A pseudorandom generator is an efficient deterministic algorithm or circuit that takes a short uniformly random seed as its input and produces a longer output that looks indistinguishable from a uniformly random string of the same length to all efficient "adversaries" that do not know the seed. More formally,

Definition 1. A function $G : \{0,1\}^k \to \{0,1\}^m$, where $k < m$, is an *ε-pseudorandom generator* against size S circuits if for every $\{0,1\}$-valued circuit D of size at most S,

$$\left|\Pr_{s\sim\{0,1\}^k}[D(G(s)) = 1] - \Pr_{y\sim\{0,1\}^m}[D(y) = 1]\right| < \varepsilon.$$

The circuit D is called a *distinguisher*, and the difference between the two probabilities is called the *advantage* of D. Since the output of a pseudorandom generator is longer than its input, its output is *statistically* distinguishable from a uniformly random string: The distinguisher that outputs 1 on input y if $y = G(s)$ for some s and 0 otherwise has advantage at least $1 - 2^{k-m}$. However this distinguisher may in general be very large. What makes this concept interesting is the requirement that the two distributions are not distinguishable efficiently.

The notion of efficient indistinguishability is a very strong one: It implies that no single bit is substantially biased, no pair of bits are substantially correlated, the majority of any odd number of bits is close to unbiased, if the string is interpreted as the adjacency matrix of the graph then the graph has no sparse cut, and so on, as all these conditions can be verified by efficient distinguishers.

There are two types of pseudorandom generators depending of the relative computational power of the generator and the distinguisher. Pseudorandom generators that are more complex than their distinguishers have applications to the deterministic simulation of randomized algorithms and proofs. Generators that are pseudorandom even against distinguishers of higher relative complexity are a central object in cryptography. The existence of both types of pseudorandom generators turns out to be closely related to questions about computational hardness.

1 Simulating randomness

Here is a possible strategy for deterministically simulating efficient randomized algorithms for decision problems, such as the algorithm for polynomial identity testing. Suppose that we have a decision problem f, a randomized algorithm A, and an input $x \in \{0,1\}^n$ such that A solves f on x with a clear majority, namely

$$\Pr_{r\sim\{0,1\}^m}[A(x,r) = f(x)] \geq 2/3$$

where m is the amount of randomness used by the algorithm on inputs of length n. If A runs in time $t(n)$ on inputs of length n then there is a circuit C_x of size $O(t(n)^2)$ such that

$$\Pr_{r\sim\{0,1\}^m}[C_x(r) = f(x)] \geq 2/3.$$

Now if $G\colon \{0,1\}^k \to \{0,1\}^m$ was a, say, 1/6-pseudorandom generator against size $O(t(n)^2)$ circuits then

$$\begin{aligned}\left|\Pr_{s\sim\{0,1\}^k}[C_x(G(s)) = f(x)] - \Pr_{r\sim\{0,1\}^m}[C_x(r) = f(x)]\right| \\ = \left|\Pr_{s\sim\{0,1\}^k}[C_x(G(s)) = 1] - \Pr_{r\sim\{0,1\}^m}[C_x(r) = 1]\right| < 1/6\end{aligned}$$

so in particular

$$\Pr_{s\sim\{0,1\}^k}[A(x,G(s)) = f(x)] = \Pr_{s\sim\{0,1\}^k}[C_x(G(s)) = f(x)] > 2/3 - 1/6 = 1/2.$$

We can now simulate A on input x deterministically by enumerating all possible outputs of $G(s)$ and observing what fraction of the time $A(x, G(s))$ accepts. If A were to accept, a majority of the outputs $G(s)$ should yield accepting computations; if A were to reject, the majority of them should yield rejecting computations. If the output of G can be computed in time t' then the simulation takes time $O(2^k(t(n) + t'(m))$. If A was a polynomial-time algorithm we can design G so that t' grows at most polynomially in m and k grows at most logarithmically in m, then we would obtain a deterministic polynomial time algorithm that simulates A on all inputs correctly decided by it with probability at least $2/3$, in particular implying that P = BPP.

2 Converting hardness into pseudorandomness

We will show how to construct G based on the assumption that there exists a certain "explicit" and "hard" decision problem f. Suppose that we had a function f that is very hard on average with respect to the uniform distribution so that no small circuit can compute f on much more than half its inputs. Intuitively, this means no small circuit can detect noticeable correlation between the inputs and outputs of f, so the distribution $(s, f(s))$, where s is uniformly random, must be indistinguishable from random from the circuit's perspective. If f can be computed efficiently (this is what we mean be explicit), the function $G(s) = (s, f(s))$ is then a pseudorandom generator.

The following lemma formalizes this intuition. For convenience we will allow negation gates to appear anywhere in the circuit.

Lemma 2. *Suppose that for every circuit C of size S,*

$$\Pr_{s\sim\{0,1\}^k}[C(s) = f(s)] < 1/2 + \epsilon/2.$$

Then the function $G : \{0,1\}^k \to \{0,1\}^{k+1}$ given by $G(s) = (s, f(s))$ is ϵ-pseudorandom against size $S - O(1)$.

The assumption is an average-case hardness assumption: In the language of Lecture 6, f does not have heuristic *circuits* of size S and error at most $1/2 - \varepsilon/2$.

Proof. Assume D is a circuit such that

$$\big|\Pr[D(s, f(s)) = 1] - \Pr[D(s, b) = 1]\big| \geq \epsilon.$$

Here, s is chosen randomly from $\{0,1\}^k$ and b is a random bit independent of s. We will use D to construct another circuit C with only $O(1)$ more gates that computes f with probability $1/2 + \varepsilon$. By assumption, C must then have size more than S so D has size more than $S - O(1)$.

First we get rid of the absolute value. By possibly replacing the circuit D by NOT D, we may assume without loss of generality that

$$\Pr[D(s, f(s)) = 1] - \Pr[D(s, b) = 1] \geq \epsilon. \tag{1}$$

Now consider the following circuit C:

C : On input s,
  Choose $b \sim \{0,1\}$ at random
    If $D(s, b) = 1$, output b
    Otherwise output a random value in $\{0,1\}$.

Since the events $D(s,b) = 0$ and $D(s,b) = 1$ are disjoint, we can write

$$\begin{aligned}
\Pr[C(s) = f(s)] &= \Pr[D(s,b) = 1 \text{ and } b = f(s)] + \tfrac{1}{2} \cdot \Pr[D(s,b) = 0] \\
&= \Pr[D(s,f(s)) = 1 \text{ and } b = f(s)] + \tfrac{1}{2} \cdot \Pr[D(s,b) = 0] \\
&= \tfrac{1}{2} \Pr[D(s,f(s)) = 1] + \tfrac{1}{2} \cdot \Pr[D(s,b) = 0] \\
&= \tfrac{1}{2} + \tfrac{1}{2} \Pr[D(s,f(s)) = 1] - \tfrac{1}{2} \cdot \Pr[D(s,b) = 1] \\
&\geq 1/2 + \varepsilon/2.
\end{aligned}$$

□

This pseudorandom generator gives us one additional bit of pseudorandomness beyond what is contained in the seed. We now show how to get more bits using a more elaborate construction.

3 The Nisan-Wigderson generator

We can now state the transformation from hard functions into pseudorandom generators. We will say that a family $G_m \colon \{0,1\}^{k(m)} \to \{0,1\}^m$ of pseudorandom generators is polynomial-time computable if there is an algorithm that on input $s \in \{0,1\}^{k(m)}$ runs in time polynomial in m (the *output length* of G_m) and outputs $G_m(s)$.

Theorem 3. *For every polynomial S and every constant $\delta > 0$ the following holds. Suppose there is a decision problem f that can be decided in time $2^{O(t)}$ on all inputs of length t, but cannot be decided on average with error at most $1/2 - 2^{-\delta t/2}$ by any circuit of size $O(2^{\delta t})$ with respect to the uniform distribution over $\{0,1\}^t$ for all sufficiently large t. Then there exists a polynomial-time computable family $G_m \colon \{0,1\}^{k(m)} \to \{0,1\}^m$ of $1/6$-pseudorandom generators against circuits of size at most $m = S(n)$.*

Such hard problems f are believed to exist, but I cannot think of any particularly natural candidate examples.

Let us first see how we can get *two* additional bits of pseudorandomness from Lemma 2: We run the generator on two independent seeds s_1 and s_2. Namely, we let $G'(s_1,s_2) = (G(s_1), G(s_2))$. If D is a distinguisher such that

$$\left|\Pr[D(G'(s_1,s_2)) = 1] - \Pr[D(y_1,y_2) = 1]\right| \geq \varepsilon$$

then it must be the case that

$$\begin{aligned}
&\left|\Pr[D(G(s_1),G(s_2)) = 1] - \Pr[D(G(s_1),y_2) = 1]\right| \geq \varepsilon/2 \\
\text{or} \quad &\left|\Pr[D(G(s_1),y_2) = 1] - \Pr[D(y_1,y_2) = 1]\right| \geq \varepsilon/2
\end{aligned}$$

and in either case we can obtain a distinguisher D' such that

$$\left|\Pr[D'(G(s)) = 1] - \Pr[D'(y) = 1]\right| \geq 1/2$$

by hardwiring the suitable input that maximizes the advantage into D. By repeating this construction using seeds $s_1, \ldots, s_t$ we can obtain $t(k+1)$ pseudorandom bits out of a seed of length tk as long as t is not too large (as the distinguishing advantage deteriorates in t.

To further shrink the seed length, we will allow parts of the strings $s_1, \ldots, s_t$ to overlap. This motivates the following definition.

Definition 4. A collection of sets $T_1, \ldots, T_m \subseteq \{1, \ldots, k\}$ is a *combinatorial design* with set size t and intersection size $t_\cap$ if $|T_i| = t$ for every i and $|T_i \cap T_j| \leq t_\cap$ for every $i \neq j$.

Given a combinatorial design, we define a pseudorandom generator $G : \{0,1\}^k \to \{0,1\}^m$ by

$$G(s) = (f(s|_{T_1}), \ldots, f(s|_{T_m}))$$

where $f : \{0,1\}^t \to \{0,1\}$ is the "hard" function and s_T is the substring of s indexed by the elements of the set T. For example, if $s = s_1 s_2 s_3 s_4$, then $s|_{2,4} = s_2 s_4$. Combinatorial designs with good parameters can be computed efficiently.

Claim 5. *For every $c > 0$ there is a family of combinatorial designs with*

$$k = 15c^2 \log m \qquad t = c \log m \qquad t_\cap = \log m.$$

Moreover, there is a deterministic algorithm that on input 1^m runs in time $m^{O(c^2)}$ and outputs the sets $T_1, \ldots, T_m$.

In particular, for every fixed c we have $k = O(\log m)$, so the seed size is exactly what we were aiming for. Let's now check that G can be computed efficiently (in time $\text{poly}(m)$). To compute G_m, we first construct the design in time $m^{O(c^2)}$. We then need to evaluate m copies of f, each on an input of size $t = c \log m$. Since we assumed that f is computable in time $2^{O(t)} = m^{O(c^2)}$, the whole computation can be done in time polynomial in m.

It remains to show that G_m is 1/6-pseudorandom against circuits of every polynomial size $S(n)$ for a suitable choice of constants c and δ. Towards a contradiction, suppose that for some circuit C of size $S(n)$ we have

$$\Pr_{s \sim \{0,1\}^k}[C(G(s)) = 1] - \Pr_{r \sim \{0,1\}^m}[C(r) = 1] \geq 1/6.$$

(As in the last proof, we can remove the absolute value without loss of generality.) Let's expand this definition:

$$\Pr_{s \sim \{0,1\}^k}[C(f(s|_{T_1}), \ldots, f(s|_{T_m})) = 1] - \Pr_{r_1, \ldots, r_m \sim \{0,1\}}[C(r_1, \ldots, r_m) = 1] \geq 1/6. \tag{2}$$

This formula does not appear all that useful. To see what is happening, we introduce of the following way of "slowly" going from the pseudorandom distribution $G(s)$ to the random distribution r: At each step, we change one input of C from pseudorandom to random. If C can distinguish $G(s)$ from r, then at some step there must be a noticeable change in the behavior of C.

More formally, we consider the following sequence of *hybrid distributions* on inputs of C:

$$\begin{array}{rcccc}
D_m: & f(s|_{T_1}), & \ldots, & f(s|_{T_{m-1}}), & f(s|_{T_m}) \\
D_{m-1}: & f(s|_{T_1}), & \ldots, & f(s|_{T_{m-1}}), & r_m \\
& \vdots & & \vdots & \vdots \\
D_0: & r_1, & \ldots, & r_{m-1}, & r_m.
\end{array}$$

These distributions are not "real"; we merely use them to help us in the analysis. Condition (2) tells us that $\Pr_{r \sim D_m}[C(r)] - \Pr_{r \sim D_0}[C(r)] \geq 1/6$. Then there must be some j between 1 and m for which $\Pr_{r \sim D_j}[C(r)] - \Pr_{r \sim D_{j-1}}[C(r)] \geq 1/6m$, that is

$$\begin{aligned}
\Pr_{s \sim \{0,1\}^k, r_{j+1}, \ldots, r_m \sim \{0,1\}}&[C(f(s|_{T_1}), \ldots, f(s|_{T_j}), \ldots, r_m) = 1] \\
&- \Pr_{s \sim \{0,1\}^k, r_j, \ldots, r_m \sim \{0,1\}}[C(f(s|_{T_1}), \ldots, r_j, \ldots, r_m) = 1] \geq 1/6m.
\end{aligned}$$

There must then exist a fixing of the values $r_{j+1}, \ldots, r_m$ that maximizes the above difference in probabilities. If we hardwire this fixing into the circuit C, we obtain a circuit C_1 of the same size such that

$$\Pr_s[C_1(f(s|_{T_1}), \ldots, f(s|_{T_{j-1}}), f(s|_{T_j})) = 1] - \Pr_{s, r_j}[C_1(f(s|_{T_1}), \ldots, f(s|_{T_{j-1}}), r_j) = 1] \geq 1/6m.$$

Let $s' = s|_{T_j}$ (this is a string of length t). There is now a fixing of all the bits of s outside s' that maximizes the above difference in probabilities. Let's hardwire these bits into C_1 and call the resulting circuit C_2. With respect to this fixing, for every $i < j$, $f(s|_{T_i})$ becomes a function of at most $10 \log m$ bits in s' (because s' intersects $s|_{T_i}$ in at most $t_\cap = \log m$ positions). Let's call this function $g_i(s')$. We then have

$$\Pr_{s'}[C_2(g_1(s'), \ldots, g_{j-1}(s'), f(s')) = 1] - \Pr_{s',r_j}[C_2(g_1(s'), \ldots, g_{j-1}(s'), r_j) = 1] \geq 1/6m.$$

Since each g_i is a function of at most $\log m$ bits, it can be computed by a circuit of size $O(2^{\log m}) = O(m)$. If we compose the circuit C_2 with the circuits for $g_1, \ldots, g_{j-1}$, we obtain a single circuit C_3 of size $S(n) + O(jm) = S(n) + O(m^2)$ such that

$$\Pr_{s'}[C_3(s', f(s')) = 1] - \Pr_{s',r_j}[C_3(s', r_j) = 1] \geq 1/6m.$$

In words, $(s', f(s'))$ is not $1/6m$-pseudorandom for size $S(n) + O(m^2)$; by Lemma 2 it follows that there is a circuit C_4 of size $S(n) + O(m^2)$ such that

$$\Pr_{s'}[C_4(s') = f(s')] \geq 1/2 + 1/6m.$$

Recall that f is a function on $t = c \log m$ bits, so we have that

$$\Pr_{s'}[C_4(s') = f(s')] \geq 1/2 + 1/6 \cdot 2^{-t/c}$$

where C_4 is a circuit of size $S(n) + O(m^2) = O(2^{2t/c})$. If we choose $\delta = 2/c$ we obtain that C_4 is a circuit of size $O(2^{\delta t})$ that predicts f with advantage $2^{-\delta t/2}$, contradicting the assumed hardness of f.

Proof of Claim 5. The sets $T_1, \ldots, T_m$ are chosen greedily: T_i is the first set of size t that has intersection size at most $t_\cap$ with all sets $T_j, j < i$. The running time is dominated by the number of possible choices for each set which is at most $2^k = m^{O(c^2)}$.

We show that a choice of T_i with the desired properties is always possible by the probabilistic method. The probability that the intersection between T_i and any fixed set T_j, $j < i$, both of size t, exceeds size $t_\cap$ is at most

$$\binom{t}{t_\cap}^2 \cdot \left(\frac{t_\cap}{k}\right)^{t_\cap} \leq (ec)^{2 \log m} \cdot (15c^2)^{-\log m} \leq \frac{1}{m}$$

so by a union bound, there must always exists a choice of T_i that has intersection less than $t_\cap$ with T_1 up to T_{i-1}. □

4 Cryptographic pseudorandom generators

In the above constructions the time that it takes to compute the pseudorandom generator G is inherently larger than the time it takes to evaluate the hard function f as G must evaluate one or several copies of f. Therefore the complexity of computing G will be larger than the complexity of distinguishing the output of G from a uniformly random string.

In cryptographic applications the pseudorandom generator is usually part of the system and the distinguisher is the adversary that wants to break the system. It is usually assumed that the adversary is willing to invest more resources into breaking the system than the honest parties use to run it. In this setting generating even one additional bit of pseudorandomness beyond the seed length is challenging.

Cryptographic pseudorandom generators can be obtained from one-way functions. The transformation works both in the model of circuits and Turing Machines. For simplicity here is the circuit variant.

Theorem 6. *For every function $f\colon \{0,1\}^n \to \{0,1\}^m$ computable by a circuit of size s there exists a function $G\colon \{0,1\}^k \to \{0,1\}^{k+1}$ computable by a circuit of size polynomial in s and n so that the following holds: If f is (S,ε)-one-way then G is ε'-pseudorandom against size S' circuits, where $S' = S - \mathrm{poly}(s,n)$ and $\varepsilon' = \varepsilon \cdot \mathrm{poly}(s,n)$.*

Once one additional bit of pseudorandomness is obtained, it is possible to increase the length of the output by applying the pseudorandom generator iteratively. Specifically, if $G\colon \{0,1\}^k \to \{0,1\}^{k+1}$ is a pseudorandom generator, then we iteratively define $G_0(s) = s$ and

$$G_{d+1}(s) = (G(\text{first } k \text{ bits of } G_d(s)), \text{last } d \text{ bits of } G_d(s)).$$

Then we can prove the following by induction on d.

Lemma 7. *If G can be computed by a circuit of size s and G is ε-pseudorandom against size S circuits then G_d is $d\varepsilon$-pseudorandom against size $S - (d-1)s$ circuits.*

Proof. For contradiction, let us suppose that G_{d+1} is not pseudorandom. So there exists a circuit D of size $S - ds$ such that

$$|\Pr[D(G_{d+1}(s)) = 1] - \Pr[D(y) = 1]| \geq (d+1)\varepsilon.$$

Here, Y is a truly random string of length $k+d+1$. Now recall that G_{d+1} was obtained by running G on the first k bits of y_d (the output of G_d) and copying the last d bits. Let $x_d = x_L x_R$, where X_L are the first k bits and y_R are the last d. Also let $y = y_L y_R$ where y_L are the first $k+1$ bits and Y_R are the last d bits. Then

$$|\Pr[D(G(x_L), x_R) = 1] - \Pr[D(y_L, y_R) = 1]| \geq (d+1)\varepsilon.$$

Now let z be a uniform string of length k independent of y. At least one of these two inequalities must hold:

$$|\Pr[D(G(x_L), x_R) = 1] - \Pr[D(G(z), y_R) = 1]| \geq d\varepsilon \qquad \text{or}$$
$$|\Pr[D(G(z), y_R) = 1] - \Pr[D(y_L, y_R) = 1]| \geq \varepsilon.$$

Suppose the first inequality holds. Then we can distinguish x from a truly random string as follows:

> Circuit D': On input u, write $u = u_L u_R$ (the first k and last d bits) and output $D(G(u_L), u_R)$.

This circuit D' has size $S - (d-1)s$ and by the first inequality

$$|\Pr[D'(x_L, x_R) = 1] - \Pr[D'(z, y_R) = 1]| = |\Pr[D(G(x_L), x_R) = 1] - \Pr[D(G(z), y_R) = 1]| \geq d\varepsilon$$

so D' distinguishes the output of $G_d(s)$ from a random string with advantage $d\varepsilon$, contradicting our inductive assumption. So the second inequality must hold. But then we can distinguish the output of G from random by the following circuit: On input u, choose a random string y_R of length d and output $D(u, y)$. By the second inequality, this is a circuit of size $S - ds + d \leq S$ that distinguishes the output of G from random with advantage ε, contradicting the pseudorandomness of G. □