Pseudo Random Generators

A pseudorandom generator is a deterministic algorithm that receives a short truly random seed and stretches it into a long string that is pseudorandom. Stated differently, a pseudorandom generator uses a small amount of true randomness in order to generate a large amount of pseudorandomness. In the definition that follows, we set n to be the length of the seed that is input to the generator and l(n) to be the output length. Clearly, the generator is only interesting if l(n) > n (otherwise, it doesn't generate any new "randomness").

Let l(·) be a polynomial and let G be a deterministic polynomial-time algorithm such that upon any input s ϵ {0, 1} n, algorithm G outputs a string of length l(n). We say that G is a pseudorandom generator if the following two conditions hold:

1. Expansion: For every n it holds that l(n) > n.
2. Pseudorandomness: For all probabilistic polynomial-time distinguishers D, there exists a negligible function negl such that: Pr[D(r) = 1] - Pr[D(G(s)) = 1] ≤ negl(n), where r is chosen uniformly at random from $\{0, 1\}^{l(n)}$, the seed s is chosen uniformly at random from $\{0, 1\}^{n}$, and the probabilities are taken over the random coins used by D and the choice of r and s.

The function l(·) is called the expansion factor of G.

Pseudorandom strings are just as good as truly random ones, as long as the seed is kept secret and we are considering only polynomial-time observers. Seed length must be long enough so that it is impossible to efficiently try all possible seeds by a polynomial time algorithm (Could be brute force). Unfortunately, we do not know how to unequivocally prove the existence of PRNGs. They can be generated under the assumption that certain long studied problems are "hard" i.e. no efficient polynomial time algorithms exist to solve them.