Pseudorandom Generators

CS 276: Introduction to Cryptography

Sanjam Garg

February 4, 2026

Overview

The Goal: Pseudorandomness

Objective

Transform a small amount of entropy into a distribution that closely resembles randomness.

The Goal: Pseudorandomness

Objective

Transform a small amount of entropy into a distribution that closely resembles randomness.

Key Idea

- Start with a small amount of entropy, known as the **seed**
- Use a **deterministic** process to generate a new distribution
- The output should appear **indistinguishable** from random

The Goal: Pseudorandomness

Objective

Transform a small amount of entropy into a distribution that closely resembles randomness.

Key Idea

- Start with a small amount of entropy, known as the **seed**
- Use a **deterministic** process to generate a new distribution
- The output should appear **indistinguishable** from random

Why This Matters

- Many cryptographic protocols need random bits
- True randomness is expensive to generate
- Can we use a short seed to generate many "random-looking" bits?

What is “Indistinguishable”?

Challenge

We need to define what it means for two distributions to be “indistinguishable” by an adversary.

What is "Indistinguishable"?

Challenge

We need to define what it means for two distributions to be "indistinguishable" by an adversary.

Two Approaches

1. **Statistical Indistinguishability**: Very strong, information-theoretic
   - Adversary is **unbounded** computationally
   - Too strong for most applications
2. **Computational Indistinguishability**: More practical
   - Adversary is **PPT** (probabilistic polynomial time)
   - This is what we'll use for PRGs

Definition of PRG

Definition 1 (Pseudorandom Generator)

A function $G : \{0,1\}^n \to \{0,1\}^{n+m}$ with $m = \text{poly}(n)$ is called a **pseudorandom generator** if:

1. G is computable in polynomial time
2. $U_{n+m} \approx G(U_n)$, where U_k denotes the uniform distribution on $\{0,1\}^k$

Definition of PRG

Definition 1 (Pseudorandom Generator)

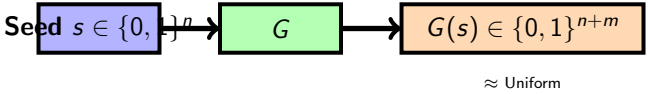
A function $G : \{0,1\}^n \to \{0,1\}^{n+m}$ with $m = \text{poly}(n)$ is called a **pseudorandom generator** if:

1. G is computable in polynomial time
2. $U_{n+m} \approx G(U_n)$, where U_k denotes the uniform distribution on $\{0,1\}^k$

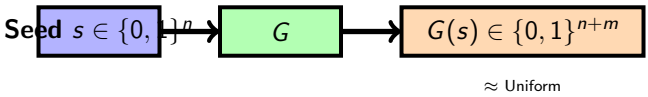
Intuition

- Takes a short seed of length n
- Outputs a longer string of length $n + m$ (where m is polynomial in n)
- Output is computationally indistinguishable from truly random

PRG: Visual Representation



PRG: Visual Representation



Key Properties

- **Expansion**: n bits $\rightarrow$ $n + m$ bits (where $m > 0$)
- **Efficiency**: Computable in polynomial time
- **Security**: Output looks random to any PPT adversary

Why PRGs are Useful

Applications

- **Stream Ciphers**: Generate long keystream from short key
- **Key Derivation**: Expand a master key into many session keys
- **Randomized Algorithms**: Provide randomness for algorithms
- **Foundational**: Building block for many cryptographic primitives

Why PRGs are Useful

Applications

- **Stream Ciphers**: Generate long keystream from short key
- **Key Derivation**: Expand a master key into many session keys
- **Randomized Algorithms**: Provide randomness for algorithms
- **Foundational**: Building block for many cryptographic primitives

The Fundamental Question

Can we construct PRGs from weaker assumptions? In particular, can we build PRGs from one-way functions?

PRG Extension: The Problem

Question

Given a PRG $G : \{0,1\}^n \to \{0,1\}^{n+1}$ that outputs just **one extra bit**, can we extend it to output many bits?

PRG Extension: The Problem

Question

Given a PRG $G : \{0,1\}^n \to \{0,1\}^{n+1}$ that outputs just **one extra bit**, can we extend it to output many bits?

Goal

Construct a PRG $F : \{0,1\}^n \to \{0,1\}^{n+l}$ where l is polynomial in n.

PRG Extension: The Problem

Question

Given a PRG $G : \{0,1\}^n \to \{0,1\}^{n+1}$ that outputs just **one extra bit**, can we extend it to output many bits?

Goal

Construct a PRG $F : \{0,1\}^n \to \{0,1\}^{n+l}$ where l is polynomial in n.

Key Insight

Use the PRG iteratively: each application gives us one more random bit and a new seed for the next iteration.

PRG Extension: Construction

Construction: PRG Extension

Given a PRG $G : \{0,1\}^n \to \{0,1\}^{n+1}$, construct $F : \{0,1\}^n \to \{0,1\}^{n+l}$ as follows:

1. Input: $S_0 \xleftarrow{\$} \{0,1\}^n$ (seed)
2. For $i = 1$ to l:
   - $(\sigma_i, S_i) := G(S_{i-1})$, where $\sigma_i \in \{0,1\}$ and $S_i \in \{0,1\}^n$
3. Output: $\sigma_1 \sigma_2 \cdots \sigma_l S_l$

PRG Extension: Construction

Construction: PRG Extension

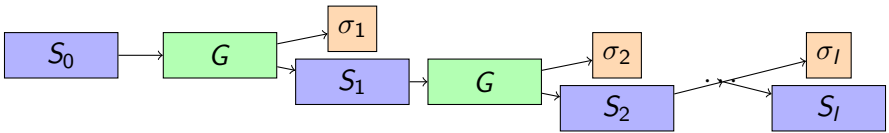
Given a PRG $G : \{0,1\}^n \to \{0,1\}^{n+1}$, construct $F : \{0,1\}^n \to \{0,1\}^{n+l}$ as follows:

1. Input: $S_0 \xleftarrow{\$} \{0,1\}^n$ (seed)
2. For $i = 1$ to l:
   - $(\sigma_i, S_i) := G(S_{i-1})$, where $\sigma_i \in \{0,1\}$ and $S_i \in \{0,1\}^n$
3. Output: $\sigma_1 \sigma_2 \cdots \sigma_l S_l$

Intuition

- Start with seed S_0
- Each iteration: use PRG to get one bit σ_i and new seed S_i
- Output all the bits $\sigma_1, \ldots, \sigma_l$ plus final seed S_l

PRG Extension: Visual Representation



Output: $\sigma_1\sigma_2\cdots\sigma_l S_l$

PRG Extension: Proof Strategy

Theorem 2

The function F constructed above is a PRG.

PRG Extension: Proof Strategy

Theorem 2

The function F constructed above is a PRG.

Proof Strategy: Hybrid Argument

- Define hybrids $H_0, H_1, \ldots, H_l$
- H_0: Output of F (all pseudorandom)
- H_l: Truly random output
- Show adjacent hybrids are indistinguishable
- By triangle inequality, $H_0 \approx H_l$

PRG Extension: Hybrid Definition

Hybrid H_i

For $i \in \{0, 1, \ldots, l\}$, define hybrid H_i:

1. Input: $S_0 \xleftarrow{\$} \{0,1\}^n$
2. $\sigma_1, \sigma_2, \ldots, \sigma_i \xleftarrow{\$} \{0,1\}$ (random bits)
3. $S_i \leftarrow S_0$
4. For $j = i + 1$ to l: $(\sigma_j, S_j) := G(S_{j-1})$
5. Output: $\sigma_1 \sigma_2 \cdots \sigma_l S_l$

PRG Extension: Hybrid Definition

Hybrid H_i

For $i \in \{0, 1, \ldots, l\}$, define hybrid H_i:

1. Input: $S_0 \xleftarrow{\$} \{0,1\}^n$
2. $\sigma_1, \sigma_2, \ldots, \sigma_i \xleftarrow{\$} \{0,1\}$ (random bits)
3. $S_i \leftarrow S_0$
4. For $j = i + 1$ to l: $(\sigma_j, S_j) := G(S_{j-1})$
5. Output: $\sigma_1 \sigma_2 \cdots \sigma_l S_l$

Key Observations

- $H_0 \equiv F$ (all bits from PRG)
- $H_l \equiv U_{n+l}$ (all bits random)
- H_i: first i bits random, rest from PRG

PRG Extension: Proof (Part 1)

Setup

Assume for contradiction that adversary $\mathcal{A}$ can distinguish H_0 from H_l with non-negligible advantage $v(n)$.
Define $\epsilon_i := \Pr[\mathcal{A}(1^n, H_i) = 1]$ for $i = 0, 1, \ldots, l$.

PRG Extension: Proof (Part 1)

Setup

Assume for contradiction that adversary $\mathcal{A}$ can distinguish H_0 from H_l with non-negligible advantage $v(n)$.
Define $\epsilon_i := \Pr[\mathcal{A}(1^n, H_i) = 1]$ for $i = 0, 1, \ldots, l$.

Triangle Inequality

$$\begin{aligned} |\epsilon_0 - \epsilon_l| &= |(\epsilon_0 - \epsilon_1) + (\epsilon_1 - \epsilon_2) + \cdots + (\epsilon_{l-1} - \epsilon_l)| \\ &\leq |\epsilon_0 - \epsilon_1| + |\epsilon_1 - \epsilon_2| + \cdots + |\epsilon_{l-1} - \epsilon_l| \end{aligned}$$

PRG Extension: Proof (Part 1)

Setup

Assume for contradiction that adversary $\mathcal{A}$ can distinguish H_0 from H_l with non-negligible advantage $v(n)$.
Define $\epsilon_i := \Pr[\mathcal{A}(1^n, H_i) = 1]$ for $i = 0, 1, \ldots, l$.

Triangle Inequality

$$\begin{aligned} |\epsilon_0 - \epsilon_l| &= |(\epsilon_0 - \epsilon_1) + (\epsilon_1 - \epsilon_2) + \cdots + (\epsilon_{l-1} - \epsilon_l)| \\ &\leq |\epsilon_0 - \epsilon_1| + |\epsilon_1 - \epsilon_2| + \cdots + |\epsilon_{l-1} - \epsilon_l| \end{aligned}$$

Key Step

Since $|\epsilon_0 - \epsilon_l| \geq v(n)$, there exists $k \in \{0, 1, \ldots, l-1\}$ such that:

$$|\epsilon_k - \epsilon_{k+1}| \geq \frac{v(n)}{l}$$

PRG Extension: Proof (Part 1)

Setup

Assume for contradiction that adversary $\mathcal{A}$ can distinguish H_0 from H_l with non-negligible advantage $v(n)$.
Define $\epsilon_i := \Pr[\mathcal{A}(1^n, H_i) = 1]$ for $i = 0, 1, \ldots, l$.

Triangle Inequality

$$\begin{aligned} |\epsilon_0 - \epsilon_l| &= |(\epsilon_0 - \epsilon_1) + (\epsilon_1 - \epsilon_2) + \cdots + (\epsilon_{l-1} - \epsilon_l)| \\ &\leq |\epsilon_0 - \epsilon_1| + |\epsilon_1 - \epsilon_2| + \cdots + |\epsilon_{l-1} - \epsilon_l| \end{aligned}$$

Key Step

Since $|\epsilon_0 - \epsilon_l| \geq v(n)$, there exists $k \in \{0, 1, \ldots, l-1\}$ such that:

$$|\epsilon_k - \epsilon_{k+1}| \geq \frac{v(n)}{l}$$

PRG Extension: Proof (Part 2)

Constructing Adversary $\mathcal{B}$

We use $\mathcal{A}$ to construct $\mathcal{B}$ that breaks the base PRG G.
$\mathcal{B}$ on input $T \in \{0,1\}^{n+1}$ (either from U_{n+1} or $G(U_n)$):

1. Sample $\sigma_1, \sigma_2, \ldots, \sigma_k \xleftarrow{\$} \{0,1\}$
2. Parse T as (σ_{k+1}, S_{k+1})
3. For $j = k+2$ to l: $(\sigma_j, S_j) := G(S_{j-1})$
4. Output: $\mathcal{A}(1^n, \sigma_1\sigma_2 \cdots \sigma_l S_l)$

PRG Extension: Proof (Part 2)

Constructing Adversary $\mathcal{B}$

We use $\mathcal{A}$ to construct $\mathcal{B}$ that breaks the base PRG G.
$\mathcal{B}$ on input $T \in \{0,1\}^{n+1}$ (either from U_{n+1} or $G(U_n)$):

1. Sample $\sigma_1, \sigma_2, \ldots, \sigma_k \xleftarrow{\$} \{0,1\}$
2. Parse T as (σ_{k+1}, S_{k+1})
3. For $j = k+2$ to l: $(\sigma_j, S_j) := G(S_{j-1})$
4. Output: $\mathcal{A}(1^n, \sigma_1\sigma_2 \cdots \sigma_l S_l)$

Analysis

- If $T \leftarrow G(U_n)$: output is from H_k
- If $T \xleftarrow{\$} U_{n+1}$: output is from H_{k+1}
- $\mathcal{B}$ runs in polynomial time

PRG Extension: Proof (Part 3)

Success Probability

$$
\begin{aligned}
&\left| \Pr[\mathcal{B}(1^n, G(U_n)) = 1] - \Pr[\mathcal{B}(1^n, U_{n+1}) = 1] \right| \\
&= \left| \Pr[\mathcal{A}(1^n, H_k) = 1] - \Pr[\mathcal{A}(1^n, H_{k+1}) = 1] \right| \\
&= |\epsilon_k - \epsilon_{k+1}| \\
&\geq \frac{v(n)}{l}
\end{aligned}
$$

PRG Extension: Proof (Part 3)

Success Probability

$$
\begin{aligned}
&\left| \Pr[\mathcal{B}(1^n, G(U_n)) = 1] - \Pr[\mathcal{B}(1^n, U_{n+1}) = 1] \right| \\
&= \left| \Pr[\mathcal{A}(1^n, H_k) = 1] - \Pr[\mathcal{A}(1^n, H_{k+1}) = 1] \right| \\
&= |\epsilon_k - \epsilon_{k+1}| \\
&\geq \frac{v(n)}{l}
\end{aligned}
$$

Contradiction

- $\frac{v(n)}{l}$ is non-negligible (since l is polynomial)
- This means $\mathcal{B}$ breaks the base PRG G
- Contradiction! Therefore, F must be a PRG.

PRG from OWP: Goal

Question

Can we construct a PRG assuming only that one-way permutations exist?

PRG from OWP: Goal

Question

Can we construct a PRG assuming only that one-way permutations exist?

Answer: Yes!

We'll construct a PRG $G : \{0,1\}^{2n} \to \{0,1\}^{2n+1}$ from a one-way permutation $f : \{0,1\}^n \to \{0,1\}^n$.

PRG from OWP: Goal

Question

Can we construct a PRG assuming only that one-way permutations exist?

Answer: Yes!

We'll construct a PRG $G : \{0,1\}^{2n} \to \{0,1\}^{2n+1}$ from a one-way permutation $f : \{0,1\}^n \to \{0,1\}^n$.

Key Ingredient

We need a **hard concentrated bit** $B(x, r)$ for the function $g(x, r) = f(x) \| r$.

PRG from OWP: Construction

Construction: PRG from OWP

Let $f : \{0,1\}^n \to \{0,1\}^n$ be a one-way permutation. Construct $G : \{0,1\}^{2n} \to \{0,1\}^{2n+1}$ as:

$$G(x, r) = f(x) \| r \| B(x, r)$$

where $x, r \in \{0,1\}^n$, and $B(x, r)$ is a hard concentrated bit for $g(x, r) = f(x) \| r$.

PRG from OWP: Construction

Construction: PRG from OWP

Let $f : \{0,1\}^n \to \{0,1\}^n$ be a one-way permutation. Construct $G : \{0,1\}^{2n} \to \{0,1\}^{2n+1}$ as:

$$G(x, r) = f(x) \| r \| B(x, r)$$

where $x, r \in \{0,1\}^n$, and $B(x, r)$ is a hard concentrated bit for $g(x, r) = f(x) \| r$.

Hard Concentrated Bit

Recall from Chapter 2: For OWP f, we can use:

$$B(x, r) = \left(\sum_{i=1}^{n} x_i r_i \right) \bmod 2$$

This is the inner product modulo 2, which is a hard concentrated bit.

PRG from OWP: Why This Works

Intuition

- $f(x)$: One-way permutation output (looks random)
- r: Truly random string
- $B(x, r)$: Hard bit that looks random given $f(x)$ and r
- Together: $(f(x), r, B(x, r))$ looks like $2n + 1$ random bits

PRG from OWP: Why This Works

Intuition

- $f(x)$: One-way permutation output (looks random)
- r: Truly random string
- $B(x, r)$: Hard bit that looks random given $f(x)$ and r
- Together: $(f(x), r, B(x, r))$ looks like $2n + 1$ random bits

Expansion

- Input: $2n$ bits (seed: x and r)
- Output: $2n + 1$ bits
- Expansion: 1 bit
- Can use PRG extension to get more bits!

PRG from OWP: Proof Setup

Theorem 3

The G constructed above is a PRG.

PRG from OWP: Proof Setup

Theorem 3

The G constructed above is a PRG.

Proof Strategy

Assume for contradiction that G is not a PRG. Then we can construct an adversary that breaks the hard concentrated bit.

PRG from OWP: Proof Setup

Theorem 3

The G constructed above is a PRG.

Proof Strategy

Assume for contradiction that G is not a PRG. Then we can construct an adversary that breaks the hard concentrated bit.

Define Hybrids

$$H_0 := G(U_{2n}) = f(x) \| r \| B(x, r), \text{ where } x, r \xleftarrow{\$} \{0,1\}^n$$

$$H_1 := f(x) \| r \| \sigma, \text{ where } x, r \xleftarrow{\$} \{0,1\}^n, \sigma \xleftarrow{\$} \{0,1\}$$

$$H_2 := U_{2n+1}$$

PRG from OWP: Proof (Part 1)

Key Observations

- Since G is not PRG, adversary $\mathcal{A}$ can distinguish H_0 from H_2
- Since f is a permutation, H_1 is uniformly distributed
- Therefore, $H_1 \equiv H_2$
- So $\mathcal{A}$ can distinguish H_0 from H_1

PRG from OWP: Proof (Part 1)

Key Observations

- Since G is not PRG, adversary $\mathcal{A}$ can distinguish H_0 from H_2
- Since f is a permutation, H_1 is uniformly distributed
- Therefore, $H_1 \equiv H_2$
- So $\mathcal{A}$ can distinguish H_0 from H_1

Distinguishing Advantage

There exists non-negligible $v(n)$ such that:

$$\left| \Pr[\mathcal{A}(H_0) = 1] - \Pr[\mathcal{A}(H_1) = 1] \right| \geq v(n)$$

PRG from OWP: Proof (Part 2)

Define H_1'

Let $H_1' = f(x) \| r \| (1 - B(x, r))$, where $x, r \xleftarrow{\$} \{0,1\}^n$.

PRG from OWP: Proof (Part 2)

Define H_1'

Let $H_1' = f(x) \| r \| (1 - B(x, r))$, where $x, r \xleftarrow{\$} \{0,1\}^n$.

Key Relationship

Since H_1 uses a random bit σ:

$$\begin{aligned}\Pr[\mathcal{A}(H_1) = 1] &= \Pr[\sigma = B(x,r)] \Pr[\mathcal{A}(H_0) = 1] \\ &\quad + \Pr[\sigma = 1 - B(x,r)] \Pr[\mathcal{A}(H_1') = 1] \\ &= \frac{1}{2} \Pr[\mathcal{A}(H_0) = 1] + \frac{1}{2} \Pr[\mathcal{A}(H_1') = 1]\end{aligned}$$

PRG from OWP: Proof (Part 2)

Define H_1'

Let $H_1' = f(x) \| r \| (1 - B(x, r))$, where $x, r \xleftarrow{\$} \{0,1\}^n$.

Key Relationship

Since H_1 uses a random bit σ:

$$\begin{aligned}\Pr[\mathcal{A}(H_1) = 1] &= \Pr[\sigma = B(x,r)] \Pr[\mathcal{A}(H_0) = 1] \\ &\quad + \Pr[\sigma = 1 - B(x,r)] \Pr[\mathcal{A}(H_1') = 1] \\ &= \frac{1}{2} \Pr[\mathcal{A}(H_0) = 1] + \frac{1}{2} \Pr[\mathcal{A}(H_1') = 1]\end{aligned}$$

Rearranging

$$\Pr[\mathcal{A}(H_1) = 1] - \Pr[\mathcal{A}(H_0) = 1]$$

$$\frac{1}{2}$$

PRG from OWP: Proof (Part 3)

Distinguishing H_0 and H_1'

From the previous slide:

$$\frac{1}{2}\left|\Pr[\mathcal{A}(H_0)=1]-\Pr[\mathcal{A}(H_1')=1]\right| = \left|\Pr[\mathcal{A}(H_1)=1]-\Pr[\mathcal{A}(H_0)=1]\right|$$
$$\geq v(n)$$

PRG from OWP: Proof (Part 3)

Distinguishing H_0 and H_1'

From the previous slide:

$$\frac{1}{2}\left|\Pr[\mathcal{A}(H_0)=1]-\Pr[\mathcal{A}(H_1')=1]\right| = \left|\Pr[\mathcal{A}(H_1)=1]-\Pr[\mathcal{A}(H_0)=1]\right| \geq v(n)$$

Therefore

$$\left|\Pr[\mathcal{A}(H_0)=1]-\Pr[\mathcal{A}(H_1')=1]\right| \geq 2v(n)$$

PRG from OWP: Proof (Part 3)

Distinguishing H_0 and H_1'

From the previous slide:

$$\frac{1}{2}\left|\Pr[\mathcal{A}(H_0)=1]-\Pr[\mathcal{A}(H_1')=1]\right| = \left|\Pr[\mathcal{A}(H_1)=1]-\Pr[\mathcal{A}(H_0)=1]\right| \geq v(n)$$

Therefore

$$\left|\Pr[\mathcal{A}(H_0)=1]-\Pr[\mathcal{A}(H_1')=1]\right| \geq 2v(n)$$

Without Loss of Generality

Assume:

$$\Pr[\mathcal{A}(H_0)=1]-\Pr[\mathcal{A}(H_1')=1] \geq 2v(n)$$

PRG from OWP: Constructing Adversary $\mathcal{B}$

Goal

Construct $\mathcal{B}$ that breaks the hard concentrated bit $B(x, r)$.

PRG from OWP: Constructing Adversary $\mathcal{B}$

Goal

Construct $\mathcal{B}$ that breaks the hard concentrated bit $B(x, r)$.

Construction

$\mathcal{B}$ on input $f(x)\|r$:

1. Sample $\sigma \xleftarrow{\$} \{0,1\}$ uniformly
2. If $\mathcal{A}(f(x)\|r\|\sigma) = 1$: output σ
3. Else: output $1 - \sigma$

PRG from OWP: Constructing Adversary $\mathcal{B}$

Goal

Construct $\mathcal{B}$ that breaks the hard concentrated bit $B(x, r)$.

Construction

$\mathcal{B}$ on input $f(x)\|r$:

1. Sample $\sigma \xleftarrow{\$} \{0,1\}$ uniformly
2. If $\mathcal{A}(f(x)\|r\|\sigma) = 1$: output σ
3. Else: output $1 - \sigma$

Intuition

- $\mathcal{B}$ guesses the hard bit by seeing which value makes $\mathcal{A}$ output 1
- If $\mathcal{A}$ distinguishes well, $\mathcal{B}$ guesses correctly with advantage

PRG from OWP: Proof (Part 4)

Success Probability of $\mathcal{B}$

$$
\begin{aligned}
&\Pr[\mathcal{B}(f(x)\|r) = B(x,r)] \\
&= \Pr[\sigma = B(x,r)] \Pr[\mathcal{A}(f(x)\|r\|\sigma) = 1 \mid \sigma = B(x,r)] \\
&\quad + \Pr[\sigma = 1 - B(x,r)] \Pr[\mathcal{A}(f(x)\|r\|\sigma) = 0 \mid \sigma = 1 - B(x,r)]
\end{aligned}
$$

PRG from OWP: Proof (Part 4)

Success Probability of $\mathcal{B}$

$$\begin{aligned}
&\Pr[\mathcal{B}(f(x)\|r) = B(x,r)] \\
&= \Pr[\sigma = B(x,r)] \Pr[\mathcal{A}(f(x)\|r\|\sigma) = 1 \mid \sigma = B(x,r)] \\
&\quad + \Pr[\sigma = 1 - B(x,r)] \Pr[\mathcal{A}(f(x)\|r\|\sigma) = 0 \mid \sigma = 1 - B(x,r)]
\end{aligned}$$

Continuing...

$$\begin{aligned}
&= \frac{1}{2} \Pr[\mathcal{A}(f(x)\|r\|B(x,r)) = 1] \\
&\quad + \frac{1}{2}(1 - \Pr[\mathcal{A}(f(x)\|r\|(1 - B(x,r))) = 1]) \\
&= \frac{1}{2} + \frac{1}{2}(\Pr[\mathcal{A}(H_0) = 1] - \Pr[\mathcal{A}(H_1') = 1]) \\
&> \frac{1}{2} + \nu(n)
\end{aligned}$$

PRG from OWP: Contradiction

Conclusion

- $\mathcal{B}$ predicts $B(x, r)$ with probability $\geq \frac{1}{2} + v(n)$
- Since $v(n)$ is non-negligible, this contradicts that B is a hard concentrated bit
- Therefore, our assumption that G is not a PRG must be false
- G **is a PRG!**

PRG from OWP: Contradiction

Conclusion

- $\mathcal{B}$ predicts $B(x, r)$ with probability $\geq \frac{1}{2} + v(n)$
- Since $v(n)$ is non-negligible, this contradicts that B is a hard concentrated bit
- Therefore, our assumption that G is not a PRG must be false
- G **is a PRG!**

Key Insight

- We reduced PRG security to hard concentrated bit security
- Since hard concentrated bits exist for OWPs (from Chapter 2), PRGs exist!
- Combined with PRG extension, we can generate polynomially many pseudorandom bits

Key Takeaways

1. **PRG Definition**: Efficient function that expands short seed into longer output that's computationally indistinguishable from random
2. **PRG Extension**: Any PRG that outputs one extra bit can be extended to output polynomially many bits
3. **PRG from OWP**: Can construct PRG from one-way permutations using hard concentrated bits
4. **Hybrid Arguments**: Powerful proof technique for showing computational indistinguishability

Next Steps

- PRGs are fundamental building blocks
- Next: Building more powerful primitives
  - Pseudorandom functions (PRFs)
  - Symmetric encryption schemes
  - Message authentication codes
- Questions?