Last Time

- Hard Core Predicates
- Computational Indistinguishability
- Prediction Advantage
Today

- Pseudorandom Distributions & Next-bit Unpredictability
- Completeness of Next-bit Test for Pseudorandomness
- Pseudorandom Generators
  - 1-bit stretch
  - Polynomial stretch
- Pseudorandom functions
Uniform distribution over \( \{0, 1\}^{\ell(n)} \) is denoted by \( U_{\ell(n)} \).

**Intuition:** A distribution is pseudorandom if it looks like a uniform distribution to any efficient test.

**Definition (Pseudorandom Ensembles)**

An ensemble \( \{X_n\} \), where \( X_n \) is a distribution over \( \{0, 1\}^{\ell(n)} \), is said to be pseudorandom if:

\[
\{X_n\} \approx \{U_{\ell(n)}\}
\]

**Looking ahead:** A PRG’s output should be pseudorandom.
Next-Bit Test

- Here is another interesting way to talk about pseudorandomness

- A pseudorandom string should pass all efficient tests that a (truly) random string would pass

- **Next Bit Test**: for a truly random sequence of bits, it is not possible to predict the “next bit” in the sequence with probability better than $1/2$ even given all previous bits of the sequence so far

- A sequence of bits *passes the next bit test* if no efficient adversary can predict “the next bit” in the sequence with probability better than $1/2$ even given all previous bits of the sequence so far
Next-bit Unpredictability

**Definition (Next-bit Unpredictability)**

An ensemble of distributions \( \{X_n\} \) over \( \{0, 1\}^{\ell(n)} \) is next-bit unpredictable if, for all \( 0 \leq i < \ell(n) \) and n.u. PPT \( A \), \( \exists \) negligible function \( \nu(\cdot) \) s.t.:

\[
\Pr[t = t_1 \ldots t_{\ell(n)} \sim X_n : A(t_1 \ldots t_i) = t_{i+1}] \leq \frac{1}{2} + \nu(n)
\]

**Theorem (Completeness of Next-bit Test)**

If \( \{X_n\} \) is next-bit unpredictable then \( \{X_n\} \) is pseudorandom.
Next-bit Unpredictability $\implies$ Pseudorandomness

\[ H_n^{(i)} := \{ x \sim X_n, u \sim U_n : x_1 \ldots x_i u_{i+1} \ldots u_{\ell(n)} \} \]

- First Hybrid: $H_n^0$ is the uniform distribution $U_{\ell(n)}$
- Last Hybrid: $H_n^{(\ell(n))}$ is the distribution $X_n$
- Suppose $H_n^{(\ell(n))}$ is next-bit unpredictable but not pseudorandom
  \[ H_n^{(0)} \not\approx H_n^{(\ell(n))} \implies \exists i \in [\ell(n) - 1] \text{ s.t. } H_n^{(i)} \not\approx H_n^{(i+1)} \]
- Now, next bit unpredictability is violated
- **Exercise:** Do the full formal proof
Definition (Pseudorandom Generator)

A deterministic algorithm $G$ is called a pseudorandom generator (PRG) if:

- $G$ can be computed in polynomial time
- $|G(x)| > |x|$
- $\{x \leftarrow \{0, 1\}^n : G(x)\} \approx_c \{U_\ell(n)\}$ where $\ell(n) = |G(0^n)|$

The stretch of $G$ is defined as: $|G(x)| - |x|$

- Can we construct PRG with even 1-bit stretch?
- What about many bits? Can we generically stretch?
PRG with 1-bit stretch

- Remember the hardcore predicate?
- It is hard to guess \( h(s) \) even given \( f(s) \)
- Let \( G(s) = f(s) \parallel h(s) \) where \( f \) is a OWF
- Some small issues:
  - \( |f(s)| \) might be less than \( |s| \)
  - \( f(s) \) may always start with prefix 101 (not random)
- **Solution:** let \( f \) be a one-way permutation (OWP) over \( \{0, 1\}^n \)
  - Domain and Range are of same size, i.e., \( |f(s)| = |s| = n \)
  - \( f(s) \) is uniformly random over \( \{0, 1\}^n \) if \( s \) is
    \[ \forall y : \Pr[f(s) = y] = \Pr[s = f^{-1}(y)] = 2^{-n} \]
    \[ \Rightarrow f(s) \) is uniform and cannot start with a fix value! \]
PRG with 1-bit stretch

- Let $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$ be a OWP
- Let $h : \{0, 1\}^* \rightarrow \{0, 1\}$ be a hardcore predicate for $f$
- **Construction:** $G(s) = f(s) \parallel h(s)$

**Theorem (PRG based on OWP)**

$G$ is a pseudorandom generator with 1-bit stretch.

- **Think:** Proof?
- **Proof Idea:** Use next-bit unpredictability. Since first $n$ bits of the output are uniformly distributed (since $f$ is a permutation), any adversary for next-bit unpredictability with non-negligible advantage $\frac{1}{p(n)}$ must be predicting the $(n + 1)$th bit with advantage $\frac{1}{p(n)}$. Build an adversary for hard-core predicate to get a contradiction.
One-bit stretch PRG $\implies$ Poly-stretch PRG

Intuition: Iterate the one-bit stretch PRG poly times

Construction of $G_{poly} : \{0, 1\}^n \to \{0, 1\}^{\ell(n)}$:

- Let $G : \{0, 1\}^n \to \{0, 1\}^{n+1}$ be a one-bit stretch PRG

\[
\begin{align*}
  s & = X_0 \\
  G(X_0) & = X_1 \parallel b_1 \\
  & \vdots \\
  G(X_{\ell(n)-1}) & = X_{\ell(n)} \parallel b_{\ell(n)} \\

g_{poly}(s) & := b_1 \ldots b_{\ell(n)}
\end{align*}
\]

Think: Proof?
Proof that $G_{poly}$ is pseudorandom

- Want: $\left\{ s \leftarrow \{0,1\}^n : G_{poly}(s) \right\} \approx_c \left\{ U_{\ell(n)} \right\}$
- Let $D$ be any non-uniform PPT algorithm.

<table>
<thead>
<tr>
<th>Experiment $H_0$</th>
<th>$s = X_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G(X_0) = X_1 | b_1$</td>
<td></td>
</tr>
<tr>
<td>$G(X_1) = X_2 | b_2$</td>
<td></td>
</tr>
<tr>
<td>$\vdots$</td>
<td></td>
</tr>
<tr>
<td>$G(X_{\ell-1}) = X_\ell | b_\ell$</td>
<td></td>
</tr>
</tbody>
</table>

Step 0:

Output $D(b_1 b_2 \ldots b_\ell)$

Claim: $\left| \Pr_s[D(G'(s)) = 1] - \Pr_s[H_0 = 1] \right| = 0.$

Proof: Input of $D$ is identically distributed in both cases. □
Proof that $G_{poly}$ is pseudorandom

**Step 1:** modify $H_0$ one line at a time.

<table>
<thead>
<tr>
<th>Experiment $H_0$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$s$ = $X_0$</td>
<td></td>
</tr>
<tr>
<td>$G(X_0)$ = $X_1|b_1$</td>
<td></td>
</tr>
<tr>
<td>$G(X_1)$ = $X_2|b_2$</td>
<td></td>
</tr>
<tr>
<td>$\vdots$</td>
<td></td>
</tr>
<tr>
<td>$G(X_{\ell-1})$ = $X_\ell|b_\ell$</td>
<td></td>
</tr>
</tbody>
</table>

Output $D(b_1b_2\ldots b_\ell)$.
Proof that $G_{poly}$ is pseudorandom

**Step 1:** modify $H_0$ one line at a time.

<table>
<thead>
<tr>
<th>Experiment $H_0$</th>
<th>Experiment $H_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s = X_0$</td>
<td>$s = X_0$</td>
</tr>
<tr>
<td>$G(X_0) = X_1</td>
<td></td>
</tr>
<tr>
<td>$G(X_1) = X_2</td>
<td></td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$G(X_{\ell-1}) = X_{\ell}</td>
<td></td>
</tr>
</tbody>
</table>

Output $D(b_1 b_2 \ldots b_{\ell})$. Output $D(u_1 b_2 \ldots b_{\ell})$.

**Claim:** $|\Pr_s[H_0 = 1] - \Pr_{s,s_1,u_1}[H_1 = 1]| \leq \mu(n)$

- Can similarly define $H_2, \ldots, H_{\ell-1}$ s.t. in $H_{\ell-1}$, $b_1 b_2 \ldots b_{\ell}$ is sampled from $U_{\ell}$
- To prove that $G_{poly}$ is PRG, it suffices to show that $H_0 \approx_c H_{\ell}$
Step 2: Hybrid Lemma

- For contradiction, suppose that $G_{poly}$ is not a PRG, i.e., $H_0$ and $H_\ell$ are distinguishable with non-negligible probability $\frac{1}{p(n)}$.
- By Hybrid Lemma, there exists $i$ s.t. $H_i$ and $H_{i+1}$ are distinguishable with probability $\frac{1}{p(n)\ell(n)}$.
- Idea: Contradict the security of $G$. 

Proof that $G_{poly}$ is pseudorandom (contd.)
Step 3: Breaking security of $G$

- For simplicity, suppose that $i = 0$ (proof works for any $i$)
- Construct $D$ to break the pseudorandomness of $G$ as follows
  - $D$ gets as input $Z\|r$ sampled either as $X_1\|b_1$ or as $s_1\|u_1$
  - Compute $X_2\|b_2 = G(Z)$ and continue as the rest of the experiment(s)
  - Output $D(rb_2 \ldots b_\ell)$

- If $Z\|r$ is pseudorandom, i.e., sampled as $X_1\|b_1 = G(s)$, then output of $D$ is distributed identically to the output of $H_0$
- Otherwise, i.e., $Z\|r$ is (truly) random, and therefore output of $D$ is distributed identically to the output of $H_1$
- Hence: $D$ distinguishes the output of $G$ with advantage $\frac{1}{p(n)\ell(n)}$ and runs in polynomial time. This is a contradiction $\square$
Concluding Remarks on PRG

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