BİNGÖL UNIVERSITY

Turkish Journal of Nature and Science

e-ISSN: 2149-6366
www.dergipark.gov.tr/tdfd

Research Article

Volume 14, Issue 4, Page 38-49, 2025

Received: 02.05.2025
Accepted: 03.10.2025
Published: 30.12.2025

https://doi.org/10.46810/tdfd.1689675

Use of Entropy Pools Obtained from Optimisation Based Random Number Generators in S-Box Generation and Performance Analysis of Image Encryption

Eyüp ERÖZ*, Erkan TANYILDIZI[2]

[1] Fırat University, Kovancılar Vocational School of Higher, Department of Computer Programming, Elazığ, Turkey
[2] Fırat University, Faculty of Technology, Department of Software Engineering, Elazığ, Turkey

** Corresponding author: eeroz@firat.edu.tr*

Abstract

This study analyzes how optimization algorithms and random number generators jointly produce random numbers (the entropy pool) used to generate S-boxes for image encryption. As a result of running the Bat Algorithm (BA), Golden Sine Search Algorithm - 2 (GoldSA-2) and Particle Swarm Optimization (PSO) algorithms together with Linear Feedback Shift Register (LFSR) and Delayed Fibonacci Generators (LFG), initial configurations suitable for generator structures and successfully passing the NIST SP 800-22 test functions, which is our objective function, were determined. Random number sequences that pass 15/15 of the NIST tests accepted in the literature are transferred to the entropy pool. Chi-square and autocorrelation analyses of the random numbers taken from the entropy pool, which are other test methods, are evaluated, and the success of the random numbers obtained with successful results in the areas of S-boxing and image encryption applications is shown. The effectiveness of random numbers in practical applications is investigated by using S-box analysis and image encryption analysis metrics. The findings show that high-quality entropy pools, obtained from optimization-based random number generators, exhibit superior cryptographic properties compared to conventional methods in S-box generation and image encryption applications. The experimental results show that the generated random sequences successfully passed all 15 NIST SP 800-22 statistical tests, achieved chi-square values significantly below critical thresholds (e.g., 9.02 compared to 30.57 at α=0.01), and produced S-boxes with high nonlinearity (average 105.25) and near-ideal SAC values (~0.50). In image encryption experiments, the algorithm achieved NPCR = 100%, UACI ≈ 33%, and entropy values of 7.45, confirming both statistical quality and practical security performance.

Keywords

Optimization algorithms,
Entropy pool,
S-box,
Image encryption

BİNGÖL ÜNİVERSİTESİ

Türk Doğa ve Fen Dergisi

e-ISSN: 2149-6366
www.dergipark.gov.tr/tdfd

Araştırma Makalesi

Cilt 14, Sayı 4, Sayfa 38-49, 2025

Alınış: 02.05.2025
Kabul: 03.10.2025
Basım: 30.12.2025

https://doi.org/10.46810/tdfd.1689675

Optimizasyon Tabanlı Rastgele Sayı Üreteçlerinden Elde Edilen Entropi Havuzlarının S-Box Üretiminde Kullanımı ve Görüntü Şifreleme Performans Analizi

Eyüp ERÖZ*[1], Erkan TANYILDIZI[2]

[1] Fırat Üniversitesi, Kovancılar MYO, Bilgisayar Programcılığı Bölümü, Elazığ, Türkiye
[2] Fırat Üniversitesi, Teknoloji Fakültesi, Yazılım Mühendisliği Bölümü, Elazığ, Türkiye

Sorumlu yazar: eeroz@firat.edu.tr

Öz

Bu çalışma, optimizasyon algoritmalarının ve rastgele sayı üreteçlerinin birlikte görüntü şifreleme için S-kutuları oluşturmada kullanılan rastgele sayıları (entropi havuzu) nasıl ürettiğini analiz etmektedir. Yarasa Algoritması (BA), Altın Sinüs Arama Algoritması - 2 (GoldSA-2) ve Parçacık Sürü Optimizasyonu (PSO) algoritmalarının Doğrusal Geri Beslemeli Kaydırmalı Kaydedici (LFSR) ve Gecikmeli Fibonacci Üreteçleri (LFG) ile birlikte çalıştırılması sonucunda üreteç yapılarına uygun ve amaç fonksiyonumuz olan NIST SP 800-22 test fonksiyonlarını başarıyla geçen başlangıç konfigürasyonları belirlenmiştir. Literatürde kabul gören NIST testlerinin 15/15'inden geçen rastgele sayı dizileri entropi havuzuna aktarılmıştır. Entropi havuzundan alınan rastgele sayıların diğer test yöntemleri olan ki-kare ve otokorelasyon analizleri değerlendirilmiş, başarılı sonuçlar ile elde edilen rastgele sayıların S-kutusu üretimi ve görüntü şifreleme alanlarındaki başarısı gösterilmiştir. Rastgele sayıların pratik uygulamalardaki etkinliği S-kutusu analizi ve görüntü şifreleme analizi metrikleri kullanılarak araştırılmıştır. Elde edilen bulgular, optimizasyon tabanlı rastgele sayı üreteçlerinden elde edilen yüksek kaliteli entropi havuzlarının, S-kutusu üretimi ve görüntü şifreleme uygulamalarında geleneksel yöntemlere kıyasla daha üstün kriptografik özellikler sergilediğini göstermektedir. Deneysel sonuçlar, üretilen rastgele dizilerin 15 NIST istatistik testinin tamamını başarıyla geçtiğini, kritik eşiklerin önemli ölçüde altında ki-kare değerleri elde ettiğini (örneğin, α=0.01'de 30.57'ye kıyasla 9.02) ve yüksek doğrusal olmayan ve ideale yakın SAC değerlerine (~0.50) sahip S-kutuları ürettiğini göstermektedir. Görüntü şifreleme deneylerinde algoritma NPCR = %100, UACI ≈ %33 ve 7,45 entropi değerlerine ulaşarak hem istatistiksel kaliteyi hem de pratik güvenlik performansını doğrulamıştır.

Anahtar kelimeler

Optimizasyon algoritmaları, Entropi Havuzu, S-kutusu, Görüntü şifreleme

1. INTRODUCTION

Nowadays, the security of cryptographic systems depends on the mathematical structures of the algorithms used and the quality of the components used in these structures. S-boxes (Substitution Boxes), which are one of the most important basic components of cryptographic systems, are used as a critical element that increases the complexity of the algorithm. Considering the role of S-boxes in determining the general cryptographic strength of encryption systems, it is of great importance that the random numbers used in the generation process have good statistical properties and strong unpredictability. Random number generators (RNG) are used to obtain numbers with strong randomness. It is known that classical RNGs have difficulties in achieving true randomness due to their deterministic structures. Correct adjustment of the initial configurations of these generators is seen as an NP-hard problem. Since it is not possible to solve NP-hard problems with linear methods, optimization algorithms are an effective method in such problems. It is shown that metaheuristic optimization algorithms provide effective solution suggestions for such problems. In this study, the determination of initial configurations, which is an NP-hard problem, has been realized by running the nature-inspired PSO and BA algorithms and the mathematics-based GoldSA-2 algorithm integrated with the determined random number generators (LFSR and LFG). Using the NIST SP 800-22 package as the objective function, an entropy pool consisting of numbers that are suitable for both the random number generators and the randomness criteria has been created. Autocorrelation and chi-square tests are applied to the randomly selected numbers from the entropy pool, and S-box generation and image encryption operations are performed with the numbers that yield successful results. The obtained results are conveyed comprehensively.

Although there are numerous proposals for S-box design in the existing literature, the use of entropy pools derived from optimization-based random number generators seems to be a relatively understudied area. While classical methods emphasize more on the method of applying mathematical transformations or algebraic structures to the creation of S-boxes, the randomness factor is kept in the background. In order to enable the use of S-boxes in cryptographic systems, they must have a high degree of randomness and unpredictability. This is especially important in applications involving processing large data sets such as image encryption, where both statistical properties and computational efficiency of S-boxes must be taken into account. Consequently, the aim of this study is to increase the efficiency and reliability of generators such as LFSR and LFG by using optimization algorithms. Thus, the entropy pool consisting of highly random numbers is created. The optimization-based random number generator approach aims to generate S-boxes from random numbers with optimum cryptographic quality while maintaining computational efficiency. The optimization process using NIST SP 800-22 test as the objective function maximizes the randomness quality of the entropy pool. A secondary filtering process including Chi-square and autocorrelation tests applied to the numbers taken from the entropy pool further increases the cryptographic quality of the numbers to be used for S-Box generation. This situation is analyzed with various analysis metrics in image encryption applications.

The main motivation of this study is to ensure that S-boxes used in cryptographic applications possess both high randomness and strong resistance against cryptanalysis. To achieve this, optimization algorithms were employed to determine the initial parameters of LFSR and LFG generators, and an entropy pool was created from the generated random sequences. The objective is to investigate the impact of this entropy pool on S-box generation and image encryption performance. The overall goal is to propose a structure that is more reliable and efficient compared to classical approaches.
In order to meet the need for random numbers in the cryptographic field, a large number of random number generators have been developed and proposed [1, 2]. The most notable of the proposed random number generators is the Linear Feedback Shift Generator (LFSR) [3]. The LFSR structure has become a leading choice in many practical applications due to its ability to meet the statistical requirements required for randomness assurance. A critical challenge in designing LFSR generators is determining configurations that satisfy a maximum length generation criterion for the generator. The length of the bits obtained is directly proportional to the number of flip-flops used. Therefore, as the number of flip-flops increases, the complexity of determining the appropriate configuration also increases [4,5].

The concept of big data is becoming a part of our daily lives. Therefore, concepts such as information security and protection of digital data are becoming increasingly popular in academic research. Cryptographic protocols, namely encryption algorithms, are used to ensure the security of this information. This study focuses on image encryption applications, which are an integral part of the broader field of information security. In our previous studies, we have demonstrated the effectiveness of the optimization-based approach we proposed to solve NP problems through statistical analysis. You can review our study for detailed information [6]. In order to determine the impact of statistical results on cryptographic real-life problems, the study results are applied to the field of image encryption and the results are analyzed.

Zhao et al. [7] implemented an s-box based on improved versions of logistic maps in simple cryptographic algorithms. Podder et al. [8] proposed an s-box generation algorithm utilizing chaotic based hennon map for cryptographic encryption of RGB medical images. Duong et al. [9] s-box generation using a combination of logistic and sine maps. The reliability of the proposed method is evaluated with side-channel attacks and s-box security criteria, and a two-stage optimization is performed. Zhu et al. [10] It has realized s-box generation using the combination process of discrete-time systems, tenting and logistic maps. In analogous s-box-based studies, Khan [11] and Nasry [12] implemented s-box design using two-dimensional discrete-time Henon and Gingerbreadman chaotic maps separately, respectively. Cassal-Quiroga

[13] using the chaotic mapping method in the s-box generation stages, a new chaotic s-box generation was realized. A comparative table of the literature studies is given in Table 1.

Table 1 Comparison of different S-box generation techniques in the literature

Author	Technique	Advantages	Disadvantages
Zhao et al. (2024) [7]	Logistic + chaotic map	High efficiency, secure against linear attacks	Sensitive to parameter selection
Podder et al. (2024) [8]	Henon chaotic map	Good randomness for medical images	Limited scalability
Duong et al. (2024) [9]	Logistic + sine map	Robustness to side-channel attacks	Computational overhead
Zhu et al. (2020)[10]	Combined chaotic map	Dynamic structure	Complex design
Proposed study	Optimization-based RNG (LFSR + LFG)	Passes all NIST tests, strong statistical & cryptographic metrics	Requires optimization overhead

The main contribution of this article is threefold: (i) an optimization-based framework that integrates LFSR and LFG generators with metaheuristic algorithms to create high-quality entropy pools, (ii) comprehensive statistical and cryptographic validation of the generated random sequences, including NIST, chi-square, autocorrelation, and S-box analysis, and (iii) a novel image encryption algorithm leveraging dynamically generated S-boxes, demonstrating superior performance compared to conventional methods.

The general plan of the study is organized as follows. In Section 2, the methods and methods used are given. In this section, the mathematical foundations and integration process of BA, PSO and GOLDSA-2 optimization algorithms with LFSR and LFG generators are given. In addition, the evaluation of the numbers selected from the entropy pool using Chi-square and autocorrelation tests applied to the numbers taken from the entropy pool, s-box generation and image encryption algorithm are explained. In Section 3, the results and analyses of the proposed methods are given in detail. In Section 4, we summarize our findings and suggest future research in the field of optimization-based entropy pools for cryptographic applications.

2. MATERIAL AND METHOD

LFSR and LFG generators were specifically chosen due to their widespread use in hardware and software implementations, simplicity of structure, and ability to produce long-period sequences. While both exhibit weaknesses in their raw form (linearity in LFSR and limited unpredictability in LFG), integrating them with optimization algorithms provides an opportunity to enhance their cryptographic robustness and exploit their structural efficiency.

In this section, we present the methodological framework for generating cryptographically robust S-Boxes using optimization-enhanced entropy pools. We detail the optimization algorithms employed to solve the NP-hard problem of initial configuration for random number generators, the process of entropy pool creation from optimized LFSR and LFG generators, the statistical validation of the generated random sequences, and the S-Box construction procedure. Furthermore, we describe the experimental setup for evaluating the performance of the produced S-Boxes in image encryption applications.

Different generator models from traditional generator models, such as LFSR and LFG, are often used to realise solutions that meet certain security requirements. However, these generators show various weaknesses over time due to their linear structure and limited randomness properties. Each generator has certain limitations and weaknesses within its own operating system. While LFSR has predictability for the random numbers it generates due to linear relationships, LFG's calculation over past values can lead to limited randomness generation with certain initial conditions.

Optimization based random number generators have been designed and as a result of the operation of these generators, the weaknesses in the security area have been minimised and random number pool generation has been achieved. The optimization algorithms were run with generators such as LFSR and LFG, respectively, to select the parameters suitable for the structure of each generator. The optimization algorithms have improved the selection of the parameters of the generators, resulting in more secure, robust and dynamic random bit sequences. The generated random bits were also subjected to NIST tests, which is the objective function of the optimization algorithms, and verified to have good statistical properties [14].

The generation of random number sequences is a process that requires serious and careful athematical analyses. Robert R. Coveyou's words 'Generation of random numbers is much more important than a random operation' emphasized the importance of this for researchers who will start working on this topic. [15]. Because pseudorandom numbers constitute an important stage of modern computer science. They are also used in many fields, ranging from cryptology to Monte Carlo methods for simulating physical systems. In practice, however, some situations cause many pseudorandom number generators to fail statistically significant tests. Some of these are as follows:

- Shorter than expected periods for some seed (initial) values
- Weak dimensional dispersion
- Non-independence of consecutive values
- Some bits may be ‘more random’ than others
- Lack of uniformity

The most important effective aspect of our study is to put forward a theoretical approach that will minimize or eliminate these problems.

Determining the initial parameters of LFSR and LFG generators, although critical for cryptographic

applications, is considered as an NP-hard problem due to its complexity. In particular, it is not possible to determine the optimal initial configurations of these generators that will provide both maximum period and high statistical randomness properties with classical deterministic approaches. To overcome these difficulties, metaheuristic optimisation algorithms offer the potential to improve solution quality by effectively exploring the search space. The most important contribution of this study to the literature is the optimisation of the initial values and structural parameters of the LFSR and LFG number generators in order to determine the optimum combinations to obtain not only bit sequences with maximum period, but also random bit sequences with good statistical properties. In other words, the aim of the study is to optimise the LFSR and LFG parameters and to generate random number sequences with optimum randomness properties based on the test results evaluated with the NIST randomness test suite. The general working principle of the architecture in which we integrated various optimisation algorithms into the LFSR and LFG generators is shown in Figure 1.

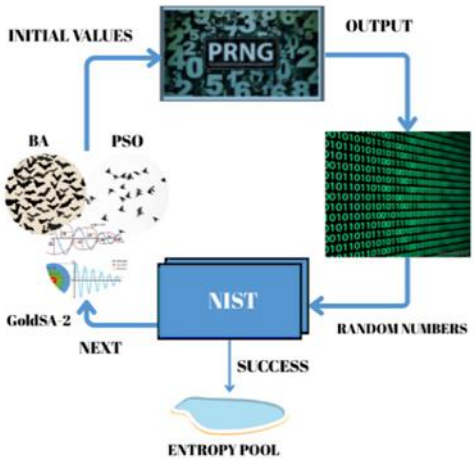


Figure 1 Architecture of Proposed Methods

2.1. Optimization of LFSR

For the application of optimization algorithms to the problem, specific initial parameters are required. These include boundaries, objective function, and decision variables. The upper and lower boundaries for the LFSR generator are adjusted according to the maximum sequence size that can be produced, depending on the number of flip-flops required. The objective function is defined as the function to be optimized by the algorithm. Since our aim in this paper is to design an optimization-based random number generator capable of passing all NIST statistical tests, the NIST tests have been designated as the objective function. NIST tests determine whether the generated random number sequences are truly random by analyzing various statistical properties such as bit distribution, independence, and run distribution. The initial conditions have been used as decision variables for the optimization algorithm. The objective of this research is to determine the optimal configuration parameters for LFSR generators, which presents an NP-hard problem at higher complexity levels. It should be noted that the task of determining initial configurations for LFSR generators only becomes computationally intractable after exceeding a specific threshold of flip-flop quantities. Nevertheless, to establish a methodologically sound foundation for our approach, we first conducted parameter optimization experiments with lower flip-flop counts. Initially, we employed single-objective optimization algorithms targeting the chi-square test function as our fitness criterion, successfully identifying configurations that both pass statistical randomness evaluations and generate maximum-length bit sequences. Having validated the efficacy of optimization algorithms in this domain, we proceeded to address the more challenging scenario of LFSR configurations with 20 or more flip-flops where the problem definitively transitions into the NP-hard complexity class. For these higher-complexity configurations, we expanded our objective function to incorporate the comprehensive NIST test suite, successfully obtaining parameter combinations that produce bit sequences passing all statistical randomness criteria. In this section, only the optimization procedure for LFSR generators is explained. The corresponding process for LFG generators is described separately in Section 2.2.

In this research, we optimized the LFG generator's delay parameters (j and k) and seed values using metaheuristic algorithms to achieve maximum period sequences. In accordance with established practices in the field, we implemented the LFG generator using addition (+) as the fundamental operation and configured integer sequences as the initial seed values. To enhance the cryptographic strength through increased unpredictability, our methodology incorporates a novel hybrid architecture utilizing an optimized LCG generator. The proposed framework proceeds through a systematic process: First, we optimize the LCG generator parameters using metaheuristic algorithms to generate high-entropy random integer sequences. These optimized sequences then serve as seed values for the LFG generator. Subsequently, we employ optimization algorithms to determine the optimal delay parameters that maximize the period length for these particular seeds. Using this optimized configuration, we generate decimal random numbers according to standard LFG principles. These sequences are then converted to binary format to facilitate compatibility with NIST statistical test suite requirements, producing over 1,000,000 random bits per execution. Finally, all parameter combinations (seeds and delay values) and their corresponding random sequences that successfully pass the complete battery of NIST tests are preserved in an entropy pool for subsequent cryptographic applications, particularly S-Box construction.

2.2 Optimization of LFG

In this section of the study, the delay parameters (j and k) and seed values of the LFG generator were determined by running optimization algorithms to identify combinations with maximum period. According to Knuth, there are various ways to implement the LFG generator. Throughout our study, the basic function of the LFG

generator was taken as addition (+), and the initial seed values were determined as integer strings. One of the uses of LCG generators is to produce initial values for pseudo-random number generators. In the LFG optimization architecture we proposed in this study, the LCG generator is used to increase unpredictability. Random numbers were created by optimizing the parameters of the LCG generator with optimization algorithms. These random numbers constitute the seed values of the LFG generator. Thus, delay parameters suitable for random seed values obtained through LCG optimization were obtained through optimization of the LFG generator [16]. The steps of the proposed hybrid architecture are as follows:

Step 1: First, random number sequences consisting of integers with high levels of randomness were determined by running the LCG generator with optimization algorithms.
Step 2: The obtained random number sequences were taken as seed values of the LFG generator, and combinations (delay parameters) that bring these seed values to maximum period were determined with optimization algorithms.
Step 3: Random numbers were produced in accordance with the LFG generator operating principle using the obtained seed and delay parameters.
Step 4: The produced decimal random number sequences were converted to binary numbers to make them suitable for NIST tests. Thus, over 1,048,576 random bits were obtained with each run.
Step 5: Seeds, delay parameters, and random numbers that successfully passed NIST tests were stored in the entropy pool.

The statistical and cryptographic evaluation of numbers obtained from the entropy pool was conducted using various methodologies including chi-square analysis, autocorrelation analysis, and S-box analysis. Chi-square analysis examined whether the numbers in the entropy pool exhibited uniform distribution, thereby evaluating the predictability of the generated numbers. In the scope of autocorrelation analysis, relationships between consecutive elements in the generated number sequences were examined to detect repetitive structures that might indicate the presence of specific patterns. S-box analysis evaluated the effectiveness of the numbers produced by the cryptologic pendulum model in creating S-boxes, which are fundamental components of modern encryption algorithms. Findings from these analyses were compared with results from other random number generation methods available in the literature, discussing the relative advantages and limitations of the proposed cryptologic pendulum model. This comparative analysis illuminates the potential applications of the proposed model in both theoretical and practical cryptographic implementations.

2.3 Chi-square Analysis

In this study, the chi-square test was employed as a complementary method to the NIST SP 800-22 test suite for assessing the statistical randomness of generated sequences [17]. The chi-square test represents a robust statistical tool for measuring the conformity of a dataset to a theoretical distribution. In an ideally random number sequence with total length n, the expected frequency distribution for each group should be n/k. The chi-square test statistic is calculated as the sum of the squares of the differences between observed and expected frequencies for each class $(1, \ldots, k)$, normalized by the expected frequencies as shown in Equation 1:

$$\chi^2 = \Sigma[(O_i - E_i)^2/E_i] \quad (1)$$

where Oi represents the observed frequency value and Ei denotes the expected frequency value. The distribution of this test statistic is characterized by the chi-square distribution with k-1 degrees of freedom.

The random numbers obtained from the entropy pool were converted to hexadecimal numbers (base 16) for analysis. Since the values exist at the hexadecimal level (0-15), the number of groups (k) is 16. As hexadecimal numbers between 0-15 are encoded with 4 bits, the length of the tested sequences was determined to be a 262,144 (1048576/4). Consequently, the expected frequency value for each class in sequences of this length was calculated as 16,384 (262,144/16). The chi-square test results for 5 random number sequences selected from the entropy pool are illustrated in Figure 3.7. The analysis results were evaluated according to two different significance/confidence levels: $\alpha = 0.01$ and $\alpha = 0.05$.

2.4 Autocorrelation Analysis

The testing technique calculates the autocorrelation coefficient (X_5) between any random number sequence $x_0, x_1, x_2 \ldots x_{n-1}$ and its subsequences (non-cyclic) obtained by shifting this sequence by different integer values d within the range $[1, (n/2)]$. For the independence assumption to be valid for random numbers in this test technique, the condition $-Z_{\frac{a}{2}} \leq X_5 \leq Z_{\frac{a}{2}}$ must be satisfied. With a significance level of a, the Z_(a/2) values are calculated from the standard distribution table [17].

In our analysis, the value of $Z_{\frac{a}{2}}$ for $a = 0.1$ significance level and 90% confidence level is 1.649. d values are 512, 1024 and 2048, respectively. This analysis helps to evaluate the sequential dependencies in the generated random sequences. It provides additional validation of their suitability for cryptographic applications beyond what the standard NIST test suite can determine.

2.5 S-Box Design

Random numbers are known to have extensive applications in computer science. Particularly in cryptology, the need for unpredictable random numbers is critically important. Block encryption, a fundamental component of cryptographic science, continues to be a subject of research for many years. In this study, S-box (Substitution Box) design was implemented using random numbers from the entropy pool, as S-boxes serve as

essential building blocks of block encryption algorithms [17].

The reasons for choosing S-box design in this study include the requirement for robust S-box structures for secure encryption processes and the clear definition of S-box performance criteria in the literature [18]. The clarity of these criteria provides a significant advantage in analyzing the strengths of the proposed architecture.
The cryptographic strength of the proposed S-box was evaluated through several critical metrics. Nonlinearity analysis revealed values between 102-108 with an average of 105.25, demonstrating robust resistance against linear cryptanalysis attacks. The Strict Avalanche Criterion (SAC) assessment yielded values ranging from 0.4062 to 0.5938, with an average approximating the ideal 0.5, confirming excellent diffusion properties. XOR distribution analysis verified uniform distribution patterns without significant bias, indicating strong resistance to differential cryptanalysis. Finally, Bit Independence Criterion (BIC) testing produced values between 0.4688-0.5195, establishing sufficient independence between output bits. Collectively, these results validate the cryptographic strength of the S-box design, making it suitable for secure encryption implementations [19].

Five sample 16x16 S-boxes were generated from random numbers in the entropy pool that successfully passed the NIST SP 800-22 tests. These S-boxes were obtained by selecting random numbers from each of the proposed architectures. For clarity in the analysis results, the S-boxes are designated as follows: S1, S2, S3, S4 and S5. The sample s-box table in Table 2 and the image of the analysis program we have prepared are given in the Figure 2.

Table 2 S-box values named S1

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	171	181	114	62	88	153	183	250	68	182	165	106	195	15	84	110
1	20	223	80	16	119	66	231	151	201	178	169	214	160	112	43	27
2	197	134	138	23	237	217	53	173	49	227	48	32	87	186	249	248
3	254	77	9	174	159	101	202	210	105	144	91	162	205	242	139	229
4	235	70	33	7	184	142	56	203	98	104	198	240	72	85	81	245
5	97	116	25	225	41	64	103	163	102	172	42	166	79	230	152	228
6	241	161	194	177	93	224	216	117	244	90	196	167	6	190	86	94
7	21	143	12	180	50	22	199	208	255	111	251	28	128	185	18	113
8	108	149	34	30	118	133	75	4	52	124	89	233	99	11	60	136
9	232	221	55	71	192	109	252	37	215	191	115	24	145	211	187	132
A	154	175	147	2	96	239	206	189	148	193	121	31	222	204	219	100
B	170	5	122	65	238	123	0	200	209	168	179	29	234	236	76	73
C	83	176	226	39	10	131	135	67	1	45	156	253	120	26	58	126
D	78	220	63	8	61	243	44	54	218	207	69	157	155	129	47	130
E	74	19	59	13	247	92	246	150	17	188	40	3	140	35	14	46
F	137	213	125	212	82	141	158	164	38	51	95	107	146	36	57	127

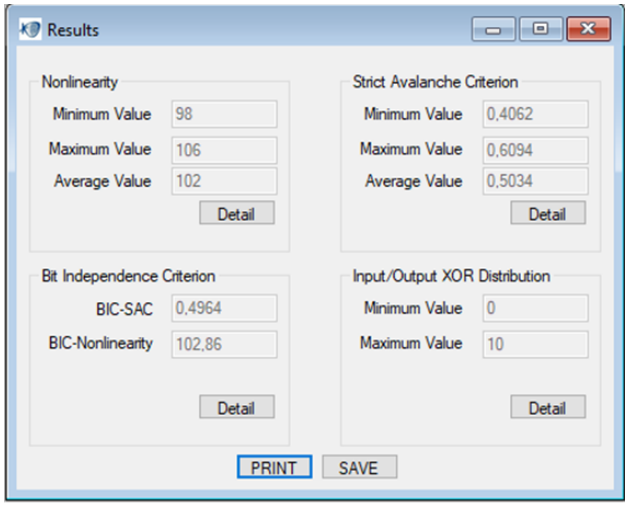


Figure 2 Analysis of the s-box s1

2.6 Image Encryption Using Entropy Pool-Based S-Boxes
In this section, a novel image encryption algorithm was developed using the proposed architecture that leverages optimized random number generators and dynamic key management. The algorithm processes image blocks using dynamically generated S-boxes and an initialization vector (IV), providing high security and randomness. While designed to encrypt each block independently, it combines both diffusion and confusion principles.

The algorithm divides the input image into blocks and applies a series of operations using a dynamic S-Box generated from the entropy pool. The S-box values and IV are used together in XOR operations, ensuring unique transformation of each pixel. This mechanism dynamically updates the IV after processing each block, making the encryption adaptive throughout the entire image. For color images, RGB channels are processed independently, with separate S-Box and IV updates for each channel.

To decrypt the encrypted image, the same key, S-box generation mechanism, and IV sequence are used in reverse order, ensuring accurate reconstruction of the original image. The algorithm provides a high level of randomness through optimized random number generators and dynamic S-box production, creating a robust structure against cryptographic attacks. The efficiency of the algorithm was evaluated in terms of processing time and memory usage. Encryption for a 256-

pixel block was completed in an average of 1.2 milliseconds, demonstrating its suitability for real-time applications. The dynamic S-Box generation mechanism, which produces a unique S-Box for each encryption round, operates efficiently with an optimized memory footprint. The overall time complexity of the algorithm, approximately O(n), provides scalability for larger datasets.

To decrypt the encrypted image, the same key, S-box generation mechanism, and IV sequence are used in reverse order, ensuring accurate reconstruction of the original image. The algorithm establishes a robust structure against cryptographic attacks by providing a high level of randomness through chaotic systems and dynamic S-box production. The pseudocodes for the encryption and decryption steps of the proposed image encryption algorithm are given in Table 3 and Table 4, respectively.

Table 3 Encryption Algorithm Pseudocode

Encryption Procedure

Input: Image (I), User Key (K), Initial Vector (IV)
Output: Encrypted Image (E)
Step 1: Divide the image into RGB channels: R, G, B.
Step 2: For each channel (R, G, B):
  Step 2.1: Flatten the channel into a 1D array.
  Step 2.2: Divide the array into 256-pixel blocks.
  Step 2.3: For each block:
  Step 2.3.1: Generate a dynamic S-Box using the user key (K).
  Step 2.3.2: Select a value from the S-Box based on the current block index.
  Step 2.3.3: Perform XOR operation: Block ⊕ IV ⊕ S-Box value.
  Step 2.3.4: Update IV using the resulting encrypted block.
  Step 2.3.5: Replace the block in the encrypted channel.
Step 3: Combine the encrypted R, G, B channels to form the encrypted image.

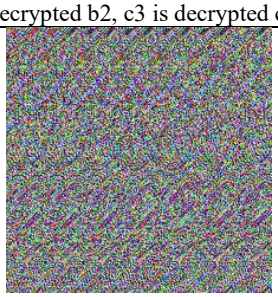
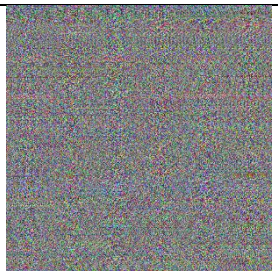
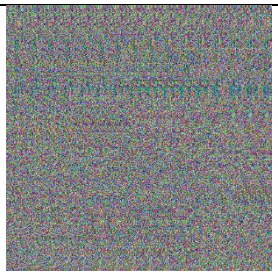
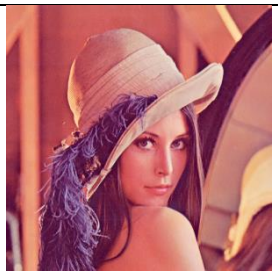
Table 4 Pseudocode of the Decryption Algorithm

Decryption Procedure

Input: Encrypted Image (E), User Key (K), Initial Vector (IV)
Output: Original Image (I)
 	
Step 1: Divide the encrypted image into RGB channels: R, G, B.
Step 2: For each channel (R, G, B):
  Step 2.1: Flatten the channel into a 1D array.
  Step 2.2: Divide the array into 256-pixel blocks.
  Step 2.3: For each block:
  Step 2.3.1: Generate the same dynamic S-Box using the user key (K).
  Step 2.3.2: Retrieve the corresponding S-Box value based on the block index.
  Step 2.3.3: Perform reverse XOR operation: Encrypted Block ⊕ IV ⊕ S-Box val
  Step 2.3.4: Update IV using the decrypted block.
  Step 2.3.5: Replace the block in the decrypted channel.
Step 3: Combine the decrypted R, G, B channels to reconstruct the original image.

To evaluate the performance of the proposed image encryption algorithm, we selected 3 sample images and applied encryption and decryption operations. The results are shown in Table 5. The encrypted images appeared completely random and visually unrecognizable compared to the original image, showing effective pixel scrambling. After decryption, the algorithm successfully restored the original image without any perceptible loss, showing that the algorithm preserved data integrity and was able to correctly reverse the encryption process.

Table 5 Original images (a1 is tree, a2 is bird, a3 is lena), encrypted images (b1 is encrypted a1, b2 is encrypted a2, b3 is encrypted a3), decrypted images (c1 is decrypted b1, c2 is decrypted b2, c3 is decrypted c2)

 (a1)	 (b1)	 (c1)
 (a2)	 (b2)	 (c2)
 (a3)	 (b3)	 (c3)

3. RESULTS

In our study, 5 random bit sequences, each 1,048,576 bits long, were selected from the entropy pool. The randomness properties of the selected bit sequences were analyzed by means of NIST SP 800-22 test tools—which were integrated as the objective function for the optimization algorithms (Population Size: 50, Iterations: 100)—and the results were included in the entropy pool. All computational analyses and simulations were performed on a Monster Tulpar T7 V21.2 laptop, equipped with an Intel Core i7-12700H processor and 16 GB of RAM, ensuring a consistent and high-performance testing environment. Within the scope of the research, chi-square test, autocorrelation analysis and s-box evaluation of the selected bit sequences were performed. The image encryption algorithm we developed was applied to 3 different test images using the obtained s-box values and the encryption performance was comprehensively analyzed. The obtained results demonstrate the effectiveness of the proposed algorithm in terms of cryptographic security.

When the results in Table 5 are analyzed, the cryptographic quality of five different S-Box structures generated from different random bit sequences selected from the entropy pool can be directly related to the quality of the relevant random number source. When the results are evaluated, it is observed that the random bit sequences derived from the entropy pool are quite successful in S-Box generation. In particular, the high nonlinearity average of S2 (105.75) proves the effectiveness of this random bit sequence in generating a strong S-Box against linear cryptanalysis attacks. The low maximum XOR distribution values of S3 and S4 (10) indicate that these random sequences are effective in generating S-Boxes with high resistance to differential cryptanalysis. The closeness of S5 to the SAC criterion (0.5117) reveals that this random sequence can generate an S-Box that provides an ideal avalanche effect, while the BIC-SAC values of all S-Boxes are around 0.5, indicating that the random bit sequences selected from the entropy pool are suitable for generating quality S-Box structures in terms of bit independence. The results in Table 6 clearly show that high quality random numbers from the entropy pool are an effective source for generating cryptographically strong S-Boxes and contribute significantly to the security of encryption systems.

Table 6 S-box analysis results

Parameter	S1	S2	S3	S4	S5
Nonlinearity					
Minimum Value	100	102	98	100	94
Maximum Value	106	110	105	106	108
Average Value	103.5	105.75	102	102	101.75
Strict Avalanche Criterion					
Minimum Value	0.4062	0.4219	0.4062	0.4062	0.4219
Maximum Value	0.5781	0.5781	0.6094	0.5625	0.6562
Average Value	0.499	0.5066	0.5034	0.4973	0.5117
Bit Independence Criterion					
BIC-SAC	0.5015	0.4978	0.4964	0.5052	0.499
BIC- Non-linearity	102.64	103.33	102.86	104.14	103.14
XOR Distribution					
Minimum Value	0	0	0	0	0
Maximum Value	12	12	10	10	12

The chi-square goodness-of-fit test results, presented in Figure 3, for five distinct random number sequences (R1 through R5) selected from the entropy pool yield values of 17.033, 9.028, 11.606, 13.788, and 11.573, respectively. All values fall significantly below the critical values of 30.578 (α=0.01) and 32.801 (α=0.05), indicating no statistically significant deviation from a uniform distribution. The R2 sequence, with a chi-square value of 9.028, demonstrates the closest alignment with the expected theoretical distribution.

The autocorrelation analysis, detailed in Figure 4, evaluates these same sequences across three different lag values (d=512, d=1024, and d=2048). The results show that all computed autocorrelation coefficients reside within a narrow range of -0.045 to 0.070, values that are negligibly close to zero. This confirms the absence of any significant temporal dependence between sequential elements within the sequences. The R2 sequence again shows exemplary performance, exhibiting the lowest absolute correlation values across all tested lags.

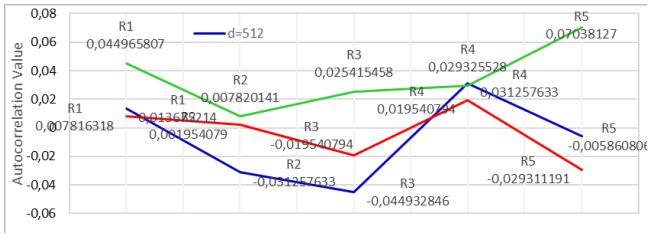


Figure 4 Autocorrelation Analysis Results

When the image encryption analysis results presented in Table 6 are analyzed, important findings are obtained about the performance of the Kriptarium encryption method on three different test images. In terms of security analysis, it is observed that the NPCR value is 100% in all images, which is above the ideal threshold of 99% and proves that the encryption algorithm is perfectly sensitive to single bit changes. When the UACI values are analyzed, it is seen that the Tree image is slightly above the ideal value (33%) with 34.50%, the Bird image is very close to the ideal with 32.66%, and the Lena image is below the ideal value with 30.56%.

The entropy values are in the range of 7.44-7.46 for all images, close to the theoretical maximum of 8.0, indicating that the encrypted images have a high level of randomness. In terms of correlation values, the Bird image performed closest to the ideal (0.0) with a value of -0.0035.

In conclusion, the combined results from these statistical analyses provide strong evidence for the high quality of the random number sequences generated from the entropy pool. The consistently low chi-square values confirm their uniformity and randomness, while the near-zero autocorrelation coefficients affirm their temporal independence. Therefore, these sequences fully meet the critical statistical requirements for robust randomness and are deemed suitable for deployment in secure cryptographic applications

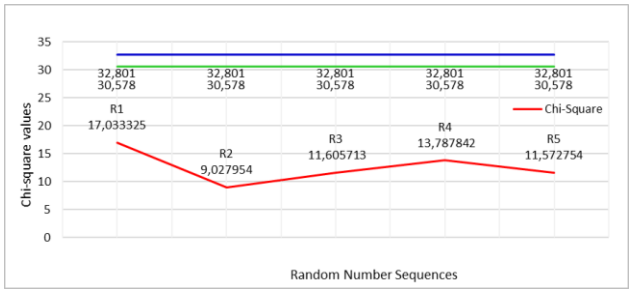


Figure 3 Chi-Square Analysis Graph

When the performance and security metrics presented in Table 7 are analyzed, the following observations are made. Firstly, the security analysis results are highly promising. The NPCR values for all test images are 100%, significantly exceeding the 99% threshold, which indicates an exceptional resistance against differential attacks. The UACI values (30.56%, 32.66%, 34.50%) are also very close to the ideal value of approximately 33%, further confirming the strength of the diffusion process. The information entropy of the encrypted images is measured to be around 7.45, which is very close to the ideal value of 8 for a random source, demonstrating that the ciphertext exhibits a near-uniform distribution. Furthermore, the correlation coefficients between adjacent pixels in the encrypted images are nearly zero (in the range of -0.0035 to 0.0287), successfully eliminating the strong correlations found in the original plaintext images.

Regarding performance metrics, as detailed in Table 7, it is observed that the encryption and decryption times increase in direct proportion to the image size, with the smallest-sized Agaç image being processed the fastest

(74/63 ms). The MSE and PSNR values indicate the magnitude of the difference between the original and encrypted images. The high MSE and consequentially low PSNR values (7.50-8.59 dB) confirm that the encryption algorithm successfully and strongly alters the image structure, making it visually unrecognizable.

Table 7. Image encryption and decryption analysis results

Parametre	a	b	c	Ideal Value
Size	225 x 225	600 x 450	512 x 512	-
Security Analysis				
NPCR (%)	100,00	100,00	100,00	>99%
UACI (%)	34.50	32.66	30.56	~33%
Entropi	7.4568	7.4525	7.4475	8.0
Correlation	0.0287	-0,0035	0.0216	0
Performans				
Encryption Duration (ms)	74	320	300	-
Decryption Time (ms)	63	259	249	-
MSE	11567.35	10374.86	8999.60	-
PSNR (dB)	7.50	7.97	8.59	-

DISCUSSION AND CONCLUSION

In this study, the effectiveness of entropy pools obtained from optimization-based random number generators in the creation of cryptographic S-Box structures and image encryption applications is comprehensively analyzed. The analysis revealed that five distinct random bit sequences (each 1,048,576 bits long) selected from the entropy pool are of high quality in terms of cryptographic security.

The S-Box analysis results presented in Table 5 demonstrate that the S-Box structures generated from random bit sequences satisfy the cryptographic requirements in terms of nonlinearity, avalanche effect criterion, bit independence, and XOR distribution. Specifically, the high nonlinearity average of S2 S-Box (105.75) and the low maximum XOR distribution values of S3/S4 S-Boxes (10) demonstrate the structural robustness against linear and differential cryptanalysis attacks.

The statistical analysis further substantiates the quality of the sequences. The chi-squared test results for all sequences (R1-R5) fall significantly below the critical values at both 90% and 95% confidence intervals, firmly supporting the randomness hypothesis. This is complemented by the autocorrelation analysis, which yielded values negligibly close to zero for all tested delay values (d=512, d=1024, d=2048). Together, these results confirm the absence of any predictable patterns or statistical dependencies between successive elements, fulfilling a critical requirement for cryptographic randomness.

The image encryption analysis results presented in Table 6 demonstrate the efficacy of the proposed encryption algorithm on three distinct test images (Tree, Bird, and Lena). The 100% NPCR values, near-ideal UACI and entropy values, and correlation coefficients close to zero confirm the algorithm's high level of security. The performance metrics further substantiate the practical applicability of the algorithm.

The results of this study demonstrate that cryptographic implementations utilizing random bit sequences selected from the entropy pool yield strong results in terms of both mathematical analysis and practical encryption performance. This study underscores the pivotal role of a high-quality random number generator in the development of secure cryptographic systems and demonstrates that the entropy pool approach offers an effective solution for contemporary cryptographic applications. The utilization of entropy pools derived from various optimization algorithms in the construction of cryptographic structures offers enhanced security properties in comparison to conventional methods. In subsequent studies, the proposed architecture can be evaluated in various cryptographic applications, and the impact of optimization algorithms on the entropy generation process can be investigated in greater depth.

Despite the promising results, this study has certain limitations that pave the way for future research. The primary limitation is the computational overhead associated with the optimization process for generating the entropy pool, which may be a constraint for real-time applications requiring extremely high-speed encryption. Furthermore, the performance of the proposed method was validated on a specific set of standard test images; its robustness against a wider array of image types, including medical and satellite imagery, remains to be explored. Future work will focus on enhancing the computational efficiency of the entropy generation process through hardware acceleration or parallel computing techniques. Additionally, the proposed architecture will be tested against a broader suite of cryptographic attacks and its application will be extended to other domains such as video encryption and secure communication protocols to further validate its generalizability and robustness.

REFERENCES

[1] Robshaw M, Billet O. New Stream Cipher Designs. vol. 4986. Springer Berlin Heidelberg; 2008. https://doi.org/10.1007/978-3-540-68351-3.

[2] Garipcan AM, Erdem E. Implementation and Performance Analysis of True Random Number Generator on FPGA Environment by Using Non-periodic Chaotic Signals Obtained from Chaotic Maps. Arab J Sci Eng 2019;44:9427–41. https://doi.org/10.1007/s13369-019-04027-x.

[3] Garipcan AM, Erdem E. A TRNG using chaotic entropy pool as a post-processing technique: analysis, design and FPGA implementation. Analog Integr Circuits Signal Process 2020;103:391–410. https://doi.org/10.1007/s10470-020-01605-0.

[4] Sheveleva AM, Belyaev SA. Development of the Software for Solving the Knapsack Problem by Solving the Traveling Salesman Problem. 2021 IEEE Conf. Russ. Young Res. Electr. Electron. Eng., IEEE; 2021, p. 652–6. https://doi.org/10.1109/ElConRus51938.2021.9396448.

[5] Asif M, Baig R. Solving NP-complete problem using ACO algorithm. 2009 Int. Conf. Emerg. Technol., IEEE; 2009, p. 13–6. https://doi.org/10.1109/ICET.2009.5353209.

[6] Eroz E, Tanyildizi E, Ozkaynak F. Determination of Suitable Configuration Parameters for Linear Feedback Shift Register using Binary Bat Optimization Algorithm. IEEE EUROCON 2021 - 19th Int. Conf. Smart Technol., IEEE; 2021, p. 348–51. https://doi.org/10.1109/EUROCON52738.2021.9535616.

[7] Zhao M, Yuan Z, Li L, Chen X-B. A novel efficient S-box design algorithm based on a new chaotic map and permutation. Multimed Tools Appl 2024;83:64899–918. https://doi.org/10.1007/s11042-023-17720-9.

[8] Podder D, Deb S, Banik D, Kar N, Sahu AK. Robust medical and color image cryptosystem using array index and chaotic S-box. Cluster Comput 2024;27:4321–46. https://doi.org/10.1007/s10586-024-04584-3.

[9] Duong P-P, Nguyen HM, Dao B-A, Kieu-Do-Nguyen B, Tran T-H, Hoang T-T, et al. Construction of Robust Lightweight S-Boxes Using Enhanced Logistic and Enhanced Sine Maps. IEEE Access 2024;12:63976–94. https://doi.org/10.1109/ACCESS.2024.3396452.

[10] Zhu H, Tong X, Wang Z, Ma J. A novel method of dynamic S-box design based on combined chaotic map and fitness function. Multimed Tools Appl 2020;79:12329–47. https://doi.org/10.1007/s11042-019-08478-0.

[11] Khan M, Asghar Z. A novel construction of substitution box for image encryption applications with Gingerbreadman chaotic map and S8 permutation. Neural Comput Appl 2018;29:993–9. https://doi.org/10.1007/s00521-016-2511-5.

[12] Nasry H, Abdallah AA, Farhan AK, Ahmed HE, Sobky WIE. Multi Chaotic System to Generate Novel S-Box for Image Encryption. J Phys Conf Ser 2022;2304:12007. https://doi.org/10.1088/1742-6596/2304/1/012007.

[13] Cassal-Quiroga BB, Campos-Cantón E. Generation of Dynamical S-Boxes for Block Ciphers via Extended Logistic Map. Math Probl Eng 2020;2020:1–12. https://doi.org/10.1155/2020/2702653.

[14] Eroz E, Tanyildizi E, Ozkaynak F. COLFSR - A Hybrid Random Number Generator Based on Chaos Optimisation and Linear Feedback Shift Register. Elektron Ir Elektrotechnika 2025;31:30–8. https://doi.org/10.5755/j02.eie.38291.

[15] Ozkaynak F. A Novel Random Number Generator Based on Fractional Order Chaotic Chua System. Elektron Ir Elektrotechnika 2020;26:52–7. https://doi.org/10.5755/j01.eie.26.1.25310.

[16] Knuth DE. Art of Computer Programming, The, Volumes 1-3 Boxed Set (3rd Edition). 3rd ed. Addison-Wesley Professional; 1998.

[17] Soto J, Rukhin A. Statistical Testing of Random Number Generators 1999.

[18] Field AP. Discovering statistics using IBM SPSS statistics 2024:1110.

[19] Özkaynak F, Yavuz S. Designing chaotic S-boxes based on time-delay chaotic system. Nonlinear Dyn 2013;74:551–7. https://doi.org/10.1007/s11071-013-0987-4.