



A new key generator algorithm based on sine map and SHA-3 for image encryption

Firat Artuğer¹ · Fatih Özkaynak²

Received: 2 April 2024 / Revised: 22 October 2025 / Accepted: 20 November 2025
© The Author(s) 2026

Abstract

In recent years, the need for secure cryptographic keys for various purposes has been constantly increasing. One of these areas is undoubtedly image encryption. In this study, a new key generation algorithm has been developed for use in image encryption algorithms. The proposed generator has a hybrid structure based on the chaotic sine map and the SHA-3 algorithm. It is aimed to produce strong keys by using the benefits of these two structures. To do this, first, bit sequences of the desired length are generated using the basic parameters of the sine map. Then, bit sequences of the same length are obtained using the SHA-3 algorithm. These two bit sequences are then subjected to an XOR operation to generate key values of the desired length. With this structure, 3 different bit sequences of 1 million length were produced. All of these bit sequences successfully passed NIST SP 800–22 tests. This key value obtained in this study has been successfully applied to a new image encryption algorithm. The results of the study demonstrate that the proposed algorithm meets important cryptographic criteria such as information entropy, correlation-histogram analysis, NPCR, and UACI values. In addition, the simple and fast structure of the algorithm in key generation was the most important difference from other studies.

Keywords Key generator · Sine map · SHA-3 · Image encryption · RNG

1 Introduction

In today's information age, new requirements emerge every day. One of these is random numbers. Random numbers are needed in many areas. For this reason, the concept of random number generator (RNG) emerged. RNGs can be an important building block of any

✉ Firat Artuğer
firartugur@munzur.edu.tr
Fatih Özkaynak
ozkaynak@firat.edu.tr

¹ Department of Computer Engineering, Munzur University, Tunceli 62200, Türkiye

² Department of Software Engineering, Firat University, Elazığ 23119, Türkiye

algorithm or application. That's why it continues to be a very important and hot topic. Researchers have been trying to design new RNGs for a long time. Especially in recent years, with the rapid development of technology, RNGs have gained great importance. Now, random numbers appear at every point in our lives. These structures are frequently used, especially in cryptography. In addition, they are frequently used in many fields such as simulation, modeling, physics, and biology. These techniques are very important, especially for the application of Monte Carlo techniques and stochastic simulations [1]. It is possible to examine RNG structures under two groups. These are hardware-based RNGs and software-based RNGs. Hardware-based RNGs generate random numbers using a hardware resource, such as processor times, mouse movements, radioactive decay, and thermal noise. Many hardware resources can be used here. This is one of the most important advantages of these structures. However, they are often complex and difficult to use. On the other hand, these generators show strong randomness properties because they are fed directly from a physical source. Therefore, their use is quite common, especially in cryptographic applications. One of their disadvantages is that when the power is cut off, the process of generating random numbers will stop. Hardware-based RNG structures are called true random number generators (TRNG).

Software-based generators, on the other hand, generate random numbers directly with algorithms and mathematical models without the need for a hardware resource. The use of software-based generators is more common than hardware-based methods. Because it will be very easy and fast to obtain. At this point, many algorithms and mathematical structures can be easily developed. However, software-based generators cannot obtain as secure number sequences as hardware-based generators. Because mathematical structures are used here. Since the same structures will result in the same number sequence, care must be taken when using them in security applications. Predicting or retrieving RNGs in a system makes it easier for an attacker to exploit vulnerabilities in the system [2]. For this reason, complex and unpredictable structures must be used when designing a software-based RNG. Software-based random number generators are called pseudorandom number generators (PRNG). PRNGs are structures that are initialized with a value called seed, and a random value is obtained each time the algorithm runs. Seed value is very important here. Because when the same seed value is used, the same number sequence is obtained. Therefore, an effective seed value must be selected. This structure is often used in modeling and simulation fields. Because in these areas, situations that often require repetition may arise. This means that the same sequence of numbers may need to be obtained many times. However, this structure may pose various risks for cryptographic applications. Therefore, it would be more appropriate to develop a PRNG in these structures without using seeds. Thus, a different sequence will be obtained each time the algorithm runs. For example, when a key is to be generated, that key value can be generated only once. In recent years, PRNG structures have been developed using many different techniques such as chaotic systems, machine learning, physical randomness, and mathematical transformations. The increasing complexity of these structures has led to safer random numbers. Some of these studies are briefly mentioned below.

A new PRNG structure was developed on FPGA using MHNN (memristive Hopfield neural network). The results obtained here have completed NIST SP800-22 tests and have shown that it can generate random numbers [3]. Chaotic maps were used to develop a new encryption structure. These chaotic maps have an adaptive symmetry structure. It has been

observed that the pseudo-random number generator developed using the Zaslavsky map works effectively [4]. A new TRNG structure based on electrolyte-controlled field effect transistors was developed. Since most of the ones obtained in this structure are obtained with random noise in the circuit, it provides a high degree of randomness. Additionally, looking at the results obtained in this study, it can be seen that all NIST tests were completed successfully [5]. F2-linear pseudo-random number generators, which are popular due to their high speed, easy construction with large state spaces, and provable theoretical properties, are discussed. However, such generators have linear art effects that fail statistical tests of linearity. Therefore, two new hand-crafted F2-linear transformations are proposed, providing linear transformations that have good statistical properties and can be efficiently programmed on exponential processors or directly in hardware [6].- A new hardware-based scheme has been developed by combining the benefits of Tent map and Bernoulli maps. HUB (Half-Unit-Biased) format is used in this scheme. This method has been shown to have a particularly significant impact on performance and to be able to obtain sufficiently random numbers - [7]. The Learning with Rounding (LWR) problem, a Lattice-based cryptographic challenge, is difficult for quantum computers to solve and could form a secure seed for the Quantum-Safe Pseudo-Random Number Generator (QSPRNG). This work involves building a QSPRNG to generate a sequence of pseudorandom bits via the Linear Feedback Shift Register (LFSR). Here, QSPRNG uses a secure seed as input and uses a Homomorphic Function to preserve the security of LFSRs' internal states [8]. EE (Elliptic Curves) encryption structures can be used in PRNG design. It has been observed that this EE-based PRNG structure developed by the researchers completed the NIST tests. Additionally, in this study, the results were tested on an image encryption algorithm, and it was confirmed that the algorithm worked effectively [9]. A subjective mapping that provides Li-Yorke chaos per unit area has been created, and a permutation algorithm that increases randomness by mixing parameters and inputs with another high-dimensional chaos has been proposed. Thus, the chaotic system created is made more random, and the key space is larger. By designing two system fusion algorithms, the output distribution of the system was improved, and finally, a compound chaotic system was obtained. A new pseudo-random number generation scheme using the compound chaotic system is presented. The proposed pseudo-random number generator (PRNG) has been evaluated using various testing tools and methods, such as NIST SP 800 – 22 test. The test results are promising because the proposed PRNG passed all these tests successfully [10].

A new PRNG structure that displays large periods has been developed. Arnold cat map was used in the development of this system. Additionally, this PRNG increases the period by using an LFSR structure [11]. A PRNG was developed with fixed-point arithmetic with a chaotic map without fixed points, optimized by the differential evolution algorithm. TestU01 tests were applied to these series, and all tests were completed successfully. Thus, this algorithm has shown that it can generate sufficiently random numbers [12]. Anti-aging controller TRNG structure is proposed. The implementation of this study can be done by obtaining random numbers through simulations using TANNER EDA Tools and ModelSim-Altera 6.4a (Quartus-II 9.0) [13]. A PRNG is designed using a prime number and some primitive roots. The PRNG has been validated for use in cryptographic applications using histograms, correlation coefficients, and the NIST statistical test suite. Later, PRNG was used in an image encryption system, and system security was tested using statistical metrics, difference attack metrics, and sensitivity to a bit change [14]. A new chaos-based TRNG

structure called SMC (sliding mode controller) has been developed. Additionally, this study was carried out on FPGA [15]. A new 2D discrete hyper-chaotic map with a linear cross-linked topological structure combined with Tent and Logistic maps was created. This map outperforms previous chaotic maps. Then, an efficient PRNG based on the proposed map was implemented on the FPGA. The sequences produced by this PRNG have a high level of randomness and have successfully passed the TestU01, Dieharder, and National Institute of Standards and Technology (NIST) SP800-22 test suites [16]. A new FPGA-based TRNG structure is proposed using dynamic key-based s-box architecture. This work has been shown by NIST tests to effectively generate random numbers [17]. A new PRNG structure has been proposed using one-dimensional chaotic maps [18]. A new PRNG structure was designed using the Hopfield Neural Network called VOFHNNUER. NIST testing was used to test the data. The outputs of the study completed all NIST tests. Additionally, the PRNG structure created in this study was applied for encryption and obfuscation. Here, an image is first encrypted and then hidden inside another image [19]. A new PRNG structure is proposed using the MHOS (Memristive Helmholtz snap) oscillator. Thanks to the chaotic behavior exhibited by the MHOS architecture, an effective PRNG structure was obtained. This PRNG has completed the NIST-800-22 test suite [20]. Based on the choice of PRNG, it was intended to ensure security through encryption of multimedia images between users. The work involves the use of various PRNG techniques between sequential processes of displacement and placement crypto operations and performs these operations using a secure process. By testing different PRNGs for encoding grayscale and color RGB images, they can be compared with other existing similar approaches. Experiments aim to investigate suitability and reliability using security benchmark standard parameters. Research shows that appropriate PRNG selection is an effective approach for image encryption [21]. A new method for generating random numbers using a modified robust logistic map (MRLM) is presented. The results obtained have completed the NIST tests [22]. A new PRNG scheme has been developed using the STM (Skew tent map). This work can generate highly random numbers of thanks to the randomness of the chaotic map. This work has also been shown to get numbers quickly. This PRNG has proven its effectiveness by completing many tests [23]. A new random number generator with a fractal-order chaotic Chua structure has been developed. This study shows that it works effectively because of the analysis. Additionally, this study showed that Fractal order chaotic systems can also be used, which can be a new alternative to discrete-time chaotic dynamics [24].

The scheme proposed in this study offers two important contributions. The first of these is to propose a new key generator. This scheme is an important algorithm that can be used especially for systems working with one-time passwords in cryptographic applications. Another important contribution of the study is the successful application of the strong key structure obtained from a new image encryption algorithm and the analysis of the results in detail. The proposed generator has completed NIST tests and effectively encrypted images in the image encryption algorithm. Other sections of this study are organized as follows. In the second section, the key planning algorithm is explained in detail. In the third section, the image encryption algorithm is given. In the fourth section, encryption results are analyzed. In the fifth section, the results are discussed.

2 Keyplanning algorithm

Today, an important component of cryptographic algorithms used to ensure security is the key planning process. In this context, designing more secure and efficient key planning algorithms is vital for information security. In this section, the proposed key generator algorithm is explained. Since the algorithm is based on the Sine map and SHA-3 algorithm, these structures are mentioned first.

2.1 Chaotic sine map

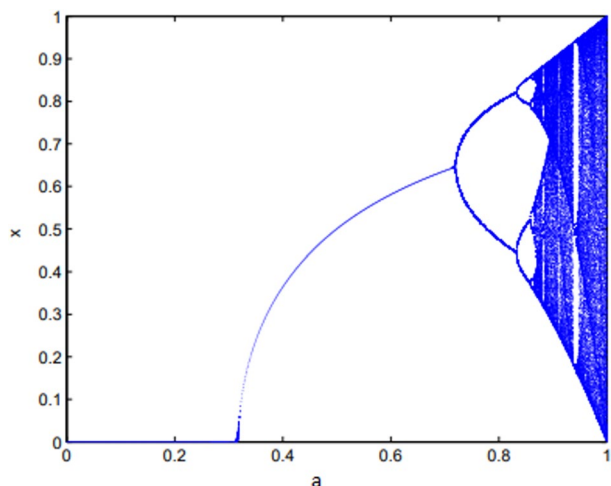
In the proposed generator algorithm, both the seed value and the 256-bit value that will enter the XOR process in the last stage are obtained with a sine map. The mathematical model of the chaotic sine map is given in Eq. 1. Additionally, the bifurcation diagram of the sine map is shown in Fig. 1. Thanks to the bifurcation diagram, the chaoticness, stability, etc. of the system can be determined. Comments can be made about behavioral characteristics. It is seen that the sine map used shows chaotic behavior with the given parameters. In this study, the range in which the sine map shows chaotic behavior, given in Eq. 1, was used. Furthermore, when the bifurcation diagram is examined, it is understood that the x values obtained when the constant a is chosen, especially after the value of 0.8, show a chaotic distribution.

$$x_{n+1} = a \sin(\pi x_n), \quad x_n \in [0,1], \quad a \in [0.85,1] \quad (1)$$

2.2 SHA-3

SHA-3, also known as Keccak, is a hashing algorithm. In other words, regardless of the length of the data, it summarizes that data into a fixed-length output. The most important feature of these algorithms is that they are one-sided. In other words, only a summary of the data can be made. However, original data is not obtained from summary data. In this way, it provides many advantages, especially for key generators. However, the most important point to note here is that the hash value of the same data will always be the same.

Fig. 1 Bifurcation diagram of the sine map



The SHA-3 algorithm is a secure hash algorithm designed by Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche [25]. Published by NIST on August 5, 2015. This algorithm has a structure that can obtain outputs of 128, 224, 256, 384, or 512 lengths according to bit values. This is a very important advantage that shows the flexibility of the algorithm and the fact that different outputs can be used according to need. Before the SHA-3 algorithm, SHA-0, SHA-1 SHA-2 algorithms were used. The SHA-0 algorithm was designed in 1993, but due to the security vulnerability found, it was updated to the SHA-1 algorithm in 1995. The SHA-1 algorithm was frequently used in various applications, especially e-signature. SHA-1 was replaced by the SHA-2 algorithm as a result of the security vulnerabilities found. The most important difference of SHA-2 was that it could produce outputs of different lengths. However, NIST announced in 2006 that a new hash algorithm was needed to be an alternative and organized a competition open to everyone in 2007. As a result of the competition, 5 candidate algorithms made it to the finals. These are Keccak, JH, Skein, Grostl, and Blake algorithms. As a result of the competition, Keccak achieved various advantages over its competitors and received the new SHA-3 title. The basic structure of the SHA-3 algorithm is given in Fig. 2.

2.3 Proposed key generator

In the proposed key generator algorithm, a similar seed value is first generated by using the initial parameters of a chaotic map. In this study, sine map was used for this purpose. The mathematical expression and initial parameters of the sine map are given in Eq. 1. This seed value is then given as input to the 256-bit output producing unit of the SHA-3 algorithm. The SHA-3 algorithm obtains a unique output of 256 bits from the chaotic seed value it takes as input. Then, another 256-bit long output is obtained with a chaotic map. In this study, sine map was used for this purpose. XOR operation is applied to the 256-bit values obtained from here and the 256-bit values obtained in the SHA-3 output, and the result is saved. In this way, the algorithm is repeated until the desired bit lengths are obtained. The flow diagram of the proposed algorithm is given in Fig. 3.

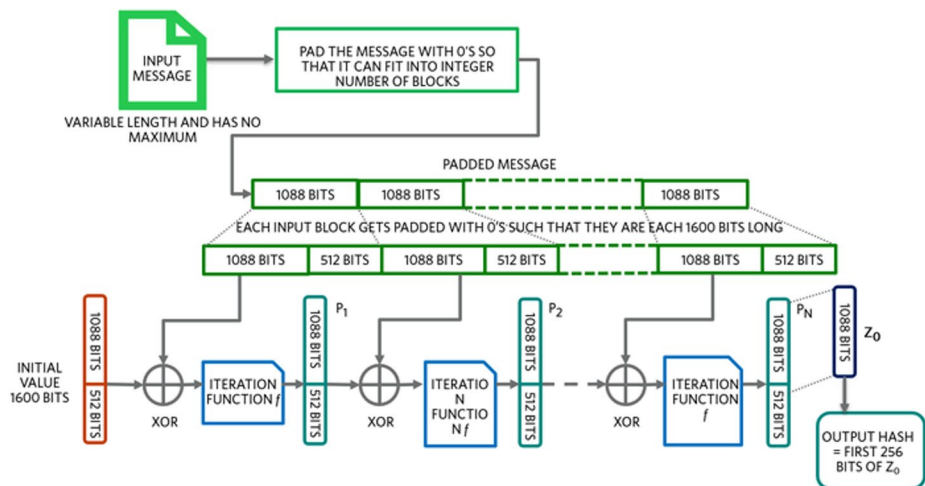


Fig. 2 The basic structure of the SHA-3 algorithm

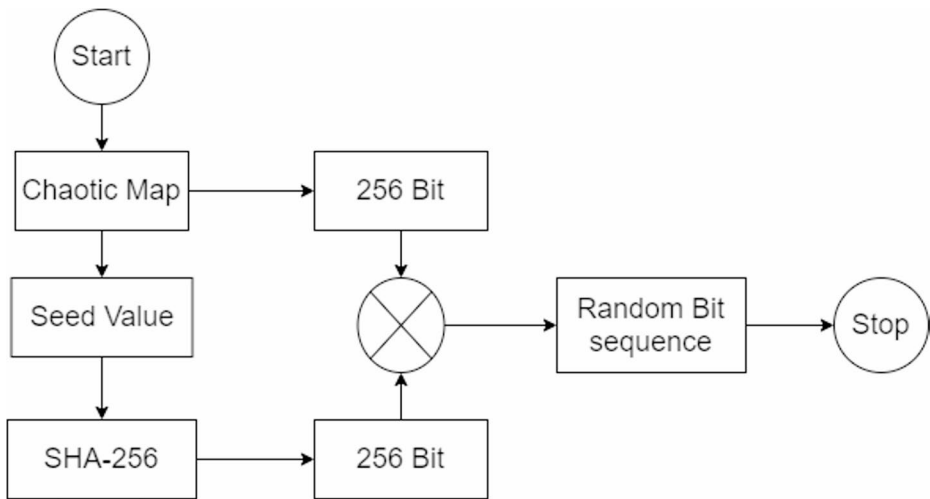


Fig. 3 Flowchart of the proposed key generator algorithm

Table 1 Python code of the algorithm that outputs a random 256-bit length

```

bit_sequence="" // 256-bit random sequence
for i in range(0,256):

    a=0.8946545468468 // randomly selected "a" constant
    x = random.random() // random value
    x_new=a*(math.sin((math.pi)*x)) // New x value generated by sine map
    x_new=(int)(10000000000*x)%256 // x value rounded to the [0-256] range

    if (x_new <128):
        bit_sequence+="1"
    else:
        bit_sequence+="0"
print (bit_sequence)
  
```

The Python code of the proposed algorithm for the 256 bits obtained in the SHA-3 output and the random 256 bits that will be XORed is given in Table 1. Here, firstly, a value is chosen within the range where a value shows random properties. Then, a random x value between 0 and 1 was chosen. After these values were taken, the mathematical formula of the sine map was applied, and a new x value was obtained. Since a total of 256 bits will be obtained, mod256 operation was used in this study. The new x value produced is multiplied by a large number and converted to an integer. Then, mod256 operation is applied to this value. If this value is less than 128, “1” is added; otherwise, “0” is added, and a random value of 256 bits is obtained when the algorithm is run for 256 iterations. Since this algorithm runs in a single cycle, it has a very fast structure with $O(n)$ complexity. This demonstrates the importance of the proposed algorithm for key generation in applications requiring high speed.

In this study, a key value of 256×8 bit length was produced and used. For this, it will be sufficient to rotate the algorithm 8 times. In this way, the proposed method gives very fast results. To understand that the key values obtained were obtained randomly, 3 different data sets with a length of 1 million were produced. NIST SP 800 – 22 [26] testing was applied to these produced data sets. All data sets have completed NIST tests. A generator that successfully passes NIST tests is a reliable option in terms of accuracy and high random distribution of random numbers. This result provides strong evidence about the unpredictability and unrepeatability of the generator. Thus, these PRNG structures to be used in encryption algorithms will form the basis of security in both data transmission and storage. NIST test results of bit sequences obtained with the proposed algorithm are presented in Fig. 4.

3 Image encryption algorithm

Image encryption has been one of the most used methods for securely storing and transmitting images from the past to the present. Sensitive images or images containing personal confidential information must be protected with effective image encryption algorithms. This has now become an indispensable part of our lives. Even if these sensitive images are captured by attackers, they are made incomprehensible thanks to encryption algorithms. In other words, the attackers obtain a meaningless image. Many approaches have been suggested for effective encryption of images. Latin square and machine learning [27], neural network [28], matrix semi tensor product theory [29], compressive sensing [30, 31], DNA coding [32–36], genetic operations [37–39], 2D hyperchaotic system [40], bitplane scrambling [41], strong S-box [42–52], optimization [53, 54], data compression [55], 5D memristive exponential hyperchaotic system [56], quantum structures [57], a discrete space map [58], chaotic system [59, 60] approaches are frequently used.

Image encryption algorithms usually encrypt the image with a key. Only people who have the key can decode the image. For this, it is very important to use effective switches.

Key Generator Algorithm		1 Million bit Dataset-1	1 Million bit Dataset-2	1 Million bit Dataset-3
No	Test	p-value	p-value	p-value
1	Frequency Test (Monobit)	0.23789	0.88210	0.34012
2	Frequency Test within a Block	0.81446	0.28272	0.85938
3	Run Test	0.59443	0.71829	0.36339
4	Longest Run of Ones in a Block	0.39885	0.32571	0.52092
5	Binary Matrix Rank Test	0.39350	0.37473	0.26078
6	Discrete Fourier Transform (Spectral) Test	0.72197	0.60115	0.76541
7	Non-Overlapping Template Matching Test	0.45869	0.22734	0.83624
8	Overlapping Template Matching Test	0.22364	0.24802	0.24279
9	Maurer's Universal Statistical test	0.84259	0.70153	0.01380
10	Linear Complexity Test	0.41765	0.70564	0.51396
11	Serial test	0.64503	0.69147	0.38502
12	Approximate Entropy Test	0.31595	0.49148	0.82975
13	Cummulative Sums (Forward) Test	0.34717	0.54740	0.54396
14	Cummulative Sums (Reverse) Test	0.34844	0.68119	0.32548
15	Random Excursions Test (State: +4)	0.63818	0.72984	0.90435
✓	Number of Successful Tests	15	15	15

Fig. 4 NIST test results of bit sequences obtained with the proposed algorithm

Encryption techniques are generally examined in two ways. The first one is block cipher structure. In these structures, data is divided into blocks, and each block is encrypted within itself. Then, the encrypted blocks are combined to obtain encrypted data. The other technique is stream encryption. Here, data is encrypted by XORing with one key at a time. Chaotic systems and complex mathematical structures are generally preferred in image encryption structures. In addition, image encryption algorithms have been tried to be made safer and more effective, especially in recent years, by using artificial intelligence techniques. Image encryption plays an important role not only in the security of personal information but also in military and commercial fields. Therefore, the development and improvement of image encryption technologies emerge as an important issue in the field of information security.

In this study, a bit-based image encryption algorithm is proposed with the key obtained by the proposed key generator. Bit-based encryption is a fast and effective encryption method, especially for applications on hardware devices. However, to ensure adequate security, keys must be secured and replaced frequently. The general architecture of the proposed algorithm is given in Fig. 5. Here, an image is first separated into its layers. An image consists of red, green, and blue layers. Each of these layers is made into an array. A key value is produced with the proposed PRNG as long as the length of these sequences. For example, if the size of the image is 256×256 , a key of length 65,536 is created. These pixel values are then XORed with the key value, respectively. In this way, after all layers are passed through the XOR function, the layers are combined, and the encrypted image is obtained. Here, the XOR function guarantees security with a secure key value. Another advantage is that decryption can be easily carried out. To decrypt the encrypted data with the XOR function, this time, the encrypted data and the key value are passed through XOR, and a clear image is obtained. The only unknown element here is the key value. The key must be random and of sufficient length. Mathematical expressions showing the encryption and decryption of data with XOR are given below.

Encryption: $A \text{ XOR } B = C$.

Decryption: $C \text{ XOR } B = A$.

Here, B represents the key, A represents the clear text, and C represents the ciphertext.

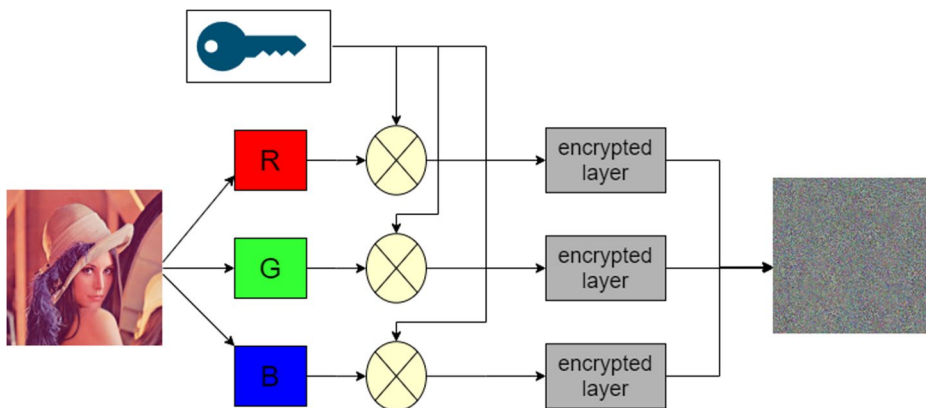


Fig. 5 Proposed image encryption algorithm

4 Analysis results

Image encryption plays a crucial role in many application areas, particularly in secure data transmission and storage. Many different encryption algorithms have been developed for this purpose. However, analysis of these algorithms is ongoing due to existing security vulnerabilities and weaknesses. In this study, a new image encryption algorithm is proposed and analyzed. The proposed algorithm is believed to have potential for use in applications such as secure data transmission and storage. In this study, four different images were encrypted using the key value obtained by the proposed generator algorithm. The original, encrypted, and decrypted versions of these images are presented in Fig. 6.

4.1 Histogram analysis

A histogram is a tool that graphically displays the distribution of color components of an image. Histogram analysis reveals how frequently the color components of the image are found in which ranges. This analysis provides a lot of information about the quality of the image. For example, if the histogram of an image is concentrated on the left side, a large portion of the tones in that image may be dark tones, indicating that the image is low contrast. The presence of density in the middle region of the histogram indicates that the image has a balanced tonal distribution. Histogram analysis also shows how much of an image's light and dark areas are separated by sharp or smooth transitions. This information can provide important clues for editing or enhancing an image. Graphs showing the histograms of the images encrypted with the proposed image encryption algorithm and the original images are given in Figs. 7, 8 and 9, and 10. As shown in these graphs, various inferences can be made because the histograms of the original images show clusters in certain regions. However, since the encrypted images have a balanced distribution, it would be very difficult for attackers to make these inferences.

4.2 Correlation analysis

Correlation graphs of original and encrypted images are given in Figs. 10, 11 and 12, and 13. A correlation plot shows the similarity of a particular aspect of an image to pixels in other directions. Thus, it is shown what the relationship between the pixels of the image is. If this graph reveals a high correlation, it means there is high similarity between pixels. That means there are similar details in those regions. However, if the correlation is low, there is little similarity between pixels. This means that the details in that part of the image are different. As a result, the correlation graph is an important technique frequently used in the field of image processing to understand similar and different points of an image. When Figs. 11, 12 and 13, and 14 are examined, the shapes in the correlation graphs of the original images are clearly recognizable. However, when looking at the correlation graphs of the encrypted images, a graph with a uniform distribution and very difficult to extract is observed.

4.3 Information entropy analysis

One of the methods that measures the distribution of information contained in an image is information entropy analysis. Here, separate calculations are generally made for the R, G,

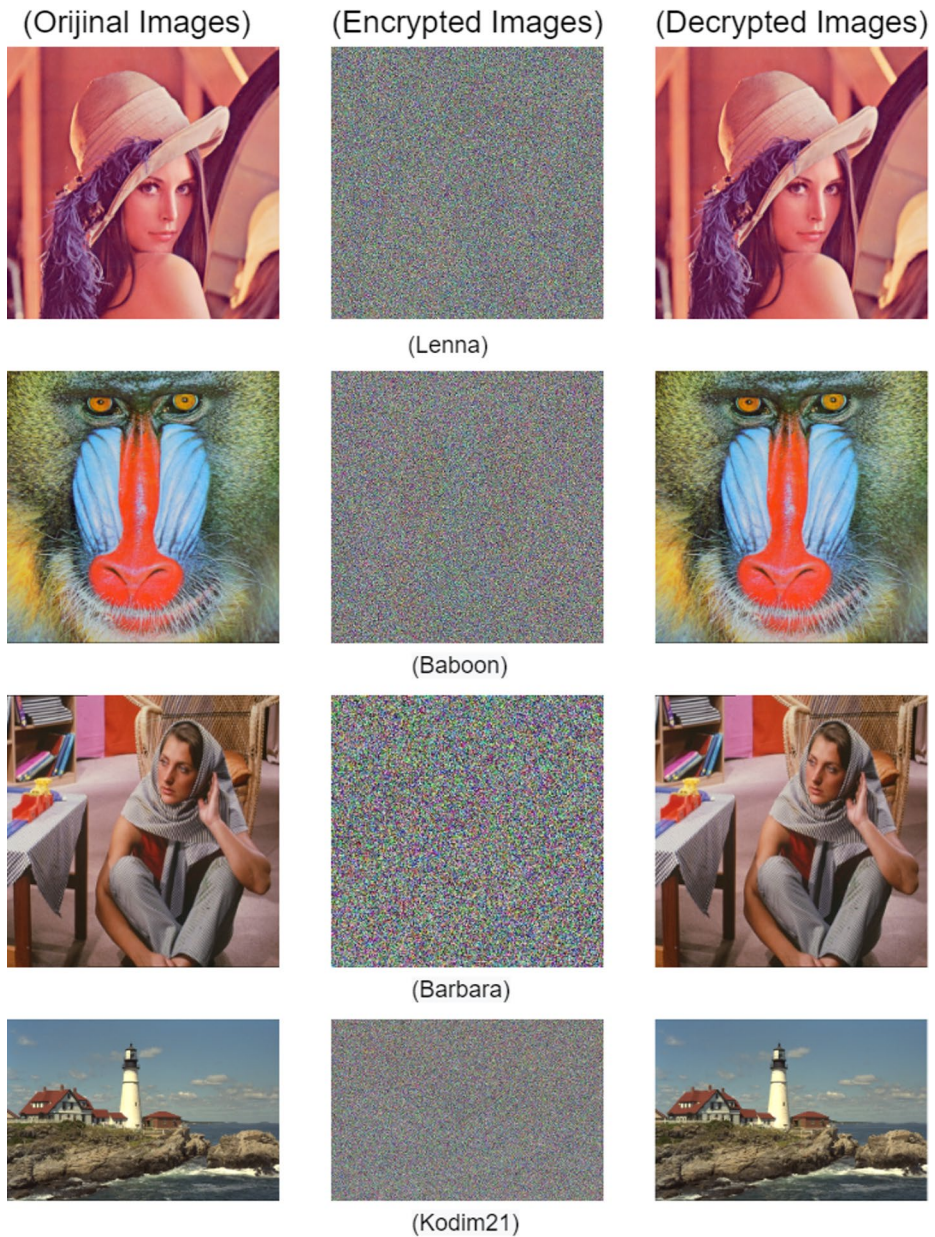


Fig. 6 Original, encrypted, and decrypted images

and B layers. If the entropy of a color layer is high, that layer carries more information. So, in an image, for example, the entropy of the R layer may be higher than the G and B layers. In this case, the red color shows more variability in the image. Conversely, if the information entropy of a layer is low, that layer carries less information. Thanks to information entropy analysis, we can have information about the information capacity of an image's

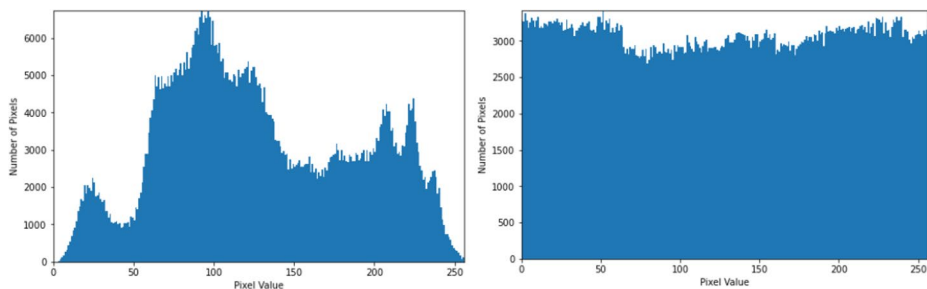


Fig. 7 Histogram graphs for the original and encrypted Lena image

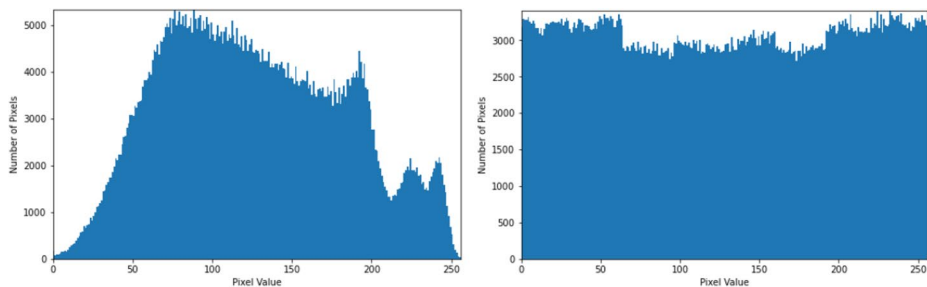


Fig. 8 Histogram graphs for the original and encrypted Baboon image

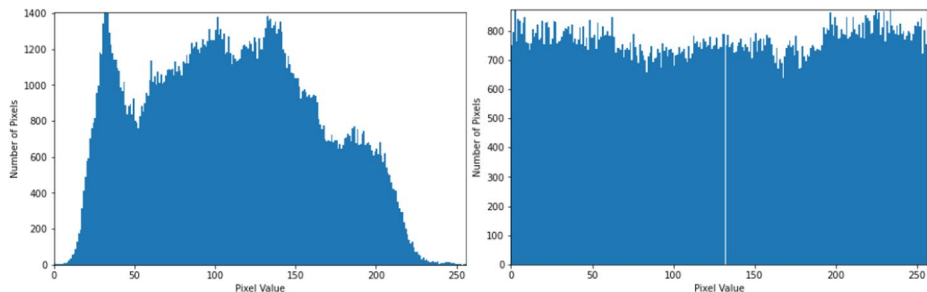


Fig. 9 Histogram graphs for the original and encrypted Barbara image

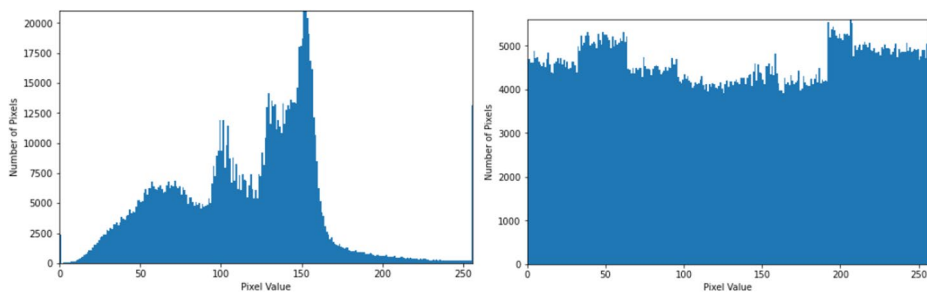


Fig. 10 Histogram graphs for the original and encrypted Kodim21 image

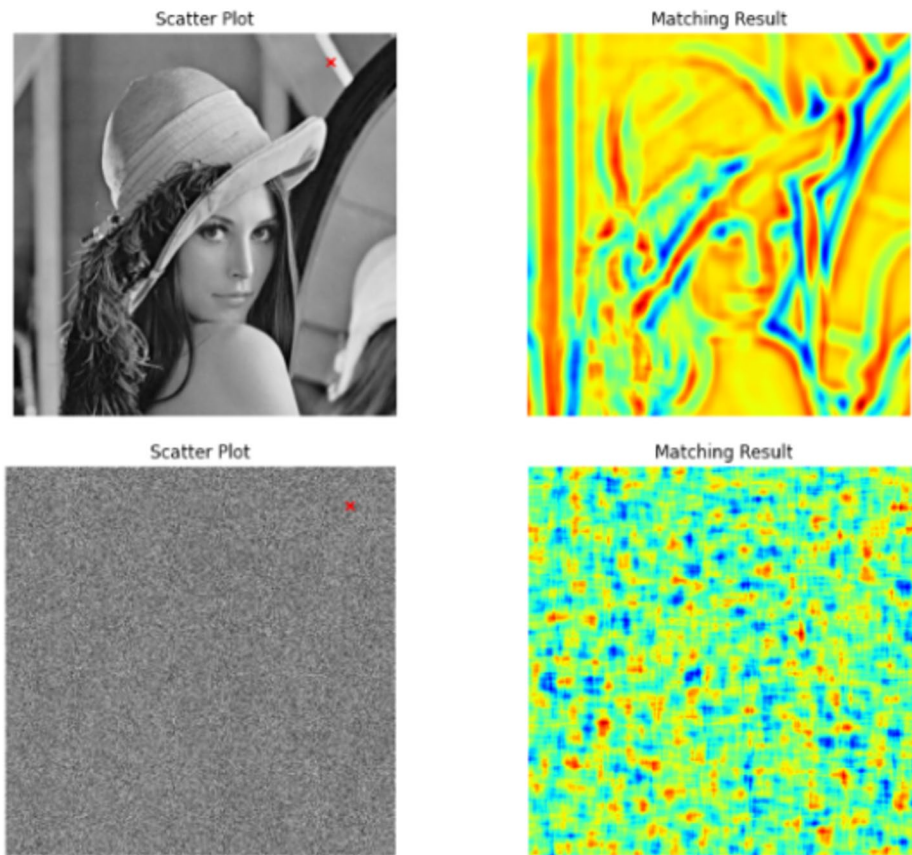


Fig. 11 Correlation graphs for the original and encrypted Lena image

colors. This helps us obtain important information through encryption, compression, or different processing methods of images.

Analysis of information entropy in an encrypted image depends on the strength of the encryption algorithm. If this encryption algorithm is based on strong foundations, that is, if it uses a strong key value, the pixel values in the encrypted image will show a random distribution. In this case, the entropy of the image will be maximum, since the relationship between pixel values or repeated patterns will disappear. However, if a weaker encryption algorithm or repeated keys are used, the pixel values in the encrypted image may not differ from the values in the original image, in which case the entropy value will be low. Looking at the literature, this value is expected to be close to 8. Information entropy values of images encrypted with the proposed method are given in Table 2. It is seen that all of these values are very close to 8 and meet this criterion.

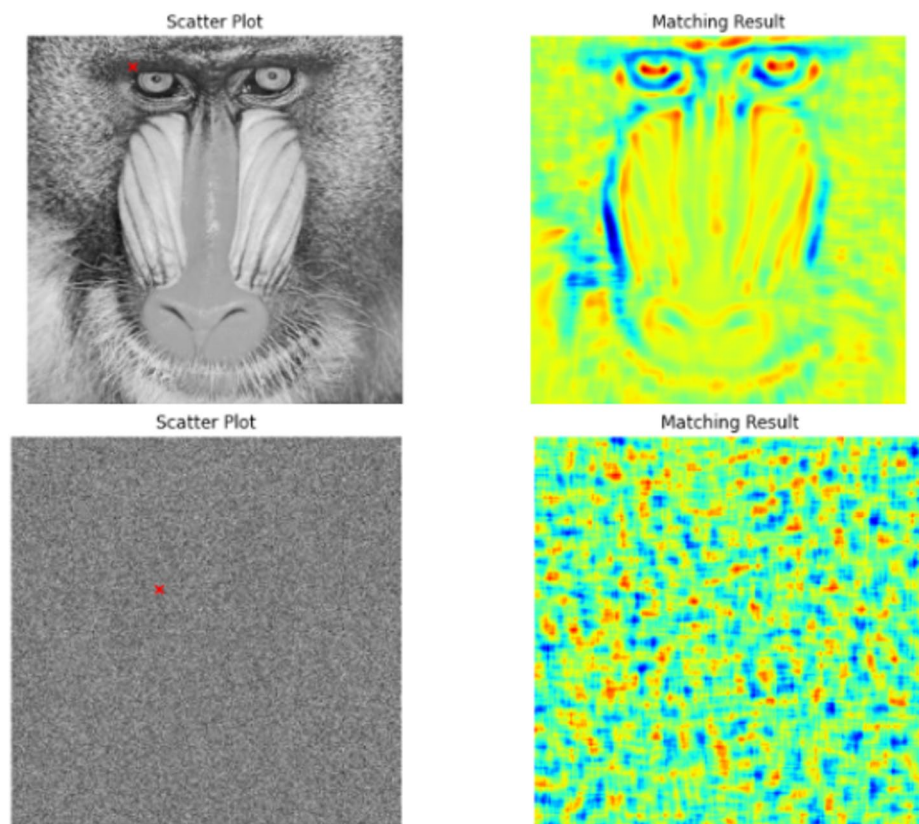


Fig. 12 Correlation graphs for the original and encrypted Baboon image

4.4 NPCR (Number of pixel change Rate) and UACI (Unified average changing Intensity) analysis

NPCR and UACI are metrics used to measure the performance of image encryption algorithms. NPCR measures the rate of change of pixels in an encrypted image. The higher the NPCR value, the higher the security of the encryption algorithm will be. If the NPCR value is high, it means that every pixel in the encrypted image is different from each other. This shows that encryption is more secure. The NPCR value should generally be 99% or higher. UACI measures the variation density of an encrypted image. The higher the UACI value, the higher the security of the encryption algorithm will be. If the UACI value is high, it means that the majority of pixels in the encrypted image have changed. This shows that encryption is more secure. NPCR and UACI values of images encrypted with the proposed image encryption algorithm are given in Table 3. It is seen that these values provide average values.

The results obtained for the Lena image were compared with some other studies in literature, and the results are presented in Table 4. These results indicate that the proposed method produces results like those of other studies. Because the primary goal of this study is to generate effective keys, a basic image encryption approach was used as an example.

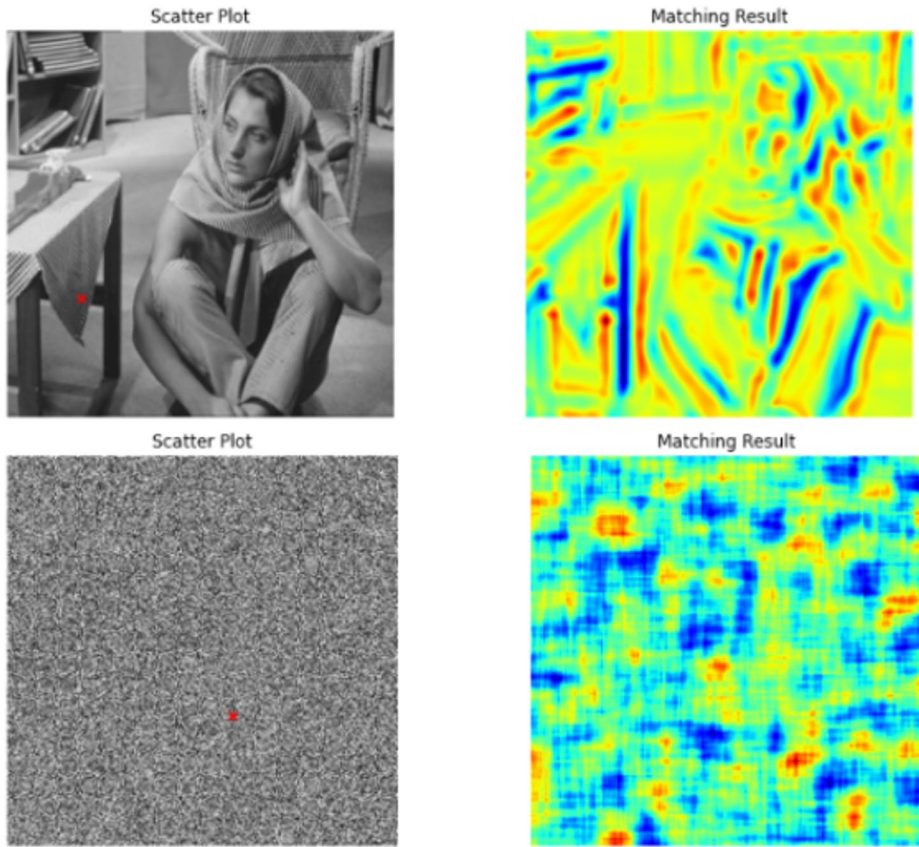


Fig. 13 Correlation graphs for the original and encrypted Barbara image

Despite this, the results are acceptable. However, more effective results could be achieved if this approach is used as a key generator in more complex image encryption algorithms in the future.

5 Conclusions

In this work, a new key generation algorithm was proposed and successfully applied to an image encryption algorithm. The proposed algorithm generates high security keys using the combination of Sine map and SHA-3 algorithms. To test such generators, bit strings usually 1 million long are obtained. Using the proposed key generator, 3 different bit strings with a length of 1 million were generated and tested. All of these produced bit sequences have successfully passed NIST SP 800–22 tests. Thus, strong keys for encryption algorithms can be obtained. In this study, a key image encryption algorithm obtained by the proposed method was used, and the encrypted images were subjected to various tests. These tests are information entropy, correlation-histogram analysis, NPCR, and UACI tests, which are among the

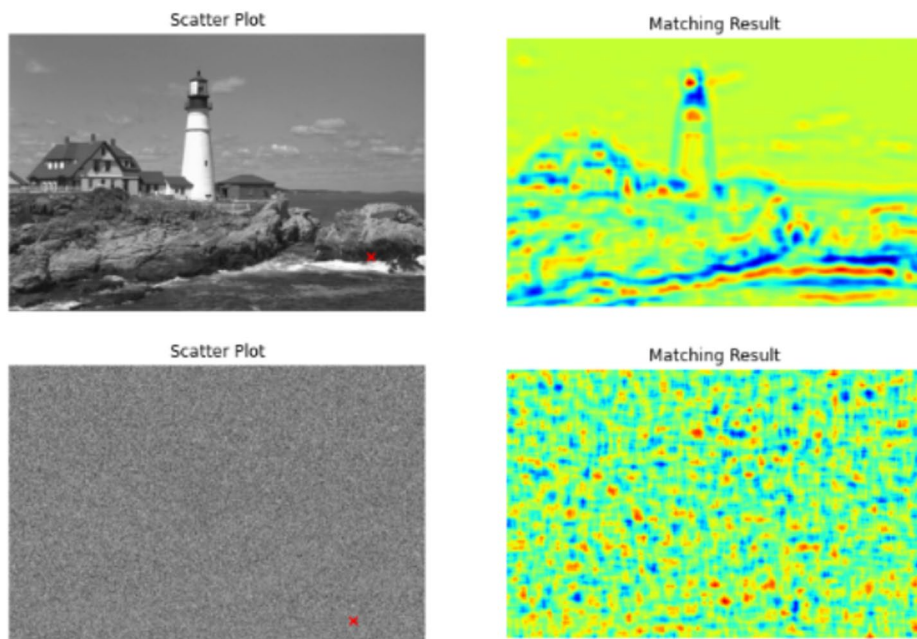


Fig. 14 Correlation graphs for the original and encrypted Kodim21 image

Table 2 Information Entropy values of encrypted images

Encrypted image	Information Entropy Value		
	<i>R</i>	<i>G</i>	<i>B</i>
Lenna	7.9971	7.9976	7.9909
Baboon	7.9970	7.9949	7.9980
Barbara	7.9950	7.9943	7.9958
Kodim21	7.9926	7.9920	7.9897

Table 3 NPCR and UACI tests

Image	NPCR	UACI
Lenna	0.9923	0.3186
Baboon	0.9923	0.3139
Barbara	0.9967	0.3150
Kodim21	0.9922	0.2990

most used tests in literature. In this study, it was observed that the encrypted images passed these tests successfully.

The proposed method demonstrates that hash algorithms can be used for key generation. At this point, older hash algorithms that are no longer used in the security field can also be used. Furthermore, the input of the hash algorithm is fed with a chaotic map to increase randomness. The values obtained from the chaotic map and the hash output are then subjected to an XOR operation to increase complexity. This allows for the generation of both fast and efficient random numbers. Based on these findings, the proposed algorithm is believed to

Table 4 Performance comparison for Lena image

Method	Information Entropy Value			NPCR	UACI
	R	G	B		
Proposed	7.9971	7.9976	7.9909	0.9923	0.3186
[27]	7.9987			0.9962	0.3347
[61]	7.9970	7.9973	7.9961	0.9962	0.3056
[32]	7.9993			0.9960	0.3347
[34]	7.9994	7.9994	7.9994	0.9960	0.3345
[48]	7.9971			0.9959	0.3344
[49]	7.9977			0.9961	0.3336
[57]	7.9992			0.9961	0.3346

serve as a roadmap for future key generators, random number generators, and encryption algorithms. In addition, the proposed algorithm can be integrated into IoT devices, one of the most popular areas today, and used in lightweight cryptography applications, with its simple and fast structure.

Author contributions F. A.: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Writing - review & editing F. Ö: Supervision Visualization, original draft, Writing - review & editing Funding acquisition, Resources, Writing - review & editing.

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK). No funding was received for conducting this study.

Data availability Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

Declarations

Competing of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Law AM, Kelton WD, Kelton WD (2007) Simulation modeling and analysis, vol 3. Mcgraw-hill, New York
2. Schindler W (2009) Random number generators for cryptographic applications. Cryptographic Eng, 5–23
3. Yu F, Zhang Z, Shen H, Huang Y, Cai S, Du S (2022) FPGA implementation and image encryption application of a new PRNG based on a memristive Hopfield neural network with a special activation gradient. Chin Phys B 31(2):020505
4. Tutueva AV, Nepomuceno EG, Karimov AI, Andreev VS, Butusov DN (2020) Adaptive chaotic maps and their application to pseudo-random numbers generation. Chaos Solitons Fractals 133:109615

5. Erozan AT, Wang GY, Bishnoi R, Aghassi-Hagmann J, Tahoori MB (2020) A compact low-voltage true random number generator based on inkjet printing technology. *IEEE Trans Very Large Scale Integr Syst* 28(6):1485–1495
6. Blackman D, Vigna S (2021) Scrambled linear pseudorandom number generators. *ACM Trans Math Softw (TOMS)* 47(4):1–32
7. Da Silva SS, Cardoso M, Nardo L, Nepomuceno E, Hübner M, Arias-García J (2023) A new Chaos-Based PRNG hardware architecture using the HUB Fixed-Point format. *IEEE Transactions on Instrumentation and Measurement*
8. Pandit AA, Kumar A, Mishra A (2023) LWR-based quantum-safe pseudo-random number generator. *J Inf Secur Appl* 73:103431
9. AbdElHaleem SH, Abd-El-Hafiz SK, Radwan AG (2022) A generalized framework for elliptic curves based PRNG and its utilization in image encryption. *Sci Rep* 12(1):13278
10. Zhao W, Chang Z, Ma C, Shen Z (2023) A pseudorandom number generator based on the chaotic map and quantum random walks. *Entropy* 25(1):166
11. Mebenga VBE, Kopparthi VR, Nzeuga HD, Fouda JAE, Dagoumguei GMD, Bitjoka GB, Sabat SL (2023) An 8-bit integer true periodic orbit PRNG based on delayed Arnold's cat map. *AEU Int J Electron Commun* 162:154575
12. De la Fraga LG (2023) A chaotic PRNG tested with the heuristic differential evolution. *Integration*
13. Sangeetha Y, Majji S, Srinagesh A, Patnala TR, Nalajala S, Prathap BR (2023) Authentication of symmetric cryptosystem using anti-aging controller-based true random number generator. *Appl Nanosci* 13(2):1055–1064
14. AbdElHaleem SH, Abd-El-Hafiz SK, Radwan AG (2022), September PRNG Using Primitive Roots of Primes and its Utilization in Chess-based Image Encryption. In 2022 27th International Conference on Automation and Computing (ICAC) (pp. 1–6). IEEE
15. Liao TL, Wan PY, Yan JJ (2022) Design and synchronization of chaos-based true random number generators and its FPGA implementation. *IEEE Access* 10:8279–8286
16. Yang Z, Liu Y, Wu Y, Qi Y, Ren F, Li S (2023) A high speed pseudo-random bit generator driven by 2D-discrete hyperchaos. *Chaos Solitons Fractals* 167:113039
17. Garipcan AM, Erdem E (2023) FPGA modeling of a novel fully-synthesizable and secure TRNG based on key-dependent s-box. *Dig Signal Process* 136:103969
18. Valle J, Machicao J, Bruno OM (2022) Chaotical PRNG based on composition of logistic and tent maps using deep-zoom. *Chaos Solitons Fractals* 161:112296
19. Kaçar S (2022) Digital circuit implementation and PRNG-based data security application of variable-order fractional Hopfield neural network under electromagnetic radiation using Grünwald-Letnikov method. *Eur Phys J Spec Top* 231(10):1969–1981
20. Chamgoué AC, Sriram S, Kuate PDK, Kingni ST, Rajagopal K (2023) Pseudo random number generator based on memristive Helmholtz snap oscillator with two lines of equilibria embedded in microcontroller. *Phys Scr* 98(5):055213
21. Gutub A, Al-Roithy B (2021) Varying PRNG to improve image cryptography implementation. *J Eng Res*. <https://doi.org/10.36909/jer.v9i3a.10111>
22. Irfan M, Ali A, Khan MA, Ehatisham-ul-Haq M, Mehmood Shah SN, Saboor A, Ahmad W (2020) Pseudorandom number generator (PRNG) design using hyper-chaotic modified robust logistic map (HC-MRLM). *Electronics* 9(1):104
23. Palacios-Luengas L, Pichardo-Méndez JL, Díaz-Méndez JA, Rodríguez-Santos F, Vázquez-Medina R (2019) PRNG based on skew tent map. *Arab J Sci Eng* 44:3817–3830
24. Ozkaynak F (2020) A novel random number generator based on fractional order chaotic Chua system. *Elektron Ir Elektrotech* 26(1):52–57
25. Bertoni G, Daemen J, Peeters M, Van Assche G (2013) Keccak. In *Advances in Cryptology—EUROCRYPT 2013: 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Athens, Greece, May 26–30, 2013. *Proceedings* 32 (pp. 313–314). Springer Berlin Heidelberg
26. NIST (2010) A statistical test suite for cryptographic applications. National Institute of Standards and Technology, pp 23–87
27. Patel S, Thanikaiselvan V (2023) Latin square and machine learning techniques combined algorithm for image encryption. *Circuits Syst Signal Process* 42(11):6829–6853
28. Chen L, Yin H, Huang T, Yuan L, Zheng S, Yin L (2020) Chaos in fractional-order discrete neural networks with application to image encryption. *Neural Netw* 125:174–184
29. Wang X, Gao S (2020) Image encryption algorithm for synchronously updating boolean networks based on matrix semi-tensor product theory. *Inf Sci* 507:16–36
30. Hadj Ibrahim A, Ali Pacha A, Hadj Said N (2023) A new fast image compression–encryption scheme based on compressive sensing and parallel blocks. *J Supercomput* 79(8):8843–8889

31. Huang X, Dong Y, Ye G, Shi Y (2023) Meaningful image encryption algorithm based on compressive sensing and integer wavelet transform. *Front Comput Sci* 17(3):173804
32. Cun Q, Tong X, Wang Z, Zhang M (2021) Selective image encryption method based on dynamic DNA coding and new chaotic map. *Optik* 243:167286
33. Jithin KC, Sankar S (2020) Colour image encryption algorithm combining Arnold map, DNA sequence operation, and a Mandelbrot set. *J Inf Secur Appl* 50:102428
34. Zhang Q, Han J (2021) A novel color image encryption algorithm based on image hashing, 6D hyperchaotic and DNA coding. *Multimedia Tools Appl* 80:13841–13864
35. Zefreh EZ (2020) An image encryption scheme based on a hybrid model of DNA computing, chaotic systems and hash functions. *Multimedia Tools Appl* 79(33–34):24993–25022
36. Zhu Y, Wang C, Sun J, Yu F (2023) A chaotic image encryption method based on the artificial fish swarms algorithm and the DNA coding. *Mathematics* 11(3):767
37. Zhang YQ, He Y, Li P, Wang XY (2020) A new color image encryption scheme based on 2DNLCLML system and genetic operations. *Opt Lasers Eng* 128:106040
38. Gupta A, Singh D, Kaur M (2020) An efficient image encryption using non-dominated sorting genetic algorithm-III based 4-D chaotic maps: image encryption. *J Ambient Intell Humaniz Comput* 11:1309–1324
39. Murali P, Niranjana G, Paul AJ, Muthu JS (2023) Domain-flexible selective image encryption based on genetic operations and chaotic maps. *Vis Comput* 39(3):1057–1079
40. Erkan U, Toktas A, Lai Q (2023) 2D hyperchaotic system based on Schaffer function for image encryption. *Expert Syst Appl* 213:119076
41. Song W, Fu C, Zheng Y, Tie M, Liu J, Chen J (2023) A parallel image encryption algorithm using intra bitplane scrambling. *Math Comput Simul* 204:71–88
42. Zhu S, Deng X, Zhang W, Zhu C (2023) Secure image encryption scheme based on a new robust chaotic map and strong S-box. *Math Comput Simul* 207:322–346
43. Zhang J, Guo J, Lu D (2023) An efficient image encryption algorithm based on S-box and DNA code. *Meas Sensors* 29:100894
44. Qobbi Y, Jarjar A, Essaid M, Benazzi A (2023) Image encryption algorithm using dynamic permutation and large chaotic S-box. *Multimedia Tools Appl* 82(12):18545–18564
45. Ali R, Jamil MK, Alali AS, Ali J, Afzal G (2023) A robust s box design using cyclic groups and image encryption. *IEEE Access* 11:135880–135890
46. Hematpour N, Gharari F, Ors B, Yalcin ME (2023) A novel S-box design based on quantum tent maps and fractional stochastic models with an application in image encryption. *Soft Comput*, 1–34
47. Gakam Tegue, G. A., Nkapkop, J. D. D., Abdel, M. A., Tsafack, N., Musheer, A., Signing, F. V., ... Tamba, J. G. (2023). A Novel Image Encryption Scheme Combining a Dynamic S-Box Generator and a New Chaotic Oscillator with Hidden Behavior. *Arabian Journal for Science and Engineering*, 1–20.
48. Hadj Brahim A, Pacha A, A., Hadj Said N (2023) An image encryption scheme based on a modified AES algorithm by using a variable S-box. *J Opt*, 1–16
49. Zhou S, Qiu Y, Wang X, Zhang Y (2023) Novel image cryptosystem based on new 2D hyperchaotic map and dynamical chaotic S-box. *Nonlinear Dyn* 111(10):9571–9589
50. Chinram R, Bano M, Habib U, Singavananda P (2023) Highly secured and quickest image encryption algorithm based on trigonometric chaotic map and S-box. *Soft Comput*, 1–13
51. Zheng J, Zeng Q (2022) An image encryption algorithm using a dynamic S-box and chaotic maps. *Appl Intell* 52(13):15703–15717
52. Shafique A, Ahmed F (2020) Image encryption using dynamic S-box substitution in the wavelet domain. *Wireless Pers Commun* 115:2243–2268
53. Singh KN, Singh OP, Singh AK, Agrawal AK (2023) EiMOL: a secure medical image encryption algorithm based on optimization and the Lorenz system. *ACM Trans Multimedia Comput Commun Appl* 19(2s):1–19
54. Sameh SM, Moustafa HED, AbdelHay EH, Ata MM (2023) An effective chaotic maps image encryption based on metaheuristic optimizers. *J Supercomputing*, 1–61
55. Yang Y, Cheng M, Ding Y, Zhang W (2023) A visually meaningful image encryption scheme based on lossless compression SPIHT coding. *IEEE Transactions on Services Computing*
56. Yu F, Xu S, Xiao X, Yao W, Huang Y, Cai S, Yin B, Li Y (2023) Dynamics analysis, FPGA realization and image encryption application of a 5D memristive exponential hyperchaotic system. *Integration* 90:58–70
57. Hao W, Zhang T, Chen X, Zhou X (2023) A hybrid NEQR image encryption cryptosystem using two-dimensional quantum walks and quantum coding. *Signal Process* 205:108890
58. Bezerra JIM, Machado G, Molter A, Soares RI, Camargo V (2023) A novel simultaneous permutation–diffusion image encryption scheme based on a discrete space map. *Chaos Solitons Fractals* 168:113160

59. Huang CK, Liao CW, Hsu SL, Jeng YC (2013) Implementation of gray image encryption with pixel shuffling and gray-level encryption by single chaotic system. *Telecommun Syst* 52:563–571
60. Sang Y, Sang J, Alam MS (2022) Image encryption based on logistic chaotic systems and deep autoencoder. *Pattern Recognit Lett* 153:59–66
61. Elkandoz MT, Alexan W (2022) Image encryption based on a combination of multiple chaotic maps. *Multimedia Tools Appl* 81(18):25497–25518

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.