Lecture 14: Pseudo-random Generators
Pseudorandom Generator: PRG

Definition (PRG)

Let $G : \{0, 1\}^n \rightarrow \{0, 1\}^{n+\ell}$ be a function that is efficient to evaluate. We say that $G$ is a pseudorandom generator, if

1. The stretch $\ell > 0$, and
2. The distribution $G(\mathbb{U}_n^{\{0,1\}})$ “appears indistinguishable” from the distribution $\mathbb{U}_n^{\{0,1\}^{n+\ell}}$ for computationally bounded adversaries.

Clarifications.

1. The input bits $s \sim \mathbb{U}_n^{\{0,1\}}$ that is fed to the PRG is referred to as the seed of the PRG
2. Intuition of a PRG: We rely on a small amount of pure randomness to jumpstart a PRG that yields more (appears to be) random bits
Note that if \( \ell \leq 0 \) then PRG is easy to construct. Note that in this case \( n + \ell \leq n \). So, \( G(s) \) just outputs the first \( n + \ell \) bits of the input seed \( s \).

The entire non-triviality is to construct \( G \) when \( \ell \geq 1 \).

Suppose \( \ell = 1 \). Note that in the case \( G \) has \( 2^n \) different possible inputs. So, \( G \) has at most \( 2^n \) different possible outputs. The range \( \{0, 1\}^{n+\ell} \) has size \( 2^{n+1} \). So, there are at least \( 2^{n+1} - 2^n = 2^n \) elements in the range that have no pre-image under the mapping \( G \). We can conclude that \( G(U_{\{0,1\}^n}) \) assigns 0 probability to at least \( 2^n \) entries in the range.

Note that the distribution \( G(U_{\{0,1\}^n}) \) is different from the distribution \( U_{\{0,1\}^{n+1}} \). A computationally unbounded adversary can distinguish \( G(U_{\{0,1\}^n}) \) from \( U_{\{0,1\}^{n+1}} \). However, for a computationally bounded adversary, the distribution \( G(U_{\{0,1\}^n}) \) appears same as the distribution \( U_{\{0,1\}^{n+1}} \).
In this class, we shall see a construction of PRG when $\ell = 1$ given a OWP $f$. In general, we know how to construct a PRG using a OWF. However, presenting that construction is beyond the scope of this course.

Note that these PRG constructions work for any OWF $f$. So, if some OWF $f$ is broken in the future due to progress in mathematics or use of quantum computers, then we can simply replace the existing PRG constructions to use a different OWF $g$. 
Let $f : \{0, 1\}^n \to \{0, 1\}^n$ be a bijection.

Suppose we sample $x \leftarrow \{0, 1\}^n$.

For any $y \in \{0, 1\}^n$, what is the probability that $f(x) = y$?

- Note that there is a unique $x'$ such that $f(x') = y$, because $f$ is a bijection.
- $f(x) = y$ if and only if $x = x'$, i.e. the probability that $f(x) = y$ is $1/2^n$.

So, the distribution of $f(x)$, where $x \leftarrow \{0, 1\}^n$, is a uniform distribution over $\{0, 1\}^n$. 

PRG Construction
We define the inner product of \( r \in \{0, 1\}^n \) and \( x \in \{0, 1\}^n \) as
\[
\langle r, x \rangle = r_1x_1 \oplus r_2x_2 \oplus \cdots \oplus r_nx_n
\]
We will state the Goldreich-Levin Hardcore Predicate without proof.

**Theorem (Goldreich-Levin Hardcore Predicate)**

If \( f \{0, 1\}^n \rightarrow \{0, 1\}^n \) is a one-way function then the bit \( b = \langle r, x \rangle \) cannot be predicted given \((r, f(x))\). This proof is beyond the scope of this course. However, students are encouraged to study this celebrated result in the future.
A note on “Predicting a bit”

- Note that it is trivial to correctly predict any bit with probability $1/2$. (Guess a uniformly random bit $z$. The probability that $z$ is identical to the hidden bit is $1/2$)

- To non-trivially predict a hidden bit, the adversary has to correctly predict it with probability at least $1/2 + \varepsilon$, where $\varepsilon = 1/\text{poly}(n)$
Recall: A pseudorandom generator (PRG) is a function $G_{n, n+\ell}: \{0, 1\}^n \rightarrow \{0, 1\}^{n+\ell}$ such that, for $x \leftarrow \{0, 1\}^n$, the output $G_{n, n+\ell}(x)$ looks like a random $(n + \ell)$-bit string.

A one-bit extension PRG has $\ell = 1$

Suppose $f: \{0, 1\}^n \rightarrow \{0, 1\}^n$ is a OWP (i.e., $f$ is a OWF and it is a bijection)

Note that the mapping $(r, x) \mapsto (r, f(x))$ is a bijection

So, the output $(r, f(x))$ is a uniform distribution if $(r, x) \leftarrow \{0, 1\}^{2n}$

Now, the output $(r, f(x), \langle r, x \rangle)$ looks like a random $(2n + 1)$-bit string if $f$ is a OWP (because of Goldreich-Levin Hardcore Predicate result)
Consider the function $G_{2n,2n+1} : \{0, 1\}^{2n} \rightarrow \{0, 1\}^{2n+1}$ defined as follows:

$$G_{2n,2n+1}(r, x) = (r, f(x), \langle r, x \rangle)$$

This is a one-bit extension PRG if $f$ is a OWP.

This construction will be pictorially represented as follows:

![Diagram](image-url)
In the previous step, we saw how to construct a one-bit extension PRG $G$.

Now, we use the previous step iteratively to construct arbitrarily long pseudorandom bit-strings.

The next slide, using the one-bit extension PRG, provides the intuition to construct $G_{2n,\ell}: \{0, 1\}^{2n} \rightarrow \{0, 1\}^{n+\ell}$, for arbitrary $\ell = \text{poly}(n)$.

The example shows only $n + \ell = 5$ but can be extended naturally to arbitrary $\ell = \text{poly}(n)$. 
Generating Long Pseudorandom Bit-Strings II

PRG Construction
This is a PRG that takes $n$-bit seed and outputs $2n$-bit string

$G_{n,2n}$ is a length-doubling PRG if $G_{n,2n} : \{0, 1\}^n \rightarrow \{0, 1\}^{2n}$ and $G_{n,2n}$ is a PRG

We can use the iterated construction in the previous slide to construct a length-doubling PRG from one-bit extension PRG
Food for thought

- Design secret-key encryption schemes where the message is much longer than the secret key