Trends in Applied Sciences Research

ISSN 1819-3579

Academic
Journals Inc.

www.academicjournals.com

Trends in Applied Sciences Research 7 (1): 68-77, 2012
ISSN 1819-3579 / DOI: 10.3923/tasr.2012.68.77

Two-layer Cellular Automata Based Cryptography

[1]Alireza Jaberi, [2]Ramin Ayanzadeh and [2]Azam S. Zavar Mousavi
[1]Department of Mathematic, Islamshahr Branch, Islamic Azad University, Islamshahr, Iran
[2]Department of Computer, Bojnourd Branch, Islamic Azad University, Bojnourd, Iran

Corresponding Author: Ramin Ayanzadeh, Department of Computer, Bojnourd Branch, Islamic Azad University, Bojnourd, Iran

ABSTRACT

Cryptography is knowledge of manipulating data to conceal secure information. It serves an essential functionality in wide variety of applications. So, several encoding and decoding methods have been proposed to enhance cryptography techniques. In this study a novel approach based on multi layer cellular automata is proposed to be used in cryptography applications. Proposed multi layers cellular automata employs interaction between two heterogeneous cellular automata to imitate Pseudo-Neumann neighborhood structure and generate trackable random integers. These random numbers are assumed as time variant keys for encoding and decoding purposes. To verify and validate performance of proposed architecture, several simulations are performed. Simulation results prove that two-layer cellular automata generate more uniform random numbers in comparison with MATLAB. Consequently, proposed architecture demonstrated desirable behavior and has less risk. Furthermore, the architecture is suitable for hardware implementations.

Key words: Cellular automata, cryptography, encryption, information technology, random numbers generator, random variables, security, time variant public key

INTRODUCTION

Cryptography is a method of concealing information in codes. It first began in 1900 B.C. when an Egyptian were using images to show the words (Rabah, 2004). Nowadays, some types of information such as credit cards, bank accounts and marshal projects are known as secure date so making them inaccessible to unauthorized people is a big challenge. Therefore, cryptography has been paid attentions as a critical paradigm. Generally, objective of cryptography is to manipulate context of messages in order to hide information (Forouzan, 2008; Rabah, 2006; Schneier, 1995). In fact, general topics of cryptography are:

- **Privacy:** Unauthorized people are not allowed to access context of information during data transformation
- **Context trustiness:** Making sure that context of messages are kept safe during cryptography phases (encoding and decoding)
- **Authentification:** Authorizing message senders
- **Deny avoidance:** Message senders could not deny what ever they have sent

Most cryptography algorithms are similar to each other. In better words, cryptography mechanisms are based on a common framework. In fact, cryptography approaches are sorted out in three major categories: namely, key free, key based and digital sign (Forouzan, 2008).

Cryptography with private key uses unique key for both encoding and decoding applications. Thus, encoding/decoding key must be shared among users. Consequently, secure channels and additional encryption techniques are needed for transferring these private keys. In current generation of cryptography with private key, there is an arithmetic relationship between encoding and decoding keys. In other view, encoding and decoding phases are performed in reverse manner. Desirable time order is the most noticeable advantage of cryptography with private key (Forouzan, 2008).

On the other hand, cryptography with public key employs two different public and private keys, for encoding and decoding. In fact, public key is shared among all authorized users and transferred with encoded message. However, private key is personal for each user and not transferred between users (Forouzan, 2008).

Furthermore, encoding and decoding keys could not be same. For instance, those messages that are encoded with public key must be decoded with recipient's private key. In Symmetric cryptography, public and private keys are retrievable based on an arithmetic relationship but asymmetric cryptography applies independent public and private keys (Abomhara *et al.*, 2010; Alsaade, 2010). Asymmetric cryptography could be performed in two different ways (Forouzan, 2008). Firstly, primary message is encoded and decoded with public and private keys, respectively. Obviously, authentifying sender is not secure because, public key is shared among all users. Secondly, primary message is encoded with private key and decoded with public key (Schneier, 1995). Despite the fact that second approach is more secure, it needs more time to be accomplished.

Digital sign is a novel method for cryptography in which, primary messages are manipulated with hash functions before encoding phase (Schneier, 1995). According to asymmetric cryptography techniques, manipulated messages are encoded and decoded with public and private keys, respectively. Consequently, message senders are detectable in digital sign (Forouzan, 2008).

Most of the traditional cryptography methods are mathematical and have their own advantages and drawbacks (Olorunfemi *et al.*, 2007; Zaidan *et al.*, 2010). During last three decades, several efforts have been performed to overcome limitations of traditional methods. Applying hardware devices in cryptography is another approach that tries to enhance reliability (Murphy *et al.*, 2006). Employing biometric features, such as finger print and iris, is other applicable technique (Islam *et al.*, 2008). However, hardware based cryptography may not be economical.

Cellular automata are discrete mathematical models which have served essential functionality in cryptography. In fact, cellular automata are applied as random variables which generate random numbers to be used as time variant private or public keys (Benkiniouar and Benmohamed, 2004). These time variant keys are employed for encoding and decoding purposes. Therefore, cellular automata based cryptography might be more complex and secured in comparison with traditional methods (Szaban *et al.*, 2005).

Cellular automata are also employed for cryptography in image processing applications (Luo *et al.*, 2010). In these applications, source images are assumed as initial configuration of cellular automata. Thus, after some fixed iterations source image will be manipulated in suitable manner. Consequently, applied transition rules must be reversible to retrieve primary images from encoded images (Islam *et al.*, 2008).

Considering that cellular automata have served desirable functionality in cryptography, some hardware implementations, based on VLSI technology, have been performed. Despite hardware implementations are more secure, they face several limitations. Specifically, VLSI applications may not be economical (Khalil-Hani *et al.*, 2008; Meng, 2011; Murphy *et al.*, 2006).

In this study, a novel two-layer cellular automata is proposed for cryptography. Proposed method uses interaction between two heterogeneous cellular automata to emulate Pseudo-Neumann neighborhood structure. Outputs of two-layer cellular automata are assumed as time variant keys that are used in cryptography applications. Proposed method can perform outstanding functionality in developing cryptography technologies.

Random numbers generators: In 1927, Tippet designed table of forty thousand random numbers to use in various applications. One hundred thousand of random numbers were generated in a table designed by Kendall in 1939. Smith followed Kendall's work and designed mechanical random generator device in 1955. The exciting point about these tables is that they were filled without any specific algorithm. In 1951, Neumann proposed a computational method (however, this method had low performance). In recent decades several algorithms were developed for random number generation as below (Ayanzadeh *et al.*, 2010; Banks *et al.*, 2004; Moghaddas *et al.*, 2008; Viega, 2003).

Linear congruential generators: Linear congruential methods use specific algorithm to generate random numbers. These algorithms are iterative and initial state is needed to start the algorithm. A sample of these algorithms is indicated in Eq. 1:

$$X_n = (aX_{n-1} + c) \quad \text{mod} \quad m \tag{1}$$

where, x_{n-i} is the generated random number in previous iteration, a and c are constant coefficients, m is Congruential module (one unit more than maximum allowed random number) and x_n is the output of algorithm. In this method generated random number extremely depends on its previous value. Maximum period of this algorithm is m (Banks *et al.*, 2004; Moghaddas *et al.*, 2008).

Multiple recursive generator: Multiple recursive generators are like linear congruential generator but this method use k random numbers from previous iterations. A multiple recursive generator is indicated in Eq. 2.

In Eq. 2, a_i are constant coefficients of algorithm, x_{n-i} is output of algorithm in (n-i)th iteration and m is congruential module (one unit more than maximum allowed random number). The advantage of this method is that maximum period of algorithm is 2^m which is much more than period of Linear congruential method (Banks *et al.*, 2004; Brent, 1994; Moghaddas *et al.*, 2008):

$$X_n = \sum_{i=1}^{k} a_i X_{n-i} \quad \text{mod} \quad m, \quad i = 1, 2, ..., k \tag{2}$$

Lagged fibonacci generator: Lagged Fibonacci generator is a special case of famous Fibonacci sequence which uses two outputs of previous iterations. Equation 3 indicates the general form of this method:

$$x_n = (x_{n-l} + x_{n-k}) \text{ mod m}, \; o < k < l \tag{3}$$

where, m and x_{n-i} are same with these parameters in multiple recursive generator method. k and l are the indexes of numbers which were generated in previous iterations. Performance of algorithm depends on selection of these values.

Summation operator in Eq. 3 can be changed by any other operator (e.g., subtraction operator). Moreover, it is possible to use binary logic operators to generate random bits. In this case if the operator is exclusive or (XOR) the method will be called transfer register generator thus the congruential operator will be neglected from equation and Eq. 3 will be changed to Eq. 4:

$$x_{n+1} = x_{n-p} \oplus x_{n-q} \tag{4}$$

Output of Eq. 1 up to 3 will be random numbers between zero and m (Banks *et al.*, 2004; Brent, 1994).

Blum blum shub random generator: This generator was introduced by Blum and his team in 1986 but due to slow functionality of this method it was never used in computer simulations. This method is widely used in cryptography. By using this method, random numbers will be generated via Eq. 5:

$$X_{n-1} = (X_x)^2 \bmod m \tag{5}$$

where, m is congruential module and usually is considered as production of two big prime numbers (Banks *et al.*, 2004; Brent, 1994).

Cellular automata: Cellular Automata (CA) are discrete computational models that contain networks of completely same cells which have interaction with together within a neighborhood structure. Various neighborhood structures are proposed till now. Some of the most popular models are: Neumann, Moore, Cole and Smith which are illustrated in Fig. 1. In Fig. 1a-d, it is assumed that neighborhood radius equals one (Ayanzadeh *et al.*, 2009; Bar-Yam, 1997; Jaberi *et al.*, 2011; Sarkar, 2000; Schifi, 2008).

Pseudo-Neumann is another neighborhood strategy which has dynamic behavior. Although, it is based on Moore model but its behavior is similar to Neumann strategy. In this novel neighborhood strategy, a random variable is used for each neighbor cell to determine active neighbor cell. In other words, number of neighbors and their positions in Pseudo-Neumann neighborhood strategy are not fixed. Figure 2a-d illustrate some instances of Pseudo-Neumann neighborhood strategy (Ayanzadeh *et al.*, 2009, 2010).

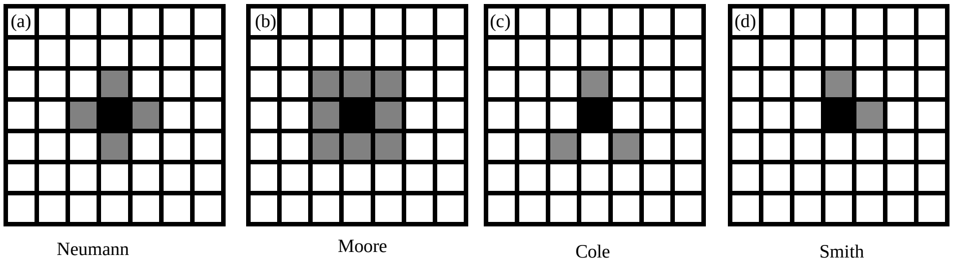


Fig. 1(a-d): Common neighborhood strategies in cellular automata

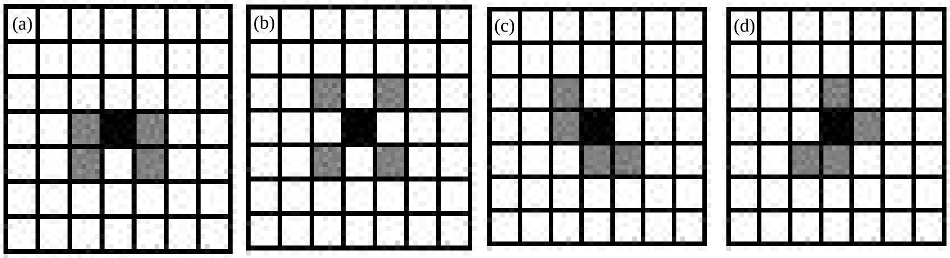


Fig. 2(a-d): Instances for Pseudo-Neumann neighborhood strategy

Table 1: Transition rules in binary CA

Rule	0	1	10	11	100	101	110	111
30	0	1	1	1	1	0	0	0
90	0	1	0	1	1	0	1	0
105	1	0	0	1	0	1	1	0
150	0	1	1	0	1	0	0	1
165	1	0	1	0	0	1	0	1

Cells state (value) is selected from a finite set. These values are changing synchronously in iterations by using some of transition rules which are same for all cells. Next states will be determined according to current values of cells and current values of neighbors (Bar-Yam, 1997; Sarkar, 2000).

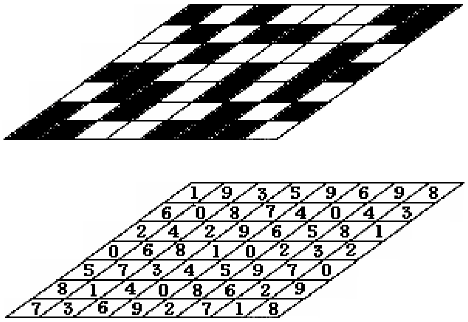
Some of the most useful Wolfram transition rules in linear binary cellular automata are illustrated in Table 1 where first row is current states of left neighbor, the cell and right neighbor, respectively. Next state of cell is indicated in other rows by using of specified rules. Using transition rules in Table 1 and starting from a random configuration leads to generate pseudo random bits. Locality of rules leads to generate pseudo random bits with desirable period (Sarkar, 2000; Schifi, 2008; Wolfram, 1986).

Two-layer cellular automata for cryptography: Proposed cellular automata are constructed from two heterogeneous layers of cellular automata. Each layer contains two dimensional cellular automata with same size.

Cells of first layer are binary and include zero or one bits. On the other hand, if the objective is generating uniform random numbers in range [0, n], as time variant keys for cryptography, then cells of second layer will include integer numbers between zero and n.

Upper bound parameter n is specified according to the coding system. For example, to apply proposed cellular automata for encoding messages with ASCII text code cells of second layer cellular automata will be in range of 0 to 255. Structure of proposed model is illustrated in Fig. 3.

Each row of binary cellular automata, in first layer, is assumed as independent linear binary cellular automata. Thus, each cell is adjacent with one right neighbor cell and one left neighbor cell. Associated values of cells of each row will be updated using one of the Wolfram transition rules 30, 90, 105, 110 or 165.



1	9	3	5	9	6	9	8
6	0	8	7	4	0	4	3
2	4	2	9	6	5	8	1
0	6	8	1	0	2	3	2
5	7	3	4	5	9	7	0
8	1	4	0	8	6	2	9
7	3	6	9	2	7	1	8

Fig. 3: Two layers CA for uniform random number generation

Binary cellular automata perform random variable functionality for second layer cellular automata. Thus, Pseudo-Neumann neighbourhood structure is employed in second layer. Despite the fact that Pseudo-Neumann neighbourhood structure demonstrate in more complex manner, it is also be trackable.

According to Pseudo-Neumann neighbourhood structure cells of second layer are active if value of cells with same positions in first layer are one and they are inactive if value of cells with same positions in first layer are zero. They will be assumed active, otherwise they will be inactive. If the cells of first layer generate uniform random bits, the cells of second layer will activate/inactive with same probability. In this case about half of the cells in Moree neighbourhood structure, four cells like in Neurnan structure, will be active. In other view, interaction between first and second layers of cellular automata leads Pseudo-Neumann neighbourhood structure to be assumed as Neumann neighborhood structure with dynamic adjacency.

States of each cell in second layer will be updated by dividing summation of cell value and values of active neighbours by n+1. Remainder of this division is considered as next value of cell. According to this rule, values of cells will be between zero and n. Furthermore, initial configuration of automata is generated randomly with uniform distribution. In addition, initial configuration must be saved for decoding purpose.

To apply proposed method, boundary values of cells in second layer are specified. In particular, lower and upper values of applied Unicode could be determined based on the languages of text messages. Moreover, a simple mapping equation can be employed in multi, language messages.

Before encoding phase, both first and second layer cellular automata must be initialized randomly. It's important to mention again that initial configuration is necessary for decoding phase too. In fact, initial configuration is used as public key in cryptography.

To encode text messages, characters of primary message are fetched sequentially. For each character or symbol, cells of both first and second layer cellular automata are updated based on their specific transition rules. Then, Unicode of fetched character is added with output of second layer cellular automata. In this case, each of the cells in second layer can be considered as output cell. Therefore, position of output cell can be participated in public key as a system parameter. After encoding phase, domain of employed Unicodes will be transferred from [a, b] to [a, b+n].

On the other hand, in decoding phase characters of encoded message are fetched sequentially. For each character, cells of both first and second layer cellular automata are updated and output of second layer is detracted from fetched Unicode. Behaviour of proposed cellular automata is deterministic. Thus, if initial configuration and transition rules of both encoder and decoder cellular automata are same then decoded message will be same with primary message.

EXPERIMENTAL RESULTS

Proposed method is based on two-layer cellular automata which applies Pseudo-Neumann neighbourhood structure. To verify and validate functionality of proposed method, some computational simulations was performed. In other words, functionality of first and second layer cellular automata in generating uniform random bits and time variant public keys for cryptography were considered. In addition, risk probability for proposed system is investigated according to the system parameters.

Functionality of first layer cellular automata: In this experiment capability of cellular automata in generating uniform random bits is considered. Therefore, binary cellular automata with 100 cells were implemented. In this simulation, neighbourhood radius is one. In addition, rule 30, in Wolfram notation, is applied for updating states of cells. Furthermore, boundary condition is considered as wrap mode. Thus, an arbitrary cell of binary cellular automata can be assumed as output of cellular automata.

To evaluate the uniformity of employing rule 30 in binary cellular automata, 103 random bits were generated and total numbers of ones which were appeared in sequence was computed. This simulation was run for one hundred times and statistical indicators such as average, standard deviation and scattering length were extracted. Statistical indicators of MATLAB software also calculated to make a benchmark for comparing uniformity of proposed binary cellular automata with random number generator of MATLAB. These statistical indicators are illustrated in Table 2.

Obviously, Table 2 indicates that quality of generated random bits using binary cellular automata is desirable. Generated random bits also follow uniform distribution. Thus, if rule 30 is used to update values of first layer in proposed model, then cells of second layer will be activated and deactivate with approximately same probability.

Functionality of second layer cellular automata: In this experiment, output of second layer cellular automata is evaluated. Objective of this evaluation is considering the uniformity of generated integer numbers which are used as time variant keys in cryptography. Therefore, sequence of integer numbers are generated by second layer of proposed model and integer random number generator of MATLAB. Then an experiment is implemented as below.

- Generate $N = 10^4$ random numbers in the range of [0,100]
- Classify the generated numbers in $c = 10$ classes with equal sizes
- Compute the frequency of numbers in each class (f_i)

After running these steps for one hundred times, average, standard deviation and scattering length of frequencies of classes are computed. Table 3 contains the experiment results. These

Table 2: Statistical features of experiment 1

Tool	Scattering length	Standard deviation	Average
Binary CA	86	10.6328	499.9484

Table 3: Statistical features of experiment 2

Method	Average	Standard deviation	Scattering length
MATLAB	926.23	111.93	504
MLCA	944.86	83.10	308

statistical indicators demonstrate generated random numbers by two-layer cellular automata are more uniform than MATLAB.

Risk of system: As discussed earlier, proposed approach is based on two-layer cellular automata with emergent behaviour. It is obvious that the system will be unsecure if and only if all of the system parameters are detected. These parameters are:

- Size of cellular automata (R×C)
- Transition rules in binary cellular automata (256 primary rules in Wolfram notation)
- Domain of values for cells in second layer (n)
- Initial configuration of both first and second layer
- Position of output cell in second layer

To make system secure, some of the parameters are transferred as public key and other parameters are assumed fixed and private. It is clear that extracting all of the parameters by unauthorized users is some how impossible. For instance, if public key consists of size of cellular automata, range of values in second layer and position of output cell, then the probability of discovering key could be calculated by Eq. 6:

$$P = \left(\frac{}{2^{RC} n^{RC}}\right)\left(\frac{1}{256}\right)\left(\frac{1}{RC}\right) \tag{6}$$

It is necessary to mention that probability of system risk is meaningful when all parts of public and private keys are detected by unauthorized users.

CONCLUSION

In this study a novel two-layer cellular automata is introduced to be used in cryptography applications. Architecture of proposed method consists of two heterogeneous two-dimension cellular automata.

First layer is binary cellular automata which specifies functionality of cells in second layer. In fact, cells of second layer are assumed active/inactive when state of cells in first layer with same position are one/zero. Therefore, second layer cellular automata employs Pseudo-Neumann neighbourhood model. Furthermore, second layer cellular automata generate time variant keys for encoding and decoding purposes. Thus, domain of states in second layer cellular automata is specified based on coding system which is used in message context.

To evaluate functionality and performance of proposed system, several computational simulations have performed. Simulation results prove that proposed method generates more uniform random integer numbers in compression with MATLAB random number generator. Consequently, output of two-layer cellular automata could be used as private key in cryptography applications. Specifically, it could serve suitable functionality in asymmetric cryptography in which, messages are encoded and decoded with private and public keys, respectively.

REFERENCES

Abomhara, M., O.O. Khalifa, O. Zakaria, A.A. Zaidan, B.B. Zaidan and H.O. Alanazi, 2010. Suitability of using symmetric key to secure multimedia data: An overview. J. Applied Sci., 10: 1656-1661.

Alsaade, F., 2010. Symmetric crypto-graphical model. Trends Applied Sci. Res., 5: 146-151.

Ayanzadeh, R., K. Hassani, Y. Moghaddas, H. Gheiby and S. Setayeshi, 2009. Innovative approach to generate uniform random numbers based on a novel cellular automata. J. Applied Sci., 9: 4071-4075.

Ayanzadeh, R., K. Hassani, Y. Moghaddas, H. Gheiby and S. Setayeshi, 2010. Multi-layer CA for normal random number generation. Proceedings of the 18th Iranian Conference on Electrical Engineering, May 11-13, Isfahan, Iran.

Banks, J., J. Carson, B.L. Nelson and D. Nicol, 2004. Discrete-Event System Simulation. 4th Edn., Prentice Hall, UK.

Bar-Yam, Y., 1997. Dynamics of Complex Systems. Addison Wesley, UK.

Benkiniouar, M. and M. Benmohamed, 2004. Cellular automata for cryptography. Proceedings of the International Conference on Information and Communication Technologies: From Theory to Applications, April 19-23, 2004, UMC., Algeria, pp: 423-424.

Brent, R.P., 1994. On the periods of generalized Fibonacci recurrences. Math. Comput. Conf., 63: 389-401.

Forouzan, B.A., 2008. Cryptography and Network Security. McGraw-Hill Inc., New York, USA.

Islam, M.R., M.S. Sayeed and A. Samraj, 2008. A secured fingerprint authentication system. J. Applied Sci., 8: 2939-2948.

Jaberi, A., R. Ayanzadeh and E. Shahamatnia, 2011. A novel fuzzy cellular automata for uniform random number generation. Proceedings of the 2nd International Conference on Contemporary Issues in Computer and Information Sciences, (CIS, 2011), Zanjan, Iran.

Khalil-Hani, M., A. Irwansyah and Y.W. Hau, 2008. A tightly coupled finite field arithmetic hardware in an FPGA-based embedded processor core for elliptic curve cryptography. Proceedings of the International Conference on Electronic Design, December 1-3, 2008, Malaysia, pp: 1-6.

Luo, H., F.X. Yu, H. Li and Z.L. Huang, 2010. Color image encryption based on secret sharing and iterations. Inform. Technol. J., 9: 446-452.

Meng, B., 2011. A survey on analysis of selected cryptographic primitives and security protocols in symbolic model and computational model. Inform. Technol. J., 10: 1068-1091.

Moghaddas, Y., R. Ayanzadeh and A.T. Hagigat, 2008. A new algorithm for improving the uniformity of random number generators based on calculation with monte carlo method. Proceedings of the of 2nd Joint Congress on Fuzzy and Intelligent Systems, (FIS, 2008), Tehran, Iran.

Murphy, G., A. Keeshan and R. Agarwal and E. Popovici, 2006. Hardware-software implementation of public-key cryptography for wireless sensor networks. Proceedings of the Irish Signals and Systems Conference, June 28-30, 2006, Ireland, pp: 463-468.

Olorunfemi, T.O.S., B.K. Alese, S.O. Falaki and O. Fajuyigbe, 2007. Implementation of elliptic curve digital signature algorithms. J. Software Eng., 1: 1-12.

Rabah, K., 2004. Steganography: The art of hiding data. Inform. Technol. J., 3: 245-269.

Rabah, K., 2006. Implementing secure RSA cryptosystems using your own cryptographic JCE provider. J. Applied Sci., 6: 482-510.

Sarkar, P., 2000. A brief history of cellular Automata. ACM Comput. Surveys (CSUR), 32: 80-107.

Schifi, J.L., 2008. Cellular Automata: A Discrete View of the World. Wiley-Interscience, USA., ISBN: 9780470168790.

Schneier, B., 1995. Applied Cryptography: Protocols, Algorithms and Source Code in C. John Wiley and Sons Inc., New York, USA.

Szaban, M., F. Seredynski and P. Bouvry, 2005. Evolving collective behavior of cellular automata for cryptography. Proceedings of the IEEE Mediterranean: Electrotechnical Conference, May 16-19, 2005, Department of Computer Science, Podlasie University, Siedlce, Malaga, pp: 799-802.

Viega, J., 2003. Practical random number generation in software. Proceedings of the 19th Annual Computer Security Applications Conference, December 8-12, 2003, USA., pp: 129-140.

Wolfram, S., 1986. Cryptography with cellular automata. Proceedings of the Advances in Cryptology, Santa Barbara, California, United States, (CRYPTO'85) Springer-Verlag, New York, USA., pp: 429-432.

Zaidan, A.A., B.B. Zaidan, A.K. Al-Frajat and H.A. Jalab, 2010. An overview: Theoretical and mathematical perspectives for advance encryption standard/rijndael. J. Applied Sci., 10: 2161-2167.