A Comparative Study between the Dynamic Behaviours of Standard Cellular Automata and Network Cellular Automata Applied to Cryptography

Heverton B. Macêdo,[1,*] Gina M. B. Oliveira,[2,†] Carlos H. C. Ribeiro[3,‡]
[1] *Goiás Federal Institute, IFG, Jataí, Brazil*
[2] *Federal University of Uberlândia, UFU, Uberlândia, Brazil*
[3] *Technological Institute of Aeronautics, ITA, São José dos Campos, Brazil*

The dynamical behavior of cellular automata (CA) transition rules are objects of study for different knowledge fields. This paper is about the development of cryptographic methods using CA-like transition rules. We investigate the dynamic behavior of rules that are not able to propagate a perturbation inserted in the initial lattice considering variations of the regular connection structure of CAs that are akin to the small-word network construction process. Extensive experimental results indicate that such modifications in the CA connection structure will produce large changes in the dynamic behavior of the evaluated rules, suggesting that it is possible to increase considerably the space of possible cryptographic keys to methods based on CA rules, provided a different topological construct for the CA lattice is considered. © 2015 Wiley Periodicals, Inc.

1. INTRODUCTION

Cellular automata (CAs)[1] can produce dynamic behaviors related to the temporal evolution of their lattice configurations that are suitable for cryptographic applications,[2] and several studies have been developed along this research line.[3–7] In a recent work,[8] we investigated the dynamic behavior of CAs with changes in the connection structure between neighboring elements, the so-called cellular automata networks (CANs). Specifically, we proposed a particular type of small-world CAN—namely, the directed small world cellular automata (DSW-CA)[8]—capable of evolving rules of CA, but propagating initial perturbations in a considerably shorter

*Author to whom all correspondence should be addressed; e-mail: hevertonbarros@gmail.com.
†e-mail: gina@facom.ufu.br
‡e-mail: carlos@ita.br

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 00, 1–19 (2015)

View this article online at wileyonlinelibrary.com. • DOI 10.1002/int.21751

time than traditional CAs, thus allowing the construction of more efficient cryptographic methods as far as computation time for encryption is concerned. However, the experiments carried out in Ref. 8 considered only a subset of the elementary rules classified as chaotic. Standard CAs have been investigated in the past,[2] and an important conclusion is that the dynamical behavior observed during their temporal evolution is related to the applied transition rule. Indeed, considering the simplest model of a standard CA known as elementary (binary and with radius 1), 256 possible rules can be classified according to their dynamical behavior. The same rules, when applied to DSW-CA networks, may change their dynamical behavior when compared with the standard CA classification due to the connection structure.

Here, we investigate DSW-CA networks under some nonchaotic standard rules. These rules present a special characteristic due to the fact that they are sensitive to a specific cell of the neighborhood (in fact, the left-extremity cell or the right one) and have been investigated as secrete keys for CA-based methods.[3–5] However, using the regular structure of standard CA, only chaotic rules with sensitivity have been applied, and some nonchaotic rules must be discarded from the space of potential secret keys since they are not able to produce the appropriate entropy to the cypher process. Our main objective is to verify if sensitive rules previously classified using the standard CA structure as of periodic or fixed-point behavior are able to produce a propagation of the initial perturbation when applied to DSW-CA networks. The main reason for conducting the present study is to further enhance the space of possible keys for cryptography applications. To measure the propagation of an initial perturbation, we used a measure of the entropy over the states of the CAN along time. Experimental results using a small set of radius 1 nonchaotic sensitive rules indicate that some of them can achieve entropy values equivalent to the rules classified as chaotic, when applied to DSW-CA networks. Additional experiments using a major sample of radius 2 sensitive CA rules confirmed that they changed their nonpropagating behavior to an appropriate level of entropy when a DSW-CA structure is used.

The rest of this paper is structured as follows. Section 2 describes basic concepts of CA and its generalization using graphs. Small-world networks and a building process to produce these kind of networks to our purpose are discussed in Section 3. Section 4 explains basics concepts of elementary CA transition rules. Experiments, analysis of the results, and discussions are given in Section 5, and Section 6 presents the concluding remarks.

2. CELLULAR AUTOMATA

The simplest and most studied CA model is certainly the so-called *elementary* CA. It evolves synchronously in discrete time from a transition rule and a single dimension cell grid (lattice) such that each cell has one of two possible states. The configuration of the CA at a given point in time is given by the configuration of the lattice with respect to the cell state values, and a transition rule is a deterministic mapping to the cell binary state at time $t + 1$ from the states of the immediate neighborhood (left and right cells plus the state of the cell itself) at time t. The

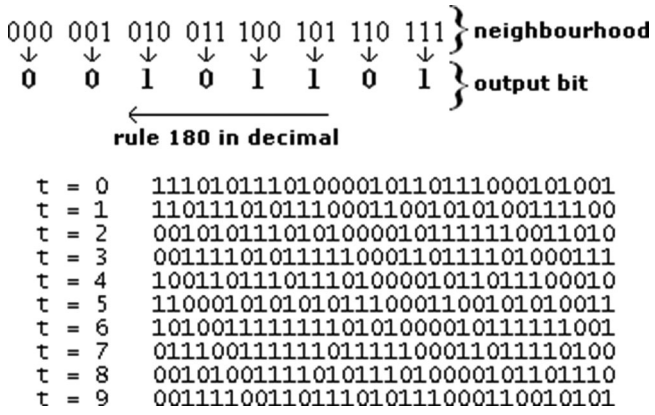

Figure 1. Cellular automaton evolution for nine steps using transition rule ⟨00101101⟩ (rule 180).

transition rule is thus responsible for updating the central cell of the neighborhood in the next time step. The number of configurations of a neighborhood is equal to 8 (2^3), totaling 256 (2^8) possible transition rules. After updating all cells of the lattice, the time step is incremented.

Elementary CAs are homogeneous, meaning that all cells of the lattice evolve using the same rule. Generally, this type of CA has a periodic boundary condition, namely, the lattice structure is circular, although it is possible to define other boundary conditions, such as keeping the extreme cells of the lattice with fixed state values. In this work, we consider the periodic boundary condition.

A cellular automaton A is formally defined[10] by the 4-tuple $A = (\Sigma, U, d, f)$, where Σ corresponds to a finite set of possible states, U is the cell neighborhood, $d \in Z^+$ is the dimension of A, and f is the local transition function responsible for the transition of states in the evolution of the CA, also called the transition rule. For elementary CAs, the set of possible states is $\Sigma = \{0, 1\}$. The neighborhood of a cell with state c^i is $U^i = (c^{i-1}, c^i, c^{i+1})$, where i corresponds to the cell position in the lattice. The transition function f maps the state of a cell c^i at i to another state in Σ ($f : \Sigma^n \rightarrow \Sigma$), according to the neighborhood U^i. Considering a cellular space of size N, the CA configuration at a given instant of time t is defined by $C_t = (c_t^0, c_t^1, \ldots, c_t^{N-1})$, where $c_t^i \in \Sigma$ corresponds to the state of cell i at time t. The dynamic evolution of the CA is performed by the global operator Φ, responsible for mapping a configuration C_t to the next time step ($\Phi : C_t \rightarrow C_{t+1}$), through the simultaneous application of the local transition function f by every cell in the lattice, that is, $C_{t+1}^i = \Phi(U_t^i)$.

Figure 1 shows an example of lattice evolution for an elementary rule. It is possible to see the mapping for all eight possible neighborhoods and their respective output bits (the next state of the central cell). Usually, CA transition rules are represented by the binary string corresponding to its output bits; in this example, ⟨00101101⟩. Alternatively, it can be represented by the decimal number corresponding to the conversion of the output bits when read from the right (neighborhood

111) to the left (neighborhood 000). In this example, $(10110100)_2 = (180)_{10}$. Thus, this rule is usually referred as the elementary rule 180. The original lattice at time step $t = 0$ is formed by the sequence $\langle 11101011101000010110111000101001\rangle$. At time $t = 1$, it is possible to see the cells updated using rule 180. The next steps correspond to successive evolutions along nine time steps.

The definitions of the CA connection structure given above can be generalized to arbitrary graph topologies, producing the so-called generalized automata networks (GANs),[10] Network-based CA,[11] or networked CA.[12]

Formally, a GAN is defined as a 4-tuple $R = (G, \Sigma, U, \{fi|i \in V\})$, where $G = (V, E)$ is a graph with a set V of vertices or nodes and a set E of (possibly directed) edges or links that define binary relations on V. Generalizing a CA to a GAN causes the neighborhood size, which is always constant for a standard CA, to have different sizes, since nodes can have different degrees. Thus, the typical rules used in CAs usually cannot express GAN relations, and as a consequence studies on GANs tend to more strongly consider topological aspects, involving concepts of graph theory and complex networks. Initial work was on random GANs (i.e., random graphs), and studies have been focused on the so-called small-world[10,13–16] and scale-free topologies.[10]

3. SMALL-WORLD NETWORKS

A small-world network can be represented by a graph with a connection pattern that is intermediate between completely regular networks and random networks.[13] More consistently, a process for constructing small-world networks was first proposed by Watts and Strogatz.[13] The initial step in constructing the network is defining a circular lattice formed by connected vertices, where each vertex i is connected to r neighbors to the left and right, thus defining the degree for every vertex i as $z^i = 2r$. Then, each network connection is reconnected to a random destination node with probability p, obeying the following restrictions: (a) self-connections are not allowed and (b) multiple connections between the same nodes are not allowed.

Networks built with p values close to 0 have greater regularity in their connection structures, while networks built with p values close to 1 tend to approach networks with highly unregular (random) connection patterns. Intermediate values of p generate networks with the so-called small-world characteristics: high local clustering, quantified as a high clustering coefficient, and a low average distance between any two nodes, quantified by a small average path length.

The clustering coefficient of a network is defined as $\mathrm{CC} = \frac{1}{n}\sum_{i=1}^{n} \tau^i$, where n is the number of nodes and τ^i is the fraction of neighbors of i that are neighbors themselves. The average path length is defined as $\mathrm{APL} = \frac{1}{n(n-1)}\sum_{i,j} d_{ij}$, where d_{ij} is the minimum distance between nodes i and j. On average, random networks produce low values for APL and CC, and highly regular and structured networks, such as regular lattices, reach high values for CC and APL. Small-world networks give rise to high values for CC and low values for APL. According to Ref. 13, networks with reconnection probabilities as low as $p \approx 0.01$ can already exhibit small world properties.

In Ref. 8, a process for constructing small-world networks adapted for generating networks that can operate on the same set of rules used by CA was presented. The process is performed on a directed graph where the output degree (origin arc) of each vertex is kept unchanged. To achieve the small-world effect, reconnections are performed only in the destination nodes with a probability p, thus allowing the use of CA rules for network evolution.

4. ELEMENTARY RULES

The set formed by the elementary rules corresponding to all possible rules of radius 1 ($R = 1$) encompasses 256 rules. Since the number of rules in the set is not too large, we can examine them in their entirety. There are different ways to classify the behavior of dynamic transition rules, the most common classification was proposed by Wolfram[2] and divides the set of rules into four classes as follows:

- Class I: Most initial configurations converge after a transient period, to a homogeneous configuration.
- Class II: Most initial configurations converge after a transient period to a fixed point or a periodic cycle of configurations, possibly with a right or left shift.
- Class III: Most initial configurations result, after a transient period, in a behavior with no easily distinguishable patterns. Rules for this behavior are called chaotic.
- Class IV: Some initial configurations result in complex localized structures, sometimes for a long number of iterations.

As the rules belonging to Class III do not have a well-defined pattern in their evolutions, they are used in the construction of many cryptographic systems based on CAs.[3–7]

Some elementary rules can be represented by additive logical operations. For example, the transition rule $\langle 01011010 \rangle$, or rule 90, can be alternatively described by the logical operation $c_{t+1}^{i} = c_t^{i-1} \oplus c_t^{i+1}$. If the transition rule can be represented using the logical XOR operator, it is called a *linear* rule. Similarly, it is a complementary rule if it can be represented using the logical operator NXOR. Nonadditive rules involve logical operators AND or OR on their descriptions.[6] Linear and complementary rules are employed in cryptographic methods based on algebraic properties of additive CAs.[6,7]

Cryptographic methods based on CA preimage calculation[3–5] usually employ rules with a property referred as sensitivity. A rule is considered sensitive if the change in the state of a neighborhood cell (in a fixed position) necessarily changes its output bit mapping. To illustrate, consider the rule 180 shown in Figure 1. The "000" neighborhood produces the output bit "0" ($\mathbf{0}00 \rightarrow \mathbf{0}$) and the change in the left neighborhood bit produces the output bit "1" ($\mathbf{1}00 \rightarrow \mathbf{1}$). If a rule retains this feature for the entire neighborhood, such as rule 180, this rule is sensitive to the left. Similarly, if the modification of the right bit of the neighborhood produces changes in the output bit, the rule is said to be sensitive to the right. Furthermore, it is possible that a rule has sensitivity on both sides. Rules with sensitivity of their extreme left or right bits are, respectively, called left-sensitive and right-sensitive rules, and are

Table I. Entropy values from bit sequences with a clear pattern and without a pattern.

Bit sequence C	Entropy $\lambda(C)$
0101010101010101	0.2500
0011001100110011	0.5000
0110000000000000	0.4054
0000000100000000	0.3278
0111010101000011	0.8632
1101001011000101	0.9062
0111110101001000	0.9062
1010110000100111	1.0000

of particular interest in this work. Such rules are used to calculate the preimage and generally produce a chaotic behavior. However, we are interested in investigating the dynamic behavior of this type of rule when they are not classified as chaotic.

5. EXPERIMENTS

Experiments were divided in two parts. In the first, we evaluate if rules classified as nonchaotic when using a standard CA connection could propagate a small initial perturbation along the cells of the DSW-CA network. In the second part, we investigate the transition between network states induced by chaotic and nonchaotic rules. For all experiments, we considered rules with sensitivity to any of the extremities, as this characteristic is associated with the possibility of these rules being used in preimage calculations in cryptography.[3–5,17,18]

5.1. Entropy

To evaluate the spread of the perturbation, we used the entropy measure proposed by Shannon.[19] The entropy of a sequence of k events is defined by: $-\sum_{i=1}^{k} p_i \times log_2 p_i$, where p_i is the probability of event i. To adapt this measure for our purpose, we define the spatial entropy of a binary word of N bits as the entropy of the occurrence of N sequences of length j. The symbol λ is used as the operator responsible for calculating the normalized entropy between 0 and 1. The operator is defined as $\lambda(C) = -\sum_{i=1}^{N} \frac{p_i(C)\times \log_2 p_i(C)}{j}$, where $j = \log_2 N$ and $p_i(C)$ corresponds to the probability of a bit sequence of size j in C. To illustrate the use of this entropy measure, Table I shows the resulting entropy values from bit sequences with a clear pattern (first four) and without a pattern (last four). We notice that standardized bit sequences and those with a more pronounced unbalance in the number of 0s and 1s reach lower λ values, and that the larger is the variation in $p_i(C)$), the larger is λ.

In the context of the experiments related herein, entropy calculation was made from the difference between the evolution of two almost identical initial configurations but for a bit change. Considering C_0 as the reference initial configuration and $C_0^{'}$ as the modified (perturbed) configuration, the difference is obtained

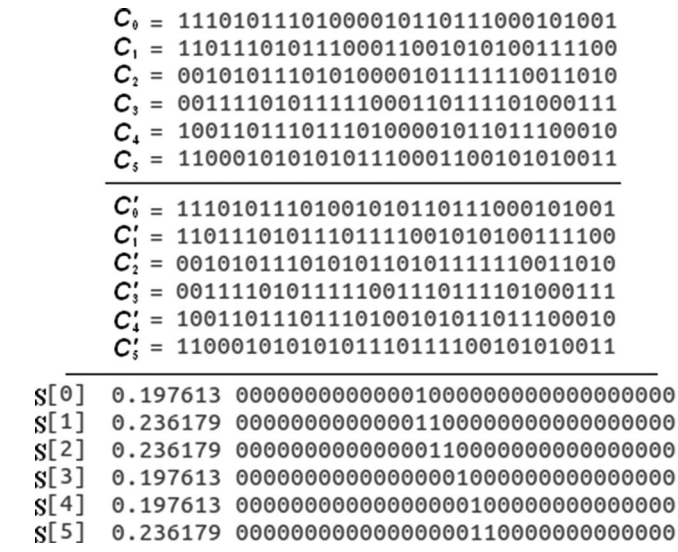

Figure 2. Evolution of original initial 32-bit lattice (C_0), a perturbed version ($C_0^{'}$), and the difference between C_0 and $C_0^{'}$ along five times using a network with $p = 0.0$.

from the XOR operation between C_0 and $C_0^{'}$. The whole process can be summarized as $S_t = \lambda(C_t \oplus C_t^{'})$, where λ is the operator for entropy calculation, $\oplus$ is the XOR operator, and t is the evolution time step. To exemplify, consider the evolution of initial 32-bit lattice $C_0 = 11101011101000010110111000101001$ along five time steps using rule 180 in a standard CA lattice (equivalent to a DSW-CA with $p = 0$). Consider also the evolution of another configuration where the lattice is perturbed with an inversion of the 14th bit (from 0 to 1), that is, $C_0^{'} = 11101011101001\mathbf{1}010110111000101001$. Figure 2 shows the evolution of C_0 e $C_0^{'}$ and the difference between the evolutions with S_t calculated for each time step t, where 0 represents unaltered states and 1 corresponds to differences between evolutions.

One can see in Figure 2 that there are few state variations for each evolution step, with a resulting low entropy even after several evolution steps. This result shows that although rule 180 is a left-sensitive transition rule, it is not adequate to be employed in cryptographic systems as this characteristic can lead someone to make a differential cryptanalysis-like attack.[7] Since rule 180 is classified as a short periodic cycles rule according to the Wolfram classification, the result of the experiment is not surprising because a network with $p = 0$ has the same connection structure of a standard CA.

Figure 3 shows the result of a similar experiment to that depicted in Figure 2. All parameters of the previous experiment were kept (initial configuration, transition rule, modified bit, and evolution time), with the exception of the connection structure of the network, which in this case was built with $p = 0.5$. Initially, the entropy is

```
S[0]  0.197613  00000000000001000000000000000000
S[1]  0.197613  00000000000001000000000000000000
S[2]  0.197613  00000000000001000000000000000000
S[3]  0.497984  00100000001000000100000000000001
S[4]  0.723346  01100110000101000001000000000001
S[5]  0.737661  00100100100100000001100101100001
```

Figure 3. The difference of evolution using an initial 32-bit lattice and a perturbed version along five times using a network with $p = 0.5$.

low because there are almost no differences between the initial lattices, but once more steps of the transition rule are applied as the entropy increases, indicating an increased propagation of the initial perturbation. However, the result shown in Figure 3 indicates that the network connection structure has great influence on the dynamic behavior of the rule, producing a result similar to the one produced by a rule classified as chaotic.[8]

Experiments in Figures 2 and 3 considered a single rule and single initial configuration sample for generating the perturbed version. Such experiments are for particular cases of networks with reconnection probabilities $p = 0$ and $p = 0.5$; however, we conducted additional experiments with a broader set of rules, initial configurations, and reconnection probabilities, namely, 512-bit lattices under 500 initial configurations (CI = 500) and reconnection probabilities $p = \{0, 0.02, 0.04, 0.06, 0.08, 0.1, 0.2, 0.4, 0.6, 0.8, 1.0\}$. For each network, initial configuration, and transition rule, we considered 170 evolution steps ($t = 170$). Subsections 5.1.1–5.1.3 present the corresponding entropy results for rules with radii 1, 2, and 3, respectively.

5.1.1. Radius 1

The set of elementary rules ($R = 1$) that have sensitivity at any of its extremities is composed by 28 rules, four of which are sensitive to both left and right. According to the Wolfram classification,[2] 20 rules in this set belong to Class III, and the other eight rules belong to Class II. The experiments performed herein investigated the eight elementary rules whose behavior is not chaotic and have sensitivity to any of its extremities. Represented in decimal, the rules are: 15, 85, 154, 166, 170, 180, 210, and 240.

Figure 4 shows the results for this set of rules. The ordinate axis corresponds to the average entropy for each of the evaluated networks and the abscissa indicates the time evolution. It can be seen that at $t = 0$ all networks have the same average entropy ($S(0) = 0.0203$), as the difference between the initial settings is a single bit value. The network with $p = 0$ maintains the low average entropy value (around 0.0225) regardless of the evolution time step, but the networks with modified structure have increasing entropy values. According to the results, the higher the probability of reconnection, the smaller the amount of time required for the network to reach its highest average entropy value, taking into consideration that the highest mean value

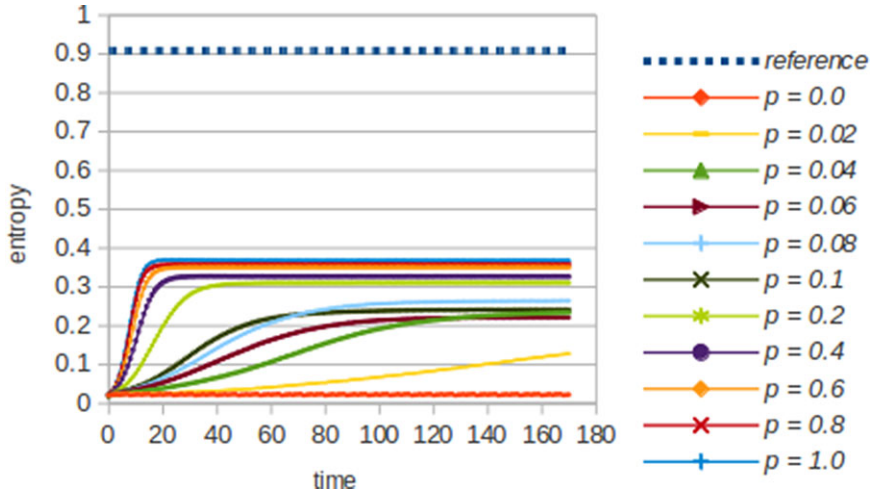

Figure 4. Average entropy by 170 evolutions—experiment with nonchaotic sensitive elementary rules.

of entropy is not the same for the networks evaluated. For example, the network with $p = 0.04$ reached the largest value of the average entropy 0.2348 at $t = 170$, while the network with $p = 1.0$ had the highest average entropy (0.3697) at $t = 29$.

Figure 4 includes a reference value that was calculated as the average entropy of 1000 randomly generated initial 512-bit configurations. Noticeably, the average entropy values achieved by the evaluated rules are well below the reference value for whatever evaluated network. Thus, new experiments were conducted to investigate the rules separately to identify if there are rules that cannot obtain high values of average entropy at all. For these new experiments, the average entropy was also calculated, but together with the minimum and maximum entropy value obtained by the rules considering *each* of the investigated networks. For brevity, we present in this article the results of the network with $p = 0$, for having the lowest average entropy after the 170 steps of evolution, and for the network with $p = 1$, for having the highest average entropy among the investigated networks.

Figure 5 shows the results of the average, minimum, and maximum entropy for each of the rules evaluated considering the network with $p = 0$ and $p = 1$. Rules are on the y-axis (total eight) and the entropy is on the x-axis (average, minimum, and maximum). The results correspond to the entropy evolution at $t = 170$.

Figures 5a and 5b show that the average entropy values of four rules are close to 0, while the other four rules can reach higher values. The results for the other networks are not presented in this article, but it is important to emphasize that this characteristic for the set of rules was also kept in other networks ($p = \{0.02, 0.04, 0.06, 0.08, 0.1, 0.2, 0.4, 0.6, 0.8\}$). Additionally, it could also be noted that the network with $p = 0$ has lower entropy values than the network with $p = 0.02$, and so forth. Finally, all networks had minimal entropy equal to 0 for one or more initial configurations.

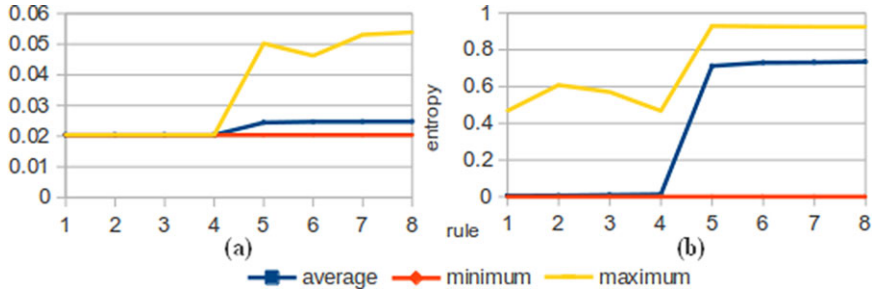

Figure 5. Average, minimum, and maximum entropy after 170 evolutions with $R = 1$. (a) $p = 0$ and (b) $p = 1$.

The results of the experiments indicate that the maximum entropy values for the network with $p = 1$ are similar to the reference value, for all the four rules that can raise the average entropy value. In fact, this result shows that four rules (out of eight), which are not classified as chaotic (considering the Wolfram classification[2]), can behave as if they were chaotic when the network connection structure is properly modified.

5.1.2. Radius 2

The set of rules used in the radius 2 experiments ($R = 2$) was generated in such a way that they are not adequate to propagate perturbations when applied to regular networks. According to the experiments performed using the method described in Ref. 17, a sensitive rule defined by a core with low entropy tends to exhibit a nonchaotic behavior and must be avoided as secret key in the hybrid cellular automata (HCA) method.[17] This observation was also confirmed when using a second cryptographic method named Variable-Length Encryption (VLE)[5], also based on preimage calculus of CA rules. The core of a sensitive rule corresponds to the half of its output bits that are enough to define the entire rule transition, as the other 50% bits are defined as a consequence of the sensitivity property. The set of 100 radius 2 rules used in the experiments reported herein was obtained using a random method to generate 16-bit strings with an entropy level between 0 and 0.5. The experiments aimed to investigate if this set of radius 2 rules can improve the propagation of any initial disturbance in DSW-CA networks.

Figure 6 shows the results similarly to ones in Figure 4, that is, the ordinate axis corresponds to the time evolution and the abscissa indicates the average entropy for each of the evaluated networks. One can observe that the network with $p = 0$ reached the lowest average entropy values for the set of evaluated rules. Unlike the results for radius 1 (Figure 4), however, the radius 2 network that had the highest average value of entropy has a reconnection probability $p = 0.08$ ($S = 0.8258$). The results show that networks with a high probability of reconnection, for example, $p = 1$, tend to increase the entropy value in fewer steps of evolution than the other networks, but upon reaching a certain point, they stop increasing the entropy value.

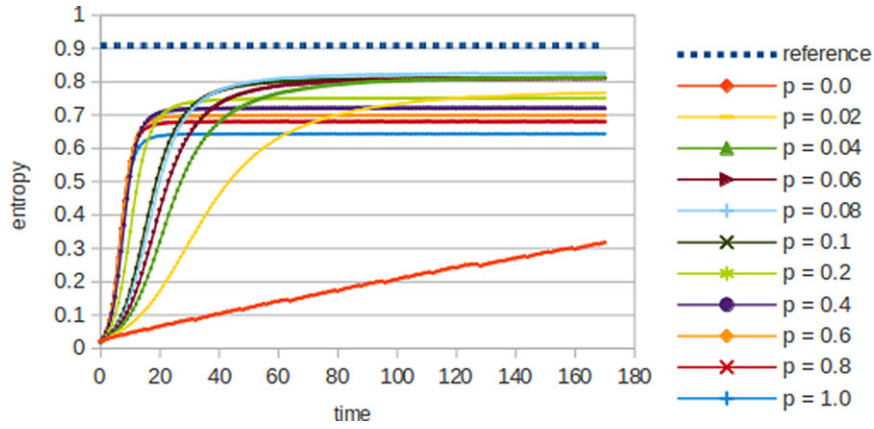

Figure 6. Average entropy by 170 evolutions—experiment with sensitive rules of radius 2 generated with core entropy below 0.5.

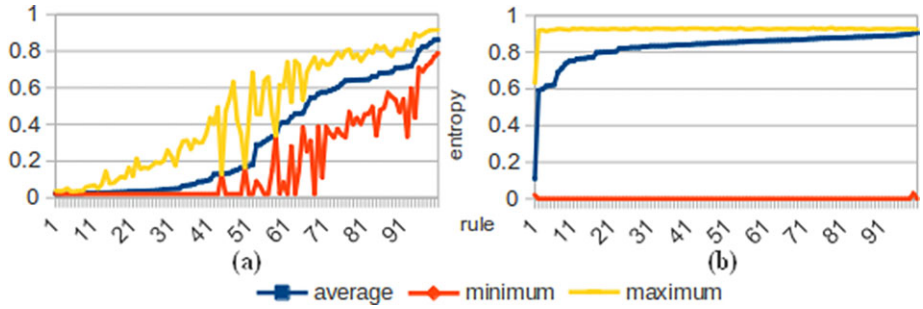

Figure 7. Average, minimum, and maximum entropy after 170 evolutions with $R = 2$. (a) $p = 0$ and (b) $p = 0.08$.

As in the radius 1 experiment, additional tests were conducted using radius 2 rules to evaluate the average, minimum, and maximum entropies for each of the investigated rules. Figure 7 presents the results of the set of rules for the network with lower average entropy ($p = 0$) and for the network that reached higher average entropy ($p = 0.08$).

In Figure 7a, which corresponds to the results of a network with regular connections ($p = 0$), one can observe that several rules have minimum entropy other than 0, and some can reach a value of minimum entropy close to 0.8. This result indicates that although the rule set has been generated in the core entropy below 0.5, there are rules with chaotic characteristics, that is, some rules are capable of propagating an initial disturbance effectively.

Comparing the average entropy shown in Figures 7a and 7b, it is evident that more than half of the rules reach average entropy over 0.8 using networks with $p = 0.08$. However, in Figure 7b, the results indicate that 98 out of 100 rules

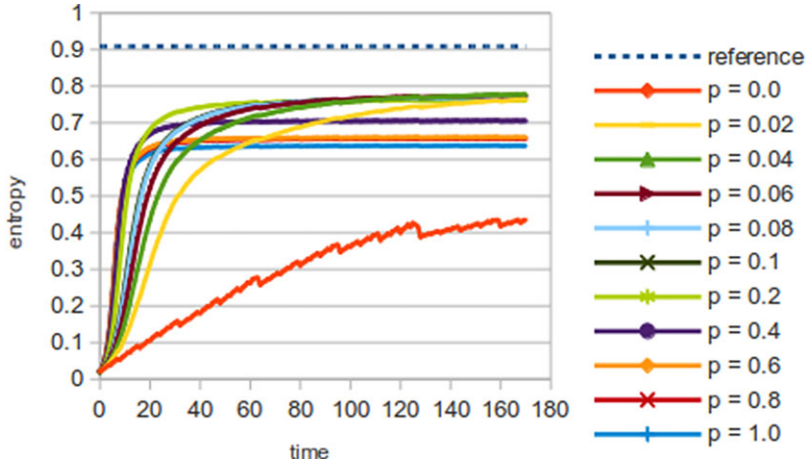

Figure 8. Average entropy by 170 evolutions—experiment with sensitive rules of radius 3 generated with core entropy below 0.5.

investigated had minimal entropy 0, and the other two had minimum entropy values near 0.

The radius 2 experiments thus indicate that the change in the connection structure for $p = 0.08$ enables increasing the average entropy of the rules, but, on the other hand, increases the probability of getting zero entropy ($S = 0$) from the set of initial lattice configurations.

5.1.3. Radius 3

The experiments with rules of radius 3 ($R = 3$) were performed using a set of 50 rules, with the sensitivity set by hand, due to the difficulty in arbitrarily to obtain a core of 128 bits with entropy below 0.5. The set of rules were generated from various cores with a large imbalance between 0s and 1s, such as the bit sequence {1000...000}, and variations of highly standardized sequences, for example, {010101...010101}, {00110011...00110011}, or {000...111...}. The objective of the experiments was to determine how such a rules, which generally have a very standardized behavior when applied to regular connection structures, behaves when operating in DSW-CA networks.

The results of experiments can be seen in Figure 8. The experiments were performed similarly to the radius 1 (Figure 4) and radius 2 (Figure 6) experiments. Similarly, we notice that networks with $p = 0$ have the lowest average entropy values considering the network evolution time. For these experiments, the network with $p = 0.04$ had the highest average entropy value ($S = 0.7788$). Analogous to radius 2, the radius 3 experiments indicate that networks with higher probability of reconnection tend to produce higher values of entropy in the first steps of evolution, however, upon reaching a certain point they tend to a stable state while maintaining the entropy values approximately constant.

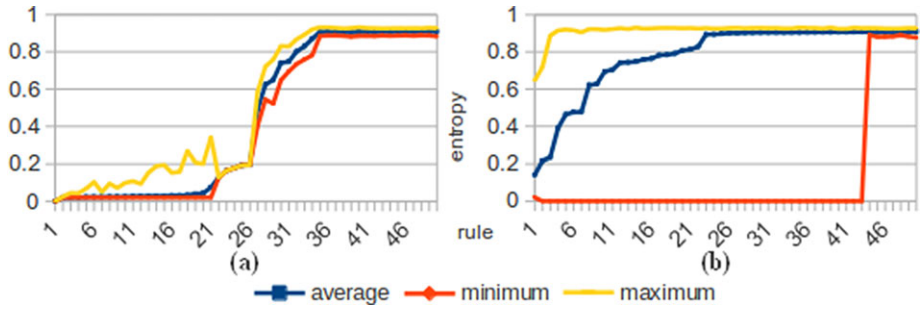

Figure 9. Average, minimum, and maximum entropy after 170 evolutions with $R = 3$. (a) $p = 0$ and (b) $p = 0.04$.

Analogously to the previous experiments, additional tests to evaluate the average, minimum and maximum entropies for each of the rules were carried out. The results of network experiments with $p = 0$ and $p = 0.04$ are in Figure 9.

The results in Figure 9a, which correspond to a regular connection structure ($p = 0$), disclose that 19 rules reach average entropy values over 0.8 and 27 rules have an average entropy below 0.5. We can note that the number of rules with minimum entropy below 0.5 is the same as observed for the average entropy (27 rules), and the number of rules with maximum entropy over 0.8 (21 rules) approaches the observed quantity for average entropy.

In results presented in Figure 9b for network with $p = 0.04$, we can see than 31 rules reached average entropy over 0.8 and seven rules have average entropy below 0.5. Also, only two rules have maximum entropy below 0.8. However, most of the rules (43 total) have minimum entropy near 0.

The results reinforce the idea that the changes in the regular network connection structure allow to increase the average entropy of the rules, but, on the other hand, makes the zero entropy more likely among the initial settings.

5.2. Transition Map

A transition map is a directed graph that represents all the transitions from all possible initial configurations for a network size N. In this context, the vertices correspond to the settings and the arcs to the transitions between settings. The number of possible transition maps generated by a network corresponds to the amount of transition rules available for a radius. For example, rules of radius 1 ($R = 1$) are able to generates 256 different transition maps for a single automata network.

Transition maps do not reveal the dynamic behavior of a rule, but are useful to graphically view the evolution flow performed by a rule. One of the major limitations of the transition maps are in the impossibility of analyzing networks with a large number of vertices. In fact, a network with 16 vertices ($N = 16$) already makes it difficult to see the connections between the $2^{16} = 65{,}536$ possible configurations.

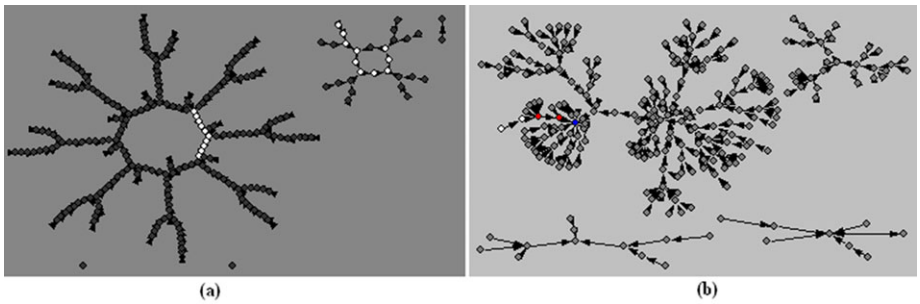

Figure 10. Transition map of rule 30 with a network of size $N = 8$. (a) $p = 0$ and (b) $p = 1$.

The main objective when performing experiments with transition maps for the present work is to investigate why the modified networks evolve to minimum entropy ($S = 0$), as observed in the results of subsection5.1. In fact, zero entropy occurs when the two initial configurations evolve to a common configuration, thus at that point there is no difference between the states in the configurations reached. Furthermore, the following evolutions also result in the same configurations, thus maintaining zero entropy whatever the number of evolutions from the next steps. However, we are interested in verifying whether the change in the network structure can be one of the reasons that causes zero entropy from the difference between two initial settings.

The experiments with transition maps were made with networks of size $N = 8$ (256 configurations/vertices) for easy viewing of the transitions between configurations. In the following sections, we present the results for experiments with radius 1 and 2.

5.2.1. Radius 1

The radius 1 experiments were performed with the elementary rules 30 ({01, 111, 000}—classified as chaotic[2]) and 154 ({01011001}—classified as periodic[2]). The rules were used in the evolution of two different types of networks, one generated with $p = 0$ and the other with $p = 1$.

Figure 10 shows the transition maps of rule 30. Some vertices were highlighted as white blobs to illustrate the evolution of the initial setting {00011100} and its variation in a single bit {000**0**1100} (fourth cell) by 10 time steps. Comparing the two transition maps we can see that the modification in the connection structure causes a large change in the transition map of rule 30.

In Figure 10a, we can see that the initial configuration and its variation are in separate components. Thus, it is impossible for these two initial configurations to result in zero entropy ($S = 0$). However, in Figure 10b, we can see that the two initial configurations are in the same component and evolved to a common configuration, thus producing zero entropy. The vertices highlighted in red correspond to states in which the initial configurations have evolved to the same state. In fact, one of the

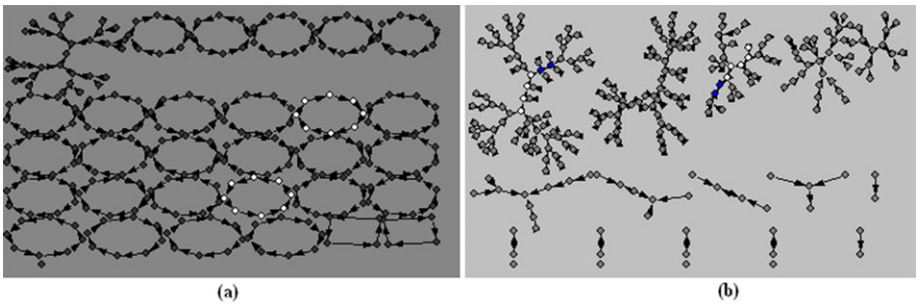

Figure 11. Transition map of rule 154 with a network of size $N = 8$. (a) $p = 0$ and (b) $p = 1$.

configuration was already on the way in which the other would evolve. After some evolutions, the two initial configurations reached an attractor configuration, marked in blue, resulting in zero entropy for the difference between the evolutions of the initial configurations.

The transition maps of rule 154 using networks with probabilities of reconnection $p = 0$ and $p = 1$ can be seen in Figure 11. As in the transition maps of rule 30, we highlighted the evolution of the initial set {00011100} and its variation {000**0**1100} by 10 time steps.

In Figure 11a, we can see that the initial settings belong to separate components. It is also observed that the transition map has various components forming cycles. New experiments to analyze the evolution of each cycle were then performed. In these new experiments, it was possible to note that the transition rule 154 (classified as periodic) produces cycles formed from the left shift of a configuration with a small variation in some bits. As an example, the initial configuration {00011100} evolves into the configuration {00111010}, which then evolves into {01110001}. This behavior causes the entropy of the difference between the evolution of the two initial configurations to remain low, because the initial configuration (which is formed by changing only 1 bit) belongs to another cycle, and next evolutions also causes only a left shift with few variations of the initial configuration. The mere fact that an initial configuration belongs to a different cycle of other initial configuration does not mean that the configurations have large differences in the arrangement of their bits.

In Figure 11b, we can see that the modification in the network connection structure provokes significant changes in the transition map of rule 154. White vertices correspond to the evolutions of the two initial settings previously mentioned. We can see in Figure 11b that the two initial settings evolve in separate components in the transition map. Also, in Figure 11b, four vertices are highlighted in blue, two in one of the components and the other two in another component. These vertices represent states attractors, or configurations that evolve between each other. If an attractor vertice is reached, the transitions switch between them. When a configuration evolves into an attractor vertex, or a small set of attractors vertices, the propagation of an initial disturbance is compromised because the next evolutions will result in repeated configurations.

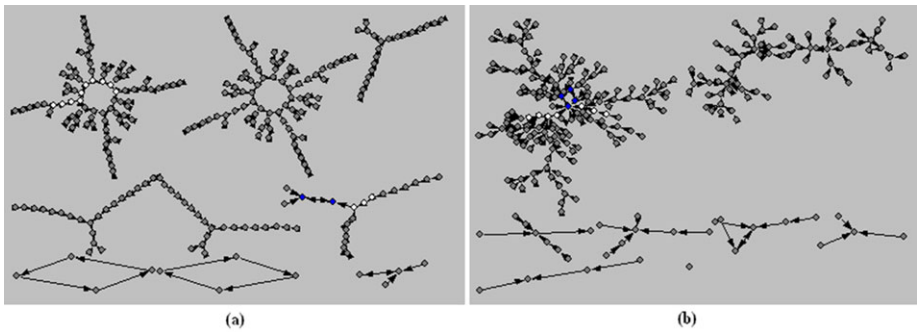

Figure 12. Transition map for core 1 with a network of size $N = 8$. (a) $p = 0$ and (b) $p = 1$.

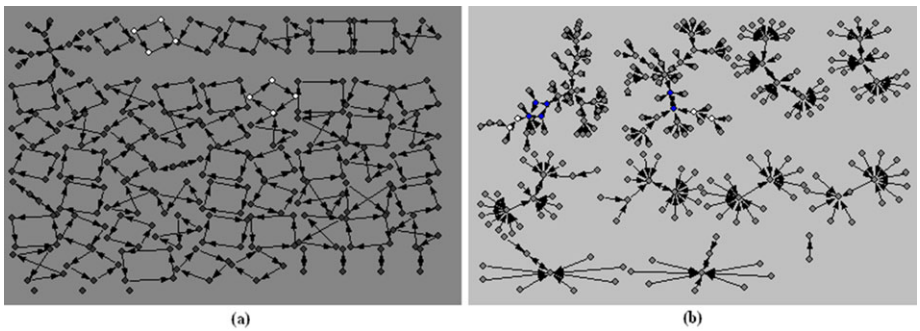

Figure 13. Transition map for core 2 with a network of size $N = 8$. (a) $p = 0$ and (b) $p = 1$.

5.2.2. Radius 2

The radius 2 experiments were also performed with two types of networks. Two rules employed in experiments on entropy (Section 5.1) were selected to produce the transition maps of radius 2 rules, namely a rule that reached entropy values close to the reference value for most experiments (core 1 {1111111111000111}) and a rule that obtained low entropy values (core 2 {0000000010000000}). The networks were generated with probabilities of reconnection $p = 0$ and $p = 1$.

Figure 12 shows the results for the transition maps for core 1, in networks generated with $p = 0$ and $p = 1$. In Figure 12a, we can see that the initial configurations evolve to a set consisting of four attractors highlighted in blue. In Figure 12b, the initial configurations have evolved into separate components, making it impossible to achieve zero entropy from the difference between evolutions.

Figure 13 shows the results for the transition maps for core 2. In the two figures, we can see that the initial configurations evolve into separate components.

5.3. Comments about Transition Map Experiments

The transition map for core 2 and $p = 0$ (Figure13a) has similarity to the transition map of rule 154 and $p = 0$ (Figure 11a). In a more detailed analysis, we can identify that the cycles shown in Figure 13a have the same characteristics as the cycles of Figure 11a, that is, the cycles are formed from bit shifts with few changes between the configurations. In fact, the two transition maps evolve configurations into separate components, but the entropy of the difference between evolutions reached low entropy values, as the initial configurations are very similar. On the other hand, the transition map of Figure 13b allows the average entropy to increase; however, it is more likely to achieve zero entropy.

Observing the transition maps of radius 1 and 2, we can notice a certain similarity between the results produced by networks with $p = 0$ when submitted to rules that are considered good to propagate a disturbance (Figures 10a and 12a). Likewise, we can see similarities between the results of networks with $p = 0$ when submitted to rules that are not good to propagate a disturbance (Figures 11a and 13a). In general, the networks with reconnection probability $p = 1$ obtained transition maps with similar characteristics for all investigated rules (Figures 10b, 11b, 12b, and 13b).

Analyzing the transition maps produced by the networks with $p = 1$, we can see that two configurations chosen arbitrarily have larger probability to evolve to zero entropy, if compared with network transition maps with $p = 0$. Other experiments with transition maps using a larger network size ($N = 12$) were performed and it was also possible to note that the networks with $p = 1$ tend to produce maps with a higher probability of zero entropy. This observation is reinforced when analyzing the results presented in subsection 5.1 in Figure 7 or results of Figure 9, wherein the networks with $p = 1$ obtained minimum entropy $S = 0$ for most of the initial configurations analyzed.

6. CONCLUSIONS

This study made a comparative analysis of the dynamic behavior of CA rules when applied to networks of CA. Empirically, we showed that the dynamic behavior of rules applied to the network CA is associated with the network connection pattern. The experiments indicate that rules considered inadequate to propagate an initial perturbation when applied to a regular structure connection and can improve the perturbation dissemination by modifying the network connections. These results corroborate the results published in Ref. 9, which in practice indicate that it is possible to increase significantly the space of possible keys for cryptographic methods based on CA and sensitive transition rules.[3–5,17,18] However, further experiments show that the modification of network connections structure increases the chances of two initial configurations evolve to the same state, which can result in a vulnerability. New studies are being conducted to construct a kind of network that has changes in its structure connections, but that eliminate the evolution of two initial configurations for the same state. Thus, it will be possible to increase the space of

keys in cryptographic methods based on CA rules, supporting the hypothesis that CA networks can be used as tools for fast and effective cryptographic applications. Initial experiments yielded promising results that will be shown and discussed in a paper currently in preparation.

Acknowledgments

Heverton B. Macêdo thanks CAPES (proc. CAPES/DINTER no. 23038.044846/2009-76). Gina M. B. Oliveira thanks FAPEMIG (proc. APQ 01910-13) and CNPq (proc. 310805/2013-9). Carlos H. C. Ribeiro thanks FAPESP (proc. 2013/13447-3) and CNPq (proc. 303738/2013-8).

References

1. Ganguly N, Sikdar B, Deutsch A, Canright G, Chaudhuri P. A survey on cellular automata. Technical Report, Centre for High Performance Computing, Dresden University of Technology; 2003.
2. Wolfram S. Universality and complexity in cellular automata. Physica D 1984;10:1–35.
3. Gutowitz H. Cryptography with dynamical systems. In: Goles E, Boccara N, editors. Cellular automata and cooperative phenomena. Dordrecht, the Netherlands: Kluwer; 1995. Vol 1, pp 237–274.
4. Oliveira GMB, Coelho AR, Monteiro LHA. Cellular automata cryptographic model based on bi-directional toggle rules. Int J Mod Phys C 2004;15:1061–1068.
5. Oliveira GMB, Martins LGA, Alt LS, Ferreira GB. Exhaustive evaluation of radius 2 Toggle rules for a variable-length cryptographic cellular automata-based model. In: Int Conf on Cellular Automata for Research and Industry (ACRI'10), Ascoli Piceno, Italy, 2010.
6. Chaudhuri P, Chowdhury D, Nandi S, Chattopadhyay S. Additive cellular automata: theory and application. IEEE Computer Society Press, Los Alamitos, CA; 1997.
7. Sen S, Shaw C, Chowdhuri R, Ganguly N, Chaudhuri P. Cellular automata based cryptosystem (CAC). In: Proc 4th Int Conf on Information and Communication Security (ICICS02), Singapore; 2002. pp 303–314.
8. Macêdo HB, Oliveira GMB, Ribeiro CHC. Dynamic behaviour of chaotic cellular automata: a comparative entropy analysis of regular lattices and small-world structures. In: Int Conf on System, Man, and Cybernetics (IEEE SMC, 2013), Manchester, England; October 13–16, 2013. pp 1566–1571.
9. Macêdo HB, Oliveira GMB, Ribeiro CHC. Dynamic behaviour of network cellular automata with non-chaotic standard rules. 2nd World Conference on Complex Systems (WCCS 2014), Agadir, Morocco; November 10–12, 2014. pp. 451–456.
10. Tomassini M. Generalized automata networks. 7th Int Conf on Cellular Automata, for Research and Industry (ACRI 2006), Perpignan, France; September 20–23, 2006. pp. 14–28.
11. Andreica A, Chira C. New majority rule for network based cellular automata. Informatica 2012;LVII(3):35–40.
12. Nochella J. Cellular automata on networks. unpublished. 2006. Available at https://www.wolframscience.com/conference/2006/presentations/materials/nochella.pdf
13. Watts DJ, Strogatz SH. Collective dynamics of "smallworld" networks. Nature 1998;393:440–442.
14. Tomassini M, Giacobini M, Darabos C. Evolution of small-world networks of automata for computation. Parallel Problem Solving from Nature VIII, Birmingham, UK; September 18–22, 2004. pp 672–681.
15. Tomassini M, Giacobini M, Darabos C. Evolution and dynamics of small-world cellular automata. Complex Syst 2005;15:261–284.
16. Gog A, Chira C. Dynamics of networks evolved for cellular automata computation. IN: HAIS (2). Lect Notes Comput Sci 2012;7209:359–368.

17. Oliveira GMB, Macêdo HB. Sistema criptográfico baseado no cálculo de preimagem em autômatos celulares n ao-homogêneos, n ao-aditivos e com dinâmica caótica. Patent deposit at INPI-Brazil under number PI0703188-2, September 2007.
18. Oliveira GMB, Macêdo HB, Branquinho A, Lima MJL. A cryptographic model based on the preimage computation of cellular automata. Automata-2008: Theory and Applications of Cellular Automata. Luniver Press, Bristol, UK; 2008. pp 139–155.
19. Shannon CE. A mathematical theory of communication. Bell Syst Tech J 1948;27:379–423 and 27:623–656.