Appeared in *Journal of Cryptology*, vol. 4, No. 2, pp. 135-149, 1991.

Local Randomness in Pseudo-random Sequences [1]

Ueli M. Maurer and James L. Massey

Institute for Signal and Information Processing
Swiss Federal Institute of Technology
CH-8092 Zürich, Switzerland

Abstract

The concept of provable cryptographic security for pseudo-random number generators that was introduced by Schnorr is investigated and extended. The cryptanalyst is assumed to have infinite computational resources and hence the security of the generators does not rely on any unproved hypothesis about the difficulty of solving a certain problem, but rather relies on the assumption that the number of bits of the generated sequence the enemy can access is limited. The concept of perfect local randomness of a sequence generator is introduced and investigated using some results from coding theory. The theoretical and practical cryptographic implications of this concept are discussed. Possible extensions of the concept of local randomness as well as some applications are proposed.

Keywords: pseudo-random number generator, provable security, local randomness.

[1] A preliminary version of this paper was presented at CRYPTO'89.

1. Introduction

It is well-known that beyond its unicity distance every cipher can in principle be broken, e.g., by an exhaustive key search, which is infeasible except for very simple ciphers. The aim of the designer of a cryptosystem is to make it secure against every attack that is practically feasible. Usually feasibility is specified by computation time, but it is conceivable that an attacker is limited by other restrictions, for instance, by his available storage capacity, by the number of ciphertext bits that he can obtain in a ciphertext-only attack (which is exactly the restriction considered by Ozarow and Wyner [15] in their recent treatment of the wire-tap channel), or by the number of bits of plaintext that he can obtain for a known-plaintext attack. Our results in Section 2 can be interpreted as showing that provably-secure (against a ciphertext-only attack) ciphers can be constructed under the restriction that the number of ciphertext bits obtainable by the enemy is smaller than the length of the key, divided by the logarithm of the ciphertext length, even when the enemy has complete freedom as to choose the locations within the ciphertext of the bits to which he has access. [To arrive at this interpretation, the output sequence of the pseudo-random number generator of Section 2 is taken as the running key in an additive stream cipher whose secret key is the "random seed" of the generator. The running key is added bit-by-bit modulo-two to the plaintext to produce the ciphertext.] This "limited ciphertext restriction", to whose formulation we were led by the work of Schnorr [19], is inappropriate for most practical applications and is much stronger than a suitable restriction on computation time. However, inasmuch as no provably-secure practical cipher has yet been devised for a computation-time restriction, the construction of provably-secure stream ciphers even for the strong limited-ciphertext restriction appears to be of interest.

Schnorr [19] presented a pseudo-random number generator whose security does not rest on any unproven (albeit plausible) assumptions, in contrast to most other proposed pseudo-random number generators [2, 3, 14]. Schnorr's generator stretches a random seed of length $k = m2^m$ to a pseudo-random sequence of length $n = 2m2^{2m}$, which cannot be distinguished from a random sequence by any statistical test that examines at most $e = 2^{m/3-(\log_2 m)^2}$ bits, even using infinite computational resources. [By a "random binary sequence" of length k we shall always mean a sequence of k binary random variables that takes on all 2^k possible values, each with probability 2^{-k}.] The length of the seed is roughly squared in this expansion, i.e., $n \approx k^2$, and the number $e+1$ of bits that must be examined by a distinguishing statistical test is roughly the third root of the seed length, i.e., $e \approx \sqrt[3]{k}$, which is very small from a cryptanalytic point of view. The generators constructed in this paper are superior to Schnorr's in two respects: the parameter e is on the order of $k/\log_2 n$ rather than only $\sqrt[3]{k}$ and the generated sequences are truly locally random rather than only (according to Schnorr's definition) locally indistinguishable from a random sequence. Rueppel [18] has pointed out the weakness of Schnorr's generator when the enemy is allowed to access k bits rather than e bits, but our interest is not in the practical security of Schnorr's generators. Rather, our interest is in exploring the theoretical questions raised in [19] from a somewhat different viewpoint.

In Section 2, we introduce the concept of a perfect local randomizer, i.e., of a sequence generator that stretches a (binary) random sequence of length k to a pseudo-random sequence of length n such that every subset of e or less bits of the generated sequence is a set of independent random bits. The concept of a perfect local randomizer corresponds to what is known in combinatorics as an orthogonal array. We use many results from coding theory to obtain explicit constructions of perfect local randomizers and to prove bounds on the achievable degree of perfect local randomization. We show that, for any choice of k, n and e satisfying $e \leq k/\log_2 n$, there exist perfect local randomizers. A topic closely related to perfect local randomization is the generation of so-called k-wise independent random variables, which was originally introduced in [8] and later also treated in [1] and [11]. The special case of pairwise independence is treated in [4], [7] and [10]. Recent theoretical interest in these schemes was motivated by their application in the construction of deterministic polynomial time algorithms from probabilistic ones for certain problems.

In the complexity-theoretic approach to pseudo-random number generation, a pseudo-random number generator is defined to be a family of sequence generators indexed by the security parameter k ($k = 1, 2, \ldots$) that stretch a sequence of k random input bits into a pseudo-random sequence of length $n(k)$ where $n(k)$ is a polynomial in k. In Section 3, we show that, for every integer t, every function $n(k)$ with $n(k) \leq k^t$ for all but finitely many k, and every $\epsilon > 0$, there exist pseudo-random number generators that stretch k-bit seeds into $n(k)$-bit sequences with the property that no statistical test (regardless of its computation time) examining not more than $e(k) = \lfloor (1-\epsilon)k \rfloor$ bits can distinguish them from random sequences. We also show that $e(k) > k$ is not achievable, thereby giving tight lower and upper bounds on the achievable $e(k)$. However, we are unable to show that the stretching function of any of these generators is computable in time polynomial in k. Our argument is a "random-coding" argument similar to that used by Shannon [20] to prove the existence of error-correcting codes with rate arbitrarily close to channel capacity that achieve an arbitrarily small block error probability, without demonstrating specific such codes. In other words, we prove the existence of pseudo-random number generators that achieve the maximum possible local randomization without presenting efficiently-computable examples. However, the linear sequence generators considered in Section 2 are easily extended to polynomial-time-computable pseudo-random number generators that achieve a local randomization of $e(k) = \lfloor k/\log_2 n(k) \rfloor$ bits rather than $e(k) = \lfloor (1-\epsilon)k \rfloor$ bits.

The restriction that Schnorr [19] puts on statistical tests, namely, that they can operate on at most a certain number of bits of the generated sequence, appears to be more information-theoretic than complexity-theoretic. This fact suggests generalizing the restriction in the following way: assume the enemy is allowed to obtain e arbitrary bits of information about the generated sequence, i.e., he is not restricted to acquiring information by examining binary digits but can, for example, obtain the value of an arbitrary random variable that does not give more than e bits of information about the sequence. Somewhat surprisingly, it turns out that under this looser restriction on the enemy's obtainable information, even for arbitrarily small e, "local" randomization cannot be achieved, as is shown in Section 4. The quotation marks here emphasize the fact that the accessed information may in this model very well be global, but its amount is limited. Similarly, if the enemy is able to obtain e arbitrary parity checks (modulo-two sums) on the sequence bits, perfect "local" randomization is shown to be impossible. Thus the results of this paper (as well as Schnorr's result) strongly rely on the assumption that the enemy's information about the sequence consists of knowing some subset of the digits in the sequence.

In Section 5, we suggest two possible applications of the proposed sequence generators. They might be excellent building blocks within practical ciphers for spreading local (pseudo-) randomness when used together with compressing transformations that guarantee confusion, and they are certainly of use wherever a secret key must be expanded (for example, in key scheduling within block ciphers).

2. Sequence Generators Achieving Perfect Local Randomness

Unlike in the literature based on, or motivated by, complexity theory, including [19], we consider in this section individual sequence generators of specific size, rather than infinite families of generators. The asymptotic case is treated in Section 3. Let I_n denote the set of binary sequences of length n, i.e., $I_n = \{0,1\}^n$. A random variable which takes on two values, both with probability 1/2, will be called a *coin-tossing random variable*, abbreviated CTRV. Throughout the whole paper, $\ln x$ and $\log x$ denote the natural logarithm and the logarithm to the base 2 of x, respectively.

Definition 1: A (k,n) *sequence generator* G is a function $G : I_k \longrightarrow I_n : \underline{z}^k \mapsto \underline{s}^n = G(\underline{z}^k)$.

Note that a (k,n) sequence generator can be interpreted as the encoder of a binary block code with 2^k codewords of length n, where we think of the randomly-selected key bits as forming the k information bits.

Definition 2: A (k,n) sequence generator G is a (k,n,e) *perfect local randomizer (PLR)* if, when the input is a sequence of k independent CTRV's, then every subset of e of the n binary output random variables is a set of e independent CTRV's. The degree of perfect local randomness of a (k,n) sequence generator G is $\max\{e : G \text{ is a } (k,n,e) \text{ PLR}\}$.

It is obvious that there exists no (k,n,e) PLR for $e > k$. For $e = k$ and $n > k$, PLR's exist in only two trivial cases: $k = 1$ (example: repeat the input bit n times) and $k = n-1$ (example: the first $n-1$ bits are the input bits and the last bit is their modulo-two sum).

Definition 3: A (k,n) sequence generator is *linear* if and only if, for all $\underline{z}_1^k, \underline{z}_2^k \in I_k$, $G(\underline{z}_1^k \oplus \underline{z}_2^k) = G(\underline{z}_1^k) \oplus G(\underline{z}_2^k)$, where $\oplus$ denotes bitwise addition modulo 2.

A linear (k,n) sequence generator can be interpreted as an encoder for a linear binary code and can be specified by the binary $k \times n$ matrix $\mathcal{G}$ such that

$$\underline{s}^n = \underline{z}^k \mathcal{G}.$$

The matrix $\mathcal{G}$ is usually called the *generator matrix* in coding theory. The following Lemma restates a well-known result in coding theory (viz., the condition for the digits in some selected e positions to be choosable as a subset of the k information digits) in terms of PLR's.

Lemma: *A linear (k,n) sequence generator G is a (k,n,e) perfect local randomizer if and only if every subset of e columns of $\mathcal{G}$ are linearly independent.*

Proof: If there exists a set of e columns of $\mathcal{G}$ that are linearly dependent, then the corresponding e bits of $\underline{s}^n$ satisfy a linear equation and are therefore not independent. Conversely, consider the submatrix formed by any set of e linearly independent columns.

One can extend this $k \times e$ submatrix to a nonsingular binary $k \times k$ matrix A by appending $k-e$ appropriate columns. If $\underline{z}^k$ is a sequence of independent CTRV's, then so also is $\underline{z}^k A$ and hence, trivially, the first e components of $\underline{z}^k A$ are independent CTRV's. $\square$

Theorem 1: *There exist linear (k, n, e) perfect local randomizers if*

$$e \;\leq\; \frac{k}{\log n} \;,$$

or if $h[(e-1)/(n-1)] < k/(n-1)$ and $e \leq (n+1)/2$. There exists no (k, n, e) linear or nonlinear PLR if $h[(e-1)/2n] \geq (k + \frac{1}{2}\log n + \frac{1}{2})/n$, which is satisfied when

$$e \;>\; 2\,\frac{k + \log n + 1}{\log n - \log(k/2)} \;,$$

where $h(x) = -x\log x - (1-x)\log(1-x)$ is the binary entropy function.

Proof: Consider a binary $k \times (n-1)$ matrix $\mathcal{G}'$ such that every subset of $e-1$ columns of $\mathcal{G}'$ is a set of linearly independent columns. The number of k-vectors that are a linear combination of at most $e-1$ columns of $\mathcal{G}'$ is upper bounded by $\sum_{i=0}^{e-1}\binom{n-1}{i}$. If this set of linear combinations does not exhaust the set of 2^k k-vectors, we can find a non-zero column that can be adjoined to $\mathcal{G}'$ to obtain a $k \times n$ matrix $\mathcal{G}$ for which every subset of e columns is linearly independent. Hence, according to the above lemma, a linear (k, n, e) PLR surely exists if

$$\sum_{i=0}^{e-1}\binom{n-1}{i} \;<\; 2^k.$$

The existence of (k, n, e) PLR's for $e \leq k/\log n$ now follows immediately from the fact that $\sum_{i=0}^{e-1}\binom{n-1}{i} < n^e$ (for $n > 1$). In the following, we will make use of the inequalities (see [22], inequalities A.24 and A.30)

$$\frac{1}{\sqrt{2n}}2^{nh(t/n)} \;\leq\; \binom{n}{t} \;\leq\; \sum_{i=0}^{t}\binom{n}{i} \;\leq\; 2^{nh(t/n)}, \tag{1}$$

where the last inequality holds for $t \leq n/2$. Letting $t = e-1$ and replacing n by $n-1$ in the last inequality proves the existence of linear (k, n, e) PLR's when $(n-1)h[(e-1)/(n-1)] < k$ and $(e-1)/(n-1) \leq 1/2$, which completes the proof of the first part of Theorem 1.

In order to prove the non-existence claim of Theorem 1 for linear (k, n, e) PLR's, we consider the number Q of all linear combinations of $\lfloor e/2 \rfloor$ or fewer columns of $\mathcal{G}$. If

$$Q \;=\; \sum_{i=1}^{\lfloor e/2 \rfloor}\binom{n}{i} \;\geq\; 2^k, \tag{2}$$

then either there exists a linear combination of $\lfloor e/2 \rfloor$ or fewer columns that equals the all-zero column or there exist at least two different linear combinations of $\lfloor e/2 \rfloor$ or fewer columns that are equal, and hence there exists a linear combination of e or fewer columns that equals the all-zero column. Thus, satisfaction of inequality (2) implies that there exists no linear (k, n, e) PLR. That (2) also implies the nonexistence of nonlinear (k, n, e) PLR's is equivalent to a result proved in [5] and called the uniform projection lemma. From (1), it follows that inequality (2) is satisfied if

$$\frac{1}{\sqrt{2n}}2^{nh(\lfloor e/2 \rfloor/n)} \;>\; 2^k$$

and thus also if

$$h\left(\frac{e-1}{2n}\right) \;>\; \frac{k+\frac{1}{2}\log n+\frac{1}{2}}{n}\,, \tag{3}$$

as can easily be verified. To complete the proof of Theorem 1, we note for $0 < x < 1$ that $h(x) \geq -x\log x$. Since $-\log[(e-1)/2n] > -\log(k/2n) = \log n - \log(k/2)$, inequality (3) is satisfied if $(e-1)(\log n - \log(k/2)) > 2k + \log n + 1$ and thus also if $e(\log n - \log(k/2)) \geq 2k + 2\log n + 1 - \log(k/2)$. Because $k \geq 1$ and thus $\log(k/2) \geq -1$, the non-existence of (k, n, e) PLR's is established when the last inequality of Theorem 1 is satisfied. □

Remarks: A well-known fact about linear codes (see [12], Chapter 1, Theorem 10) is that, given any parity-check matrix for the code, the minimum distance equals the minimum positive integer d such that there exists a set of d columns in the parity-check matrix of the code that are linearly dependent or, equivalently, the maximum d such that every subset of $d-1$ columns are linearly independent. By definition, a parity-check matrix for a linear code is the encoding matrix of the dual code. From the above lemma, we conclude that the degree e of perfect local randomness of a linear (k, n) sequence generator is one less than the minimum distance d of the dual code to the code encoded by this generator. This dual code is a linear code with dimension $n-k$, i.e., a $[n, n-k, d]$ linear code with 2^{n-k} codewords (see [12], p. 9). In other words, every linear (k, n, e) PLR is an encoder of the dual of a linear $[n, n-k, e+1]$ code and conversly, every encoder of the dual of a linear $[n, n-k, d]$ code is a $(k, n, d-1)$ PLR. Note that the existence proof given here for linear (k, n, e) PLR's amounts to a proof of the well-known Gilbert-Varshamov existence bound for linear codes (see [12], Ch. 1, Theorem 12), which states that given n and k there exists a binary linear $[n, k, d]$ code if $\sum_{i=0}^{d-2}\binom{n-1}{i} \;<\; 2^{n-k}$. More generally, every bound on the minimum distance of linear $[n, n-k, d]$ codes can directly be transformed into a bound on the degree of perfect local randomness of linear (k, n) sequence generators. The best table known to the authors of minimum distances achievable with linear codes is that of [21].

Although the problem of determining the maximal achievable degree of perfect local randomness of any linear (k, n) sequence generator is equivalent to the problem of determining the maximal achievable minimum distance d of a linear binary $[n, n-k, d]$ code, the corresponding two problems for nonlinear PLR's and codes are not equivalent and much less is known about the first of them. It is therefore somewhat surprising that the Hamming bound, a well-known upper bound on the achievable minimum distance of a code (linear or nonlinear) with $n-k$ information bits and codeword length n, is correspondingly valid for the maximal degree e of perfect local randomness of any (k, n, e) PLR. The Hamming (or sphere packing) bound (see [12], Ch. 1, Theorem 6), which follows from the fact that all spheres of radius $(d-1)/2$ (the number of errors guaranteed to be correctable by the code) must be disjoint, states that there exists no binary code with 2^{n-k} codewords of length n and minimum distance d if

$$\sum_{i=0}^{\lfloor (d-1)/2 \rfloor} \binom{n}{i} \;>\; 2^k\ .$$

Because $e = d-1$, this bound is equivalent to the bound (2) on the maximal degree of perfect local randomness of a (k, n, e) PLR, although the latter was obtained in a different way that applies only for linear PLR's. It is an open problem to find a stronger upper bound on the achievable degree of perfect local randomness, for instance, one equivalent in strength to the McEliece-Rodemich-Rumsey-Welch upper bound [13] on the achievable minimum distance of a code, which is significantly better than the Hamming bound. Clearly any upper bound on the minimum distance d gives an upper bound on the degree of perfect local randomness e that can be achieved by linear PLR's, the interesting question is whether the same bound applies to nonlinear PLR's as well.

Note that the lower and upper bounds on the achievable degree e of perfect local randomness given in Theorem 1 differ by a factor of approximately 2 when $\log k << \log n << k$, which is the situation of greatest interest.

Theorem 1 gives an existence bound for good linear PLR's. Although the proof of the Gilbert-Varshamov bound is in principle constructive, its application for finding good PLR's for general k and n requires computation time exponential in k and n. The following theorem exhibits an infinite polynomial-time constructable class of linear (k, n, e) PLR's for which $e > k/\log n$, i.e., whose degree of perfect local randomness is approximately equal to the value guaranteed by the Gilbert-Varshamov lower bound.

Theorem 2: *The encoder of an extended Reed-Solomon code over $GF(2^m)$ with e information symbols, codeword length 2^m and design distance $2^m - e + 1$ is a linear $(me, m2^m, e)$ perfect local randomizer when the symbols are appropriately represented by m binary digits.*

Proof: Extended Reed-Solomon codes over $GF(2^m)$ (see [12]) are *maximum distance separable*, i.e., every subset of e codeword digits may be chosen as the e information digits. By appropriately representing every digit of $GF(2^m)$ as a binary m-tuple, the Reed-Solomon code becomes a binary linear code with $k = me$ information bits and codeword length $n = m2^m$ such that, for random information bits, every subset of e m-bit blocks of the codeword is random. Thus certainly every subset of e bits is random. □

Remark: The maximum-distance-separable property of Reed-Solomon codes derives from the fact that any $k \times n$ generator matrix $\mathcal{G}$ is such that every k columns of $\mathcal{G}$ form a Vandermonde matrix. Other authors have noted the usefulness of the properties of Vandermonde matrices [4, 8] or of BCH codes [1] in connection with k-wise independence of random variables.

The PLR's of Theorem 2 can be compared fairly with the pseudo-random number generator suggested by Schnorr [19] since the parameters k and n can be chosen to coincide. Schnorr's generator is a family of $(m2^m, 2m2^{2m})$ sequence generators (m is the security parameter) such that no test examining at most $2^{m/3-(\log m)^2}$ bits can distinguish the output sequence from a random sequence. An extended Reed-Solomon code over $GF(2^{2m})$ with 2^{m-1} information symbols corresponds to a $(m2^m, 2m2^{2m}, 2^{m-1})$ PLR that not only achieves true local randomness instead of only indistinguishability from randomness but also gives a degree of perfect local randomness greater than the third power of that guaranteed by Schnorr. The smallest value of m for which Schnorr's lower bound is nontrivial is $m = 162$ where the number of bits that must be examined by a distinguishing statistical test is $e = 2$ (out of approximately 10^{100} bits), compared to $e = 2^{161}$ for the Reed-Solomon code. (This example illustrates that the practical significance of asymptotic results in cryptology must always be carefully evaluated.)

In the following, we discuss nonlinear perfect local randomizers. A (k, n, e) PLR (linear or not) is the encoder of a binary block code with 2^k codewords such that for every subset of e positions, every e-bit pattern occurs exactly 2^{n-e} times. Such a configuration is also known as an *orthogonal array* [17] of size 2^k, n constraints, 2 levels, strength e and index 2^{n-e}. As in the earlier treatment of linear PLR's, some results from coding theory can be applied in the nonlinear case. MacWilliams ([12], Chapter 5) introduced a transform for the distance distribution of a code that yields, for linear codes, the distance (or, equivalently, the weight) distribution of the dual code. The significance of the transform of the distance distribution of a nonlinear code is not obvious since there exists no dual code for a nonlinear code. However, surprisingly enough, if one defines the dual distance d' of a code as the minimum distance value for which the transformed distance distribution is not zero, then one obtains precisely what we are looking for: the degree of perfect local randomness of an encoder for the code considered as a sequence generator is $e = d' - 1$. This remarkable result is due to Delsarte [6].

The question whether for large k and n there exist nonlinear (k, n) sequence generators whose degree of perfect local randomness is greater than that of every linear (k, n) sequence generator is open. However, there do exist some nonlinear PLR's superior to the best linear PLR's. The so-called Kerdock codes $\mathcal{K}(m)$ are $(2^m, 2^{2m}, 2^{m-1} - 2^{(m-1)/2})$ nonlinear codes for all even $m \geq 4$ that yield $(2m, 2^m, 5)$ nonlinear PLR's as shown by determining the dual distance d' of these nonlinear codes (see [12], Ch. 15, Theorem 24 and Corollary 29). The so-called punctured Preparata codes $\mathcal{P}(m)^*$ similarly yield $(2^m - 2m, 2^m - 1, 2^{m-1} - 2^{m/2-1})$ nonlinear PLR's for all even $m \geq 4$ (see [12], Ch. 15, Theorem 32). The Delsarte-Goethals codes $\mathcal{DG}(m, d)$ with $d = (m-2)/2$ yield $(3m-1, 2^m, 7)$ nonlinear PLR's for all even $m \geq 4$ (see [12], pp. 476-477). Thus, $\mathcal{K}(4)$, $\mathcal{K}(6)$, $\mathcal{K}(8)$, $\mathcal{P}(6)^*$, $\mathcal{P}(8)^*$, $\mathcal{DG}(4,1)$ $\mathcal{DG}(6,2)$ and $\mathcal{DG}(8,3)$ are $(8,16,5)$, $(12,64,5)$, $(16,256,5)$, $(52,63,27)$, $(240,255,119)$, $(11,16,7)$, $(17,64,7)$ and $(23,256,7)$ nonlinear PLR's, respectively. ¿From the table in [21], we conclude that the best linear $(8,16,e)$, $(12,64,e)$, $(52,63,e)$, $(11,16,e)$ and $(17,64,e)$ PLR's satisfy $e = 4$, $4 \leq e \leq 5$, $25 \leq e \leq 26$, $e = 7$ and $5 \leq e \leq 7$, respectively. The $(8,16,5)$ PLR (also known as the Nordström-Robinson code) and the $(52,63,27)$ PLR thus beat the best linear PLR's with the same k and n. It is unknown to the authors whether $\mathcal{K}(m)$, $\mathcal{P}(m)^*$ and $\mathcal{DG}(m,(m-2)/2)$ are superior to the best linear PLR's for infinitely many m, or for all $m \geq 2$, $m \geq 2$, and $m \geq 3$, respectively.

3. Locally-Randomized Pseudo-random Number Generators

Section 2 was devoted to sequence generators that stretch, for fixed k and n, a k-bit secret random key to an n-bit sequence. Since the framework of complexity theory is based on the analysis of asymptotic behavior, a *pseudo-random number generator* G is often defined [3, 19] as an infinite class $G = \{G_k : k \geq 1\}$ of $(k, n(k))$ sequence generators G_k, where n is a polynomial function of the index k and where the computation time of each sequence generator is upper bounded by a polynomial function of k. Similarly, a *statistical test* $S^G = \{S_k^G : k \geq 1\}$ for the pseudo-random number generator G [24] is an infinite class of probabilistic algorithms S_k^G which take as input a binary sequence of length $n(k)$ and emit a binary output. G is said to pass the statistical test S^G if and only if, for all polynomials P and for all but a finite number of integers k,

$$\left| p_k^{S^G,G} - p_k^{S^G,R} \right| < \frac{1}{P(k)},$$

where $p_k^{S^G,G}$ denotes the probability that S_k^G emits a 1 if the input is the sequence generated by G_k for a random k-bit input, and where $p_k^{S^G,R}$ denotes the probability that S_k^G emits a 1 if the input is a random sequence of length $n(k)$.

Definition 4: Let $e(k)$ be any positive integer-valued function. We shall call a pseudo-random number generator G *degree* $e(k)$ *locally-randomized* if G passes every (not necessarily time-bounded) statistical test that examines not more than $e(k)$ of the $n(k)$ bits.

Corollary to Theorem 1: *Let t be any positive integer. For any function $n(k)$ satisfying $n(k) \le k^t$ for all but finitely many k, there exist degree $e(k) = \lfloor k/(t \log k) \rfloor$ locally-randomized pseudo-random number generators.*

Proof: The corollary is an immediate consequence of Theorem 1 and the fact that a k-bit row vector can be multiplied by a binary $k \times n(k)$ matrix in time polynomial in k.

Theorem 3: *There exist no degree $e(k)$ locally-randomized pseudo-random number generators having $e(k) > k$ for infinitely many k.*

Proof: Since the statistical test need not be time-bounded, it can compute any function $I_{e(k)} \longrightarrow \{0,1\}$. Let the test's output be 1 if and only if the first $e(k)$ bits agree with the corresponding bits of one of the at most 2^k sequences that can be generated by the pseudo-random number generator. Then $p_k^{S^G,G} = 1$. But all $2^{e(k)}$ possible values of the first $e(k)$ bits are equally likely when $\underline{s}^{n(k)}$ is a random sequence so that $p_k^{S^G,R} \le 2^{k-e(k)}$, since S_k^G outputs 1 for at most 2^k input values. For $e(k) > k$, $p_k^{S^G,R} \le 0.5$ and thus $|p_k^{S^G,G} - p_k^{S^G,R}| \ge 0.5$. □

The following theorem shows that the degree of perfect local randomness can be arbitrarily close to the upper bound k. However, the existence proof is non-constructive since it is based on a random coding argument, and therefore the polynomial-time computability of the generator cannot be guaranteed.

Theorem 4: *Let t be any positive integer. For every $\epsilon > 0$ and for any function $n(k)$ satisfying $n(k) \le k^t$ for all but finitely many k, there exist degree $e(k) = \lfloor (1-\epsilon)k \rfloor$ locally-randomized, not necessarily polynomial-time computable, pseudo-random number generators.*

Proof: A $(k, n(k))$ sequence generator can be considered to be an ordered list of 2^k binary sequences of length $n(k)$. We will show that if $n(k)$ is upper bounded by a polynomial in k and if $P(k)$ is any polynomial in k, then, for sufficiently large k and for virtually all of the $2^{2^k n(k)}$ $(k, n(k))$ sequence generators, the best (not time-bounded) statistical test not examining more than $e(k) = \lfloor (1-\epsilon)k \rfloor$ bits achieves a distinguishing probability $|p_k^{S^G,G} - p_k^{S^G,R}|$ smaller than $1/P(k)$. Hence, there exists a degree $\lfloor (1-\epsilon)k \rfloor$ locally-randomized pseudo-random number generator if the polynomial time computability is not required.

The statistical test S_k^G consists of a possibly probabilistic and adaptive strategy for determining a set of $e(k)$ observed bit positions and of a possibly probabilistic function $f^{G_k} : I_{e(k)} \longrightarrow \{0,1\}$ that assigns to every bit pattern $u \in I_{e(k)}$ the value 1 with probability

p_1^u and the value 0 with probability $1 - p_1^u$. Hence,

$$\left|p_k^{S^G,G} - p_k^{S^G,R}\right| = \left|\sum_{u \in I_{e(k)}} m_u 2^{-k} p_1^u - \sum_{u \in I_{e(k)}} 2^{-e(k)} p_1^u\right| = 2^{-k}\left|\sum_{u \in I_{e(k)}} (m_u - 2^{k-e(k)}) p_1^u\right| ,$$

where m_u is the number of sequences of G_k having the pattern u in the $e(k)$ positions selected by S_k^G. Obviously $\sum_{u \in I_{e(k)}} m_u = 2^k$. The term $|p_k^{S^G,G} - p_k^{S^G,R}|$ is maximized by choosing $p_1^u = 1$ for those u where $m_u - 2^{k-e(k)} \geq 0$ and by choosing $p_1^u = 0$ for all remaining u. (Note that this choice corresponds to a deterministic function f^{G_k}.) Therefore, for every function $f^{G_k} : I_{e(k)} \longrightarrow \{0,1\}$,

$$\left|p_k^{S^G,G} - p_k^{S^G,R}\right| \;\leq\; 2^{-k} \sum_{u \in I_{e(k)} : m_u \geq 2^{k-e(k)}} (m_u - 2^{k-e(k)}).$$

Consider now the random experiment of randomly selecting a $(k, n(k))$ sequence generator G_k, i.e., of randomly selecting 2^k binary sequences of length $n(k)$. In the following, we upper bound the probability that the selected generator G_k contains a set of $e(k)$ positions such that there exists a function f^{G_k} operating on these $e(k)$ positions whose distinguishing probability $|p^{f^{G_k},G_k} - p^{f^{G_k},R}|$ is greater than $2^{-\delta k}$ for a given $\delta > 0$. This event is the union over all $\binom{n(k)}{e(k)}$ possibilities of selecting $e(k)$ positions of the events that such a function exists for a particular set of $e(k)$ positions, e.g., the first $e(k)$ bits. Thus, the union bound yields

$$\begin{aligned} P[|p^{f^{G_k},G_k} - p^{f^{G_k},R}| \geq 2^{-\delta k}] \;&\leq\; \binom{n(k)}{e(k)} P\left[2^{-k} \sum_{u \in I_{e(k)} : m_u \geq 2^{k-e(k)}} (m_u - 2^{k-e(k)}) \geq 2^{-\delta k}\right] \\ &\leq\; \binom{n(k)}{e(k)} P\left[m_u - 2^{k-e(k)} \geq 2^k 2^{-\delta k}/2^{e(k)} \text{ for some } u \in I_{e(k)}\right] \\ &\leq\; \binom{n(k)}{e(k)} 2^{e(k)} P\left[m_{\underline{0}} \geq 2^{k-e(k)}(1 + 2^{-\delta k})\right]. \end{aligned} \tag{4}$$

where here m_u is the number of sequences having the pattern u in the first $e(k)$ positions and where $\underline{0}$ denotes the all-zero pattern. The second inequality follows from the fact that if the sum of $2^{e(k)}$ positive numbers is greater or equal to S, then at least one of them has to be greater or equal to $S/2^{e(k)}$; and the third inequality results from another application of the union bound: the events $m_u \geq 2^{k-e(k)}(1+2^{-\delta k})$ are equiprobable for all $2^{e(k)}$ values of u and thus the probability that at least one of these events occurs is upper bounded by $2^{e(k)}$ times the probability that any particular one of these events occurs, for example, the probability that $m_{\underline{0}} \geq 2^{k-e(k)}(1 + 2^{-\delta k})$. Since the events that the first $e(k)$ bits of the i-th sequence of G_k are all zero are independent and equiprobable for $1 \leq i \leq 2^k$, we can apply the Chernoff bound. From [22], inequality A.19, we obtain

$$P[m_{\underline{0}} \geq 2^{k-e(k)}(1 + 2^{-\delta k})] \leq e^{-2^k X}, \qquad \text{where} \qquad X = p \ln \frac{p q_0}{p_0 q} - \ln \frac{q_0}{q} \tag{5}$$

and where $p_0 = 2^{-e(k)}$, $p = (1 + 2^{-\delta k}) p_0$, $q_0 = 1 - p_0$ and $q = 1 - p$. Rewriting X and using $\ln(1+y) \geq y - \frac{2}{3}y^2$ for $y \geq 0$, $\ln(1-y) \leq -y$ for $y < 1$ and $\ln(1-y) \geq -y - y^2$ for

$y \leq 0.5$, we find for $2^{-e(k)}(1+2^{-\delta k}) \leq 0.5$ that

$$\begin{aligned}
X &= p\ln(p/p_0) + (1-p)[\ln(1-p) - \ln(1-p_0)] \\
&= 2^{-e(k)}(1+2^{-\delta k})\ln(1+2^{-\delta k}) \\
&\quad +[1-2^{-e(k)}(1+2^{-\delta k})][\ln(1-2^{-e(k)}(1+2^{-\delta k})) - \ln(1-2^{-e(k)})] \\
&\geq 2^{-e(k)}(1+2^{-\delta k})(2^{-\delta k} - \tfrac{2}{3}2^{-2\delta k}) \\
&\quad +[1-2^{-e(k)} - 2^{-e(k)-\delta k}][-2^{-e(k)}(1+2^{-\delta k}) - 2^{-2e(k)}(1+2^{-\delta k})^2 + 2^{-e(k)}] \\
&\geq \tfrac{1}{3}2^{-e(k)-2\delta k} - \tfrac{2}{3}2^{-e(k)-3\delta k} - 2^{-2e(k)}(1+2^{-\delta k})^2. \qquad (6)
\end{aligned}$$

Combining inequalities (4), (5) and (6) and using $\binom{n(k)}{e(k)} \leq n(k)^k$ yields

$$P[|p^{f^{G_k},G_k} - p^{f^{G_k},R}| \geq 2^{-\delta k}] \;\leq\; n(k)^k 2^{e(k)} e^{-2^k X} \;=\; e^{-2^k X + e(k)\ln 2 + k\ln n(k)}. \qquad (7)$$

Let $\epsilon > 0$, $e(k) = \lfloor (1-\epsilon)k \rfloor$ and $0 < \delta < \epsilon/2$ and let $\gamma = \epsilon - 2\delta > 0$. Then

$$2^k X \;\geq\; \tfrac{1}{3}2^{(\epsilon-2\delta)k} - \tfrac{2}{3}2^{(\epsilon-3\delta)k+1} - 2^{-2((1-\epsilon)k+1)}(1+2^{-\delta k})^2$$

and thus $2^k X > a 2^{\gamma k}$ for every $a < 1/3$ and sufficiently large k. If $n(k)$ is upper bounded by a polynomial in k, the exponent $-2^k X + e(k)\ln 2 + k \ln n(k)$ on the right side of (7) is negative for sufficiently large k. Therefore, there must exist a pseudo-random number generator such that, for sufficiently large k,

$$\left| p_k^{S^G,G} - p_k^{S^G,R} \right| \;<\; 2^{-\delta k}$$

for all statistical tests S^G that examine at most $e(k) = \lfloor (1-\epsilon)k \rfloor$ bits of the generated sequence with no constraints on the location of these examined bits. Note that the distinguishing probability decreases exponentially with k. □

Remark: Theorem 4 can also be proved if the specification $e(k) = \lfloor (1-\epsilon)k \rfloor$ is replaced by the specification $e(k) = \lfloor k - c(\log k)^\alpha \rfloor$ for some $\alpha > 1$; one needs only to replace $2^{-\delta k}$ in the proof by $2^{c'(\log k)^\alpha}$ for $c' < c/2$.

It is an open problem whether polynomial-time-computable degree $e(k)$ locally-randomized pseudo-random number generators exist for which $\lim_{k\to\infty} e(k)\log k\,/k > 0$, for instance, with $\lim_{k\to\infty} e(k)/k > C$ for some constant C with $0 < C \leq 1$. We conjecture that the answer is yes. Piveteau [16] has recently considered locally-randomized pseudo-random number generators in a setting where all computations are polynomially bounded and proved that there exist locally-randomized pseudo-random number generators if and only if there exist pseudo-random number generators.

4. Extensions of the Concept of Local Randomization

So far we have considered statistical tests that are limited in the total number of bits of the pseudo-random sequence that are examined during the execution. This corresponds to a known-plaintext attack with a limited amount of plaintext data available when the pseudo-random sequence is the "running key" in an additive stream cipher. In general, however, the nature of the enemy's a priori and/or obtainable information about the plaintext is global rather than structured in binary digits. For example, he might know that the plaintext satisfies certain parity checks (e.g., reduced ASCII code). It would therefore be desirable to extend the results of Sections 2 and 3 to purely information-theoretic results by allowing the statistical test to obtain the value of any random variable not giving more than e bits of information about the pseudo-random sequence (or, equivalently, about the plaintext sequence in the additive stream cipher described above). The following theorem shows that unfortunately such an extension is not possible.

Theorem 5: *For every (k,n) sequence generator G, there exists a function f^G using one bit of information about the generated sequence, whose distinguishing probability $|p^{f^G,G} - p^{f^G,R}|$ is lower bounded by $1 - 2^{k-n}$.*

Proof: Assume f^G can only obtain the output of the binary-valued function $I_n \longrightarrow \{0,1\}$ that assigns to every $\underline{s}^n \in I_n$ the value 1 if and only if $\underline{s}^n$ is a sequence that can be generated by G. Certainly the result of a binary-valued function can at most give one bit of information. By simply feeding this input bit through to the output without processing, f^G achieves $p^{f^G,G} = 1$ and $p^{f^G,R} \leq 2^{k-n}$ since the cardinality of the set of all sequences that can be generated by G is upper bounded by 2^k. □

Theorem 5 shows not only that the amount of information that the enemy is allowed to obtain about the generated sequence but also the way in which the enemy can access information must be restricted appropriately if statements similar to theorems 1 and 4 should be proved. A possible relaxation of the restriction that the enemy obtains information about the pseudo-random sequence by observing only bits could, for example, be that he is allowed to obtain at most e parity checks, i.e., linear combinations, on the sequence bits. But even for this model, perfect "local" randomness cannot be achieved because n binary CTRV's are jointly independent if and only if every non-trivial linear combination of these CTRV's is a CTRV (see [23], or the XOR-Lemma in [5]). On the other hand, if the enemy is not allowed to obtain arbitrary bits of the sequence but only subblocks of a certain length, i.e., if the basic alphabet is the set of binary m-tuples rather than the binary alphabet, then (k,n) sequence generators achieving the information-theoretically maximal degree of perfect local randomness k can sometimes be achieved. In coding theory, schemes having this property (e.g., Reed-Solomon codes) are called maximum distance separable, cf. [12]. The problem of determining the minimal alphabet size such that there exists a (k,n,k) PLR for given k and n is open.

Two other ways of generalizing the concept of a (k,n,e) perfect local randomizer would be either to drop perfectness, i.e., to allow slight deviations from the uniform distribution, or to require perfect local randomness only "almost everywhere", i.e., only for all but a small fraction of the subsets of e bits. A suitable definition for the first generalization is given in the following:

Definition 5: A (k,n) sequence generator is a (k,n,e,δ) *local randomizer* if, when the input is a sequence of k independent CTRV's, then for every subset S of e of the n output random variables, $H(S) \geq e - \delta$, where $H(S)$ is Shannon's entropy [20] of the set S of random variables.

5. Applications and Conclusions

We are by no means suggesting that the sequence generators presented in Section 2 be used as practical pseudo-random sequence generators. The two main reasons for this reticence are first that one cannot often validly assume that an enemy is restricted to obtaining only a few bits of the sequence and second that most of our proposed schemes are linear and therefore easily unmasked by a simple appropriate parity check involving $e+1$ bits. The latter weakness could be obviated by application of an invertible nonlinear transformation on the sequence space, but the first problem is intrinsic. Nevertheless, there are two potential practical cryptographic applications of the proposed perfect local randomizers. We first note that they are expanding transformations providing, in a certain sense, ideal "diffusion". If combined with appropriate compressing transformations providing "confusion", they might be excellent building blocks for practical ciphers. The second possible application is their use in key scheduling schemes (e.g., within block ciphers) where a small secret key must be stretched to a long key.

In this paper we have explored the concept of local randomization which leads to provable security, but only for a weak notion of security. We note also that, once more in cryptologic research, results borrowed from the theory of error-correcting codes have turned out to be useful.

Acknowledgement

We would like to thank Moni Naor for drawing our attention to references [1] and [4] and Andi Loeliger for helpful comments.

References

[1] N. Alon, L. Babai and A. Itai, *A fast and simple randomized parallel algorithm for the maximal independent set problem*, Journal of Algorithms, Vol. 7, pp. 567-583, 1986.

[2] L. Blum, M. Blum and M. Shub, *A simple unpredictable pseudo-random number generator*, SIAM J. on Computing, Vol. 15, pp. 364-383, 1986.

[3] M. Blum and S. Micali, *How to generate cryptographically strong sequences of pseudo-random bits*, SIAM J. on Computing, Vol. 13, pp. 850-864, 1984.

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

[5] B. Chor, O. Goldreich, J. Hastad, J. Freidmann, S. Rudich and R. Smolensky, *The bit extraction problem or t-resilient functions*, Proc. 26th ann. Symp. on Foundations of Computer Science, pp. 396-407, 1985.

[6] P. Delsarte, *An algebraic approach to the association schemes of coding theory*, Philips Research Reports Supplements, No. 10, 1973.

[7] A. Joffe, *On a sequence of almost deterministic pairwise independent random variables*, Proc. Amer. Math. Soc., Vol. 29, No. 2, pp. 381-382, July 1971.

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

[9] E. Kranakis, *Primality and cryptography*, Stuttgart and New York: Wiley-Teubner Series in Computer Science, 1986.

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

[11] M. Luby, *A simple parallel algorithm for the maximal independent set problem*, SIAM J. on Computing, Vol. 15, No. 4, pp. 1036-1053, Nov. 1986.

[12] F.J. MacWilliams and N.J.A. Sloane, *The theory of error-correcting codes*, Amsterdam, New York, Oxford: North-Holland Publishing Company, Fifth Printing, 1986.

[13] R.J. McEliece, E.R. Rodemich, H.C. Rumsey and L.R. Welch, *New upper bounds on the rate of a code via the Delsarte-MacWilliams inequalities*, IEEE Trans. Info. Th., Vol. IT-23, pp. 157-166, 1977.

[14] S. Micali and C.P. Schnorr, *Efficient, perfect random number generators*, Preprint MIT, Universität Frankfurt, Nov. 1988.

[15] L.H. Ozarow and A. D. Wyner, *Wire-tap channel II*, AT&T Bell Lab. Tech. J., Vol. 63, No. 10, pp. 2135-2157, Dec. 1984.

[16] J.-M. Piveteau, *Local pseudorandom generators*, Preprint, ETH Zürich, 1989.

[17] D. Raghavarao, *Constructions and combinatorial problems in Design of Experiments*, New York: Wiley, 1971.

[18] R.A. Rueppel, *On the security of Schnorr's pseudo-random generator*, to appear in Proc. of EUROCRYPT'89.

[19] C.P. Schnorr, *On the construction of random number generators and random function generators*, Proc. EUROCRYPT'88, Lecture Notes in Computer Science, Vol. 330, Springer Verlag, pp. 225-232, 1988.

[20] C.E. Shannon, *A mathematical theory of communication*, Bell Syst. Tech. J., Vol. 27, pp. 379-423 and 623-656, 1948.

[21] T. Verhoeff, *An updated table of minimum-distance bounds for binary linear codes*, IEEE Trans. Info. Th., Vol. IT-33, pp. 665-680, 1987.

[22] J.M. Wozencraft and B. Reiffen, *Sequential Decoding*, MIT Press, Cambridge, MA, 1961.

[23] G.Z. Xiao and J.L. Massey, *A spectral characterization of correlation-immune combining functions*, IEEE Trans. Inform. Theory, Vol. 34, pp. 569-571, 1988.

[24] A.C. Yao, *Theory and applications of trapdoor functions*, Proc. 23rd IEEE Symposium on Foundations of Computer Science, pp. 80-91, 1982.