Available online at http://www.journalcra.com

International Journal of Current Research
Vol. 3, Issue, 9, pp.086-089, August, 2011

INTERNATIONAL JOURNAL OF CURRENT RESEARCH

ISSN: 0975-833X

RESEARCH ARTICLE

CIPHER SCHEME HYBRID ADDITIVE CELLULAR AUTOMATA

Meenakshidevi, M*

Department of Computer Science and Engineering, Alagappa University, Karaikudi, Tamilnadu, India

ARTICLE INFO

Article History:
Received 19th May, 2011
Received in revised form
8th June, 2011
Accepted 5th July, 2011
Published online 23rd August, 2011

Key words:
Cryptography,
Cellular Automata,
Block cipher.

ABSTRACT

This paper presents an originally encryption system implemented on a structure of hybrid additive programmable cellular automata (HAPCA). As the development of cellular automata (CA) applications is generally an experimental effort, the research implies the exploration through simulation of the huge space of cellular automata local rules and global states. The encryption and decryption modules are identically and the cryptosystem is featured by its large key space and high speed due to cellular automata's parallel information processing. The method supports both software and hardware implementation. In this paper we present a fully functional software application for the data encryption of yahoo messenger conversations.

INTRODUCTION

In this age of information, communications and electronic connectivity, security is a topic of general interest that should never be underestimated. The security of databases, of data communications, of Internet connections, of scientific research and of personal e-mail and phone calls are examples for cases in which the encryption of data/information plays a major role. Therefore, cryptography has become an important field of theoretical research and applications development, not only in military communications as it was at its origins, but also for the business sector and private individuals. Because of its importance, cryptography is nowadays a science by itself, strongly related to other modern research fields as complexity theory, chaos, dynamical systems, computing theory and so on. The state of the art for the field of cryptography is probably classified as it has military applications, but for the public domain a good reference can be found in [1], [2] and [3].

1.1 Cryptographic Techniques

Cryptographic techniques are divided into two categories: symmetric-key (or secret-key) and asymmetric- key (or public-key).They differ each other as follows: the symmetric algorithms use the same key for both encryption and decryption (or the decryption key is easily derived from the encryption key), whereas asymmetric algorithms use a different key for encryption and decryption, and the decryption key cannot be derived from the encryption key without knowing some additional secret information. Based on these algorithms there are two classes of symmetric-key encryption schemes: block ciphers and stream ciphers. Block ciphers breaks up the message into blocks of the fixed length and encrypt one block at a time. On the contrary, the stream ciphers encrypt a single bit of plain text at a time. This paper deals with symmetric-key block encryption. CA have been used so far in both symmetric-key and public-key cryptography. Our goal is to develop an alternative cryptogram based on HAPCA, in which several CA technologies such as Wolfram approach [4], transform-based approach [5] and five evolution rules are combined in some way to form a cryptogram. A relevant relationship between the cellular automata and cryptography was revealed by Shanon in his fundamental early work [6].

2. PROGRAMMABLE CELLULAR AUTOMATA

CA is a bio-inspired paradigm highly addressing the soft computing and hardware for a large class of applications including information security.From the days of Von Neumann and Stanislav Ulam who firstly proposed the concept of cellular automata (CA) –cellular spaces, to the recent book of Stephen Wolfram "A New Kind of Science" [4], the simple structure of CA has attracted researchers from different field of interests. CA consists of a regular uniform n-dimensional lattice(or array). At each site of the lattice (cell), a physical cell quantity takes values. This physical quantity is the global state of the CA, and the value of this quantity at each is the local state of this cell. Each cell of the CA is restricted to local neighborhood interactions only, and as a result it is incapable of immediate global communication. The neighborhood of the cell is taken to be the cell itself and some or all of the immediately adjacent cells. The cells evolve in discrete time steps according to some deterministic rule that depends only on local neighbours. In effect, each cell consists

***Corresponding author:** srimeenuwonder@gmail.com

of a storage element (D flip-flop) and a combinational logic (CL) implemented the next-state functions (see Fig. 1). The combinational logic is called the "rule" of the CA.

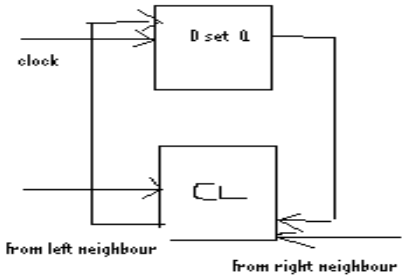

Fig 1: CA cell

The next-state function describing a rule for a three neighborhood CA cell can be expressed as follows:

$$a_i(t+1) = f[a_i(t), a_{i+1}(t), a_{i-1}(t)] \qquad (1)$$

Where i is the position of an individual cell in one-dimensional array of cells, t is the time step, and f is the rule of CA. If the rule of a CA involves only XOR logic, then it is called a linear rule. Rules involving XNOR logic are referred to as complement rules. A segments. As results, the operation of CA can be represented by a state-transition graph. Figure 2 shows the state transition graph of a null boundary condition CA.

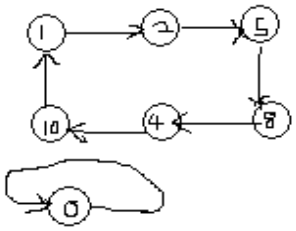

Fig 2: The state transition graph

As the state space is discrete, on a finite lattice of any given size the number of possible configurations is necessarily finite. This means that, in principle at least, all possible configurations of the system can be enumerated and the step-by-step evolution can be represented by connecting those points with directed. Each node of the graph represents one of the possible states of the CA. The direct edges of the graph correspond to a single time step transition of the automata. Depending on the initial state, the CA can follow a certain trajectory inside the states space and can enter in different attractor basins. The PCA was firstly introduced in [8], where the CL of each cell is not fixed but controlled by a number of control signals such that different functions (i.e. rules) can be realized on the same structure. As a matter of fact, PCA are essentially a modified CA structure. It employs some control signals on a CA structure. By specifying certain values of control signals at run time, a PCA can implement various functions dynamically in terms of different rules.

3. PCA ENCRYPTION ALGORITHM

CA with all its cells having linear rules is called a linear CA, whereas a CA having a combination of XOR and XNOR rules is called additive CA. If all the cells obey the same rule, then the CA is said to be a uniform CA, otherwise, it is a hybrid CA. A CA is said to be a null boundary CA if both the left and right neighbor of the leftmost and rightmost terminal cell is connected to logic 0- state. In practice, a control program, stored in memory, can be employed to activate the switches. For example, the state 1 or 0 of the bit i-th of a memory word either opens or closes the switch that controls that cell. Basically, such a structure is referred as a programmable CA (PCA). The encryption method proposed here is based on the fact that the CAs from class II exhibit periodic behavior (i.e. each state lays in some cycle). In these cases, their evolution depends essentially of the initial state, but we can say that after a while the initial state is "forgotten", in sense that the initial state cannot be retrievable through analyses of the current configuration. The proposed encryption system it is implemented using a combination of three cellular automata: a CA-PRNG and two one-dimensional PCA. The structure of the proposed encryption system is presented in Fig.3

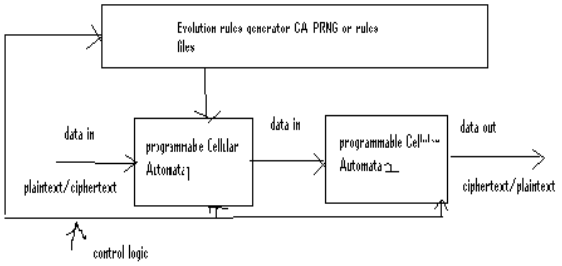

Fig 3: Structure of the proposed encryption system

In the block cipher scheme, one 8-bit message block is enciphered by one enciphering function. The PCA control signals are activated with the help of the signals that are generated with the CA-PRNG or rules file. For the sake of simplicity, the enciphering function has two fundamental transformations FTs (PCA = 2) to operate on 8-bit data. For high security applications, more FTs are preferred. The block cipher (decipher) procedure can be defined as follows:

1. Load the PCA with one byte plaintext (cipher text) from I/O. The initial block of the message is the initial state of the PCA. The global configuration of the PCA represents the encrypted message.
2. Load a rule configuration control word from CA-PRNG into the PCA.
3. Run the PCA for 1 ... 7
4. Repeat steps 2 and 3 for 2 times.
5. Send one byte ciphertext (plaintext) to I/O. If not end of the plaintext (ciphertext) go to step 1. Otherwise, stop the process.

3.1. EIGHT CELL PCA (PCA PIPELINE)

The block cipher algorithm presented in this paper is constructing using 2 PCA from class II, with rules 51, 60,102.

Table I. Rules that update the next state of the cells of the ca-prng

0e	6e	8e	0e	0	13e	173	19c	1c7	0f0
1c	70	9c	0e1	0f1	117	13f	174	19d	1ce
1d	71	9d	0e2	0f2	11c	147	19e	1cf	175
1e	72	9e	0e3	0f3	11d	14e	176	19f	1d7
2e	73	0ae	0e4	0f4	11e	14f	177	1a7	1dc

Two 8-cell PCAs are cascaded to form a CA pipeline. With the pipeline, two CA FTs can be performed simultaneously. That means one enciphering function can be done in a single pipeline. The PCA presented here was obtained after a lot of simulations with different structural and behavioral parameters. The rules specify the evolution of the CA from neighborhoods configuration to the next state and these are presented in Table I.

The corresponding combinational logic of rules 51, 60,102 for PCA can be expressed as follows

$$a_i(t+1) = a_i(t) - ® 51 \quad (2)$$

$$a_i(t+1) = a_i(t) \Leftrightarrow a_i - 1(t) - ® 60 \quad (3)$$

$$a_i(t+1) = a_i(t) \Leftrightarrow a_i + 1(t) - ® 102 \quad (4)$$

where i is the position of an individual cell in one-dimensional array of cells, t is the time step, N is the length of the CA. The PCA is a null boundary CA configured with the rules 51, 60 and 102 and its state-transition diagram consists of equal circles of even length. The null boundary 8-cell CA with rule configuration <51, 102, 102, 102, 51, 102, 51 and 102> generates cycles depicted in Fig. 4.

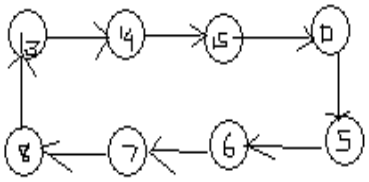

Fig4. The state transitions diagram of a non-maximum-length PCA

In the state-transition graph, the CA has two equal length cycles; each cycle has a cycle length 8. This phenomenon is a basic requirement of the enciphering scheme. For example, if we employ this CA as our enciphering function and define a plaintext as its original state, it goes to its intermediate state after four cycles. This is the process of enciphering. After running another four cycles, the intermediate state returns back to its original state, so the cipher text is deciphered into plaintext. This is the process of deciphering. An 8-cell CA configured with rules 51, 60 (or 102) has 512 kinds of configurations, but only 156 of them have cycle length 8. The others have cycles of 2, 4, 16 or combination of them. In this encryption algorithm we use only the configurations of the rules that generate cycles of length 8.So the system designer is free to take any number in the 156 combinations to enhance the security of the system. Because of the fact that the PCA does not generate sequences of maximum-length for all the possible combinations (512) of the rules we must apply from the file or CA-PRNG only the combinations (156) that generate cycles of length 8. The rules with 8-cycle length can be seen in Table II. It consists of a D flip-flop and a logic

Table II. Rules with 8 cycles length in hexadecimal

Rules	7 111	6 110	5 110	4 100	3 011	2 010	1 001	0 000
51	0	0	1	1	0	0	1	1
60	0	0	1	1	1	1	0	0
102	0	1	1	0	0	1	1	0
	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

combinational circuit (LCC). The LCC includes multiplexers and XNOR logic gates to implement the rules of CA and to control the loading of data and operation of the CA. When the load control signal (LoadData) is "logic 1", data is loaded into D flip-flop. When LoadData is "logic 0", data is run into the cell according to the rules applied to the rule control signals (S1, S0) and the states of neighbourhoods. After an established number of cycles (1 to 7), the data on the Q output of the flip-flop is sent out and new data is loaded in. In this paper we decided to connect together sixteen cells in order to build two 8-cells PCA. In the 8-cell PCA, the data path is 8 bits (Data_In[0:7]), there are 9 rules configuration signals and one load data control signal (LoadData). Because all the cells share the same common rule signal S1, only rule 51 or one of the rules 60 or 102 can be applied at a given time. The left and right input terminals of every cell are connected to left and right neighbours DataOut terminals, thus it is configured as a 3-neighbourhood CA. The left and right terminals of the leftmost and rightmost cells are connected to "logic 0"providing a null boundary condition.

3.2. RULES GENERATOR (CA-PRNG OR RULES FILE)

The rule generator is designed to provide encryption rules for the encryption system. The rule generator consists of a modulus 156 counter and a file with 156 lines with rules or a CA used as a key stream generator. We can also use a CA as a key stream generator, CA pseudo-random number generator (PRNG) that combines in some way two rules (the rules 90 and 150), to provide the key sequence [9]. It has established that the maximum-length CAs generates patterns having a high quality of pseudo-randomness. The corresponding combinational logic of rules 90, 150 for CA can be expressed as follows:

$$\mathbf{a_i(t+1) = a_i - 1(t) \Leftrightarrow a_i + 1(t) - ® 90} \quad \mathbf{(5)}$$

$$\mathbf{a_i(t+1) = a_i - 1(t) \Leftrightarrow a_i(t) \Leftrightarrow a_i + 1(t) - ® 150} \quad \mathbf{(6)}$$

The evolution rules (keys of the block cipher) for the two PCA are selected from a file rules or from this CA. The operation of CA can be represented by a state-transition graph. Each node of the transition graph represents one of the possible states of the CA. The direct edges of the graph correspond to a single time step transition of the automata.The outputs of the CA-PRNG are connected to the selection signals (S1, S0) of the two PCA. If the rules are read from the file, first are loaded into a list structure. When the encryption process begins, rules are read out in sequence and sent to the PCA arrays.

CONCLUSIONS

This paper presents a methodology for the development of particular crypto scheme with PCA. The encryption technique presented in this paper demonstrates the power of the new block cipher cryptosystem based on programmable cellular automata. The cryptogram was tested and verified using Yahoo messenger conversations and a primary encouraging result was the perfect concordance between encryption and decryption presented in software simulation. In the PCA encryption algorithm, the same cipher-text may be generated from different plain-text, and any cipher-text may give rise as well to different plain-text depending on the different PCA's rule configuration.

REFERENCES

[1] William Stallings, "Cryptography and Network Security", Prentice- Hall, New Jersey, 2003.

[2] Brian A. LaMacchia, ".NET Framework Security", Addison Wesley, USA, 2002.

[3] M. Seredinsky and P. Bouvry, "Block encryption using reversible cellular automata", ACRI 2004 The Netherlands - Amsterdam, LNCS 3305, pp. 785–792, October 2004.

[4] S. Wolfram, "A new kind of science", Wolfram Media Inc., 2002.

[5] O. Lafe, "Cellular automata transforms: theory and applications in multimedia compression, encrypt and modeling", Kluwer Academic Publisher, 2000.

[6] C. Shannon, "Communication Theory of Secrecy Systems", Bell Sys.Tech.J.28,pag.656–715,1949. (netlab.cs.ucla.edu/wiki/files/shannon1949.pdf).

[7] Franciszek Seredynski, Pascal Bouvry, Albert Y. Zomaya, "CellularProgramming and Symmetric Key Cryptography Systems", in GECCO 03 (Genetic and Evolutionary Computation Conference), Chicago, IL, USA, July 12-16, 2003. Proceedings, Part II. Lecture Notes in Computer Science 2724 Springer 2003, ISBN 3-540-40603-4, pp. 1369-1381, 2003.

[8] S. Nandi, B. K. Kar and P. Pal Chaudhuri, "Theory and applications of cellular automata in cryptography", IEEE Transactions on Computers, 43(12):1346-1356, 1994.

[9] P.D. Hortensius, R.D. McLeod, Podaima, "Cellular Automata Circuits for Built-in Self-test", IBM J. RES. DEVELOP., Vol. 34,No.2/3, 1990, 389-405.

****