International Journal of Technical Innovation in Modern Engineering & Science (IJTIMES)
Impact Factor: 5.22 (SJIF-2017),e-ISSN:2455-2585
International Conference on
Recent Explorations in Science, Engineering And Technology (ICRESET'19)
Volume-5, Special Issue-March, 2019.

Generating Cryptographic Numbers Using Cellular Automaton

Dr.G.Selvavinayagam

Department of Information Technology, KGiSL Institute of Technology

Abstract - The paper presents new results concerning application of cellular automata (CAs) to the numbers using cipher cryptography.CA are applied to generate pseudo-random numbers sequence (PNS) which is used during the encryption process. One dimensional, non-uniform CAs is considered as a generator of pseudorandom number sequences (PNSs) used in cryptography with the secret key. The quality of PNSs highly depends on a set of applied CA rules. Rules of radius r = 1 and 2 for non-uniform one dimensional CAs have been considered. The search of rules is performed with use of evolutionary technique called cellular programming. As the result of collective behavior of discovered set of CA rules very high quality PNSs are generated. The quality of PNSs outperforms the quality of known one dimensional CA-based PNS generators used in the secret key cryptography. The extended set of CA rules which was found makes the cryptography system much more resistant on breaking a cryptography key.

Keywords - Cellular automata; Cellular programming; Random number generators; Symmetric key; cryptography

The paper is organized as follows. The following section presents the idea of an encryption process based on Vernam cipher and used in CA-based secret key cryptosystems. Section 3 outlines the main concepts of CAs, overviews current state of applications of CAs in secret key cryptography and states the problem considered

In this paper Section IV outlines evolutionary technique called cellular programming and shows how this technique is used to discover new CA rules suitable for encryption process. Section V contains the analysis of results and concludes the paper.

I. INTRODUCTION

Confidentiality is mandatory for a majority of network applications for example commercial uses of the internet. Two classes of algorithms exist on the market for Data encryption: secret key systems and public key systems. An emerging cryptography techniques used in both types of system. One of such a promising cryptography techniques are cellular automata. Cellular automata are highly parallel and distributed systems which are able to perform complex computations. New perspectives in this area have been opened when evolutionary techniques appeared and have been used to design automatically CA based system.

CAs were proposed for public key cryptosystems by Guan and Kari [9] .In such systems two keys are required: one key is used for encryption and other is used for decryption, and one of them is held in private, other is published. However the main concern of this paper is secret key cryptosystems. In such system the same key is used for encryption and decryption. The encryption process is based on the generation of pseudorandom bit sequences, and CA is used for this purpose. In the context of secret key systems, CA were first studied by wolfram [17], and later by Nandi et al. [20] and Gutowitz [8]. Recently they were a subject of study by Tomassini and his colleagues [12]. This paper extends these recent studies and describes the application of one-dimensional (1D) CAs for the secret key cryptography.

II. SECRET KEY CRYPTOGRAPHY

Let P be a plain-text message consisting of m bits P1 P2. . . Pm, and k1 k2 . . . km be a bit stream of a key K. Let C i be the ith bit of a cipher-text obtained by applying a (exclusive-or) enciphering operation: Ci = Pi Ki

The original bit P_i of a message can be recovered by applying the same operation on c_i with use of the same bit stream key k: $P_i = Ci , K_i$

The enciphering algorithm called Vernam cipher is known to be [5, 9] perfectly safe if the key stream is truly unpredictable and is used only one time.

III. CELLULAR AUTOMATA AND CRYPTOGRAPHY

One-dimensional CA is in a simplest case a collection of two-state elementary automata arranged in a lattice of the length *N*, and locally interacted in a discrete time *t.* For each cell *i* called a central cell, a neighborhood of a radius *r* is defined, consisting of $n_i = 2r + 1$ cells, including the cell *i*. When considering a finite size of CAs a cyclic boundary condition is applied, resulting in a circle grid as shown in Figure 1.

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. $q_i^{t+1} = f (q_i^{\ t}, q_{i1}^{\ t}, q_{i2}^{\ t}, \ldots, q_{in}^{\ t})$, and a transition function *f* , called a *rule*, which defines a rule of updating a cell *i*. A

length *L* of a rule and a number of neighborhood states for a binary uniform CAs is $L = 2^n$, where $n = n_i$ is a number of cells of a given neighborhood, and a number of such rules can be expressed as 2^L. For CAs with e.g. $r = 2$ the length of a rule is equal to $L = 32$, and a number of such rules is 2^{32} and grows

very fast with *L*. When the same rule is applied to update cells of CAs, such CAs are called uniform CAs, in contrast with non uniform CAs when different rules are assigned to cells and used to update them.

Wolfram was the first to apply CAs to generate PNSs. He used uniform, 1D CAs with r = 1, and rule 30. Hortensius and Nandi et al. [20] used nonuniform CAs with two rules 90 and 150, and it was found that the quality of generated PNSs was better than the quality of the Wolfram system. Recently Tomassini and Perrenoud [12] proposed to use nonuniform, 1D CAs with r = 1 and four rules.2

In this study we continue this line of research. We will use finite, 1D, non uniform CAs. However, we extend the potential space of rules by consideration of two sizes of rule neighborhood, namely neighborhood of radius r = 1 and 2. To discover appropriate rules in this huge space of rules we will use CP.

IV. CELLULAR PROGRAMMING ENVIRONMENT

A. Cellular programming

CP is an evolutionary computation technique similar to the diffusion model of parallel genetic algorithms and introduced to discover rules for non uniform CAs. Fig.2 shows a CP system implemented [2] to discover such rules. In contrast with the CP used in [12] the system has a possibility to evaluate non uniform rules of two types. The system consists of a population of *N* rules (left) and each rule is assigned to a single cell of CAs (right). After initiating states of each cell, i.e. setting an initial configuration, the CAs start to evolve according to assigned rules during a predefined number of time steps. Each cell produces a stream of bits, creating this way a PNS.

After stopping evolving CAs all PNSs are evaluated. The entropy E_h is used to evaluate the statistical quality of each PNS. To calculate a value of the entropy each PNS is divided into subsequences of a size h. In all experiments the value $h = 4$ was used. Let l be the number of values which can take each element of a sequence (in our case of binary values of all

where P_{hj} is a measured probability of occurrence of a sequence h_j in a PNS. The entropy achieves its maximal value $E_h = h$ when the probabilities of the k_h possible sequences of the length h are equal to $1/l^h$. The entropy will be used as a fitness function of CP.

A single PNS is produced by a CA cell according to assigned rules and depends on a configuration ci of states of CAs. To evaluate statistically reliable value of the entropy, CAs run with the same set of rules C times for different configurations ci, and finally the average value of entropy is calculated and serves as a fitness function of each rule from the population of rules.

After evaluation of a fitness function of all rules of the population genetic operators of selection, crossover and mutation are locally performed on rules. The evolutionary algorithm stops after some predefined number of generations of CP.

The algorithm can be summarized in the following way:

1) Initiate randomly *population* of *N* rules of type

2) or type 2 (r = 2), or both types, and create CAs consisting of *N* cells

2. Assign *k*ith rule from the CP population to *k*ith cell of CAs

3. **for** $i = 1 \ldots C$ **do** { create randomly configuration c_i of CAs evolve CAs during M time steps evaluate entropy of each PNS }

4. Evaluate fitness function of each rule

5. Apply locally to rules in a specified sequence genetic operators of selection, cross-over and mutation

6. If STOP condition is not satisfied return to 2.

B. Discovery of rules in 1D, non uniform CAs

In all conducted experiments a population of CP and the size of non uniform CAs were equal to 50 and the population was processing during 50 generations. The CAs with initial random configuration of states and a set of assigned rules evolved during $M = 4096$ time steps. Running CAs with a given set of rules was repeated for $C = 300$ initial configurations. Fig. 3 shows an example of running CP for the evolutionary neighborhood $i - 3$, $i - 2$, i, $i + 2$, $i + 3$. One can see that whole CAs is able to produce very good PNSs after about 40 generations (see, the average value avg of the entropy close to 4).

A typical result of a single run of an evolutionary process starting with a random rules assigned to cells of CAs is discovering by CP a small set of good rules which divide the cellular space of CAs into domains-areas where the same rules, short ($r = 1$) or long ($r = 2$), live together (see Table 1). Evolutionary process is continued on borders of domains

where different rules live. This process may result in increasing domains of rules which are only slightly better than neighboring rules, which domains will decrease and finally disappear .

This happens in particular when two neighboring domains are occupied respectively by the same short rules and the same long rules. The search space of short rules is much smaller than the search space of the long rules. Therefore better short rules are discovered faster than better long rules, and for this reason long rules are gradually replaced by short rules. To limit this premature convergence of short rules, the short and long rules are initially randomly assigned to cells in the proportion of 1:3 in all subsequent experiments.

The purpose of the experiments which followed was to discover an enlarged set of rules (to enlarge the key space of cryptography system) which working collectively would produce very high quality PNSs. It was noticed that in a single run of CP the evolutionary algorithm produces typically a small set of rules with a very high value of the entropy. In the result of evolutionary searching process a set of 8 short rules (including 5 rules found by [16]) and a set of 39 long rules was found.

given test. For this purpose uniform CAs consisting of 50 cells evolved during 65536 time steps with each single discovered rule. Each PNS produced by CAs was divided into 4-bit words and tested on general statistical tests such as the entropy, v2 test, serial correlation test [6] (some weaker rules after this testing were removed).

The best scores were achieved by rules 30, 86, 101, 153 and by 8 long rules. Rules 90,105,150 and 65 working separately in uniform CA obtained good results in test of entropy and long runs test, quite good results in serial correlation test and monobit test but were week in X2 test ,poker test, runs test.sult weak in v2 test, poker test and runs test. However this set of rules work-ing collectively in non uniform CAs achieves good results (see, Table 2). For this reason only 10 rules were removed from discovered set of rules which have passed the FIPS 140-2 standard testing. These rules were worse than Tomassini and Perrenoud rules. However passing all statistical tests does not exclude a possibility that the PNS is not suitable for cryptographic purposes. Before a PNS is accepted it should pass special cryptographic tests. Therefore rules which passed tests were next submitted to a set of Marsaglia tests [7]—a set of 23 very strong tests of randomness implemented in the Diehard program. Only 11rules passed all 23 Marsaglia tests. These are short rules 30, 86, 101, and long rules 869020563, 1047380370, 1436194405, 1436965290, 1705400746, 1815843780, 2084275140 and 2592765285.

The purpose of the last set of experiments was a selection of a small set of short and long rules for non uniform CAs which working collectively would provide a generation of very high quality PNSs suitable for the secret key cryptography. Simple combination of different rules which passed all Marsaglia tests in non uniform CAs have shown that resulting PNSs may have worse statistical characteristic than PNSs obtained using uniform CAs. On the other hand, experiments with Tomassini and Perrenoud rules show that rules that separately are working worse can provide better quality working collectively. For these reasons rules 153 and some long rules which obtained very good results in general tests but not passed all Marsaglia tests were also accepted for the set of rules to search a final set of rules. In the result of combining rules into sets of rules and testing collective behavior of these sets working in no non uniform CAs the following set of rules has been selected: 86, 90, 101, 105, 150, 153, 165 ($r = 1$), and 1436965290 ($r = 2$). Among the rules are 4 rules discovered in [16]. The set of found rules have been tested again on statistical and cryptographic tests using non uniform CAs with random assignment of rules to CA cells. Table II presents the results of testing this new set of rules and compares the results with ones obtained for Tomassini and Perrenoud rules. One can see that results of testing both sets on general tests and FIPS 140-2 tests are similar. However, the main difference between these results can be observed in passing Marsaglia test.

The secret key K which should be exchanged between two users of considered CA- based cryptosystem consists of a pair of randomly created vectors: the vector R_i informing about assigning 8 rules to N cells of CAs and the vector C(0) describing an initial binary state of CA cells. The whole key space has therefore the size 8^N x 2^N. The key space is much larger than the key space (4^N x 2^N) of 1D CA-based system

V. CONCLUSION

CA are an attractive approach for cryptographic applications. They are simple, modular logic systems that can generate good quality pseudorandom bit streams as required in robust cryptographic systems. In the paper we have reported results of the study on applying CAs to the secret key cryptography. The purpose of the study was to discover a set of CA rules which produce PNSs of a very high statistical quality for a CA-based cryptosystem which is resistant on breaking a cryptography key. The main assumption of our approach was to consider non uniform 1D CAs operating with two types of rules. Evolutionary approach called CP was used to discover suitable rules. After discovery of a set of rules they were carefully selected using a number of strong statistical and cryptographic tests. Finally, the set consisting of 8 rules has been selected. Results of experiments have shown that discovered rules working collectively are able to produce PNSs of a very high quality outperforming the quality of known 1D CA-based secret key cryptosystems, which also are much more resistant for breaking cryptography keys that know.

REFERENCES

[1] A. Menezes, P. van Oorschot, S. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996

[2] A. Mroczkowski, Application of cellular automata in cryptography, Master Thesis, Warsaw University of Technology, 2002 (in Polish).

[3] A.K Das, A. Sanyal, and P.P. Chaudhuri, "On the Characterization of Cellular Automata, " Information Science, 1991.

[4] A.K Das and P.P Chaudhuri, "Efficient Characterization of Cellular Automata", Proc. IEE(Part E), IEE, Stevenge, U.K., Vol. 137, Jan. 1990, pp. 81-87.

[5] B. Schneier, Applied Cryptography, Wiley, New York, 1996.

[6] D.E. Knuth, The Art of Computer Programming, in: Seminumerical Algorithms, vols 1 and 2,Addison-Wesley, 1981.

[7] G Marsaglia, Diehard. Available from http://stat.fsu.edu/~geo/ diehard.html (1998).

[8] H. Gutowitz, Cryptography with dynamical systems, in: E. Goles, N. Boccara (Eds.), Cellular Automata and Cooperative Phenomena, Kluwer, 1993. [16].

[9] H. Nishio. "Real Time Sorting of Binary Numbers by One-dimensional Cellular Automata", Technical Report, Kyoto Univ., Japan, 1981.

[10] J. Kari, Cryptosystems based on reversible cellular automata, Personal communication, 1992.

[11] M. Mitchell, An Introduction to Genetic Algorithms (Complex Adaptive Systems), MIT Press, ISBN:0262133164.

[12] M. Sipper, M. Tomassini, Generating parallel random number generators by cellular programming, International Journal of Modern Physics C 7 (2) (1996) 181-190.

[13] M. Tomassini, M. Perrenoud, Stream ciphers with one- and two-dimensional cellular automata, in: M. Schoenauer et al. (Eds.), Parallel Problem Solving from Nature--PPSN VI, LNCS 1917, Springer, 2000, pp. 722-731.

[14] P. Sarkar, A brief history of cellular automata, ACM Computing Surveys 32 (1) (2000) 80-107.

[15] P. Guan, Cellular automaton public-key cryptosystem, Complex Systems 1 (1987) 51-56.

[16] P. Pal Chaudhuri, D. Roy Chowdhury, S. Nandi, and S. Chatterjee, "Additive Cellular Automata – Theory and Applications", volume 1. IEEE Computer society Press, CA, USA, ISBN 0-8186-7717-1,1997.

[16] R. Sommerhalder and S.C van Westrhenen, "Parallel Language Recognition in Constant Time by Cellular Automata", Acta Informatica, Vol. 6, 1983, ppl 397-407.

[17] S. Wolfram, Cryptography with cellular automata, in: Advances in Cryptology: Crypto'85 Proceedings, LNCS 218, Springer,

[18] S. Wolfram. "Statistical Mechanics of Cellular Automata". Rev. Mod. Phys., Vol. 55, July 1983, pp. 601–644.

[19] S. Chakraborty, D.R. Chowdhury, and P.P. Chaudhuri, "Theory and Application of Non-Group Cellular Automata for Synthesis of Easily Testable Finite State Machines", IEEE Trans. Computers, Vol. 45, No.7, July 1996, pp. 769-781.

[20] S. Nandi, B.K. Kar, P.P. Chaudhuri, Theory and applications of cellular automata in cryptography, IEEE Transactions on Computers 43 (1994) 1346-1357.

[21] Lalith, T. (2010). Key Management Techniques for Controlling the Distribution and Update of Cryptographic keys. International Journal of Advanced Computer Science and Applications - IJACSA, 1(6), 163-166.

[22] Hajami, A., & Elkoutbi, M. (2010). A Council-based Distributed Key Management Scheme for MANETs. International Journal of Advanced Computer Science and Applications - IJACSA, 1(3).

[23] Meshram, C. (2010). Modified ID-Based Public key Cryptosystem using Double Discrete Logarithm Problem. International Journal of Advanced Computer Science and Applications - IJACSA, 1(6).