Journal of Computer Science
https://dergipark.org.tr/en/pub/bbd
https://doi.org/10.53070/bbd.990990

Journal of Computer Science
ISSN: 2548-1304
Volume: IDAP-2021 , Issue: Special , pp: 53-60
@Anatolian Science

# Randomness Analysis With Runge Kutta Methods

Cemile İNCE*[1], Kenan İNCE[2], Davut HANBAY[3]

[1]Department of Information Technology, İnönü University, Malatya, Turkey

[2]Department of Software Engineering, İnönü University, Malatya, Turkey

[3]Department of Computer Engineering, İnönü University, Malatya, Turkey

(cemile.ince@inonu.edu.tr, kenanince@gmail.com, davut.hanbay@inonu.edu.tr)

Received: Sep.3, 2021 Accepted: Sep.16, 2021 Published: Oct.20, 2021

**Abstract—** Chaotic systems are widely used in encryption because of their sensitivity to initial conditions and parameters, high ergodicity, mixing properties, and highly complex structures. Various analyzes are available to understand whether a system is chaotic. The most used analyzes are time series analysis, phase portraits, Lyapunov exponents, and bifurcation diagrams. Modeling of chaotic systems is also possible with numerical analysis methods. These methods are; Houses, Heun, 4th and 5th degree Runge Kutta methods are the most common differential solution methods.

**Keywords:** Random Number Generators, Runge Kutta, Chaotic Maps, Floating Point Numbers

## 1. Introduction

In today's world, where the use of the Internet has become widespread, data is increasing in proportion to the usage. The size of data in digital environments is increasing day by day. The huge increase in data sizes has brought with it the fact that the required storage areas for data storage have increased, and that the concept of information, which is important in data, must be kept, and that it is inevitable to keep personal information such as personal data.

Due to the nature of data transmission, it is open to monitoring and intervention. There are many methods for encrypting data. DES for encryption of text data; There are many methods such as AES, Blowfish. Although these algorithms are very suitable for text encryption, they are not suitable for encrypting large data such as pictures and videos. Because the large size means both a waste of time and partial decoding of the text is not considered to be fully understood, while even partial decoding of the image data is sufficient condition for understanding the content [1].

While personal data such as fingerprint, retina and iris are stored in databases, keeping them without encryption has brought along many security vulnerabilities. Akgul et al.[2] In their study, using chaotic system infrastructure, vein image encryption and vein images within the scope of personal data were encrypted in the database and stored in the microcomputer environment. They generated random numbers before encryption and passed NIST 800-22 tests for the analysis of the security of these numbers. There are image encryption studies using the S-Box designed with random number generators. Li et al. [3] In 2020, they propose a 32-bit positive random number generator called PL_PWLCM for image encryption based on chaotic maps. In the study, they proposed a four-pixel row and column-based diffusion algorithm for image encryption. While carrying out this study, it was stated that there was a high correlation between image pixels, and therefore advanced encryption standards such as DES, AES and RSA were not appropriate [3]. S-Boxes are one of the most important algorithms for block ciphers. Random number generators are used for chaos-based S-box generation. There are studies that show various security vulnerabilities in image encryption of block cipher algorithms. Chen et al. [4] designed a chaos-based QoS and security-enhancing algorithm to increase the security of data transmission over RTP (real time protocol). Tong et al. [5] proposed a hyper-chaotic system for wireless networks; They used cubic and logistic chaotic maps together. Although the S-box design is widely used in image encryption, it is difficult to implement. As an alternative to S-Boxes, chaos-based systems, which are extremely sensitive to initial conditions, have become widespread in recent years [6]. It is also possible to use chaotic maps as an alternative to S-boxes. There are maps such as logistic map, henon map, tent map, duffing map, Arnold cat map to produce chaotic maps. Zhang Y. [7]. talked about the advantages of chaotic systems in generating random numbers in his study in 2016, stated that S-box was insufficient in encryption, chaos maps were used to eliminate this disadvantage, and linear chaotic maps were used while creating these maps. In order to increase the spread of pixel densities, Arnold transform is used for encryption of digital images in the proposed study [6], while s-box is used for partial encryption, while partially encrypted image is created by applying Arnold transform 10 times for full encryption. Ginittting and Dillak [8] proposed a secure algorithm based on a hash encryption algorithm using the RC4 algorithm and the logistic map algorithm for digital picture encryption. Experimental results showed that the proposed hash algorithm could not visually identify the encrypted image, but it eliminated the correlation between the plain image and the encrypted image. In another study using Baker's chaotic map [9], it was stated that chaos is used to expand the clutter and confusion in the image, dynamic permutation map is used due to its sensitivity to initial conditions, and dynamic permutation map has a good potential for designing S-boxes.

One of the ways to create chaotic encryption is to use boolean algebra functions. Khan et al. [10] focused around the successful and fast chaotic binary boolean function to ensure information security. The study, in which chaotic maps were used for chaotic structure, stated that boolean functions were used to create key fields. In addition, in the study, traditional methods require several rounds in the encryption process; Therefore, the importance of encryption algorithms that are fast and highly resistant to cryptanalysis in real-time applications is mentioned. Stating that S-box structures are not sufficient for image encryption, Farwa S. et al. reported that a new and secure image encryption is created by combining it with algebraic encryption with pixel mixing effect using Arnold cat maps transformation. In addition to creating static encryption, it is possible to create these structures dynamically. In another study where image encryption is performed using dynamic s-box, Z_257 transform is used to create the S-box algebraic structure [11]. While a single chaotic map will be used in chaotic system design, more than one chaotic map can be used in the same system design. Structures in which more than one chaotic system is used are generally called hyper-chaotic systems. There are two important problems in hyper-chaos-based encryption algorithms: The first is the periodic corruption of the chaotic sequence. Since the processor precision is finite, the chaotic sequence can turn into a periodic function or a fixed point. To solve this problem, [11] proposed the LFSR shift register. The second important problem is that the system must have a finite field inversion. For inversion, the se author uses the Z_257 algebrabic structure. This operation is called inverse Z_N operation.

Random number generators, which have an important place in encryption, need to generate random numbers and these numbers must provide some security tests. There are internationally accepted statistical tests in the literature. The most widely used statistical tests are NIST tests.

## 2. Random Number Generators

It is possible to classify random numbers as real random numbers obtained depending on physical conditions and pseudo-random numbers obtained by software. Pseudo-random numbers are derived from a seed, commonly called a seed.

Since their implementation is software-based, their costs are low. However, finding the initial value brings with it serious security vulnerabilities [12].

True random number generators are meta-stability, phase jitter (jilter), user interactions, chaos, using non-deterministic physical randomness sources; In short, they are hardware-generated numbers. Being able to generate hardware increases the cost, but it is difficult to generate random numbers with good statistical properties from true random number generators, despite their cryptographic distinctiveness such as unpredictability and non-reproducibility. Because the internal deviations and correlations produced from the entropy source prevent them from showing statistically uniform distribution. To overcome this situation, GRSUs are usually post-processed, which requires extra time and cost [13]

In this study, successful results were obtained by generating chaos-based pseudo-random numbers with Runge Kutta methods and subjecting them to the internationally accepted NIST 800-22 tests. One-million-bit length PRNGs (pseudo random number generator), which are produced based on chaos and passed the tests successfully, are one of the indispensable software elements of security equipment.

## 2.1.Chaotic Systems

In recent years, in addition to existing chaotic systems, chaotic systems produced by making changes in existing systems have been developed. Akgül and Pehlivan created their own chaotic equations and examined the chaotic properties [14]. Other existing studies have proposed various chaotic equations by using existing chaotic equations or by mixing these equations to produce hybrid chaotic systems [15] [10] [11] [16].

In this study, random numbers generated on the basis of 4 in the runge kutta and 5 in the runge kutta were passed through the NIST tests and mathematically generated random number generators were obtained. NIST tests consist of 15 sub-statistical tests. A set length of one million bits of binary strings is required to perform some subtests. Successfully generated random numbers were passed through NIST tests and successful results were obtained.

## 2.2.Generating of PRNG With RK4 and RK5

In the fourth-order runge kutta method, many different fourth-order runge kutta methods emerge according to the selection of a1, a2, a3, a4. Along with flour, k1, k2, k3,k4 also varies. The general Runge Kutta equation given in Equation 1.

$$y(x_i + \Delta h) = y(x_i) + (a_1 * k_1 + a_{2*}k_2 + a_3k_3 + a_4 * k_4) \qquad (1)$$

The equation for obtaining the general formula and coefficients with the most commonly used coefficients method is as follows:

$$y(x_i + \Delta h) = y(x_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) * \Delta h) \qquad (2)$$

$$k_1 = f(x_i\,, y_i) \qquad y' = f(x, y) \qquad (3)$$

$$k_2 = f(x_i + \frac{1}{2} * \Delta h, y_i + \frac{1}{2} * k_1 * \Delta h) \qquad (4)$$

$$k_3 = f(x_i + \frac{1}{2} * \Delta h, y_i + \frac{1}{2} * k_2 * \Delta h) \qquad (5)$$

$$k_4 = f(x_i + \Delta h\,, y_i + k_3 * \Delta h) \qquad (6)$$

### 2.3. Lorenz Attractor

Basically, when the mathematical equations of runge kut 4 are applied to the lorenz attractor, the x-y-z 3-phase chaotic scheme is obtained as in the vaccine. Lorenz equation:

x=sigma(y-x)
y=x*(p-z)-y;

$$z=x*y-(B*z) \tag{7}$$

Has a three-dimensional equation. When the Lorenz chaotic equation provides the initial values σ=10, B=8/3, p=28, z=9, x=0, y=0.1, the system starts to show chaoticity.

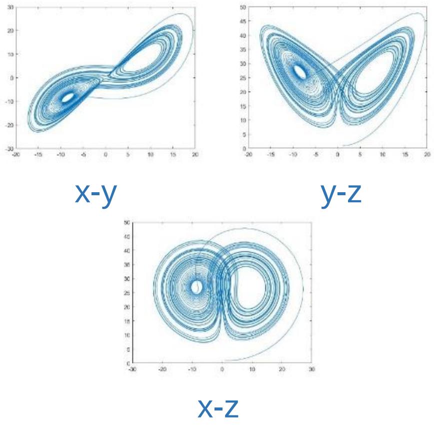


**Figure 1:** Application of RK4 differential equation to Lorenz Attractor

Lorenz Attractor is a 3D chaotic equation. When the RK4 algorithm is applied to the Lorenz Chaotic equation, separate floating-point numbers are obtained for x, y, z depending on the h sampling interval. The resulting floating-point numbers were converted to binary format and made suitable for NIST tests. The first 10 examples of random numbers generated for the generated floating-point numbers are as in the table:

**Table 1**: x, y, z Generated Floating Point Numbers and Converting to Binary Format Example Table

| **Example** | **X** | **Y** | **Z** |
|---|---|---|---|
| 1 | 1.012567191 | 1.259917799 | 0.984890972 |
| 2 | 1.04882371 | 1.523997131 | 0.97311422 |
| 3 | 1.107208854 | 1.79830989 | 0.965158951 |
| 4 | 1.186868017 | 2.088540142 | 0.961737225 |
| 5 | 1.287557057 | 2.400154464 | 0.963806064 |
| 6 | 1.409570658 | 2.738545614 | 0.97260817 |
| 7 | 1.553690062 | 3.109153968 | 0.989731122 |
| 8 | 1.72114638 | 3.517569455 | 1.017186524 |
| 9 | 1.913596203 | 3.96961501 | 1.057511855 |
| 10 | 2.133106543 | 4.471410648 | 1.113898918 |

The floating-point number format is regulated according to the IEEE 754 standard. It is realized in two different ways, single precision 32 bit and double precision 64 bit. number expressed in 32 bits; It consists of 1-bit sign, 8 bits exponent, 23-bit significant bit. After converting the float numbers to binary format, the binary format for the sample ten numbers is as in the table below:

**Table 2**: Converting float numbers obtained from x, y, z phases to binary format

|  | X | Y | Z |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 1 | 00111111100000011001101111001101 | 00111111101000010100010011111101 | 00111111011111000010000111010001 |
| 2 | 00111111100001100011111111011011 | 00111111110000110001001001010111 | 00111111011110010001111000000011 |
| 3 | 00111111100011011011100100000101 | 00111111111001100010111100000101 | 00111111011101110001010010101000 |
| 4 | 00111111100101111110101101001011 | 01000000000001011010101010100100 | 00111111011101100011010001101001 |
| 5 | 00111111101001001100111010101011 | 01000000000110011001110000100001 | 00111111011101101011101111111111 |
| 6 | 00111111101101000110110011010000 | 01000000001011110100010001010101 | 01111110111100011111100110110001 |

Each number in the generated binary format consists of 32 bits.

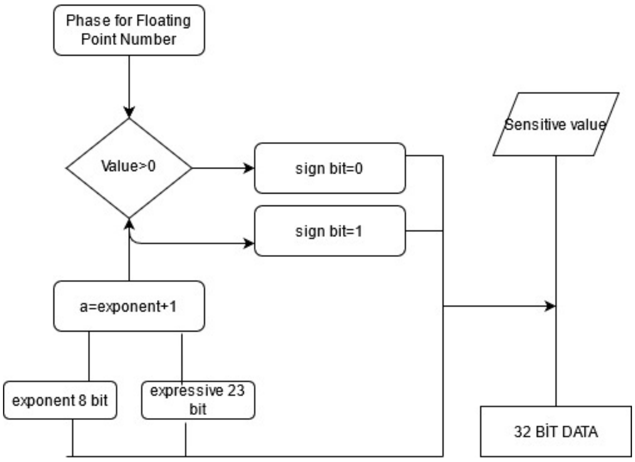
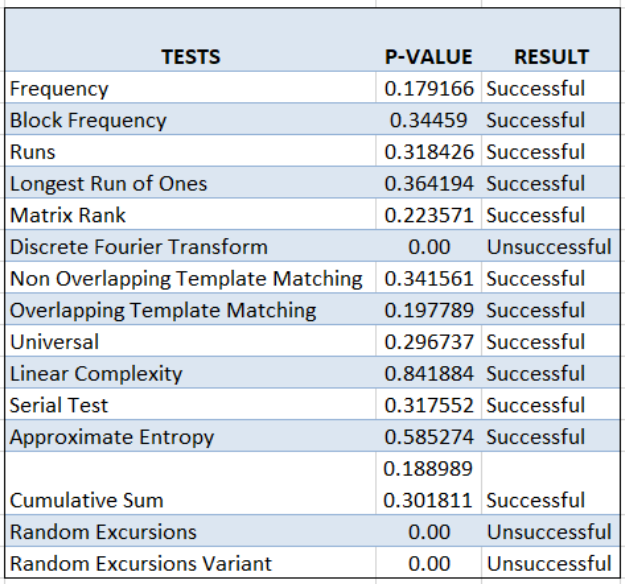


**Figure 2:** Flow Chart Used to Generate Float Numbers

## 3. NIST Tests of Produced PRNGs

The unpredictability of randomness in cryptography means that there is no statistical relationship between samples. Random numbers, on the other hand, are numbers that do not have a relationship or correlation. The use of random numbers is important in the field of cryptology. Random number generators are used in many areas such as key field determination, key distribution, initial value determination, authentication, and encrypted storage of personal data in the database. Pseudo-random number generators are obtained by obtaining random number generators from software, and real random number generators are obtained by obtaining from hardware sources. Hybrid number generators are obtained by using the two generators together.

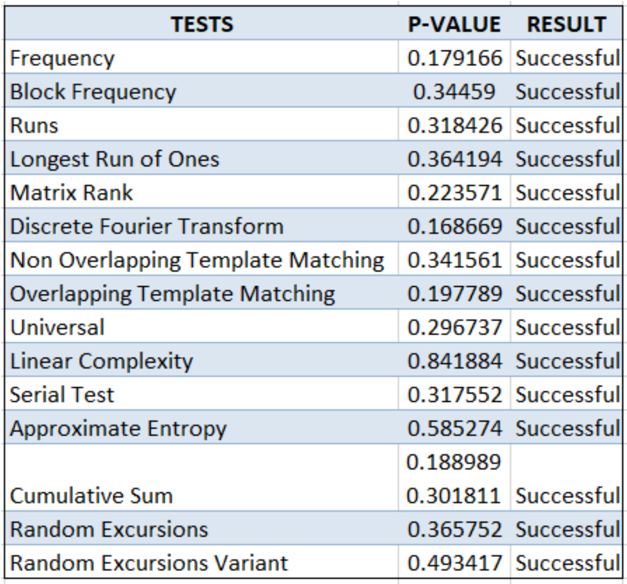
Chaos-based random numbers were generated with floating point numbers obtained using the RK4 algorithm. The produced float numbers were converted to binary mode and made suitable for NIST tests. NIST tests consist of 15 internationally accepted statistical tests. The table obtained as a result of the test was as follows:



| TESTS | P-VALUE | RESULT |
|---|---|---|
| Frequency | 0.179166 | Successful |
| Block Frequency | 0.34459 | Successful |
| Runs | 0.318426 | Successful |
| Longest Run of Ones | 0.364194 | Successful |
| Matrix Rank | 0.223571 | Successful |
| Discrete Fourier Transform | 0.00 | Unsuccessful |
| Non Overlapping Template Matching | 0.341561 | Successful |
| Overlapping Template Matching | 0.197789 | Successful |
| Universal | 0.296737 | Successful |
| Linear Complexity | 0.841884 | Successful |
| Serial Test | 0.317552 | Successful |
| Approximate Entropy | 0.585274 | Successful |
| Cumulative Sum | 0.188989<br>0.301811 | Successful |
| Random Excursions | 0.00 | Unsuccessful |
| Random Excursions Variant | 0.00 | Unsuccessful |

**Figure 3:** NIST Statistical Test Results of Issues Generated Based on RK4.

When the figure is examined, it is seen that 12 tests passed successfully, except for the fourier, random excursions and random excursions variant tests, which are among the NIST statistical tests. When the reasons for the three failed tests are examined, it is clear that the fourier test basically examines the periodicity of the sequence, the cumulative total randomness in the random test, and the number of n cycles in the cumulative total random walk in the random excursions variant test. In order to pass these tests, by examining the binary sequences we produced with the RK4 method, it was seen that the most chaoticity was in the X and Z phases. In this study, the binary sequence produced by XORing the X and Z phases was passed through the NIST tests again, and the following results were obtained.



| TESTS | P-VALUE | RESULT |
|---|---|---|
| Frequency | 0.179166 | Successful |
| Block Frequency | 0.34459 | Successful |
| Runs | 0.318426 | Successful |
| Longest Run of Ones | 0.364194 | Successful |
| Matrix Rank | 0.223571 | Successful |
| Discrete Fourier Transform | 0.168669 | Successful |
| Non Overlapping Template Matching | 0.341561 | Successful |
| Overlapping Template Matching | 0.197789 | Successful |
| Universal | 0.296737 | Successful |
| Linear Complexity | 0.841884 | Successful |
| Serial Test | 0.317552 | Successful |
| Approximate Entropy | 0.585274 | Successful |
| Cumulative Sum | 0.188989<br>0.301811 | Successful |
| Random Excursions | 0.365752 | Successful |
| Random Excursions Variant | 0.493417 | Successful |

**Figure 4**: Results of repeated NIST tests after XORing

## 4. Results

According to the test results, Random numbers produced with Runge Kutta 4 have successfully passed the NIST-800.22 ver 1a tests, known as randomness tests. There are many random number generation methods in the literature. The more chaotic, the greater the randomness and predictability. It is possible to generate random numbers with chaotic maps, as well as generating random numbers based on mathematical foundations, and the results are successful. Moreover, RK4, which can be proved mathematically, is one of the leading methods that can be proved mathematically in random number generation methods in the literature with these features.

In future studies, the randomness of numbers obtained by mathematically generated RK4, RK5, Galois Field (Irreducible Polynomial) methods and which method is more efficient can be investigated.

## References

[1] E. M. Esin, "Knutt / Durstenfeld Shuffle Algoritmasının Resim Şifreleme Amacıyla Kullanılması," Politek. Derg., vol. 12, no. 3, pp. 151–155, 2009, doi: 10.2339/2009.12.3.

[2] A. Akgül, M. Z. Yıldız, Ö. F. Boyraz, E. Güleryüz, S. Kaçar, and B. Gürevin, "Microcomputer-based encryption of vein images with a non-linear novel system," J. Fac. Eng. Archit. Gazi Univ., vol. 35, no. 3, pp. 1369–1385, 2020, doi: 10.17341/GaziMfd.558379.

[3] H. Li, L. Deng, and Z. Gu, "A Robust Image Encryption Algorithm Based on a 32-bit Chaotic System," IEEE Access, vol. 8, pp. 30127–30151, 2020, doi: 10.1109/ACCESS.2020.2972296.

[4] S. Chen, X. X. Zhong, and Z. Z. Wu, "Chaos block cipher for wireless sensor network," Sci. China, Ser. F Inf. Sci., vol. 51, no. 8, pp. 1055–1063, 2008, doi: 10.1007/s11432-008-0102-5.

[5] X. Tong, Z. Wang, Y. Liu, M. Zhang, and L. Xu, "A novel compound chaotic block cipher for wireless sensor networks," Commun. Nonlinear Sci. Numer. Simul., vol. 22, pp. 120–133, 2015.

[6] Z. Liu et al., "Color image encryption by using Arnold transform and color-blend operation in discrete cosine transform domains," Opt. Commun., vol. 284, no. 1, pp. 123–128, 2011, doi: 10.1016/j.optcom.2010.09.013.

[7] Y. Zhang, "The unified image encryption algorithm based on chaos and cubic S-Box," Inf. Sci. (Ny)., vol. 450, pp. 361–377, 2018, doi: 10.1016/j.ins.2018.03.055.

[8] R. U. Ginting and R. Y. Dillak, "Digital color image encryption using RC4 stream cipher and chaotic logistic map," 2013 Int. Conf. Inf. Technol. Electr. Eng., pp. 101–105, 2013.

[9] A. Jolfaei and A. Mirghadri, "Image Encryption Using Chaos and Block Cipher," Comput. Inf. Sci., vol. 4, no. 1, pp. 172–185, 2010, doi: 10.5539/cis.v4n1p172.

[10] M. Khan, T. Shah, and S. I. Batool, "Construction of S-box based on chaotic Boolean functions and its application in image encryption," Neural Comput. Appl., vol. 27, no. 3, pp. 677–685, 2016, doi: 10.1007/s00521-015-1887-y.

[11] Y. Liu, X. Tong, and J. Ma, "Image encryption algorithm based on hyper-chaotic system and dynamic S-box," Multimed. Tools Appl., vol. 75, no. 13, pp. 7739–7759, 2016, doi: 10.1007/s11042-015-2691-5.

[12] I. Cicek, A. E. Pusane, and G. Dundar, "A novel design method for discrete time chaos based true random number generators," Integr. VLSI J., vol. 47, no. 1, pp. 38–47, 2014, doi: 10.1016/j.vlsi.2013.06.003.

[13] K. I. Farhana Sheikh Leonel Sousa, Ed., "Circuits and Systems for Security and Privacy," .

[14] A. Akgul and I. Pehlivan, "A New Three-Dimensional Chaotic System Without Equilibrium Points, Its Dynamical Analyses and Electronic Circuit Application," Teh. Vjesn., vol. 23, pp. 209–214, 2016, doi: 10.17559/TV-20141212125942.

[15] N. Munir, M. Khan, T. Shah, A. S. Alanazi, and I. Hussain, "Cryptanalysis of nonlinear confusion

component based encryption algorithm," Integration, vol. 79, no. February, pp. 41–47, 2021, doi: 10.1016/j.vlsi.2021.03.004.

[16] H. G. Mohamed, D. H. Elkamchouchi, and K. H. Moussa, "A novel color image encryption algorithm based on hyperchaotic maps and mitochondrial DNA sequences," Entropy, vol. 22, no. 2, pp. 7279–7297, 2020, doi: 10.3390/e22020158.