Improving Middle Square Method RNG Using Chaotic Map

Hamed Rahimov¹, Majid Babaie¹, Hassan Hassanabadi²
¹Department of Computer Engineering, Shahrood University of Technology, Shahrood, Iran
²Physics Department, Shahrood University of Technology, Shahrood, Iran
E-mail: hrahimov@shahroodut.ac.ir, hrahimov@gmail.com
Received January 23, 2011; revised March 10, 2011; accepted March 12, 2011

Abstract

One of the classic approaches in PRNGs is the middle square method in which with a simple mathematical model generating pseudorandom numbers in high speed and minimum correlation between output numbers. Despite these unique characteristics, the method contains weaknesses that a broader application of this algorithm will face. In this paper is studied middle square method and then a logistic chaotic map is introduced with its specific features and its improved weaknesses via using these characteristics. Finally the NIST tests suite s are presented, in order to detect the specific characteristics expected from truly random sequences.

Keywords: Middle Square Method (MSM), Random Number Generator, Logistic Map, NIST Tests Suite

1. Introduction

Nowadays, the world is connected together by networks. The main problem in the present networks is the security protocols. This is due to growing interest in the use of digital chaotic systems offering the possibility to support the security of cryptographic algorithms for data encryption, like those presented in the networking security protocols [1,2].

The best ways to increase security in modern communication methods is using the cryptography algorithms. Most of these new algorithms use chaotic maps as pseudorandom number generators (PRNGs) [3,4] to obtain a binary stream including symmetric encryption algorithm [5]. PRNGs are very important in several fields like statistical studies, simulations, cryptography and solving specific mathematical problems [6-8]. They may be based on unpredictable physical resources such as physical phenomenon or on mathematical algorithms like “Halton algorithm” [6]. However, truly random numbers (i.e. physical phenomenon) are not gained by a specific algorithm [8,9] as they are used in the encryption systems [9].

In order to clarify the purpose of the proposed algorithm, one can see some of methods in References [13-27].

With this background, it is understood that in the recent years the PRNG’s algorithm developed by some
physical algorithm such as chaos theory and produced random numbers rapidly and with less correlation [13-27]. This paper investigates an efficient method to produce random numbers with the combination of middle Square Method (MSM) and chaotic logistic map. Hence in the next section, we describe the random number’s application in the cryptography algorithm.

2. Middle Square Method

Middle Square Method is a one of the best methods of generating pseudorandom numbers. But in the some practical situations it is not a good method, since its period is usually very short (as shown in the Figure 2 and Figure 3). This mathematical algorithm was described by John von Neumann in 1949 [28].

In this method for generating a sequence of ten digit pseudorandom numbers, in the first step a four digits starting value is created and squared (e.g. $A \times A = C$) so that an eight digits number is produced. Then by a simple mathematical model generated the middle four digits of the result and it would be the next number in the sequence (e.g. $D = C \mod 10^{n/2}$) and returned as the result. By repeating this process we are able to generate more random numbers. One of the main problems occurs when the middle n/2 digits (in this example, four digits) are all zero and the generator yields zero outputs forever. For testing this algorithm we take some random numbers and describe the weaknesses in our examples which would describe $a = 8439$ for first example (as shown in Figure 2) and $a = 6395$ for second example (as shown in Figure 3) [28].

Sequence one, with core number (8439) according with Figure 2. After generating 214 random numbers we have an unlimited loop generating “0”.

Bearing in mind the limited random numbers generated by this method, in this paper proposed a novel chaotic system based on logistic map.

Most of these discrete chaotic cryptosystem algorithms use a chaotic map as PRNGs to generate a keystream which is used for encryption of a plaintext message to produce a ciphertext [15]. The secret key of such systems constitutes the initial values and/or the system parameters [5]. In this paper, a novel sequence generator is proposed which is based on a single one-dimensional logistic map.

Sequence three, with core numbers (6395); according with Figure 3 after generating 216 random numbers we have a loop with length 602 (i.e. where numbers: “217” and “819”, we have generated “0.125”, so the length of the loop is 602).

This paper is structured as follows. In Section 3, the properties of the logistic map are discussed. In Section 4, a complete description of the chaotic middle square method is presented. The statistical properties of these sequences are discussed in Section 5 and finally the result of NIST test presented in Table 1.

3. Chaotic Logistic Map

As we know, the logistic map is the one of the simplest
and most studied nonlinear that was introduced as a demographic model. The logistic map is given by that is shown in Equation (1):

\[ f(x_n) = x_{n+1} = r \cdot x_n \cdot (1 - x_n) \] (1)

Where \( x_n \) is the state variable, which generates values in the interval \([0, 1]\) and \( r \) is called system parameter which can have any value between one and four \([10, 29]\).

There is a non-trivial fixed point where \( 3.4 \leq r \leq 4 \) which is seen in Figure 4. Finally for \( r = 4 \), we observe that chaos values are generated in the complete range of \([0, 1]\). In this paper, the logistic map has application of

<table>
<thead>
<tr>
<th>Test</th>
<th>P-value</th>
<th>Pass rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>0.804645</td>
<td>0.9890</td>
</tr>
<tr>
<td>Block-Frequency</td>
<td>0.322901</td>
<td>0.9900</td>
</tr>
<tr>
<td>CuSums-forward</td>
<td>0.265567</td>
<td>0.9900</td>
</tr>
<tr>
<td>CuSums-backward</td>
<td>0.090388</td>
<td>0.9895</td>
</tr>
<tr>
<td>Rans</td>
<td>0.569766</td>
<td>0.9915</td>
</tr>
<tr>
<td>Long run</td>
<td>0.066673</td>
<td>0.9940</td>
</tr>
<tr>
<td>Rank</td>
<td>0.248649</td>
<td>0.9915</td>
</tr>
<tr>
<td>FFT</td>
<td>0.000159</td>
<td>0.9995</td>
</tr>
<tr>
<td>Overlapping templates</td>
<td>0.906745</td>
<td>0.9905</td>
</tr>
<tr>
<td>Universal</td>
<td>0.971354</td>
<td>0.9885</td>
</tr>
<tr>
<td>Approximate entropy</td>
<td>0.558502</td>
<td>0.9925</td>
</tr>
<tr>
<td>Serial 1</td>
<td>0.357000</td>
<td>0.9875</td>
</tr>
<tr>
<td>Serial 2</td>
<td>0.864494</td>
<td>0.9905</td>
</tr>
<tr>
<td>Liner complexity</td>
<td>0.671820</td>
<td>0.9920</td>
</tr>
</tbody>
</table>

Figure 4. Bifurcation plot of the Logistic map in \( x_n \in [0,1] \) \( r \in [3.4,4] \).

4. Proposed Chaotic PRNG

Middle square method has some big problems \([28]\) that were mentioned in Section 3 with the notice of these problems in MSM and consider to unique properties of chaotic logistic map in this section presented a chaotic solution for solving these problems.

When efficient values are chosen for numbers in the core that have no loop with the specific length, next we can correct the unlimited loop (when PRNG joust “0” generated in sequence) by exchanging one of the core’s numbers \([28]\).

Assume the logistic chaotic map with \( r = 4 \) in \( x_0 = 0.6 \) Generates a random number and replaces it with a number in \( A \)’s place and continues generating random numbers without any unlimited loop and the loop with a particular length. These problems can be solved by implementation of chaotic iteration that is generated in parallel.

Middle square method algorithm described in Equation (2) also seen below, having the number (i.e. \( A \)).

\[
\begin{align*}
C &= A \times A \Rightarrow D = C^n10^2 \\
E &= (C-D)/10^2 \Rightarrow Result = E\%10^n \\
Num_t &= Result/10^n \\
&\text{if } (Num_t == 0) \text{ then } \\
&\{ \\
x_{n+1} &= x_n r (1-x_n) \\
B &= x_{n+1} \\
x_n &= x_{n+1} \\
&\} \\
\end{align*}
\]

5. NIST Statistical Tests Suite

The NIST tests suite is a statistical package involving 15 tests and is based on hypothesis testing. Also The NIST tests suite focus on a variety of different types of non-randomness. In the present case, the tests consist determining whether or not a specific sequence of zeroes and ones is random \([8]\). The theoretical background of its distribution is based on null hypothesis which is determined by mathematical methods and corresponding probability value (P-value).

The P-value is the probability that a perfect RNG for each test would produce a sequence less random than the sequence that was tested, given the kind of non-randomness assessed by the test. If the P-value for a test is determined to be equal zero, that means the sequence ap-
pears to be exactly non-random and a P-value to be 1, then the sequence has perfect randomness [8].

6. Conclusions

In this paper we have proposed a pseudorandom number generator (PRNG) based on the combination of chaotic logistic map and Middle Square Method, the chaotic system iterated independently starting from initial conditions could help to generate appropriate values; we have also tested the generated sequences using the NIST tests to detect the unique characteristics expected from truly random bit sequences. The results of statistical testing are reliable so has been used as a part of encryption system to generate secret key and have an efficient algorithm. Finally the result of their statistical tests presented in the Table 1.

7. References


[23] V. V. Kolesov, R. V. Belyaev and G. M. Voronov, “A Digital Random-Number Generator Based on the Chaotic


