R. Dogaru and Ioana Dogaru, „Efficient and Cryptographically Secure Pseudorandom Number Generators based on Chains of Hybrid Cellular Automata Maps", in Proceedings of the 2014 10th International Conference on Communications (COMM), final version of the paper is available on IEEE Xplore - index:
http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6866716&queryText%3DEfficient+and+Cryptographically

Efficient and Cryptographically Secure Pseudorandom Number Generators based on Chains of Hybrid Cellular Automata Maps

DRAFT

Radu Dogaru[1,*], Ioana Dogaru[1]
[1]*University "Politehnica" of Bucharest, Natural Computing Laboratory, Applied Electronics and Information Engineering, Bucharest, Romania*
[*]Corresponding author (E-mail : radu_d@ieee.org)

***Abstract*— In this paper we consider several solutions for designing cryptographically good pseudo random number generators (PRNG) with low complexity implementations. Our solutions are based on hybrid cellular automata (HCA) maps. The first solution (ensuring maximal throughput) is based on creating chains of HCA maps, i.e. the nonlinear map is changed dynamically being controlled by another HCA map within a chain. The second solution is simpler but somehow reduces the throughput and it is based on the down-sampling of a single HCA output with a factor *d*. It is shown that both solutions pass all statistical tests of NIST.**

***Keywords*-component; cryptography; random number generator; chaos; cellular automata; NIST statistical test suite**

I. INTRODUCTION

Designing good pseudo-random number generators is the focus of many researchers. A good pseudo-random number generator (PRNG) must be cryptographically secure, a property that is often certified if it passes the standard battery of NIST tests [1]. But on the other hand, there are other requirements such as the throughput and hardware complexity. Modern stream ciphers require throughputs in the order of Gigabytes per second, i.e. the possibility to provide a new PRNG bit (or word) at a very high frequency rate while ensuring the cryptographic requirements. Consequently, it is important to define PRNG architectures capable to pass the NIST tests while maintaining a very low architectural complexity and high throughput.

In a recent study [2] various proposals submitted to eSTREAM competitions were analyzed from cryptographic perspective with a focus on both chaotic maps and cellular automata. Although, apparently they are different categories, it is shown in [3] and in section II that when implemented in a digital system they represent the same discrete-time nonlinear dynamical system with a finite word representation of the state vector.

In [2] the main disadvantages listed for chaotic maps were a high implementation complexity (due to the complicated arithmetic operators) and their relatively low throughput, partially due to the need to de-correlate signals using under-sampling (down sampling) with relatively large sample distances *d* [4] . Consequently, the apparent advantage of using the high-dimensional state space (still a discrete one) associated with floating-point representations is counterbalanced by the complexity of implementing the maps, also impeding on throughputs. This critique is given in [2] for most of the Baptista-scheme [5] chaotic maps. Only the Rabbit fixed-point chaotic map [6] is mentioned in [2] as very close to meeting the highest cryptographic requirements (as it was cited among the finalist of the eSTREAM competition). It is worth mentioning an original approach in [14] where the running-key method applied to chaotic tent maps was shown to produce good PRNG passing the battery of NIST tests

Cellular Automata solutions for cryptographically safe PRNG were also proposed in previous papers (see [7] for a recent solution and previous work overview) as a more compact and low complexity alternative to chaotic maps. According to [2] linear CA solutions (such as those proposed in [8]) should be avoided since they are prone to have a low immunity to algebraic and other kind of attacks. Consequently, most of the recent work moves in the direction of using CA with nonlinear cells (a similar direction is taking place in replacing LFSR with NLFSR). For instance, the solution in [7] selects from a larger family of 5-cell neighborhood CA previously found in [9] as good random number generators and uses a down-sampling method (see section III) with d=4 to generate bit-streams that were further analyzed with the NIST battery of tests. Only 3 CA cells were finally selected as meeting the cryptographic criteria. Although authors in [7] do not provide comments on linearity of their cells, using our CA analysis tools [10] it was found that their cells are indeed nonlinear, although they are not conservative (consequently they have many transient states). The throughput of their solution is 4-times lower than the clock frequency.

A modern tendency in ensuring good properties for CA-based PRNG is the use of hybrid architectures (i.e. using several different types of cells). Although some authors [7] consider hybrid architectures as having higher complexity, our HCA solutions [11] adds only one additional input to each cell controlling if the cell output is or not negate. It is the simplest possibility to define hybrid CA.

Herein we continue in the direction previously reported in [3,10,11] showing that by chaining the HCA results in a PRNG capable to pass all of the NIST battery of tests with a low implementation complexity and a maximal throughput given by the clock frequency. Section II reviews the definition of hybrid cellular automata providing the basic PRNG unit to be used as basic building block for cryptographically secure PRNG. Section III discussed two approaches to improve security while Section IV summarizes the evaluation of our solutions from the perspective of NIST tests.

II. CELLULAR AUTOMATA AS NONLINEAR MAPS

Cellular automata are particular cases of nonlinear automata maps evolving in a discrete time and having a discrete state variable represented with a finite number of bits *n*. An exemplification of this concept is given in Fig. 1. Chaotic maps, particularly the fixed-point integer chaotic maps also fit in this general formalism (assuming that all state variables are represented with a finite *n* number of bits, as it is the case in any digital system).

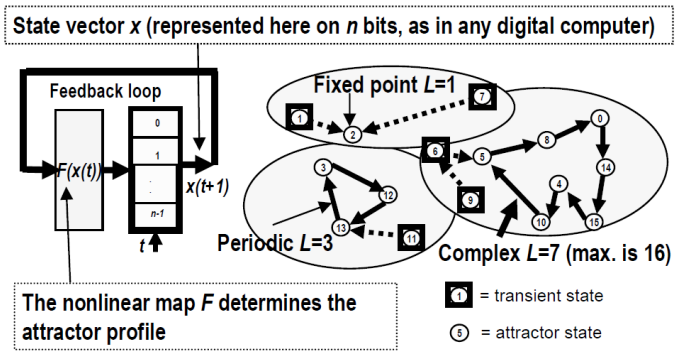

Figure 1. General structure of a nonlinear dynamical system and the state space profile induced by the nonlinear feedback F.

The CA maps differ from classic y=F(x) maps only in the particular manner of defining the nonlinear function F. In the case of CA, F is the collective result of applying *n* times the rule (or cell) functions in the *n* neighbourhoods consisting in *m* adjacent cells. Consequently, CA maps have a low implementation complexity reduced to *n*-times the devices allocated to one single cell. As shown previously [12] in FPGA implementations, one HCA cell is allocated to only one basic logical element.

A. Hybrid Cellular Automata (HCA) as conservative dynamical systems

The HCA model first introduced in [11] is based on iterating the next very simple equation:

$$x_i(t+1) = m_i \oplus Cell\left(x_{i-1}(t), x_i(t), x_{i+1}(t)\right), ID\right) \quad (1)$$

which applies synchronously to all *n* CA cells (a cell is identified by an index $i \in \{1,2,..n\}$ and has a binary state $x_i(t)$). In the above equation a *m=3* cell neighbourhood is considered (3 inputs per cell) but the model can be easily expanded to a larger number of inputs (e.g. we recently explored the much larger space of 5 inputs cells [10]). A periodic boundary condition is assumed i.e. the leftmost cell (*i=1*) is connected to the rightmost one (*i=n*). The binary *mask vector* $\mathbf{m} = [m_1, m_2, .., m_n]$ provides the hybrid character of the CA and it acts as a control parameter for the associated map, as seen in Figs 3 and 4. The particular rule function $y = Cell(x_{i+1}, x_i, x_{i-1}, ID)$ among all 2^m possible functions is specified by the cell ID. As shown in [10,12] an algebraic normal form representation can be associated with each particular ID, the algebraic normal form having several advantages such as: i) it provides a simple and efficient way for hardware description; ii) clearly shows whether a cell is linear or not – a linear cell has only 1'st degree terms.

It is the rule equation (and its associated ID) the one giving the desired dynamics. Depending on the neighborhood size one may find less or more rules providing CA with good cryptographic properties.

A pre-selection of rules (among all 256 possible if the neighborhood *m*=3 cells, and among all 4 billion possibilities for the case *m*=5) is done with CA analysis tools according to some criterions, as detailed in [10]. The most important criterions in selecting rules is the presence of chaotic behavior and of the nonlinear dynamics conservative property. It means that no state is a transient one, i.e. all states are included in several cycles (see Fig.1). The CA will run on the longest cycle which also supposed to have a high degree of chaos.

The typical analysis for one single ID lasts on the order of fraction of seconds which makes such tool very useful for a pre-selection (NIST tests take on the order of tens of minutes).

So far we were able to locate several tens of rules providing very good conservative PRNG among all 4-inputs in 5-cell neighborhood rules.

Herein we focus on one of the simplest rule (ID=101). As discussed in [11] the mask vector can be optimized to maximize the period of the operating cycle as required in certain specialized application (when the PRNG acts as a chaotic counter) [13] but for the purpose of good PRNG arbitrary mask vectors can be chosen (representing part of the key, along with initial register states and cell/rule functions). The architecture of the HCA and its simplified notation are given in Fig. 2.

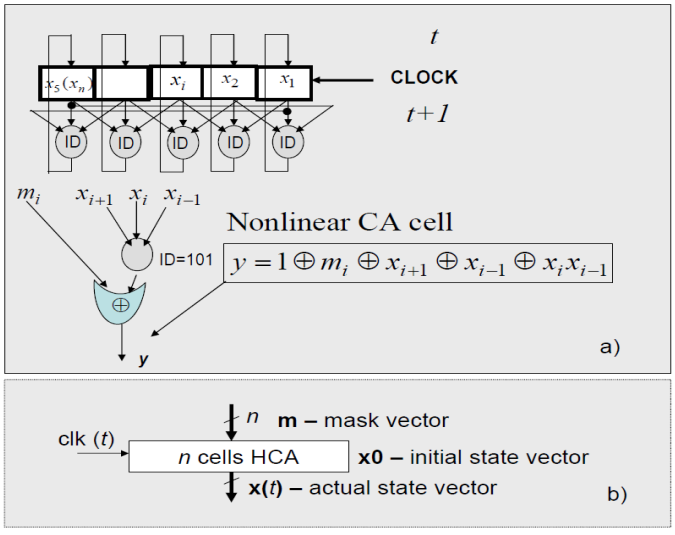

Figure 2. a) HCA architecture and b) its simplified notation

B. *HCA as fixed-value integer chaotic maps*

If the CA state-vector is regarded as a fixed-point number x represented on n bits, map curves y=F(x) may be drawn as seen in Fig.3 for different masks of the corresponding HCA. It is interesting to note the sophisticated, fractal shape of F which is obtained using a very low complexity hardware (n Boolean functions with 4 inputs, each representing the CA cells).

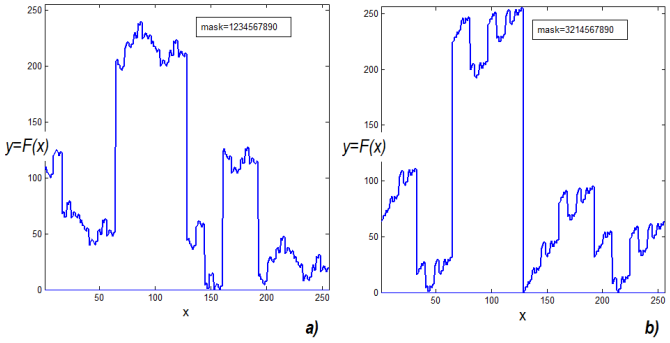

Figure 3. HCA maps for the case of Boolean cell $y_i = 1 \oplus m_i \oplus x_{i+1} \oplus x_{i-1} \oplus x_i x_{i-1}$ and two different mask vectors.

III. SOLUTIONS FOR IMPROVING THE CRYPTOGRAPHIC PROPERTIES

Before analyzing in more detail the proposed solutions it is worth mentioning that the simplest PRNG solution is the HCA with a very large number n of cells. But the next solutions will ensure good cryptographic properties even for relatively small number n of bits (cell) as detailed bellow. Figure 4 presents these two approaches.

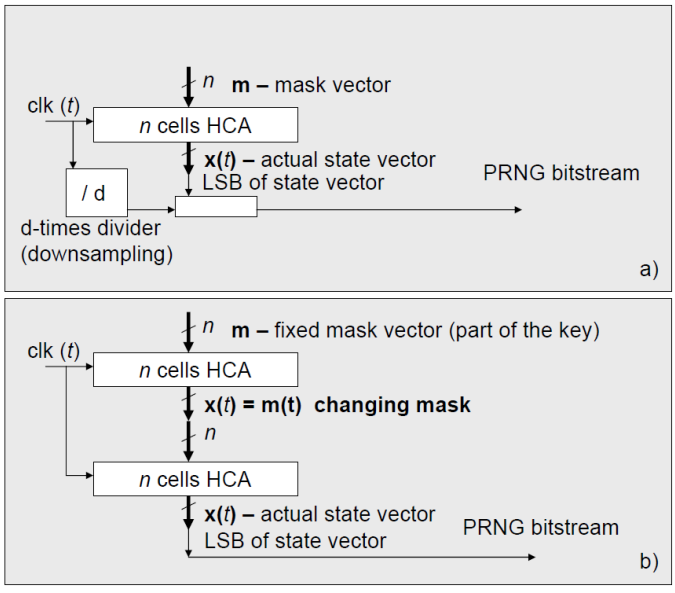

Figure 4. Improving PRNG based on HCA maps: a) using downsampling with a d factor (throughput also decreasing d times); b) using chains of HCA maps (masks are dyankically changing).

i) **Down-sampling** - This solution is rather simple and it aims at de-correlating consecutive bits or words in the PRNG stream by simply sending 1 of d consecutive samples. The main disadvantage of this solution is a decrease of the throughput, proportional to the factor d.

A careful analysis of this solution using tests for statistical independence to determine the optimal d value was carried out in recent works [4]. We noticed that although high d-values (tens) were reported for the case of chaotic maps, as seen in section IV values as low as $d=2$ suffice to pass the NIST tests in the case of HCA maps. In the framework of the CA a similar solution was proposed in [8] and recently employed with $d=4$ in [7].

ii) Chains of HCA – this solution is novel and exploits the existence of a mask vector with the same size n as the state vector. Consequently, a chain with 2 ore more stages (2-stages are shown in Fig.4b) is built where the masks are dynamically changing ensuring that the output HCA has a variable nonlinear map which changes at each clock iteration. This solution has the advantage of maintaining the highest throughput rate (equal to the clock rate) but requires M-times more devices (where M is the number of stages).

IV. ASSESEMENT USING THE NIST BATTERY OF TEST

In order to evaluate PRNGs based on HCA, for each case M=100 bit-streams (sequences) of 1 million bits each were generated (in the form of a single file with 10^8 bits). Running the NIST battery of tests results in a file "finalAnalysisReport.txt" reporting two values for each of the 188 tests applied to the set of binary sequences: i) The proportion P of sequences passing a test, and ii) the distribution of P-values (called next p-value-T). A significance level α = 0.01, as recommended by NIST, was used for the analysis of P-values obtained from various tests.

In order to accept that a certain PRNG passes the NIST test based on 100 sequences with 10^6 bits each, certain thresholds are defined according to NIST procedures [1] for both P proportions and P-values as follows: The range of acceptable proportion is 0.99 ± 0.02985 ([0.96015,1.01985]); The lower threshold for p-values-T (indicating the uniformity of the distribution of P-values) is in this case 10^{-4}.

From a more practical perspective a specific test is passed if for that test $P > 0.96$ and $p-value-T > 10^{-4}$. If at least one test fails it is assumed that the corresponding PRNG fails the cryptographic requirements. **On the opposite, having a good cryptographic PRNG corresponds to passing ALL 188 tests**.

The next table summarizes several variants of PRNG based on HCA with neighborhood 3 and the nonlinear rule function (for the cell i) $y_i = m_i \oplus x_{i+1} \oplus x_{i-1} \oplus x_i x_{i-1}$ and the results of applied the above NIST tests:

PRNG short name	Details of the PRNG architecture		# OF FAILED TESTS (in 188)
	Type of architecture	*PRNG parameters*	
REF	Single HCA	n=31, mask=1200041, d=1	2 (FFT, 1 nonoverlap)
A	Chain HCA 2 stages	n=31, mask=1200041,	NONE

PRNG short name	Details of the PRNG architecture		# OF FAILED TESTS (in 188)
	Type of architecture	*PRNG parameters*	
		d=1	
B	Downsampling HCA 1 stage, d=2	n=31, mask=1200041, d=2	NONE
C	Chain HCA 3 stages	n=15, mask=12041, d=1	NONE
D	Chain HCA 3 stages	n=31, mask=12041, d=1	NONE

An example of the specific NIST test result in the case of PRNG-A is given in Fig.6

A more detailed distribution of *p-values-T* is given in Fig. 5 For reference, the results of the NIST tests for the Blum-Blum-Shub (BBS) PRNG (among the best, but computationally intensive) in the same conditions are also given . It is clear that for all PRNGs passing the whole battery of tests the results are comparable.

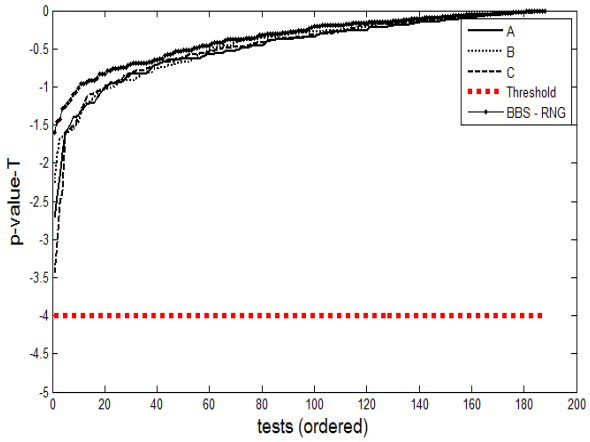

Figure 5. Distributions of (logarithms of) p-value-T values for all 188 NIST tests in the case of PRNGs A,B and C. Note that in all cases the values are above the threshold. The tests are reordered such that *P* values are sorted in an increasing order.

RESULTS FOR THE UNIFORMITY OF P-VALUES AND THE PROPORTION OF PASSING SEQUENCES

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST
9	11	10	4	14	9	12	10	12	9	0.699313	99/100	Frequency
11	12	6	9	11	14	7	12	11	7	0.719747	99/100	BlockFrequency
9	9	11	10	8	9	6	15	8	15	0.554420	98/100	CumulativeSums
11	11	10	6	12	5	8	7	17	13	0.224821	99/100	CumulativeSums
11	11	8	14	5	17	8	9	8	9	0.304126	99/100	Runs
12	8	7	9	12	17	9	8	8	10	0.534146	99/100	LongestRun
9	2	10	17	10	12	14	6	6	14	0.032923	100/100	Rank
11	10	9	12	15	13	9	8	8	5	0.595549	99/100	FFT
11	8	10	8	11	14	11	12	6	9	0.851383	100/100	NonOverlappingTemplate
9	11	5	7	13	15	6	11	12	11	0.419021	99/100	NonOverlappingTemplate
13	10	6	8	11	14	12	12	7	7	0.616305	96/100	NonOverlappingTemplate
...												
17	5	11	8	11	9	10	7	11	11	0.419021	98/100	OverlappingTemplate
11	10	9	6	11	12	15	9	11	6	0.678686	99/100	Universal
11	10	14	10	11	7	15	8	5	9	0.514124	99/100	ApproximateEntropy
10	6	8	10	5	4	7	3	3	6	0.350485	61/62	RandomExcursions
4	9	6	11	6	5	10	4	3	4	0.213309	62/62	RandomExcursions
10	6	7	6	3	3	12	9	3	3	0.060239	60/62	RandomExcursions
.....												
4	6	7	7	5	5	7	6	3	12	0.437274	61/62	RandomExcursionsVariant
5	7	3	9	8	5	4	9	7	5	0.671779	62/62	RandomExcursionsVariant
.....												
6	7	11	8	6	11	13	14	10	14	0.455937	100/100	Serial
7	7	8	8	17	10	9	7	13	14	0.275709	100/100	Serial
15	10	8	9	10	9	14	5	12	8	0.534146	100/100	LinearComplexity

The minimum pass rate for each statistical test with the exception of the random excursion (variant) test is approximately = 96 for a sample size = 100 binary sequences.

The minimum pass rate for the random excursion (variant) test is approximately = 59 for a sample size = 62 binary sequences.

For further guidelines construct a probability table using the MAPLE program provided in the addendum section of the documentation.

Figure 6. Results of applying the NIST tests to the PRNG-A (some of the "Nonoverlapping", "Random Excursions" tests were removed for space reasons).

V. CONCLUDING REMARKS

Very low complexity PRNG solutions meeting cryptographic requirements (passing the NIST battery of tests) were presented. Among these solutions the original solution of HCA chaining has the advantage of maintaining the maximal throughput given by the clock signal while achieving good cryptographic properties even for small number *n* of cells by adequately choosing the number of stages. Such PRNGs are easily scalable to arbitrary *n* values. The resulting PRNG is highly nonlinear being thus similar to NLFSR solutions and consequently is expected to have a high immunity to various kind of attacks. In terms of FPGA implementation, a 2-chain HCA solution for *n* =31 cells was implemented on a Digilent Basys2 board (with a Spartan-3E FPGA chip) resulting in a compact allocation of only 62 LUTs. Other details about FPGA implementations and NIST tests based on signal sequences generated directly on FPGA boards are given in [15].

REFERENCES

[1] A. Rukhin et al., "A statistical test suite for random and pseudorandom number generators for cryptographic applications", NIST Special Publication 800-22 (with revisions dated April, 2010).

[2] Matt Henricksen, "A Critique of Some Chaotic-Map and Cellular Automata-Based Stream Ciphers", in Proceedings ASIAN 2009, LNCS 5913, pp. 69-78, 2009.

[3] R. Dogaru, "HCA101: A chaotic map based on cellular automata with binary synchronization properties", in Proceedings of The 8'th Int'l Conference on Communications-COMM2010, 10-12 June, Bucharest, Romania, pp. 41–44.

[4] Adriana Vlad, A. Luca and M. Frunzete, "Computational Measurements of the Transient Time and of the Sampling Distance That Enables Statistical Independence in the Logistic Map", Lecture Notes in Computer Science, 2009, Volume 5593-2009, pp. 703–718.

[5] M.S. Baptista, "Cryptography with Chaos", Physics Letters A, vol. 240, pp. 50–54, 1998.

[6] M. Boesgaard, M. Vesterager, T. Pedersen, J. Christiansen, and O. Scavenius, "Rabbit: a new high-performance stream cipher", In: Johansson, T. (ed.) FSE 2003, LNCS, vol. 2887, pp. 325–344. Springer, Heidelberg (2003)

[7] Ping Ping, Feng Xu and Zhi-Jian Wang, "Generating High-Quality Random Numbers by Next Nearest-Neighbor Cellular Automata", Proceedings of the 2nd International Conference On Systems Engineering and Modeling (ICSEM-13), pp. 0838-0842, 2013.

[8] M. Tomassini, M. Sipper, M. Zolla, M. Perrenoud, "Generating high-quality random numbers in parallel by cellular automata", Future Generation Computer System 16, pp. 291-305, 1999.

[9] F. Seredynski, P. Bouvry, A. Zomaya, "Cellular automata computation and secret key cryptography", Parallel Computation 30, pp. 753-766, 2004.

[10] R. Dogaru and Ioana Dogaru, "Applications of Natural Computing in Cryptology: NLFSR based on Hybrid Cellular Automata with 5-cell Neighborhood", PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A, Volume 14, Special Issue 2013, pp. 365–372.

[11] R. Dogaru, "Hybrid Cellular Automata as Pseudo-Random Number Generators with Binary Synchronization Property", in Proceedings of the International Symposium on Signals Circuits and Systems, Iasi Romania, July 2009, pp. 389-392.

[12] Ioana Dogaru and R. Dogaru, "Algebraic Normal Form for Rapid Prototyping of Elementary Hybrid Cellular Automata in FPGA", in Proceedings ISEEE 2010 (September 2010, Galati, Romania), pp. 273-276, 2010.

[13] R. Dogaru, Ioana Dogaru, H. Kim, "Chaotic Scan: A Low Complexity Video Transmission System for Efficiently Sending Relevant Image Features, IEEE Trans. on Circuits and Systems for Video Technology, Vol.20, Issue 2, pp. 317–321, 2010.

[14] Adriana Vlad, A. Luca, O. Hodea, R. Tataru, "Generating Chaotic Secure Sequences using Tent Map and a Running-key Approach", PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A, Volume 14 (Special Issue), pp. 295–302, 2013.

[15] Ioana Dogaru, R. Dogaru, "FPGA Implementation and Evaluation of two Cryprographycally Secure Hybrid Cellular Automata", in Proceedings of COMM2014 - Intl. Conference on Communications, Bucharest, May 2014.