Appeared in *Advances in Cryptology - EUROCRYPT '92*, Lecture Notes in Computer Science, Berlin: Springer-Verlag, vol. 658, pp. 239-255, 1992.

# A Simplified and Generalized Treatment of Luby-Rackoff Pseudorandom Permutation Generators

Ueli M. Maurer

Institute for Theoretical Computer Science
ETH Zürich
CH-8092 Zürich, Switzerland
Email address: maurer@inf.ethz.ch

**Abstract.** A paper by Luby and Rackoff on the construction of pseudorandom permutations from pseudorandom functions based on a design principle of the DES has recently initiated a burst of research activities on applications and generalizations of these results. This paper presents a strongly simplified treatment of these results and generalizes them by pointing out the relation to locally random functions, thereby providing new insight into the relation between probability-theoretic and complexity-theoretic results in cryptography. The first asymptotically-optimal construction of a locally random function is presented and new design strategies for block ciphers based on these results are proposed.

**Keywords.** Locally random function, Pseudorandom function, Pseudorandom permutation, Luby-Rackoff permutation generator.

# 1. Introduction

In a celebrated complexity-theoretic paper [9], Luby and Rackoff described a construction of a pseudorandom permutation generator from any pseudorandom function generator that was motivated by a study of the Data Encryption Standard (DES, cf. [4]). Much research has recently been based on this paper (e.g., [11], [12], [13], [14]). The main goal of the present paper is to give a simplified and generalized treatment of the results of [9] by suggesting an information-theoretic rather than complexity-theoretic interpretation based on the concept of locally random functions whose treatment is an independent goal of this paper. It is shown that the proof of the Main Lemma of [9], which originally required three pages of highly technical definitions, claims and arguments, can be strongly simplified and interpreted essentially as an application of the birthday paradox, thus providing much more insight. Moreover, the central proposition of [9], which was used to establish the relation between probability-theoretic arguments and complexity-theoretic results and which was unfortunately stated without proof, is shown to be unnecessary and somewhat misleading.

Local randomness is an important concept in theoretical computer science with several applications. Intuitively, a family of functions is locally random of degree $k$ if for every set of at most $k$ arguments, the function values for these arguments for a randomly (from the family) chosen function are independent and uniformly distributed. In other words, a randomly (from the family) chosen function behaves precisely like a truly random function as long as it is evaluated for at most $k$ arguments. Similarly, a sequence generator is locally random of degree $k$ [10] if for a randomly selected seed, every subset of $k$ (or less) digits is completely random. Clearly, a locally random sequence generator can be obtained from a locally random function by "reading out" the function values for a given enumeration of the arguments, but the converse is not true in general because a sequence generator need not have the property that arbitrary digits can be accessed efficiently (only consecutive digits must be efficiently computable).

The usefulness of local randomness has previously been observed (e.g., [1], [3], [6], [7]) and was referred to as $k$-wise independence. However, our treatment is more general in that (1) families of functions that are only "almost" locally random of degree $k$ and (2) polynomial-time computable functions with super-polynomial degree of local randomization are considered, allowing applications in complexity theory as well as for the design of practical block ciphers.

The results of Luby and Rackoff are discussed in Section 3. Locally random functions are introduced in Section 4, and an alternative interpretation and generalization of the results of Luby and Rackoff based on this new concept are described Section 5. Some further applications of locally random functions and a new design strategy for block ciphers are discussed in Section 6.

## 2. Terminology

Our terminology is similar to that of [9]. Let $\{0,1\}^n$ denote the set of binary strings of length $n$, let $F^n$ denote the set of all $2^{n2^n}$ functions $\{0,1\}^n \to \{0,1\}^n$, and let $P^n$ denote the subset of functions of $F^n$ that are permutations of $\{0,1\}^n$, i.e., invertible or one-to-one. For $f_1 \in F^n$ and $f_2 \in F^n$, $f_1 \circ f_2$ denotes the composition of $f_1$ and $f_2$, i.e., $f_1 \circ f_2(x) = f_2(f_1(x))$.

For two binary strings $a$ and $b$, $a \bullet b$ denotes their concatenation and when $a$ and $b$ have the same length, $a \oplus b$ denotes their bit-by-bit *exclusive or*. The string consisting of the $t$ rightmost bits of a string $a$ is denoted by $[a]_t$. In particular, $[i]_t$ for a non-negative integer $i < 2^t$ denotes the representation of $i$ by $t$ bits (with possible leading zeroes).

When an argument of a function is replaced by a set of arguments this will denote the multiset of resulting function values. In all cases, a random choice of an object $x$ from a set or multiset $S$ of objects (denoted by $x \in_R S$) will be such that each object is equally likely to be chosen, taking into account multiple occurrencies in multisets. We refer to Section 4 of [9] for definitions of a pseudorandom number (or bit) generator (PRNG), of a pseudorandom function generator (PRFG) and of a pseudorandom permutation generator (PRPG). A function $f : \mathbb{N} \to \mathbb{N}$ is called superpolynomial if for every polynomial $Q$, $f(n) > Q(n)$ for all sufficiently large $n$. Finally, $\#S$ denotes the cardinality of the set or multiset $S$, and all logarithms in this paper are to the base 2.

## 3. Luby-Rackoff Pseudorandom Permutation Generators

Levin [8] gave a construction of a PRNG from any one-way function, and Goldreich, Goldwasser and Micali [5] devised a method for constructing a PRFG from any PRNG and hence, by Levin's result, also from any one-way function. (As a by-product of this research a simpler construction of a PRFG from any PRNG will be described in Section 4.) A PRNG can be used for encryption in a so-called additive stream cipher but a PRFG cannot directly be used for (block) encryption because pseudorandom functions are not invertible in general. Luby and Rackoff considered the problem of constructing a secure block encryption algorithm, i.e., a (secure) pseudorandom permutation generator, from any (secure) PRFG, and hence from any PRNG or from any one-way function. We refer to [9] for definitions.

Motivated by the round structure of the Data Encryption Standard DES (cf. [4]), Luby and Rackoff defined a mapping $H : F^n \times F^n \times F^n \to P^{2n}$ assigning every triple of functions in $F^n$ a permutation in $P^{2n}$. Let $L$ and $R$ denote the left and right half of a $2n$-bit string $L \bullet R$ and let for $f \in F^n$ the permutation $\overline{f} \in P^{2n}$ be defined as

$$\overline{f}(L \bullet R) = R \bullet [L \oplus f(R)],$$

i.e., the right half of the argument appears unchanged as the left half of the result and the right half of the result is equal to $L \oplus f(R)$. This corresponds to one round of DES. For a list of functions, $f_1, \ldots, f_s \in F^n$, let the function (actually a permutation) $\psi(f_1, \ldots, f_s) : \{0,1\}^{2n} \to \{0,1\}^{2n}$ be defined by

$$\psi(f_1, \ldots, f_s) \;=\; \overline{f}_1 \circ \cdots \circ \overline{f}_s,$$

i.e., $\psi(f_1, \ldots, f_s)(L \bullet R) \;=\; \overline{f}_s(\overline{f}_{s-1}(\cdots \overline{f}_1(L \bullet R) \cdots))$. Note that $H$ can now be defined by $H(f_1, f_2, f_3) = \psi(f_1, f_2, f_3)$ (cf. Figure 1), where

$$\psi(f_1, f_2, f_3)(L \bullet R) = [R \oplus f_2(L \oplus f_1(R)] \bullet [L \oplus f_1(R) \oplus f_3(R \oplus f_2(L \oplus f_1(R)))].$$

Luby and Rackoff considered the problem of distinguishing, by use of an oracle circuit, a function randomly chosen from $F^{2n}$ from a function randomly chosen from the much smaller set $\psi(F^n, F^n, F^n)$. An oracle circuit $C_{2n}$ is a circuit with oracle gates, i.e., gates with a $2n$-bit input and a $2n$-bit output, where all oracle gates in a circuit evaluate the same fixed function in $F^{2n}$ (for details see [9]). Let

$$P[C_{2n}(f) = 1 : f \in_R \psi(F^n, F^n, F^n)]$$

and

$$P[C_{2n}(f) = 1 : f \in_R F^{2n}]$$

denote the probabilities that $C_{2n}$ outputs 1 if the oracle gates are evaluated for a function chosen randomly from $\psi(F^n, F^n, F^n)$ and from $F^{2n}$, respectively. We hope that this notation, which differs slightly from that of [9], is more intuitive. The Main Lemma of [9] is as follows.

**Main Lemma of [9].** *Let $C_{2n}$ be an oracle circuit with $k$ oracle gates such that no input value is repeated to an oracle gate. Then*

$$\left| P[C_{2n}(f) = 1 : f \in_R \psi(F^n, F^n, F^n)] - P[C_{2n}(f) = 1 : f \in_R F^{2n}] \right| \leq k^2/2^n.$$

From the following discussion it will become clear that the restriction to circuits whose oracle gates must have different inputs, and hence also the proposition stated (unfortunately without proof) in [9] above the Main Lemma, are unnecessary and somewhat misleading. The result can be stated as a purely probability-theoretic result having no direct relation to complexity theory, and will in Section 5 be interpreted as a result on locally random functions.

Let $g : (\{0,1\}^{2n})^k \to \{0,1\}$ be a function taking as input $k$ $2n$-bit strings. For a given set of $k$ arguments $x_1, \ldots, x_k$, let in analogy to the above definitions

$$P[g(f(x_1), \ldots, f(x_k)) = 1 \colon f \in_R \psi(F^n, F^n, F^n)]$$

and

$$P_g \ \stackrel{\triangle}{=} \ P[g(f(x_1), \ldots, f(x_k)) = 1 \colon f \in_R F^{2n}] \tag{1}$$

be defined as the probabilities that $g(f(x_1), \ldots, f(x_k)) = 1$ when $f$ is chosen randomly from $\psi(F^n, F^n, F^n)$ and from $F^{2n}$, respectively. Note that $P_g$ can alternatively be defined as

$$P_g \ = \ P[g(r_1, \ldots, r_k) = 1]$$

where $r_1, \ldots, r_k$ are independent and randomly selected from $\{0,1\}^{2n}$. Again equivalently, $P_g$ can also be defined as

$$P_g \ = \ \frac{\#\{(r_1, \ldots, r_k) \in (\{0,1\}^{2n})^k : g(r_1, \ldots, r_k) = 1\}}{2^{2nk}}.$$

**Lemma 1.** *For every function $g : (\{0,1\}^{2n})^k \to \{0,1\}$ and for every set of $k$ arguments $x_1, \ldots, x_k$,*

$$\Big| P[g(f(x_1), \ldots, f(x_k)) = 1 \colon f \in_R \psi(F^n, F^n, F^n)] \ - \ P_g \Big| \ \le \ k^2/2^n.$$

Clearly, Lemma 1 is also true for every function $g : (\{0,1\}^{2n})^{k'} \to \{0,1\}$ with $k' < k$. It demonstrates that there exists no set of $k$ arguments, whether adaptively chosen or not, and whether distinct or not, that would allow an oracle circuit with these arguments as the inputs to the oracle gates to achieve

$$\Big| P[C_{2n}(f) = 1 \colon f \in_R \psi(F^n, F^n, F^n)] - P[C_{2n}(f) = 1 \colon f \in_R F^{2n}] \Big| \ > \ k^2/2^n.$$

The Main Lemma of [9] is hence an immediate consequence of Lemma 1. (It is easy to see that the converse is also true.) Moreover, it is obvious that probabilistic strategies cannot be better than deterministic ones for distinguishing a function from a random function since the deterministic function $g$ could be defined as that resulting for the optimal choice of the randomizer.

*Proof of Lemma 1.* Let $f_1, f_2$ and $f_3$ be functions randomly chosen from $F^n$, and let $f = \psi(f_1, f_2, f_3)$. Let $x_i = L_i \bullet R_i$ for $1 \le i \le k$ be the $k$ arguments of $f$, and define $S_i, T_i$ and $V_i$ for $1 \le i \le k$ as follows (cf. Figure 1):

$$S_i \ = \ L_i \oplus f_1(R_i),$$

$$T_i \ = \ f_2(S_i) \oplus R_i$$

and

$$V_i \;=\; f_3(T_i) \oplus S_i.$$

Note that when the evaluation of $f$ for the argument $x_i$ is viewed as a three-round process (similar to three rounds of DES), the outputs of the first, second and third round are $R_i \bullet S_i$, $S_i \bullet T_i$ and $T_i \bullet V_i = f(L_i \bullet R_i)$, respectively. We may for the rest of the proof assume without loss of generality that the $x_i$, $1 \leq i \leq k$, are distinct. Choosing identical arguments provides no new information and can thus certainly not help.

Figure 1. Computation of $\psi(f_1, f_2, f_3)(L \bullet R)$ as $T_i \bullet V_i$. For the sake of simplicity, the swaps of left and right half between rounds are not shown explicitly.

Let $\mathcal{E}_S$ and $\mathcal{E}_T$ denote the events that $S_1, \ldots, S_k$ are distinct and that $T_1, \ldots, T_k$ are distinct, respectively, and let $\mathcal{E}$ be the event that both $\mathcal{E}_S$ and $\mathcal{E}_T$ occur. If $\mathcal{E}_S$ occurs, then $T_1 = R_1 \oplus f_2(S_1), \ldots, T_k = R_k \oplus f_2(S_k)$ are completely random because $f_2$ is a random function and hence $f_2(S_1), \ldots, f_2(S_k)$ are completely random. Similarly, if $\mathcal{E}_T$ occurs, then $V_1 = S_1 \oplus f_3(T_1), \ldots, V_k = S_k \oplus f_3(T_k)$ are completely random because $f_3$ is a random function. Thus if both $\mathcal{E}_S$ and

$\mathcal{E}_T$ occur, $f(x_1) = T_1 \bullet V_1, \ldots, f(x_k) = T_k \bullet V_k$ are completely random and thus $f = \psi(f_1, f_2, f_3)$ behaves precisely like a function chosen randomly from $F^{2n}$. Therefore the distinguishing probability is upper bounded by

$$\left| P[g(f(x_1), \ldots, f(x_k)) = 1 \colon f \in_R \psi(F^n, F^n, F^n)] - P_g \right| \;\leq\; 1 - P[\mathcal{E}].$$

We now derive an upper bound on $1 - P[\mathcal{E}] = P[\overline{\mathcal{E}}]$, where $\overline{\mathcal{E}}$ denotes the complementary event of $\mathcal{E}$. $\overline{\mathcal{E}}$ is the union of the $\binom{k}{2}$ events $\{S_i = S_j\}$ for $1 \leq i < j \leq k$ and the $\binom{k}{2}$ events $\{T_i = T_j\}$ for $1 \leq i < j \leq k$. The probability of the union of several events is upper bounded by the sum of the probabilities, and hence

$$1 - P[\mathcal{E}] = P[\overline{\mathcal{E}}] \;\leq\; \sum_{1 \leq i < j \leq k} P[S_i = S_j] + \sum_{1 \leq i < j \leq k} P[T_i = T_j]. \tag{2}$$

For $i \neq j$ we have

$$P[S_i = S_j] \;=\; \begin{cases} 2^{-n} & \text{if } R_i \neq R_j \\ 0 & \text{if } R_i = R_j. \end{cases} \tag{3}$$

Note that when $R_i = R_j$, then $P[S_i = S_j] = 0$ since by assumption $L_i \bullet R_i \neq L_j \bullet R_j$ and hence $L_i \neq L_j$. Equation (3) shows that

$$P[S_i = S_j] \leq 2^{-n}$$

for $i \neq j$. By a similar argument we obtain

$$P[T_i = T_j] \leq 2^{-n}$$

for $i \neq j$. The total number of terms on the right side of (2) is $2\binom{k}{2} = k(k-1) < k^2$. Lemma 1 follows. $\Box$

An interpretation and generalization of this result based on locally random functions, which are introduced in the following section, will be presented in Section 5.

# 4. Random Functions and Locally Random Functions

A random function $r : \{0,1\}^n \to \{0,1\}^n$ is a function that assigns to all arguments $x \in \{0,1\}^n$ independent and completely random values $r(x) \in \{0,1\}^n$. The trivial implementation of a random function as a table requires the generation of $n2^n$ random bits during a precomputation phase and $n2^n$ bits of memory to store the table.

A random function can alternatively be implemented as a device (or procedure) that, when given as input an argument $x$ that was never given before, generates a random output $r(x)$ and stores the pair $(x, r(x))$ in a table ordered according to $x$, and when given as input an argument $x$ for which $r$ was previously evaluated, outputs $r(x)$ stored in the table. An advantage of the latter implementation is that when $r$ needs to be evaluated for at most $t$ arguments, $2tn$ bits of memory are required and at most $2tn$ random bits need to be generated. However, the computation time for each argument is $O(\log t)$ compared to $O(1)$ for an implementation based on a pregenerated table (of exponential size), and hence depends on the number $t$ of arguments.

When the computation time of an algorithm accessing a random function, implemented as described above, is polynomially (in $n$) bounded, so are the total computation time and memory requirements of the resulting algorithm, including the random function. In other words, although a random function seems at first to require an exponential amount of memory, any polynomial-time algorithm using random functions can be implemented in polynomial time and polynomial space. This observation is the key argument of the proof of Theorem 1 of [9]. We would like to point out (without making further use of this result) that the same observation can be used to present a construction of a PRFG from a PRNG that is much simpler (albeit less practical) than that proposed by Goldreich, Goldwasser and Micali [5] for proving the following proposition.

**Proposition 1** [5]. *Pseudorandom function generators exist if and only if pseudorandom number generators exist.*

Randomness is often an expensive and limited resource. Moreover, a dependence of the function evaluation time and memory requirement on the number $t$ of arguments for which a function is evaluated is most often intolerable. Therefore, an important concept is that of a locally random function, i.e., a function that behaves like a random function as long as it is evaluated for at most $k$ arguments for some parameter $k$.

**Definition 1.** A family $\mathcal{F}_{\mathcal{Z}} = \{f_z : z \in \mathcal{Z}\}$ of functions $f_z : \{0,1\}^n \to \{0,1\}^m$ is an $(n, m, k)$ *locally random function (LRF)* with key space $\mathcal{Z}$ if for every subset $\{x_1, \ldots, x_k\}$ of $\{0,1\}^n$, $f_z(x_1), \ldots, f_z(x_k)$ are uniformly distributed over $\{0,1\}^m$ and jointly statistically independent, when $z$ is randomly selected from $\mathcal{Z}$.

$\mathcal{F}_{\mathcal{Z}}$ could alternatively be viewed as a single function $\mathcal{Z} \times \{0,1\}^n \to \{0,1\}^n$. A random function $\{0,1\}^n \to \{0,1\}^n$ is an $(n, n, 2^n)$ LRF. The restriction to binary digits is made without essential loss of generality. The above definition is purely combinatorial, i.e., no restriction on the computation time is made. LRFs will be generalized below to take into account both minor deviations from complete randomness of any $k$ function values and efficient (i.e., polynomial-time) computability.

An important question is for which choices of parameters $n, m, k$ and $|\mathcal{Z}|$ there exist LRFs. Because it is impossible to expand deterministically a sequence of random bits into a longer sequence of (independent) random bits, it is obvious that for an $(n, m, k)$ LRF with key space $\mathcal{Z}$,

$$|\mathcal{Z}| \;\geq\; 2^{km} \tag{4}$$

must hold. It may appear to be somewhat surprising that, for any $n, m$ and $k$ with $m = n$ or $m$ a multiple of $n$, equality in (4) can be achieved. This follows from the following well-known proposition, which can be proved by observing that the $d+1$ coefficients of a polynomial of degree $d$ over a field can be interpolated from any set of $d+1$ arguments and the corresponding polynomial values. When $m < n$, equality in (4) cannot be achieved.

**Proposition 2.** *Let $p_0, \ldots, p_{k-1}$ be randomly selected $n$-bit strings. The function*

$$p : \{0,1\}^n \to \{0,1\}^n : \; x \mapsto p(x) = p_{k-1}x^{k-1} + \cdots + p_1 x + p_0,$$

*where all quantities are considered as representations of elements of the finite field $GF(2^n)$, is a $(n, n, k)$ LRF with minimal key space $\mathcal{Z} = \{0,1\}^{kn}$.*

For the sake of completeness we state the following proposition, which is an immediate consequence of Theorem 1 in [10]. Let $\mathcal{Z} = \{0,1\}^v$, i.e., the key consists of $v$ binary digits.

**Proposition 3.** *There exists a $(n, m, k)$ LRF if*

$$mk \;\leq\; \frac{v}{n + log_2 m}$$

*and there exists no $(n, 1, k)$ LRF if*

$$k \;>\; \frac{2(v + n + 1)}{n - \log_2 v + 1}.$$

In order to state our results on PRFGs and PRPGs in terms of LFRs, we need to generalize the concept of LRFs in two different ways. As a first generalization, the condition of true randomness of any $k$ function values must be somewhat relaxed. Instead of introducing the new concept of "almost" locally random functions we generalize LRFs by introducing a fourth parameter, $\epsilon$, believing that this generalization will be intuitive rather than ambiguous. (Note that a $(n, m, k, 0)$ LRF will be the same as a $(n, m, k)$ LRF.)

**Definition 1'.** A family $\mathcal{F}_{\mathcal{Z}} = \{f_z : z \in \mathcal{Z}\}$ of functions $f_z : \{0,1\}^n \to \{0,1\}^m$ is an $(n, m, k, \epsilon)$ *locally random function* with key space $\mathcal{Z}$ if for all functions $g :$

$(\{0,1\}^m)^k \rightarrow \{0,1\}$ and for every subset $\{x_1,\ldots,x_k\}$ of $\{0,1\}^n$, for $z$ randomly selected from $\mathcal{Z}$,

$$\Big|P[g(f_z(x_1),\ldots,f_z(x_k))=1]-P[g(r_1,\ldots,r_k)=1]\Big| \;\leq\; \epsilon,$$

where $r_1,\ldots,r_k$ are independent and randomly selected from $\{0,1\}^m$.

Note that

$$P[g(f_z(x_1),\ldots,f_z(x_k))=1]=\frac{\#\{z\in\mathcal{Z}:g(f_z(x_1),\ldots,f_z(x_k))=1\}}{\#\mathcal{Z}}.$$

Clearly, an $(n,m,k,\epsilon)$ LRF is also an $(n,m,k',\epsilon')$ LRF for every $k'\leq k$ and $\epsilon'\geq\epsilon$. Moreover, a $(n,m,k,\epsilon)$ LRF can easily be modified by deleting some output bits to yield a $(n,m',k,\epsilon)$ LRF for any $m'<m$. Conversely, a $(n,cm,k,\epsilon)$ LRF with key space $\mathcal{Z}^c$ can for $c>1$ be obtained from a $(n,m,k,\epsilon)$ LRF $\mathcal{F}_{\mathcal{Z}}$ with key space $\mathcal{Z}$ by a simple concatenation of $c$ copies of $\mathcal{F}_{\mathcal{Z}}$ with independent keys.

A second generalization of LRFs is necessary in order to be consistent with other asymptotic definitions in complexity theory, in particular those used in [9].

**Definition 2.** A *locally random function generator (LRFG)* with key length function $l(n)$ and degree of local randomization $k(n)$ is a family $\mathcal{F}=\{\mathcal{F}^n_{\{0,1\}^{l(n)}}: n\in\mathbb{N}\}$, where $\mathcal{F}^n_{\{0,1\}^{l(n)}}$ is a $(n,n,k(n),\epsilon(n))$ LRF with key space $\{0,1\}^{l(n)}$ that is (for every given argument and key) computable in time polynomial in $n$, independent of the number of previous evaluations, where $\epsilon(n)$ vanishes faster than $1/Q(n)$ for every polynomial $Q(n)$ (i.e., $1/\epsilon(n)$ is superpolynomial in $n$).

# 5. Complexity-theoretic Applications of Locally Random Functions

The construction of [9] for a PRPG is based on the following observation which can be formalized. Let $\mathcal{F}$ be a LRFG (or LRPG) with key length function $l(n)$ whose degree of local randomization $k(n)$ is superpolynomial in $n$ (which implies that $l(n)$ is superpolynomial in $n$). Let $\mathcal{F}'$ be the PRFG (or PRPG) resulting from $\mathcal{F}$ when the $n\cdot l(n)$ random bits are substituted by a function generator $\mathcal{G}$ generating the corresponding amount of (pseudorandom) bits. Then $\mathcal{F}'$ is a PRFG (or PRPG) under the assumption that $\mathcal{G}$ is a PRFG.

For PRFGs, this construction seems to be of little value because a PRFG is required for constructing another PRFG. For the case of a PRPG, however, this observation allows to relax the problem of constructing a PRPG to the two problems of constructing a PRFG and a LRFG with superpolynomial degree of local randomization that is also a permutation generator.

For the construction of a LRF described in Proposition 2, the evaluation time is proportional to the degree $k$ of local randomization because for every $x$, $f(x)$ depends on every internal random bit, i.e., on all the coefficients of the polynomial $p(x)$. A different construction for which every function value depends only on an negligible fraction of the key bits must hence be used for obtaining a superpolynomial degree $k$ of local randomization while retaining the polynomial evaluation time, as is required for a generalized interpretation of the Luby-Rackoff results. Before presenting such constructions we point out that the mapping $H : F^n \times F^n \times F^n \rightarrow P^{2n}$ gives a construction of LRFGs from other LRFGs.

**Theorem 1.** *Let $\mathcal{F}_i$ for $i = 1,2,3$ be three independent $(n,n,k,\epsilon_i)$ LRFs. $\psi(\mathcal{F}_1,\mathcal{F}_2,\mathcal{F}_3)$ is a $(2n,2n,k,\epsilon)$ LRF for all $k$, where $\epsilon = k^2 2^{-n} + \epsilon_1 + \epsilon_2 + \epsilon_3$.*

*Proof.* The fact that $\mathcal{F}_i$ is a $(n,n,k,\epsilon_i)$ LRF can be expressed as

$$\Big| P[g(f(x_1),\ldots,f(x_k))=1\colon f \in_R \mathcal{F}_i] \;-\; P_g \Big| \;\leq\; \epsilon_i \tag{5}$$

where $P_g$ is defined in (1) with $F^{2n}$ replaced by $F^n$. Since using a randomized strategy for distinguishing $\mathcal{F}_i$ from a random function cannot be better than using the best deterministic function $g$, as mentioned before, inequality (5) implies that

$$\Big| P[g(f(x_1),\ldots,f(x_k))=1\colon f \in_R \psi(F^n,F^n,F^n)] \\ -P[g(f(x_1),\ldots,f(x_k))=1\colon f \in_R \psi(\mathcal{F}_1,F^n,F^n)]\Big| \leq \epsilon_1$$

and

$$\Big| P[g(f(x_1),\ldots,f(x_k))=1\colon f \in_R \psi(\mathcal{F}_1,F^n,F^n)] \\ -P[g(f(x_1),\ldots,f(x_k))=1\colon f \in_R \psi(\mathcal{F}_1,\mathcal{F}_2,F^n)]\Big| \leq \epsilon_2$$

and

$$\Big| P[g(f(x_1),\ldots,f(x_k))=1\colon f \in_R \psi(\mathcal{F}_1,\mathcal{F}_2,F^n)] \\ -P[g(f(x_1),\ldots,f(x_k))=1\colon f \in_R \psi(\mathcal{F}_1,\mathcal{F}_2,\mathcal{F}_3)]\Big| \leq \epsilon_3.$$

The proof is completed by combining these three inequalities with Lemma 1 and observing that these four probability differences define four adjacent (but possibly overlapping) subintervals of $[0,1]$ whose total span can be at most the sum of the four interval lengths. $\square$

Note that the functions of $\psi(\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3)$ are actually permutations. Lemma 1 follows immediately as a corollary of this theorem since $F^n$ with key space $\{0,1\}^{n2^n}$ is an $(n, n, 2^n, 0)$ LRF.

Instead of implementing the functions $f_1, f_2, f_3$ in the Luby-Rackoff construction $H(f_1, f_2, f_3)$ directly as some pseudorandom functions, the construction $H$ can be applied iteratively. For instance, a pseudorandom permutation $f : \{0,1\}^{4n} \to \{0,1\}^{4n}$ can be implemented as $f = H(f_1, f_2, f_3)$ where $f_1 = H(f_{11}, f_{12}, f_{13})$, $f_2 = H(f_{21}, f_{22}, f_{23})$ and $f_3 = H(f_{31}, f_{32}, f_{33})$ and where $f_{ij}$ are pseudorandom functions $\{0,1\}^n \to \{0,1\}^n$. Let

$$H^{(s)} : (F^n)^{3^s} \to P^{2^s n}$$

be the $s$-fold iterative application of the Luby-Rackoff construction $H$ which requires $3^s$ functions $\{0,1\}^n \to \{0,1\}^n$ as inputs. The following Corollary to Theorem 1 gives a characterization of this iterative construction as a result on locally random functions.

**Corollary 1.** *When $H^{(s)}$ is applied to $3^s$ independent $(n, n, k, \epsilon)$ LRFs then the resulting function is a $(2^s n, 2^s n, k, \epsilon')$ LRF where $\epsilon' = k^2 \sum_{i=1}^{s} 3^{i-1} 2^{-2^{s-i} n} + 3^s \epsilon$.*

Although the mapping $H$ serves the originally intended purpose of proving an important complexity-theoretic result, randomness is used wastefully: The degree of local randomization is only on the order of the square root of the number of key bits. The best (in terms of efficient use of key bits) previously known asymptotic construction is the LRFG $\{\psi(F^n, F^n, F^n, F^n, F^n) : n \in \mathbb{N}\}$ with superpolynomial degree of local randomization and with key length function $l(n) = 5n2^n$, which was proved in [11] to have degree $k(n) = \Omega(2^{2n/3}) = \Omega(l(n)^{2/3})$ of local randomization.

In the following we present an alternative construction of LRFGs that achieves a local randomization of degree $k(n) = \Omega(l(n)^\alpha)$ for any $\alpha < 1$. Such LRFGs lead to alternative constructions of PRFGs and PRPGs based on PRFGs.

Let $d$ be a parameter of the following construction, let $r_i : \{0,1\}^t \to \{0,1\}^n$ for $1 \leq i \leq d$ be random functions, let $c = \lceil \log_2 d \rceil$ and let $P$ be a $(n + c, t, 2d)$ LRF. For example, $P$ could be implemented as $P(\xi) = [p(\xi)]_t$ (the $t$ least significant bits of $p(\xi)$), where $p$ is a polynomial $p(u) = p_{2d-1}u^{2d-1} + \cdots + p_1 u + p_0$ of degree $2d - 1$ over $GF(2^{n+c})$ and the key of $P$ consists of the $2d$ coefficients $p_{2d-1}, \ldots, p_o$. The total number of random bits required for implementing $r$ and $P$ is hence $l(n) = 2^t dn + 2d(n + c)$.

**Theorem 2.** *The family of functions $\mathcal{F}^{(n)} = \{f_z : z \in \{0,1\}^{l(n)}\}$ defined by*

$$f_z(x) \;=\; \sum_{i=0}^{d-1} r_i(P(x \bullet [i]_c)),$$

*where the sum is bit-wise modulo 2 and the $l(n) = 2^t dn + 2d(n+c)$ bits of $z$ are used in some manner to implement the functions $r_i$ and as the key of $P$, is a $(n, n, k, \gamma(k))$ LRF for all $k$, where $\gamma(k) = k^{d+1}2^{-dt}$.*

*Proof.* Let $a_{ij}$, $1 \le i \le d$, $1 \le j \le k$ be the input to function $r_i$ when $\mathcal{F}^{(n)}$ is evaluated for the $j$th argument $x_j$. Let $\mathcal{E}$ be the event that for every $x_j$, $1 \le j \le k$, there exists an $i_j$, $1 \le i_j \le d$, such that $a_{i_j j} \ne a_{i_j m}$ for all $m \ne j$. If $\mathcal{E}$ occurs, then for $1 \le j \le k$ at least one of the terms in the sum forming $f_z(x_j)$ is a random variable that is completely random and independent of all the other terms occurring in the evaluations of $f_z(x_1), \ldots, f_z(x_k)$, and hence $f_z(x_1), \ldots, f_z(x_k)$ are independent and completely random. The complementary event $\overline{\mathcal{E}}$ is the union over $1 \le j \le k$ and over $m_i \in \{1, \ldots, j-1, j+1, \ldots, k\}$ for $1 \le i \le d$ of the $k(k-1)^d$ events

$$\{a_{ij} = a_{im_i} \text{ for } 1 \le i \le d\}.$$

Because the $a_{ij}$ are $2d$-wise independent, each of these events has probability $2^{-dt}$ and hence

$$P[\overline{\mathcal{E}}] \;\le\; k^{d+1}2^{-dt}. \quad \square$$

The following argument demonstrates that the construction of Theorem 2 yields an asymptotically optimal locally random function. Let

$$\mathcal{G} = \{\mathcal{F}^{(n)} : n \in \mathbb{N}\}, \tag{6}$$

where $\mathcal{F}^{(n)}$ is the LRF from Theorem 2 with $d = t$ and $t$ is any function of $n$ such that $2^t/t$ is superpolynomial in $n$. For instance, $t(n) = \lceil (\log n)^{1+\delta} \rceil$ for some fixed $\delta > 0$. The key length function of $\mathcal{G}$ is

$$l(n) \;=\; 2t(n + \lceil \log_2 t \rceil) + 2^t tn \;=\; \Theta(2^t tn).$$

For

$$k(n) = 2^{t(t-1)/(t+1)}$$

we have $\gamma(k) = 2^{-t}$. It is straight-forward to prove that

$$\lim_{n\to\infty} \frac{l(n)^\alpha}{k(n)} = 0$$

for all $\alpha < 1$ and thus we have the following result.

**Corollary 2.** *$\mathcal{G}$ as defined in (6) is a LRFG with degree of local randomization $k(n) = \Omega(l(n)^\alpha)$ for any $\alpha < 1$.*

We suggest as an open problem to devise a LRFG with superpolynomial key length function $l(n)$ and local randomization of degree $k(n) = \Omega(l(n))$.

One can prove that even when $r_1, \ldots, r_d$ are taken to be the same random function $r : \{0,1\}^t \to \{0,1\}^n$ rather than independent random functions, the resulting family of functions is a LRF also satisfying $k(n) = \Omega(l(n)^\alpha)$ for any $\alpha < 1$.

**Theorem 3.** *The family of functions $\mathcal{F}^{(n)} = \{f_z : z \in \{0,1\}^{l(n)}\}$ defined by*

$$f_z(x) \;=\; \sum_{i=0}^{d-1} r(P(x \bullet [i]_c)),$$

*where the $l(n) = 2^t n + 2d(n+c)$ bits of $z$ are used in some manner to implement the function $r$ and as the key of $P$, is a $(n, n, k, \gamma(k))$ LRF, where $\gamma(k) = k^{d+1}(2d2^{-t})^d$, for all $k \geq \sqrt{2^t/d}$.*

## 6. Concluding Discussion

An important application of locally random functions is in the area of probabilistic algorithms, where randomness is often an expensive resource and therefore simulated by a pseudorandom generator with a random seed. In the analysis of a probabilistic algorithm that uses blocks of random bits at various stages it is sometimes sufficient to require that the random blocks be only $k$-wise independent rather than jointly independent. For instance, the "birthday paradox" holds not only for independent random birthdays but also when the birthdays are only pairwise independent. Furthermore, pairwise independence of a set of random variables is sufficient for proving that the variance of the sum of the random variables equals the sum of the variances (e.g., see [3]).

Local randomness has also several applications in cryptography. A first (in the author's opinion misinterpreted) cryptographic application is the design of cipher systems "provably secure" against enemies with unlimited computational resources. Schnorr [13] suggested to simulate the random keystream of the one-time pad by a keystream that is only locally random. If an eavesdropper can examine at most $k$ (arbitrarily chosen) bits of the keystream, where $k$ is the degree of local randomization, such a system offers the same perfect security as the one-time pad, even if the eavesdropper has infinite computing power. Of course, as is pointed out in [10] where Schnorr's idea is generalized, the drawback of such a system is that clearly $k$ cannot be greater than the length of the secret key (the seed) and thus the assumption that an eavesdropper cannot obtain more than $k$ keystream bits is generally completely unrealistic. Another example is the "provably secure" block cipher described in [14] which suffers from an even stronger weakness because the number of plaintext-cryptogram pairs an eavesdropper is allowed to obtain is upper bounded by only the square root of the key size. Loosely speaking, an enemy is guaranteed to spend at least

100 years breaking the cipher if the user of the system is willing to spend 10'000 years for only loading the secret key into the system. Clearly, if such a long secret key were available, the users would be better off using a one-time pad to begin with.

A related but much more important cryptographic application of local randomness is the design of conventional cryptographic algorithms using a secret key of only moderate size. The basic idea, which could be further formalized, is to design a system that uses an (impractically) large amount of secret random bits and to prove it secure against enemies with unlimited computational resources for a suitable definition of security. If the random bits are replaced by pseudorandom bits generated by a pseudorandom number generator or a pseudorandom function generator with only a short secret key, the system can clearly not retain its unconditional security. However, failure of this modified system to be *computationally* secure for the same definition of security implies failure of the pseudorandom (number or function) generator to be *computationally indistinguishable* from a random generator since any breaking algorithm for the cryptosystem would yield a distiguishing algorithm for the pseudorandom generator. Therefore, the modified system can be proved secure under the assumption that the component pseudorandom generators are secure. Although no pseudorandom generator has been proved secure, to rely on such an unproven assumption may be worth-while as it allows to clarify the principles on which a cipher's security is based.

The most trivial and widely used application of the described idea are conventional additive stream ciphers which can trivially be "proved" secure under the assumption that the keystream generator is a pseudorandom number generator according to [2]. Another less trivial application is for the design of block ciphers. A block cipher can be obtained from an efficiently computable locally random function by replacing the random bits by one or several pseudorandom function.

A further cryptographic application of local randomness may be for the key scheduling in secret-key ciphers where a relatively short key must be stretched to a sequence of subkeys (e.g. round keys of a block cipher).

## Acknowledgment

Jacques Patarin has independently found a different construction of a LRFG with superpolynomial key length function $l(n)$ and local randomization of degree $k(n) = \Omega(l(n)^\alpha)$ for any $\alpha < 1$.

# References


[1] N. Alon, O. Goldreich, J. Hastad and R. Peralta, Simple constructions of almost $k$-wise independent random variables, *Proceedings of the 31st IEEE Symposium on Foundations of Computer Science*, pp. 544-553, 1990.

[2] M. Blum and S. Micali, How to generate cryptographically strong sequences of pseudo-random bits, *SIAM Journal on Computing*, Vol. 10, pp. 96-113, 1981.

[3] B. Chor and O. Goldreich, On the power of two-point based sampling, *Journal of Complexity*, Vol. 5, No. 1, pp. 96-106, 1989.

[4] D.E. Denning, *Cryptography and Data Security*, Addison-Wesley, Reading, MA, 1983.

[5] O. Goldreich, S. Goldwasser and S. Micali, How to construct random functions, *Journal of the Association for Computing Machinery*, Vol. 33, pp. 792-807, 1986.

[6] A. Joffe, On a set of almost deterministic $k$-independent random variables, *The Annals of Probability*, Vol. 2, No. 1, pp. 161-162, 1974.

[7] H.O. Lancaster, Pairwise statistical independence, *Ann. Math. Statist.*, Vol. 36, pp. 1313-1317, 1965.

[8] L.A. Levin, One-way functions and pseudorandom generators, *Proc. 17th ACM Symposium on Theory of Computing*, pp. 363-364, 1985.

[9] M. Luby and C. Rackoff, How to construct pseudorandom permutations from pseudorandom functions, *SIAM Journal on Computing*, Vol. 17, No. 2, pp. 373-386, 1988.

[10] U.M. Maurer and J.L. Massey, Local randomness in pseudo-random sequences, *Journal of Cryptology*, Vol. 4, No. 2, pp. 135-149, 1991.

[11] J. Patarin, Etude des générateurs de permutations basés sur le Schéma du D.E.S., Ph. D. Thesis, INRIA, Domaine de Voluceau, Le Chesnay, France, 1991. An extract appeared in: J. Patarin, New results on pseudorandom permutation generators based on the DES scheme, *Advances in Cryptology – CRYPTO'91*, J. Feigenbaum (Ed.), Lecture Notes in Computer Science, Vol. 576, Springer-Verlag, pp. 301-312, 1992.

[12] J. Pieprzyk, How to construct pseudorandom permutations from single pseudorandom functions, *Advances in Cryptology – EUROCRYPT'90*, I.B. Damgård (Ed.), Lecture Notes in Computer Science, Vol. 473, Springer-Verlag, pp. 140-150, 1991.

[13] C.P. Schnorr, On the construction of random number generators and random function generators, *Advances in Cryptology – EUROCRYPT'88*, C.G. Günther (Ed.), Lecture Notes in Computer Science, Vol. 330, Springer-Verlag, pp. 225-232, 1988.

[14] Y. Zheng, T. Matsumoto and H. Imai, Impossibility and optimality results on constructing pseudorandom permutations, *Advances in Cryptology - EUROCRYPT'89*, J.-J. Quisquater et al. (Eds.), Lecture Notes in Computer Science, Vol. 434, Springer-Verlag, pp. 412-421, 1990.