How to Construct Pseudo-random Permutations from Pseudo-random Functions

Michael Luby
Charles Rackoff
Department of Computer Science
University of Toronto
Toronto, Canada M5S 1A4

ABSTRACT

Let F^n be the set of all functions from n bits to n bits. Let f^n specify for each key k of a given length a function $f_k^n \in F^n$. We say f^n is **pseudo-random** if the following two properties hold:

(1) Given a key k and an input α of length n, the time to evaluate $f_k^n(\alpha)$ is polynomial in n.

(2) If a random key k is chosen, f_k^n "looks like" a random function chosen from F^n to any algorithm which is allowed to evaluate f_k^n at polynomial in n input values.

Let P^{2n} be the set of permutations (1-1 onto functions) from $2n$ bits to $2n$ bits. Let p^{2n} specify for each key k of a given length a permutation $p_k{}^{2n} \in P^{2n}$. We present a simple method for describing p^{2n} in terms of f^n. The method has the property that if f^n is pseudo-random then p^{2n} is also pseudo-random. The method was inspired by a study of the security of the Data Encryption Standard. This result, together with the result of Goldreich, Goldwasser and Micali [GGM], implies that if there is a pseudo-random number generator then there is a pseudo-random invertible permutation generator. We also prove that if two permutation generators which are "slightly secure" are cryptographically composed, the result is more secure than either one alone.

H.C. Williams (Ed.): Advances in Cryptology - CRYPTO '85, LNCS 218, p. 447, 1986.