

Cryptographic pseudo-random sequences from the chaotic Hénon map

MADHEKAR SUNEEL

PGAD, Defence Research and Development Organization, DRDL Complex,
Kanchanbagh, Hyderabad 500 058
e-mail: suneel@ieee.org

MS received 20 November 2008; revised 9 April 2009

Abstract. A scheme for pseudo-random binary sequence generation based on the two-dimensional discrete-time Hénon map is proposed. Properties of the proposed sequences pertaining to linear complexity, linear complexity profile, correlation and auto-correlation are investigated. All these properties of the sequences suggest a strong resemblance to random sequences. Results of statistical testing of the sequences are found encouraging. An estimate of the keyspace size is presented.

Keywords. Chaos; nonlinear difference equations; random number generation; stream ciphers; cryptography.

1. Introduction

Pseudo-random number sequences are useful in many applications including Monte-Carlo simulation, spread spectrum communications, steganography and cryptography. Conventionally, pseudo-random sequence generators based on linear congruential methods and feedback shift-registers are popular (Knuth 1998). For cryptographic applications, several algorithms such as ANSI X9.17 and FIPS 186 are found to be popular (Menezes *et al* 1997). In recent times, several researchers have been exploring the idea of using chaotic dynamical systems for this purpose (Falcioni *et al* 2006, Kocarev 2001, Woodcock & Smart 1998). The random-like, unpredictable dynamics of chaotic systems, their inherent determinism and simplicity of realization suggest their potential for exploitation as pseudo-random number generators.

Cryptographic schemes based on chaos have been classified as (i) *discrete-time discrete-value* schemes (ii) *discrete-time continuous-value* schemes and (iii) *continuous-time continuous-value* schemes (Dachselt & Schwarz 2001). In this paper, a scheme for obtaining a pseudo-random binary sequence from the two-dimensional chaotic Hénon map is explored. A cryptographic application of the scheme can be classified under the first of the three classes.

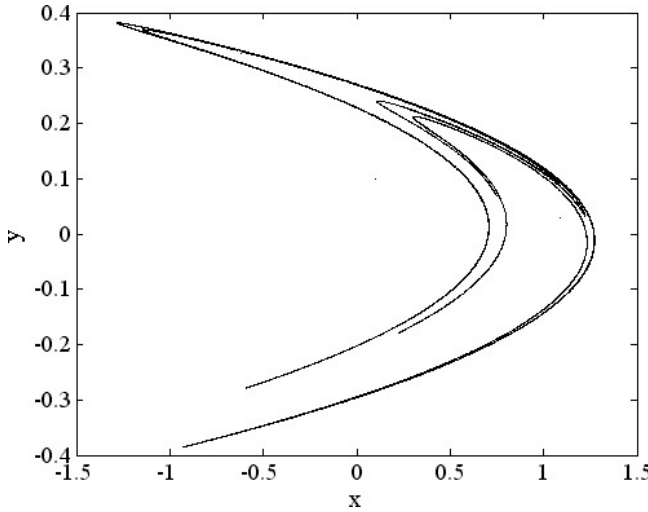


Figure 1. The strange attractor of Hénon.

2. The Hénon map

An N -dimensional discrete-time dynamical system is an iterative map $f : \mathfrak{R}^N \rightarrow \mathfrak{R}^N$ of the form

$$X_{k+1} = f(X_k), \tag{1}$$

where $k = 0, 1, \dots$ is the discrete time and $X \in \mathfrak{R}^N$ is the state. Starting from X_0 , the initial state, repeated iteration of (1) gives rise to a series of states known as an orbit. An example is the Hénon map, a two-dimensional discrete-time nonlinear dynamical system represented by the state equations (Hénon 1976, Kathleen 1997, Peitgen 2004, Crutchfield *et al* 1986, Gleick 1997, Kumar 1996)

$$\begin{aligned} x_{k+1} &= -\alpha x_k^2 + y_k + 1, \\ y_{k+1} &= \beta x_k. \end{aligned} \tag{2}$$

Here, (x, y) is the two-dimensional state of the system. The state-plane diagram for $\alpha = 1.4$ and $\beta = 0.3$ for this map is shown in figure 1. The diagram is a strange attractor popularly known as the Hénon attractor.

In this paper, the Hénon map shall be considered as a representative example of 2-dimensional chaotic maps for the generation of pseudorandom sequences.

3. Pseudo-random sequence generation scheme

Generating a pseudorandom binary sequence from the orbit of a chaotic map essentially requires mapping the state of the system to $\{0, 1\}$. For the Hénon map, consider the two bits b_x and b_y derived respectively from the x and y state-variables as follows:

$$b_x = \begin{cases} 1 & \text{if } x > \tau_x; \\ 0 & \text{if } x \leq \tau_x. \end{cases} \tag{3}$$

$$b_y = \begin{cases} 1 & \text{if } y > \tau_y; \\ 0 & \text{if } y \leq \tau_y. \end{cases} \tag{4}$$

Here, τ_x and τ_y are appropriately chosen threshold values for state-variables x and y . τ_x should be chosen such that the likelihood of $x > \tau_x$ is equal to that of $x \leq \tau_x$. The median of a large set of numbers has precisely this property. Therefore, we choose τ_x as the median of a large number (T) of consecutive values of x . Similarly, we assign to τ_y , the value of the median of T consecutive values of y . Thus, two streams of bits $S_x = \{b_x^i\}_{i=1}^\infty$ and $S_y = \{b_y^i\}_{i=1}^\infty$ are obtained from the map. Consider the bit-stream B_x formed by choosing every P^{th} bit of S_x , i.e. $B_x = \{b_x^{Pi}\}_{i=1}^\infty$. Consider the similarly formed bit-stream $B_y = \{b_y^{Pi}\}_{i=1}^\infty$. Let us denote the j^{th} bit of these two sequences respectively as $B_x(j)$ and $B_y(j)$. Then, the pseudo-random output bit O is chosen as per the following rule:

$$O(j) = \begin{cases} B_x(j) & \text{if } B_y(j - 2) = 0 \text{ and } B_y(j - 1) = 0; \\ \overline{B_x}(j) & \text{if } B_y(j - 2) = 0 \text{ and } B_y(j - 1) = 1; \\ B_y(j) & \text{if } B_y(j - 2) = 1 \text{ and } B_y(j - 1) = 0; \\ \overline{B_y}(j) & \text{if } B_y(j - 2) = 1 \text{ and } B_y(j - 1) = 1. \end{cases} \tag{5}$$

Here, $\overline{B_x}$ and $\overline{B_y}$ respectively denote the logical inverse of B_x and B_y . For $j = 0$, $B_y(-2)$ and $B_y(-1)$ can arbitrarily be assumed to be 0.

The author has found that generated pseudorandom sequences have good statistical properties when P is large. The author has used P between 75 and 5000 depending on the available time for computation and the length of the sequence required. In this paper, sequences generated using this method are called *Hénon map sequences*.

4. Linear complexity properties

A Linear Feedback Shift Register (LFSR) is said to *generate* an N -bit sequence W if for some initial state, the first N bits of the output sequence of the LFSR are the same as W (Golomb 1964, Golomb 1982). The length of the shortest LFSR that generates W is known as its *linear complexity*.

The author has measured the linear complexity of a large number of even-length Hénon map sequences using the Berlekamp–Massey Algorithm (Massey 1969). The linear complexities obtained for each sequence-length were found to follow a certain probabilistic pattern. In particular, the probability of the linear complexity C of an N -bit sequence being equal to c ($c < N$), when N is even, was found to be very close to

$$P(C = c) = \begin{cases} (0.5)^{N-2c+1} & \text{if } c \leq N/2; \\ (0.5)^{2c-N} & \text{if } c > N/2. \end{cases} \tag{6}$$

To illustrate the correctness of this conjecture, the experimentally determined distribution of linear complexities for 64-bit Hénon map sequences is shown in figure 2a alongside the conjectured distribution in figure 2b that has been computed using (6). The mean value of linear complexities obtained in this experiment was found to be 32.2083. The expectation of linear complexity for a 64-bit random sequence is 32.2222 and is found to be very close to that of the Hénon map sequences (Menezes *et al* 1997). The variance of linear complexities of 64-bit sequences obtained by experiment was 1.0811 against 1.0617 for random sequences.

Similarly, the probability of the linear complexity C assuming the value c ($c < N$) when N is odd was found to be very close to

$$P(C = c) = \begin{cases} (0.5)^{N-2c+1} & \text{if } c < (N + 1)/2; \\ (0.5)^{2c-N} & \text{if } c \geq (N + 1)/2. \end{cases} \tag{7}$$

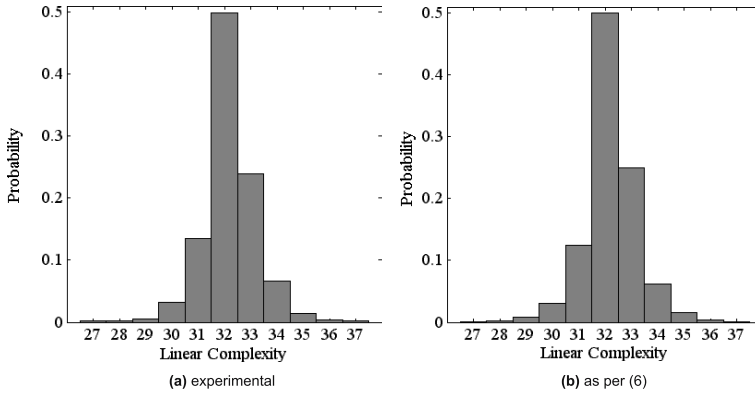


Figure 2. Probability distribution of linear complexity: 64-bit sequences.

Again, in figure 3a, the experimental distribution is shown along with the experimentally determined distribution in figure 3b for 65-bit Hénon map sequences. The experimentally measured mean of linear complexities of these sequences was 32.7663 against the expected 32.7778 for random sequences. Also, the measured variance of linear complexities stands at 1.1177 against 1.0617 for random sequences (Menezes *et al* 1997).

Let W be an N -bit binary sequence. Let $W_i (i = 1, 2, \dots, N)$ denote the subsequence of W consisting of its first i bits. Let C_i denote the linear complexity of W_i . Then the sequence of linear complexities $(C_1, C_2, C_3, \dots, C_N)$ is known as the linear complexity *profile* of W . For random sequences, the linear complexity profile is expected to be very close to the $C_i = i/2$ line (Menezes *et al* 1997). The linear complexity profile of a sample 553-bit sequence was determined using the Berlekamp–Massey Algorithm. The obtained profile is shown in figure 4 and can be seen to be very close to the $C_i = i/2$ line.

5. Correlation properties

Consider two N -bit binary sequences U and V . Let A be the number of bit-by-bit agreements between the two. The number of bit-by-bit disagreements must be $D = (N - A)$. Then the

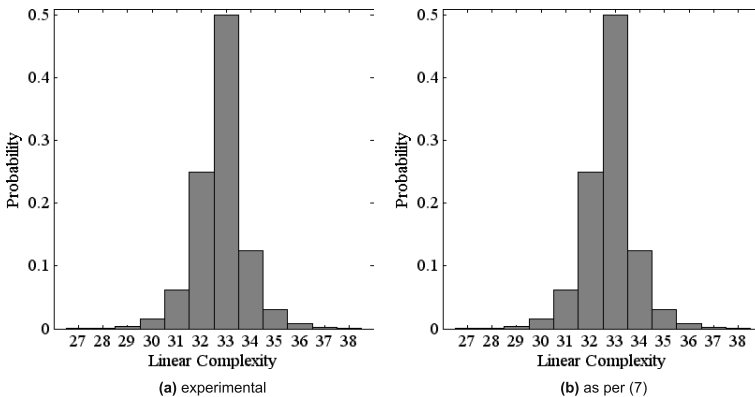


Figure 3. Probability distribution of linear complexity: 65-bit sequences.

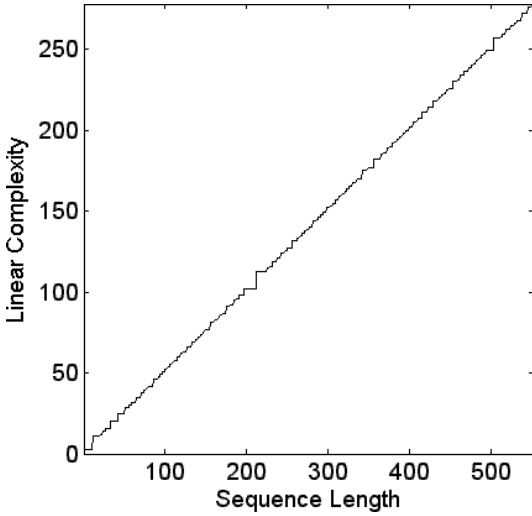


Figure 4. Linear complexity profile.

correlation θ of the two sequences is defined as

$$\theta(U, V) = (A - D)/N. \tag{8}$$

Theorem 1. *If U and V are two N -bit random binary sequences, the probability of their correlation θ assuming the value Θ is given by*

$$P(\theta = \Theta) = \sqrt{\frac{2}{N\pi}} e^{-N\Theta^2/2} \tag{9}$$

when

$$\Theta \in \begin{cases} \{0, \pm \frac{2}{N}, \pm \frac{4}{N}, \dots, \pm 1\} & \text{for even } N, \\ \{\pm \frac{1}{N}, \pm \frac{3}{N}, \pm \frac{5}{N}, \dots, \pm 1\} & \text{for odd } N. \end{cases}$$

The probability is zero for other values of Θ .

Proof. Since the probability of occurrence of a one is the same as the probability of occurrence of a zero in a truly random sequence, the probability of occurrence of an agreement ($p = 0.5$) is the same as the probability of occurrence of a disagreement ($1 - p = 0.5$) in a pair of such sequences. Therefore, the number of agreements A in a pair of such sequences is a random variable that follows the binomial distribution with mean $\mu = N/2$ and standard deviation $\sigma = \sqrt{N}/2$. Therefore,

$$P(A = r) = \frac{{}^N C_r}{2^N}. \tag{10}$$

Applying the Normal approximation to the binomial distribution (Keeping 1962, Ramasubramanian 1997),

$$P(A = r) \approx \sqrt{\frac{2}{N\pi}} e^{-2(r-N/2)^2/N}. \tag{11}$$

The correlation θ is related to the number of agreements A as

$$\theta = \frac{2A}{N} - 1. \tag{12}$$

As $A \in \{0, 1, 2, \dots, N\}$, $\theta \in \{-1, (\frac{2}{N} - 1), (\frac{4}{N} - 1), \dots, 1\}$. Therefore, $\theta \in \{0, \pm\frac{2}{N}, \pm\frac{4}{N}, \dots, \pm 1\}$ for even N and $\theta \in \{\pm\frac{1}{N}, \pm\frac{3}{N}, \pm\frac{5}{N}, \dots, \pm 1\}$ for odd N . Clearly, the probability of θ assuming any other value is zero. By (11) and (12), if Θ belongs to the above set of valid values,

$$P(\theta = \Theta) = \sqrt{\frac{2}{N\pi}} e^{-N\Theta^2/2} \tag{13}$$

which completes the proof.

The correlation between pairs of Hénon map sequences was experimentally determined and the probability distribution was found to be very close to that of (9). As an illustration, the probability distribution for 127-bit Hénon map sequences and the expected distribution for random sequences are shown in figure 5.

The correlation of a sequence with a cyclic-shift of itself is known as its *cyclic auto-correlation*. Let W be an N -bit sequence and W^j denote W cyclically right-shifted by j bits. Then the cyclic auto-correlation *function* of W is defined as $R(j) = \theta(W, W^j)$. The (cyclic) auto-correlation function of a random sequence is expected to be unity at $j = 0$ and close to zero at all other values of j . This indeed was found to be the case with Hénon map sequences. The auto-correlation function of a 2000-bit Hénon map sequence is shown in figure 6. In the figure, a negative value of shift signifies cyclic left-shifting by an amount equal to the magnitude.

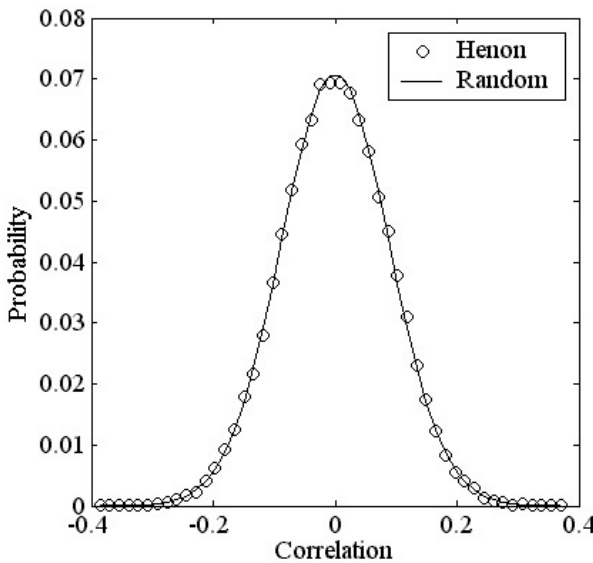


Figure 5. Probability distribution of correlation. The legend *Random* is used for a plot of (9).

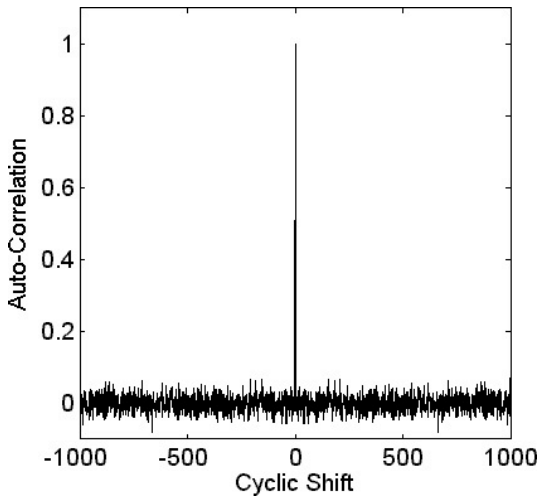


Figure 6. Auto-correlation function.

6. Statistical testing

A number of sequences were generated by the algorithm and were subjected to statistical tests. The tests carried out were Menezes *et al*'s basic tests of randomness (Menezes *et al* 1997), FIPS 140-1 (FIPS PUB 140-1) recommended battery of tests and National Institute of Standards and Technology (NIST) battery of tests (Rukhin *et al* 2001).

Menezes *et al*'s set of tests consists of frequency, serial, poker, runs and auto-correlation tests. While the first four tests were carried out once for a given sequence, the auto-correlation test was carried out for all possible shifted-versions of the sequence. The result of each test is a test statistic which is compared with a threshold value (one-sided test). The tests were carried out at a significance level of 0.01, hence 1% of failures were expected even for random sequences.

FIPS 140-1 recommends monobit, poker, runs and long-run tests. Each test is a two-sided test where a test-statistic is required to lie within an interval.

The NIST statistical test suite consists of a battery of sixteen tests namely 1) frequency (monobit) test, 2) frequency test within a block, 3) runs test, 4) test for longest run of ones in a block, 5) binary matrix rank test, 6) discrete Fourier transform (spectral) test, 7) non-overlapping template matching test, 8) overlapping template matching test, 9) Maurer's universal statistical test, 10) Lempel-Ziv compression test, 11) linear complexity test, 12) serial test, 13) approximate entropy test, 14) cumulative sums test, 15) random excursions test and 16) random excursions variant test.

7. Statistical test results

7.1 Menezes *et al*'s basic tests of randomness

Two 128-bit sequences R_1 and R_2 were generated using parameters shown in table 1. The notation used for test-statistics is same as that in Menezes *et al* (1997). The statistic X_5 of the auto-correlation test is a function of the value of the relative circular shift. In this case, the value of the relative circular shift is shown in brackets. For example, the statistics obtained

Table 1. Parameters used for Menezes' test samples.

Parameter	R_1	R_2
α	1.40	1.20
β	0.30	0.30
x_0	-0.75	-0.75
y_0	-0.02	0.32
$B_y(-2)$	0	0
$B_y(-1)$	1	1
P	24	24
T	1000	1000

by running the auto-correlation test on the sequence and its 3-shifted version is denoted by $X_5(3)$. A similar notation is used for the statistic X_3 of the poker test where X_3 depends on the length of the non-overlapping sub-sequences. The results of the tests on the sequences are shown in table 2.

7.2 FIPS 140-1

Testing for compliance with FIPS 140-1 is required to be carried out with sequences of 20,000 bits. Five sequences S_1 through S_5 were generated using parameters shown in table 3. The results of the tests on the sequences are shown in table 4.

7.3 NIST statistical test suite

For each test-run of the test-suite, a sequence of 2×10^8 bits was generated and the test-suite was configured to consider this sequence as 200 sequences of 1×10^6 bits each. This set of 200 sequences was subjected to statistical testing in each case. In this manner, the distribution of failures could also be examined by the test-suite and appropriate analysis could be carried out.

The test was carried out on two sets U_1 and U_2 of 200-samples each. The parameters used for generating these sets of sequences are shown in table 5.

For a sample of 200 sequences, the minimum proportion of sequences required to pass all the tests of the suite other than the random excursions (variant) test is 0.968893. The proportion of sequences passing these tests was found to be larger than this threshold. For the random excursions (variant) test, the required minimum passing proportion was found to be 0.961540 and 0.962864 for U_1 and U_2 respectively. U_1 and U_2 were found to meet this requirement also. The author has noticed, however, that some of the sequences that pass the FIPS 140-1 and Menezes' tests do not pass the NIST suite of tests. For example, a set of sequences generated using the same parameters as R_1 was found to fail in the NIST suite. The author has found that passing the NIST suite requires a more careful choice of the parameters. In this case, increasing the value of T to a sufficiently large value (about 10000) resulted in the sequences passing the NIST suite.

8. Keyspace size

The properties of Hénon map sequences presented in the preceding sections demonstrate that they are potential candidates for cryptographic applications. A Hénon map sequence,

Table 2. Testing for Menezes' basic tests of randomness.

Statistic	Expected	R_1	Result (R_1)	R_2	Result (R_2)
X_1	< 6.634897	0.125000	Pass	6.125000	Pass
X_2	< 9.210340	0.213583	Pass	6.245079	Pass
$X_3(2)$	< 11.344867	4.625000	Pass	7.625000	Pass
$X_3(3)$	< 18.475307	1.809524	Pass	9.428571	Pass
X_4	< 9.210340	0.890713	Pass	5.373956	Pass
$X_5(1)$	$ X_5(\cdot) < 2.326348$	0.443678	Pass	-0.088736	Pass
$X_5(2)$	$ X_5(\cdot) < 2.326348$	0.178174	Pass	1.069045	Pass
$X_5(3)$	$ X_5(\cdot) < 2.326348$	0.268328	Pass	-0.98387	Pass
$X_5(4)$	$ X_5(\cdot) < 2.326348$	-1.257237	Pass	-0.179605	Pass
$X_5(5)$	$ X_5(\cdot) < 2.326348$	-0.631169	Pass	-0.811503	Pass
$X_5(6)$	$ X_5(\cdot) < 2.326348$	0	Pass	-0.724286	Pass
$X_5(7)$	$ X_5(\cdot) < 2.326348$	0.272727	Pass	0.272727	Pass
$X_5(8)$	$ X_5(\cdot) < 2.326348$	-0.730297	Pass	-0.912871	Pass
$X_5(9)$	$ X_5(\cdot) < 2.326348$	-0.09167	Pass	0.09167	Pass
$X_5(10)$	$ X_5(\cdot) < 2.326348$	1.288804	Pass	0	Pass
$X_5(11)$	$ X_5(\cdot) < 2.326348$	0.09245	Pass	-0.64715	Pass
$X_5(12)$	$ X_5(\cdot) < 2.326348$	0	Pass	0	Pass
$X_5(13)$	$ X_5(\cdot) < 2.326348$	0.466252	Pass	-0.466252	Pass
$X_5(14)$	$ X_5(\cdot) < 2.326348$	0.749269	Pass	-0.561951	Pass
$X_5(15)$	$ X_5(\cdot) < 2.326348$	-0.658505	Pass	-0.282216	Pass
$X_5(16)$	$ X_5(\cdot) < 2.326348$	0.188982	Pass	-0.377964	Pass
$X_5(17)$	$ X_5(\cdot) < 2.326348$	1.233905	Pass	-1.993232	Pass
$X_5(18)$	$ X_5(\cdot) < 2.326348$	0.381385	Pass	-2.097618	Pass
$X_5(19)$	$ X_5(\cdot) < 2.326348$	-0.287348	Pass	-0.478913	Pass
$X_5(20)$	$ X_5(\cdot) < 2.326348$	-1.539601	Pass	-0.96225	Pass
$X_5(21)$	$ X_5(\cdot) < 2.326348$	1.06341	Pass	-0.290021	Pass
$X_5(22)$	$ X_5(\cdot) < 2.326348$	0.194257	Pass	-1.748315	Pass
$X_5(23)$	$ X_5(\cdot) < 2.326348$	1.85421	Pass	-2.04939	Pass
$X_5(24)$	$ X_5(\cdot) < 2.326348$	0.784465	Pass	0.392232	Pass
$X_5(25)$	$ X_5(\cdot) < 2.326348$	1.280928	Pass	0.68973	Pass
$X_5(26)$	$ X_5(\cdot) < 2.326348$	1.782266	Pass	-0.594089	Pass
$X_5(27)$	$ X_5(\cdot) < 2.326348$	-0.298511	Pass	-0.298511	Pass
$X_5(28)$	$ X_5(\cdot) < 2.326348$	-0.4	Pass	-1.8	Pass
$X_5(29)$	$ X_5(\cdot) < 2.326348$	1.306549	Pass	1.105542	Pass
$X_5(30)$	$ X_5(\cdot) < 2.326348$	0.808122	Pass	0.606092	Pass
$X_5(31)$	$ X_5(\cdot) < 2.326348$	1.929158	Pass	-0.507673	Pass
$X_5(32)$	$ X_5(\cdot) < 2.326348$	0	Pass	-0.408248	Pass
$X_5(33)$	$ X_5(\cdot) < 2.326348$	-0.102598	Pass	0.307794	Pass
$X_5(34)$	$ X_5(\cdot) < 2.326348$	0	Pass	-0.206284	Pass
$X_5(35)$	$ X_5(\cdot) < 2.326348$	0.933257	Pass	-2.384989	Fail
$X_5(36)$	$ X_5(\cdot) < 2.326348$	0.625543	Pass	-0.208514	Pass
$X_5(37)$	$ X_5(\cdot) < 2.326348$	-0.314485	Pass	-0.943456	Pass
$X_5(38)$	$ X_5(\cdot) < 2.326348$	-0.843274	Pass	-1.897367	Pass
$X_5(39)$	$ X_5(\cdot) < 2.326348$	-0.317999	Pass	-0.317999	Pass
$X_5(40)$	$ X_5(\cdot) < 2.326348$	-1.918806	Pass	-0.639602	Pass
$X_5(41)$	$ X_5(\cdot) < 2.326348$	1.393746	Pass	-0.964901	Pass
$X_5(42)$	$ X_5(\cdot) < 2.326348$	-0.862662	Pass	0.215666	Pass
$X_5(43)$	$ X_5(\cdot) < 2.326348$	-0.325396	Pass	-0.325396	Pass
$X_5(44)$	$ X_5(\cdot) < 2.326348$	-1.309307	Pass	-0.872872	Pass
$X_5(45)$	$ X_5(\cdot) < 2.326348$	1.426935	Pass	-1.426935	Pass
$X_5(46)$	$ X_5(\cdot) < 2.326348$	-0.441726	Pass	-0.220863	Pass

Table 2. (Continued).

Statistic	Expected	R_1	Result (R_1)	R_2	Result (R_2)
$X_5(47)$	$ X_5(\cdot) < 2.326348$	-1	Pass	0.777778	Pass
$X_5(48)$	$ X_5(\cdot) < 2.326348$	-0.67082	Pass	-0.447214	Pass
$X_5(49)$	$ X_5(\cdot) < 2.326348$	-1.237597	Pass	-1.012579	Pass
$X_5(50)$	$ X_5(\cdot) < 2.326348$	0.452911	Pass	0.679366	Pass
$X_5(51)$	$ X_5(\cdot) < 2.326348$	-1.025645	Pass	0.797724	Pass
$X_5(52)$	$ X_5(\cdot) < 2.326348$	-0.458831	Pass	0.458831	Pass
$X_5(53)$	$ X_5(\cdot) < 2.326348$	-0.11547	Pass	-0.57735	Pass
$X_5(54)$	$ X_5(\cdot) < 2.326348$	-0.464991	Pass	-1.394972	Pass
$X_5(55)$	$ X_5(\cdot) < 2.326348$	-0.819288	Pass	-0.117041	Pass
$X_5(56)$	$ X_5(\cdot) < 2.326348$	0.235702	Pass	-0.471405	Pass
$X_5(57)$	$ X_5(\cdot) < 2.326348$	-1.780172	Pass	0.593391	Pass
$X_5(58)$	$ X_5(\cdot) < 2.326348$	-0.717137	Pass	0.239046	Pass
$X_5(59)$	$ X_5(\cdot) < 2.326348$	-1.324244	Pass	-1.083473	Pass
$X_5(60)$	$ X_5(\cdot) < 2.326348$	-0.242536	Pass	0.242536	Pass
$X_5(61)$	$ X_5(\cdot) < 2.326348$	1.588203	Pass	-0.855186	Pass
$X_5(62)$	$ X_5(\cdot) < 2.326348$	-0.246183	Pass	-1.723281	Pass
$X_5(63)$	$ X_5(\cdot) < 2.326348$	0.620174	Pass	0.124035	Pass
$X_5(64)$	$ X_5(\cdot) < 2.326348$	-0.5	Pass	0.5	Pass

for example, can be directly Exclusive-ORed, bit-by-bit, with a data sequence of the same length. Such a cipher is popularly known as the Vernam Cipher (Kippenhahn 1999, Mollin 2001). The values of α, β, x_0, y_0 and the sampling-factor P together can form the key. In this section, an attempt is made to estimate the size of the keyspace for such a cipher.

The author has found that α in (1.16, 1.41) and β in (0.2, 0.3) are useful for generating sequences with the desired statistical properties. Also, for estimating the keyspace size, let us assume (x_0, y_0) is taken from a rectangular area around the strange attractor of figure 1. Choices of (x_0, y_0) outside this area also yield satisfactory results, as can be seen from several examples presented in this paper, but the keyspace size determined under this assumption gives a useful lower estimate for the actual value. In particular, we assume x_0 to lie in $(-1, 1)$ and y_0 to lie in $(-0.35, 0.35)$. P can be assumed within (80, 1000). Though these limits are in no way binding or accurate, they should give a reasonable estimate of the size of the keyspace.

Table 3. Parameters used for FIPS 140-1 test samples.

Parameter	S_1	S_2	S_3	S_4	S_5
α	1.23	1.40	1.40	1.40	1.41
β	0.25	0.25	0.30	0.30	0.21
x_0	-1.0	-1.0	-1.0	-1.0	-1.0
y_0	1.0	1.0	1.0	1.0	1.0
$B_y(-2)$	0	0	0	0	0
$B_y(-1)$	1	1	1	1	1
P	84	84	84	24	24
T	1000	1000	1000	1000	1000

Table 4. Testing for FIPS 140-1.

Statistic	Expected	S_1	S_2	S_3	S_4	S_5
n_1	$9654 < n_1 < 10346$	9938	10107	9944	10099	10020
X_3	$1.03 < X_3 < 57.40$	17.2544	13.0304	12.9792	14.7328	6.6624
B_1	$2267 < B_1 < 2733$	2572	2473	2560	2480	2454
G_1	$2267 < G_1 < 2733$	2452	2524	2534	2554	2447
B_2	$1079 < B_2 < 1421$	1192	1231	1200	1286	1268
G_2	$1079 < G_2 < 1421$	1302	1264	1226	1262	1310
B_3	$502 < B_3 < 748$	640	643	653	636	616
G_3	$502 < G_3 < 748$	611	577	638	582	580
B_4	$223 < B_4 < 402$	277	319	320	312	312
G_4	$223 < G_4 < 402$	328	315	306	336	314
B_5	$90 < B_5 < 223$	140	165	162	150	156
G_5	$90 < G_5 < 223$	156	165	168	148	159
B_6	$90 < B_6 < 223$	180	159	134	159	164
G_6	$90 < G_6 < 223$	151	146	157	142	159
Result	Pass/Fail	Pass	Pass	Pass	Pass	Pass

The size of the keyspace then depends on the precision of the computing platform on which the cipher algorithm is implemented. With 32-bit floating-point numbers of the IEEE format, the smallest possible increment (ϵ) is $\epsilon_{32} \approx 1.1921 \times 10^{-7}$. With 64-bit floating-point numbers this value is $\epsilon_{64} \approx 2.2204 \times 10^{-16}$. Let $\hat{\alpha}$ indicate the size of the interval over which α can span i.e. $\hat{\alpha} = 1.41 - 1.16$. Let $\hat{\beta}$, \hat{x}_0 , \hat{y}_0 and \hat{P} have correspondingly similar meaning. Since P is an integer, $\hat{P} = 1000 - 80$. Then, the number of representable values of α on the applicable computing-platform can be computed as $K_\alpha = \hat{\alpha}/\epsilon$. Such a calculation can easily be carried out using logarithms. Similarly, K_β , K_{x_0} and K_{y_0} can also be calculated. $K_P = \hat{P}$. Since the parameters are used together, the size of the keyspace $K = K_\alpha \times K_\beta \times K_{x_0} \times K_{y_0} \times K_P$. The logarithm of this figure, to the base 2, gives an estimate of the length of a single binary key which contains all the information required to generate the pseudo-random sequence. The value of $\log_2(K)$ for 32-bit and 64-bit precision was found to be 97 and 213 respectively. The size of the keyspace can be further increased by increasing the precision of floating-point representation.

Table 5. Parameters used for NIST test samples.

Parameter	U_1	U_2
α	1.40	1.398
β	0.30	0.283
x_0	-1	0.26
y_0	1	0.29
$B_y(-2)$	0	0
$B_y(-1)$	1	1
P	117	111
T	1000	1000

9. Concluding remarks

Though chaotic orbits of discrete-time maps are non-periodic in nature, because of finite precision of digital computers the orbits actually turn out to be periodic. The average period of an orbit of a two-dimensional map can be expected to be longer than that of a one-dimensional map. To overcome the inevitable periodicity in digital computation, Shujun *et al* have used simple LFSR-based perturbation generators to perturb the parameters of the dynamical systems used (Shujun *et al* 2001). Though they have used perturbed one-dimensional maps in couple-chaotic-system-based pseudorandom sequence generators, the same technique should be directly applicable to the present algorithm and is a feasible way of defeating the periodicity.

The periodicity inherent in digital-computer implementations is not a problem in maps realized on an analog-computer. The Hénon map, due to its polynomial form, can be realized in the form of an electronic circuit using analog-multipliers, sample-and-hold blocks and operational amplifiers. Such a realization of the logistic map has already been studied (Suneel 2006). The advantage of an analog realization of chaotic maps is that due to natural variations in circuit parameters and conditions, practically identical systems provided with practically identical conditions generate sequences that quickly diverge and become un-correlated within a short time. Therefore, such implementations may actually turn out to be *truly* random (as opposed to pseudorandom) sequence generators.

The choice of the Hénon map for the work in this paper was rather arbitrary. The author believes that similar results should also be attainable with other two-dimensional maps.

The linear complexity and correlation properties of the proposed sequences suggest a strong similarity to random sequences. Results of statistical tests carried out confirm this further. A large size of the keyspace suggests strong candidature for cryptographic applications, especially for stream cipher cryptography. Possible cryptanalysis techniques for these sequences is an open subject.

The author thanks his colleagues A P Dabhade, K V Suresh, D Venu Gopal and R S Chandrasekhar for discussions.

References

- Crutchfield J P, Farmer J D, Packard N H, Shaw R S 1986 Chaos. *Sci. Amer.* 255: 38–49
- Dachselt F, Schwarz W 2001 Chaos and cryptography. *IEEE Trans. Circ. Sys. - I* 48: 1498–1509
- Falcioni M, Palatella L, Pigolotti S, Vulpiani A 2006 Properties making a chaotic system a good pseudo random number generator, ePrint *arXiv:nlin.CD/0503035*
- FIPS PUB 140-1 1994 Federal information processing standards publication - *Security requirements for cryptographic modules*, U.S. Department of Commerce
- FIPS PUB 140-2 2001 Federal information processing standards publication - *Security requirements for cryptographic modules*, U.S. Department of Commerce
- Gleick J 1997 *Chaos* (London: Vintage)
- Golomb S W 1964 *Introduction to digital communication* in S W Golomb (ed) Digital communications with space applications (Englewood Cliffs: Prentice Hall) 1–16
- Golomb S W 1982 *Shift register sequences* (Laguna Hills: Aegean Park Press) 1–108
- Hénon M 1976 A two-dimensional mapping with a strange attractor. *Commun. Math. Phys.* 50: 69–77

- Kathleen T A, Sauer T D, Yorke J A 1997 *Chaos—An introduction to dynamical systems* (New York: Springer-Verlag) 43–91, 163–166, 238–239
- Keeping E S 1962 *Introduction to statistical inference* (Princeton: D. Van Nostrand) 64–66
- Kippenhahn R 1999 *Code breaking: A history and exploration* (Hyderabad: Universities Press) 201–204
- Knuth D E 1998 *The art of computer programming, Volume 2/Seminumerical algorithms*, 3rd ed (Reading: Addison Wesley) 9–37
- Kocarev L 2001 Chaos-based cryptography: A brief overview. *IEEE Circ. and Sys. Mag.* 1: 6–21
- Kumar N 1996 *Deterministic chaos: Complex chance out of simple necessity* (Hyderabad: Universities Press)
- Massey J L 1969 Shift register synthesis and BCH decoding. *IEEE Trans. Inform. Theory* 15: 122–127
- Menezes A J, van Oorschot P C, Vanstone S A 1997 *Handbook of applied cryptography* (Boca Raton: CRC Press) 198–200
- Mollin R A 2001 *An introduction to cryptography* (Boca Raton: Chapman and Hall/CRC) 115–116
- Peitgen H O, Jürgens H, Saupe D 2004 *Chaos and fractals, New frontiers of science* 2nd ed (New York: Springer-Verlag) 609–627
- Ramasubramanian S 1997 The normal distribution: From binomial to normal. *Resonance* 2: 15–24
- Rukhin A, Soto J, Nechvatal J, Smid M, Barker E, Leigh S, Levenson M, Vangel M, Banks D, Heckert A, Dray J, Vo S 2001 NIST special publication 800-22: *A statistical test suite for random and pseudorandom number generators for cryptographic applications*, National Institute of Standards and Technology
- Shujun L, Xuanqin M, Yuanlong C 2001 Pseudorandom bit generator based on couple chaotic systems and its applications in stream cipher cryptography. *Progress in Cryptology: INDOCRYPT-2001 The 2nd International Conference on Cryptology in India*, Indian Institute of Technology, Madras, Chennai, India
- Suneel M 2006 Electronic circuit realization of the logistic map. *Sādhanā* 31(1): 69–78
- Woodcock C F, Smart N P 1998 p -adic chaos and random number generation. *Experimental Math.* 7: 333–342