IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Mathematical Modelling of Engineering Problems
Vol. 10, No. 6, December, 2023, pp. 2121-2126
Journal homepage: http://iieta.org/journals/mmep

A Novel Approach of 1-D Cellular Automata in Cryptosystem

Gaverchand Kukaram, Venkatesan Ramasamy*

Department of Mathematics, College of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur 603203, Tamil Nadu, India

Corresponding Author Email: venkater1@srmist.edu.in

https://doi.org/10.18280/mmep.100623

Received: 14 June 2023
Revised: 20 September 2023
Accepted: 8 October 2023
Available online: 21 December 2023

Keywords:
cellular automata, cryptography, encryption, decryption, attacks

ABSTRACT

Cryptosystems worldwide employ techniques for the encryption and decryption of sensitive data, relying extensively on secret keys. In this context, the generation of a randomized, secured secret key and its size hold paramount significance in ensuring confidentiality, data integrity, and resistance to a plethora of security attacks, rendering it arduous for potential intruders to predict key sequences. This study aims to generate the highly secured randomized secret key, minimize the time complexity and ensure efficiency of the cryptosystem. In the key generation methodology presented, a novel approach was introduced, taking into account the receiver's credentials and employing the elementary cellular automata (CA). Rule 150 of CA was strategically leveraged to generate a secret key, undergoing numerous iterations to bolster security, intricacy, and to compound the predictability challenge of the key. Python was the chosen medium for the implementation of the proposed model. Time complexity was rigorously evaluated, and a comparative analysis was conducted against established cryptographic algorithms, notably Rivest-Shamir-Adleman (RSA) and Advanced Encryption Standard (AES), to ascertain efficiency. Subsequently, a frequency analysis, underpinned by a letter frequency distribution chart, was undertaken to confirm the randomness of the resultant ciphertext (CT). To further validate the robustness of the proposed model, security assessments encompassing brute force attacks, CT attacks, known plaintext (PT) attacks, and chosen PT attacks were meticulously examined. Cumulative findings corroborate the heightened security and efficiency of the proposed model in contrast to its predecessors. Anticipated future research horizons include the potential incorporation of CA in domains such as image and video encryption, blockchain technology, cryptocurrency, and digital signatures, aiming to cultivate superior security infrastructures.

1. INTRODUCTION

In today's rapidly evolving digital age, the paramount significance of security cannot be understated. A universal desire has been identified to ensure data is stored securely and that private communications or data transactions occur with heightened security. Nonetheless, various avenues exist through which data breaches can occur. In this context, cryptography emerges as a critical tool to shield data and transactions from potential intruders during online communications. Historically, the vast realm of cryptography has been enriched by diverse techniques proposed by myriad researchers. Fundamentally, cryptography pertains to the practice of safeguarding communication by encoding information, rendering it accessible and comprehensible exclusively to authorised entities. Herein, the CIA triad (Confidentiality, Integrity, Authenticity) is recognised as a foundational concept.

Cryptography, as a discipline, can be broadly categorised into symmetric key cryptography (otherwise termed private-key cryptography) and asymmetric key cryptography (referred to as public-key cryptography). Both these categories are deployed to secure digital communications. Notably, cryptosystems founded upon CA have been demonstrated to possibly outperform classical systems in terms of efficacy [1] Within the domain of symmetric key cryptography, CA has been utilised for the creation of S-boxes [2] and the generation of pseudo-random numbers. Noteworthy contributions in this regard include the proposition by Kotoulas et al. [3] of a pseudo-random number generator leveraging a 1-D cellular automaton. This innovation reportedly facilitated the real-time generation of high-quality random numbers. An enhancement in randomisation quality was observed upon the integration of programmable cellular automata (PCA) [4].

In the burgeoning field of cryptography, encryption algorithms based on CA consistently demonstrate promising results. The mathematical underpinnings of cryptography were introduced by Silverman et al. [5]. A myriad of CA applications has been elucidated, including roles in image encryption [6], bio-hash code generation [7], watermarking [8], and various other cryptographic procedures [9]. Notably, Data Encryption Standard (DES) and blowfish algorithms,

along with their foundational functions, were examined in depth by Nie and Zhang [10]. An intriguing comparison between DES and AES algorithms using CA was made by Panda et al. [11]. However, this model faced vulnerabilities against security breaches, primarily due to the dwindling robustness of AES and DES. Parashar et al. [12] accentuated the indispensability of CA rules within block cipher algorithms, which fall under the umbrella of symmetric key cryptography. They further integrated non-complemented and hybrid CA rules to conceive group CA.

The incorporation of CA within cryptographic methodologies has been expounded upon in various studies [13, 14]. Furthermore, several applications of symmetric key cryptography have been scrutinised, focusing on the intricate structure of CA rules [15]. Kumersan et al. [16] presented a comprehensive analysis of various CA concepts, discussing their relevance in cryptography. However, this analysis did not extend to security implications. Distinct cryptographic strategies using face anti-magic labelling were unveiled by Kuppan et al. [17], while Rao et al. [18] advanced a novel image encryption approach through visual cryptography, leveraging the Least Significant Bit (LSB) technique in the spatial domain. In another study, Rama et al. [19] augmented the use of CA for DES key generation, primarily benefitting from the randomness induced by CA rule 30, although a limitation was observed in the reduced key size. Despite the efforts of Stanica and Anghelescu [20] in proposing a cryptographic algorithm grounded on hybrid 1-D CA and subsequent NIST testing, as well as hardware and software implementations, the model was marred by high time complexity, and a comprehensive security analysis remained conspicuously absent.

A critical evaluation of the aforementioned literature reveals intrinsic limitations: elevated time and space complexity, inadequately sized keys not generated through random processes, gaps in security analysis, and a lack of authentication. Most notably, the crucial frequency analysis, pivotal for verifying model randomness, often went unaddressed.

The core ambition of the present study revolves around addressing these identified drawbacks, culminating in the introduction of an enhanced cryptosystem harnessing 1-D CA. Emphasis is placed upon the development of encryption and decryption schemas, the generation of a 128-bit secret key using 1-D CA rules, and a rigorous comparison against established symmetric key algorithms. The integration of CA into cryptography is anticipated to bolster the overall security, efficiency, and randomness of the proposed cryptosystem. Of particular note is the strategic use of characters from receiver credentials and the attributes of CA rule 150, augmenting randomness during the key generation phase.

This document is structured as follows: Section 2 elucidates the preliminary concepts, supplemented with illustrative examples. Section 3 delves into the intricacies of the proposed model, elucidating relevant algorithms. Section 4 is dedicated to an exhaustive analysis of potential attacks, presenting algorithmic outputs. Finally, the conclusions drawn from the study are encapsulated in Section 5.

2. PRELIMINARIES

2.1 CA

CA was first introduced by Von Neuman [21] and subsequently developed further by Wolfram [22, 23]. Defined as a dynamical system in which both space and time are discrete, CA is composed of a collection of cells, each of which possesses a finite set of states. These states within the cells are updated simultaneously in discrete time intervals, governed by a local interaction rule that remains consistent across all cells.

A CA is charaterised by a 4-tuple $\{L, Q, N, f\}$:
where,

L is a lattice or regular grid where cells are organised.

Q is a set of finite states.

N is a neighbourhood relation (in 1-D cases, the neighbourhood of the current cell comprises its two adjacent cells).

f is a local transition rule.

f: $S^k \rightarrow S$.

k is the number of cells in the neighborhood of each cell.

S^k is the set of all possible combinations of the states.

2.2 1-D CA

1-D CA is unique in their linear configuration, and it encompasses two possible states (0 or 1). Each state possesses three neighbours: the state itself and the two adjacent states. At consistent time intervals, denoted by t, each cell's state undergoes modification for the subsequent state in different time steps. The alteration in each cell's state at time interval $t+1$ is influenced by the following rules:

- The state's current status during the prior time step.
- The previous time step's left neighbour.
- The previous time step's right neighbour.

With 1-D CA consisting of three cells, there are 2^3=8 potential patterns. Typically, there exist 2^8=256 CA rules. Consequently, these rules can be represented as decimal numbers ranging from 0 to 255 via a 3-variable function.

The illustration of rule 150 is as follows:

$f(111)=1$
$f(110)=0$
$f(101)=0$
$f(100)=1$
$f(011)=0$
$f(010)=1$
$f(001)=1$
$f(000)=0$

In the above defined rule 150, the values inside the brackets represent all potential neighbourhood patterns for each state. The values situated on the right are employed to transition each state from the zeroth to the first generation. Rule-150's representation is depicted in Figure 1.

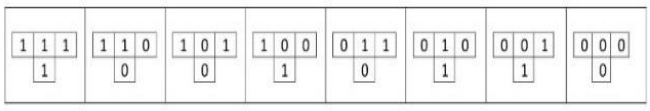

Figure 1. CA rule 150

Subsequently, for the next state of time step t, rule 150 (10010110_2) is applied. The mathematical representation of this next state, or rule 150, is articulated as:

$$t + 1 = (p + q + r) mod 2 \quad (1)$$

where,

$t+1$ represents the succeeding state.

p signifies the left neighbour.
q represents the current neighbour.
r stands for the right neighbour.

For an initial state defined as 00101001_2, iteration is performed using Eq. (1), with the results after five iterations documented in Table 1. Table 2 portrays the subsequent state values for CA rules 110, 150, and 180.

Table 1. Result of the initial state after five iterations

Initial State	0	0	1	0	1	0	0	1
	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Iteration 1	1	1	1	0	1	1	1	1
Iteration 2	1	1	0	0	0	1	1	1
Iteration 3	1	0	1	0	1	0	1	1
Iteration 4	0	0	1	0	1	0	0	1
Iteration 5	1	1	1	0	1	1	1	1

Table 2. Illustration of CA numbering rule for next state

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Rule	111	110	101	100	011	010	001	000
110	0	1	1	0	1	1	1	0
150	1	0	0	1	0	1	1	0
180	1	0	1	1	0	1	0	0

2.2.1 Characteristics of CA rule 150

Rule 150, an elementary rule among CA, is renowned for its inherent chaotic and complex behaviour. It is frequently utilised within cryptography for the creation of intricate, asymmetric, and non-repetitive patterns. By employing this rule, a key is generated after several iterations.

2.3 Permutation

Permutation is characterised as a rearrangement process of set elements, such that every element in the initial set is mapped uniquely to an element in another set, fulfilling a bijective mapping criterion. This ensures a one-to-one correlation between the elements of the original and the new set. Within symmetric-key cryptography, permutations are regularly applied to convert data into an encrypted or scrambled format.

3. PROPOSED MODEL

A brief overview of the proposed scheme is provided: A 128-bit secret key is generated within this model. Subsequently, the encryption and decryption algorithms are elucidated in detail, with outcomes demonstrated using the Python programming language. Following this, results are analysed and conclusions regarding the model are drawn. The workflow of the proposed model is outlined in Figure 2.

3.1 Key generation

Upon receipt of the receiver's credentials, the sender generates a 128-bit key, believed to enhance security. To generate this key, input is initially taken from receiver credentials, ensuring a length of 16 to achieve the 128-bit key. These credentials are then converted to ASCII values, which are further transformed to binary values. Progression requires the definition of a CA rule. CA rule 150 is employed, with initial states sourced from the 8-bit binary values. This rule is applied to each initial state to determine the subsequent states. The state-updating process is depicted in Figure 1. After performing up to 10 iterations, the final iterated 8-bit values are amassed and concatenated into a singular binary string. If the resultant string exceeds 128-bits, only the initial 128 bits are considered as the secret key.

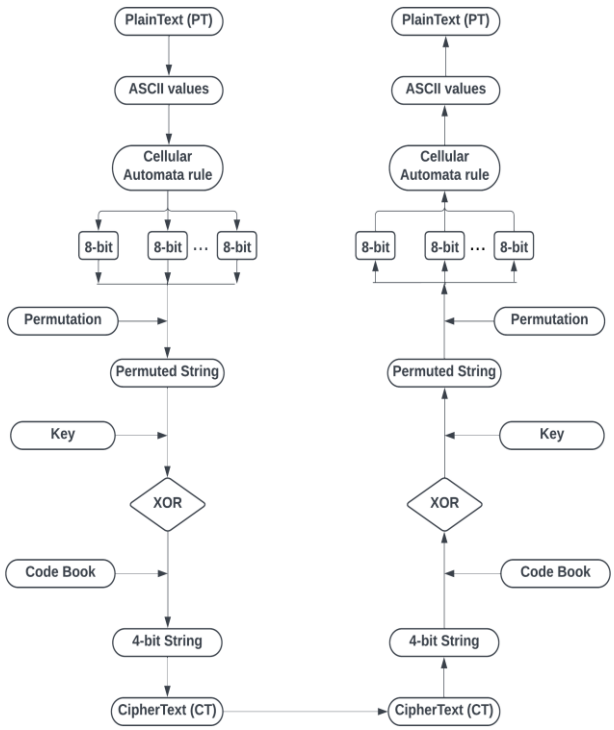

Figure 2. A flowchart detailing the encryption and decryption processes within this model

3.2 Encryption

Table 3. Random conversion of 4-bit binary string into alphabets

4-Bit Binary String	English Alphabets	4-Bit Binary String	English Alphabets
0000	QEWR	1000	AYJQ
0001	PLKJ	1001	GQJK
0010	FLRK	1010	FKEP
0011	UXCV	1011	LQSF
0100	YJFY	1100	VCJE
0101	UZRC	1101	QZYX
0110	AWYF	1110	SGVJ
0111	XPWZ	1111	REWY

The encryption process is outlined as follows: The sender inputs PT into the encryption algorithm. This input is first transformed into its corresponding American Standard Code for Information Interchange (ASCII) values and subsequently converted to binary form. A permutation order (PO) (PO [3, 1, 2, 0, 6, 7, 5, 4]) is then applied to each 8-bit binary segment, yielding a new binary string. An exclusive OR (XOR) operation is performed between the permuted string and the key, generating XOR results in binary form. Each 4-bit segment of these binary values is transformed into characters, referencing the corresponding values from codebook Table 3, to produce the necessary CT. The encrypted text, accompanied by certain parameters, is encrypted again using the receiver's

public key and transmitted to the intended recipient. If the permuted string length exceeds 128-bits, it is segmented into 128-bit blocks with the XOR operation applied to each block.

3.3 Decryption

For decryption, the CT serves as the primary input. Each 4-character segment within the CT is cross-referenced with codebook Table 3, enabling character-to-binary conversion. An XOR operation between the binary representations and the secret 128-bit key is then executed. The result undergoes a permutation operation to produce a new binary string. Subsequently, every 8-bit segment of this binary string undergoes conversion to its ASCII counterpart. Lastly, these ASCII values are substituted with their corresponding characters to recover the original PT.

Algorithm 1: Key generation

1: An input of length 16 is selected from the provided credentials.
2: This input is then transformed into its corresponding ASCII values.
3: These ASCII values are subsequently converted into 8-bit binary strings.
4: CA rule 150 is employed, converting each 8-bit binary string into a new string, which stands as the 10^{th} iteration of the value from step 3.
5: The final iterated values from step 4 are concatenated to yield a 128-bit key.

Algorithm 2: Encryption

Input: PT
Output: CT
1: Characters within the PT are converted to their respective ASCII values.
2: These ASCII values are transformed into their equivalent binary forms.
3: A PO is applied, altering the positions within each 8-bit string.
4: An XOR operation is executed between the permuted string from the previous step and the 128-bit key.
5: The resulting XOR values are mapped to their corresponding characters using the codebook denoted as Table 3.
6: The CT is subsequently generated.

Algorithm 3: Decryption

Input: CT
Output: PT
1: The CT is considered.
2: The codebook, Table 3, is employed to retrieve the corresponding XOR value mapping for the CT.
3: An XOR operation is executed between the 128-bit key and the XOR value, yielding a permuted string.
4: A permutation operation is applied on the result from step 3.
5: Each permuted 8-bit string is converted back into its ASCII value.
6: The ASCII values are further converted into corresponding letters, resulting in the recovery of PT.

4. RESULTS AND DISCUSSION

The results derived from the proposed algorithms, encompassing key generation, encryption, and decryption, are delineated in Figure 3.

4.1 Frequency analysis

Within cryptographic studies, frequency analysis serves as a pivotal technique to ascertain the randomness of the CT. This randomness is affirmed by juxtaposing the observed frequency of the generated CT against the anticipated frequency of the English alphabet. The CT from Figure 3 was subjected to this analysis. The observed frequency (Table 4) was deduced by dividing the number of instances each alphabet manifested in the CT by the aggregate count of letters within the CT. Upon comparison, a significant deviation between the expected and observed frequency was identified, suggesting an enhanced difficulty for potential adversaries to discern the PT or to successfully decrypt the encoded message.

Total number of letters in the CT: 400.

$$\text{Observed frequency} = \frac{Occurence\ of\ each\ letter}{400}$$

4.2 Security analysis

This segment endeavours to affirm that the model introduced remains impervious to a spectrum of prevalent attacks.

4.2.1 Brute force attack

During this mode of attack, an intruder endeavours to enumerate the possibilities of the clandestine key. However, given the 128-bit size of the key in the proposed model, the assailant would necessitate 2^{128} computations to identify the key, rendering the task virtually insurmountable. Such a feature elucidates the robustness of this model's security framework.

4.2.2 CT attack

The attacker is granted access to several encrypted messages, but remains devoid of the corresponding PT or the concealed key. In the proposed model, several stages, including permutation, XOR, and the utilization of a codebook, are engaged to encrypt data. These intricacies further bolster the resilience of the model against such assaults.

4.2.3 Known PT attack

For this form of assault, an intruder gains partial access to both the CT and the corresponding PT, with the aim to decipher the concealed key. Notably, the model under study produces a plethora of CT characters for a singular PT character. Moreover, the CA rule is employed during the key generation process, which amplifies the model's resistance to this kind of intrusion.

4.2.4 Chosen PT attack

Under this scenario, an attacker possesses the liberty to select arbitrary PT and retrieve their respective CT. Despite being more potent than the known-PT attack, the proposed model remains robust. The adoption of PO and XOR operation for PT encryption complicates the attacker's attempts to deduce the CT from the PT.

4.3 Performance analysis

This section juxtaposes the performance metrics of the introduced model against extant cryptographic algorithms.

Evaluations were conducted on a Dell Inspiron, equipped with 8GB RAM, 1TB HDD, 128GB SSD, and running on Windows 10 OS. Python was harnessed for these evaluations, leveraging standard libraries and tools such as string manipulation, bitwise operations, and time measurement. Data sourced from a prominent dataset, containing textual data congruent for appraising the proposed model, was utilised. Analyses were undertaken on five diverse data sets, classified based on their size. Each dataset was subjected to 50 iterations to ascertain an average. The objective was to discern whether the proposed method consistently outperformed traditional encryption algorithms, both in terms of speed and memory consumption. A comparative exposition is presented in Table 5, while Figure 4 offers a visual representation of the time and memory comparisons between the extant and the proposed models.

Figure 3. Output of the proposed model

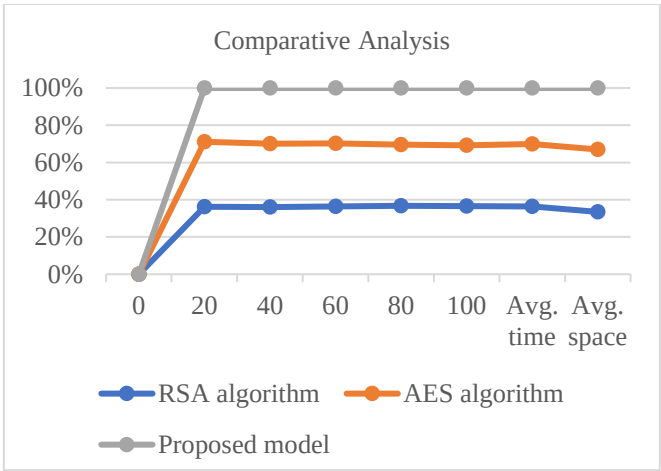

Figure 4. Time and space comparison

Table 4. Frequency analysis

Alphabet	Number of Times Occurred	Expected Frequency	Observed Frequency
A	19	0.0817	0.0475
C	14	0.0278	0.035
E	19	0.1270	0.0475
F	30	0.0223	0.075
J	33	0.0015	0.0825
K	22	0.0077	0.055
L	16	0.0403	0.04
P	22	0.0193	0.055
Q	29	0.0010	0.0725
R	22	0.0599	0.055
S	10	0.0633	0.025
U	10	0.0276	0.025
V	13	0.0098	0.0325
W	32	0.0236	0.08
X	24	0.0015	0.06
Y	46	0.0197	0.115
Z	28	0.0007	0.07

Table 5. Comparative analysis

PT (Length)	RSA Algorithm	AES Algorithm	Proposed Model
20	50	48	39.8
40	52	49	43
60	55	51	45
80	57	51	47.1
100	58	52	48.8
Average time (ms)	54.4	50	44.74
Average space (MB)	81.21	81.03	79.98

5. CONCLUSIONS

In the presented study, a 1-D CA model was introduced, employing linearity rules. Through this methodology, a random key of 128-bit dimensions was generated, subsequently utilised for encryption and decryption. The randomness of the generated CT was subsequently assessed through frequency analysis.

The security assessments conducted elucidated the robustness and efficacy of this innovative model, demonstrating its resilience against a multitude of cyber threats. Comparative evaluations indicated a favourable reduction in time and space complexity when juxtaposed with conventional algorithms. A notable relationship between cryptography and CA was illuminated through this study. The hallmark of this investigation lies in its capacity to generate a randomized secret key, a pivotal element in enhancing cryptographic security. Such findings bear profound implications for secure data communication, encompassing facets such as data encryption, data integrity, and confidentiality.

The overarching merits of the model delineated herein encompass heightened security measures, increased resilience, enhanced processing speed, diminished storage requirements, and a random secret key generation mechanism, often referred to as key management in cryptographic parlance. Future

endeavours are anticipated to encompass the exploration of diverse CA rules, expanding the current 1-D paradigm into a 2-D framework. Such an evolution would necessitate the adaptation to a grid structure, the formulation of innovative rules for cell state transitions within the grid, and the intricate management of encryption keys, all the while considering the resultant security and performance ramifications. Comprehensive testing and thorough documentation remain paramount to ensure the unwavering security and functionality of any extended cryptographic system.

REFERENCES

[1] Anghelescu, P. (2012), Hardware implementation of programmable cellular automata encryption algorithm. In IEEE International Conference on Telecommunications and Signal Processing, Prague, Czech Republic, pp. 18-21. https://doi.org/10.1109/TSP.2012.6256189

[2] Szaban, M., Seredynski, F. (2010). CA-based generator of S-boxes for cryptography use. In 2010 IEEE International Symposium on Parallel and Distributed Processing, Workshops and Phd Forum (IPDPSW), Atlanta, GA, USA, pp. 1-8. http://doi.org/10.1109/IPDPSW.2010.5470699

[3] Kotoulas, L., Tsarouchis, D., Sirakoulis, G.C., Andreadis, I. (2006). 1-D cellular automata for pseudo random number generation and its reconfigurable hardware implementation. In 2016 IEEE International Symposium on Circuits and Systems, Kos, Greece, p. 4. https://doi.org/10.1109/ISCAS.2006.1693661

[4] Anghelescu, P. (2011). Encryption algorithm using programmable cellular automata. In IEEE World Congress on Internet Security, London, UK, pp. 223-239. http://doi.org/10.1109/WorldCIS17046.2011.5749858

[5] Silverman, J.H., Pipher, J., Hoffstein, J. (2011). An Introduction to Mathematical Cryptography. Springer, New York, USA. https://doi.org/10.1007/978-0-387-77993-5

[6] Nandi, S., Roy, S., Dey, N, Nath, S., Chakraborty, S., Kaara, W.B.A. (2014). 1-D group cellular automata-based image encryption technique. In 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies, Kanyakumari, India, pp. 521-526. https://doi.org/10.1109/ICCICCT.2014.6993017

[7] Dey, N., Nandi, B., De, M., Das, A., Chaudhuri, S.S. (2013). Bio-Hash code generation from electrocardiogram features. In 2013 3rd IEEE International Advance Computing Conference, Ghaziabad, India, pp. 732-735 https://doi.org/10.1109/IAdCC.2013.6514317

[8] Acharjee, S., Chakraborty, S., Ray, R., Nath, S., Dey, N. (2014). Watermarking in motion vector for security enhancement of medical videos. In 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies, Kanyakumari, India, pp. 532-537. https://doi.org/10.1109/ICCICCT.2014.6993019

[9] Nandi, S., Kar, B.K., Pal Pushkar, P. (1994). Theory and applications of cellular automata in cryptography. IEEE Transactions on Computers, 43(12): 1346-1537. http://doi.org/ 10.1109/12.338094

[10] Nie, T.Y., Zhang, T. (2009). A study of DES and blowfish encryption algorithm. In TENCON 2009-2009 IEEE Region 10 Conference, Singapore, pp. 1-4. https://doi.org/10.1109/TENCON.2009.5396115

[11] Panda, S.P., Sahu, M., Rout, U.P., Nanda, S.K. (2011). Equivalence of DES and AES algorithm with cellular automata. International Journal of Communication Networks and Security, 1(1): 47-52. https://doi.org/10.47893/IJCNS.2011.1008

[12] Parashar, D., Roy, S., Dey, N., Jain, V., Rawat, U.S. (2018). Symmetric key encryption technique: A cellular automata based approach. In Advances in Intelligent Systems and Computing. Springer, Singapore. https://doi.org/10.1007/978-981-10-8536-9_7

[13] Wolfram, S. (1985). Cryptography with cellular automata. In Proceedings of the Conference, CRYPTO'85 on Advances in Cryptography. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, pp. 429-432. http://doi.org/10.1007/3-540-39799-x_32

[14] Tomassini, M., Perrenoud, M. (2001). Cryptography with cellular automata. Applied Soft Computing, 1(2): 151-160. http://doi.org/10.1016/S1568-4946(01)00015-1

[15] Roy, S., Nandi, S., Dansana, J., Pattnaik, P.K. (2014). Application of cellular automata in symmetric key cryptography. In 2014 International Conference on Communication and Signal processing, Melmaruvathur, India, pp. 3-5. http://doi.org/10.1109%2FICCSP.2014.6949906

[16] Kumaresan, K., Gopalan, N.P. (2017). An analytical study of cellular automata and its applications in cryptography. International Journal of Computer Network and Information Security, 10(12): 45-54. http://doi.org/10.5815/ijcnis.2017.12.06

[17] Kuppan, R., Shobana, L., Cangul, I.N. (2020). Encrypting and decrypting algorithms using strong face graph of a tree. International Journal of Computer Mathematics: Computer Systems Theory, 5(4): 225-233. http://doi.org/10.1080/23799927.2020.1807606

[18] Rao, K.S., Sridhar, M. (2021). A novel image encryption using parity based visual cryptography. Ingenierie des Systems d'Information, 26(1): 135-142. https://doi.org/10.18280/isi.260115

[19] Rama, R., Bala Suyambu, J., Arokiaraj, A., Saravana, S. (2013). A study of DES algorithm with cellular automata. International Journal of Innovative Management, information & Production, 4(1): 10-16.

[20] Stanica, G.C., Anghelescu, P. (2023). Cryptographic algorithm based on hybrid one-dimensional cellular automata. Mathematics, 11(6): 1481. https://doi.org/10.3390/math11061481

[21] Von Neuman, J. (1996). Theory of Self-Reproducing Automata. University of Illinois Press, London, UK.

[22] Wolfram, S. (1986). Theory and Applications of Cellular Automata. World Scientific, Singapore.

[23] Wolfram, S. (2002). A New Kind of Science. Wolfram Media Inc.