Journal of Theoretical and Applied Information Technology
15th April 2017. Vol.95. No 7

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

GENERATING AND EXPANDING OF AN ENCRYPTION KEY BASED ON KNIGHT TOUR PROBLEM

[1, 2]ALI SHAKIR MAHMOOD, [1, 3]MOHD SHAFRY MOHD RAHIM

[1]Faculty of Computing, University Technology Malaysia, Johor Bahru, Malaysia

[2]Department of Computer Science, College of Education, AL-Mustansiriyah University, Baghdad, Iraq

[3]UTM-IRDA Digital Media Center, University Technology Malaysia, Johor Bahru, Malaysia

ABSTRACT

The encryption key considers as a vital part in designing of a cryptosystem. Whereas these keys must be random as can as possible. The ability to regenerate the same sequence with small initial value is still a major problem that faces the designer of encryption key system. The current paper designs a new method of random number generator with the ability to expand the generated encryption key to fit the proper image size. The knight tour problem was employed as a random number generator and used for encryption key expansion. The expansion process contains two steps, first one generate a random number with (64 × 64) key size and the second step consider the boundary numbers as from the previous step to initiate the knight tour as a second time, the second step continue until the image size was reached. Generated random numbers acquired from the knight tour problem have been subjected to the NIST 800.22 statistical test and successfully passed all statistical tests without requiring any additional processing. Per these results, it has been proved that the proposed system meets the security requirement and can be used in cryptographic applications. Furthermore, the knight tour generator provides a small initial value with the ability to regenerate the same sequence when feed up with the same initial value.

Keywords: *Random Number, Encryption Key, Knight Tour, Key Expansion, (NIST) Randomness Tests.*

1. INTRODUCTION

Random numbers are extensively used for several applications, such as encryption key, simulating and modeling, numerical analysis, as well as for selecting random samples from larger data sets [1]. Earlier, several methods are proposed for generating a random encryption key, that generating based on mathematical computational or physical activities [2]. Irrespective of the procedure, it is required that the generated encryption keys must be random with unpredictable sequences for the next occurrences, the appearance probability of any element in the sequence is equal to other elements in the same sequence, the random sequence cannot be reproduced unless the same initial value is used [3, 4]. The other properties of an encryption key generator are related to the portability, efficiency, less memory size, homogeneity and disjoint subsequence without any correlation between

generated sequences of different initial values [4, 5].

These generators have many drawbacks with encryption including the requirement of long initial seed like a series of parameters or piece of the image to begin their operation [6, 7]. The presence of small initial seed is advantageous because it can easily to save and delivered to the other parties. Making the encryption key small and easy transfer to the recipients is considered as a major problem in key management [8, 9]. The other limitation is related to the ability for regenerating the same sequence from the alike initial values. Most of the generators are unable to regenerate the same sequence because of their dependence on a physical action such as wind speed or system clock, keystrokes, and mouse position on the screen etc. [10]. To surmount such limitations, of randomness, small initial key and the ability to generate the same sequence, the knight tour (KT) generator is proposed, which is overcomes the problems in most previous works; these problems include a low level of randomness, large initial seed and incapacity to regenerate the same sequence when feed up with the same initial seed [11, 12].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The proposed method generates a pseudo random number sequence by implementing the knight tour problem and expands the generated key to fit the proper plain image size. The knight tour problem is a mathematical problem involving the identification of a sequence of moves of a knight (on a chessboard) based on the knight movement rule in a chess game. Followed by two steps of key expansion. The produced sequence is tested with a NIST statistical test suite and compared with other works to verify that such sequence meets the specification of encryption keys. The obtained results indicate that the proposed method could successfully generate random encryption key with good statistical properties and high linear complexity.

2. RELATED WORK

Several methods for generating an encryption key have been proposed in recent years. Most of this methods are implemented in hardware rather than software [13]. These methods provide a maximum periodic of random sequence and high throughput rate while adhering to established statistical standards by applying a seeding mechanism. To generate encryption keys, previous generator used prime number theory [14], audio and video sources [11], mouse motion and one-dimensional chaotic map [10], biometric feature of human iris [15] and even the contents of input/output buffers [16].

However, one of random encryption key generator that depend on hardware was introduced in [17], that hardware used a discrete Boolean chaotic oscillator printed on circuit boards combined with Lyapunov exponents. As well as the CMOS process was used to handle this generator, which consists of a core chaotic oscillator and a source follower buffer. The NIST statistical test [18], a standard way to validate the properties of generated sequences for cryptographic applications, is performed with the measured data. The resulting random bit sequence passes the widely accepted statistical tests used to evaluate cryptographic random number generators.

On the other hand, there is several methods based on software implementation with the ability of overcoming an important issue in key management such as the capability of regenerate the same key sequence when the same initial values were supplied and the initial key size it was important in designing of encryption key generator. Several researchers used the knight tour problem to satisfy this requirement, where knight tour used to generalize encryption key space by using mathematical method. Analytic and experimental results show that the algorithm based on outlet number is more effective for seeking the solution of generalized knight [19].

The mixing between SCAN method and knight tour are introduced in [20] work, where the generated path used to per mutate the image pixels. The encryption key generated from the amplitude values of a chosen noise audio file. The effectiveness of proposed method is validated by several statistical analyses. High pixel scrambling can be achieved by considering higher dimension chessboard and white noise audio signals. The proposed method is suitable for real time data transmission due to fast of encryption.

This work introduces a new approach in the field of image encryption by extending the generated key based on knight tour problem. The designed method used two privet encryption keys instead of one key for enhances image encryption method. Key generation is far more easily utilized in here, used for real time encryption like online activities [21, 22]. There is another work also used the knight tour to generate an encryption key combined with slip encryption filter. Where the center of the slip encryption filter template matrix move along with the knight tour slip matrix to do the confuse operation like the image filter, consequently the image was encrypt [23]. The gray images also encrypted by knight tour problem conjunct with another external key, plain image was partitioned to four sub-images where each sub employ by knight tour to scramble and substitute the sub pixels with starting point as a secret key [24].

Moreover, some of the previous generators fail in randomness statistical tests because the generated sequences are insufficiently random or the expansion procedure suffer from high level of redundancy because using of repetition patterns. To overcome such issues and obtain a good encryption key, this paper proposes the use of the knight tour problem to generate and expand an encryption key, that is provide a high level of randomness, need small initial value and regeneratable, as well as the expansion method easy for implementation and no need for farther expansion algorithm but use the same knight tour, which is described in detail in the next sections.

3. KNIGHT TOUR PROBLEM

Historically, the term Knight Tour Problem was first coined in the ninth century in a Sanskrit poem named as Kavyalankara written by Indian poet

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Rudrata. A model for knight's tour was also found on a half-board which was offered in various forms. As was frequently the practice in lavish Sanskrit poetry, the syllabic examples of this poem clarify a totally distinctive theme, for this situation an open knight's tour on a half-chessboard [25].

It is needless to mention that the KT is an arithmetic problem connected to the movement of a knight on a chessboard. The knight is placed on a blank board and paddles following the chess game's knight movement rules, where every square need be visited just once [26], as demonstrated in Figure-1.

According to the knight location on the chessboard, the knight can classify to five different movement sets as illustrated pictorially. As shown in Figure 1-(a), the white knight can be moved to two locations in case it is at the chessboard corner. When the knight's position is closer to the center it possesses more choices to move as shown in red and green knight where that can move to eight and sex respectively. As well as Figure 1-(b) illustrates the articulation of all the possible moves in a single illustration.

The process of solving knight's tour is a widespread problem known to computer scientists. Several strategies are adopted for solving knight tour problem including the divide-conquer method [27], intelligence searching-backtracking algorithm [28], depth-first search algorithm [29], heuristic approach with a minimal outlet [30], genetic algorithms [31], and ant colony optimization algorithm [32] to cite a few.

4. SOLUTION OF THE KNIGHT TOUR PROBLEM

The filling process of the standard size of chessboard can be done based on sets of eight boundary conditions are introduced for solving KT problem, where the knight path is determined during the chessboard. The detail description of knight tour key generating processes demonstrates in Figure 2. Firstly, to initiate the knight must be choose the initial position for the knight on the chessboard as shown in Figure 2-(a), where this represent the first position. The knight movement sequences are stored in a key matrix of size (8 × 8) which follows the standard chess game rules as illustrate in Figure 2-(d). The generated knight moves are considered as an encryption and decryption keys of (8 × 8) dimension, based on the standard chessboard.

The movements of the knight rock can be computed based on eight conditions, where each condition is related to certain direction for the next move. Algorithm 1 explains the pseudo-code for knight movements for filling the chessboard that leads to the KT solution and produced the (8 × 8) array filled with the knight rock movements.

The conditions stated in Algorithm 1 concerned only with the knight moves and to determine the next movement for the knight rock. There are more conditions needs to employed for filling and completing the entire chess board such as those who determine the cell its allocated or not, or that sets of conditions used to step back for one movement and start again when reached to the dead end on a standard chessboard.

Although it can generate 64 pixels to be used later as encryption key but the image size is bigger than 64 pixels. Thus, another method is required for expanding the knight sequence that satisfies the requirements of the plain image for encryption and decryption. To overcome this limitation, a new method for key expansion is introduced as described hereunder.

5. KEY EXPANSION

As aforementioned, the key generated via KTG must fulfil the requirement of plain image size to use as an encryption key. Thus, a method needs to be developed for enlarging the key size achieved by using the KTG. To make the encryption key fit to plain image size is performed in two stages as described below.

5.1 First Expansion Stage

The proposed method for expand the key size describes in Figure 3, where the first expansion stage that generates (64 × 64) keys. The first enlarge stage starts with using each cell in knight tour array that generated previously as an initial position to generate a new (8 × 8) encryption key. In other words, each generated key is considered as an initiation to the KT for the second time to generate a set of keys with size (64 × 64).

The obtained encryption key size from the first expansion stage is not satisfy the dimension of the traditional plain image size, where the proposed method needs an encryption keys fit to the plain image size. Therefore, called the need to develop another expansion method that expand an encryption keys fit to the image dimensions, as explains in the next sub-section named as second expansion stage. Also, the key length must be long

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

enough so that an attacker cannot try all possible combinations.

5.2 Second Expansion Stage

The second phase of key expansion process is illustrated in Figure 4. This stage used the outer boundaries of (64 × 64) keys generated from the previous stage. Several cells gained from the outer boundaries are 64 for each side with a total of 252 cells of four sides.

As demonstrate in Figure 4, there are several cells in the boundaries are considered as an initial value to the KT which generated a new (8 × 8) keys and then arranged them sequentially to become an (80 × 80) array of keys. The cells located in the corner position such as (1, 1), (1, 64), (64, 64) and (64, 1) are used to generate a (8 × 8) key matrix to fill the gap in the corner. Then, the next cell is chosen as an initial one to generate next (8 × 8) key matrix. Thus, the cells are gradually increased for different directions for each boundary row and column.

The second stage is continued to appropriately consider the cells in the outer boundaries and the KT is initiated by these values to generate (8 × 8) keys. This increased the key dimension by eight cells in four directions at each time. This process is repeated until the key size is matched to the plain image size. Figure 5 demonstrates the overall process of key generation based on KT, each color represents a certain process.

As noted before the algorithm used for expansion it is the same algorithm used for generating, where get the benefit of easy to implement and no need for further algorithms.

For instance, and based on the Figure 5, when the plain image size is (256 × 256) the second step of expansion process is repeated twelve times to generate the proper key size. These generated keys are now ready for used in image encryption.

6. RESULT AND DISCUSSION

The encryption key being the representation of specific information of any cryptosystem needs to operate successfully. Thus, it is important to analyze those keys used to encrypt the image. This section explains the procedures that are used to analyses the qualities of the produced pseudo random sequence, in other words, the approaches that are used to testify the randomness of the generated random number stream.

The ideal randomness of the generated stream is checked using various statistical tests suite following the special publication of National Institute of Standards and Technology (NIST). Commonly, this suite includes 14 tests, which verifies assorted types of possible non-randomness that may occur in the random stream. A sequence consider as random if the P-value is larger than 0.001 [18]. Table 1 summarizes the results obtained using these statistical tests performed on the proposed key generator with initial value in the location of (1, 1) for three different size such as (256 × 256), (512 × 512) and (1024 × 1024) pixels.

These results clearly indicate that the developed key generator and expander reveal perfect randomness and successfully pass almost random statistical tests. Moreover, the P-value (the values obtained from statistical test in columns 2, 3 and 4 in Table 1) is greater than threshold randomness value as mentioned earlier. It is implying that the sequence is truly a random one with 90% confidence, but there is some observation in the P-value of frequency test, block frequency and linear complexity. It is equal or near from each other in all generated sequence because, the knight tour generates the same set of random numbers with different positions each time, that is lead to same number of zeros and ones initiate to frequency test, therefore the P-value are equal for different size.

Thus, the generated keys are suitable for used in cryptography applications especially in image encryption. It is noticeable that the KTG generator is approved by all the NIST statistical tests. For more understanding to the numbers tabulated in Table 1, the graphical representation of the third column values (because that image size is very familiar to use during sending and receiving images) compare with the NIST values get from CMOS generator [17] and shuffling phase [33] in Figure 6.

The visualization stated in Figurer 6, that compare between the NIST tests results of the proposed KTG with the tests results come from CMOS generator and shuffle map, that is clearly show the proposed method look like best acting than the other work and can be considered as a second proven of the excellent abilities 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

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

This work addressed the use of knight tour problem as a random number generator and expander to fit the proper image size. The proposed method overcomes the current challenges in key generators; these challenges are small initial size, unpredictability and the ability to regenerate the same sequence, when the same initial value was initiated and the initial size of the seed value. In addition to the previous reasons the level of randomness is still an essential characteristic when designing an encryption key generator. The true random numbers of encryption key make a cryptosystem immune against several attacks like brute force attack. The experimental results of proposed generator and expander provide an excellent level of randomness based on the NIST randomness tests suite. Where more than 90% of randomness tests indicate that the proposed generator and expander acting like random number generator and most of tests result passes the randomness threshold value. The comparison of a proposed generator and expander with another similar work shows that the proposed method has better performance in terms of randomness and very flexible of expanding because it expands 8 pixels in each direction at a time. Therefore, the proposed method is very suitable for used in cryptography systems. In the future, we hope to use other mathematical problems like multi knight or Sudoku game algorithm that can also be used as a random number generator.

ACKNOWLEDGMENT

The authors are grateful to the Ministry of Higher Education and Scientific Research, Iraq, for providing a Ph.D. sponsorship. Furthermore, I would like also to thank Faculty of Computing, School of Postgraduate and Research Management Centre of Universiti Tenologi Malaysia under GUP Q.J130000.2428.02G28.

REFERENCES

[1] Tong, X., et al., *An Image Encryption Scheme Based on Hyperchaotic Rabinovich and Exponential Chaos Maps.* Entropy, 2015. 17(1): p. 181-196.

[2] Takagi, T., *Introduction to Public-Key Cryptography*, in A Mathematical Approach to Research Problems of Science and Technology. 2014, Springer. p. 35-45.

[3] Rahman, M.T., 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.

[4] SaberiKamarposhti, M., et al., *Using 3-cell chaotic map for image encryption based on biological operations*. Nonlinear Dynamics, 2014. 75(3): p. 407-416.

[5] Brent, R.P., *Some integer factorization algorithms using elliptic curves*. arXiv preprint arXiv:1004.3366, 2010.

[6] Sui, L., et al., *Asymmetric double-image encryption based on cascaded discrete fractional random transform and logistic maps*. Optics express, 2014. 22(9): p. 10605-10621.

[7] Rakhmadi, A., et al., *Connected component labeling using components neighbors-scan labeling approach*. Journal of Computer Science, 2010. 6(10): p. 1099.

[8] Kremer, S., R. Künnemann, and G. Steel, *Universally composable key-management*, in Computer Security–ESORICS 2013. 2013, Springer. p. 327-344.

[9] Radwan, A.G., S.H. AbdElHaleem, and S.K. Abd-El-Hafiz, *Symmetric encryption algorithms using chaotic and non-chaotic generators: a review*. Journal of advanced research, 2016. 7(2): p. 193-208.

[10] Xingyuan, W., Q. Xue, and T. Lin, *A novel true random number generator based on mouse movement and a one-dimensional chaotic map*. Mathematical Problems in Engineering, 2012. 2012.

[11] Chen, I.-T., *Random numbers generated from audio and video sources*. Mathematical problems in engineering, 2013. 2013.

[12] Sharifara, A., M.S.M. Rahim, and Y. Anisi. *A general review of human face detection including a study of neural networks and Haar feature-based cascade classifier in face detection*. in Biometrics and Security Technologies (ISBAST), 2014 International Symposium on. 2014. IEEE.

[13] De Schryver, C., et al., *A hardware efficient random number generator for nonuniform distributions with arbitrary precision*. International Journal of Reconfigurable Computing, 2012. 2012: p. 12.

[14] Kumar, R. 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, 2015. 8(1): p. 161-192.

[15] Taherdoost, H., et al., *Definitions and Criteria of CIA Security Triangle in Electronic Voting System*. International Journal of Advanced Computer Science and Information Technology (IJACSIT) Vol, 2013. 1: p. 14-24.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[16] Pardo, J.L.G., *Introduction to Cryptography with Maple*. 2012: Springer Science & Business Media.

[17] Park, M., J.C. Rodgers, and D.P. Lathrop, *True random number generation using CMOS Boolean chaotic oscillator*. Microelectronics Journal, 2015. 46(12): p. 1364-1370.

[18] 18. Rukhin, A., 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, 2013. 1(03).

[19] Bai, S., et al. *Generalized knight's tour problem and its solutions algorithm*. in 2006 International Conference on Computational Intelligence and Security. 2006. IEEE.

[20] Sivakumar, T. and R. Venkatesan, *A New Image Encryption Method Based on Knight's Travel Path and True Random Number*. Journal of Information Science and Engineering, 2016. 32(1): p. 133-152.

[21] Singh, M., A. Kakkar, and M. Singh, *Image Encryption Scheme Based on Knight's Tour Problem*. Procedia Computer Science, 2015. 70: p. 245-250.

[22] Kumar, J. and S. Nirmala, *A novel and efficient method based on knight moves for securing the information contents of images—A parallel approach*. Journal of Information Security and Applications, 2016. 30: p. 105-117.

[23] Delei, J., B. Sen, and D. Wenming. *An Image Encryption Algorithm Based on Knight's Tour and Slip Encryption-filter*. in Computer Science and Software Engineering, 2008 International Conference on. 2008. IEEE.

[24] Pareek, N.K., *Knight"s Tour Application in Digital Image Encryption*. International Journal of Advanced Research in Computer Science and Software Engineering, 2015. 5(9): p. 208-213.

[25] Sastry, P.N., *Kavyalankara Of Bhamaha (Sastry)(Ed. With Eng. Tr. And Notes)*. 1970: Motilal Banarsidass Publ.

[26] Mahmood, A.S. and M.S.M. Rahim, *State of the Art of Image Ciphering: A Review*. International Journal of Computer Science Issues (IJCSI), 2014. 11(2): p. 74.

[27] Diaconu, A.-V., A. Costea, and M.-A. Costea, *Color image scrambling technique based on transposition of pixels between RGB channels using Knight's moving rules and digital chaotic map*. Mathematical Problems in Engineering, 2014. 2014.

[28] Zhu, L., et al. *A novel algorithm for scrambling digital image based on cat chaotic mapping*. in 2006 International Conference on Intelligent Information Hiding and Multimedia. 2006. IEEE.

[29] Gordon, V.S. and T.J. Slocum. *The knight's tour-evolutionary vs. depth-first search*. in Evolutionary Computation, 2004. CEC2004. Congress on. 2004. IEEE.

[30] Zhang, J. and Z. Zhang, *Ant Colony Algorithms and Logistics Distribution Solutions*, in Communications and Information Processing. 2012, Springer. p. 734-740.

[31] Al-Gharaibeh, J., Z. Qawagneh, and H. Al-Zahawi. *Genetic Algorithms with Heuristic-Knight's Tour Problem. in GEM. 2007*. Citeseer.

[32] Jiang, D., S. Bai, and W. Dong. *An ant colony optimization algorithm for knight's tour problem on the chessboard with holes*. in 1st International Workshop on Education Technology and Computer Science. 2009.

[33] Kanso, A. and M. Ghebleh, *An efficient and robust image encryption scheme for medical applications.* Communications in Nonlinear Science and Numerical Simulation, 2015. 24(1): p. 98-116.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table (1) *NIST Statistical Test Results for Three Different Sizes with Same Initial Value*

	Key Size		
	256 × 256	**512 × 512**	**1024 × 1024**
Test Name	**P-Value**	**P-Value**	**P-Value**
Approximate Entropy	0.6024	0.3076	0.2796
Block Frequency	0.7892	0.7901	0.7883
Cumulative Sums	0.8936	0.8895	0.9105
Discrete Fourier Transform	0.5839	0.4109	0.4021
Frequency	0.3954	0.3954	0.3954
Linear Complexity	0.6491	0.5508	0.6709
Longest Run	0.6531	0.6495	0.6597
Non Overlapping Template	0.7832	0.8094	0.8108
Overlapping Template	0.7004	0.7097	0.7031
Random Excursions	0.5807	0.5292	0.6594
Rank	0.4034	0.4389	0.8603
Runs	0.8972	0.9127	0.8072
Serial	0.4024	0.4904	0.4502
Universal Statistical	0.7809	0.6493	0.7305

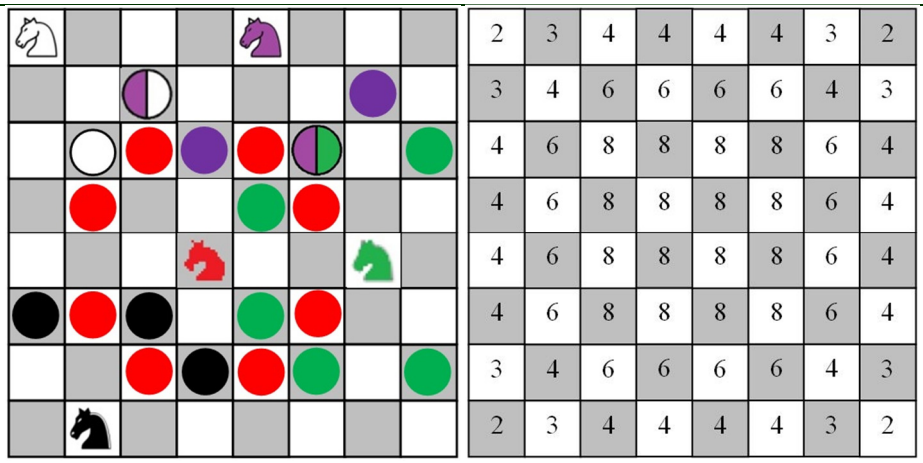
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Algorithm (1) *Pseudo-Code for Calculating the Movements of Knight on Chessboard*

```
Input: Current position (x, y)
Output: Next position (x̄, ȳ)
    Begin
        // Upward and Right
            If (x+1) <= (x_Size-1) and (y+2) <= (y_Size-1) Then
                x̄ ←x + 1
                ȳ ←y + 2
            End if
        // Upward and Left
            If (x-1) >= 0 and (y+2) <= (y_Size-1) Then
                x̄ ←x – 1
                ȳ ←y + 2
            End if
      // Right and Upward
            If (x+2) <= (x_Size-1) and (y+1) <= (y_Size-1) Then
                x̄ ←x + 2
                ȳ ←y + 1
            End if
        // Right and Downward
            If (x+2) <= (x_Size-1) and (y-1) >= 0 Then
                x̄ ←x + 2
                ȳ ←y – 1
            End if
        // Downward and Right
            If (x+1) <= (x_Size-1) and (y-2) >= 0 Then
                x̄ ←x + 1
                ȳ ←y – 2
            End if
        // Downward and Left
            If (x-1) >= 0 and (y-2) >= 0 Then
                x̄ ←x – 1
                ȳ ←y – 2
        // Left and Upward
            If (x-2) >= 0 and (y+1) <= (y_Size-1) Then
                x̄ ←x – 2
                ȳ ←y + 1
            End if
      // Left and Downward
            If (x-2) >= 0 and (y-1) >= 0 Then
                x̄ ←x – 2
                ȳ ←y – 1
            End if

            End if
    End.
```

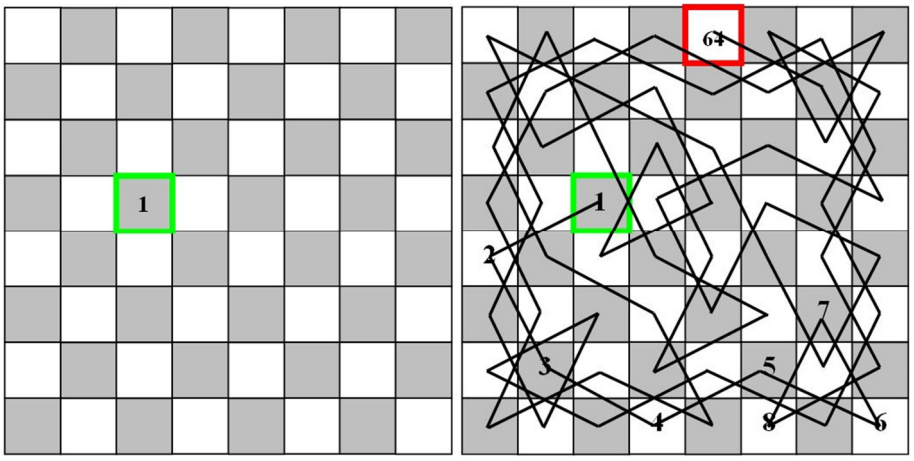
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195



2	3	4	4	4	4	3	2
3	4	6	6	6	6	4	3
4	6	8	8	8	8	6	4
4	6	8	8	8	8	6	4
4	6	8	8	8	8	6	4
4	6	8	8	8	8	6	4
3	4	6	6	6	6	4	3
2	3	4	4	4	4	3	2

(a) (b)

Figure (1) *Knight Tour Problem: (a) Five Example of Location of Knight on Chessboard and (b) Knight Possible Movements on Chessboard*



(a) (b)

52	41	14	31	64	27	12	29
15	32	53	50	13	30	63	26
40	51	42	47	54	61	28	11
33	16	1	60	49	58	25	62
2	39	48	43	46	55	10	57
17	34	19	38	59	44	7	24
20	3	36	45	22	5	56	9
35	18	21	4	37	8	23	6

(c)

52	41	14	31	64	27	12	29
15	32	53	50	13	30	63	26
40	51	42	47	54	61	28	11
33	16	1	60	49	58	25	62
2	39	48	43	46	55	10	57
17	34	19	38	59	44	7	24
20	3	36	45	22	5	56	9
35	18	21	4	37	8	23	6

(d)

Figure (2) I*mplementation of KT Generator: (a) Selection of Initial Position, (b) Knight Movements of Chessboard, (c) Chessboard Sequence After Knight Tour Movement, and (d) Generated Keys*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

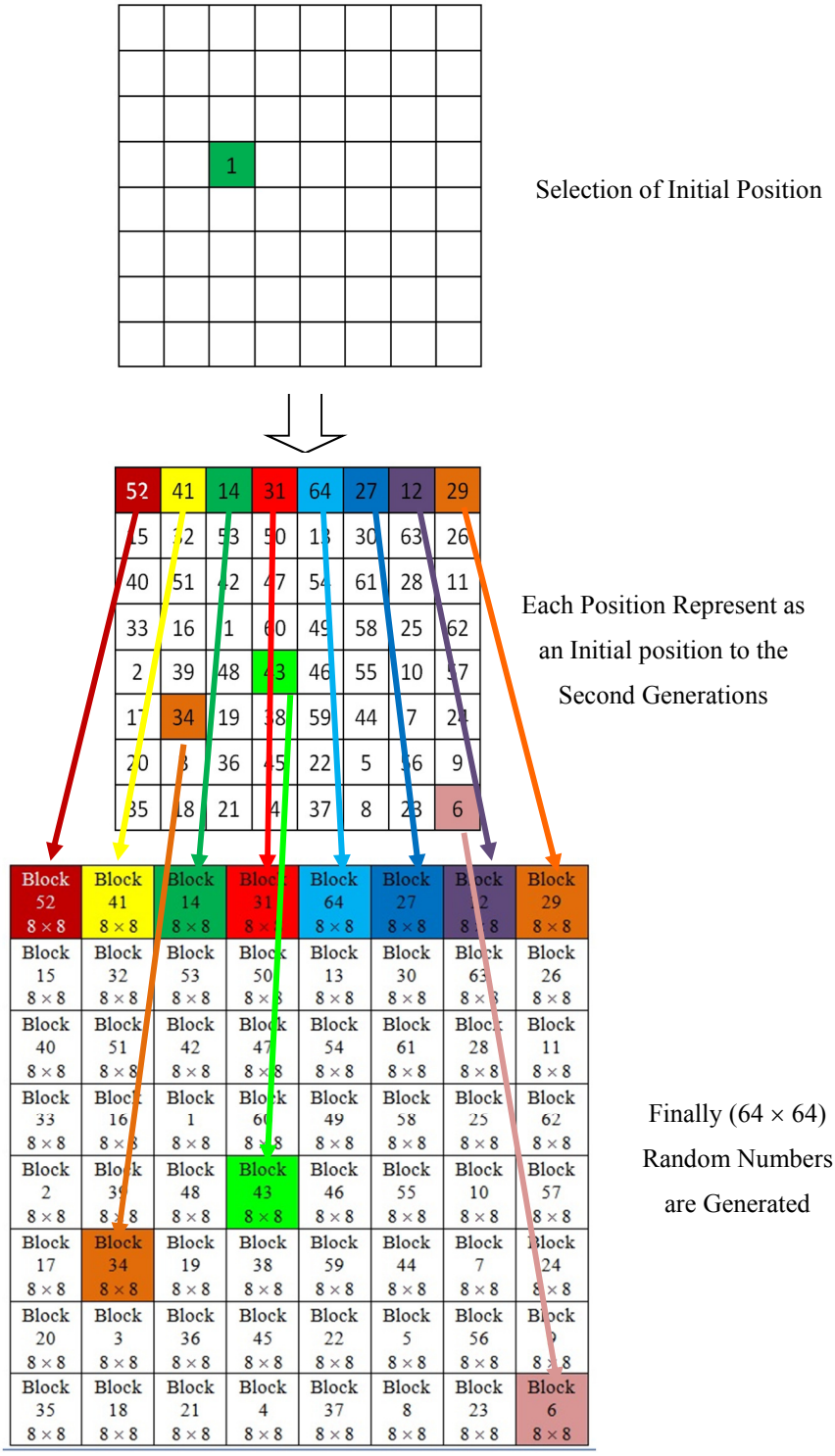


Figure (3) *First Stage of Key Expansion*

Journal of Theoretical and Applied Information Technology
15th April 2017. Vol.95. No 7

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

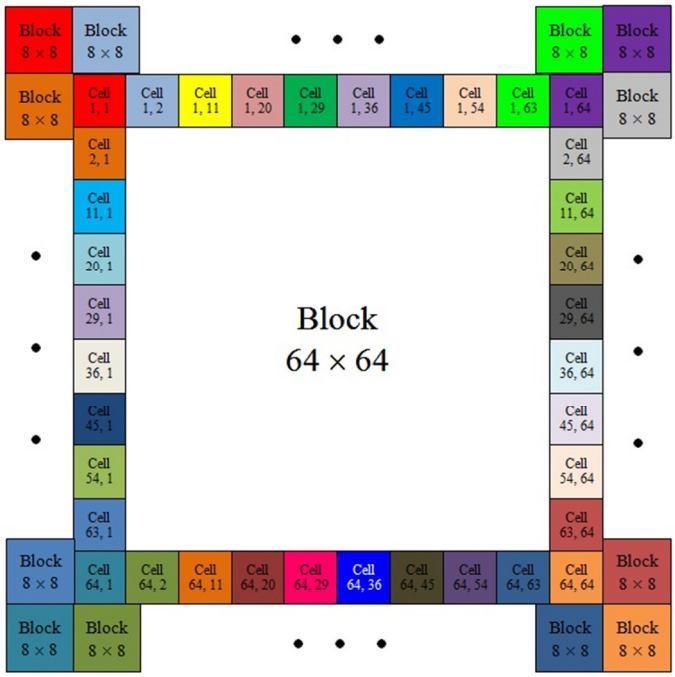


Figure (4) *Second Stage of Key Expansion*

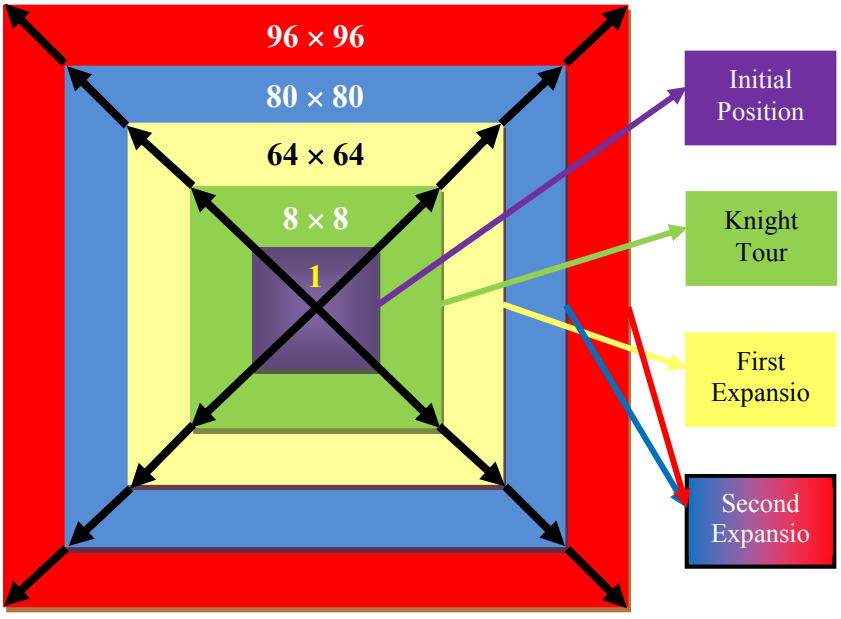


Figure (5) *Overall Process of Key Generating and Expansion*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

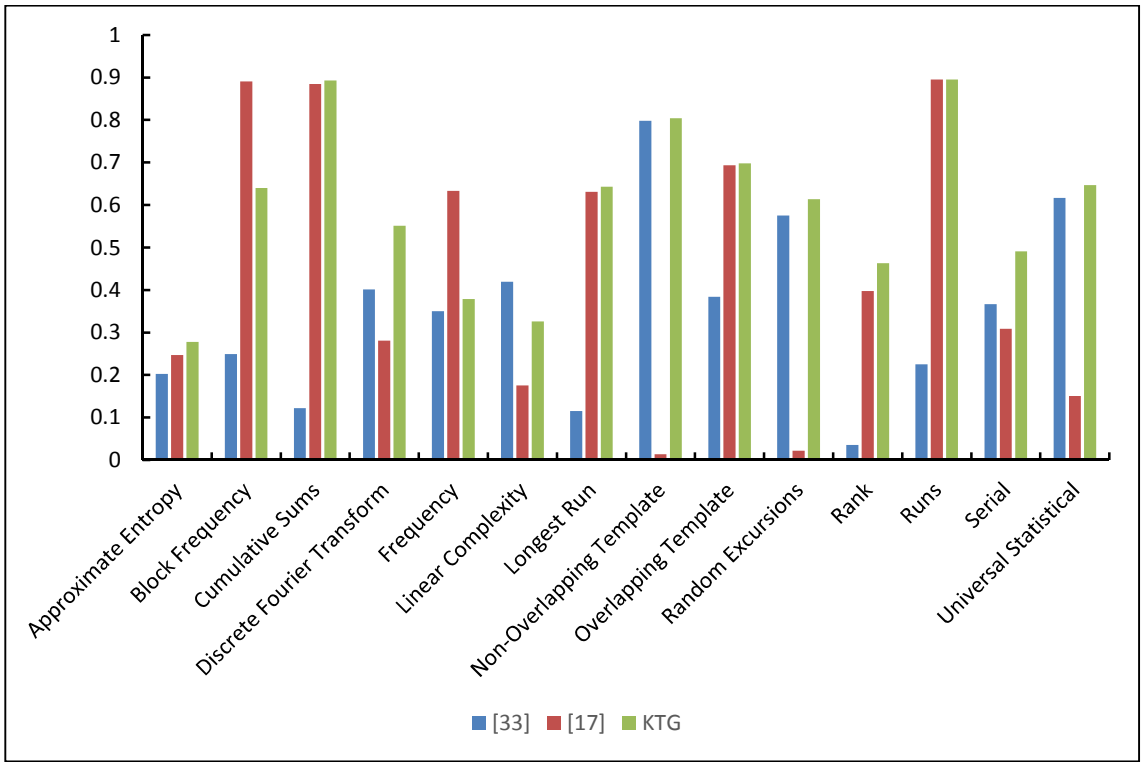


Figure (6) *Comparing of NIST Statistical Tests Results of the Proposed Key Generator with CMOS Generator Tests Results*