \left\{ x\colon \Pr[r \xleftarrow{\$} \{0,1\}^n : \mathcal{A}(f(x), r) = h(x,r)] \geqslant \frac{3}{4} + \frac{\varepsilon(n)}{2} \right\}$$

- $\Pr[x \in S] \geqslant \varepsilon(n)/2$
- Inverter $\mathcal{B}$:
  - Let $a := \mathcal{A}(f(x), e_i + r)$ and $b := \mathcal{A}(f(x), r)$, for $r \xleftarrow{\$} \{0,1\}^n$

Warmup Proof (2)

- Assumption: Given $g(x,r) = (f(x),r)$, adversary $\mathcal{A}$ outputs $h(x,r)$ with probability $3/4 + \varepsilon(n)$ (over choices of (x,r))
- Define set S:

$$S := \left\{ x\colon \Pr[r \xleftarrow{\$} \{0,1\}^n : \mathcal{A}(f(x),r) = h(x,r)] \geqslant \frac{3}{4} + \frac{\varepsilon(n)}{2} \right\}$$

- $\Pr[x \in S] \geqslant \varepsilon(n)/2$
- Inverter $\mathcal{B}$:
  - Let $a := \mathcal{A}(f(x), e_i + r)$ and $b := \mathcal{A}(f(x), r)$, for $r \xleftarrow{\$} \{0,1\}^n$
  - Compute $c := a \oplus b$

Warmup Proof (2)

- Assumption: Given $g(x, r) = (f(x), r)$, adversary $\mathcal{A}$ outputs $h(x, r)$ with probability $3/4 + \varepsilon(n)$ (over choices of (x, r))
- Define set S:

$$S := \left\{ x \colon \Pr[r \xleftarrow{\$} \{0,1\}^n : \mathcal{A}(f(x), r) = h(x, r)] \geqslant \frac{3}{4} + \frac{\varepsilon(n)}{2} \right\}$$

- $\Pr[x \in S] \geqslant \varepsilon(n)/2$
- Inverter $\mathcal{B}$:
  - Let $a := \mathcal{A}(f(x), e_i + r)$ and $b := \mathcal{A}(f(x), r)$, for $r \xleftarrow{\$} \{0,1\}^n$
  - Compute $c := a \oplus b$
  - $c = x_i$ with probability $\frac{1}{2} + \varepsilon$ (Union Bound)

Warmup Proof (2)

- Assumption: Given $g(x,r) = (f(x), r)$, adversary $\mathcal{A}$ outputs $h(x,r)$ with probability $3/4 + \varepsilon(n)$ (over choices of (x,r))
- Define set S:

$$S := \left\{ x\colon \Pr[r \xleftarrow{\$} \{0,1\}^n : \mathcal{A}(f(x), r) = h(x,r)] \geqslant \frac{3}{4} + \frac{\varepsilon(n)}{2} \right\}$$

- $\Pr[x \in S] \geqslant \varepsilon(n)/2$
- Inverter $\mathcal{B}$:
  - Let $a := \mathcal{A}(f(x), e_i + r)$ and $b := \mathcal{A}(f(x), r)$, for $r \xleftarrow{\$} \{0,1\}^n$
  - Compute $c := a \oplus b$
  - $c = x_i$ with probability $\frac{1}{2} + \varepsilon$ (Union Bound)
  - Repeat and take majority to obtain x_i^* s.t. $x_i^* = x_i$ with prob. $1 - \mathsf{negl}(n)$

Warmup Proof (2)

- Assumption: Given $g(x,r) = (f(x),r)$, adversary $\mathcal{A}$ outputs $h(x,r)$ with probability $3/4 + \varepsilon(n)$ (over choices of (x,r))
- Define set S:

$$S := \left\{ x : \Pr[r \xleftarrow{\$} \{0,1\}^n : \mathcal{A}(f(x),r) = h(x,r)] \geqslant \frac{3}{4} + \frac{\varepsilon(n)}{2} \right\}$$

- $\Pr[x \in S] \geqslant \varepsilon(n)/2$
- Inverter $\mathcal{B}$:
  - Let $a := \mathcal{A}(f(x), e_i + r)$ and $b := \mathcal{A}(f(x), r)$, for $r \xleftarrow{\$} \{0,1\}^n$
  - Compute $c := a \oplus b$
  - $c = x_i$ with probability $\frac{1}{2} + \varepsilon$ (Union Bound)
  - Repeat and take majority to obtain x_i^* s.t. $x_i^* = x_i$ with prob. $1 - \mathsf{negl}(n)$
  - Output $x^* = x_1^* \ldots x_n^*$

Full Proof

Homework!

Food for Thought on PRGs

- OWF $\implies$ PRG: [Impagliazzo-Levin-Luby-89] and [Hastad-90]

Food for Thought on PRGs

- OWF $\implies$ PRG: [Impagliazzo-Levin-Luby-89] and [Hastad-90]
- More Efficient Constructions: [Vadhan-Zheng-12]

Food for Thought on PRGs

- OWF $\implies$ PRG: [Impagliazzo-Levin-Luby-89] and [Hastad-90]
- More Efficient Constructions: [Vadhan-Zheng-12]
- Computational analogues of Entropy

Food for Thought on PRGs

- OWF $\implies$ PRG: [Impagliazzo-Levin-Luby-89] and [Hastad-90]
- More Efficient Constructions: [Vadhan-Zheng-12]
- Computational analogues of Entropy
- Non-cryptographic PRGs and Derandomization: [Nisan-Wigderson-88]

Going beyond Poly Stretch

Going beyond Poly Stretch

- PRGs can only generate polynomially long pseudorandom strings

Going beyond Poly Stretch

- PRGs can only generate polynomially long pseudorandom strings
- Think: How to efficiently generate exponentially long pseudorandom strings?

Going beyond Poly Stretch

- PRGs can only generate polynomially long pseudorandom strings
- Think: How to efficiently generate exponentially long pseudorandom strings?

Idea: Functions that index exponentially long pseudorandom strings

Random Functions

- $\mathcal{F}_n :=$ set of all functions that map inputs from $\{0,1\}^n$ to $\{0,1\}^{\ell(n)}$

Random Functions

- $\mathcal{F}_n :=$ set of all functions that map inputs from $\{0,1\}^n$ to $\{0,1\}^{\ell(n)}$
- Think: What is $|\mathcal{F}_n|$?

Random Functions

- $\mathcal{F}_n :=$ set of all functions that map inputs from $\{0,1\}^n$ to $\{0,1\}^{\ell(n)}$
- Think: What is $|\mathcal{F}_n|$?
- A random function is $f \xleftarrow{\$} \mathcal{F}_n$

Oracle Algorithms

- Oracle O maps queries $q \in \{0,1\}^n$ to $\{0,1\}^{\ell(n)}$

Oracle Algorithms

- Oracle O maps queries $q \in \{0,1\}^n$ to $\{0,1\}^{\ell(n)}$
- Oracle algorithm $\mathcal{A}$ with "oracle access" to O is denoted as $\mathcal{A}^O$

Oracle Algorithms

- Oracle O maps queries $q \in \{0,1\}^n$ to $\{0,1\}^{\ell(n)}$
- Oracle algorithm $\mathcal{A}$ with "oracle access" to O is denoted as $\mathcal{A}^O$
- Time measure: Querying and receiving an answer from O takes unit time

Oracle Algorithms

- Oracle O maps queries $q \in \{0,1\}^n$ to $\{0,1\}^{\ell(n)}$
- Oracle algorithm $\mathcal{A}$ with "oracle access" to O is denoted as $\mathcal{A}^O$
- Time measure: Querying and receiving an answer from O takes unit time
- Think: Definition of PPT and n.u. PPT for oracle algorithms

Oracle Indistinguishability

Definition (Oracle Ensemble)

A sequence $\{O_n\}_{n\in\mathbb{N}}$ is an oracle ensemble if $\forall n \in \mathbb{N}$, O_n is a distribution over the set of all functions $f : \{0,1\}^n \to \{0,1\}^{\ell(n)}$

Oracle Indistinguishability

Definition (Oracle Ensemble)

A sequence $\{O_n\}_{n\in\mathbb{N}}$ is an oracle ensemble if $\forall n \in \mathbb{N}$, O_n is a distribution over the set of all functions $f : \{0,1\}^n \rightarrow \{0,1\}^{\ell(n)}$

Definition (Oracle Indistinguishability)

Oracle ensembles $\{O_n^0\}$ and $\{O_n^1\}$ are computationally indistinguishable if for every n.u. PPT oracle machine D, there exists a negligible function $\mu(\cdot)$ s.t.:

$$\left|\Pr\left[f \leftarrow O_n^0 : D^f\left(1^n\right) = 1\right] - \Pr\left[f \leftarrow O_n^1 : D^f\left(1^n\right) = 1\right]\right| \leqslant \mu(n)$$

Pseudorandom Functions

Intuition: An efficiently computable function that "looks like" a random function

Pseudorandom Functions

Intuition: An efficiently computable function that "looks like" a random function

Definition (Pseudorandom Functions)

A family of functions $\{f_s : \{0,1\}^n \rightarrow \{0,1\}^{\ell(n)}\}$ is a pseudorandom function (PRF) if:

Pseudorandom Functions

Intuition: An efficiently computable function that "looks like" a random function

Definition (Pseudorandom Functions)

A family of functions $\{f_s : \{0,1\}^n \rightarrow \{0,1\}^{\ell(n)}\}$ is a pseudorandom function (PRF) if:

- **Efficient Computation:** There exists a PPT F s.t. $F(s, x)$ efficiently computes the function $f_s(x)$

Pseudorandom Functions

Intuition: An efficiently computable function that "looks like" a random function

Definition (Pseudorandom Functions)

A family of functions $\{f_s : \{0,1\}^n \rightarrow \{0,1\}^{\ell(n)}\}$ is a pseudorandom function (PRF) if:

- **Efficient Computation:** There exists a PPT F s.t. $F(s,x)$ efficiently computes the function $f_s(x)$
- **Indistinguishability:**

$$\left\{ s \xleftarrow{\$} \{0,1\}^n : f_s \right\} \approx \left\{ f \xleftarrow{\$} \mathcal{F}_n : f \right\}$$

Pseudorandom Functions

Intuition: An efficiently computable function that "looks like" a random function

Definition (Pseudorandom Functions)

A family of functions $\{f_s : \{0,1\}^n \to \{0,1\}^{\ell(n)}\}$ is a pseudorandom function (PRF) if:

- **Efficient Computation:** There exists a PPT F s.t. $F(s,x)$ efficiently computes the function $f_s(x)$
- **Indistinguishability:**

$$\left\{ s \xleftarrow{\$} \{0,1\}^n : f_s \right\} \approx \left\{ f \xleftarrow{\$} \mathcal{F}_n : f \right\}$$

Typically, $\ell(n)$ will be equal to n

PRF from PRG [Goldreich-Goldwasser-Micali]

Goal: Construct a PRF $\{f_s : \{0,1\}^n \to \{0,1\}^n\}$ from a length-doubling PRG $G : \{0,1\}^n \to \{0,1\}^{2n}$

PRF from PRG [Goldreich-Goldwasser-Micali]

Goal: Construct a PRF $\{f_s : \{0,1\}^n \to \{0,1\}^n\}$ from a length-doubling PRG $G : \{0,1\}^n \to \{0,1\}^{2n}$

Construction of f_s:

- $G(s) = G_0(s), G_1(s)$ where $G_0, G_1 : \{0,1\}^n \to \{0,1\}^n$

PRF from PRG [Goldreich-Goldwasser-Micali]

Goal: Construct a PRF $\{f_s : \{0,1\}^n \rightarrow \{0,1\}^n\}$ from a length-doubling PRG $G : \{0,1\}^n \rightarrow \{0,1\}^{2n}$

Construction of f_s:

- $G(s) = G_0(s), G_1(s)$ where $G_0, G_1 : \{0,1\}^n \rightarrow \{0,1\}^n$
- $f_s(x) := G_{x_n}\left(G_{x_{n-1}}\left(\cdots G_{x_1}\left(s\right)\cdots\right)\right)$

PRF from PRG [Goldreich-Goldwasser-Micali]

Goal: Construct a PRF $\{f_s : \{0,1\}^n \to \{0,1\}^n\}$ from a length-doubling PRG $G : \{0,1\}^n \to \{0,1\}^{2n}$

Construction of f_s:

- $G(s) = G_0(s), G_1(s)$ where $G_0, G_1 : \{0,1\}^n \to \{0,1\}^n$
- $f_s(x) := G_{x_n}\left(G_{x_{n-1}}\left(\cdots G_{x_1}\left(s\right)\cdots\right)\right)$
- Think: Proof?

Food for Thought on PRFs

- PRFs from number-theoretic assumptions [Naor-Reingold97], lattices [Banerjee-Peikert-Rosen12]

Food for Thought on PRFs

- PRFs from number-theoretic assumptions [Naor-Reingold97], lattices [Banerjee-Peikert-Rosen12]
- PRFs with "Punctured" Keys [Sahai-Waters14]

Food for Thought on PRFs

- PRFs from number-theoretic assumptions [Naor-Reingold97], lattices [Banerjee-Peikert-Rosen12]
- PRFs with "Punctured" Keys [Sahai-Waters14]
- Constrained PRFs [Boneh-Waters13,Kiayias-Papadopoulos-Triandopoulos-Zacharias13,Boyle-Goldwasser-Ivan14]

Food for Thought on PRFs

- PRFs from number-theoretic assumptions [Naor-Reingold97], lattices [Banerjee-Peikert-Rosen12]
- PRFs with "Punctured" Keys [Sahai-Waters14]
- Constrained PRFs [Boneh-Waters13,Kiayias-Papadopoulos-Triandopoulos-Zacharias13,Boyle-Goldwasser-Ivan14]
- Related-key Security [Bellare-Cash10]: Should evaluation of $f_s(x)$ help predict $f_{s'}(x)$?

Food for Thought on PRFs

- PRFs from number-theoretic assumptions [Naor-Reingold97], lattices [Banerjee-Peikert-Rosen12]
- PRFs with "Punctured" Keys [Sahai-Waters14]
- Constrained PRFs [Boneh-Waters13,Kiayias-Papadopoulos-Triandopoulos-Zacharias13,Boyle-Goldwasser-Ivan14]
- Related-key Security [Bellare-Cash10]: Should evaluation of $f_s(x)$ help predict $f_{s'}(x)$?
- Key-homomorphic PRFs [Boneh-Lewi-Montgomery-Raghunathan13]: Given $f_s(x)$ and $f_{s'}(x)$, compute $f_{g(s,s')}(x)$