1 Topic Covered

Creating a PRG in 3 steps:

- Creating a PRG with constant stretch from a PRG with stretch 1
- Creating a PRG with polynomial stretch from a PRG with stretch 1
- Creating a PRG with stretch 1 from a OWF

2 Increasing the stretch of a PRG

2.1 Previous Definitions

**Definition 1** We define computational indistinguishability \( X \approx Y \) between ensembles \( X = \{X_n\}_{n \in \mathbb{N}} \) and \( Y = \{Y_n\}_{n \in \mathbb{N}} \) as \( \forall \text{ PPT } D, \exists \varepsilon(n) = \text{negl}(n) \) such that

\[
|\Pr[D(X_n) = 1] - \Pr[D(Y_n) = 1]| \leq \varepsilon(n)
\]

\[\Diamond\]

**Definition 2** A function \( G : \{0,1\}^* \to \{0,1\}^* \) is a PRG with stretch \( \ell(n) \) if

\[
G(U_n) \approx U_{n+\ell(n)}
\]

where \( U_m \) denotes the uniform distribution over \( \{0,1\}^m \).

\[\Diamond\]

2.2 Increasing stretch from 1 to a constant

**Theorem 1** If \( \exists \text{ PRG } G \text{ with 1-bit stretch, then } \forall \ell(n) = \text{poly}(n), \exists \text{ PRG } G^\ell \text{ with } \ell(n)-bit stretch.**

**Proof:** (constant \( \ell \)) Using the following construction, we define \( G^\ell(x_0) = (b_1, b_2, \ldots, b_\ell, x_\ell) \)

Or in psuedocode:

\[
G^\ell(x_0) = \begin{cases} 
\text{for } i \in \{1, \ldots, \ell\} \\
(x_i, b_i) := G(x_{i-1}) \\
\text{output } (b_1, \ldots, b_\ell, x_\ell)
\end{cases}
\]
To prove this is a PRG, we need to show that if we could break $G^\ell$ then we could break $G$.

Recall:
**Hybrid argument:** If $X \approx Y$ and $Y \approx Z$, then $X \approx Z$.

We will define some hybrid, in-between distributions then show that every step of the chain is computationally indistinguishable from the next. We define:

\[
\begin{align*}
H^0_n &:= G^\ell(U_n) \\
H^i_n &:= b_1, \ldots, b_i \leftarrow \{0,1\}^n \\
&\quad \quad \quad \quad \quad \quad \quad (b_{i+1}, \ldots, b_\ell, x_\ell) := G^{\ell-i}(x_i) \\
H^\ell_n &:= U_{n+\ell}
\end{align*}
\]

We want to show that any two adjacent hybrids are indistinguishable. Here’s a representation of the difference between two:

**Claim 1** \( \forall i \in \{0, 1, \ldots, \ell - 1\}, H^i \approx H^{i+1} \)

**Idea:** If we can distinguish between hybrids, we can distinguish between \((x_{i+1}, b_{i+1}) = G(x_i)\) and \((x_{i+1}, b_{i+1})\) being uniformly random. This is the only difference between Hybrids \(H^i\) and \(H^{i+1}\).

**Proof:** We define a PPT function \(f_i\) as

\[
f_i(x_{i+1}, b_{i+1}) = \begin{cases} \\
& b_1, \ldots, b_i \leftarrow \{0,1\}^n \\
& \text{for } j \in \{i + 2, \ldots, \ell\} \\
& (x_j, b_j) := G(x_{j-1}) \\
& \text{output } (b_1, \ldots, b_\ell, x_\ell) \end{cases}
\]

We note that the distribution of \(f_i(U_{n+1})\) is related to \(H\), in particular

\[
\begin{align*}
f_i(U_{n+1}) &\equiv H^{i+1}_n \\
f_i(G(U_n)) &\equiv H^i_n
\end{align*}
\]
Where “≡” means equal distributions.
Last time we claimed that if $X \approx Y$ and $f$ is a PPT function then $f(X) \approx f(Y)$. By this claim and assumption of security of $G$, we know $H^i \approx H^{i+1}$. Now we know

$$H^0 \approx H^1 \approx \ldots \approx H^\ell$$

and by the hybrid argument

$$G^\ell(U_n) \equiv H^0 \equiv H^\ell \equiv U_{n+\ell}$$

Which proves $G^\ell$ is a PRG.

2.3 Increasing stretch from 1 to a polynomial

However, that proof only works for constant $\ell$. We now want to extend the proof to any polynomial $\ell(n)$. (Side note: we are only dealing with the cases where $\ell(n)$ is computible in polynomial time.) We use almost the exact same construction as last time, just changing $\ell$ to $\ell(n)$:

$$G^\ell(x) = \begin{cases} 
\text{for } i \in \{1, \ldots \ell(n)\} \\
\quad (x_i, b_i) := G(x_{i-1}) \\
\text{output } (b_1, \ldots b_{\ell(n)}, x_{\ell(n)})
\end{cases}$$

The analysis is almost the same, but now our hybrids look like:

$$\{H_i^n\}_{n \in \mathbb{N}, i \in \{0, \ldots \ell(n)-1\}}$$

**Claim 2** If for all polynomials $i(n)$ such that $i(n) \in \{0, \ldots \ell(n) - 1\}$ we have

$$\{H_n^{i(n)}\}_{n \in \mathbb{N}} \approx \{H_n^{i(n)+1}\}_{n \in \mathbb{N}}$$

then

$$\{H_0^n\}_{n \in \mathbb{N}} \approx \{H_\ell^n\}_{n \in \mathbb{N}}$$

We need this claim because while we could use the hybrid argument for a known number of ensembles, now the number of hybrid ensembles depends on $n$.

**Proof:** Let $D$ be a PPT distinguisher between $\{H_0^n\}_{n \in \mathbb{N}}$ and $\{H_\ell^n\}_{n \in \mathbb{N}}$.

$$\left| \Pr[D(H_0^n) = 1] - \Pr[D(H_\ell^n) = 1] \right| = \sum_{i=0}^{\ell(n)-1} \Pr[D(H_i^n) = 1] - \Pr[D(H_i^{i+1}) = 1] \leq \sum_{i=0}^{\ell(n)-1} \frac{\Pr[D(H_i^n) = 1] - \Pr[D(H_i^{i+1}) = 1]}{\delta_i} \leq \ell(n) \cdot \left| \Pr[D(H_n^{i^{(n)}}) = 1] - \Pr[D(H_n^{i^{(n)+1}}) = 1] \right|$$
Where $i^*(n) = \arg \max_{i \in \{0, \ldots, \ell(n) \}} \delta_i^i$.

Essentially, we are bounding every term in the sum by the worst case term. Since by assumption, $|\Pr[D(H_n^{i^*(n)}) = 1] - \Pr[D(H_n^{i^*(n)+1}) = 1]|$ is negligible, we can conclude that $\ell(n) \cdot \text{negl}(n)$ is also negligible.

To prove $\{H_n^{i(n)}\}_{n \in \mathbb{N}} \approx \{H_n^{i(n)+1}\}_{n \in \mathbb{N}}$ would be the same as proving $H_i \approx H_{i+1}$ in the fixed $\ell$ case (Claim 1), but there is an additional difficulty: $i(n)$ may not be efficiently computable.

There are at least two ways different ways we could deal with this:

1. Use the non-uniform model of computation, which equips a TM with some fixed lookup value of $n$. This can also be viewed as a family of algorithms indexed by $n$.

2. Instead of changing our model of computation, we can make a stronger claim by using a weaker assumption:

**Claim 3** Let $I_n$ be uniform over $\{0, \ldots, \ell(n-1)\}$. If $H_{I_n} \approx H_{I_n+1}$ then $H_n^0 \approx H_n^{\ell(n)}$

**Proof:** (Similar to Claim 2).

\[
|\Pr[D(H_n^0) = 1] - \Pr[D(H_n^{\ell(n)}) = 1]| \\
= \sum_{i=0}^{\ell(n)-1} |\Pr[D(H_n^i) = 1] - \Pr[D(H_n^{i+1}) = 1]| \\
= \sum_{i=0}^{\ell(n)-1} \left| \Pr[D(H_n^{I_n}) = 1 \mid I_n = i] - \Pr[D(H_n^{I_n+1}) = 1 \mid I_n = i] \right| \\
= \ell(n) \cdot \sum_{i=0}^{\ell(n)-1} \left| \Pr[D(H_n^i) = 1, I_n = i] - \Pr[D(H_n^{i+1}) = 1, I_n = i] \right| \\
= \ell(n) \cdot |\Pr[D(H_n^{I_n}) = 1] - \Pr[D(H_n^{I_n+1}) = 1]| \\
= \text{negl}(n)
\]

Now to finish the proof that $G^\ell$ is a PRG we need to show $H_{I_n} \approx H_{I_n+1}$

We change our definition of $f_i$ to $f_{I_n}$

\[
f_{I_n}(x, b) = \begin{cases}
\text{pick } i \leftarrow I_n \\
(x_{i+1}, b_{i+1}) := (x, b) \\
b_1, \ldots, b_{I_n} \leftarrow \{0, 1\}^n \\
\text{for } j \in \{I_n + 2, \ldots, \ell\} \\
(x_j, b_j) := G(x_{j-1}) \\
\text{output } (b_1, \ldots, b_\ell, x_\ell) 
\end{cases}
\]
The rest of the proof is identical to before. Using Claim 3, we know
\[ G_\ell^{(n)}(U_n) \equiv H_n^0 \approx H_n^{\ell(n)} \equiv U_{n+\ell(n)} \]
Which shows \( G_\ell \) is a PRG for any computible \( l(n) = \text{poly}(n) \).

3 Creating a PRG from a OWF

We’ve shown that PRG’s of larger stretch can be constructed from a PRG with 1-bit stretch. Now we need to construct such a PRG from a OWF. It’s slightly suprising that this can be done, since the requirement of uniformity doesn’t seem to be provided by a OWF.

**Definition 3** A OWF \( f : \{0,1\}^* \to \{0,1\}^* \) is a one way permutation (OWP) when both

- \(|f(x)| = |x| \quad \forall \ x \)
- \( \forall \ x \neq x', f(x) \neq f(x') \)

\( \Diamond \)

Note that this definition implies that \( f \) is one-to-one and onto.

**Idea:** We construct \( G = (f(x), hc(x)) \) for some \( hc : \{0,1\}^* \to \{0,1\} \).

We want to exploit the fact that there is some information in \( x \) that is unknown and hard to recover.

As a first attempt, would defining \( hc(x) = x[1] \) produce a good PRG? Unfortunately, this won’t work for arbitrary OWP \( f \). As a counterexample, let \( f' \) be a OWP, and \( f(x) = (x[1], f'(x[2 \ldots n])) \). We can show that \( f'(x) \) is a valid OWP, since a preimage of \( f' \) would result in a preimage of \( f \), but \( G(x) = (f(x), hc(x)) \) would always output equal first and last bits, so \( G \) could be easily distinguished from \( U_n \), and wouldn’t be a PRG.

To be continued: finding a good \( hc(x) \ldots \)