Simple method of selecting totalistic rules for pseudorandom number generator based on nonuniform cellular automaton

Mirosław Szaban¹

Accepted: 18 February 2021 / Published online: 22 March 2021
© The Author(s) 2021

Abstract

This paper is devoted to selecting rules for one-dimensional (1D) totalistic cellular automaton (TCA). These rules are used for the generation of pseudorandom sequences, which could be useful in cryptography. The power of pseudorandom number generator (PRNG) based on nonuniform TCA can be improved using not only one rule but a large set of rules. For this purpose, each subset of rules should be analyzed with its assignment to cellular automaton (CA) cells should be analyzed. We examine each of the subsets of totalistic rules, consisting of rules with neighborhood radius equal to 1 and 2. The entropy of bitstreams generated by the nonuniform TCA points out the best set of rules appropriate for the TCA-based generator. The paper also presents the method of simple selection of CA rules based on a cryptographic criterion known as a balance. The proposed method selects a maximal size of the set of available CA rules for a given neighborhood radius and suitable for PRNG. The method guarantees to avoid conflicting assignments of rules resulting in the creation of unwanted stable bit sequences, and provides high-quality pseudorandom sequences. This technique is used to verify the subsets of rules selected experimentally. Verified rules are proposed for 1D TCA-based PRNG as a new subset of best nonuniform TCA rules. New picked, examined, and verified subset of rules could be used in TCA-based PRNG and provide cryptographically strong bit sequences and huge key space.

Keywords Cellular automaton · Pseudorandom number generator · Totalistic rules · Cryptography

✉ Mirosław Szaban
miroslaw.szaban@uph.edu.pl

¹ Institute of Computer Science, Siedlce University of Natural Sciences and Humanities, Siedlce, Poland

1 Introduction

The digital world surrounding us is present in every area of human activity. Every minute digital data are created in a large amount. In these data, the main part is not public but private. So the need for securing the safety and privacy of digital information stored or transmitted over global networks is growing. Cryptographic techniques are among others used to provide information security, being essential component of any secure communication tools. Nowadays, two core cryptography systems are used: secret and public-key systems. An extensive overview of currently known or emerging cryptography techniques used in both types of systems can be found, e.g., in [10]. One promising cryptography technique is the application of cellular automata (CA).

Intended for public-key cryptosystems, CA was proposed by Guan [2], and Kari [6]. Such systems require two types of keys: one key for encryption and the other one for decryption. One is held in private, the other rendered public. The main concern of this paper is cryptosystem with a secret key, also called the symmetric key cryptography systems. The main idea of cryptography using a symmetric key is that both sides of the cryptographic process apply the same key to encrypt and decrypt the message. The key is secret and most secure, because only two persons can use it, while other people can only know the encrypted message, which is too challenging to encrypt without knowing the key. In this study, we continue Vernam's approach to cryptography with the secret key. In these approach P is a plain-text message consisting of m bits where $(p_1 p_2 \dots p_m)$ and $(k_1 k_2 \dots k_m)$ are a bitstreams of a key K . The cipher-text C is created bit-by-bit with applying enciphering operation XOR (exclusive-or) as follows: $C = P XOR K$ (and bit-by-bit: $c_i = p_i XOR k_i$). The original bit P of a message can be recovered by applying the same operation XOR on C (bit of a cipher-text) using the same bitstream key K : $P = C XOR K$ (and bit-by-bit: $p_i = c_i XOR k_i$). Vernam Cipher's encoding algorithm (see, [8, 10]) is known as perfectly safe if the keystream is genuinely unpredictable and used only once. The encryption process is based, in particular, on the generation of high-quality pseudorandom bit sequences, and CA can be effectively used for this purpose. In this paper, we answer the following questions: how to select an appropriate set of CA rules providing near pure randomness of key bitstreams; and how to obtain such a key with a length large enough to encrypt real-world amounts of data.

CA for symmetric cryptography was first studied by Wolfram [20], who proposed 1D CA-based PRNG with rule 30 and later by Habutsu et al. [3], Hortensius et al. [4], and Nandi et al. [9], who proposed rules 90 and 150. Next, this subject was studied by Tomassini et al. [17, 18], where the set of rules was enlarged to rules: 90, 105, 150, 165. Afterward, in the paper [11] authors presented a new larger set of rules {86, 90, 101, 105, 150, 153, 165, 1436194405}, discovered with use of evolutionary technique called cellular programming (CP) [12]. This set of rules consists of rules with the neighborhood of radius $r = 1$ and $r = 2$ (last rule in the set). Correspondingly, this set gives similar results in the sense of passed tests like entropy test and FIPS 140-2 (standard tests for basic analysis of the PRNG's

quality), but offered larger keys (different bit sequences) than previous proposals. Lately, in the paper [14] were presented techniques of appropriate selection of rules for one-dimensional (1D) cellular automata (CA), which is used for generation PRNG. These techniques are based on a cryptographic criterion known as a balance. The paper [14] presents a new set of CA rules with neighborhood radius $r = 2$. For a selected set of CA rules, the statistical testing approach was applied. As a result, the whole general set and each subset of selected rules can be used in CA-based PRNG and provide cryptographically strong bit sequences. Similar statistical testing approach for analyzing CA usefulness in cryptography and as PRNG was used in papers: [1, 5, 7], etc.

Cryptographic techniques and ciphers require secure keys, being high pseudorandom, or almost random sequences. Many generators of such keys are known, but they did not supply present-day demands on quantity of applied keys. CA is known to be a powerful tool creating such large amounts of number sequences. In the literature, for this purpose are used an elementary CA [21]; also totalistic CA [19] seems to be promising and able to enlarge the set of existing tools for generating PRNS.

The paper is organized as follows. The next section outlines the concept of 1D Totalistic CA and its relation with CA-based symmetric cryptography. In Sect. 3, a construction of TCA-based PRNG is described. Section 4 describes the sets of quality tests for examining obtained sequences of bits. Section 5 presents the results of the research. Proper totalistic rules, testing these rules, and analysis of their cryptographic quality are described here. The last section concludes the paper.

2 Totalistic cellular automata and symmetric cryptography

A cellular automaton (CA) is a discrete, dynamical system consisted of identical cells arranged in a regular grid, in one or more dimensions [21]. In this paper, one-dimensional CA is considered. 1D CA is, in the simplest case, a collection of two-state elementary cells arranged in a lattice of the length N and locally interacting in a discrete time t . For each cell, i , called a central cell, a neighborhood of a radius r is defined. The neighborhood consists of $n = 2r + 1$ cells, including the cell i . A cyclic boundary condition is applied to a finite size CA, which results in a circle grid. According to a local rule defined on a neighborhood, initial states of all cells (an initial configuration of a CA) and states of cells are updated synchronously at discrete time steps. In this paper, finite CA with the totalistic type of CA rule (TCA) [19] is considered. It is assumed that a state q_i^{t+1} of a cell i at the time $t + 1$ depends only on states of its neighborhood at the time t , i.e.,

$$q_i^{t+1} = TF_i\left(\sum_{j=-r}^r q_{i+j}^t\right) = TF_i(q_{i-r}^t + \dots + q_{i-1}^t + q_i^t + q_{i+1}^t + \dots + q_{i+r}^t), \quad (1)$$

where TF_i is a totalistic transition function, also called a totalistic rule, defining how to update the cell i . The length L of a totalistic rule and the number of neighboring states for a binary CA is $L = 2(r + 1)$, where n is a number of given neighborhood cells. The number of such rules can be expressed as 2^L . For CA with, e.g., $r = 1$

($r = 3$) the length of the rule is equal to $L = 4$ ($L = 8$), while number of such rules is 2^4 (2^8) and grows very fast with L .

TCA can change its state in time using a single rule assigned to all CA cells, and it is called a uniform TCA. If two or more different rules are given to update cells, TCA is called nonuniform TCA. Wolfram system [20] was uniform; the other mentioned above systems were nonuniform. This paper is dedicated to nonuniform TCA and shows the analysis of such kind of TCA.

3 A concept of 1D TCA-based PRNG

Similarly, like in the case of elementary CA [4, 9, 11, 14, 17, 18, 20], let us consider a PRNG based on 1D TCA with a lattice of the length N consisting of cells locally interacting in a discrete-time t . A rule (rules) of the TCA controlling cells are described as in Eq. 1. A corresponding seed of the generator will consist of few elements. The first element is an initial configuration of CA, the second is a set of TCA rules, the third is an index of the cell (i), which generate bit sequence used for encryption, and the last is a number of time steps (T), which correspond to the length of a bit sequence. During CA work, fixed cell i changes its state. The next state of the cell i create the bit sequence. Such proposed construction is TCA-based PRNG. The TCA-based PRNG should generate cryptographically strong bit sequences independently to an initial CA configuration, i.e., a selected cell i , and time steps T . The set of totalistic rules for managing the CA should be carefully chosen, and a particular assignment of rules to cells should not be conflicting. To examine these requirements, key streams generated by selected rules will be put under dedicated for these purpose cryptographic tests, like the Entropy test (presented in this paper). They could also be examined by the NIST SP 800-22 tests and a Diehard set of Marsaglia tests (which will be realized in future works).

4 Quality tests for number generators

4.1 The entropy test

The entropy E_h task is to specify the statistical quality of each pseudorandom number sequences (PNS). We used Shannon's equation of even distribution as an entropy function. To calculate the entropy's value, each PNS is divided into subsequences of size h ($h = 4$). Let k be the number of values, which can construct a single element of a sequence (for binary values $k = 2$) and k^h a number of possible states of each sequence of length h (if $h = 4$ than $k^h = 16$). E_h can be calculated in the following way:

$$E_h = - \sum_{j=1}^{k^h} p_{h_j} \log_2 p_{h_j}, \quad (2)$$

where p_{h_j} is a probability of occurrence of a sequence h_j in a PNS.

The entropy achieves its maximum $E_h = h$, when the probabilities of the h_j (possible sequences of the length h) are equal to $\frac{1}{k^h}$.

5 Selection of totalistic rules for application in nonuniform TCA-based PRNG by passing tests

During the experiments reported in [13], were obtained sets of uniform TCA rules characterized by the Entropy test's best values, the NIST SP 800-22rev1a tests. These sets of rules were obtained automatically based on test results, without analyzing TCA's structure with applied rules. Selected rules could be used to BitStream Generator as a seed of TCA-based PRNG, and the session keys number for every single rule is equal $1^N * 2^N$, where the N is the number of TCA cells. So, to enlarged the set of session keys for TCA-based PRNG should be used nonuniform TCA, whereas seeds are applied not one single rule but a set of rules (more than one rule), which managed the changes of TCA cells during the time steps. In such a case, the session keys number is equal $k^N * 2^N$, where k is a number of rules in nonuniform TCA.

5.1 Experimental results

The starting point is an analysis of a nonuniform CA with all totalistic rules. The research cover TCA rules with neighborhood radius $r = 1$, and an examination of all subsets of these 16 rules (i.e., $\{r0, ..., r15\}$). In the next stage also all subsets of TCA rules with $r = 2$ are examined. This whole set consists of 64 TCA rules. However, the number of subsets to analyze is the sum of each of combinations $C(n, k) = \frac{n!}{k!(n-k)!}$, where n is the number of rules in the whole set, and k is a number of rules in the subset. In the nonuniform, TCA the cooperation of rules is essential. So, if one rule cooperates with each of the other rules in a two-element subset, then probably they could cooperate with rules in a subset larger than two numbers of rules. Cooperation of rules means that such rules applied in nonuniform TCA will generate high-quality bitstreams. Therefore, in the next experiments a two-element subset of nonuniform TCA rules will be analyzed.

In all performed experiments, CA size was equal to 100, and CA was working in 4096-time steps (a value suitable for Entropy test), which examine the distribution of subsequences consisted of 4 elements in the whole bit sequence. From one CA run one-bit sequence obtained from states of randomly selected CA cell was selected. Each rule test was repeated 10000 times with random initial configuration of CA state, and the average Entropy of each rule was calculated.

During the experiments each of 120 subsets containing 2 rules with $r = 1$ was analyzed, and only 5 subsets of rules were selected, because average entropies of other subsets are lower than 3,8. Table 1 presents values of minimal, average, and maximal entropy for each best two-element subsets and one the worst subset of rules with $r = 1$ for TCA. We can see that only five from all subsets of TCA rules have a good quality, and obtained entropy values are near to maximal equal to 4. The best

Table 1 Values of entropy for selected pairs of nonuniform TCA rules with $r = 1$, obtained from 10000 tests with the random initial configuration of CA state

Rules	Binary rules	Entropy min.	Entropy ave.	Entropy max.
t5, t10	0101, 1010	3,996492559	3,999860445	3,999954371
t5, t2	0101, 0010	3,366156548	3,953761099	3,965234439
t5, t11	0101, 1011	3,901125467	3,945968610	3,983327578
t10, t2	1010, 0010	3,861017601	3,933497519	3,960603068
t10, t11	1010, 1011	3,684155449	3,909619841	3,933592499
t11, t2	1011, 0010	0,051211863	3,683436868	3,724036994
...	...	...	...	...
t4, t8	0100, 1000	0,002896606	0,142102885	0,278299729

subsets of rules are: $\{t5, t10\}$, $\{t5, t2\}$, $\{t5, t11\}$, $\{t10, t2\}$, $\{t10, t11\}$, among them the highest average entropy equal to 3,999860445 has the subset $\{t5, t10\}$. Other rules are characterized by not enough high value of entropy. As we can see in Table 1, we have four rules ($\{t2, t5, t10, t11\}$) which joined in pairs are characterized by a high value of entropy, except the couple $\{t11, t2\}$, for which the minimal entropy is almost equal to 0. It means that stable or cyclic bit sequence was generated. We can conclude that rules $t2$ and $t11$ do not cooperate with each other like other rules mentioned above.

In Table 2 we can see presented value of entropy for nonuniform TCA for subsets of rules selected from set of cooperating rules. Table 2 holds results for subsets composed of four $\{t5, t10, t2, t11\}$, or three $\{t5, t10, t2\}$ and $\{t5, t10, t11\}$ cooperating with each other TCA rules. Despite the higher number of rules in these subsets, the values of entropy are higher for subset composed of two rules $\{t5, t10\}$ (see, Table 1).

The next stage of experiments was performing the entropy test on two-element subsets from the whole set of TCA rules with $r = 2$. During the experiments, each of 2016 subsets containing 2 rules with $r = 2$ was analyzed. The 125 subsets of rules, with average entropy not lower than 3,9 were selected. In this subset of 125 couples of rules, the best average value of entropy and best minimal entropy are characterized by the couples presented in Table 3. As we can observe, the rules composed of pairs crate set of the strongest rules $\{t10, t11, t20, t21, t42, t43, t52$ and $t53\}$, except the pairs $\{(t20, t52), (t52, t53)\}$ with very low entropy values. Each couple from this set is presented in Table 3.

Table 2 Values of entropy calculated for subsets of rules, selected from the set of cooperated nonuniform TCA rules with $r = 1$, obtained from 10000 tests with the random initial configuration of CA state

Rules	Binary rules	Entropy min.	Entropy ave.	Entropy max.
t5, t10,	0101, 1010	3,725403281	3,947546359	3,980636366
t2, t11	0010, 1011			
t5, t10,	0101, 1010	3,870986994	3,967697942	3,986950426
t2	0010			
t5, t10,	0101, 1010	3,750574365	3,988016194	3,998255285
t11	1011			

Table 3 Values of entropy for selected pairs of nonuniform TCA rules with $r = 2$, obtained from 10000 tests with the random initial configuration of CA state

Rules	Binary rules	Entropy min.	Entropy ave.	Entropy max.
t21, t42	010101, 101010	3,997536552	3,999570372	3,999996948
t10, t53	001010, 110101	3,966649561	3,999399189	3,999792413
t20, t43	010100, 101011	3,962663789	3,998624173	3,999569667
t10, t21	001010, 010101	3,980969539	3,996978264	3,997720656
t11, t52	001010, 110100	3,948529775	3,996603310	3,998405877
t21, t43	010101, 101011	3,961628073	3,996283475	3,998897691
t42, t43	101010, 101011	3,977019780	3,995220043	3,996396892
t10, t42	001010, 101010	3,971103642	3,994672342	3,997471220
t21, t53	010101, 110101	3,922131630	3,989958200	3,994333684
t11, t20	001011, 010100	3,962841223	3,989924098	3,993098170
t10, t43	001010, 101011	3,932817212	3,989819484	3,991688961
t21, t52	010101, 110100	3,977018763	3,988493053	3,992059429
t20, t42	010100, 101010	3,939659537	3,987008202	3,992980897
t10, t52	001010, 110100	3,833273892	3,979045690	3,982094763
t11, t53	001011, 110101	3,922849989	3,978805785	3,992356714
t43, t53	101011, 110101	3,914536918	3,978145219	3,984487805
t43, t52	101011, 110100	3,964579863	3,978006487	3,992314314
t42, t53	101010, 110101	3,918778773	3,975426609	3,983267180
t20, t53	010100, 110101	3,921563093	3,973513439	3,976810524
t11, t21	001011, 010101	3,946582377	3,972045196	3,995190297
t11, t42	001011, 101010	3,958004806	3,969870354	3,992219684
t20, t21	010100, 010101	3,926115240	3,969799518	3,992821014
t10, t20	001010, 010100	3,929293073	3,968819021	3,978152365
t42, t52	101010, 110100	3,864345235	3,967529080	3,974254542
t10, t11	001010, 001011	3,845671782	3,957197012	3,965069576
t11, t43	001011, 101011	3,924609436	3,946618529	3,969862542
t20, t52	010100, 110100	0,083436848	0,097931947	0,118192591
t52, t53	110100, 110101	0,083855096	0,089813895	0,118929786

Moreover, we can see that the best values of entropy (minimal, average, and maximal) present the pair $\{t21, t42\}$. Also, in the subset of 125 pairs of rules occasionally rules: $\{t5, t12, t13, t18, t19, t22, t23, t25, t26, t28, t30, t33, t35, t36, t37, t38, t44, t50$ and $t51\}$ have appeared.

Figure 1 presents the graphical representation of TCA work. We can see states of TCA during the time steps. We can observe each of the single cells and their states during the time steps. In the cases (a) and (b) on the diagrams we can't see stable or cyclic bit sequences. This fact confirms corresponding entropy values near to maximal (equal to 4), see Table 3. In the opposite cases (c) and (d), diagrams are showing us vanishing randomness and stable and also cyclic bit sequences. As was mentioned above, TCA rules from these pairs do not cooperate

Fig. 1 Space-times diagrams of nonuniform TCA with 100 cells and 100 time steps managed by set of rules: (a) $\{(r21, r42)\}$, (b) $\{(r10, r53)\}$, (c) $\{(r20, r52)\}$ and (d) $\{(r52, r53)\}$. White squares are representing 0's and black 1's in sequences

with each other, so low values characterize entropies of such generated sequences (see, last rows of Table 3).

Table 4 presents values of entropy for maximal set of rules for nonuniform TCA, selected from set of cooperating rules. We can see high entropy values, average, and minimal and maximal obtained during the executed experimental tests. Figure 2 presents the space-time diagram of nonuniform TCA with rules applied from this set. As we can see, there are no stable or periodic sequences of bits, 0's are represented by white, and 1's by black squares.

To be sure that the TCA rules did not generate stable or cyclic sequences of bits, let us look at them from the perspective of one of the most crucial

Table 4 Values of entropy corresponding to subsets of rules selected from the set of cooperating nonuniform TCA rules with $r = 2$, obtained from 10000 tests with the random initial configuration of CA state

Rules	Entropy min.	Entropy ave.	Entropy max.
t10, t11, t20, t21, t42, t43, t52, t53	3,99323316236544	3,99980168294887	3,9998985308231

Fig. 2 Space-time diagrams created for nonuniform TCA with a number of cells and time steps equal to 200. TCA was managed by a set of rules $\{r10, r11, r20, r21, r42, r43, r52 \text{ and } r53\}$. White squares are representing 0's and black 1's in sequences

cryptographic criteria—a balance of the TCA. This approach analyzes the behavior of rules in the sense of balance in TCA rules structure.

5.2 Balance of 1D CA-based PRNG

Balance (regularity) is one of the important criteria which should be fulfilled by a Boolean function used in ciphering (see, [22]). This criterion says that each output bit (0 or 1) of a Boolean function should appear an equal number of times for all possible input values. For the CA-based generator, it means that independently on the initial configuration (except consisted of only 0s or only 1s), CA should achieve the state where the number of 0s and 1s in a CA state is equal to $\frac{N}{2}$, and should maintain this state in the next time steps. The balance of the CA could be satisfied by using the balanced rules of CA. CA rule as a Boolean function $f : Z_2^n \rightarrow Z_2$ maps n binary inputs (neighborhood state of CA) to a single binary output. So, the balance of a Boolean function is measured using its Hamming Weight and is defined as:

$$HW(f) = \sum_{x \in B^n} f(x). \quad (3)$$

A Boolean function is balanced when its Hamming Weight is equal to 2^{n-1} . In means that balanced rule has in binary form an equal number of 0s and 1s. As we can see the rules described in [9, 11, 15–17, 20] are balanced for both $r = 1$ and $r = 2$. For a single rule as rule of CA-based PRNG, a balance is necessary criterion except formal tests, but for a set of rules it is insufficient.

5.3 Selection of totalistic rules for application in TCA-based PRNG by analysis of structure and balance

Rules for TCA-based PRNG selected during the tests, despite passing the appropriate tests, sometimes generated bit sequences with low randomness. Since the tests are examining only bit sequences chosen randomly from thousands generated by the generator, sometimes, incidental cyclic bit sequence was not considered or averaged results from many tests do not reflect its proper characteristic. So occasionally, even a good rule applied in TCA can create cyclic or partially stable bit sequences. In the paper [14] we presented a method of selection of Elementary CA rules based on a cryptographic criterion known as a balance. The proposed method selects a maximal size set of available CA rules for a given neighborhood radius and suitable for PRNG. This technique guarantees to avoid wrong assignments of rules resulting in the creation of unwanted stable bit sequences and provides cryptographically strong bit sequences and huge keyspace. This method introduces a relationship for good bit mixing. A rule for CA-based PRNG should perform the same number of the subsequent cell state transitions: $0s \rightarrow 0s$, $0s \rightarrow 1s$, $1s \rightarrow 0s$ and $1s \rightarrow 1s$. Therefore, for Elementary CA, the outputs 0s and 1s should be selected in a special configuration, resulting in CA rule. This method cannot be identically transferred to creating the TCA rules, because in TCA, does not exist a direct relationship between neighborhood structure and effect of work. Elementary CA rule from a strict neighborhood structure creates an adequate state of CA cell. In TCA the neighborhood structure is not essential; instead, the sum of elements in the

neighborhood plays a vital role. From here, for different neighborhood structures with the same sum of 0s and 1s, the TCA rule gives the same effect and generates identical states of TCA cells.

Let's analyze a TCA rule with $r = 1$ as a Boolean function. In Table 5 the letters $a, \dots, d \in \{0, 1\}$ correspond to outputs of a Boolean function. When $a + \dots + d = 2$, then the Elementary CA rule is balanced, but for TCA, the balance means $a * 1 + b * 3 + b * 3 + d * 1 = (1 + 3 + 3 + 1)/2 = 4$, because the totalistic rule creates the same CA state for few different neighborhood structures. For a good mixing of bits, a rule for TCA-based PRNG should perform the same number of the following cell state transitions: $0s \rightarrow 0s$, $0s \rightarrow 1s$, $1s \rightarrow 0s$ and $1s \rightarrow 1s$. The totalistic rule for few neighborhood structures creates the same CA state; therefore, for these output 0s and 1s should be selected in $a, \dots, d$ in a special configuration. A bad schedule might cause that rule will produce stable sequences during an interaction with other rules (see, Fig. 1 (c), (d)). Elementary CA for a single neighborhood state generates one output (see, the paper [14]). However, in the TCA case, one single output is created by the neighboring states.

TCA rules to be balanced should have two 0s and two 1s in binary form. Moreover, the same number of 1s generating 1s, 1s generating 0s, 0s generating 1s, and 0s generating 0s, which determine the situation where the same number of neighborhood states generate 1s and 0s. So, to satisfy the above requirements, a rule should fulfill the following logical sentence, based on designation Table 5:

$$(a \neq b) \wedge (c \neq d) \wedge (a \neq d). \quad (4)$$

Let $\{0, 1, X\}$ be the binary alphabet, where X is the mask, and means X masking each value in the alphabet. From above, when we have a situation, where the part of TCA state is, e.g., $\{..0111.. \}$ and neighborhood configuration is $\{..X11X.. \}$ (from $\{X11\}$ and $\{11X\}$), then two middle cells of TCA will produce sequences of 1s during the TCA work, etc. So, in formula 4: $(a \neq b)$ protects rule against input configurations of neighborhood: $\{11X\}$ and $\{X11\}$ translates in TCA $1s \rightarrow 1s$, and leads to generation of unwanted stable sequence of 1s; $(c \neq d)$ protects against input configurations $\{00X\}$ and $\{X00\}$ translates $0s \rightarrow 0s$, leading to generation of unwanted stable sequence of 0s. Also in binary alphabet, from formula 4 we receive relationship $(b \neq c)$, which protects rule against input configurations of neighborhood: $\{10X\}$ and $\{X01\}$ translating $0s \rightarrow 0s$, leading to generation of unwanted stable sequence

Table 5 TCA rule with $r = 1$ as a Boolean function

Input	111	110	100	000
		101	011	
		011	001	
Sum of the elements in neighborhood structure	3	2	1	0
Number of neighborhood structures corresponding to sum of elements	1	3	3	1
Output	a	b	c	d

of 0s, and configurations $\{01X\}$ and $\{X10\}$ translating $1s \rightarrow 1s$, leading to generation of unwanted stable sequence of 1s. Formula 4 could be simplified to the following logical sentence:

$$(a = c) \wedge (b = d) \wedge (a \neq b). \quad (5)$$

Let us solve the formula 5.

Let $a = 0$ then $c = 0$ and $b = d = 1$, and this way we obtain the binary TCA rule $(0101)_2$, i.e., a decimal rule $t5_{10}$.

Let $a = 1$ then $c = 1$ and $b = d = 0$, and this way we obtain the binary TCA rule $(1010)_2$, i.e., a decimal rule $t10_{10}$.

As we can see, analyzed rules have a neighborhood radius equal to 1. To summarize, only two of these rules are free from a wrong configuration in nonuniform TCA-based PRNG, leading to generating stable or cyclic sequences of bits. We can obtain the same result by analyzing, e.g., the entropy of generated bit sequences, see Table 1 (first row of data), where we can see high values of entropies for this set of rules. Entropies for these rules far exceed values for other presented in the same table. Nevertheless, experimental surveys need much more time than the method of analysis with applying balance criterion, which gives a possibility to easily select rules.

Analogously, applying the above method, we can create a suitable logical formula for TCA rules with neighborhood $r = 2$. Table 6 presents a TCA rule with $r = 2$ as Boolean function.

On the base of Eqs. 4 and 5 for CA rules with $r = 1$, we can create the suitable logical formula for CA rules with $r = 2$. The dependencies between outputs of the TCA rule with $r = 2$, which during cooperation with other rules do not create stable sequences present following logical sentence:

$$(a \neq f) \wedge (a \neq b) \wedge (b \neq e) \wedge (b \neq c) \wedge (c \neq d). \quad (6)$$

The letters $a, b, c, d, e, f \in \{0, 1\}$ correspond to outputs of a Boolean function. When $a * 1 + b * 5 + c * 10 + d * 10 + e * 5 + f * 1 = 32/2 = 16$ then the rule is balanced. The dependencies between outputs of the TCA rule with $r = 2$ such that its interaction with other rules do not generate stable sequences was achieved in two selected TCA rules: $t21$, $t42$. But, when we give a possibility not to satisfy the balance a little bit, i.e., to be not 16 but for example $17 = 16 + 1$ or $15 = 16 - 1$, with a specific degree of freedom equal to 1. Then, constructed this way rules will be not

Table 6 TCA rule with $r = 2$ as a Boolean function

Input	1111	1110	1100	1100	1000	0000
		...	...	...	...	
		0111	0011	0001	0001	
Sum of the elements in neighborhood structure	5	4	3	2	1	0
No. of neighborhood structures corresponding to sum of element	1	5	10	10	5	1
Output	a	b	c	d	e	f

balanced but quasi-balanced, resulting in making the changes for Boolean function outputs in $a, f \in \{0, 1\}$. Therefore, Eq. 6 is changing to the following form:

$$(b \neq e) \wedge (b \neq c) \wedge (c \neq d), \quad (7)$$

by reducing the relationship $(a \neq f) \wedge (a \neq b)$.

The solutions of these logic sentences leads to set of TCA rules: $\{t10, t11, t20, t21, t42, t43, t52 \text{ and } t53\}$ corresponding to set presented in Table 4 discovered during the time-consuming experiments. This set of rules was obtained automatically based on test results (see the previous section) and confirmed by analyzing TCA's structure with applied rules. Selected rules could be applied to BitStream Generator as a seed of TCA-based PRNG. The session keys number for a selected set of TCA rules is equal $8^N * 2^N$, where the N is the number of TCA cells, while the number of session keys for every single rule, is only equal to $1^N * 2^N$. We could conclude that nonuniform TCA with a large session key space could increase the protected information's safety.

In this same easy way, everyone can select the best sets of TCA rules free from the creation of cyclic or stable sequences for various TCA neighborhood radius.

6 Conclusion

This paper includes the analysis of nonuniform totalistic CA as a base of PRNG. During the experiments were selected subsets of TCA rules for neighborhood radius equal to 1 and 2. These sets are characterized by higher than for others subsets pseudorandomness measured by Entropy values.

Compositions of TCA rules in sets selected by the experimental method were confirmed by the theoretical approach based on a cryptographic criterion known as a balance. The structures of a neighborhood in TCA and balance (regularity) as relationships between inputs and outputs of TCA cells and rules were analyzed. Performed analyses showed which rules of nonuniform TCA are suitable for generating bit sequences with the highest pseudorandomness.

It was shown that the set of discovered TCA rules with $r = 2$ could be used in TCA-based PRNG as a seed. The selected set of rules provide better quality than other analyzed sets, supporting pseudorandomness of bit sequences and huge session key space.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Formenti E, Imai K, Martin B, Yunès J-B (2014) Advances on random sequence generation by uniform cellular automata. In: *Computing with New Resources - Essays Dedicated to J. Gruska on the Occasion of His 80th Birthday*, pp 56–70
2. Guan P (1987) Cellular automaton public-key cryptosystem. *Complex Syst* 1:51–56
3. Habutsu T, Nishio Y, Sasae I, Mori S (1991) A secret key cryptosystem by iterating a chaotic map. In: *Proceedings of Eurocrypt'91*, LNCS 547, pp 127–140
4. Hortensius RD, McLeod RD, Card HC (1989) Parallel random number generation for VLSI systems using cellular automata. *IEEE Trans Comput* 38:1466–1473
5. Hosseini SM, Karimi H, Jahan MV (2014) Generating pseudorandom numbers by combining two systems with complex behaviors. *J Inf Secur Appl* 19(2):149–162
6. Kari J (1992) Cryptosystems based on reversible cellular automata
7. Leporati A, Mariot L (2014) Cryptographic properties of bipermutative cellular automata rules. *J Cell Autom* 9(5–6):437–475
8. Menezes A, van Oorschot P, Vanstone S (1996) *Handbook of applied cryptography*. CRC Press, Boca Raton
9. Nandi S, Kar BK, Chaudhuri PP (1994) Theory and applications of cellular automata in cryptography. *IEEE Trans Comput* 43:1346–1357
10. Schneier B (1996) *Applied cryptography*. Wiley, New York
11. Seredynski F, Bouvry P, Zomaya A (2004) Cellular automata computation and secret key cryptography. *Parallel Comput* 30:753–766
12. Sipper M, Tomassini M (1996) Generating parallel random number generators by cellular programming. *Int J Modern Phys C* 7(2):181–190
13. Szaban M (2019) Pseudorandom number generator based on totalistic cellular automaton. In: Victor M (ed) *Parallel Computing Technologies*, LNCS 11657, pp 360–370. <https://doi.org/10.1007/978-3-030-25636-4>
14. Szaban M, Seredynski F (2018) Designing conflict free cellular automata-based PRNG. *J Cell Autom* 13(3):229–246
15. Szaban M, Seredynski F, Bouvry P (2006) Collective behaviour of rules for cellular automata-based stream ciphers. In: *IEEE Congress on Evolutionary Computation (CEC 2006)*, July 16–21, 2006. Vancouver, Canada, IEEE, pp 486–490
16. Szaban M, Seredynski F, Bouvry P (2006) Evolving collective behavior of cellular automata for cryptography. In: *The 13 IEEE mediterranean electrotechnical conference (MELECON 2006)*, Benalmadena (Malaga), Spain, May 16–19, pp 799–802
17. Tomassini M, Perrenoud M (2000) Stream ciphers with one- and two-dimensional cellular automata. In: Schoenauer M et al (eds) *Parallel Problem Solving from Nature - PPSN VI*. LNCS 1917. Springer, Berlin, pp 722–731
18. Tomassini M, Sipper M (2000) On the generation of high-quality random numbers by two-dimensional cellular automata. *IEEE Trans Comput* 49(10):1140–1151
19. Wolfram S (1983) Statistical mechanics of cellular automata. *Rev Mod Phys* 55:601–644
20. Wolfram S (1986) Cryptography with cellular automata. In: *Crypto '85 Proceedings*, LNCS 218. Springer, Berlin, pp 429–432
21. Wolfram S (2002) *A new kind of science*. Wolfram Media
22. Youssef A, Tavares S (1995) Resistance of balanced S-boxes to linear and differential cryptanalysis. *Inf Process Lett* 56:249–252

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.