Investigation of Strength and Security of Pseudo Random Number Generators

K Sathya1*, J Premalatha2 and Vani Rajasekar1
1* Assistant Professor, Department of Computer Technology- UG
2 Professor, Department of Information Technology
3 Assistant Professor, Department of Computer Science and Engineering, Kongu Engineering College, Erode, India-638060.
*Email: pearlhoods@gmail.com

Abstract – Security is a key factor in today’s fast communicating world. Many cryptographic algorithms are tested and put into use efficiently. Random numbers are used in diverse forms like nonces, secret key, initialization vector, etc. They find place in encryption, digital signature, hashing algorithms. A deterministic algorithms takes an intial seed value as input and produces pseudo random numbers with falsely induced randomness. This research work extensively surveys large set of state-of-the-art PRNGs and categorizes them based on methodology used to produce them. We compared their statistical results obtained from various statistical test tools like NIST SP 800-22, TestU01. Finally, security analyses of various PRNGs were done quantitatively for their key space, key sensitivity, entropy, speed of bit generation, linear complexity. At last, we concluded the results with some future directions for researchers to carry out their research in improving the PRNGs.

Keywords – Random number generators, PRNG, statistical analysis, cryptography, security analysis of PRNG

1. Introduction
The sequence of numbers with unpredictable output has greater significance in various fields of technology like Monte Carlo simulations[1, 2], Cryptography[3, 4], Gaming theory[5], Statistics[6], etc. The random numbers are hard to predict and all the numbers generated must have equal probability to make them effective candidate. Today’s fast moving world demands on the fly security of information that is being shared. This paper extensively analyses the random number generators used in various algorithms of cryptography.

Cryptography uses random numbers in various forms like secret key, nonces, initialization vector, etc. Random numbers with complete unpredictable behaviour and uniform distribution are required to make the cryptosystems withstand any kind of attacks. Random numbers can be generated in either of two ways. Firstly from the true source of random like thermal noise, atmospheric pressure, SRAM start-up pattern, etc. They are called True Random Number Generators (TRNG)[7]. Secondly by feeding a seed value into a deterministic algorithm. They are called Pseudo Random Number Generator (PRNG)[8].

PRNGs can be constructed in multiple ways using various mathematical formulations. PRNGs are cost effective and faster to implement that makes them more favourable in faster communications
than their counterpart TRNGs. Many cryptographic protocols for key generation, key distribution, secure encryption, digital signature integrate either specialized random number generator algorithm into them or utilize the readily available random number generators embedded in the digital systems.

Since these cryptographic protocols’ security is now greatly dependent on random numbers, it is the responsibility of random number generators to produce high entropy numbers with unpredictable behaviour. Weak construction of random number generator can be vulnerable to reconstruct the keys generated and break the cryptosystem[9]. Various security analyses[10, 11] and statistical randomness tests[12, 13] have been developed to extensively test the generators before they are deployed.

PRNG need some seed value to initiate their sequence generation. Though this idea may seem simple, they are not secure as the system is completely predictable. To overcome this issue, many researchers have proposed various techniques like processing the seed value[14], modify the deterministic algorithm[15], combine different algorithms[16] to make the system look more complex. In this paper various methods of producing pseudo random numbers are extensively studied and detailed analyses of their statistical properties and security are presented.

Following this section, the remainder of this paper are grouped as follows: History of generating pseudo random numbers is described in Section 2, classification of PRNGs is presented and various published techniques in PRNG are studied in Section 3, statistical analyses of all techniques are analysed in Section 4, comparison of all the techniques in terms of performance, security are made in Section 5 and conclusion of this survey with brief discussion of future directions in improving PRNGs is given in Section 6.

2. History of PRNG

PRNG is a deterministic algorithm generating random number sequence with an initial seed value. The sequence generated is not truly random in nature owing to their dependence on the previous state of algorithm. However, these sequences are found useful for many fields of science owing to their simple nature. Hence they are named as Pseudo Random numbers. Traditionally pseudo random numbers were generated with mathematical linear equations[17, 18]. Some researches proposed to build secure PRNGs with block ciphering algorithms[19]. These PRNGs are known to produce good quality random numbers suitable for secure communications. PRNGs must be analysed mathematically to ensure their randomness properties before putting into use. Later days, PRNGs were constructed with seed from random sources[20] like sensors. These random numbers had improved randomness due to their random seeds. To make PRNGs more efficient and perform better in software, many chaotic system based PRNG[21] were proposed to generate random numbers.

3. Classification of PRNGs

The authors have made a detailed survey on various PRNGs and their testing methodologies. From this study, six major categories of random number generators are identified and listed as below:

i. Chaotic map based PRNG
ii. Polynomial equation based PRNG
iii. Hardware based PRNG
iv. Nature Inspired PRNG
v. Cryptographic algorithm based PRNG

This article surveys a huge variety of PRNGs that are implemented by various mathematical models. Each category of PRNG is explained in detail with their own advantages and disadvantages. They are analysed in various parameters like randomness, key space, entropy, security, performance, etc. These results are of helpful for researchers to focus their ideas into improving the weaknesses of surveyed PRNGs. It also provides directions for future research.
3.1 Chaotic map based PRNG

Chaotic maps are well known for their chaotic behaviour[22]. Chaotic maps are controlled by parameters and they produce output in iterative manner. Depending on the number of dimensions it produces outputs; chaotic maps are categorized into one-dimension, two-dimension, and three-dimension chaotic maps and so on.

Researchers have used wide varieties of chaotic maps like Logistic map[22], Henon map[23], Skew Tent map[24], Zig-Zag map[25], Lorenz map, Sawtooth map[26], Bernoulli map, etc. These maps have chaotic behaviour naturally, however when implemented on digital systems, they tend to lose their chaotic behaviour owing to digitization of output values.

To overcome the problems of chaotic degradation due to digitalization, various techniques have been proposed to improve their chaotic behaviour. Chaotic maps are modified in their basic structure or parameters, mix two or more chaotic maps or deploy higher order dimensional chaotic maps. These techniques are detailed below.

3.1.1 Modified chaotic maps. Using only the chaotic map as a deterministic algorithm to generate pseudo random numbers have many flaws in the domain of randomness, performance, key space. These flaws are mainly contributed by the digital implementation of chaotic maps where the chaotic attractors are represented using double-precision floating point number. When the attractors are represented in floating point 64-bits, the remaining bits are truncated to fit the memory location in a computer. This leads to loss of chaotic behaviour and the sequence generated tend to have shorter periods.

One way to improve the periodicity of random sequences is to modify the chaotic map by adding some other input like arcsine function[27], Physical Unclonable Function (PUF)[28], constant perturbation[29-33], combine multiple dimensions into single output[34-36], modifying their mathematical structure with additional operations[15, 37-42], time-varying delay function[43, 44], fractal order[45, 46].

The alternative way to improve the periodicity of chaotic maps is to modify their parameters. Ordering of parameter has improved the key length in [47]. Good parameters were chosen based on Kolmogorov Entropy measures to maintain good chaos in Sine map, Tent map and Logistic map[48]. Parameters can be dynamically varied by means of AES S-box as in [49], multiplying a probability value as in [50], choosing from set of parameter values as in [51, 52], mixing and confusing the parameters as in [53]. These modifications were proven to be improving the periodicity of generated sequences and have larger key spaces to withstand various attacks.

3.1.2 Combined chaotic maps. Single chaotic map may seem to be inefficient in producing good statistical results. Hence researchers also developed the idea of combining more than one chaotic map to improve their chaos. Combining the maps can be done in many forms so that output of one map perturbs the other map to have healthier chaos with minimal digital degradation. Cascading of two or more chaotic maps have minimized the degradation and was optimized to have faster generation speed in [54-56]. Other techniques include combine two or more different maps as in [57-63], combine a chaotic system with another true random source like Ring oscillator[64], coupling of maps[65-69], getting additional perturbation from hash function[70], parallel implementation of separate maps and combining their output as in [71, 72], combine some nonlinear operations into the map as in [73], output of multiple maps are multiplexed and shifted as in [74, 75]. Combining chaotic maps with these different techniques have improved their usability in cryptography for generating random keys and encrypt the images with improved efficiency.

3.1.3 Higher dimensional chaotic maps. With the advancement in computer systems in recent years, researchers have tried to put hyperchaotic systems in use to generate random numbers. A chaotic system with more than one Lyapunov attractor is termed as Hyperchaotic. Such hyperchaotic
systems have further more chaotic behaviour owing to presence of multiple attractors. Various studies have been done to utilize these hyperchaotic systems in cryptography. Five dimensional Hamiltonian chaotic system was proposed in [76], Logistic map is modified into four dimensions as in [77], modified into three dimensions as in [78]. Some hyperchaotic systems were perturbed to overcome the digital degradation as in [79, 80]. Use of memristive hyperchaotic system was proposed in [81]. Hyperchaotic systems were also providing good random sequences that are more suitable for applications like image and video encryption where higher rate of generation is required.

3.2 Polynomial equation based PRNG
PRNGs traditionally were built upon mathematical polynomial equations. Some of well known such generators are Linear Congruential Generator (LCG), Mersenne Twister, Linear Feedback Shift Register (LFSR). These kinds of generators can be implemented easily and efficiently in hardware and also in software. However, they have severe flaws like lesser periodicity, non-uniform distribution of random bits, high correlation between its outputs, etc. To make them potential candidates in cryptography owing to their simple nature, many researchers have proposed various ideas of improving them.

Two LCGs were coupled with variable inputs and implemented in hardware component like Field Programmable Gate Array (FPGA) to improve their periodicity[82]. Residue Number System (RNS) was applied to polynomials over finite field in [83] to increase the periodicity. Chaotic system was used to generate initial vectors for LFSR in [84, 85] that increased entropy of the random sequence. LFSR was combined with a layer of bit-reorganization and a non-linear function to efficiently encrypt videos with less data loss in [37]. Mersenne Twister is fed with low-entropy seeds taken from Static Random Access Memory (SRAM) of Radio Frequency Identification (RFID) tags to generate dynamic secret keys[86].

De Burjin block is combined with a Nonlinear Feedback Shift Register (NLFSR) to make them lightweight for suitable use in Wireless Sensor Networks (WSN) and RFID tags[87]. Composited de Burjin sequences is used to generate random numbers in [88]. LFSRs were modified by genetic algorithm to increase the length of output sequences in [89]. Sensor seeds were also proposed to be used as a source of entropy for LCG in [20, 90], they are more suitable for WSN were sensor data are readily available for use.

3.3 Hardware based PRNG
Hardware components can be of good source for producing randomness. Physical Unclonable Function (PUF) is a physical component that uniquely generates a fingerprint for every semiconductor devices like microprocessor. Ring oscillators with odd number of NOT gates connected in a chain fashion produces alternating 1s and 0s that may be taken as source of our random sequence.

A new hardware based chaotic system was proposed in [91] where Hamiltonian cycle was deleted within the N-cube and added one permutation to improve the chaotic behaviour of the proposed system. PUF generated random sequences were proven to be resistant to machine learning attacks [92] as its unpredictable behaviour is not easily learnt by machines. Ring Oscillator is coupled with hash functions to produce random sequence with good statistics in [93].

3.4 Nature inspired PRNG
Nature inspired computing is where inspiration to solve a particular problem was taken from a natural phenomenon. Such nature inspired algorithms try to implement the natural phenomenon to provide a novel solution to problem-solving techniques. Some of the most commonly inspired concepts are Cellular Automata (CA), cellular neural networks, genetic algorithm, DNA computing, quantum computing, etc. In recent years, researchers have come forward to put in use the nature inspired algorithms in generating random sequences with better statistical properties.

Cellular neural networks have been proposed to generate random sequences in six dimensions[94] with higher performance. Hybrid CA was proposed in [95] where non-linear feedback is coupled with
63 cells. This technique had higher throughput and was resistant to algebraic attacks. Self-programmable CA was utilized in [96] to generate faster bit rates with lesser energy. CA was combined with Langton’s ants where ants produced the disturbance in CA. Effect of varying the number of ants into CA has been extensively examined in [97]. Another nature inspired concept named Genetic Algorithm (GA) is applied to LFSR to increase the length of random output sequence in [89]. This technique proved to generate more secure keys for encryption. Another GA was proposed[98] to use calculation of entropy as fitness function to generate high entropy random sequence. Recurrent Neural Network (RNN) was added with Long Short-Term Memory (LSTM) to produce irrational number sequence[99]. This again fed to a hash function to generate the random sequence.

Quantum walks similar to random walks have their state transition with randomness coming from quantum superposition of states or reversible evolution or collapsing of wave functions. These quantum walks are also used in the field of generating random sequence as in [100]. Some researchers proposed to modify the quantum walks using quantum hash function as in [101]. Quantum chaotic map were proposed[46] to produce superior quality random sequences with good statistical results. They were also proven to have longer periodicity. However, using quantum chaotic maps must be examined carefully as they were shown to have 99% of weaker keys in [102].

3.5 Cryptographic algorithm based PRNG
PRNGs can also be constructed from cryptographic algorithms like hash, stream ciphers and block ciphers. These algorithms have multiple iterations and need some input in form of message and secret key. The result may be a hash value in case of hash function or an encrypted data in case of cipher algorithms. Hash value or encrypted data are considered to be good random sequence.

S-box of Advanced Encryption Standard (AES) algorithm is used along with piecewise Logistic map in [49] to improve the density of probability distribution and maintains a balance between efficiency and security. Grostl hash function have been proposed to be used with logistic map and Skew tent map to improve the statistical randomness of output sequence in [70]. Secure Hash Algorithm (SHA) -256 was used in[103] to generate random bit sequence by obtaining seed data from SRAM. These sequences were tested to be cryptographically secure. AES algorithm along with additional input from Sprott 94 G True Random Number Generator (TRNG) produced superior results in [104]. In a similar fashion, input from TRNG like Ring Oscillator was fed to Keccak hash function to produce cryptographically strong random number sequence in[93].

4. Statistical analyses
Random numbers generated by all these algorithms need to be statistically tested to verify their randomness property. An ideal PRNG must have uniform distribution of random numbers and independence between the numbers generated. Many statistical tools are readily available to measure the randomness of produced sequences. Most commonly used tools are NIST Statistical test suite 800-22[13], TestU01[105]. These software packages are free to use and they evaluate the randomness of a PRNG. NIST Statistical test suite 800-22 has 15 battery tests that analyse a PRNG in different dimensions. Each test produces a P-value based on which it can be determined whether the sequence has passed the test. TestU01 also performs an empirical analysis of RNG. This section compares the statistical analysis results of various PRNGs discussed above. A PRNG passing only few batteries of a test doesn’t mean it is unfit for use. Its properties can be improved in future by examining closely on its weakness. Table 1. depicts the PRNGs that have passed all the 15 battery tests of NIST and PRNGs that need minor improvements to overcome their failed results.
Table 1. NIST Test Results Analysis

| PRNGs                              | Passed all 15 tests | Failed few tests | NIST test not done |
|------------------------------------|---------------------|------------------|-------------------|
| Chaotic map based PRNG             | [24, 27, 28, 30-32, 34, 36-42, 47-58, 60-70, 74, 76-81] | [15, 29, 72]     | [45, 71, 73, 75]  |
| Polynomial equation based PRNG     | [20, 82, 83, 87, 90] | -                | [84, 88, 89]      |
| Hardware based PRNG                | [91]                | -                | [92]              |
| Nature Inspired PRNG               | [46, 94, 97-101]    | [95, 96]         | [102]             |
| Cryptographic algorithm based PRNG | [93, 103, 104]      | -                | -                |

Table 2. shows the list of PRNGs that passed all three tests – Alphabit, Rabbit and Smallcrush battery of tests. It also lists down some PRNGs that failed to pass all battery tests with inherent weaknesses.

Table 2. TestU01 Test Result Analysis

| Passed all 3 battery tests | Failed few tests |
|----------------------------|------------------|
| [15, 24, 31, 46, 47, 58, 59, 65-67, 71-73, 77, 85, 95] | [27, 56, 60, 101] |

5. Security analysis

5.1 Key sensitivity
PRNGs must be very sensitive to any change in their input seed values. With their internal state remaining constant, a diminutive alteration in the key or seed value should have a greater impact on its output. This behaviour is preferred in cryptographic applications, since smaller changes in input will highly affect the resulting sequence thus able to detect such errors. PRNG when constructed must be analysed for its sensitivity to changes in key value. While checking for sensitivity, researchers modify the key value in a micro level and find the correlation between original sequence and modified sequence. If the correlation between them is close to 0, then the sequences are completely unrelated and have higher sensitivity to initial key. Table 3. lists the PRNGs that have undergone the sensitivity test.

5.2 Linear complexity
Linear Complexity refers to the minimal LFSR length required to regenerate the resulting random sequence. It is recommended that higher the linear complexity, higher the unpredictability of PRNG. To be more ideal, a PRNG generating n-bit sequence can have a linear complexity equal to almost half of its length (n/2). Table 3. lists the PRNGs that have been tested for linear complexity and proven to have n/2 linear complexity. If a PRNG fails to have an almost n/2 linear complexity, it can be concluded to be more predictable and prone to security attacks.
Table 3. Key Sensitivity and Linear Complexity Analysis

| PRNG            | Key Sensitivity | Linear Complexity |
|-----------------|-----------------|-------------------|
| [24, 34, 37, 52, 55, 60, 61, 65] | ✓               | ✓                 |
| [15, 27, 31, 32, 35, 40, 42, 49, 53, 54, 57, 59, 62, 63, 66, 67, 73, 77-81, 85, 95, 97, 101] | ✓               | ✗                 |

5.3 Key space
The key space of a PRNG refers to the number of keys that can be totally generated before getting repeated. For a PRNG to withstand many cryptanalytic attacks, it is recommended to have key space greater than $2^{128}$. Larger the key space, higher the chance of being secure against differential attacks. Key space of a PRNG is dependent on its initial parameters, number of internal states. Fig 1. shows the key space of various PRNGs. It is evident that with advancement in digital platforms, it is easier to achieve higher key space.

Figure 1. Key Space Analysis of PRNGs

5.4 Entropy
Unpredictability of a PRNG can be measured in terms of its entropy. Entropy refers to the amount of information present in the given sequence. A PRNG generating n-bit random numbers can have totally $2^n$ possible random numbers. An ideal PRNG must have maximum entropy in each random sequence generated to ensure its unpredictable behaviour. It can be concluded that, a PRNG generating n-bit random number must have entropy close to n for it to be more unpredictable. Figure 2. shows the entropy achieved by various PRNGs. In general PRNG can be designed to generate 8,16,32,64,128 bit random numbers. The comparison made here in this section considers 8-bit PRNGs.
5.5 Speed

The rate at which a PRNG generates the random bits determines its speed. A PRNG generating large number of bits in short time find its usefulness in high speed network communications. These applications require random bits in order of Gigabits per second to make faster communication. PRNGs are required to be designed to meet the speed needs of particular applications. In Table 4, speeds of various proposed PRNGs are listed out to help researchers identify which category of PRNGs can produce random bits at required speed.

![Figure 2. Entropy Analysis of PRNGs](image)

Table 4. Speed Comparison of PRNGs

| PRNG | Speed (Mbps) |
|------|--------------|
| [54] | 80,000       |
| [91] | 6950         |
| [47] | 2.9          |
| [58] | 21.5054      |
| [65] | 3.8955       |
| [15] | 1.7          |
| [66] | 138.0864     |
| [49] | 164.3064     |
| [83] | 449.236      |
| [79] | 0.5017       |
| [86] | 0.01356      |
| [81] | 62.5         |
| [87] | 408          |
| [85] | 14.48        |
| [61] | 2.1105       |
| [103] | 598.1        |
| [80] | 367.88       |
| [31] | 34.18        |
| [35] | 2.1026       |
| [46] | 873.05       |
| [28] | 832          |
5.6 Resistance to security attacks

PRNGs are subjected to various attacks with the goal of obtaining the next random number to be generated. Some of the well known attacks on PRNG include Differential Attack (DA), Brute Force Attack (BFA), Side-Channel Attack (SCA), Chosen Plaintext Attack (CPA), etc. When a PRNG is designed, it is tested for its resistance to various attacks. Table 5. shows various PRNGs and their resistance to various attacks.

| PRNG                  | DA | BFA | SCA | CPA |
|-----------------------|----|-----|-----|-----|
| [54, 63]              | ✓  | ✓   | ✗   | ✗   |
| [49, 58, 62, 77, 101] | ✓  | ✗   | ✗   | ✗   |
| [36, 46, 66]          | ✓  | ✗   | ✗   | ✓   |
| [78, 85]              | ✗  | ✗   | ✗   | ✓   |
| [103]                 | ✗  | ✗   | ✓   | ✗   |
| [80]                  | ✗  | ✓   | ✗   | ✗   |

6. Conclusion

This paper detailed the analysis of various pseudorandom number generators[106] that are put into use in cryptographic applications[107-110]. We have first introduced the types of random number generators and their potential benefits and drawbacks. Later in the section how PRNGs evolved mathematically and found profound place in many areas of science. Next we categorized the techniques into six major categories of producing pseudo random numbers based on chaotic systems, polynomial equations, hardware, nature-inspired, and cryptographic algorithms. Then an exhaustive review of large varieties of PRNGs under each category was explained. Every PRNG must be analysed for its statistical properties, key space, entropy, linear complexity, key sensitivity, etc. A section comparing the statistical properties of all discussed PRNGs has been proposed. Finally the security analyses of all discussed PRNGs are summarized under each security parameter. This survey can be effectively utilised to make a brief comparison of various categories of PRNGs and identify the flaws that need improvement in future.

References
[1] Demchik V 2011 Computer Physics Communications 182 692-705.
[2] Kargaran H, Minuchehr A, Zolfaghari A 2016 AIP Advances 6 045101.
[3] Aljohani M, Ahmad I, Basheri M, Alassafi MO 2019 IEEE Access 7 39794-805.
[4] Figotin A, Gordon A, Molchanov S, Quinn J, Stavrakas NJMCM, Applications 2001 7 167-76.
[5] Lorek P, Słowik M, Zagórski F 2017 International Conference on Mathematical Aspects of Computer and Information Sciences (Springer) pp.425-37
[6] Figotin A, Gordon A, Molchanov S, Quinn J, Stavrakas N 2001 Monte Carlo Methods Applications 7 167-76.
[7] Garcia-Bosque M, Pérez A, Sánchez-Azqueta C, Celma S 2017 Sensors 17 646.
[8] Babaei M, Farhadi M 2011 International Journal of Communications, Network System Sciences 4 616.
[9] Dorrendorf L, Gutterman Z, Pinkas B 2009 ACM Transactions on Information System Security 13 1-32.
[10] Dodis Y, Pointcheval D, Ruhault S, Vergniaud D, Wichs D 2013 Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security pp.647-58
[11] Lambić D, Janković A, Ahmad M 2018 Journal of Electronic Testing 34 709-15.
[12] McCullough B. A review of TESTU01. JSTOR; 2006.
[13] Rukhin A 2001
[14] Silva RM, Crespo RG, Nunes MS 2009 International Journal of Communication Networks Distributed Systems 3 301-18.
[15] Murillo-Escobar M, Cruz-Hernández C, Cardoza-Avendaño L, Méndez-Ramírez R 2017 Nonlinear Dynamics 87 407-25.
[16] Wortman P, Yan W, Chandy J, Tehranipoor F 2018 IET Computers Digital Techniques 12 289-96.
[17] Boyar J 1989 Journal of Cryptology 1 177-84.
[18] Matsumoto M, Nishimura T 1998 ACM Transactions on Modeling Computer Simulation 8 3-30.
[19] Lee W-K, Cheong H-S, Phan RC-W, Goh B-M 2016 Cluster Computing 19 335-47
[20] Hong SL, Liu C 2015 IEEE Access 3 562-8.
[21] Guyeux C, Wang Q, Bahi JM 2010 International Conference on Web Information Systems and Mining (Springer) pp.202-11
[22] Phatak S, Rao SS 1995 Physical review E 51 3670
[23] Benedicks M, Carleson L 1991 Annals of Mathematics 133 73-169.
[24] Palacios-Luengas L, Pichardo-Méndez J, Díaz-Méndez J, Rodríguez-Santos F, Vázquez-Medina 2019 Arabian Journal for Science and Engineering 44 3817-30.
[25] Beirami A, Nejati H, Ali W 2012 Electronics Letters 48 1537-8.
[26] Jessa M 2006 IEEE Transactions on Circuits and Systems I: Regular Papers 53 1140-50.
[27] Zhu S, Zhu C, Cui H, Wang W 2019 IEEE Access 7 34141-52.
[28] Kalanadhabhatta S, Kumar D, Anumandla KK, Reddy SA, Acharyya A 2020 IEEE Transactions on Very Large Scale Integration Systems
[29] Dastgheib MA, Farhang M 2017 Nonlinear Dynamics 89 2957-66.
[30] Tutueva AV, Nepomuceno EG, Karimov AI, Andreev VS, Butusov DN 2020 Chaos, Solitons and Fractals 133 109615.
[31] Lambić D, Nikolić M 2019 Filomat 33 2257-68.
[32] Irfan M, Ali A, Khan MA, Ehatisham-ul-Haq M, Mehmood Shah SN, Saboor A, et al. 2020 Electronics Letters 9 104.
[33] Meranza-Castillón M, Murillo-escobar M, López-Gutiérrez R, Cruz-Hernández C 2019 AEU-International Journal of Electronics and Communications 107 239-51.
[34] Hanma R 2017 Journal of Information Security Applications 35 119-29.
[35] Riaz M, Ahmed J, Shah RA, Hussain A 2018 Wireless Personal Communications 99 85-93.
[36] García-Martínez M, Campos-Cantón E 2015 Nonlinear Dynamics 82 2119-31.
[37] Xu H, Tong X, Meng X 2016 Optik 127 9305-19.
[38] Yuan Z, Li H, Miao Y, Hu W, Zhu X 2017 International Journal of Bifurcation and Chaos 27 1750210.
[39] Machicao J, Bruno OM 2017 Chaos: an interdisciplinary journal of nonlinear science 27 053116.
[40] Moysis L, Volos C, Jafari S, Munoz-Pacheco JM, Kengne J, Rajagopal K, et al. 2020 Entropy 22 474.
[41] Rezk AA, Madi A, Radwan AG, Soliman AM 2020 AEU-International Journal of Electronics Communications 113 152947.
[42] Meranza-Castillón M, Murillo-escobar M, López-Gutiérrez R, Cruz-Hernández C 2019 AEU-International Journal of Electronics and Communications 107 239-51.
[43] Lv X, Liao Y, Yang B 2018 Nonlinear Dynamics 94 325-45.
[44] Palacios-Luengas L, Pichardo-Méndez J, Díaz-Méndez J, Rodríguez-Santos F, Vázquez-Medina R 2019 Arabian Journal for Science and Engineering 44 3817-30.
[45] Ozkaynak F 2020 Elektronika ir Elektrotechnika 26 52-7.
[46] Akhshani A, Akhavan A, Mobaraki A, Lim S-C, Hassan Z 2014 Communications in Nonlinear Science and Numerical Simulation 19 101-11.
[47] Barani MJ, Ayubi P, Valandar MY, Irani BY 2020 Journal of Information Security and Applications 53 102509.
[48] Lan R, He J, Wang S, Liu Y, Luo X 2018 IEEE Transactions on Circuits and Systems II: Express Briefs 66 492-6.
[49] Wang Y, Liu Z, Ma J, He H 2016 Nonlinear Dynamics 83 2373-91.
[50] Moysis L, Tutueva A, Volos C, Butusov D, Munoz-Pacheco JM, Nistazakis H 2020 Symmetry 12 829.
[51] Garcia-Bosque M, Pérez-Resa A, Sánchez-Azqueta C, Aldea C, Celma S 2018 IEEE Transactions on Instrumentation and Measurement 68 291-3.
[52] Liu L, Miao S, Hu H, Deng Y 2016 IET Information Security 10 87-94.
[53] Wang L, Cheng H 2019 Entropy 21 960.
[54] François M, Defour D, Negre C 2014
[55] Alawida M, Samsudin A, Teh JS 2020 Information Sciences 512 1155-69.
[56] Alawida M, Samsudin A, Teh JS 2019 Nonlinear Dynamics 96 601-13.
[57] Abdullah HA, Abdullah HN, Mahmoud Al Jawher WA 2020 International Journal of Communication Systems 33 e4236.
[58] Huang X, Liu L, Li X, Yu M, Wu Z 2019 Complexity 2019
[59] Zhou Y, Hua Z, Pun C-M, Chen CP 2014 IEEE transactions on cybernetics 45 2001-12.
[60] Liu Y, Luo Y, Song S, Cao L, Liu J, Harkin J 2017 International Journal of Bifurcation and Chaos 27 1750033.
[61] Alawida M, Samsudin A, Teh JS 2019 IEEE Access 7 150609-22.
[62] Xu H, Tong X-J, Zhang M, Liu Y, Wang Z 2018 International Journal of Bifurcation and Chaos 28 1850172.
[63] François M, Grosjes T, Barchiesi D, Erra R 2014 Communications in Nonlinear Science and Numerical Simulation 19 887-95.
[64] Avaroglu E, Tuncer T, Bedri Ozer A, Turk M 2014 Journal of Microelectronics, Electronic Components and Materials 44 303-11.
[65] Huang X, Liu L, Li X, Yu M, Wu Z 2019 Mathematical Problems in Engineering 2019
[66] Lv X, Liao X, Yang B 2018 Nonlinear Dynamics 94 325-41.
[67] Hua Z, Zhou Y 2015 IEEE transactions on cybernetics 46 3330-41.
[68] Elmanfaloty RA, Abou-Bakr E 2019 Chaos, Solitons and Fractals 118 134-44.
[69] Avaroglu E 2017 Turkish Journal of Electrical Engineering and Computer Sciences 25 633-43.
[70] Özkalınak F 2014 Nonlinear Dynamics 78 2015-20.
[71] Guyeux C, Couturier R, Héam P-C, Bahi JM 2015 The journal of Supercomputing 71 3877-903.
[72] Öztürk I, Kilç R 2015 Nonlinear Dynamics 80 1147-57.
[73] Hua Z, Zhou Y 2017 IEEE Transactions on Circuits and Systems I: Regular Papers 65 235-46.
[74] Rezk AA, Madian AH, Radwan AG, Soliman AM 2019 AEU-International Journal of Electronics and Communications 98 174-80.
[75] Purkayastha T, De D, Das K 2016 Microprocessors and Microsystems 45 32-44.
[76] Dong E, Yuan M, Du S, Chen Z 2019 Applied Mathematical Modelling 73 40-71
[77] Wang Y, Zhang Z, Wang G, Liu D 2019 International Journal of Bifurcation and Chaos 29 1950124.
[78] Sahari ML, Boukemara I 2018 Nonlinear Dynamics 94 723-44.
[79] Zhao Y, Gao C, Liu J, Dong S 2019 Chaos, Solitons and Fractals: X 4 100023.
[80] Liu Y, Tong X 2016 IET Information Security 10 433-41
[81] Yu F, Li L, He B, Liu L, Qian S, Huang Y, et al. 2019 IEEE Access 7 181884-98.
[82] Panda AK, Ray KC 2019 IEEE Transactions on Instrumentation and Measurement 69 1011-9.
[83] Ma S, Liu J, Yang Z, Zhang Y, Hu J 2018 Science China Information Sciences 61 082304.
[84] Mondal B, Mandal T 2020 Multimedia Tools and Applications 1-24.
[85] Alhadawi HS, Zoolkipli MF, Ismaill SM, Lambić D 2019 Cryptologia 43 190-211.
[86] Ozcanhan MH, Unluturk MS, Dalkilic G 2016 Advances in Electrical and Computer Engineering 16 81-91.
[87] Mandal K, Fan X, Gong G 2016 ACM Transactions on Embedded Computing Systems 15 1-28.
[88] Mandal K, Gong G 2015 IEEE Transactions on Computers 65 2725-38.
[89] Sudeepa K, Aithal G, Rajinikanth V, Satapathy SC 2020 Pattern Recognition Letters
[90] Wallace K, Moran K, Novak E, Zhou G, Sun K 2016 IEEE Internet of Things Journal 3 1189-201.
[91] Bakiri M, Guyeux C, Couchot J-F, Marangio L, Galatolo S 2018 IEEE Transactions on Industrial Informatics 14 3754-65.
[92] Wen Y, Yu W 2019 Electronics Letters 55 515-7.
[93] Yakut S, Tuncer T, Özer AB 2019 Elektronika ir Elektrotechnika 25 40-6.
[94] Hu G, Peng J, Kou W 2019 International Journal of Computational Intelligence Systems 12 643-8.
[95] Dogaru R, Dogaru I 2015 PROCEEDINGS OF THE ROMANIAN ACADEMY SERIES A-MATHEMATICS PHYSICS TECHNICAL SCIENCES INFORMATION SCIENCE 16 367-74.
[96] Petrica L 2018 Journal of Parallel and Distributed Computing 111 251-9.
[97] Hosseini SM, Karimi H, Jahan MV 2014 Journal of Information Security and Applications 19 149-62.
[98] KÖSEMEN C, DALKILIÇ G, AYDIN Ö 2018 Turkish Journal of Electrical Engineering and Computer Sciences 26 2500-11.
[99] Jeong Y-S, Oh K-J, Cho C-K, Choi H-J 2020 The Journal of Supercomputing 1-19.
[100] Yang Y-G, Zhao Q-Q 2016 Scientific reports 6 1-11.
[101] Yang Y-G, Xu P, Yang R, Zhou Y-H, Shi W-M 2016 Scientific reports 6 19788.
[102] Lambić Đ 2018 Nonlinear Dynamics 94 1117-26.
[103] Chen S, Li B, Zhou C 2018 Microprocessors and Microsystems 59 57-68.
[104] Avaroğlu E, Koyuncu I, Özer AB, Türk M 2015 Nonlinear Dynamics 82 239-48.
[105] L’Ecuyer P, Simard R 2013
[106] Sathya K, Premalatha J, Rajasekar V 2015 2015 IEEE Workshop on Computational Intelligence: Theories, Applications and Future Directions (WCI) (IEEE) pp.1-5
[107] Rajasekar V, Varadhagapanathy S, Sathya K, Premalatha J 2016 2016 3rd International Conference on Recent Advances in Information Technology (RAIT) (IEEE) pp.394-7
[108] Premalatha J, Sathya K, Rajasekar V 2016 International Journal of Computers and Communications 10 18-22.
[109] Usha M, Vanitha C 2016 Wireless Personal Communications 89 27-43.
[110] Dhiviya S, Malathy S, Monikha M 2018 4th International Conference on Computing Communication and Automation (ICCCA) (IEEE) pp.1-6