Pseudorandom number generator based on Arnold cat map and statistical analysis

Erdinç AVAROĞLU*
Department of Computer Engineering, Faculty of Engineering, Mersin University, Mersin, Turkey

Received: 30.07.2015 • Accepted/Published Online: 05.02.2016 • Final Version: 24.01.2017

Abstract: Pseudorandom number generators (PRNGs) generate random bit streams based on deterministic algorithms. Any bit stream generated with a PRNG will repeat itself at a certain point, and the bit streams will become correlated. As a result, all bit streams generated in this manner are statistically weak. Such weakness leads to a strong connection between PRNGs and chaos, which is characterized by ergodicity, confusion, complexity, sensitivity to initial conditions, and dependence on control parameters. In this study, we introduce a PRNG that generates bit sequences by sampling two Arnold cat map outputs. The statistical randomness of bit streams obtained using this PRNG was verified by statistical analyses such as the NIST test suite, the scale index method, statistical complexity measures, and autocorrelation. The generated bit streams successfully passed all the analytical tests and can be safely used for the many applications of randomness.

Key words: Pseudorandom number generator, Arnold cat map, NIST statistical test, scale index, statistical complexity measure, autocorrelation

1. Introduction

Randomness means being unknown with any certainty. In other words, it means unpredictability in cryptography and a lack of relationship between samples in statistics [1,2]. Random numbers, on the other hand, are numbers that are generated with equal probability so that there is no correlation between the numbers. Random numbers have been widely used in the fields of simulation, sampling, numerical analysis, decision-making, entertainment, computer programming, and cryptography. Random numbers must be unpredictable and irreproducible (nonperiodic) with good statistical properties and uniform distribution [3].

Various number generators such as pseudorandom number generators (PRNGs), true random number generators (TRNGs), and hybrid random number generators have been developed to generate random numbers. TRNGs generate random numbers by using a physical process as a source of noise. A hybrid random number generator is a random number generator that uses random numbers generated by a TRNG as a seed in a PRNG [4]. A PRNG uses a seed obtained from a source of entropy as the random input and generates bit streams that cannot be distinguished from those generated by TRNGs by means of calculations. Since these generators are deterministic, the output value cannot exceed the seed value introduced. Moreover, their streams repeat themselves after a while, which means that the system is periodic. A nonperiodic system requires a rather large memory. However, this requirement causes the system to slow down. The system safety of PRNGs depends on the unpredictability of seeds and the complexity of the functions used in the system. Moreover, the parameters of

*Correspondence: eavaroglu@gmail.com
the functions must be carefully selected. The algorithms used by PRNGs include linear feedback shift registers, midsquare methods, and linear congruential generators [3]. Aside from these generators, chaos, particularly, is used to generate random numbers in PRNGs due to its properties. These are [5]:

- Ergodicity and confusion
- Sensitivity to initial condition/control parameter
- Deterministic dynamics (deterministic processes lead to random-like (pseudorandom) behaviors)
- Structure complexity and algorithm complexity

Dependence on extreme initial conditions makes chaotic systems very attractive for PRNGs. Chaos is used for many PRNG designs such as piecewise linear maps [6–9], logistic maps [10,11], and z-logistic maps [12]. In recent studies, the design of chaos-based PRNGs continues. Sun and Lu designed a PRNG system based on spatial chaotic maps for cryptographic use [13]. Patidar et al. developed a PRNG design that generates bit streams through a comparison of outputs obtained with two chaotic standard maps with different initial conditions [14]. Guyeux et al. developed a PRNG design based on chaotic iterations and a combination of XORShift and ISAAC generators [15], and Pareek et al. developed a PRNG design based on a cross-coupled chaotic tent map [16]. López et al. developed a PRNG design based on coupled chaotic map (a modification of a piecewise linear map) [17]. For the PRNG design offered in [18], deterministic chaotic systems were used. Using three logistic maps, François and Defour modified Patidar’s [14] PRNG design based on two logistic maps and showed that their results were more successful [19]. François et al. made PRNG designs composed of a chaotic logistic map [20] and a mixture of three chaotic maps [21].

These studies indicate a strong link between chaos and PRNGs. Especially for PRNG design, chaotic maps have been used most. This is because a chaotic map is fast and iterative. Therefore, this study introduces a PRNG design based on a chaotic Arnold cat map. Using two Arnold cat maps, a bit stream was generated by passing the random numbers through the compare section. An output bit stream was generated by sampling the generated bit streams according to the selection rule in the sampler section. The NIST 800-22 test suite was used to verify that the bit stream generated is statistically random. In addition, the scale index method, statistical complexity measures, and a correlation test were used to determine whether the generated bit streams are nonperiodic and correlated.

This article is organized as follows: Section 2 introduces the Arnold cat map. Section 3 describes the PRNG design. Section 4 describes the statistical analyses used with the generated bit streams and presents the results of these analyses. Finally, Section 5 evaluates these results.

2. Arnold cat map

Recently, chaos has been commonly used in random number generators [22–26]. This is because the generation and storage of chaotic number streams has proven to be fast and easy and does not require storing long number streams. Only a few functions (chaotic maps) and a few parameters (initial conditions) are sufficient even for very long streams. Moreover, it is possible to generate a large number of number streams easily by simply changing the initial conditions. These advantages have led to using chaos as a random number generator [27].

This study used the Arnold cat map discovered by Arnold and Avez [28]. The map is named after Vladimir Arnold, who demonstrated its effects in the 1960s using an image of a cat, as in Figure 1.
To define the Arnold cat map, we first need to define a torus and phase space. A torus is the surface obtained by revolving a circle in three-dimensional space around a disconnected axis that is coplanar with the circle. A phase space represents all possible states of a system, and each state corresponds to one unique point. The map can be now defined as a discrete system in which the trajectories in phase space are stretched and folded to obtain a torus. The mathematical definition of the Arnold cat map is shown in Eqs. (1)–(4).

Let $X = \begin{bmatrix} x \\ y \end{bmatrix}$. $X$ is an $n \times n$ matrix, and the Arnold cat map transformation is as follows:

$$
\Gamma : \begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \mod n = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \mod n
$$

(1)

$$
\Gamma : (x, y) \rightarrow (x + y, x + 2y) \mod n
$$

(2)

This system can be represented like this:

$$
x_m = (2x_m + y_m) \mod n
$$

(3)

$$
y_m = (x_m + y_m) \mod n
$$

(4)

Since this is a chaotic map, made of a discrete system, it expresses the dynamics of chaos. The initial conditions will affect the map, and its outputs will appear to be random.

3. The proposed PRNG

A random number generator is a tool, a natural source or an algorithm. The random numbers generated by a random number generator are expected to be unpredictable and irreproducible with good statistical properties. Some generators use natural sources (nondeterministic), while others use specific algorithms (deterministic). Any bit stream generated with a deterministic system will repeat itself at a certain point, and the bit streams will become correlated. As a result, all bit streams generated in this manner are statistically weak. To resolve these problems, this section introduces a PRNG that uses the Arnold cat map. Figure 2 shows the proposed PRNG design.
The proposed PRNG is composed of three sections: the random number generation section, the compare section, and the sampler section. In the random number generation section, Arnold cat map 1 and Arnold cat map 2 use different initial conditions to generate random numbers. In this study, random bit streams were generated for 100 different initial conditions and then they were analyzed. However, for the results presented in this article the initial conditions are $x_0 = 0.9$, $y_0 = 0.4$, $z_0 = 0.3$, and $w_0 = 0.6$, and mod 1 is used since we want the generated numbers to be in the range of $[0, 1]$.

The bit stream is generated by comparing the random numbers generated in the random number generation section in the compare section following the rules given in Eqs. (5) and (6).

For Arnold cat map 1:  
$$g(x_{m+1}, y_{m+1}) = \begin{cases} 1 & \text{if } x_{m+1} \geq y_{n-1} \\ 0 & \text{if } x_{m+1} < y_{n-1} \end{cases}$$  
(5)

For Arnold cat map 2:  
$$g(z_{m+1}, w_{m+1}) = \begin{cases} 1 & \text{if } z_{m+1} \geq w_{n-1} \\ 0 & \text{if } z_{m+1} < w_{n-1} \end{cases}$$  
(6)

Here, $n$ is the length of the bit stream.

In the sampler section, the bit streams generated in the compare section are sampled by using the sampler rule in Eq. (7) to obtain the output bit stream $b_i$.

$$\text{sampler rule} = \begin{cases} b_i & \text{if } a_i = 1 \\ \text{discard} & \text{if } a_i = 0 \end{cases}$$  
(7)

Table 1 shows an example of the sampler rule in use.

| $a_i$: Generated bit stream with Arnold cat map 1 | 10111000111011011011011010 |
| $b_i$: Generated bit stream Arnold cat map 2 | 00101110110011101010101111 |
| Sampler rule | 0 -101 — 110 -11- 1-10- 010- 1- |
| Output bit stream | 0101110111100101 |
4. Statistical analysis
Any bit stream generated by PRNGs is expected to be highly random, unpredictable, irreproducible, and uncorrelated. These properties, however, must be analyzed carefully. This section analyzes the bit stream generated by the proposed PRNG.

4.1. Randomness test (NIST test suite)
The applicable statistical tests are used to determine whether a generated bit stream is random or not. For this purpose, various test suites such as FIPS140 [30], NIST [1], Diehard [31], TestU01 [32], and others have been developed. The best test known to date is the NIST 800-22 test suite released by the National Institute of Standards and Technology. The NIST 800-22 test suite includes 15 tests and requires success in all of these tests. The studies of PRNGs with chaotic maps in the literature are declared successful when they pass all the NIST tests successfully. In this study, the bit streams generated by Arnold cat map 1 and Arnold cat map 2 did not pass all 15 tests. However, the bit streams obtained by sampling the bit streams from Arnold cat map 1 and Arnold cat map 2 according to the sampler rule did successfully pass the entire NIST test suite. The test results obtained are presented in Table 2. Table 2 also contains the NIST test success rate of bit streams generated from 100 different initial conditions.

4.2. Scale index method
The scale index technique was proposed by Benitez [33]. This technique enables the obtaining of information about the periodic nature of generated number series. The technique, which depended on the continuous wavelet transform (CWT) and wavelet multiresolution analyses, is given below [34]. The scale $s$ and $f$ at time $u$ in the CWT and scalogram are defined as shown in Eqs. (8) and (9) [24,33,34].

$$W_f(u,s) := \langle f, \phi_{u,s} \rangle := \int_{-\infty}^{+\infty} f(t) \phi_{u,s}^*(t) dt \quad (8)$$

$$S(s) := ||W_f(u,s)|| = \left( \int_{-\infty}^{+\infty} |W_f(u,s)|^2 du \right) \quad (9)$$

$S(s)$ is the CWT's energy. The inner scalogram is defined by Eq. (10).

$$S^{in}(s) := ||W_f(u,s)||_{j(s)} = \left( \int_{\varepsilon(s)}^{d(s)} |W_f(u,s)|^2 du \right)^2 \quad (10)$$

Here, $J(s) = [c(s), d(s)] \subseteq I$ indicates the maximal subinterval in $I$. The support of $\psi_{u,s}$ is included in $I$ for all $u \in j(s)$. $S^{in}$ as defined in Eq. (11) is normalized.

$$\tilde{S}^{in}(s) = \frac{S^{in}(s)}{(d(s) - c(s))^2} \quad (11)$$

The scale index of $f$ in the scale interval $[s_0, s_1]$ can be described as shown in Eq. (12).

$$i_{scale} := \frac{S(s_{min})}{S(s_{max})} \quad (12)$$

637
### Table 2. NIST test results of the proposed PRNG and success rates for 100 different initial conditions.

<table>
<thead>
<tr>
<th>Test</th>
<th>Arnold cat map 1</th>
<th>Arnold cat map 2</th>
<th>Output bit streams (b)</th>
<th>Success rates for 100 different initial conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>P-value Result</td>
<td>P-value Result</td>
<td>P-value Result</td>
<td>Result Proportion</td>
<td></td>
</tr>
<tr>
<td>Frequency</td>
<td>-</td>
<td>Not passed</td>
<td>0.980 Passed</td>
<td>0.95</td>
</tr>
<tr>
<td>Frequency test within a Block</td>
<td>-</td>
<td>Not passed</td>
<td>0.664 Passed</td>
<td>0.95</td>
</tr>
<tr>
<td>Runs</td>
<td>-</td>
<td>Not passed</td>
<td>0.117 Passed</td>
<td>0.95</td>
</tr>
<tr>
<td>Test for the longest run of ones in a block</td>
<td>-</td>
<td>Not passed</td>
<td>0.222 Passed</td>
<td>0.95</td>
</tr>
<tr>
<td>Binary matrix rank</td>
<td>0.988</td>
<td>Passed</td>
<td>0.456 Passed</td>
<td>0.95</td>
</tr>
<tr>
<td>Discrete Fourier transform</td>
<td>-</td>
<td>Not passed</td>
<td>0.283 Passed</td>
<td>0.95</td>
</tr>
<tr>
<td>Nonoverlapping template matching</td>
<td>-</td>
<td>Not passed</td>
<td>0.024 Passed</td>
<td>0.95</td>
</tr>
<tr>
<td>Overlapping template matching</td>
<td>-</td>
<td>Not passed</td>
<td>0.0117 Passed</td>
<td>0.95</td>
</tr>
<tr>
<td>Maurer’s universal statistical</td>
<td>-</td>
<td>Not passed</td>
<td>0.117 Passed</td>
<td>0.95</td>
</tr>
<tr>
<td>Linear complexity</td>
<td>0.153</td>
<td>Passed</td>
<td>0.295 Passed</td>
<td>0.95</td>
</tr>
<tr>
<td>Serial 1</td>
<td>-</td>
<td>Passed</td>
<td>-</td>
<td>0.95</td>
</tr>
<tr>
<td>Serial 2</td>
<td>-</td>
<td>-</td>
<td>0.203 Passed</td>
<td>0.95</td>
</tr>
<tr>
<td>Approximate entropy</td>
<td>-</td>
<td>Not passed</td>
<td>-</td>
<td>0.90</td>
</tr>
<tr>
<td>Cumulative sums</td>
<td>-</td>
<td>Not passed</td>
<td>0.992 Passed</td>
<td>0.95</td>
</tr>
<tr>
<td>Random excursion</td>
<td>-</td>
<td>Not passed</td>
<td>0.028</td>
<td>0.90</td>
</tr>
<tr>
<td>Random excursion variant</td>
<td>-</td>
<td>Not passed</td>
<td>0.028</td>
<td>0.90</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Passed</td>
<td>0.028</td>
<td>0.90</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

i\textsubscript{scale} should be in the range of 0 ≤ i\textsubscript{scale} ≤ 1. If the i\textsubscript{scale} obtained from the proposed system is 0 or near 0, the system is nonperiodic. If it is 1 or near 1, the system is defined as periodic.

The scale index plot of the 100 different bit streams obtained from the proposed PRNG is shown in Figure 3. All the values are 1 or near 1. This shows that the bit streams are nonperiodic.

### 4.3. Statistical complexity measure

The statistical complexity measure (SCM), based on H-disorder and Q-disequilibrium, was introduced in previous studies [34,35]. The SCM is a quantifier of the probability distribution P. The SCM is expressed by Eq. (13).
$$C[P] = Q[P] * H[P]$$  \hspace{1cm} (13)

For the probability distribution $P = \{P_i, I = 1 \ldots N\}$ and its information measure $S$, the disorder $H$ is defined as in Eq. (14).

$$H[P] = S[P]/S_{max}$$  \hspace{1cm} (14)

$S_{max} = S[P_e]$, where $P_e = [1/N, \ldots, 1/N]$ is the equilibrium distribution so that $(0 < H[P] < 1)$ is true. $S$ is the Shannon entropy as in Eq. (15).

$$S[P] = -\sum_{i=1}^{N} P_i \log_2 P_i$$  \hspace{1cm} (15)

$Q$ is defined as the distance in probability space. To calculate $Q$ with Eq. (16), Euclidean distance ($D_E$) is used.

$$Q[P] = Q_E[P, P_e] = Q_0^{(E)}D_E[P, P_e] = Q_0^{(E)}\sum_{i=1}^{N} \left\{P_i - \frac{1}{N}\right\}^2$$  \hspace{1cm} (16)

$Q_0^{(E)} = N/(N-1)$ such that $0 \leq Q_E \leq 1$.

For nonperiodic streams, SCM and $Q$ must be zero or near zero, while $H[P]$ must be 1 or near 1.

The results of the SCM performed with 16-bit and 32-bit blocks of the bit streams obtained from the proposed PRNG are presented in Figures 4 and 5, respectively.

The results show that for both the 16-bit and 32-bit blocks the SCM and $Q$ are zero or near zero, while $H[P]$ is 1 or near 1. This shows that the generated bit stream is nonperiodic.

### 4.4. Autocorrelation test

Correlation is an indication of a linear relationship between two variables or more and has a value between $+1$ and $-1$. If the value is 0 or near 0, then there is no linear relationship between the variables. The aim of this test is to check the correlation between the generated bit stream $b_i$ and its shifted version. We assume that $d$ is a constant integer and $1 \leq d \leq (n/2)$. The mathematical expressions of the test are given in Eqs. (16) and (18) [36].

$$A(d) = \sum_{i=0}^{n-d-1} b_i \oplus b_{i+d}$$  \hspace{1cm} (17)
Figure 4. The 16-bit statistical complexity measure results: (a) Shannon entropy, (b) disequilibrium, (c) statistical complexity.

Figure 5. The 32-bit statistical complexity measure results: (a) Shannon entropy, (b) disequilibrium, (c) statistical complexity.

Here, $\oplus$ is the XOR operator and $n$ is the length of the bit stream. The following random variable is defined:

$$ X_5 = \frac{2[A(d) - (n - d)/2]}{\sqrt{n - d}} $$  \hspace{1cm} (18) $$

If $\{b_i\}$ is a true random stream and $n \to \infty$, this random variable has a normal distribution $N(0, 1)$. Assuming $\alpha = 0.05$, if $|X_5| < 1.6449$, the test is successful [36].

The correlation test results for the proposed PRNG are shown in Table 3.
### Table 3. Autocorrelation test results for the proposed PRNG.

<table>
<thead>
<tr>
<th>Test</th>
<th>D value</th>
<th>X5 value</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autocorrelation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>-0.175</td>
<td>0.175</td>
<td>Success</td>
</tr>
<tr>
<td>10</td>
<td>0.363</td>
<td></td>
<td>Success</td>
</tr>
<tr>
<td>13</td>
<td>-0.527</td>
<td>0.527</td>
<td>Success</td>
</tr>
<tr>
<td>20</td>
<td>0.518</td>
<td></td>
<td>Success</td>
</tr>
<tr>
<td>25</td>
<td>-0.585</td>
<td></td>
<td>Success</td>
</tr>
<tr>
<td>100</td>
<td>0.576</td>
<td></td>
<td>Success</td>
</tr>
<tr>
<td>500</td>
<td>-0.076</td>
<td></td>
<td>Success</td>
</tr>
<tr>
<td>1000</td>
<td>-0.06</td>
<td></td>
<td>Success</td>
</tr>
</tbody>
</table>

5. Conclusion

This study introduces a PRNG design based on the Arnold cat map. The randomness of the bit streams generated by the proposed PRNG were measured using specific analytical methods. The results of these analyses show the bit streams generated by means of Arnold cat map 1 and Arnold cat map 2 did not pass the NIST tests, but they managed to pass the tests successfully after sampling according to the proposed system’s sampler rule. The bit streams generated were checked for nonperiodicity, and all 100 bit streams displayed nonperiodicity in the scale index. Furthermore, the SCM results show that the system is not periodic. The final analysis searched for a correlation between the zeros and ones in the generated bit stream and showed that there is no such correlation. In conclusion, the bit streams generated by the proposed PRNG can be used in a variety of applications. The analyses have shown that the proposed system can also be used in cryptographic systems since it is simple and user-friendly and it can generate high-quality random number streams. The only drawback of the proposed system is that the bit rate decreases after sampling.

References

