Article can be accessed online at http: //www.publishingindia.com

Hybrid Cryptosystem using Cellular Automata Transformations on Graphs

M. Phani Krishna Kishore*, Sunayana Budhiraju**

Abstract

Digital security has become a prime area of research as large amount of data is shared over the internet and variety of agents are getting connected. The need to develop systems to reduce the tradeoff between the security, space and time is never ending. Graphs are known for their versatility in applications and cellular automata for their agility to provide speed and complexity to the process. It is against this backdrop an attempt is made to demonstrate a new hybrid cryptosystem using the properties of cellular automata transformation on graphs. The method also combines public and private key encryption processes to provide flexibility and strength.

Keyword : Cellular Automata, Graphs, Cryptosystems, Information Security

Introduction

The explosive growth in the data flow over the internet in the recent past demands more and more sophisticated security systems. The variety of gadgets and systems that are getting connected to internet compels one on the need to design variety of algorithms and processes to suit the requirement, and as such a generic approach to security is no longer valid.

The tradeoff between strength of the algorithm or process and time has to be narrowed down. In this direction quest for new algorithms look a never ending process. Mathematical principles which suit these processes are continuously explored since the origin of cryptosystems. Several methods can be found in literature based on algebra, number theory, geometry, discrete dynamical system, graph theory and so on.

The public key and private key based cryptosystems are widely used in various domains. Each of them has its own strengths and weakness. Hybrid Systems are also used for greater safety. A modest attempt is made in this paper to develop a new hybrid crypto system that combines both public and private key concepts. Usually these combinations reduce the processing speed of algorithms. Cellular automata are more suitable in such cases as they offer greater complexity with less processing time. Properties of graphs needs to be explored more in this direction as they offer sound mathematical models to deal with a large group of interconnected objects.

Chebyshev polynomials are known for their special properties. They are being used to produce chaotic maps which in turn are suitable for cryptosystem.

This paper presents, the concept of transformations on graphs with the help of cellular automata. In this context the properties of transformations are studied and they are used to build a hybrid cryptosystem that uses both public and private key concepts. The concepts of ElGamal cryptosystem based on chebyshev polynomials have also been used. The strength of the algorithm is analyzed. The proposed method on comparison with AES resulted in reasonable security with less time complexity.

While section Two focuses on review of related work in brief, section three touches upon preliminaries. Similarly section four analyses the concept of graph transformations using cellular automata, and section five details the proposed method exercised alongside the algorithm designed.

* Professor, Department of Information Technology, Gayatri Vidya Parishad College of Engineering(A), Madhurawada, Visakhapatnam, Andhra Pradesh, India. Email: kishorempk73@gvpce.ac.in

** Email: sunayana.b05@gmail.com

The strength of the algorithm is compared with AES algorithm in section six.

Related Work

Cryptosystems using Cellular Automata (CA) has been initiated in 1980s. P.Guan [1] developed cryptosystems using public key mechanism.

In secret key systems, CA has been used to generate Pseudo Random sequences. Wolfrom [2, 3], has extensively studied the applications of CA, in particular to cryptosystems. Nandi et al., [4] also worked extensively on these concepts. Symeon Bozapalidis et al, [5] studied a relational graphoid, also defined Automata on directed hyper graphs. Priyadarsini PLK et al.,[6] developed new algorithms using Hadamard encoding by generating random Hadamard matrices of order 16 from strongly regular graphs that have a specific property. The cellular automata on cayley graphs are studied from computational point of view by Zsuzsanna Roka[7]. In their study they established a sufficient condition for the following: given two Cayley graphs, every cellular automation on the first graph can be simulated by a cellular automation on the second and vice-versa.

Ljupco Kocarev et al., designed Public key encryption with chaos [8] which is based on chaotic maps. The working of this algorithm is based on semi group property of Chebyshev maps. These algorithms are proven to be as secure as RSA algorithm. S.Vairachilai et al., presented a public key cryptosystem using chebyshev polynomial based on edge information [9] that is secure, practical and is used for both Encryption and Digital Signature. The cellular automata concepts are applied to graphs by Klaus Sutner [10]. In his work he studied the cellular automata with additive rules on finite undirected graphs, in which the problem of deciding whether a given configuration has a predecessor is addressed. Carsten Marr et al., in Cellular Automata on graphs [11] reviews the previous results of cellular automata on graphs by establishing relation between the network architecture and Dynamics. Graphs obtained in simulated evolution procedure from Eidos-Reny (ER) graphs and then selecting low entropies of Cellular Automata dynamics are studied.

The cellular transformations on graphs in general is still a promising area that offers complexity and speed for the algorithms.

In the present work, the transformations on graphs using one dimensional cellular automata are studied via adjacency matrices and a hybrid cryptosystem is developed which uses both the public and private key mechanisms.

Preliminaries

Cellular Automata(CA)

Several approaches to describe cellular automata are available in literature. A Cellular Automata is a regular lattice of cells (grids) that changes their state synchronously, according to a local update rule that specifies the new state of each cell based on the old states and its neighbors. CAs are dynamic systems in which space and time are discrete, they exhibit some inherent features like parallelism, locality, simplicity, unpredictability and homogeneity. Processing with cellular automata is naturally too fast, efficient in hardware and software implementations.

Cellular Automata are classified into two types based on the usage of rules. If all the cells obey the same rule it is known as uniform cellular automata otherwise non-uniform cellular automata.

Several types of boundary conditions are also studied by researchers. A Cellular Automata with periodic boundary is one in which the extreme cells lie adjacent to each other.

The state of all cells at time 't', is called configuration of CA at time 't' and is denoted CA^t. The next state of the CA is denoted by CA^{t+1}. Most of the researchers considered only two states for a cell {0, 1} for their applicability through programming. The state of a cell at the next time step is determined by the transition function along with current state of the cell and states of surrounding neighborhood cells. This phenomenon is represented as follows:

$$CA_i^{t+1} = f(C_{i-r}^t, C_{i-r+1}^t, \ldots, C_{i-}^t, C_i^t, C_{i+1}^t, \ldots C_{i+r}^t) \quad (1)$$

Where C_i^t means cell at time 't', C_i^{t+1} means cell at time 't+1', r is the neighborhood radius. If the radius is taken as one (r=1 i.e. 3-neighborhood) with one dimensional cellular automata then the next state of CA is represented as

$$CA_i^{t+1}=f(C_{i-1}^t, C_i^t, C_{i+1}^t) \quad (2)$$

The present paper examines only one dimensional cellular automata with circular boundary having neighborhood radius as 1.

Chebyshev Polynomial

Polynomials over finite field are used in cryptography to enhance the security of the cryptographic system [12]. Chebyshev polynomials are a sequence of orthogonal polynomials which can be defined recursively.

The Chebyshev polynomials of first kind are defined by the recurrence relation:

$T_0(x) = 1$, $T_1(x) = x$ and $T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$ (3)

Chebyshev polynomials are used to transmit the key.

Cellular Transformations on Graphs

Let G = (V, E) be a directed graph. Let $A_G = [a_{ij}]$ be the adjacency matrix of the graph G.

Throughout the remaining part of the paper directed graphs are considered.

A one-dimensional cellular automation rule K, is applied on each of the rows of the matrix with circular boundary.

Let $r_1, r_2, \ldots, r_m$ be the rows of the matrix A_G. Let $s_1, s_2, \ldots, s_m$ be the rows obtained after applying the automation rule K. Let B be the matrix with rows $s_1, s_2, \ldots, s_m$. Consider the graph H with B_H as the adjacency matrix. Mathematically the transformation can be defined as

K: G → H, where, $A_G = [g_{ij}]$, $B_H = [h_{ij}]$ and $[h_{ij}] = K([g_{ij-1}, g_{ij}, g_{ij+1}])$.

In principle the rules can be applied in several ways. For instance these rules can be applied column wise and also with different boundary conditions that may produce new kind of graphs. Similarly, two dimensional automata rules can also be applied on the adjacency matrix. The results can also be studied on directed as well as undirected graphs. In the present paper only one dimensional rules are applied on rows and analyzed graph transformations.

The process is illustrated through the adjacency matrices in the following example.

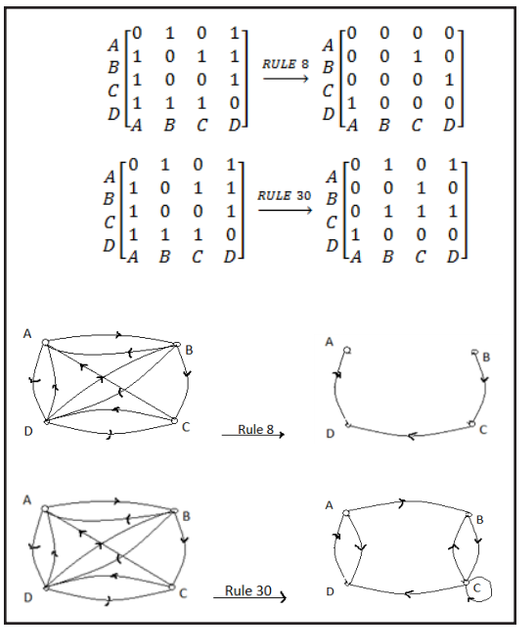


Fig. 1: Cellular Transformations on Graphs

The transformations are studied on a variety of graphs and the following observations are made. The exhaustive study of transformations on all types of graphs with all rules is quite a challenging task.

Transformations are applied on certain graphs and observations are noted.

Transformations

In this section the transformations on different types of graphs are studied with all the rules. The results are tabulated in terms of categories of rules and the edges in the corresponding transformed graph.

In the following tables 1-8 different graphs are considered and the patterns of change in number of edges for the corresponding rules (0-255) are noted.

Table 1: Directed Graph with 5 Vertices and 20 arcs with a Pair of Directed Arcs between each Pair of Vertices

Edges	10	15	20	25
Rule number [0-255]	128	8*(2k+1) and 8*3k for k≥0	2+k, k≥0 except, {8*(2k+1) k≥0 and 8*4k k≥1}	32*(2k+1) k≥0

The above observations also represents an undirected complete graph with 5 vertices and 10 edges.

Table 2: Directed Bipartite Graph with 7 Vertices and 18 Arcs with a Pair of Directed Arcs between each Pair of Vertices

Edges	37	40	41	42	46	49
Rule number [0-255]	4+8k, k>=0	3+2k, k>=0	32+64k, k>=0	16+32k, 18+32k 2+4k k>=0	8*(2k+1) 8,24	128

The above table also represents an undirected bipartite graph with 7 vertices and 9 edges.

Table 3: Cyclic Graph with 6 Edges and 12 Directed Arcs such that Two Directed Cycles Exists in Opposite Directions

Edges	0	6	12
Rule number [0-255]	8k,k=1,2,3 8*(2k+1),k>=0	K {k!=(4+8m), 8*(2m+1), m>=0, 8*m, m=1,2,3}	4+8k, k>=0

The above table also represents an undirected cyclic graph with 6 vertices and 6 edges

Table 4: Directed Graph with 6 Vertices (Two Groups of 3 Vertices each) and 12 Pairs of Directed Arcs between Vertices of Opposite Groups

Edges	0	8	16
Rule number [0-255]	8k {except 8*(2+4k)}, k>=0	2k {except 8k}, 8*(2+4k),k>=0	2k+1, k>=0

The above table also represents Bipartite graph with 6 vertices and 6 edges.

The above observations indicate that the changes in the number of edges from transformed graph and that of the original graph is highly sensitive to the graph and the rule.

Boolean Formulae

Stephen wolfram [2] and Evangelos Georgiadis [13] identified Boolean expressions for the one dimensional rules that are minimal in the sense of less number of operation.

On the same lines the changes in the edges of the graph under the transformations can also be expressed.

As an illustration, if rule 30 is used then the resultant edge can be identified by the following:

$$e_{ij} = (e_{ij-1} \wedge \bar{e}_{ij} \wedge \bar{e}_{ij+1}) \vee (\bar{e}_{ij-1} \wedge e_{ij}) \vee (\bar{e}_{ij-1} \wedge e_{ij+1}) \quad (5)$$

Where $e_{ij} e_{ij}$ denote the presence of the edge and $\bar{e}_{ij}\, \bar{e}_{ij}$ absence of the edge.

Using the above expression the resultant adjacency matrix and hence the resultant graph can be identified under a given cellular transformation.

Boolean expressions can be written for all one dimensional CA rules in terms of graph transformations by following the above nation and hence graphs can be identified.

Proposed Method

This section presents the proposed hybrid cryptosystem. In this method encryption of the plain text is based on cellular automata transformations over graphs. In this, the public key generated using the 1D-cellular automata rules. Similarly the private key is generated using the ElGamal mechanism applying Chebyshev polynomials over finite field.

The plain text is divided into blocks, each containing same size which are transformed into matrices using ASCII code with necessary padding with zeros accordingly. The public key would be the information containing set of rules, iterations to be applied and the set of graphs. The private key is containing chosen set of rules, iterations and a graph. The private key is encrypted using ElGamal Algorithm applying Chebyshev polynomials over finite field.

ElGamal Algorithm using Chebyshev Polynomial for Key Exchange

The algorithm presented here is given by Pina Bergamo et al., [14].

Let A denote the sender and B the receiver.

Step 1. B selects a large integer 's'. He then chooses a random number $x \in [-1, 1]$ and calculates $T_s(x)$

Encryption method:

Step 2. The public key would be (x, $T_s(x)$). 'M' denotes the message as a number M ∈ [-1, 1] and selects a large integer r.

Step 3. Then A calculates $T_r(x)$, $T_{r.s}(x)$ (= $T_r(T_s(x))$) and calculates X = M * $T_{r.s}(x)$.

Step 4. He then sends cipher text C = ($T_r(x)$, X) to B.

Decryption method:

Step 5. B uses the private key 's' to compute $T_{s.r}(x)$ = $T_s.(T_r(x))$.

Step 6. The message 'M' is computed by formula M=X / $T_{s.r}(x)$.

Text Encryption

A(sender) publishes sets of rules, iterations and graphs so that each receiver can choose his own combination to choose from. The receiver B selects the key matrix [rule number(R), iterations (I), graph (G)], from the public key consisting of combinations of rules, iterations and graphs and encrypts using the Elgammal encryption system as described above and send it to A. A decrypts the key message and identifies the key to be used to encrypt the text message for B. Let G_1 be the graph obtained by using the cellular automata rule R on G with I number of iterations. Now the first block of text (T_1) is encrypted using G_1 to generate G_2 as G_1 (XoR) T_1 and the process is repeated in the chaining mode sequentially for all blocks. On the other side, receiver obtains G_1, using the key matrix, and all the blocks of text are decrypted in the reverse process sequentially.

Two or more such key matrices can be created to enhance the security and the same be used in a specified sequence as desired by the receiver so that even if one key is compromised the complete message fails to get retrieved.

Algorithm

Let A be the sender and B be the receiver. B wishes to get the information from A securely,

Let A publish the public key consisting of a set of rules, iterations and the set of graphs.

Step 1: B selects a set of keys consisting of specific rules, iterations and the graphs published by A.

Step 2: The set of key matrices {R_1, I_1, G_1},{R_2, I_2, G_2} so on., together with the sequence to be followed {K_1, K_2, K_3…,K_n}, are encrypted by B using ElGammal algorithm as given in 5.1 and transmitted to A.

Step 3: A decrypts the message sent by B and retrieves the key matrices and the sequence.

'A' performs the following

Step 4: The plain text is divided into blocks as $text_1$, $text_2$,… ,$text_n$

Step 5: Generate matrices using ASCII code for the input text.

Step 6: Using the key matrix obtain a graph D_1 by applying rules R_1 of cellular automata I_1 number of times on the chosen graph G_1 in the key matrix and similarly D_2 using R_2, I_2, G_2.

Step 7: Now the first block is encrypted using this D_1 to generate D_2 , D_2 is used to generate D_3 and so on until all blocks are exhausted in the specified sequence using chaining mode and are transmitted to B.

Step 8: At the receiver side, using the key matrix, receiver obtains D_1, D_2 by using key matrices.

Step 9: Entire text is decrypted using D_1, D_2 and sequence.

Cryptanalysis

The Graphs are chosen for transformation in such a way that they produce a larger complexity, that is ensured by keeping the proportion of one's and number of zero's after transformation are as close as possible.

Cipher text only: Attacker has the knowledge of multiple cipher texts but do not have the knowledge of corresponding plain text.

In the brute force attack, the attacker has to try M×N×O number of combinations where M denote the number of rules, N denote the number of graphs and O denote the number of iterations. Large numbers of N,O can be displayed as there are 2^{100} possible graphs for a simple 10×10 matrix. Also the set of graphs can be refreshed from time to time.

2. Known plain text: Attacker has the knowledge of some plain text M, cipher text C pair, only a part of the text get retrieved even in this as two sets of keys are being used. Further, to enhance security more number of keys

are to be used in a specified sequence, as the portion of the exposure is minimal in this case.

Key streams like one key matrix for one block of text may also be used depending on the security requirement of the users.

Linear Cryptanalysis: Linear cryptanalysis is performed on the proposed method as follows,

Considering the input of S-box (Encryption mechanism) as plain text bits (X_i) and the output as cipher text bits (Y_j) the key chosen is of matrix 4×4, the probability of occurrence of zero's and one's is given by the following expression:

Here, i=1 to 16, j=1 to 16, the total number of combinations obtained are 2^{16} Here,' ⊕' represents XOR operation.

All the bit sequences are classified into linear expressions and the following probability of occurrence is observed as detailed below.

The linear cryptanalysis measures the probability of occurrence of zeor's and one's, and the given system is good if both the probabilities are close to 0.5. In the present case the average probability of zero's is 0.3511 which is 70% close to 0.5 and similarly the average probability of one's is 0.6488 which is 77% closer to 0.5. Hence the method can be considered safer.

Table 5. The Linear Expressions Identified and the Corresponding Probabilities in Linear Cryptanalysis

Linear Expression	Probability of occurance	
	zero's	one's
$c_i=y_i \oplus(x_i \oplus x_{i+1} (\oplus x_j)^*)$	0.3907	0.6092
$d_j=x_i \oplus(y_i \oplus y_{i+1}(\oplus y_j)^*)$	0.391	0.609
$x_i \oplus y_j$	0.3125	0.6875
$c_i \oplus d_j$	0.3125	0.6875
$c \oplus(d_i \oplus d_{i+1}(\oplus d_j)^*)$	0.39	0.61
$d \oplus(c_i \oplus c_{i+1}(\oplus c_j)^*)$	0.31	0.69

Confusion gives a measure of dependency of cipher text on several parts of the key.

Diffusion gives a measure of dependency of cipher text on plain text, that is, if the plain text is changed then several character of the cipher text may change.

The analysis noted down shows the confusion on different graphs (of size 10×10) keeping the rule 30 applied on them with only one iteration.

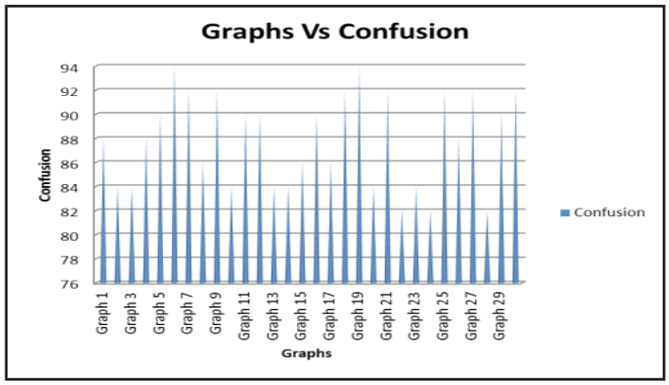


Fig. 2: Different Graphs of Size 10×10 on X-axis and Number of Bits on Y-axis

It is clear from the analysis that confusion depends on the nature of the graph and it ranges between 82 to 94 against a maximum of 100 possible.

The confusion analysis with different a rule keeping a 10×10 graph and 10 iterations as constant is shown below.

From the analysis it is observed that all the rules can be classified into (specific to the graph and iterations) 4 categories based on the number of bits changed as given in the following table.

Table 6: Classification of Rules Based on the Number of Bits Changed

Number of bits	36	37	61	64
Rule number [0-255]	1, 128	4+4k, k>=0 (except 16+16k, k>=0,k!=3,11)	2+4k, k>=0 16+16k, (k>=0, K!=3,11)	2k+1, k>=0

Further the confusion is also observed with different sizes of graphs and with different iterations with the rule 30.

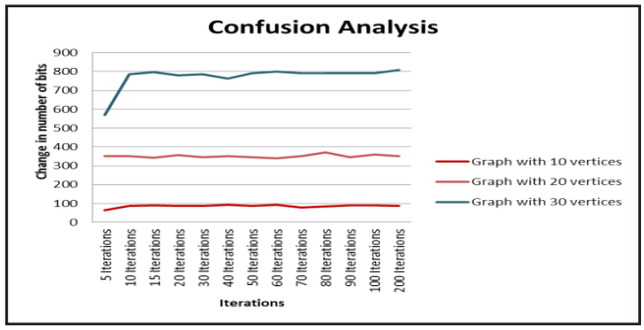


Fig. 3: Number of Iterations on X-axis and Change in the Number of Bits on Y-axis

From the above graph it can be observed that as the size of the graph (adjacency matrix) increases greater confusion is achieved and in each case, confusion is above 90% of the total available bits. However, there is no much change with the increase in iterations.

The Diffusion analysis is shown in the figure below.

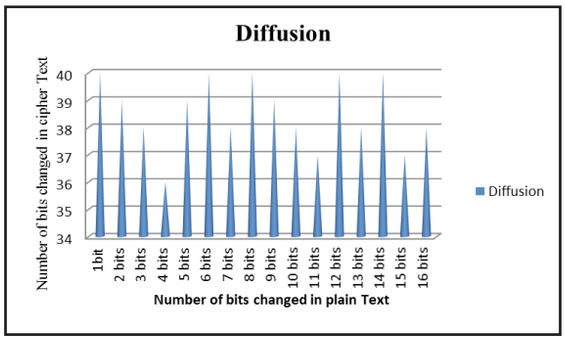
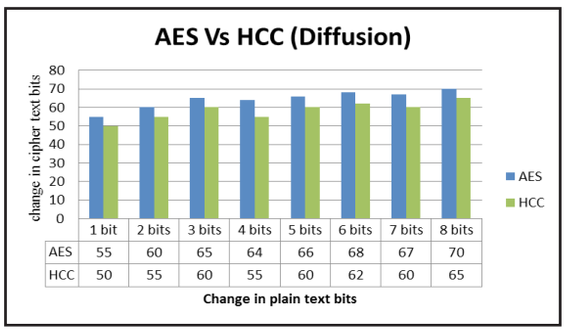


Fig. 4: Number of Bits Changed in Plain Text on X-axis and Number of Bits Changed in Cipher Text on Y-axis

From the graph it can be observed that the diffusion is specific to the graph and the rule.

Diffusion analysis when HCC is compared with AES algorithm is given as follows:



	1 bit	2 bits	3 bits	4 bits	5 bits	6 bits	7 bits	8 bits
AES	55	60	65	64	66	68	67	70
HCC	50	55	60	55	60	62	60	65

Fig. 5: Comparison with AES

It can be observed from thea analysis that AES is having slightly higher diffusion (<10%) when compared with proposed method.

Analysis of HCC with AES with respect to the execution time is as shown below:

Execution Environment:

Operating System: Ubuntu 14.04, RAM: 2GB, Hard Disk: 500GB

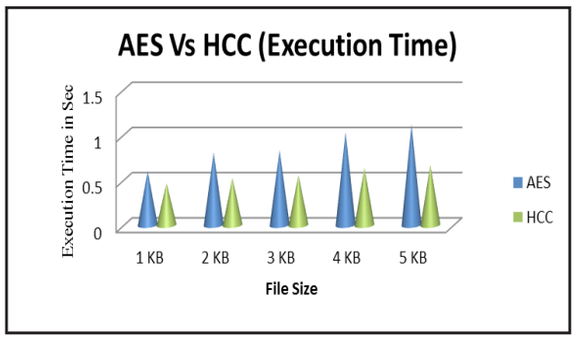


Fig. 6: Comparison with Respect to Execution Time (in seconds)

From the above analysis it is observed that HCC exhibits 10% lesser diffusion than AES but computational speed is 49% more than that of AES.

Conclusion

The new concept of transformations on graphs exhibits the potential to design new cryptosystems having less time complexity with similar security strengths comparable to existing systems. However as only one dimensional Cellular Automata is applied, there is a scope for further probe into the possibilities of building better methods with two dimensional Cellular Automata.

References

1. Guan, P. (1987). Cellular automaton public-key cryptosystem. *Complex Systems 1,* (pp. 51-56).
2. Wolfram, S. (2002). *A New Kind of Science.* Wolfram Media, Inc., (pp. 884-885).
3. Wolfram, S. (1986). Cryptography with cellular automata. *In Advances in Cryptology: Crypto 85-344 Proceedings, LNCS 218, Springer*, (pp. 429-432).
4. Nandi, S., Kar, B. K., & Chaudhuri, P. P. (1994). Theory and applications of cellular automata in cryptography. *IEEE Transactions on Computers,* 43, (pp. 1346-1357).
5. Bozapalidis, S., & Kalampakas, A. (2008). Graph automata. *Theoretical Computer Science*, 393(1-3), 147-165.
6. Priyadarsini, P. L. K., & Ayyagari, R. (2013). Cihpers based on Special Graphs. *Proceedings of the 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, (pp 460- 465).

7. Sutner, K. (1988). Additive automata on graphs. *Complex Systems*, 2(6), 649-661.
8. Marr, C., & Hutt, M. T. (2012). Cellular automata on Graphs: Topological properties of ER graphs evolved towards low-entropy dynamics. *Entropy*, 14, 993-1010.
9. Kocarev, L., Makraduli, J., & Amato, P. (2005). Public Key Encryption based on Chebyshev Polynomials. *Circuits Systems Signal Processing*, 24(5), 497-517.
10. Georgiadis, E. (2007). A note on minimal boolean formula size of one-dimensional cellular automata. *Journal of Cellular Automata*, 4(2), 1-4.
11. Bergamo, P., D'Arco, P., De Santis, A., & Kocarev, L. (2005). Security of Public-Key Cryptosystems based on Chebyshev Polynomials. *IEEE Transactions on Circuits and Systems-I*, July, 52(7), 1382-1393.