Journal of Information Hiding and Multimedia Signal Processing ©2019 ISSN 2073-4212
Ubiquitous International Volume **10**, Number **1**, January **2019**

Design Random Number Generator Utilizing The Futoshiki Puzzle

Ali Shakir Mahmood

Computer Science Department, College of Education,
Mustansiriyah University, Baghdad, Iraq

asmjhm2006@uomustansiriyah.edu.iq

Received March 2018; revised May 2018

Abstract. *This paper discusses and demonstrate the proposed method, for random number generator. The random number generator trying to discard the relationship between consecutive digit occurrences, moreover, these random sets can't be repeated except if the equivalent generator algorithm with same initial values are used. The designed generator stunned the earlier problems of an initial key size, minimal periodic and the capability to regenerate the similar sequence. The proposed generator employing the idea of Futoshiki puzzle game, this puzzle originated from Japan, trying to fill the equal dimension board with numbers (1 ... board size) without any repetition of digits in the row and column, also there is an expansion method to meet the users demands. The generated random numbers can be used in several aspects of security, such as encryption or password generator for network administrators. The generated random numbers must have subjected to several randomness tests called NIST statistical test suite to check the level of randomness of the produced sets, whereas, successfully passed all statistical tests were crossing the threshold P-value. The statistical properties indicate that the Futoshiki generator is substantially successful in producing random number with respectable statistical outcomes, high linear complexity and provides a small initial value with the capability to regenerate the equivalent sequence when feed up with the same initial value.*
Keywords: Futoshiki puzzle, Random number generator, Encryption key, Key expansion, NIST

1. **Introduction.** The random number is commonly cast for several implementations, like an encryption key for security applications, mathematical analysis, simulation and modulation, for picking random pieces from larger data sets [1, 2]. These sets of numbers can remain produced by examining random physical phenomena, like, wind speed, temperature and daylight sum level. This category of random generator named as a True Random Number Generator (TRNG). The TRNG need for more devices produce a set of random numbers and some shortages in the capability to re-producing the same series of random numbers unless the similar initial key is used. The reproduce of an equivalent set are impossible since the random set produced according to the natural physical phenomenon. Alternative type of these generators, called Pseudo Random Number Generator (PRNG), uses some mathematical procedures to produce the random sequence. PRNG is more appropriate for generating encryption key because it can be regenerate the same random sequence in both source and destination stations, furthermore, no need for additional equipment to generate this kind of random number. This type of generators needs

to initially seed to start the first iteration on generating process. The PRNGs can be categorized within the periodic generators, where they regenerate the similar sequence after a certain number of rounds [3]. The random number generators depicted in many general properties. Firstly, the succeeding number in the set cannot be prophesied. Secondly, the possibility of element appearance in the sequence is equal to another element in the equivalent sequence. Thirdly, the initial value is compulsory needed to regenerate the same sequence [4]. Many algorithms for random number generating was employed, such as genetic algorithms, neural networks and shift registers [5-8]. The proposed method in the current paper overwhelms the problems in furthermost earlier works; these problems comprise a large size for initial seed and the capability to re-generate the equivalent set when the identical initial values are fed to the system [9-12]. The proposed technique generates a set of pseudo random numbers by implementing the theory Futoshiki puzzle. The Futoshiki puzzle like several common logic puzzles, this one invents from Japan. It means "not equal" and is an indication to greater than and less than signs that are used as the typical feature of this puzzle. The produced sets of random numbers are examined with a statistical test suite and security analysis is performed to verify that such sequence meets the specification of random numbers and these statistical tests indicate it is suitable for using as an encryption key or not. The obtained results show that the proposed method could successfully generate pseudo random numbers with good statistics and security properties and high linear complexity. This paper is ordered into some sections. Section 2 briefly examines previous works on random number generators. Section 3 provides a detailed explanation of Futoshiki puzzle. Section 4 provides the general definition of the solution of Futoshiki puzzle and the proposed system. Section 5 describes the proposed key expansion method. Section 6 discusses the details of experimental results. Finally, Section 7 presents the conclusions.

2. **Related Work.** Numerous algorithms for random numbers generating have been proposed in recent years. Utmost of the methods are employed in software rather than hardware [13, 14]. These approaches provide a supreme periodic of random sequence and higher quantity level of randomness while adhering to established statistical standard tests by applying a seeding mechanism. To generate random numbers, earlier generators use the prime number theory [15], initial sources come from audio and video [10], mouse motion [9], chaotic map [16, 17], human biometric features [18] and even the contents of input/output buffers [19]. Nevertheless, most of these random generators required large initial values to start [10, 20]. This requirement is unacceptable if the generator is used to produce a cryptography key [21]. In key management for cryptographic purpose, in which these keys need to be switched between sender and recipient, a large initial key will require additional efforts or resources to be sent; this requirement slows down delivery and consider as a breach of the system security. Furthermore, most of the previous generators are 're-generate able' [9, 16]. When the equivalent initial value feed up to the system, hypothetically, the same sequence of random numbers must be reproduced. Hence, generators that depend on a physical initial seed (i.e. Temperature and wind speed) would be incapable of regenerating the identical key sequence because the similar initial value is intolerable to obtain. Additionally, many of the earlier generators failures in randomness statistical tests because the generated sequences are unsatisfactorily random [22]. To overcome these issues and obtain a good random sequence, this paper proposes the use of the Futoshiki puzzle, which is described in detail in the next sections.

3. **Futoshiki Puzzle.** This puzzle is a board-based puzzle game, also known as the Unequal. It is played on a square board having a given fixed size as described in Figure

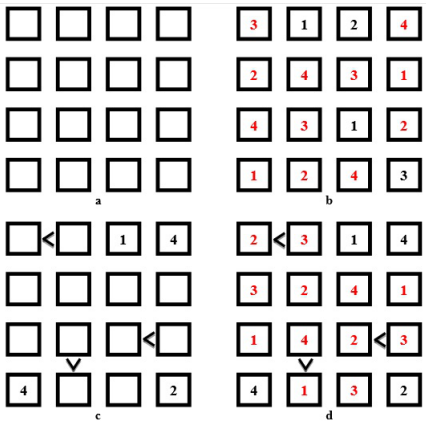

FIGURE 1. Futoshiki puzzle examples (a) Without signs conditions. (b) Answer of (a) puzzle. (c) With signs conditions. (d) Answer of (b) puzzle.

1-a, where the number of columns is equal to the number of rows $(n \times n)$,where $n \in Z$. The game aim is to find the allocated numbers inside the boards empty cells; each cell is filled with a digit between 1 and the boards size Futo[i,j] in (1,. . . ,n). On each row and columns, each digit appears exactly one time as illustrate in Figure 1-b. At the launching of the game some digits might be discovered, these digits, consider as an initial key where sent to the other parties to regenerate the equivalent sequence to decrypt the secret massage [23]. The board might also contain some inequalities between the board cells, some mathematical signs are used to give some challenges to this game such as greater than and less than signs (¡,¿). On the grid were located between the cells as demonstrate in Figure 1-c; these inequalities should be followed and used as hints to determine the remaining secreted digits. As well as each Futoshiki puzzle having one solution, where can get the benefits of the ability of regeneration the same sequence more than one time, these abilities are so important for the security system designers were used as an encryption key and send only the initial key better than sending the all encryption keys to other parties.

4. **Futoshiki Puzzle Solution.** The Futoshiki puzzle, also known as (More or Less) or Unequal is a logic puzzles origin from Japan. The name means inequality. The objective is to fill the square grid board with numbers so that each row and column is filled with unique digits, where each digit must be appearing exactly one time. The (¡,¿) symbols provide additional suspicions that show you the connection between adjacent squares these inequalities between square cells must be respected. There are several algorithms to answer such puzzle such as neural networks [24], genetic algorithms [25], recursion [26] and so on. The solution of Futoshiki puzzle from a mathematical point of view can be expressed as the following, where L a set of the inequality signs such that each inequality sign is located between two adjacent cells. Let Q_L be the set of all the ordered pairs of two adjacent cells such that at least one of them is empty in L, that is,

$$Q_L = |((i,j),(\bar{i},\bar{j})) \in n^2 \times n^2|(1)$$

Where (i,j) and $(\bar{i},\bar{j})$ are adjacent, and at least one of (i,j) and $(\bar{i},\bar{j})$ is empty in L.

Also, suppose a subset Q of Q_L, a sign set when $(i,j) and (\bar{i},\bar{j}) \in Q$ implies $(\bar{i},\bar{j}) and (i,j) not \in Q$. Each $(i,j) and (\bar{i},\bar{j}) \in Q$ represents a "smaller than" inequality sign such that (i,j) should be assigned a smaller integer than $(\bar{i},\bar{j})$.

For every pair $(i,j) and (\bar{i},\bar{j})$ of adjacent cells in Q, either

```
Input: Initial key as (row, col) and the board size (n)
Output: Set of numbers that fill the Futoshiki board
  Begin
    // Initial key position
      row ← key the x coordinate;
      col ← key the y coordinate;
      n ← board size (Futoshiki Matrix Dimension);
      Counter ← 0;
    // Start filling process
      Create a plain matrix Futoshiki with size (n × n)
      For row =1 → n; increment by 1 do
        For col =1 → n; increment by 1 do
          While (Counter <= n) do
            Set the Counter number to the board (row, col, Counter);
            If the conditions satisfy, Then
              Futoshiki [row, col] ← Counter
            End if;
            Counter ← increment Counter by 1;
          End While;
        End for;
      End for;
  End.
```

FIGURE 2. Futoshiki puzzle solution.

$$(A \oplus L)_{ij} = 0 or (A \oplus L)_{\overline{(ij)}} = 0 (2)$$

or,

$$(A \oplus L)_{ij} < 0 or (A \oplus L)_{\overline{(ij)}} (3)$$

Note that Q contains at most one inequality sign between any two adjacent cells, and, it contains no inequality sign between two adjacent cells such that both cells are given integers by L; such an inequality sign would be redundant in the puzzle. The current study chooses a recursion procedure to solve the Futoshiki puzzle as shown in Algorithm-1, because the recursion reduces the program code statements and make it so easy to write and understood as well as reduce the debug code time and reduces time complexity [26].

The generated numbers are considered as an encryption and decryption keys of $(n \times n)$ dimension. There are more conditions needs to employ for filling and completing the entire Futoshiki board such as those who determine the cell its allocated or not, or that set of conditions used to step back for one movement and start again when reached to the dead end. This kind of generators is very useful when trying to generate random numbers used as an encryption key especially used in image encryption. Where the image is subdivided into equal blocks where each block can use a generated set comes from the solving of Futoshiki puzzle, therefore, the image size blocks must equal in dimension with the Futoshiki board size. While the generated keys are 64 random number comes from (8×8) Futoshiki board size, but the image size is bigger than 64 pixels. Therefore, additional method is required for expanding the generated sequence that satisfies the requirements of the plain image size for encryption and decryption. To overcome this limitation, a new method for key expansion is introduced as described hereunder.

5. **Key Expansion.** As above-mentioned, the generated key by using Futoshiki puzzle must fulfill the requirement of the plain image in term of image size to use as an encryption key. Therefore, the suggested method needs to be developed for enlarging the key size achieved by using the Futoshiki puzzle. To make the encryption key fit to plain image size is performed as described below. The proposed method for expanding the generated key size is described in Figure 2, where after the initial value was set to the Futoshiki puzzle Figure 2-a, the puzzle will start fill the missed squares to fill the entire board as shown in

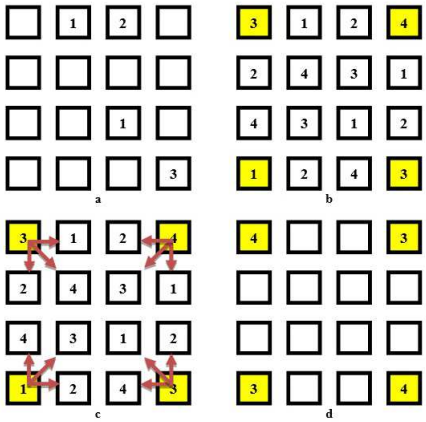

FIGURE 3. The key expansion process (a) Initial stage. (b) Answer of (a) puzzle. (b) Testing the neighbour corner for the biggest number. (d) Initial values to Futoshiki puzzle for generate the nest stage.

Figure 2-b, to expanding the generated key propose to use the corners value of the board (3, 4, 1, 3). The triple neighbour's method is supposed and used to set the initial to the next stage of generating. Each corner has three neighbour's numbers as descriptive in Figure 2-c, select the bigger value to be set as an initial for the next stage to generate (4×4) random number. At some time, the bigger number may be repeated or selected more than one time in the same board where that will break the Futoshiki rules in the appearance of digit once in each row and columns. To overcome this problem, we will choose the next smaller digit to the biggest one. The first corner in Figure 2-c, was in position (1, 1) and contain numbers 3, the boundary to that digit are (1, 4 and 2) the biggest number is 4, then the initial value of the position (1, 1) is 4. For the position (1, 4) the neighbour of these digits is (2, 3 and 1), where the largest number is 3 and consider as an initial to the position (1, 4). The third corner is (4, 1) and contain 1, as well as the neighbours are (4, 3 and 2) and the largest number is 4 but cannot use 4 to initial the position (4, 1) because the Futoshiki rule not allowed to use the same number in the row and column, therefore, select the smallest number then the bigger one, where initiate the position (4, 1) by 3. Finally, the last position is (4, 4) and the neighbours is (4, 1 and 2), the largest number is 4 and consider as an initial to the position (4, 4), the second stage of generating is describing in Figure 2-d, the expansion process of testing the corner neighbours and select the bigger to use as an initial to generate the new set of random number will continue until the desire size was reached.

In term of board size the baggier board of Futoshiki puzzle need more than four digits to start the next stage of generating [1], the number of initial digits are be double when the board size was doubled. Therefore, when the size was (8×8) need to start with the 8 digits as a minimum number of initial keys and the selection stage for initial keys also will be doubled, as shown in Figure 3-a and b, while the Futoshiki puzzle rule was maintained.

6. **Result and Discussion.** The encryption key being the representation of specific information of any cryptosystem needs to work successfully. Thus, it is important to investigate those keys used in the encryption process. This section explains the tests that are used to evaluate the qualities of the produced pseudo random sequence, in other words, this section testifies the approaches that are used to generate a random number stream. The ideal randomness of the generated stream is checked using various statistical test suites such as DIEHARD test suite [28], TestU01 test suite [29] and the most popular randomness test produce in the special publication of the "National Institute of Standards

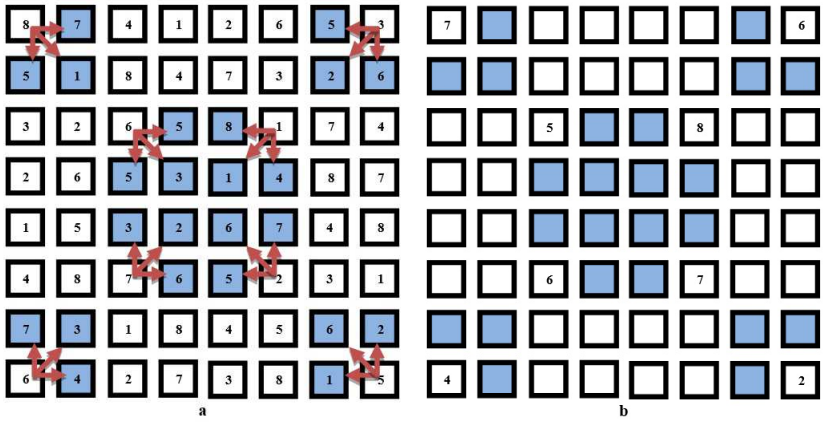

FIGURE 4. Triple neighbour's expansion stage (a) when n=8, (b) The initial key for the next stage contains 8 digits.

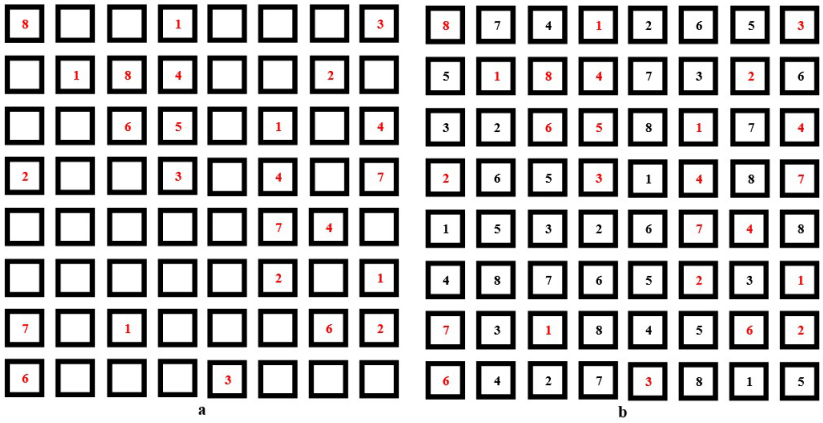

FIGURE 5. Practical example for Futoshiki puzzle with n=8 (a) Initial state. (b) First stage generator.

and Technology" (NIST). Commonly, this suite includes 14 tests, which verifies assorted types of possible non-randomness that may occur in the random stream [30]. A sequence considers as random if the P-value is larger than 0.001 [31]. The practical example was stated in Figure 3, where the puzzle size was n=8 and the initial state visualized in Figure 4-a, the first step of generating figured in Figure 4-b.

The first stage expansion started in the Figure 5-a, where eight digits were selected to consider as an initial to the next stage of generating, where the next stage figured in Figure 5-b.

To generate a block of (256×256) need to generate and expand the proposed method 8192 times. These statistical tests performed on the proposed key generator for three distinct sizes such as (256×256), (512×512) and (1024×1024) pixels. The achieved results are summarized in Table-1, where these results are got by applying the NIST tests on a random number produced by the Futoshiki puzzle.

These results tabulated in Table 1, obviously show that the developed key generator and expander reveal the perfect level of randomness and successfully pass almost random statistical tests. Moreover, the P-value (the values obtained from a statistical test in columns 2, 3 and 4 in Table 1) is greater than the threshold randomness value as mentioned earlier. It is indicated that the sequence is truly a random one with 90 confidence, while the P value is greater than 0.01, but there is some observation on the P-value of frequency

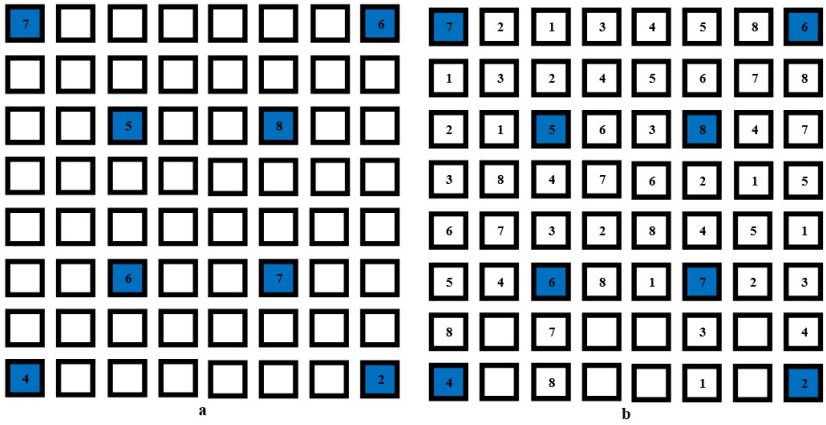

FIGURE 6. Practical example for Futoshiki puzzle expansion with n=8 (a) Initial state. (b) First stage generator.

TABLE 1. The obtained results of the NIST statistical test for three varied sizes with the same initial value.

Key Size			
Test Name	**P-Value** 256×256	**P-Value** 512×512	**P-Value** 1024×1024
Approximate Entropy	0.5024	0.2076	0.1796
Block Frequency	0.6892	0.6901	0.6883
Cumulative Sums	0.7936	0.7895	0.8105
Discrete Fourier Transform	0.4839	0.3109	0.3021
Frequency	0.2954	0.2954	0.2954
Linear Complexity	0.5491	0.4508	0.5709
Longest Run	0.5531	0.5495	0.5597
Non-Overlapping Template	0.6832	0.7094	0.7108
Overlapping Template	0.6004	0.6097	0.6031
Random Excursions	0.4807	0.4292	0.5594
Rank	0.3034	0.3389	0.7603
Runs	0.7972	0.8127	0.7072
Serial	0.3024	0.3904	0.3502
Universal Statistical	0.6809	0.5493	0.6305

test, block frequency and linear complexity. It is equal in value or near each other in all generated sequence because, the proposed Futoshiki puzzle generates the same set of random numbers with various positions each time, that is, lead to the same number of ones and zeros that was tested during the NIST test suite, therefore the P-value is equal for varied sizes. To ensure that the proposed Futoshiki puzzle generator can perform better than others in term of quality, where consider the results of NIST test to compare with other works [2] and [3], as described previously in literature part. These papers are used the random number as an encryption key where used image to apply the encryption algorithms, but they generate random number and test them by using NIST statistical test, the comparison is visualized in Figure 6.

Figure 6 describe the comparison between the NIST tests results of the proposed Futoshiki puzzle generator and the tests results come from other mentioned generators, that is obviously illustrate the proposed method look like best in acting as a random number

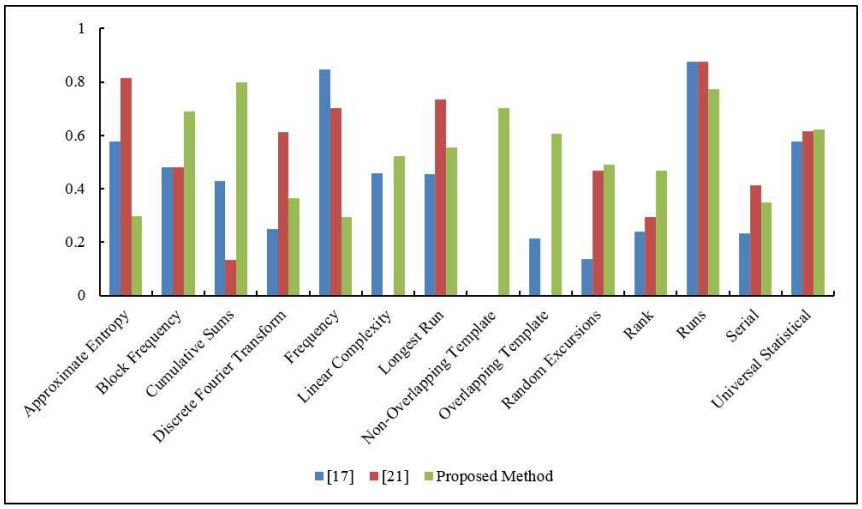

FIGURE 7. Comparing of NIST statistical tests results of the proposed key generator with other works.

generator than the other work and can be considered as a second proven of the excellent randomness properties of the proposed method, furthermore, the generated random numbers can be used as an encryption key epically in case of image encryption because the generated key fit to the image dimension.

7. **Conclusion.** In this paper, proposed an innovative random number generator. This generator employs the Futoshiki puzzle to generate a set of numbers under certain conditions of one occurrence in a separate row and column in the board under the size and. Each time Futoshiki puzzle will generate a set of $(n \times n)$ random number, in case of need more random numbers, therefore, required to expand the generated random number by using the proposed triple neighbour's method. The proposed system was analysed by both theoretical and experimental methods. The generated sets of random numbers are subjected to the NIST statistical tests for checking the level of randomness and remarkably passes most of these tests. The obtained results come from the NIST tests show an excellent performance in term of random number also, compared the performance with some other recently proposed random number generator. Based on these advantages, the proposed generator can be used as a random number generator to generate an encryption key used especially in image encryption systems.

Acknowledgment. The author would like to thank Mustansiriyah University (www.uomustansiriyah.edu.iq) Baghdad-Iraq for its support in the present work.

REFERENCES

[1] A. S. Mahmood, and M.S. M. Rahim, State of the Art of Image Ciphering: A Review. *International Journal of Computer Science Issues (IJCSI),* vol. 11, no. 2, 2014.

[2] X. Tong, et al., An Image Encryption Scheme Based on Hyperchaotic Rabinovich and Exponential Chaos Maps. *Entropy*, vol. 17, no. 1, pp. 181-196, 2015.

[3] K. Li, Performance analysis and evaluation of random walk algorithms on wireless networks. *International Journal of Foundations of Computer Science*, vol. 23, no. 04, pp. 779-802, 2012.

[4] M. T. Rahman, et al. TI-TRNG: Technology Independent True Random Number Generator. *in Proceedings of the The 51st Annual Design Automation Conference on Design Automation Conference.* 2014. ACM.

[5] R.F. Toso, and M.G. Resende, A C++ application programming interface for biased random-key genetic algorithms. *Optimization Methods and Software*, 2014(ahead-of-print, pp. 1-13.

[6] T. Hrdy, M. Prazan, and pp. Holoubek, Random number generator, 2014, US Patent, vol. 20, no. 140,324,934.
[7] D. E. ubrova, M. Naslund, and G. Selander. Secure and efficient LBIST for feedback shift register-based cryptographic systems. *in Test Symposium (ETS), 2014 19th IEEE European. 2014.* Paderborn: IEEE.
[8] H. A. Shenoy, R. Srikanth, and T. Srinivas, Efficient quantum random number generation using quantum indistinguishability. *Fluctuation and Noise Letters*, 2013, vol. 12, no. 04, pp. 1350020.
[9] W. Xingyuan, Q. Xue, and T. Lin, A novel true random number generator based on mouse movement and a one-dimensional chaotic mapp. *Mathematical Problems in Engineering*, 2012. 2012.
[10] I.-T. Chen, Random numbers generated from audio and video sources. Mathematical problems in engineering, 2013. 2013.
[11] Zhao, T., et al., Image encryption using fingerprint as key based on phase retrieval algorithm and public key cryptography. *Optics and Lasers in Engineering*, 2015, vol. 72: pp. 12-17.
[12] E. Anceaume, et al., Dependability evaluation of cluster-based distributed systems. *International Journal of Foundations of Computer Science*, vol. 22, no. 05, pp. 1123-1142, 2011.
[13] T. E. Tkacik, A hardware random number generator. in International Workshop on Cryptographic hardware and embedded systems. 2002. Springer.
[14] C. De Schryver, et al., A hardware efficient random number generator for nonuniform distributions with arbitrary precision. *International Journal of Reconfigurable Computing.* 2012, pp. 12, 2012.
[15] R. Kumar, and M. Dhiman, Secured Image Transmission Using a Novel Neural Network Approach and Secret Image Sharing Technique. International Journal of Signal Processing, *Image Processing and Pattern Recognition*, vol. 8, no. 1, pp. 161-192, 2015.
[16] H. Hu, L. Liu, and N. Ding, Pseudorandom sequence generator based on the Chen chaotic system. *Computer Physics Communications,* 2013, vol. 184, no. 3, pp. 765-768.
[17] L. Liu, and S. Miao, A new simple one-dimensional chaotic map and its application for image encryption. Multimedia Tools and Applications, 2018, pp. 1-18.
[18] H. Taherdoost, et al., Definitions and Criteria of CIA Security Triangle in Electronic Voting System. International Journal of *Advanced Computer Science and Information Technology (IJACSIT),* vol,1, pp. 14-24, 2013.
[19] Pardo, J. L. G., Introduction to Cryptography with Maple. 2012: Springer Science and Business Media.
[20] Y. Deng, , et al., Analysis and Design of Digital Chaotic Systems With Desirable Performance via Feedback Control. Systems, Man, and Cybernetics Society, 2015. PP(99).
[21] B. Yang, and X. Liao, A new color image encryption scheme based on logistic map over the finite field ZN. Multimedia Tools and Applications, 2018, pp. 1-19.
[22] pp. J. Pashley, , On generating random sequences. *A Handbook for Data Analysis in the Behaviorial Sciences:* vol. 1: Methodological Issues Volume 2: Statistical Issues, 2014, pp. 395.
[23] Sutanto, H. Combination of BFS and Brute Force Algorithm Implementation in Futoshiki Puzzle Game. *in Seminar Nasional Aplikasi Teknologi Informasi (SNATI).* 2013.
[24] Y. I. A. Osman, , et al., Experimental Study of the Cloud Influence on PV Grid Connected System. *Smart Grid and Renewable Energy,* vol. 9, no. 01, pp. 1, 2018.
[25] Abdel-Raouf, O., I. El-Henawy, and M. Abdel-Baset, A novel hybrid flower pollination algorithm with chaotic harmony search for solving sudoku puzzles. *International Journal of Modern Education and Computer Science*, vol. 6, no. 3, pp. 38, 2014.
[26] S. Blank, M. Azmoodeh, and N.F. Alamo-Pritchard, S*ystem and method for generating and using solvable puzzle forms.* 2017, Google Patents.
[27] K. Haraguchi,, The number of inequality signs in the design of Futoshiki puzzle. *Journal of information processing*, , vol. 21, no. 1, pp. 26-32.
[28] G. Marsaglia, Diehard, A battery of tests of randomness., httpp
stat. fsu. edu/pub/diehard.[Accessed September 15th, 2013], 1996.
[29] R. Simard, TestU01: AC library for empirical testing of random number generators. *in ACM Transactions on Mathematical Software.* 2007.
[30] A. Rukhin, , et al., Statistical test suite for random and pseudorandom number generators for cryptographic applications, *NIST special publication.* 2010.
[31] A. Rukhin, , et al., NIST Special Publication 800-22 Revision 1a: A statistical test suite for random and pseudorandom number generators for cryptographic applications.(2010). *Date of access,* vol. 1, no. 03, 2013.