Pseudo-random Generators from One-way Functions

Michael Luby
International Computer Science Institute
Berkeley, California, U.S.A.

Abstract

One of the basic primitives in cryptography and other areas of computer science is a pseudo-random generator. The usefulness of a pseudo-random generator is demonstrated by the fact that it can be used to construct a private key cryptosystem that is secure even against chosen plaintext attack. A pseudo-random generator can also be used to conserve random bits and allows reproducibility of results in Monte Carlo simulation experiments. Intuitively, a *pseudo-random generator* is a polynomial time computable function g that stretches a short random string x into a much longer string $g(x)$ that "looks" just like a random string to *any* polynomial time adversary that is allowed to examine $g(x)$.[1] Thus, a pseudo-random number generator can be used to efficiently convert a small amount of true randomness into a much longer string that is indistinguishable from a truly random string of the same length to any polynomial time adversary.

On the other hand, there seem to be a variety of natural exampleꞷ of another basic primitive; the one-way function. Intuitively, a function f is *one-way* if: (1) given any x, $f(x)$ can be computed in polynomial time; (2) given $f(x)$ for a randomly chosen x, it is not possible on average to find an inverse x' such that $f(x') = f(x)$ in polynomial time. It has not been proven that there are any one-way functions, but there are a number of problems from number theory, coding theory, graph theory and combinatorial theory that are candidates for problems that might eventually be proven to be one-way functions.

We show how to construct a pseudo-random generator from *any* one-way function. The journal version of this work (in preparation) is the combination of the results announced in the conference papers [ILL, Impagliazzo Levin Luby] and [H, Håstad].

References

[1] Impagliazzo, R., Levin, L. and Luby, M, "Pseudo-random number generation from one-way functions", 21^{rst} *STOC*, 1989, pp 12-24.

[2] Håstad, J. "Pseudo-Random Generators under Uniform Assumptions", 22nd *STOC*, 1990, pp 395-404.

[1] This should be contrasted with the classical definition of a pseudo-random generator. A classical pseudo-random generator is required to pass a particular set of statistical tests, but does not necessarily satisfy the more general requirement that it pass all polynomial time tests. This is a particularly important distinction in the context of cryptography, where the adversary must be assumed to be as malicious as possible, with the only restriction on tests being computation time.

J. Feigenbaum (Ed.): Advances in Cryptology - CRYPTO '91, LNCS 576, p. 300, 1992.