Lecture 4: PRG *Construction*

Notes by Yael Kalai

MIT - 6.5620
Lecture 4 (September 15, 2025)

> ***Warning:*** This document is a rough draft, so it may contain bugs. Please feel free to email me with corrections.

Recap

Last lecture we covered the following:

1. We defined the notion of a pseudorandom generator.

   Definition 1. An efficient (poly-time computable) deterministic function $G : \{0,1\}^\lambda \to \{0,1\}^{n(\lambda)}$ is said to be a pseudorandom generator if the following two conditions hold:

   - It is expanding; i.e., $n(\lambda) > \lambda$.
   - It is pseudorandom, i.e.
   $$\{G(U_\lambda)\}_{\lambda\in\mathbb{N}} \approx \{U_{n(\lambda)}\}_{\lambda\in\mathbb{N}},$$
   where U_ℓ is the uniform distribution over $\{0,1\}^\ell$.

2. We proved that pseudorandomenss is equivalent to the following "next-bit unpredictability" property.

   Definition 2. An expanding function $G : \{0,1\}^\lambda \to \{0,1\}^{n(\lambda)}$ is next-bit unpredictable if for every poly-size adversary $\mathcal{A}$ there exists a negligible function μ such that for every $\lambda \in \mathbb{N}$ and every $i \in [n(\lambda)]$
   $$\Pr_{U\leftarrow\{0,1\}^\lambda}[\mathcal{A}(G(U_\lambda)_{[i-1]}) = G(U_\lambda)_i] = 1/2 + \mu(\lambda)$$
   where $G(U_\lambda)_{[i-1]}$ denotes the first $i-1$ bits of $G(U_\lambda)$ and $G(U_\lambda)_i$ denotes the i'th bit of $G(U_\lambda)$.

3. We proved that if there exists a PRG $G : \{0,1\}^\lambda \to \{0,1\}^{n(\lambda)}$ then there exists a computationally secure encryption scheme with key space $\mathcal{K}_\lambda = \{0,1\}^\lambda$ and message space $\mathcal{M}_\lambda = \{0,1\}^{n(\lambda)}$, defined by
   $$\mathsf{Enc}(k,m) = G(k) \oplus m$$
   and
   $$\mathsf{Dec}(k,c) = G(k) \oplus c.$$

Today:

In this lecture (and next), our goal is to construct a PRG

$$G : \{0,1\}^\lambda \to \{0,1\}^{n(\lambda)}$$

for an arbitrary polynomial expanding function $n : \mathbb{N} \to \mathbb{N}$.

We will focus on constructing a PRG G that expands by a single bit; i.e,

$$G : \{0,1\}^\lambda \to \{0,1\}^{\lambda+1}$$

This is enough since we can generically increase the stretch of a PRG.

Stretching a PRG

One can increase the stretch of a PRG by applying a PRG to the output of the PRG. Namely, given a PRG

$$G : \{0,1\}^\lambda \to \{0,1\}^{\lambda+1}$$

one can construct a PRG

$$G^k : \{0,1\}^\lambda \to \{0,1\}^{\lambda+k}$$

where

$$G^k(U_\lambda) = G(G(\dots G(U_\lambda)\dots)).$$

Theorem 3. *[2] If G is a* PRG *then for every polynomial $k = k(\lambda)$, G^k is a* PRG*.*

Proof. Fix a polynomial $k = k(\lambda) \geq 1$. We need to prove that G^k is a PRG. The fact that G^k is stretching and efficiently computable follows from the fact that G satisfies these properties, together with the fact that $k(\lambda) \leq \text{poly}(\lambda)$.

It is tempting to prove that it is pseudorandom by induction on k, as follows:

Base case: $k = 1$. By assumption

Induction step: Suppose $G^{k-1} : \{0,1\}^\lambda \to \{0,1\}^{\lambda+k-1}$ is pseudorandom, and we will prove that $G^k : \{0,1\}^\lambda \to \{0,1\}^{\lambda+k}$ is pseudorandom. Fix a poly-size adversary $\mathcal{A}$. Then for every $\lambda \in \mathbb{N}$

$|\Pr[\mathcal{A}(G^k(U_\lambda)) = 1] - \Pr[\mathcal{A}(U_{\lambda+k}) = 1]| =$
$|\Pr[\mathcal{A}[G(G^{k-1}(U_\lambda)) = 1] - \Pr[\mathcal{A}(U_{\lambda+k}) = 1]| =$
$|\Pr[\mathcal{A}[G(G^{k-1}(U_\lambda)) = 1] - \Pr[\mathcal{A}[G(U_{\lambda+k-1}) = 1] + \Pr[\mathcal{A}[G(U_{\lambda+k-1}) = 1] - \Pr[\mathcal{A}(U_{\lambda+k}) = 1]| \leq$
$|\Pr[\mathcal{A}[G(G^{k-1}(U_\lambda)) = 1] - \Pr[\mathcal{A}[G(U_{\lambda+k-1}) = 1]| + |\Pr[\mathcal{A}[G(U_{\lambda+k-1}) = 1] - \Pr[\mathcal{A}(U_{\lambda+k}) = 1]| \triangleq$
$\mu_{k-1}(\lambda) + \mu_1(\lambda).$

Note that μ_{k-1} is a negligible function by the induction hypothesis, and μ_1 is a negligible function by assumption that G is a PRG, and the sum of negligible functions is negligible.

Why is this argument flawed? Induction only works for a constant k! The "correct" way to prove this is via a hybrid argument, as follows:

$$|\Pr[\mathcal{A}(G^k(U_\lambda)) = 1] - \Pr[\mathcal{A}(U_{\lambda+k}) = 1]| =$$

$$|\sum_{i=0}^{k-1} \Pr[\mathcal{A}(G^{k-i}(U_{\lambda+i})) = 1] - \Pr[\mathcal{A}(G^{k-(i-1)}U_{\lambda+i-1}) = 1]| \leq$$

$$\sum_{i=0}^{k-1} |\Pr[\mathcal{A}(G^{k-i}(U_{\lambda+i})) = 1] - \Pr[\mathcal{A}(G^{k-(i-1)}U_{\lambda+i-1}) = 1]| \leq$$

$$\sum_{i=0}^{k-1} \mu_i(\lambda) = \text{negl}(\lambda).$$

where the fact that each μ_i is negligible follows from the induction hypothesis.

□

The above theorem implies that it suffices to construct a PRG that has a single bit stretch. The following remarkable theorem is known.

Theorem 4. *[2] Pseudorandom generators exist assuming the existence of one-way functions.*

This theorem has a beautiful but very complicated proof. We will prove a simplified version that assumes the existence of one-way permutations, defined below.

Definition 5. A function $f : \{0,1\}^* \to \{0,1\}^*$ is a permutation if it is length preserving and bijective; namely, for every $\lambda \in \mathbb{N}$ and for every $x \in \{0,1\}^\lambda$ it holds that $f(x) \in \{0,1\}^\lambda$, and for every $y \in \{0,1\}^\lambda$ there is a unique $x \in \{0,1\}^\lambda$ such that $f(x) = y$.

Definition 6. A one-way permutation (OWP) $f : \{0,1\}^* \to \{0,1\}^*$ is a one-way function that is also a permutation.

We prove the following theorem.

Theorem 7. *Pseudorandom generators exist assuming the existence of a* OWP.

By Theorem 3, to prove Theorem 7 it suffices to construct a PRG that stretches by a single bit assuming the existence of a OWP.

PRG *Construction with a Single Bit Stretch*

Let $f : \{0,1\}^* \to \{0,1\}^*$ be a one-way permutation. Suppose that f has a *hardcore predicate* $P : \{0,1\}^* \to \{0,1\}$.

Definition 8. $P : \{0,1\}^* \to \{0,1\}$ is a *hardcore predicate* of $f : \{0,1\}^* \to \{0,1\}^*$ if it is efficiently computable and for every poly-size $\mathcal{A}$ there exists a negligible function $\mu : \mathbb{N} \to [0,1]$ such that for every $\lambda \in \mathbb{N}$,

$$\Pr[\mathcal{A}(f(U_\lambda)) = P(U_\lambda)] \leq \frac{1}{2} + \mu(\lambda).$$

Given a one-way permutation f with a hardcore predicate P, let

$$G(U_\lambda) = f(U_\lambda) \circ P(U_\lambda)$$

Theorem 9. *G is a pseudorandom generator.*

Proof. G is expanding by definition, and the fact that it is efficiently computable follows from the fact that both f and P are efficiently computable. Thus, it remains to prove that G is pseudorandom, or equivalently that G satisfies the next-bit unpredictability property.

To this end fix any poly-size adversary $\mathcal{A}$. We need to prove that there exists a negligible function μ such that for every $\lambda \in \mathbb{N}$ and every $i \in [\lambda + 1]$

$$\Pr[\mathcal{A}(G(U)_{[i-1]}) = G(U)_i] \leq \frac{1}{2} + \mu(\lambda)$$

The fact that f is a one-way permutation implies that for every $i \leq [\lambda]$

$$\Pr[\mathcal{A}(G(U)_{[i-1]}) = G(U)_i] = \frac{1}{2}.$$

For $i = \lambda + 1$, the fact that P is a hardcore predicate of f implies that there exists a negligible function μ such that

$$\Pr[\mathcal{A}(G(U)_{[\lambda]}) = G(U)_{\lambda+1}] = \Pr[\mathcal{A}(f(U_\lambda)) = P(U_\lambda)] \leq \frac{1}{2} + \mu(\lambda),$$

as desired. □

In theorem 7 we assumed there exists a one-way permutation but we did not assume that it has a hardcore predicate. Thankfully, Goldreich and Levin proved that every one-way function has a hardcore predicate!

Theorem 10. *[1] If f is a one-way function, then the following randomized predicate*

$$P(x, r) := x \cdot r \mod 2 = \sum_{i \in \lambda} x_i r_i \mod 2$$

is a hardcore predicate for f. Namely, for every poly-size $\mathcal{A}$ there exists a negligible function $\mu : \mathbb{N} \to [0,1]$ such that for every $\lambda \in \mathbb{N}$

$$\Pr_{U_\lambda \leftarrow \{0,1\}^\lambda}[\mathcal{A}(f(U_\lambda), r) = P(U_\lambda, r))] \leq \frac{1}{2} + \mu(\lambda)$$

References

[1] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC)*, pages 25–32, Seattle, Washington, USA, 1989.

[2] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.