International Conference on Computational Science, ICCS 2010

A CA Randomizers Based on parallel CAs with Balanced Rules

Farhad Maleki*, Ali Mohades, M. E. Shiri, Afsane Bijari

Faculty of Information Technology, Payame Noor University, Nakhel Ave, Minicity, 19395-4697, Tehran, Iran
Department of Computer Science, Faculty of Mathematics and Computer Sciences, Amirkabir University of Technology, 424 Hafez Ave, 15914 Tehran, Iran

Abstract

In this paper a class of cellular automata rules is defined and proposed to be used in CA randomizers. The quality of the proposed rules is shown by study of symmetric rules of radius one and two. In addition, a non-uniform CA randomizer is constructed with the proposed symmetric rules of radius two. The high quality of the generated random numbers are shown by a battery of statistical tests. Moreover, it is shown that the proposed CA randomizer is more secure against cryptanalysis attacks.

Keywords: Cellular automata; Rule; Genetic Algorithm; Pseudorandom number Generator; CA Randomizer; Cryptography; Stream Ciphers;

1. Introduction

Random numbers are needed in a variety of applications including Cryptography, Monte Carlo, Simulation, etc. High quality random numbers are also essential for cryptographic applications. There are two classes of key-based cryptosystems, symmetric and asymmetric key [1]. Cellular Automata (CAs) have been proposed for asymmetric key cryptosystems by Guam [2] and Gutowitz [3].

Symmetric cryptosystems are categorized into Block ciphers and Stream ciphers [1]. In the context of Block ciphers, Cellular Automata (CAs) have been widely studied [4, 5]; in addition, we have proposed an image encryption system by special kind of cellular automata [6]. In the area of stream ciphers CAs have been used as random number generators [7, 8, 9, 10, 11]. At first, Wolfram proposed a random number generator by one-dimensional uniform CA with rule 30 [7], and later, non-uniform CA randomizers based on rules 90 and 150 have been proposed to construct a stream cipher [8, 9] and it was shown that the quality of generated random numbers was better than the quality of generated random numbers by Wolfram. In addition, an evolutionary technique called cellular programming has been used to find appropriate rule for non-uniform CAs and, as a result, four rules 90, 105, 165, and 150 have been used to construct a high quality random number generator [10]. Because of the huge size of the secret key space the proposed scheme was more robust against direct cryptanalysis. Beside, cellular programming have been used to find a combination of rules of radius one and two which construct a random number

* Corresponding author. Tel.: +98 918 7401022; fax: +988325227290.
E-mail addresses: farhad.maleki@aut.ac.ir & farhad.maleki@yahoo.com.

© 2012 Published by Elsevier Ltd. Open access under CC BY-NC-ND license.

doi:10.1016/j.procs.2010.04.045
generator that was more robust against direct cryptanalysis [11].

In the field of cryptography by cellular automata, some rules have been used, but there is no special group or class of appropriate rules which can be proposed to be used in CA randomizers. The main concern of this paper is to define and propose a special class of rules which can be used in CA randomizers to produce high quality random sequences. In addition, the quality of these rules is examined by study of appropriate rules of radius one and two for CA randomizers. Besides, a non-uniform CA randomizer is constructed with the proposed symmetric rules of radius two, and the high quality of generated random sequences is shown by standard statistical tests.

The paper is organized as follows. Section 2 gives an overview of CAs. Section 3 presents the basic concepts of genetic algorithm. In section 4, k-imbalanced and balanced rules are defined; in addition, rules of radius one for a one-dimensional Boolean CA are ranked, and it is shown that the high-ranking rules are balanced rules or rules with low degree of imbalancity. In addition, High quality rules of radius two are studied by the use of Genetic algorithm. Section 5 presents a one-dimensional non-uniform CA randomizer by the proposed rules in section 4. In Section 6, the high quality of the proposed CA randomizer is shown by a battery of statistical tests. Finally, section 7 ends the paper with a short discussion and conclusion.

2. One-Dimensional Cellular Automata

A One-Dimensional finite CA is a discrete dynamical system formed by a one-dimensional array of \( n \) identical objects called cells [12]. Each cell takes its value from a finite set \( S \), called the State Set. A CA is named Boolean if \( S = \{0, 1\} \); the \( i \)-th cell is denoted by \(<i>\), and the state of cell \(<i>\) at time \( t \) is denoted by \( a_i^t \). For each cell \(<i>\) called central cell, a symmetric neighborhood of radius \( r \) is defined by equation 1:

\[
V_i = \{i-r, i-r+1, \ldots, i, i+r\}
\]

(1)

The CA evolves deterministically in discrete time steps and the value of each cell \(<i>\) is updated by a local transition function \( f_i \), called rule, which for a symmetric neighborhood with radius \( r \) is defined as follows (equation 2):

\[
a_i^{t+1} = f(a_{i-r}, \ldots, a_i, \ldots, a_{i+r})
\]

(2)

or equivalently by equation 3:

\[
a_i^{t+1} = f(V_i^{t+1})
\]

(3)

Such that \( V_i \) is as follows (equation 4):

\[
V_i = (a_{i-r}, \ldots, a_i, \ldots, a_{i+r})
\]

(4)

To represent a symmetric rule of radius \( r \) for a Boolean CA, a binary string of length \( L \) is used. Where \( L=2^{2r+1} \). It is obvious that the number of all symmetric rules of radius \( r \) is equal to \( 2^L \). Table 1 shows the rule 90 of radius one \((r=1)\). The number of all symmetric rules of radius one is equal to \( 2^8 \). The rule space grows rapidly such that the length of a symmetric rules of radius two is \( 32(2^{2r+1}) \), and the number of these rules is equal to \( 2^{32} \).

Table 1. The Rule representation of Boolean symmetric rule 90 of radius one

<table>
<thead>
<tr>
<th>Neighborhood Number</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V_i )</td>
<td>111</td>
<td>110</td>
<td>101</td>
<td>100</td>
<td>011</td>
<td>010</td>
<td>001</td>
<td>000</td>
</tr>
<tr>
<td>( f(V_i) )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

If all CA cells obey the same rule, then the CA is said to be a uniform CA; otherwise, it is a non-uniform CA; in addition, a CA is said to be a CA with periodic boundary condition if the extreme cells are adjacent to each other (Fig 1). Also the state of all cells at time \( t \) is called configuration of CA at time \( t \) and it is denoted by \( C_t \). If a CA rule involves only XOR logic, it is called a linear rule; rules involving XNOR logic are referred to complemented rules. A CA with all cells having linear rules is called linear CA, whereas a CA having a combination of linear and complemented rules is called an additive CA [13]. Fig 1 shows a one-dimensional Boolean CA with rule 90,
boundary condition (as it depicted by blue arrow in Fig. 1) and with 8 cells. The configuration of CA at time \( t \) and \( t+1 \), and also a symmetric neighborhood of radius one dashed box) is depicted.

![Fig. 1. A one-dimensional Boolean CA with rule 90, boundary condition and 8 cells](image)

### 3. Genetic algorithm

Genetic algorithm (GA) is often used to find solutions in large discrete search spaces that are too big to iterate completely [14, 15]. Because of the fact that the search spaces growth exponentially with the radius of CA rules, we use GA for rule discovery in CAs.

GA can be regarded as a randomized search procedure that is commonly used to solve the optimization problems. A solution in the problem domain corresponds to an individual in a GA which is represented by a chromosome containing many genes. An objective function called the fitness function is used to evaluate the quality of each chromosome. In general, GA is mainly comprised of the following three operators, namely, reproduction, crossover, and mutation. Reproduction retains the current chromosome's genes, crossover assembles existing genes into new combinations, and mutation produces new genes. The procedure of GA is started by specifying an initial population in the first generation, and during each next generation, the individuals in the population undergo the activities of reproduction, crossover and mutation, to produce their offspring. Then a fitness function is applied to each offspring to determine its quality. The individuals with high quality will survive and form the population of the next generation. The process will repeat for many times until a predefined requirement is satisfied, or a constant number of iterations are exceeded.

### 4. Imbalanced and balanced rules

An appropriate rule for a CA randomizer needs to preserve the randomness of a random configuration during evolution steps. In this section, we define a class of CA rules and propose them as appropriate rules for CA randomizers.

**Definition 1.** Let \( R \) be a CA rule of radius \( r \). Degree of imbalancity for \( R \) is denoted by \( I_r(R) \) and defined as follows (equation 5):

\[
I_r(R) = 2^{2r} - \tau
\]

\[
\tau = \sum_{i=0}^{2^r-1} a_i \quad \text{and} \quad R = a_{2^r-1}a_{2^r-2}...a_1a_0
\]

\((a_{2^r-1},...,a_1,a_0\) is the binary rule string of rule \( R \))
Definition 2. Let $R$ be a CA rule of radius $r$. $R$ is a balanced rule if $I_r(R) = 0$ otherwise, $R$ is an Imbalanced rule.

Definition 3. Let $R$ be a CA rule of radius $r$. $R$ is $k$-Imbalanced if $I_r(R) = k$.

For example, if $R_1 = 01011000$ and $R_2 = 01011010$ both of radius one ($r=1$), then $I_1(R_1) = 1$ and $I_1(R_2) = 0$. $R_1$ is a $1$-Imbalanced rule and $R_2$ is a balanced rule.

4.1. High Quality Rules and Balanced Rules of Radius One

To compare high quality rules and balanced rules, we need to define high quality rules first. It is obvious that a rule $R$ is a high quality rule if it can produce high quality random sequences when it is used as a CA rule for a CA Randomizer. Unfortunately, we do not have any unambiguous definition of randomness, so we have to use some criteria to overcome the ambiguity in the concept of high quality random sequences.

Entropy has been used as a criterion to overcome this ambiguity [9, 10]. To calculate the entropy of a random sequence $S$, it is divided into subsequences of length $h$. Let $L$ be the number values which each element of sequence can take, so $L^h$ is the number of all states for a subsequence of length $h$. In a random sequence, we expect to have all subsequences with almost equal probability of $1/h$ and therefore the maximum entropy. The entropy can be calculated by equation 6:

$$E_h(S) = \sum_{i=1}^{L^h} p_{h_j} \left( \log(p_{h_j}) \right)$$

Where $p_{h_j}$ is a measured probability of occurrence of a subsequence $h_j$ in $S$. The Entropy achieves its maximal value ($E_h = h$) when the probability of each possible sequence of length $h$ is $1/L^h$. All CA rules are ranked, for two uniform one-dimensional CAs with 50 and 64 cells, by Rule Test Scheme which is as follows:

**Rule Test scheme**

**Input:** $R$ as a CA rule for $C$ (a CA with $N$ cell)

**Output:** $q$ as a real number to show the quality of $R$

**Begin**

Set $S$ as an empty string
$q = 0$

for $i := 1$ to $k$

Begin
Set $C$ with a random initial configuration
for $j := 1$ to $M$

Begin
evolve $C$ with rule $R$ one time
Set $S(j) =$ state of central cell of $C$
End
$q_0 =$ The entropy of random sequence $S$
$q = q + q_0$
End
$q = q/K$

**End**
Rule Test scheme uses the state of central CA cell \((N/2)\) to construct random sequences. Because the generated random strings strongly depend on initial configuration, Rule Test scheme uses \(K\) random initial configurations to evaluate statistically reliable value of entropy. The proposed scheme constructs one random string for each initial configuration. Finally, the quality of each rule \(R\) is computed as average value of all calculated entropy values.

If a rule gets a low rank (low entropy) in rule ranking, it cannot be an appropriate rule for CA randomizers. Also high-ranked rules are good candidate to be used in CA randomizers. In experiments, \(K=1024\), \(M=4096\), \(h=4\), and \(l=2\) were used for two CA with 50 and 64 cells. Rules with high entropy in Rule Test scheme are high-ranked and rules with low entropy are low-ranked rules. All 256 symmetric rules, of radius one, were ranked for CAs with 50 and 64 cells. Table 1 shows the first 20 high ranked rules for CAs with 50 and 64 cells.

As it depicted in Table 1, Degree of imbalancity for almost all high ranked rules are zero and these rules are balanced. This result shows that there is a direct relationship between balanced rules and high quality rules for using in CA randomizers. Fig. 2 shows the degree of imbalancity for all ranked rules from high–ranked to low-ranked for a CA with 50 cells. The result shows that the balanced rules are appropriate rules to be used in CA randomizers.

![Fig. 2. Rule with high degree of imbalancity are not appropriate rules for CA randomizers](image)

Table 2. Rule-ranking based on Rule Test scheme for symmetric rules of radius one

<table>
<thead>
<tr>
<th>No.</th>
<th>Rule (50 Cells)</th>
<th>(I_1) (Rule)</th>
<th>Rule (64 Cells)</th>
<th>(I_1) (Rule)</th>
<th>No.</th>
<th>Rule (50 Cells)</th>
<th>(I_1) (Rule)</th>
<th>Rule (64 Cells)</th>
<th>(I_1) (Rule)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>90</td>
<td>0</td>
<td>75</td>
<td>0</td>
<td>11</td>
<td>169</td>
<td>0</td>
<td>225</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>60</td>
<td>0</td>
<td>86</td>
<td>0</td>
<td>12</td>
<td>149</td>
<td>0</td>
<td>106</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>101</td>
<td>0</td>
<td>149</td>
<td>0</td>
<td>13</td>
<td>225</td>
<td>0</td>
<td>22</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>45</td>
<td>0</td>
<td>89</td>
<td>0</td>
<td>14</td>
<td>195</td>
<td>0</td>
<td>151</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>89</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>15</td>
<td>135</td>
<td>0</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>102</td>
<td>0</td>
<td>101</td>
<td>0</td>
<td>16</td>
<td>120</td>
<td>0</td>
<td>166</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>153</td>
<td>0</td>
<td>169</td>
<td>0</td>
<td>17</td>
<td>106</td>
<td>0</td>
<td>154</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>30</td>
<td>0</td>
<td>135</td>
<td>0</td>
<td>18</td>
<td>150</td>
<td>0</td>
<td>170</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>75</td>
<td>0</td>
<td>45</td>
<td>0</td>
<td>19</td>
<td>105</td>
<td>0</td>
<td>85</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>86</td>
<td>0</td>
<td>120</td>
<td>0</td>
<td>20</td>
<td>165</td>
<td>0</td>
<td>210</td>
<td>0</td>
</tr>
</tbody>
</table>
4.2. High Quality Rules and Balanced Rules of Radius Two

In this section the quality of symmetric rules of radius two is studied. Due to the huge size of rule space $2^{2^{12}}$, it is impossible to search all rule space for rules by high radius. For example for rules of radius two the size of rule space is equal to $2^{12}$. In addition, Rule Test Scheme uses $2^{10}$ initial configurations, the CA is evolved $2^{12}$ time steps, and time complexity of calculating entropy is a coefficient of the generated random string $2^{12}$. So because of exponential time complexity $2^{12} * 2^{10} * 2^{12} * 2^{12}$, it is impossible to search all rule space. To overcome this time complexity, Genetic Algorithm is used to find high quality rules of radius two.

In this study, the chromosomes are CA rules of radius two for a one-dimensional uniform Boolean CA, so that each rule is represented by its binary rule string. Also Rule Test scheme is used as fitness function. The process of rule discovery by GA starts with a random initial generation of 100 chromosomes; in addition, the single point crossover, and the mutation rate of 0.05 are used. Also the values of $K$ and $M$ in Rule Test scheme, as the fitness function, are set to 1024 and 4096, respectively.

The goal of rule discovery is to find an appropriate set of rules that can be used in CA randomizers to produce high quality random numbers. Table 3 shows the proposed rules based on the rule discovery technique by GA.

Table 3. The four proposed rules by GA

<table>
<thead>
<tr>
<th>No.</th>
<th>Rule</th>
<th>Rule String</th>
<th>Degree of Imbalancity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>845585581</td>
<td>0011001001100110011011001011011100101101</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1436194405</td>
<td>0011010110011011001101100110110101</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>2791922090</td>
<td>101001100110011001101100110110111010</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>869020620</td>
<td>0011001111001100001100111001100111001</td>
<td>0</td>
</tr>
</tbody>
</table>

As it depicted in Table two, all proposed rules of radius two are balanced rules, and this observation acknowledges that balanced rules are good candidate to be used in CA randomizers.

5. The proposed Random Number Generator

In this section, we propose a random number generator based on non-uniform one-dimensional Boolean CA. The proposed random number generator uses a combination of the proposed rules in section 4 (Table 3) and their complements.

PARALLEL CA RANDOMIZER scheme uses two CAs ($C$ and $D$). One with $N = 2^t$ cells and another CA is a CA with $t = \log_2 N$ cells. Both CAs take their rules form a set of eight rules- four rules from Table 2 and their complements- therefore, three bits are needed to show each cell’s rule. In addition, the CAs are initialized by random configurations. The proposed scheme needs $3*(N + \log_2 n)$ bits to show the rule configuration for both CAs. Moreover, CAs use $N + \log_2 n$ bits for initial configurations. So the Key length is equal to $4*(N + \log_2 n)$.

To generate a random bit, each time both CAs are evolved in parallel. After each evolution the configuration of CA $D$ is used as a binary value which is correspond with a cell $<K>$ of $C$. for example for a CA D with 8 bit and the following configuration (As it depicted in Fig. 3) $K$ is 218=(11011010). Therefore state of cell $<218>$ from CA $C$ is used as a random binary value.

| 1 1 0 1 1 0 1 0 |

Fig. 3. Configuration of CA D with 8 cells determine the cell $<K>$ which is used to produce a random binary value.
To generate a random sequence of length M, both CAs are evolved in parallel for M times. After each evolution, K is calculated using the configuration of D and then the state of cell <K> is used to generate a random binary value. The proposed CA Randomizer (PARALLEL CA RANDOMIZER) is as follows:

PARALLEL CA RANDOMIZER scheme

Input:

C: A one-dimensional non-uniform CA with \( N = 2^t \) cells
C0: A random initial configuration for C
RSC: A random rule configuration for C
D: A one-dimensional non-uniform CA with \( t = \log_2 N \) cells
D0: A random initial configuration for D
RSD: A random rule configuration for D
M: Size of output string

Output:

S: A random string of length M as the output of PARALLEL CA RANDOMIZER scheme

Begin

Set C with initial configuration C0
Set D with initial configuration D0
Assign each cell of C a rule by RSC
Assign each cell of D a rule by RSD
For i:=1 to M

Begin

Evolve C and D as parallel one time
K=the equivalent number with configuration of D
Set \( S(i) \) as the state of cell <K>

End

End
6. Quality of the proposed scheme

The PARALLEL CA RANDOMIZER scheme has a huge key space such that it is robust against direct cryptanalysis which tries to find the key by using all possible keys. In addition, when a random combination of rules with their complements are used for non-uniform CAs, without knowledge of each cell’s rule one cannot make an efficient hypothesis about the next state of the cell (It is obvious that an efficient hypothesis is one with the probability greater than 0.5).

All CA randomizers [7, 8, 9, 10, 11] use constant cell (cells) to generate random string, and as a direct result, this helps a cryptanalyst to gain some information about the state of some cells in some time steps. The proposed scheme selects the cell which should be used for generating the output bit randomly by using the configuration of $D$, so a cryptanalyst cannot gain any information about the state of CA cells and it make the scheme more robust against cryptanalysis.

In addition, the proposed scheme generates high quality random strings. Fig. 4 visually depicts the quality of generated random strings with PARALLEL CA RANDOMIZER scheme. To construct this image, each generated random bit is corresponded to a pixel of a black and white image. The pixel color is set to white, if its corresponding bit is 1; otherwise, it is set to black.

Moreover, The NIST test suite as a battery of statistical tests for random number generators is used to test the quality of the PARALLEL CA RANDOMIZER scheme [16]. The proposed random number generator successfully passed all sixteen tests, including Frequency, Block Frequency, Runs, Longest Run, Binary Matrix Rank, Discrete Fourier Transform, Non-overlapping Template Matching, Overlapping Template Matching, Universal Statistical, Lempel-Ziv Compression, Linear Complexity, Serial, Approximate Entropy, Cumulative Sums, Random Excursion, Random Excursions Variant.

Fig. 4. The generated black and white image by PARALLEL CA RANDOMIZER scheme
7. Conclusion

In this paper, we defined and proposed balanced rules as appropriate rules to be used in CA randomizers. The study of one-dimensional CA rules of radius one and two acknowledged that balanced rules are high quality rules to be used in CA randomizers. In addition, the PCAR scheme which is constructed by balanced rules of radius two and their complements generates high quality random numbers. This acknowledges that balanced rules are good candidate to be used in CA randomizers.

Although we propose balanced rules for CA randomizers, these rules can be used as good candidate for all CA cryptosystems. In addition, during rule discovery process, we can confine the search space to balanced rules where there is more likely to find appropriate rules. Moreover, we propose the study of balanced rules and their properties as future work.

References