More on PRGs

Recommended reading: KL, Section 6.4

Recap: Last time we saw that we can use a PRG to construct a computationally secure encryption scheme with key shorter than the message.
Now, we need to construct PRGs!

Unpredictability vs. Indistinguishability

Let's discuss an equivalent perspective on pseudorandomness that will be useful later on.

Def (next-bit unpredictability):

A function $G: \{0,1\}^* \to \{0,1\}^*$ computable by a deterministic polynomial-time algorithm and mapping n bits to $\ell(n) > n$ bits is unpredictable if for every PPT algorithm A and every index i there is a negligible function $\varepsilon(n)$ such that

$$\Pr_{y \leftarrow G(U_n)}[A(y_1, \ldots, y_{i-1}) = y_i] \leq \frac{1}{2} + \varepsilon(n).$$

Thm: G is a PRG iff it is unpredictable.

Proof:

PRG $\Rightarrow$ unpred: Homework!

Unpred $\Rightarrow$ PRG: We're going to use an important proof technique in crypto called an "hybrid argument".

Suppose that G is not a PRG. This means that there is a PPT algorithm D and a polynomial $p(n)$ such that

$$\Pr(D(G(U_n))=1) - \Pr(D(U_{\ell(n)})=1) \geq 1/p(n)$$

We need to use D to construct a PPT algo D' that predicts <u>some</u> bit of $G(U_n)$ given the previous bits.

We will do this by morphing $U_{\ell(n)}$ into $G(U_n)$ bit by bit.

For $i = 0, \ldots, \ell(n)$, sample $Y^{(i)}$ as follows:

- Sample $s \leftarrow U_n$ and compute $Y = G(U_n)$
- For $j \leq i$, set $Y_j^{(i)} = Y_j$.
- For $j > i$, set $Y_j^{(i)} = Y_j^*$ with $Y_j^* \leftarrow \{0,1\}$

Intuition: $Y^{(i)}$ corresponds to the random variable where the first i bits are pseudorandom and the last $\ell(n)-i$ bits are completely random.

Then, we have $Y^{(0)} = U_{\ell(n)}$ and $Y^{(\ell(n))} = G(U_n)$.

If $p_i = \Pr(D(Y^{(i)}) = 1)$, then we know that

$$\frac{1}{p(n)} \leq p_{\ell(n)} - p_0 = \sum_{i=1}^{\ell(n)} (p_i - p_{i-1}).$$

This implies that there is i s.t. $p_i - p_{i-1} \geq \frac{1}{\ell(n) \cdot p(n)} = \text{non-negl.}$

Fix any such i.

We now show that D can be used to predict Y_i given $Y_1, \ldots, Y_{i-1}$, where we recall that $Y = G(U_n)$.

Consider the PPT algo D' behaving as follows on input $Y_1, \ldots, Y_{i-1}$:

- Sample $B_i, B_{i+1}, \ldots, B_{\ell(n)} \leftarrow \{0,1\}$
- Run $b' \leftarrow D(Y_1, \ldots, Y_{i-1}, B_i, B_{i+1}, \ldots, B_{\ell(n)})$
- Output B_i if $b' = 1$, else output $1 - B_i$.

We have

$$\begin{aligned}
&\Pr(D'(Y_1, \ldots, Y_{i-1}) = Y_i) \\
&= \frac{1}{2} \Pr(D(Y_1, \ldots, Y_{i-1}, B_i, \ldots, B_{\ell(n)}) = 1 \mid B_i = Y_i) \\
&\quad + \frac{1}{2} \Pr(D(Y_1, \ldots, Y_{i-1}, B_i, \ldots, B_{\ell(n)}) = 0 \mid B_i \neq Y_i) \\
&= \frac{1}{2} \Pr(D(Y^{(i)}) = 1) + \frac{1}{2} \Pr(D(\bar{Y}^{(i)}) = 0) \\
&= \frac{1}{2} + \frac{1}{2} (\underbrace{\Pr(D(Y^{(i)}) = 1)}_{p_i} - \underbrace{\Pr(D(\bar{Y}^{(i)}) = 1)}_{p_i'}) \quad (*)
\end{aligned}$$

where $\bar{y}^{(i)}$ is exactly like $y^{(i)}$ except that $\bar{y}_i^{(i)} = 1 - y_i^{(i)}$.

On the other hand,

$$\underbrace{\Pr(D(y^{(i-1)}) = 1)}_{p_{i-1}} = \frac{1}{2} p_i + \frac{1}{2} p_i'$$

Since $\frac{1}{\ell(n) p(n)} \leq p_i - p_{i-1} = \frac{1}{2}(p_i - p_i')$, we conclude from (*) that

$$\Pr(D'(y_1, \ldots, y_{i-1}) = y_i) = \frac{1}{2} + \frac{1}{2}(p_i - p_i') \geq \frac{1}{2} + \frac{1}{2 p(n) \ell(n)}$$

∎

In the proof above, the $y^{(i)}$'s are called **hybrids**. We're basically showing that $y^{(i-1)} \approx y^{(i)}$ for all i, which then implies $U_{\ell(n)} = y^{(0)} \approx y^{(\ell(n))} = G(U_n)$. We'll see more examples of this technique later.

The Blum-Micali Construction

We now present the Blum-Micali construction, which reduces constructing PRGs with arbitrary expansion to constructing "hardcore predicates". → a very basic object we'll try to construct concretely in the next lectures!

Def (hardcore predicate).

A predicate $P: \{0,1\}^* \to \{0,1\}$ is a hardcore predicate for $f: \{0,1\}^* \to \{0,1\}^*$ if for any PPT adversary A there exists a negligible function $\varepsilon(n)$ such that

$$\Pr_{x \leftarrow U_n}\left(A(1^n, f(x)) = P(x)\right) \leq \frac{1}{2} + \varepsilon(n),$$

and P is computable by a deterministic poly-time algo.

PRG with 1-bit expansion from hardcore predicates for permutations

Thm: If f is a permutation (↗ computable by deterministic poly-time algo. f maps n-bit strings to n-bit strings) and P is a hardcore predicate for f, then $G(s) = (f(s), P(s))$ is a PRG.

Proof: Homework! Follows easily from the unpredictability perspective on PRGs. + the fact that if $x \leftarrow U_n$ & f is a permutation, then $f(x)$ is uniform over $\{0,1\}^n$.

So, permutation + hardcore predicate gives a PRG with 1 bit expansion. How can we get more expansion?

Thm: If f is a permutation computable by a det. poly-time alg. & P is a hardcore predicate for f, then for any polynomial $\ell(n)$ there exists a PRG with expansion $\ell(n)$.

Proof: We use the unpred. perspective on PRGs again.

Fix $\ell(n)$ & for a given $s_0 \in \{0,1\}^n$ define

$$s_i = f(s_{i-1}), \quad i \geq 1$$

Then, we define $G(s_0) = P(s_{\ell(n)}), P(s_{\ell(n)-1}), \ldots, P(s_1)$

Intuition: We apply the Base PRG $G'(s_0) = (f(s_0), P(s_0))$ a total of $\ell(n)$ times reusing the first n bits of seed & retaining the last output bit in each call to G'.

Suppose that G is not a PRG. Then, there is a PPT adv. A & index i & poly $p(n)$ such that

$$\Pr\left(A(P(s_{\ell(n)}), \ldots, P(s_{\ell(n)-i+1})) = P(s_{\ell(n)-i})\right) \geq \frac{1}{2} + \frac{1}{p(n)}$$

(◇)

We're going to use A to construct a PPT distinguisher A' that breaks the hardcore predicate P.

On input $f(x)$ for $x \leftarrow U_n$, A' proceeds as follows:

→ Sets $x_1 = f(x)$ ⇝ $x_j = f(x_{j-1})$ for $j \geq 2$.

→ Runs $b' \leftarrow A(P(x_i), P(x_{i-1}), \ldots, P(x_1))$

→ Outputs b'

We claim that $\Pr_{x \leftarrow U_n}(A'(f(x)) = P(x)) \geq \frac{1}{2} + \frac{1}{p(n)}$.

To see this, note that $(x, x_1, \ldots, x_i) \sim (s_{\ell(n)-i}, s_{\ell(n)-i+1}, \ldots, s_{\ell(n)})$ follow the same joint dist. Now, invoke $(\diamondsuit)$.

$\blacksquare$