IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 19 November 2023, accepted 20 December 2023, date of publication 22 December 2023, date of current version 3 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3346319

RESEARCH ARTICLE

Automated Chaos-Driven S-Box Generation and Analysis Tool for Enhanced Cryptographic Resilience

YıLMAZ AYDıN[1] **AND FATIH ÖZKAYNAK**[1,2]
[1]Department of Software Engineering, Fırat University, 23119 Elâzığ, Turkey
[2]Kriptarium Arge Yazılım Danışmanlık Savunma Sanayi ve Ticaret Ltd. Şti., 23119 Elâzığ, Turkey
Corresponding author: Fatih Özkaynak (ozkaynak@firat.edu.tr)

This work was supported in part by the Scientific and Technological Research Council of Turkey under Grant 121E600 and Grant 122E337.

ABSTRACT In a rapidly advancing world of technology, information security studies have become the backbone of the digital age, and steps in this area are critical. In this context, cryptography, in particular, plays a key role in ensuring the confidentiality, integrity and authentication of data. s-box structures provide a certain diversity and security layer in encryption algorithms, forming one of the key elements in this area. This study focuses on the design and analysis of s-box structures, examining the potential impact of chaos theory-based structures on encryption systems. First, it provides a comprehensive classification of existing s-box design proposals in the literature, and explores the contribution of chaos theory to the security features of these structures. The original contribution of the study is the results obtained with the help of the developed analysis and design program. The program optimizes levels of complexity, randomness, and resistance, and demonstrates the resistance of these new structures to cryptanalysis attacks. The paper also draws attention to open issues in the field of chaos-based s-box design and provides a road map for future research. It is estimated that all these findings will provide a common motivation for researchers in the relevant literature and constitute the basis for many practical practices.

INDEX TERMS Chaos theory, cryptography, information security, substitution box.

I. INTRODUCTION

The relationship between information security and the digitization of society shows a remarkable parallel. The process began with the use of computers by limited experts in the 1960s and early 1970s, and gained new dimensions with the spread of personal computers in the 1980s. However, awareness of information security at that time was at levels to be lost. The 1990s, with the opening of the Internet to the public, accelerated the exchange of information, but also increased cyber-attacks. In the 2000s, cybersecurity awareness rose with the rapid increase in Internet use, and companies began to take various security measures to protect personal information. The 2010s, with the rise of big data analytics and cloud computing, allowed more data to be stored and processed in a digital environment. This has made data security issues even more important. In the 2020s, however, the rise of technologies such as artificial intelligence and the Internet of Things (IoT) made cyber threats more complex and made it compulsory to update information security strategies [1]. In the future, the emergence of new technologies, such as quantum computers, has made more research and development in this area a must, given the fact that it will bring more challenges in information security issues [2].

Cryptology is one of the cornerstones of information security and therefore continues to be a continuously evolving field. In collaboration with other sub-disciplines within the discipline of information security, cryptography is constantly undertaking new research to provide effective defense against emerging technologies and threats [3]. Substitution box (s-box) structures are notable as part of cryptographic algorithms. These structures allow certain bits or bytes to be replaced with a different value during encryption, which forms an important layer of encrypting security [4]. s-boxes,

The associate editor coordinating the review of this manuscript and approving it for publication was Aneel Rahim.

when properly designed and implemented, can increase the security of encryption algorithms. As a result, s-box design work has had a long historical development process in the field of cryptography. In the classical cryptography era, it was defined by simple encryption methods used with mechanical machines and electromechanical devices. However, the Feistel network structure, developed by Horst Feistel in the 1970s, laid the foundation of modern block encryption algorithms, and s-boxes became an important part of this network structure. During this period, the development of the Data Encryption Standard (DES) attracted particular interest in the design of the s-box. The safety of the DES was studied, through the mathematical properties of the s-boxes. This period marks the beginning of mathematical analysis in s-box design. In the following years, the Advanced Encryption Standard (AES), a new encryption standard, increased the interest in the design of the s-box [5]. During this period, work on how to design s-boxes in terms of resistance and safety has formed an important area. Nowadays, s-box designs are being redesigned to be resilient to future threats, especially under the influence of new perspectives such as quantum cryptography and post-quantum encryption. Among these efforts are chaotic s-box design work aimed at increasing the security levels of encryption algorithms and providing an effective defense mechanism against emerging threats.

This article deals with chaos-based substitution box designs and examines the existing literature in this field in a comprehensive way. An analysis program that can evaluate the s-box design criteria is required to be able to conduct such a review. In this context, the previously developed chaos analysis and design program [6] has been upgraded to a higher level and is designed to automatically generate more robust s-box structures. The role of chaos theory in s-box design has been further highlighted through our analytical and design generator software. Our program successfully integrates chaos theory to increase complexity, strengthen randomness properties, and optimize resistance levels. Another genuine contribution of the study to the literature was the results we obtained with the program we developed. It has been shown that suggesting new and powerful chaos-based s-box designs can enhance the security levels of encryption algorithms and resist cryptanalysis attacks. Another important point in the study is that it offers a critical assessment of chaos-based s-box literature, covering about twenty years. This drew attention to unresolved open problems in the area of chaos-based s-box design, and provided a projection for future work on this. It is estimated that this will enable researchers to further advance their work in this field.

The rest of the study is organized as follows. The purpose of the second section is to provide a comprehensive classification of existing proposals in the literature based on how the basic design methodologies emerged, the mathematical details of the metrics used as the success measure of the s-box structures, the relationship of these metrics with attack scenarios, and chaos, creating a solid starting point for both new researchers planning to work in this field and other researchers having the very unknown design details in the s-box literature. The third section shares details of the s-box design and analysis software developed to meet the requirements that are shaped in the framework of all these results. The fourth section discusses the potential contributions to literature of new s-box structures obtained using s-box design and analysis software. The fifth section focuses on open problems in chaos-based s-box design, and presents a projection of future work on this. This is thought to enable researchers to move their work in this field further. In the last section, all the results obtained were evaluated and the study was summarized.

II. SUBSTITUTION BOX STRUCTURES

s-box structures are considered one of the cornerstones of cryptography, and in encryption algorithms they transform input data into output data through a complex conversion. They essentially provide a non-linear transformation to solve the truthfulness problems of encryption algorithms. This non-linearity helps to increase resistance to differential and linear attacks in cryptoanalysis. In addition, s-box structures are used to increase the key dependency of encryption algorithms, which means that key changes should lead to major changes in s-Box outputs. s-box structures also support the diffusion and confusion principles of encryption algorithms, allowing small changes in input bits to have a broad impact on output bits. These structures are designed to enhance the security of cryptographic algorithms by strengthening mathematical properties, controlling the distribution of identical values, or increasing their randomness properties. As a result, s-box structures have a decisive influence on the security, resilience and performance of cryptographic algorithms, and play a fundamental role in the field of information security [7], [8].

Different techniques used in the design of s-boxes are often studied under three main categories: algebraic, geometric, and combinatorial methods [9]. S-boxes are designed to be created using mathematical processes. For example, modular arithmetic or linear algebra operations focus on the creation of s-boxes that provide specific characteristics. Geometric methods create s-boxes using geometric structures and concepts. In these methods, s-box tables can be designed by creating a specific matrix structure based on flat geometry or the properties of shapes. Combinatory methods generate s-boxes using different combinations and permutations. s-box structures can be created by combinatory operations, such as replacing or replacing input and output values within a particular structure. Each method aims to design s-boxes using different mathematical or logical operations, and this diversity allows different s-box structures to be designed according to the requirements of cryptographic algorithms. In parallel with this information, an alternative classification process has been tried to be presented below:

- Permutation and Substitution Method: In this method, a s-box is generated using the processes of permutation (change of location) and substitution combined. This involves changing the input bits in a specific way and then converting them to the output with a special mathematical transformation.

- Logical and Mathematical Operations: s-box structures can be created using Boolean logical operators (such as AND, OR, XOR, NOT) and mathematical operations (modular arithmetic, linear and nonlinear transformations). These methods focus on obtaining the desired s-box structures using various mathematical operations.
- Random Generatio: In some cases, s-box structures are developed by random creation and subsequent improvement of these structures with specific parameters or metrics. Heuristic algorithms such as simulated annealing can be used in this process.
- Matrix-based Design Approaches: Matrix operations, especially linear algebra techniques, are often used in the generation of s-box structures. It is commonly used in special matrix operations, and in particular in the design of the linear shape of the s-box, to certain characteristics.
- Chaos Theory and Nonlinear Dynamic Systems: Inspired by chaos theory and nonlinear dynamic systems, s-box structures can be designed. The properties of chaotic systems can be used to enhance properties such as randomness, complexity, and non-reality.

Each of these methods aims to design s-box structures using different mathematical and information techniques. These methods can be combined or adapted in different ways to enhance the safety, resistance and complexity features of the s-boxes [10].

While describing the relationship between any particular algorithm and the s-box design is usually more possible when there is a preference specific to a clearly defined standard or algorithm, this section attempts to present some relationships primarily from a general perspective. The logic behind the selected s-box design is often focused on strengthening security features, improving cryptographic metrics such as non-truthfulness, randomness, and resistance. Which S-box design is chosen for which algorithm is usually based on a specific standard, performance requirements, or security priorities [11]. For example, a specific s-box structure may be preferred to enhance the non-truth and resistance properties. Other features such as key dependency, confusion, and randomness can also affect the preferred s-box structure. These choices depend on the security requirements and performance targets of a specific encryption algorithm.

Table 1 shows that there are s-box structures in very different structures [12]. However, this paper focused on developing a software on the design and analysis of AES-like s-box structures. The most important factor in setting such a focus is that the Rijndael algorithm is an accepted standard for cryptographic security, and that the s-box structure has been supported over the years by mathematical analysis and cryptographical tests [12]. The AES s-box framework is carefully selected to be resistant to cryptographic attacks and designed to provide resistance to differential, linear cryptanalysis. In addition, AES-like s-box structures have been optimized for high performance and compliance with widely accepted standards. These structures, supported by mathematical analysis and in long-term use, have been widely recognized for their reliability because of their wide scope of use and a wide academic-industrial review process. For these reasons, AES-like s-box structures often provide more security than other alternatives in terms of reliability, security, and compatibility. After the idea of producing AES-like s-box structures were adopted as the focus of the study, the need to assess how well the s-boxes designed in the first phase are suited to important metrics such as security, resistance, complexity and randomness emerged. As a result of comprehensive evaluations of literature, the following key points have emerged to assess the role and importance of the s-box analysis program in cryptology studies:

- Security Assessment: The analytical programs to be developed should have the capacity to assess the security levels of the designed s-boxes. The s-box designs, which are weak in security, may be sensitive to potential attacks and cryptanalysis techniques. Analysis programs should help determine the level of security of the designed s-box by simulating possible attack scenarios and using mathematical assessments.
- Resistance and Confusion Analysis: Assessing the resistance and confusion levels of s-box designs is a fundamental measure in cryptology studies. Analysis programs must apply a variety of mathematical

TABLE 1. Commonly known S-box structures and basic properties.

Algorithm	S-box structure	Properties
AES	Nyberg Proposition, irreducible polynomials and inversion	It is known for its structure determined using inversion and irreducible polynomials.
Blowfish	32-bit specially prepared S-box tables	It contains special structures associated with the key.
Serpent	8x8 matrix-based structure	Matrices produce output based on predetermined values.
Twofish	key-associated S-box structures	It contains S-box structures in which the key acts through special operations and permutations.
Camellia	Logical and mathematical operations	It is created using Boolean logic operators and mathematical operations.
DES	Special structures and permutations	It is determined using permutations and special structures.
IDEA	16-bit tables	It contains specially prepared 16-bit tables.
RC6	Special structures and logical operations	It contains special structures created by logical operations.

techniques to determine and optimize these metrics. High resistance and confusion levels will make the s-box more resistant to crypto-analysis attacks.

- Randomity and Distribution Analysis: Random behavior of s-boxes may reduce the predictability of attackers. Analysis programs should be able to determine how close the designs are to the expected random behavior by evaluating the random characteristics of the s-boxes and the bit distribution.
- Performance Optimization: Analysis programs should be able to evaluate the performance of designed s-boxes and optimize them to speed up processes and use resources efficiently.
- Cryptanalysis Testing: Analysis programs should be able to assess whether the designed s-boxes are resistant to cryptoanalysis tests. These tests should be designed to measure levels of resistance to attacks, especially differential and linear cryptographic analysis.
- Compliance with standards: Analysis programs should be designed to verify compliance with specific cryptographic standards (e.g. NIST standards). This could provide an advantage in determining whether the designed s-box meets generally accepted safety standards.

It is an undeniable fact that analytics programs that can meet these requirements will have a significant impact on the security and performance of designed cryptographic algorithms. This will provide critical insights to cryptologists and security professionals on how to make their designs stronger and more resilient, thereby helping them develop effective defenses in the field of information security. However, since making these evaluations is not an easy process, the five main metrics used for s-box analysis in line with the need for quantitative evaluation criteria are:

- Nonlinearity: This metric measures the nonlinearity property of the s-box. An irregularity means that the slightest change between the input and output of an s-box causes the greatest change in the output. High nonlinearity values indicate a stronger s-box in terms of non-linearity.
- Bijection: These metric checks that the s-box generates an output value for each input value and that each output is matched to an input value. If the s-box generates a different output for each input and each output corresponds to an input, this shows the bijective property.
- SAC (Strict Avalanche Criterion): SAC measures how many changes a single bit changes in an s-box input causes in the output. High SAC values indicate that single-bit changes in inputs have a broad impact on outputs.
- Bit Independence Criterion (BIC): A bit independence criterion measures how independent the input and output bits of an s-box are from each other. If any of the s-box input bits change, the output bits are expected to change as randomly and independently as possible. This may indicate a statistically random distribution.
- XOR Distribution: This metric examines the rate of distribution in the outputs after the application of the XOR operation to the inputs of the s-box. A balanced XOR distribution ensures that the s-box is effectively distributed and that there is sufficient differentiation between inputs.

These metrics are used to evaluate and analyze the cryptographic security features of an s-box. The study of s-box designs according to these metrics helps create powerful and resilient encryption algorithms. In the following subsections, the details of each metric are examined in detail [4], [7], [8], [9], [10], [11].

A. NONLINEARITY

The nonlinearity value used to measure the non-linearity property of a s-box structure is determined by calculating the maximum of the absolute values of the minimum value of the linearity functions in the matrix obtained by the Walsh-Hadamard transformation of all the linear combinations between input and output values. The Walsh-Hadamard transformation is used to calculate the value of nonlinearity. This transformation is to create a linear table representing all possible differences between the input and output values of the s-box. As a first step, the input and output values are represented as bit vectors. For example, a 4-bit s-box has 16 input and output values, each of which is 4 bits long. A linear table is obtained by applying the Walsh-Hadamard transformation of these values. In another expression, the Walsh-Hadamard transform forms a matrix representing all the linear combinations between input and output values. A rationality function is then calculated for each linear combination. This calculation is done by minimizing the difference between the absolute value of each combination and half the number of bits of 2. In the final stage, the minimum values of the vertical functions represent the nonlinearity value of the s-box. Nonlinearity is calculated by taking the maximum of the absolute values of the minimum values derived from these functions. Table 2 presents a pseudo code that can be used to calculate the value of nonlinearity.

The value of nonlinearity calculated as a result of the analysis is preferred to be higher. Because the high nonlinearity value indicates the s-box's non-realistic attribute and can help it to be resistant to cryptanalysis attacks. However, an ''ideal'' nonlinearity value is not constant in all cases and may vary depending on the requirements of the cryptographic algorithm used. The nonlinearity value of the s-box structure of the AES is selected in accordance with the criteria specified in the design phase and is carefully designed to meet the cryptographic security requirements. The AES nonlinearity value is calculated as 112. The objective of the study is to produce AES-like s-box structures.

B. BIJECTION

The bijection criterion is a critical metric in s-box design and contains several important details. If an s-box is bijective,

TABLE 2. Pseudo code for nonlinearity criterion.

```
Function Calculate_Nonlinearity(S):
  n = 8  # Bit length of the AES S-box
  max_nonlinearity = 2^(n-1)

  result = 0
  for a = 1 to 2^n-1:
    if a == 1 then
      continue  # Exclude the case when a = 0
    min_sum = infinity  # Initialize minimum value
    for x = 1 to 2^n-1:
      if x == 0 then
        continue  # Exclude the case when x = 0
      fx = S[x]
      fx_xor_a = S[x XOR a]
      current_sum = AbsoluteValue(fx XOR fx_xor_a)

      if current_sum < min_sum then
        min_sum = current_sum

    if min_sum > max_nonlinearity / 2 then
      min_sum = max_nonlinearity - min_sum
    if min_sum > result then
      result = min_sum

  return result
```

it means that each input bit series must match a different output bit series. This criterion is based on the principle that the s-box must match each input value with a unique output value. Each input must have a single output and each output must have one single input, so that each input bit series corresponds to one output, and every output bit series to one input. This principle ensures that there is no duplicate matching; any input is not matched to more than one output, and any output does not match multiple inputs. This ensures that there is no collision and that each input matches a different output. The bijection criterion enhances the security features of the s-box, as it is resistant to linear and differential attacks because there is a unique match between each input and output bit series. Therefore, the bijection criterion in s-box design is critical for reliability and cryptographic stability. Several different alternatives can be observed to evaluate the bijective criterion of a s-box structure. These are:

- Matching Test: A matching test can be performed to determine whether each input bit series of the s-box matches a different output bit series. This test shows that the output is different for each input.
- Inverse Matching Test: Inverse matching test can also be performed. This means that each output bit series is checked to match a different input bit series.
- Collision Analysis: A collision analysis can be performed to determine whether there is any collision in the s-box. A collision is when two or more inputs or outputs match the same value.

The pseudo codes that can be used for these tests are given in Table 3.

TABLE 3. Pseudo codes for bijection criterion.

```
Function MatchingTest(S):
  For i = 0 to length(S) - 1:
    For j = 0 to length(S) - 1:
      If i ≠ j and S[i] = S[j]:
        Return False
  Return True

Function InverseMatchingTest(S):
  For i = 0 to length(S) - 1:
    For j = 0 to length(S) - 1:
      If i ≠ j and S[i] = S[j]:
        Return False
  Return True

Function CollisionAnalysis(S):
  For i = 0 to length(S) - 1:
    For j = 0 to length(S) - 1:
      If i ≠ j and S[i] = S[j]:
        Return True
  Return False
```

C. STRICT AVALANCHE CRITERION

The Strict Avalanche Criterion (SAC) is a metric that measures how much change a s-box changes in an input bit. The SAC determines the susceptibility of the s-box to single-bit changes. The SAC measures the non-linear properties of the s-box and evaluates the impact of single-bit changes on the s-box output. This metric is important to understand how strong or resistant the s-box is cryptographically.

The step-by-step process of calculating the SAC could be as follows:

- Creating Linear and Non-Linear Input Changes: All possible combinations representing changes in an input bit are created. For example, combinations involving the change of a single input bit from 0 to 1 or from 1 to 0.
- Application of Input Changes: Every possible input change is applied to the s-box and changes to the output are saved.
- Measurement of the Avalanche Effect: For each input change, the number of changes in the output bits is recorded.
- Calculating Average Avalanche Value: The sum of output changes caused by all input changes is calculated.
- Normalization of average Avalanche Value: The total value obtained is divided by the number of output bits of the s-box. This shows the average effect of average output changes on an input change.

These steps represent a general method for calculating the SAC. These steps are repeated for each input bit, thereby determining how much change the s-box changes to a single input bit. The pseudocode for how the SAC value can be calculated is shown in Table 4.

According to Table 4, we first need to determine the number of possible input and output values for the s-box. For example, if the s-box has 8-bit input and output values, there are $2^8 = 256$ possible input and output values. Next, we need to iterate through each possible input value and determine the number of bits that change in the output when a single bit is changed in the input. For example, if the input value 0×00 maps to the output value 0xFF and we change the input

TABLE 4. Pseudo codes for SAC criterion.

```
function calculate_SAC(sbox):
  num_inputs = 8
  num_outputs = 8
  sac = 0

  for i from 0 to num_inputs - 1:
    for j from 0 to num_outputs - 1
      input_diff = 1 shifted left by i
      output_before = sbox[input_diff]
      output_after = sbox[input_diff XOR (1 shifted left by j)]
      output_diff = output_before XOR output_after
      sac += count_bits_set(output_diff)

  return sac / (num_inputs * num_outputs)
```

value to 0×01, the output value would change to 0xFE. In this case, the number of bits that change in the output would be 1. After we have determined the number of bits that change in the output for each possible input value, we can calculate the SAC of the s-box using the Eq. (1):

$$\mathrm{SAC} = \mathrm{SUM}(\mathrm{Ni} * (\mathrm{n} - \mathrm{Ni}))/(\mathrm{n} * (\mathrm{n} - 1)) \tag{1}$$

where n is the number of possible input and output values and Ni is the number of bits that change in the output when the i-th input bit is changed. For example, if our s-box has 8-bit input and output values and the number of bits that change in the output when the input value 0×00 is changed is 1, the SAC of the s-box would be calculated as Eq. (2):

$$\mathrm{SAC} = \mathrm{SUM}(1 * (256 - 1))/(256 * (256 - 1)) \tag{2}$$

The SAC of the s-box is a value between 0 and 1. The SAC value should ideally be 0.5 This value represents the average of the change made at the output by any bit change in the s-box's inputs. If the SAC is closer to 0.5 this may indicate that the s-box is more balanced and safer. Because any change in the input makes a stronger mix that affects the output more.

D. BIT INDEPENDENCE CRITERION

The Bit Independence Criterion (BIC) is used to assess how independent the input and output bits of an s-box are from each other. This criterion measures that when the input bits of the s-box change, the output bits change as randomly and independently as possible. Thus, it may be a statistically random distribution indicator. A probability table is first created to measure how independent the probability values are for each combination of input and output bits. A probability table containing combinations of all inputs and outputs for the respective s-box is produced. For each input/output combination, the corresponding probability value is then calculated, with the degree of independence being calculated by matching the combined probability of the input and output. For example, P(a, b) = P (a) ∗ P(b), where "a" represents the input and "b" the output values. When calculating the degree of independence, the probability varies from the expected value. The absolute value of this difference constitutes the value of the bit independence criterion. For all degrees of independence, absolute differences are calculated and the sum of these values is taken. Total absolute difference represents the resulting value of the bit independence criterion. A higher BIC may indicate that the input and output bits are more independent, so the s-box behaves more statistically randomly. This could increase the level of security of the s-box.

While the BIC refers to the Bit Independence Criterion, additions such as BIC-SAC and BIC-Nonlinearity represent versions of the bit independence criterion associated with different analyses. These can be obtained as follows:

- BIC-SAC (Bit Independence Criterion - Strict Avalanche Kriterion): BIC-SAC refers to the relationship between the Bit Independence and Strict SAC criterion. The SAC measures how much change a single bit change in the inputs of an s-box creates in the output bits. BIC-SAC examines the relationship between SAC and BIC. The relationship between the BIC value and the SAC is usually expressed as follows: the higher that BIC, the better the s-box's SAC property. High BIC may indicate that the change in a single input bit is more likely to affect the output bits, and that this may be an undesirable situation.
- BIC-Nonlinearity (Bit Independence Criterion - Nonlinearity): BIC-Nonlinearity refers to the relationship between the Bit Independence Criterion and the nonlinearities of a s-box. Nonlinearity refers to the nonlinear property of an s-box. A higher nonlinearity value may indicate that the s-box is more resistant to linear cryptanalysis. The relationship between the BIC value and nonlinearity is usually expressed as follows: that Higher BIC values are often associated with high non linearity. This may indicate that the high independence criterion (BIC) may represent a s-box with a stronger nonlinearity attribute.

These relations indicate the connection of the Bit Independence Criterion with different cryptographic properties. Relationships between BIC, SAC and nonlinearity provide insight into the different analyses and properties of the s-box.

E. XOR DISTRIBUTION

The XOR distribution table is a table showing the distribution of the inputs and outputs of an s-box in the XOR process. This table shows the distribution of outputs for each input pair resulting from the XOR operation. The XOR distribution table is used to study the properties of the s-box, such as the relationship between non-linearity and input-output. To create a XOR distribution table, the XOR operation is first performed for each input pair: for every input value in the s-box with another expression, it is performed between this input value and all other input values. For example, the XOR operation between 0x00 and 0x01 is: 0x00-XOR-0x01=0x01. The XOR operations of the output values are then recorded. The resulting XOR results are compared to outputs when these inputs are applied to the s-box, and the

XORs of these outputs are calculated. For example, if the output value matched with 0x01 is 0x05, then the XOR operation is: In the final stage, the output XOR values obtained as a result of the XOR operation of each input pair are placed in the corresponding cells in the table. At the end of this step, a picture is obtained. This table shows the distribution of outputs from each input pair resulting from the XOR operation. Thanks to the XOR distribution table, properties such as truthfulness, randomness, and input-output independence can be evaluated. A properly dispersed XOR distribution table may indicate that the s-box is stronger and more complex.

The pseudo code in Table 5 takes a specific s-box structure and calculates the XOR process for each input pair.

TABLE 5. Pseudo codes for XOR distribution Table.

```
function create_XOR_distribution_table(sbox):
  num_inputs = 8
  num_outputs = 8
  xor_table = initialize_empty_table(num_inputs, num_outputs)

  for input_1 from 0 to num_inputs - 1:
    for input_2 from 0 to num_inputs - 1:
      input_diff = input_1 XOR input_2
      output_1 = sbox[input_1]
      output_2 = sbox[input_2]
      output_diff = output_1 XOR output_2
      xor_table[input_diff][output_diff] += 1

  return xor_table
```

The s-box structure of AES is known to have all values 4 in the XOR distribution table, especially for 8-bit inputs and outputs. This is due to a particular feature of the s-box structure of the AES and is one of its mathematical structures. The s-box of the AES ensures that the output XOR value is also complementary when the input XOR is complementary. That is, the output value between an input value and its complementary value is the same as that between the XOR resulting from the operation and the XOR resulting of the operation. This attribute causes all values to be shown as 4 in the XOR distribution table. This is due to a specific mathematical feature of the s-box structure of AES, and is designed for cryptographic security. However, the fact that all values in the XOR distribution table are 4 is not always called an "ideal" situation. While there is no "ideal" XOR distribution table value for each cryptographic situation, it is often more desirable to have specific XOR values at low frequencies. This may indicate that the s-box is more complex, non-linear, and more resistant to cryptanalysis attacks.

III. s-box ANALYSIS AND GENERATION PROGRAM

There is a variety of software for performance analysis of s-box structures [13], [14], [15], [16], [17]. Although each of these software is superior in different ways, a program was developed to analyze s-box structures such as AES, after it was found that there are certain requirements that existing programs do not meet, given a specific purpose. In the first version, only the AES-like s-box structure was developed because Rijndael's proposed 8bit input 8-bit output design approach to the s-box has the ideal features for the standard AES, as outlined in the second part. As an alternative to the AES s-box structure, the s-box structures need to have design metrics that compete with AES s-box structures. It is therefore a basic assumption that the program developed meets the following characteristics.

- Comprehensive Properties Assessment: It is important for the program to be able to comprehensively evaluate the cryptographic properties of s-boxes (precision, non-linearity, differential and linear correlation, differential and linear properties, balanced distribution, etc.). It should provide a wide range of analysis across different metrics and criteria.
- Performance and Speed: Efficient analysis programs are essential. It should be able to evaluate large-scale s-box structures quickly and effectively and deliver results within a reasonable timeframe.
- Flexibility and adjustability: It is important for programs to work with different s-box structures and to be able to analyze s-box structures of different lengths and sizes. It is also useful that they can be customized for different analysis needs.
- Reliable Results and Accuracy: Analysis programs should produce accurate results and be reliable. The algorithms and metrics in their evaluations must comply with standards accepted and approved by safety experts.
- User-Friendly Interface: It is important that users can easily use the program and understand the results. A user-friendly interface ensures that the program is used effectively.
- Expandability: Analysis programs should be designed to adapt to new discoveries or developments. New features and analysis methods should be easy to add or update.

Taking these factors into consideration, the new version of the software shown in Figure 1 has been prepared as part of this software.

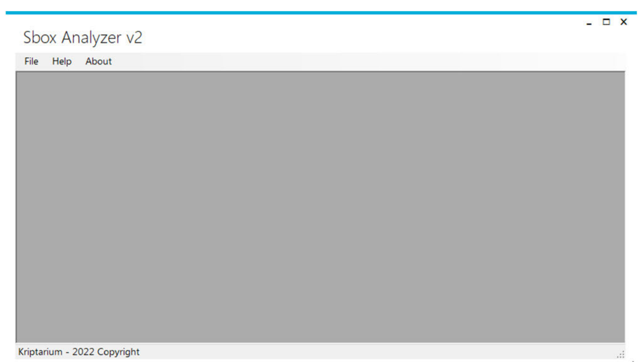

FIGURE 1. Main form of s-box analysis and generation software.

Figure 1 illustrates a straightforward yet highly functional interface that has been developed. The program's generator and analysis sections are easily accessible through the top menu. Additionally, a help menu is incorporated within the program. Developed within the Visual Studio development environment using the C# language, the program requires the installation of.Net Framework 4.8 or a higher version for operation. Supplementary files for download, helpful documents, indexed publications featuring s-box structures, and a comprehensive video tutorial outlining the program's functionalities can be accessed through the tool's web page [18].

The updated program offers three distinct methods for conducting s-box analysis. Firstly, as depicted in Figure 2, analysis can be initiated by reading data from a file. This file can contain data in integer, hexadecimal, or binary formats, separated by spaces, commas, semicolons, or new lines. Secondly, within the 'Discrete Time Chaotic Systems' section, an analysis module is instantly accessible upon s-box generation without the need to save it to a file. Thirdly, within the 'Discrete Time Chaotic Systems' module, a sequential automatic generator can undergo background analysis, a method that will be elaborated upon in subsequent sections. After determining the data in the first and second methods, the analysis screen is started as shown in Figure 3. Simultaneously, the program tests whether the data is bijective—ensuring that each number within the relevant sequence is unique. If the data lacks bijectivity, a warning is issued indicating "the selected s-box is not bijective." Conversely, if the data adheres to bijective criteria, the s-box data is visually presented on the screen as a 16 × 16 matrix. Users can select specific criteria for calculation at the top of the form field and proceed to the analysis screen by clicking the 'ANALYZE' button. Figure 4 demonstrates a summary of the analysis results obtained from the sample data."

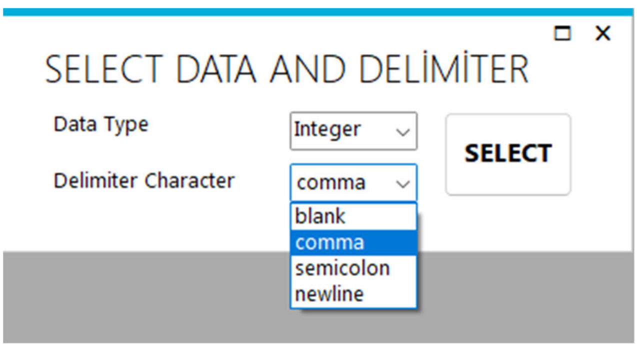

FIGURE 2. Input data screen.

The detailed results for each test are individually presented in Figures 5, 6, 7, and 8. In this paper's sample test, the AES standard s-box structure was utilized to showcase the program's efficient functionality. The analysis of the nonlinearity property of the s-box structure is detailed in Figure 5 with performance measurements. This measurement is very important for the s-box because s-box structures are the only nonlinear component in many cryptographic algorithms. Nonlinearity is expected to be high and the maximum nonlinearity value that can be achieved is 112.

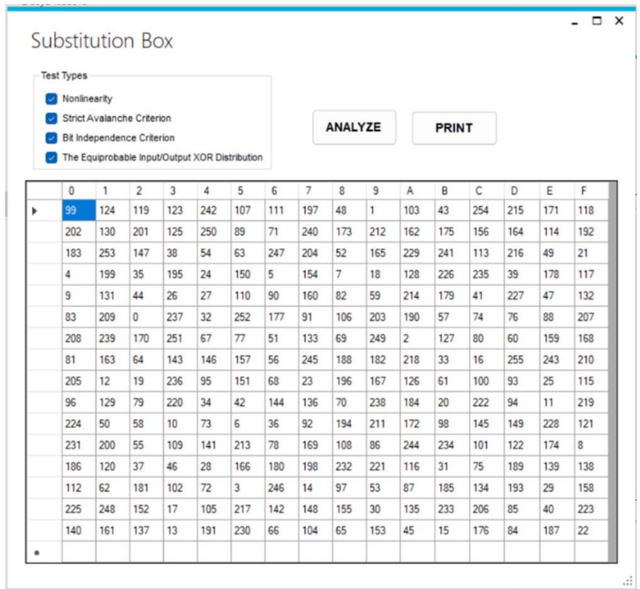

FIGURE 3. s-box preparation screen.

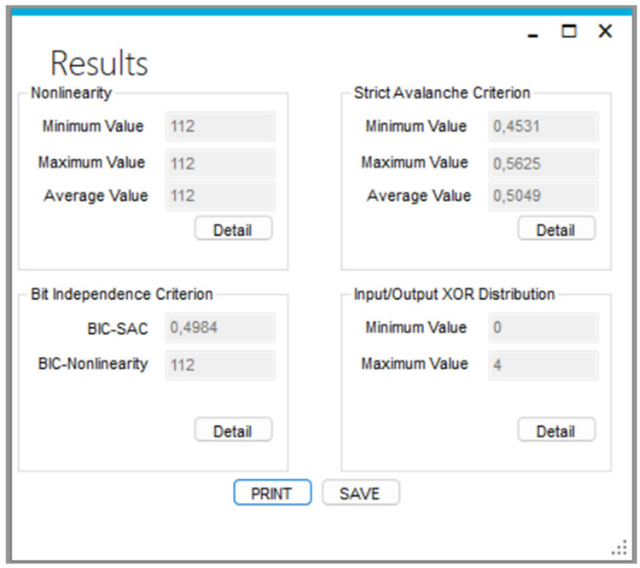

FIGURE 4. Main analysis summary screen.

FIGURE 5. Analysis report for nonlinearity criterion.

Another important measurement tool is the strict avalanche criterion (SAC). SAC analyzes the rate of change in the output bits when only one bit is changed in the input bits and is optimally expected to be 0.5. A SAC value of 0.5 means that changing one bit in the input bits changes half of the output bits and the detailed results are shown in Figure 6.

Strict Avalanche Values

0,5156	0,4688	0,5156	0,5312	0,4531	0,4531	0,5312	0,5156
0,5156	0,4844	0,5156	0,5312	0,5	0,5156	0,5312	0,5625
0,4531	0,5625	0,5	0,4688	0,4531	0,5156	0,4688	0,5156
0,5625	0,5	0,4688	0,4531	0,5156	0,4688	0,5156	0,5312
0,4531	0,4844	0,5625	0,5	0,5	0,4688	0,4688	0,4844
0,4844	0,4531	0,5	0,5312	0,5	0,5469	0,5312	0,5312
0,4531	0,5	0,5312	0,5	0,5469	0,5312	0,5312	0,4844
0,5	0,5312	0,5	0,5469	0,5312	0,5312	0,4844	0,5156

FIGURE 6. Analysis report for SAC criterion.

BIC SAC Values

0	0,4805	0,4902	0,4883	0,5078	0,4961	0,5039	0,4922
0,4805	0	0,5	0,5117	0,498	0,5117	0,4941	0,4922
0,4902	0,5	0	0,5273	0,4961	0,5215	0,498	0,5098
0,4883	0,5117	0,5273	0	0,4668	0,5195	0,4648	0,4883
0,5078	0,498	0,4961	0,4668	0	0,4785	0,5098	0,498
0,4961	0,5117	0,5215	0,5195	0,4785	0	0,4941	0,5312
0,5039	0,4941	0,498	0,4648	0,5098	0,4941	0	0,4844
0,4922	0,4922	0,5098	0,4883	0,498	0,5312	0,4844	0

BIC Nonlinearity Values

0	112	112	112	112	112	112	112
112	0	112	112	112	112	112	112
112	112	0	112	112	112	112	112
112	112	112	0	112	112	112	112
112	112	112	112	0	112	112	112
112	112	112	112	112	0	112	112
112	112	112	112	112	112	0	112
112	112	112	112	112	112	112	0

FIGURE 7. Analysis report for BIC criterion.

Input/Output XOR Distribution Table

4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	0

FIGURE 8. Analysis report for Input-output XOR values.

Another test criterion, the "bit independence criterion (BIC)" is shown in Figure 7. This test evaluates the "nonlinearity" and "avalanche criterion" of both inputs and outputs. The optimal values of BIC-nonlinearity and BIC-SAC are the same as the optimal values of nonlinearity and strict avalanche criterion. Therefore, a BIC-nonlinearity value of 112 and a BIC-SAC value of 0.5 are expected.

The final test is the difference distribution table (XOR table) test, which is used in conjunction with the existing tests to demonstrate the robustness of s-box structures to differential cryptanalysis. The largest value in the table is expected to be as small as possible, and a sample distribution is shown in Figure 8.

In the analysis results screen of the program, the results can also be printed out or saved to a file via the "SAVE" and "PRINT" buttons, so that the values can be evaluated together.

Another notable feature of the program involves the batch s-box testing module, enabling bulk testing of s-box structures within a designated folder. Prior to file selection, users are prompted to specify the number formats and brackets within the files, ensuring uniformity across the dataset. The program then sequentially analyzes these files, saving the results as.csv files. Figure 9 provides an illustration of this module's functionality. Moreover, the program can process files containing data of the same number type, each line representing an individual s-box. An important contribution of this study to the field is the performance analysis of chaos-based s-box structures facilitated by the proposed analysis program. The analysis results, as depicted in Table 6, offer a comparative reference for researchers entering this domain, aiding them in benchmarking their newly proposed s-box structures against prior studies.

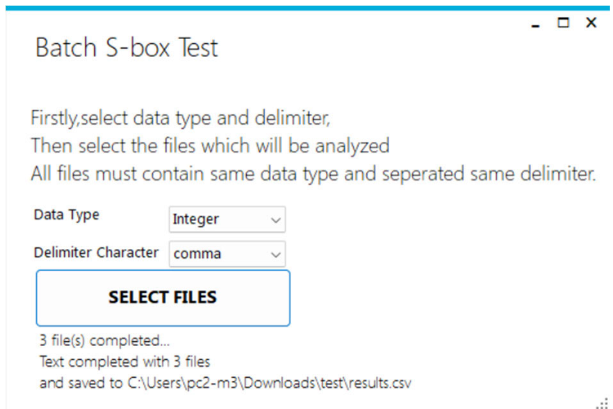

FIGURE 9. Batch test module.

Table 6 also includes some important results for the chaos-based s-box literature. For example, studies with the ∗ symbol next to them use optimization algorithms as a conversion process. For this reason, the success metrics of these algorithms are expected to be much higher.

IV. THE NEW PROPOSED CHAOS DRIVEN s-box STRUCTURES

The study makes a significant contribution through its s-box generator tools. While various algorithms exist in the literature for chaos-based s-box generators, the available generator functions remain limited. This research presents an s-box generator that utilizes random distribution, a continuous-time chaotic system-based generator, and a discrete-time chaotic systems-based generator. A notable innovation lies in the enhancement of the discrete-time s-box generator module, an improvement over the previous version. The number of discrete-time chaotic maps has been expanded to six, and two new s-box generator algorithms have been incorporated. The continuous-time systems employed include the Lorenz

TABLE 6. Performance analysis results of existing chaos-based s-box structures in the literature.

Ref. S-box	Strict Avalanche Criterion			Nonlinearity			Bit Independence Criterion		I/O XOR
	min	max	avg	Min	Max	Avg	SAC	Non.	
Ref. [4]	0,3906	0,6562	0,5027	102	108	104,75	0,4987	103	12
Ref. [6]	0.3906	0.5938	0.5034	102	108	104,70	0.4972	103.3	10
Ref. [10]	0,4219	0,6094	0,4995	108	110	109,25	0,5017	103,43	10
Ref. [11]	0,3906	0,5938	0,4978	98	106	102,75	0,502	103,36	12
Ref. [19]	0.4531	0.5625	0.5049	112	112	112,00	0.5046	112	4
Ref. [20]	0.4375	0.5625	0.5007	112	112	112,00	0.4997	112	4
Ref. [21]	0.4375	0.5625	0.4998	112	112	112,00	0.5026	112	4
Ref. [22]	0.4531	0.5625	0.5049	112	112	112,00	0.4992	112	4
Ref. [23]	0.4531	0.5625	0.5049	112	112	112,00	0.4992	112	4
Ref. [24]	0.4219	0.5469	0.5115	112	112	112,00	0.5027	108	8
Ref. [25]			0.4998	112	112	112,00	0.504	112	4
Ref. [26]	0.4375	0.5547	0.4998	112	112	112,00	0.504	112	4
Ref. [27]	-	-	0.504	112	112	112,00	0.4890	112	4
Ref. [28]	-	-	0.4951	112	112	112,00	-	112	4
Ref. [29]	0.4375	0.5625	0.5053	112	112	112,00	0.5013	112	4
Ref. [30]	0.438	0.563		112	112	112,00		112	4
Ref. [31]*	0,4375	0,5625	0,5022	112	112	112,00	0,5005	112	4
Ref. [32]*	0,4531	0,5781	0,5005	106	114	112,00	0,5005	104,21	10
Ref. [33]	0.4375	0.5781	0.5053	110	112	111,50	0.5053	111.3	4
Ref. [34]	0.4063	0.5781	0.5068	110	112	111,25	0.4950	111.5	10
Ref. [35]	-	-	0.5007	110	112	111,25	0.5034	102.5	10
Ref. [36]*	0,4531	0,5781	0,5103	110	112	110,75	0,4995	111,29	6
Ref. [37]	-	-	0.5046	108	112	110,50	-	-	8
Ref. [38]*	0,4062	0,5781	0,4929	110	112	110,50	0,4964	104,21	8
Ref. [39]*	0.4219	0.5781	0.4953	110	112	110,25	0.5021	104.07	10
Ref. [40]	0,4375	0,5938	0.5068	106	112	109,50	0,499	106.8	8
Ref. [41]*	0,4375	0,5781	0,5002	108	110	109,25	0,4975	109,71	6
Ref. [42]	0.4531	0.5156	0.5012	108	112	109,00	0.5012	104	8
Ref. [43]	0.4062	0.5938	0.5017	106	110	108,50	0.4971	104	10
Ref. [44]*	-	-	0.4995	106	110	108,50	0.5011	103.85	10
Ref. [45]*	0.3906	0.5781	0.491	108	110	108,50	0.5048	103.78	10
Ref. [46]*	0.4063	0.5781	0.5068	108	108	108,00	0.4950	90	10
Ref. [47]	0.4258	0.5175	0.5007	104	110	108,00	0,5006	112	12
Ref. [48]	0,4219	0.5781	0.5073	106	110	108,00	0.5020	104	10
Ref. [49]	0.4453	0.5313	0.4988	106	110	108,00	0.4969	102.85	12
Ref. [50]	0.4063	0.5781	0.4990	106	110	108,00	0.4961	104.2	10
Ref. [51]*	0.3750	0.6094	0.5093	106	110	107,50	0.5025	103.7	10
Ref. [52]	0.4065	0.5785	0.5	106	110	107,50	0.5048	105	10
Ref. [53]	0,3906	0,5938	0,4980	104	110	107,50	0,5022	103.5	10
Ref. [54]*	0,4219	0,5731	0,4944	106	108	107,50	0.5001	104.3	10
Ref. [55]*	0.4063	0.5625	0.5015	106	110	107,00	0.5010	105.5	10
Ref. [56]	-	-	0.5101	106	110	107,00	0.5006	106.2	10
Ref. [57]			0.4993	104	110	107,00	0.5050	100	10
Ref. [58]	0.4932	0.5625	0.4932	106	108	107,00	-	102.2	10
Ref. [59]	0,46094	0,53516	0,50056	104	108	106,75	0,50241	104	10
Ref. [60]	0.4219	0.6250	0.5034	106	108	106,70	0.4951	104	10
Ref. [61]	0.3909	0.6094	0.4941	106	108	106,70	0.4957	103.5	10
Ref. [61]	0.4063	0.4971	0.4063	106	108	106,70	0.4994	103.2	10
Ref. [62]	0.4062	0.625	0.4976	104	108	106,70	0.504	103.5	10
Ref. [63]	0.4219	0.5781	0.5015	106	110	106,70	0.5029	104.07	10
Ref. [64]*	0.4375	0.6406	0.5120	104	110	106,50	0.4984	105.2	10
Ref. [65]	0.4375	0.5938	0.4978	106	108	106,50	0.5003	104.2	10
Ref. [66]	0.4063	0.5781	0.4990	106	108	106,50	0.5033	103.5	10
Ref. [67]	0.4219	0.5938	0.5039	104	110	106,20	0.5023	102.3	10
Ref. [68]	0.4219	0.5781	0.501	106	108	106,20	0.5288	100	10
Ref. [69]	0.4375	0.5625	0.5059	104	110	106,20	0.5026	104	10
Ref. [70]	0.4219	0.5938	0.5002	102	108	106,00	0.4968	105.4	10
Ref. [71]	0.4375	0.625	0.5197	104	110	106,00	0.5014	104.2	10
Ref. [72]	0.4062	0.6406	0.5012	104	110	106,00	0.4977	103.5	10
Ref. [73]	0.4219	0.5781	0,4985	106	106	106,00	0,5029	103,14	10
Ref. [74]	0.5027	0.5938	0.4062	102	108	105,75	0.4972	102,36	10
Ref. [75]	0.4219	0.5938	0.4976	104	108	105,70	0.5032	104	10
Ref. [76]	0.4063	0.5781	0.5022	100	110	105,50	0.4983	107	32
Ref. [77]	0.4062	0.5937	0.4926	98	110	105,50	0.4994	105.7	32
Ref. [78]	0.4063	0.6094	0.5010	102	110	105,50	0.4988	104.3	12
Ref. [79]	0.4375	0.5781	0.5056	102	108	105,30	0.4971	104	10
Ref. [80]	0.4531	0.5469	0.4987	104	108	105,25	0.4990	102.6	10
Ref. [81]	0.4375	0.5625	0.5037	102	108	105,25	0.4994	102.6	10
Ref. [74]	0.4062	0.6094	0.5073	98	108	105,25	0.4986	103,86	10
Ref. [82]	0.4063	0.5781	0.5059	102	108	105,20	0.5013	104.3	12
Ref. [83]	0.4063	0.5938	0.5012	104	106	105,00	0.4994	103.4	10
Ref. [84]	0.4750	0.6093	0.5046	102	106	105,00	0.5004	103.6	10
Ref. [85]	0.4290	0.5850	0.4990	100	107	104,80	0.4890	104.7	12
Ref. [86]	0.4218	0.6093	0.4978	100	108	104,75	0.5009	103,6	12
Ref. [87]	0.3906	0.5937	0.5056	102	108	104,70	0.5021	104.1	12
Ref. [88]*	0.3906	0.5938	0.4037	100	108	104,70	0.4965	105	32
Ref. [89]	0.4218	0.5781	0.4982	100	108	104,70	0.4942	103.1	10
Ref. [90]	0,3906	0,5938	0,5034	102	108	104,75	0,4972	103,36	10
Ref. [91]	0.4219	0.6406	0.498	102	108	104,50	0.5013	104.6	12
Ref. [92]	0,4219	0,6094	0,502	102	106	104,25	0,4992	102,64	10
Ref. [93]*	0.3750	0.6093	0.4980	102	106	104,00	0.4971	103.2	10
Ref. [94]	0.2813	0.6094	0.4954	98	108	104,00	0.4967	102	12
Ref. [95]	0.3750	0.6250	0.4946	100	106	104,00	0.4990	102.5	10
Ref. [96]	0.4825	0.5175	0.5018	102	106	104,00	0.5019	103.5	10
Ref. [97]	0.4218	0.6093	0.5039	98	108	104,00	0.5078	104	12
Ref. [98]	0.3906	0.5781	0.5026	100	106	104,00	0.5033	103.2	10
Ref. [99]	0,2344	0,6094	0.5	96	108	104,00	0.498	102.8	10
Ref. [100]	0.3906	0.5781	0.5058	101	108	103,80	0.4958	102.6	14
Ref. [101]	0.4140	0.6328	0.5036	101	106	103,80	0.5037	103.4	10
Ref. [102]	0.4140	0.6015	0.4987	99	106	103,30	0.4995	103.3	10
Ref. [103]	0.4062	0.625	0.5058	99	106	103,30	0.5037	103.6	12
Ref. [104]	0.4218	0.5937	0.5048	100	106	103,20	0.5009	103.7	10
Ref. [105]	0.3671	0.5975	0.5058	98	108	103,20	0.5031	104.2	12
Ref. [106]	0.4218	0.6093	0.5	100	106	103,00	0.5024	103.1	14
Ref. [107]	0.3906	0.625	0.5039	96	106	103,00	0.5010	100.3	12
Ref. [108]	0.4062	0.5937	0.5012	98	108	103,00	0.4988	104.1	12
Ref. [109]	0,4062	6094	0,488	102	106	102,70	0,5027	105,75	8
Ref. [110]	0.3906	0.6719	0.5178	96	106	102,50	0.4026	102.5	54
Ref. [111]	0.3281	0.6016	0.4836	98	108	102,30	0.4992	100	14
Ref. [112]	0.4219	0.6094	0.5059	96	108	102,25	0.5050	103.5	16
Ref. [113]	0.125	0.625	0.4812	84	106	100,00	0.4962	101.9	16
Ref. [114]	0.125	0.625	0.4812	84	106	100,00	0.4962	101.9	16
Ref. [115]	0,4531	0,5625	0,5049	112	112	112	0,5017	112	4
Ref. [116]	0,4531	0,5625	0,5049	112	112	112	0,5034	112	4
Ref. [117]	0,3438	0,6094	0,5027	104	110	106,75	0,4976	103,57	10
Ref. [118]	0,4375	0,5781	0,5042	108	110	109,75	0,4953	109,86	6
Ref. [119]	0,4062	0,5625	0,5017	102	108	104,75	0,5077	103,5	10
Ref. [120]	0,4375	0,5625	0,4978	112	116	114	0,4976	103,86	12
Ref. [121]	0,4531	0,5625	0,5049	112	112	112	0,4984	112	4
Ref. [122]	0,4062	0,5781	0,4944	102	110	106,5	0,5052	103,36	12
Ref. [123]	0,4219	0,5625	0,5066	108	112	110,5	0,4972	106,43	10
Ref. [124]	0,375	0,5938	0,4937	108	110	109	0,5001	103,86	10
Ref. [125]	0,4375	0,5625	0,5034	112	112	112	0,4985	112	4
Ref. [126]	0,4375	0,5781	0,501	110	112	111,25	0,4989	103,79	10
Ref. [127]	0,3594	0,6094	0,4973	100	108	105	0,4971	104,21	10
Ref. [128]	0,4062	0,5781	0,5029	102	112	108	0,5012	104,43	14
Ref. [129]	0,3906	0,625	0,4929	100	106	104,5	0,4999	103,5	12
Ref. [130]	0,4375	0,5469	0,502	116	116	116	0,5013	104,86	10
Ref. [131]	0,4062	0,5938	0,4983	106	110	107,75	0,5007	104,14	10
Ref. [132]	0,4219	0,6094	0,501	106	108	106,25	0,5001	103,14	12
Ref. [133]	0,4375	0,5625	0,5029	110	112	111,75	0,4997	111,57	4
Ref. [134]	0,3906	0,5938	0,498	108	112	109,5	0,5041	104,21	10
Ref. [135]	0,3906	0,5938	0,4973	106	110	107,5	0,4989	103,57	10
Ref. [136]	0,4062	0,5938	0,4954	102	108	103,75	0,4976	103,14	10
Ref. [137]	0,4062	0,5781	0,501	106	110	106,5	0,4987	103,93	10
Ref. [138]	0,4062	0,5938	0,5076	100	106	103,75	0,501	103,79	14
Ref. [139]	0,4219	0,5938	0,4985	106	110	108,5	0,5005	104	10
Ref. [140]	0,4375	0,5625	0,4998	112	112	112	0,4939	112	4
Ref. [141]	0,375	0,5781	0,4998	110	112	111,25	0,4958	102,57	12
Ref. [142]	0,4375	0,5781	0,5056	112	112	112	0,4991	104	10
Ref. [143]	0,4062	0,5781	0,4973	110	112	111	0,4977	103,43	10
Ref. [144]	0,4062	0,5781	0,4985	110	112	110,75	0,4964	103,86	12
Ref. [145]	0,4375	0,5781	0,5144	110	112	110,25	0,4996	104,14	10

System, Chua System, and Chaotic Labyrinth Rene Thomas System. For discrete-time systems, enhanced formulas are applied to the Logistic map, Circle map, Sine map, Cosine map, Square map, and Tent map, as outlined in Table 7, illustrating the formulas of the chaotic maps used [146].

TABLE 7. Pseudo codes for XOR distribution Table.

Name	Equation	μ
Logistic map	$y_{n+1} = \mu y_n(1 - y_n)$	4.0
Square map	$y_{n+1} = 1 - \mu y_n^2$	2.0
Cosine map	$y_{n+1} = \cos(\mu y_n)$	6.0
Tent map	$y_{n+1} = \mu \min\{y_n, 1 - y_n\}$	1.9999
Sine map	$y_{n+1} = -\mu \sin(y)$	4.0
Circle map	$y_{n+1} = y_n - \mu \sin(y_n)$	4.5

The method proposed in Özkaynak [61] is used as the s-box design algorithm. The low complexity of the algorithm is an advantage and it has been observed that it achieves high performance values. The working logic of the algorithm is explained step by step.

Step 1. One of 6 different discrete-time chaotic systems or 3 different continuous-time chaotic systems is selected.

Step 2. State variables are calculated according to initial values and control parameters

Step 3. A state variable is selected for continuous-time chaotic systems

Step 4. The selected state variable is normalized between 0 and 255 with the mod 256 function

Step 5. If the value obtained is not in the table, add it, otherwise go to the next loop

Step 6. Repeat steps 4-6 until the entire table is filled

In the discrete-time chaotic systems module of the study, three new maps were added and three different methods were developed for initial values. The first one is s-box structures generated with random numbers generated entirely by the system. The generated s-box can be saved to a file with the "Save" button. The second is to generate an s-box with a random initial value generated by the system according to the selected map and the module is shown in Figure 10.

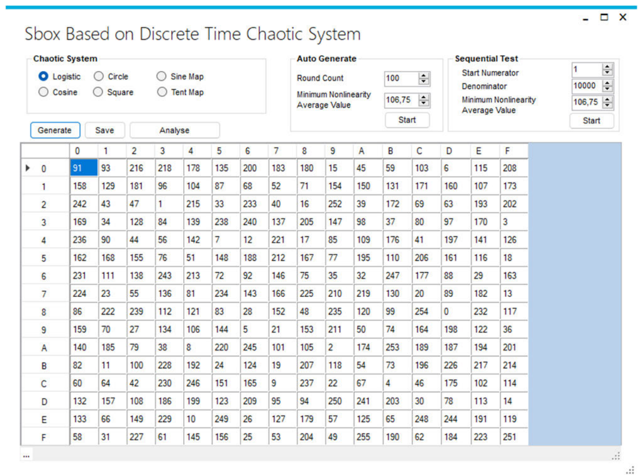

FIGURE 10. s-box auto generation module.

One of the six chaotic maps detailed in Table 7 is selected, and following the selection of initial conditions, the s-box is generated using the aforementioned algorithm and displayed on-screen. The generated values can be saved to a file by clicking the 'Save' button. Alternatively, the 'Analyze' button allows direct transfer of generated values to the analysis screen without saving them first. The study introduces two innovative options for automatic s-box generation to yield more results. Both options involve generating the s-box, and if it meets specified criteria, it is saved to a file. In the 'Auto Generate' section, users input the number of s-boxes to generate ('s-box Count') and set a minimum average nonlinearity target ('Minimum Nonlinearity Average Value'). Clicking 'Start' initiates continuous generation of s-box structures according to the algorithm. If the generated s-box meets the conditions, it's automatically saved to the program's folder, streamlining thousands of manual operations. The last method involves s-box generation through the sequential test approach. Within the 'Sequential Test' section, users input the initial numerator and denominator, along with the targeted minimum nonlinearity value. The system progressively increments the numerator by one and divides it by the denominator, utilizing the resulting decimal value as the initial value for the selected map. This method ensures unique, rule-following seed values. If the test achieves the targeted nonlinearity value, the initial value and map name are saved to the file. The Table 8 displays the efficient results obtained through this method.

V. DISCUSSION AND OPEN PROBLEMS

As part of the study, the chaos-based s-box literature was tried to be discussed with all the details. Table 6 aims to present an overview of a broad literature. The fundamental motivation behind the emergence of so many studies is to achieve a design that can compete with the AES s-box structure in terms of design metrics. Because the AES s-box structure can fail in modern attack scenarios such as algebraic attacks and application attacks [1], [2]. A capable attacker can leverage artificial intelligence and machine learning techniques to obtain the algorithm's secret key using side-channel information such as heat, light, noise, and power consumption. Therefore, random selection-based s-box designs have started to become increasingly popular. The new s-box studies produced within the scope of this study should also be evaluated within this context. During the study, thousands of s-box structures were generated. The performance metrics of only around 50 of these structures are provided in the fourth section. It is expected that these structures will offer significant contributions in various practical applications.

The studies listed in Table 6 can be grouped into three main categories: s-box structures derived mathematically, s-box

TABLE 8. Analysis results for generated new s-boxes.

Map	Initial Value	Average NL Value	Average BIC	Average SAC	Max XOR
S1 (Logistic)	0,357509274721758	106,75	0,495	0,4944	10
S2 (Logistic)	0,236992890213294	107	0,5045	0,4998	10
S3 (Logistic)	0,960471185864424	107,25	0,4997	0,5037	10
S4 (Logistic)	0,693900570784018	107	0,4967	0,4976	10
S5 (Logistic)	0,638498122052582	107	0,4964	0,5002	10
S6 (Logistic)	0,555400448787434	107	0,5025	0,499	10
S7 (Logistic)	0,683940849656210	107	0,4982	0,5085	10
S8 (Logistic)	0,316059150343790	107	0,4982	0,5085	10
S9 (Logistic)	0,201514357597987	107,5	0,4981	0,498	10
S10 (Cosine)	0,614051578452646	106,75	0,4927	0,5015	10
S11 (Cosine)	0,0794776156715299	106,75	0,5007	0,4995	10
S12 (Cosine)	0,61801145965621	107,25	0,5006	0,5085	10
S13 (Cosine)	0,660769498454043	107	0,5033	0,5042	10
S14 (Cosine)	0,440495666121349	107	0,502	0,5007	10
S15 (Tent)	0,866304010879674	106,75	0,4966	0,5083	10
S16 (Tent)	0,967220983370499	106,75	0,5002	0,4998	10
S17 (Tent)	0,63131106066818	107	0,4987	0,5044	10
S18 (Tent)	0,694436555776438	107	0,5001	0,5083	10
S19 (Tent)	0,305563444223562	107	0,5001	0,5083	10
S20 (Tent)	0,345178335006620	107,25	0,4971	0,5034	10
S21 (Circle)	0,0167494975150745	106,75	0,5003	0,4956	10
S22 (Circle)	0,727558173254802	106,75	0,4991	0,4956	10
S23 (Circle)	0,331160065198044	107	0,4994	0,4998	10
S24 (Circle)	0,771366401740751	107	0,5007	0,4973	10
S25 (Circle)	0,764514593591379	107	0,5031	0,5012	10
S26 (Circle)	0,531853108112973	107	0,4987	0,5039	10
S27 (Circle)	0,464199002427932	107	0,4973	0,501	10
S28 (Circle)	0,979492574207922	107,25	0,5033	0,501	10
S29 (Sine)	0,984160443507582	107	0,4926	0,4995	10
S30 (Sine)	0,602343134392237	107	0,5026	0,5063	10
S31 (Sine)	0,580659741527237	107	0,501	0,5012	10
S32 (Sine)	0,499334018647478	107	0,4971	0,5085	10
S33 (Sine)	0,347030283152072	107	0,4957	0,502	10
S34 (Sine)	0,140396068910071	107	0,4963	0,4968	10
S35 (Sine)	0,502193938569720	107	0,5019	0,5081	10
S36 (Sine)	0,187522749363018	107,25	0,5019	0,5039	10
S37 (Sine)	0,864484065478036	106,75	0,4995	0,5073	10
S38 (Sine)	0,0316090517284481	107	0,5007	0,4973	10
S39 (Sine)	0,495635130946072	107	0,5001	0,5005	10
S40 (Square)	0,289715887955137	107	0,498	0,491	10
S41 (Square)	0,358737955337251	107,25	0,5066	0,4988	10
S42 (Square)	0,574731907506590	107,25	0,4964	0,5039	10
S43 (Square)	0,331338722515770	107,5	0,4973	0,5	10
S44 (Square)	0,908972730818076	107	0,4987	0,5066	10
S45 (Square)	0,553003409897703	107	0,5029	0,4973	10
S46 (Square)	0,530384088477346	107,25	0,4968	0,5044	10

designs based on specific transformations, and s-box designs based on chaotic transformations. Within designs based on chaotic transformations, there are subcategories such as discrete-time, continuous-time, hyper-chaotic, time-delay, fractional-order, spatiotemporal, and other transformations. However, due to the lower design metrics compared to mathematically derived designs, studies using optimization techniques to significantly enhance the design metrics are considered as a separate group. A recent development in this field is the proposed post-processing techniques to enhance the performance of chaotic-based s-box structures. Showing that designs based on post-processing techniques can improve the nonlinearity property of s-box structures as much as the nonlinearity property of the AES s-box structure has brought a new dimension to the literature [141], [142], [143], [144], [145]. From this point, it is anticipated that addressing the following open problems in future studies will significantly contribute to the relevant literature.

Open Problem 1: An exploration of alternative post-processing techniques should be conducted to list their advantages and disadvantages. A comparison should be made between these techniques and optimization-based approaches to determine the best practices in this field.

Open Problem 2: For chaos-based s-box literature, the fundamental metric considered so far has been the nonlinearity criterion. Following recent studies that have shown improvements in the nonlinearity value up to 112, it is now necessary to demonstrate that the same success can be achieved within the XOR distribution table. When reviewing Table 6, it is observed that in chaos-based studies, the lowest nonlinearity value is 10. Efforts should be focused on studies that could reduce this value to 4.

Open Problem 3: Efforts to reduce the maximum value in the XOR distribution table to below 10 should be reiterated for optimization-based approaches. Specifically, it is observed that the studies in the literature tend to employ single-objective optimization algorithms. There is a need to reconsider contributions to the literature using an objective function based on all s-box design metrics.

Open Problem 4: We are currently in an era of artificial intelligence where significant advancements are being made in various fields using AI algorithms. However, successful AI applications require reliable data sources. Through the proposed s-box generation software in this study, thousands of different s-box structures meeting specific requirements can be easily generated. By transforming these generated s-box structures into an s-box dataset, the general characteristics of successful s-box structures can be learned using a black-box model. This allows for the acquisition of robust s-box structures from different perspectives.

Open Problem 5: The study conducted analyses using five widely accepted metrics for evaluating s-box structures. However, there are many alternatives, such as the Bent Index Criterion, that can be used in the evaluation of s-box structures. The proposed software has been designed in an expandable form. Future studies should delve into how different metrics can be integrated into the s-box analysis process in detail.

VI. CONCLUSION

Because many of the encryption algorithms are the underlying constructors, research has been carried out on how changes in the design of s-box structures can affect the resilience of the cryptographic algorithms. These researches revealed that new and more complex s-box designs could provide improved resistance, but also require more computing power. This revealed an effort to strike a balance between security and performance. As a result, research into s-box structures aims to raise security standards in the field of cryptography and create more effective defences against new threats. In parallel, the chaos-based s-box design has been the subject of serious research over the last two decades, especially after the deterministic structure of mathematical-based s-box projects has been shown to be a disadvantage in application-oriented attacks such as side channel analysis. However, the fact that the performance metrics of chaos-based s-box structures are not as good as the AES s-box structure occupies the researchers' agenda as a problem awaiting solution in the literature.

The aim of this study was to develop a s-box analysis and design program that could form the basis of literature for this problem. The enhanced analytical tool meets a critical requirement by providing a method that can evaluate core cryptographic properties for AES-like s-box structures. The program enables researchers to evaluate s-box structures using verticality, non-linearity, differential properties, XOR distribution, and other important metrics. Thanks to this ability, it will have a standard assessment approach for all researchers for chaos-based s-box literature. The miscalculation of the cryptographic properties of some of the previously suggested s-box structures confirms once again the need for such a standard evaluation program.

One of the most important advantages of the enhanced analytical tool is that chaos theory is effectively integrated into the s-box design phase and represents an innovative approach to s-box design. The study suggests the possibility of increasing the security levels of the s-box structures by simultaneously analyzing several s-boxes and producing s-box structures that provide a certain nonlinearity value. Thousands of s-box structures have been produced with the program. Newly designed s-box structures tend to have higher complexity, randomness, and non-realistic characteristics. This has been seen as an important step towards increasing the security levels of modern encryption algorithms and making them more resistant to cryptanalysis attacks. As a result, the development of chaos theory-based s-box analysis and design programs could enhance the security levels of cryptographic systems, and enable development of next-generation encryption algorithms. This study represents an important step in this direction and aims to be an inspiration for future research. Another important motivation for the study to contribute to future work is the sharing of a list of open problems that could be collaborated in this field in the future. With more robust s-box structures to be developed to solve these problems, it is envisaged that a wide range of practical work can be influenced, from image encryption to key generators, from block encryptions to the design of symmetrical encrypting algorithms.

REFERENCES

[1] I. Verbauwhede and B. S. Ors, "Side-channel analysis attacks on hardware implementations of cryptographic algorithms," in *Wireless Security and Cryptography: Specifications and Implementations*. Boca Raton, FL, USA: CRC Press, 2007.

[2] Y. Bürhan and F. Özkaynak, "The effects of knowledge extraction approaches on cryptanalysis studies and analysis of the success of chaos-based countermeasures," in *Studies in Computational Intelligence*, vol. 1040. Cham, Switzerland: Springer, 2023, pp. 189–202, doi: 10.1007/978-3-031-07707-4_23.

[3] C. Paar and J. Pelzl, *Understanding Cryptography*. Berlin, Germany: Springer, 2010.

[4] E. Turan, M. K. Özdemir, B. Karakaya, and F. Özkaynak, "A substitution-box structure based on solar panel data," *Turkish J. Sci. Technol.*, vol. 17, no. 1, pp. 143–149, Mar. 2022, doi: 10.55525/tjst.1034034.

[5] J. Daemen and V. Rijmen, "The design of Rijndael: The advanced encryption standard (AES)," in *Information Security and Cryptography*, 2nd ed. Berlin, Germany: Springer, 2020.

[6] F. Özkaynak, "An analysis and generation toolbox for chaotic substitution boxes: A case study based on chaotic labyrinth rene Thomas system," *Iranian J. Sci. Technol., Trans. Electr. Eng.*, vol. 44, no. 1, pp. 89–98, Mar. 2020, doi: 10.1007/s40998-019-00230-6.

[7] C.-K. Wu and D. Feng, "Boolean function representation of S-boxes and Boolean permutations," in *Boolean Functions and Their Applications in Cryptography*. Berlin, Germany: Springer, 2016.

[8] T. W. Cusick and P. Stanica, *Cryptographic Boolean Functions and Applications*. Oxford, U.K.: Elsevier, 2009.

[9] K. Nyberg, "Differentially uniform mappings for cryptography," in *Advances in Cryptology—EUROCRYPT 1993* (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Berlin, Germany: Springer, 1994, doi: 10.1007/3-540-48285-7_6.

[10] F. Artuger and F. Özkaynak, "An effective method to improve nonlinearity value of substitution boxes based on random selection," *Inf. Sci.*, vol. 576, pp. 577–588, Oct. 2021, doi: 10.1016/j.ins.2021.07.036.

[11] F. Artuger and F. Özkaynak, "A method for generation of substitution box based on random selection," *Egyptian Informat. J.*, vol. 23, no. 1, pp. 127–135, Mar. 2022, doi: 10.1016/j.eij.2021.08.002.

[12] L. R. Knudsen and M. J. B. Robshaw, *The Block Cipher Companion*. Berlin, Germany: Springer, 2011.

[13] S. Picek, L. Batina, D. Jakobović, B. Ege, and M. Golub, "S-box, SET, match: A toolbox for S-box analysis," in *Information Security Theory and Practice. Securing the Internet of Things* (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Berlin, Germany: Springer, 2014, doi: 10.1007/978-3-662-43826-8_10.

[14] R Core Team. (2013). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. [Online]. Available: http://www.R-project.org/

[15] F. Lafitte. The Boolfun Package: Cryptographic Properties of Boolean Functions. R Cran. Accessed: Dec. 10, 2023. [Online]. Available: http://cran.rproject.org/package=boolfun

[16] F. Lafitte, D. Heule, and J. Hamme, "Cryptographic Boolean functions with R," *R J.*, vol. 3, no. 1, p. 44, 2011, doi: 10.32614/rj-2011-007.

[17] W. Stein, *Sage Mathematics Software*. Newbury Park, CA, USA: Sage, 2017.

[18] F. Özkaynak. *Chaotic Key Tool*. [Online]. Available: http://www.chaotickeys.com/sbox.html

[19] J. Daemen and V. Rijmen, *The Design of Rijndael*. Berlin, Germany: Springer, 2002, doi: 10.1007/978-3-662-04722-4.

[20] L. Cui and Y. Cao, "A new s-box structure named affine-power-affine," *Int. J. Innov. Comput. Inf. Control*, vol. 3, no. 3, pp. 751–759, 2007.

[21] M. T. Tran, D. K. Bui, and A. D. Duong, "Gray S-box for advanced encryption standard," in *Proc. Int. Conf. Comput. Intell. Secur. (CIS)*, 2008, pp. 253–258, doi: 10.1109/CIS.2008.205.

[22] I. Hussain, T. Shah, M. A. Gondal, and H. Mahmood, "An efficient approach for the construction of LFT s-boxes using chaotic logistic map," *Nonlinear Dyn.*, vol. 71, nos. 1–2, pp. 133–140, Jan. 2013, doi: 10.1007/s11071-012-0646-1.

[23] I. Hussain, T. Shah, M. A. Gondal, and H. Mahmood, "Efficient method for designing chaotic s-boxes based on generalized Baker's map and TDERC chaotic sequence," *Nonlinear Dyn.*, vol. 74, nos. 1–2, pp. 271–275, Oct. 2013, doi: 10.1007/s11071-013-0963-z.

[24] A. Belazi, A. A. Abd El-Latif, A.-V. Diaconu, R. Rhouma, and S. Belghith, "Chaos-based partial image encryption scheme based on linear fractional and lifting wavelet transforms," *Opt. Lasers Eng.*, vol. 88, pp. 37–50, Jan. 2017, doi: 10.1016/j.optlaseng.2016.07.010.

[25] A. Anees and Y.-P.-P. Chen, "Designing secure substitution boxes based on permutation of symmetric group," *Neural Comput. Appl.*, vol. 32, no. 11, pp. 7045–7056, Jun. 2020, doi: 10.1007/s00521-019-04207-8.

[26] I. Hussain, "True-chaotic substitution box based on Boolean functions," *Eur. Phys. J. Plus*, vol. 135, no. 8, p. 663, Aug. 2020, doi: 10.1140/epjp/s13360-020-00666-4.

[27] M. S. M. Malik, M. A. Ali, M. A. Khan, M. Ehatisham-Ul-Haq, S. N. M. Shah, M. Rehman, and W. Ahmad, ''Generation of highly nonlinear and dynamic AES substitution-boxes (s-boxes) using chaos-based rotational matrices,'' *IEEE Access*, vol. 8, pp. 35682–35695, 2020, doi: 10.1109/ACCESS.2020.2973679.

[28] N. Siddiqui, F. Yousaf, F. Murtaza, M. Ehatisham-ul-Haq, M. U. Ashraf, A. M. Alghamdi, and A. S. Alfakeeh, ''A highly nonlinear substitution-box (s-box) design using action of modular group on a projective line over a finite field,'' *PLoS ONE*, vol. 15, no. 11, Nov. 2020, Art. no. e0241890, doi: 10.1371/journal.pone.0241890.

[29] N. Siddiqui, H. Khalid, F. Murtaza, M. Ehatisham-Ul-Haq, and M. A. Azam, ''A novel algebraic technique for design of computational substitution-boxes using action of matrices on Galois field,'' *IEEE Access*, vol. 8, pp. 197630–197643, 2020, doi: 10.1109/ACCESS.2020.3034832.

[30] I. Ullah, N. A. Azam, and U. Hayat, ''Efficient and secure substitution box and random number generators over mordell elliptic curves,'' *J. Inf. Secur. Appl.*, vol. 56, Feb. 2021, Art. no. 102619, doi: 10.1016/j.jisa.2020.102619.

[31] A. Mahboob, M. Asif, I. Siddique, A. Saleem, M. Nadeem, D. Grzelczyk, and J. Awrejcewicz, ''A novel construction of substitution box based on polynomial mapped and finite field with image encryption application,'' *IEEE Access*, vol. 10, pp. 119244–119258, 2022, doi: 10.1109/ACCESS.2022.3218643.

[32] A. R. Alharbi, S. S. Jamal, M. F. Khan, M. A. Gondal, and A. A. Abbasi, ''Construction and optimization of dynamic s-boxes based on Gaussian distribution,'' *IEEE Access*, vol. 11, pp. 35818–35829, 2023, doi: 10.1109/ACCESS.2023.3262313.

[33] A. Javeed, T. Shah, and A. Ullah, ''Construction of non-linear component of block cipher by means of chaotic dynamical system and symmetric group,'' *Wireless Pers. Commun.*, vol. 112, no. 1, pp. 467–480, May 2020, doi: 10.1007/s11277-020-07052-4.

[34] A. Razaq, A. Ullah, H. Alolaiyan, and A. Yousaf, ''A novel group theoretic and graphical approach for designing cryptographically strong nonlinear components of block ciphers,'' *Wireless Pers. Commun.*, vol. 116, no. 4, pp. 3165–3190, Feb. 2021, doi: 10.1007/s11277-020-07841-x.

[35] M. Ahmad and E. Al-Solami, ''Evolving dynamic s-boxes using fractional-order Hopfield neural network based scheme,'' *Entropy*, vol. 22, no. 7, p. 717, Jun. 2020, doi: 10.3390/E22070717.

[36] A. Haque, T. A. Abdulhussein, M. Ahmad, M. W. Falah, and A. A. A. El-Latif, ''A strong hybrid s-box scheme based on chaos, 2D cellular automata and algebraic structure,'' *IEEE Access*, vol. 10, pp. 116167–116181, 2022, doi: 10.1109/ACCESS.2022.3218062.

[37] M. Ahmad and E. Al-Solami, ''Improved 2D discrete hyperchaos mapping with complex behaviour and algebraic structure for strong s-boxes generation,'' *Complexity*, vol. 2020, pp. 1–16, Dec. 2020, doi: 10.1155/2020/8868884.

[38] M. Ahmad, R. Alkanhel, W. El-Shafai, A. D. Algarni, F. E. A. El-Samie, and N. F. Soliman, ''Multi-objective evolution of strong s-boxes using non-dominated sorting genetic algorithm-II and chaos for secure telemedicine,'' *IEEE Access*, vol. 10, pp. 112757–112775, 2022, doi: 10.1109/ACCESS.2022.3209202.

[39] Y. Wang, Z. Zhang, L. Y. Zhang, J. Feng, J. Gao, and P. Lei, ''A genetic algorithm for constructing bijective substitution boxes with high nonlinearity,'' *Inf. Sci.*, vol. 523, pp. 152–166, Jun. 2020, doi: 10.1016/j.ins.2020.03.025.

[40] M. Ahmad, I. A. Khaja, A. Baz, H. Alhakami, and W. Alhakami, ''Particle swarm optimization based highly nonlinear substitution-boxes generation for security applications,'' *IEEE Access*, vol. 8, pp. 116132–116147, 2020, doi: 10.1109/ACCESS.2020.3004449.

[41] M. Kang and M. Wang, ''New genetic operators for developing s-boxes with low boomerang uniformity,'' *IEEE Access*, vol. 10, pp. 10898–10906, 2022, doi: 10.1109/ACCESS.2022.3144458.

[42] D. Lambic, ''A novel method of s-box design based on chaotic map and composition method,'' *Chaos, Solitons Fractals*, vol. 58, pp. 16–21, Jan. 2014, doi: 10.1016/j.chaos.2013.11.001.

[43] E. Al Solami, M. Ahmad, C. Volos, M. Doja, and M. Beg, ''A new hyperchaotic system-based design for efficient bijective substitution-boxes,'' *Entropy*, vol. 20, no. 7, p. 525, Jul. 2018, doi: 10.3390/e20070525.

[44] H. S. Alhadawi, M. A. Majid, D. Lambic, and M. Ahmad, ''A novel method of s-box design based on discrete chaotic maps and cuckoo search algorithm,'' *Multimedia Tools Appl.*, vol. 80, no. 5, pp. 7333–7350, Feb. 2021, doi: 10.1007/s11042-020-10048-8.

[45] H. S. Alhadawi, D. Lambic, M. F. Zolkipli, and M. Ahmad, ''Globalized firefly algorithm and chaos for designing substitution box,'' *J. Inf. Secur. Appl.*, vol. 55, Dec. 2020, Art. no. 102671, doi: 10.1016/j.jisa.2020.102671.

[46] Y. Wang, K.-W. Wong, C. Li, and Y. Li, ''A novel method to design s-box based on chaotic map and genetic algorithm,'' *Phys. Lett. A*, vol. 376, nos. 6–7, pp. 827–833, Jan. 2012, doi: 10.1016/j.physleta.2012.01.009.

[47] X. Zhang, Z. Zhao, and J. Wang, ''Chaotic image encryption based on circular substitution box and key stream buffer,'' *Signal Process., Image Commun.*, vol. 29, no. 8, pp. 902–913, Sep. 2014, doi: 10.1016/j.image.2014.06.012.

[48] Y. Tian and Z. Lu, ''s-box: Six-dimensional compound hyperchaotic map and artificial bee colony algorithm,'' *J. Syst. Eng. Electron.*, vol. 27, no. 1, pp. 232–241, Feb. 2016.

[49] M. A. B. Farah, R. Guesmi, A. Kachouri, and M. Samet, ''A new design of cryptosystem based on s-box and chaotic permutation,'' *Multimedia Tools Appl.*, vol. 79, nos. 27–28, pp. 19129–19150, Jul. 2020, doi: 10.1007/s11042-020-08718-8.

[50] S. Ibrahim, H. Alhumyani, M. Masud, S. S. Alshamrani, O. Cheikhrouhou, G. Muhammad, M. S. Hossain, and A. M. Abbas, ''Framework for efficient medical image encryption using dynamic s-boxes and chaotic maps,'' *IEEE Access*, vol. 8, pp. 160433–160449, 2020, doi: 10.1109/ACCESS.2020.3020746.

[51] Y. Tian and Z. Lu, ''Chaotic s-box: Intertwining logistic map and bacterial foraging optimization,'' *Math. Problems Eng.*, vol. 2017, pp. 1–11, Jan. 2017, doi: 10.1155/2017/6969312.

[52] A. Javeed and T. Shah, ''Design of an s-box using Rabinovich–Fabrikant system of differential equations perceiving third order nonlinearity,'' *Multimedia Tools Appl.*, vol. 79, nos. 9–10, pp. 6649–6660, Mar. 2020, doi: 10.1007/s11042-019-08393-4.

[53] A. H. Zahid, E. Al-Solami, and M. Ahmad, ''A novel modular approach based substitution-box design for image encryption,'' *IEEE Access*, vol. 8, pp. 150326–150340, 2020, doi: 10.1109/ACCESS.2020.3016401.

[54] H. A. Ahmed, M. F. Zolkipli, and M. Ahmad, ''A novel efficient substitution-box design based on firefly algorithm and discrete chaotic map,'' *Neural Comput. Appl.*, vol. 31, no. 11, pp. 7201–7210, Nov. 2019, doi: 10.1007/s00521-018-3557-3.

[55] M. Ahmad, D. Bhatia, and Y. Hassan, ''A novel ant colony optimization based scheme for substitution box design,'' *Proc. Comput. Sci.*, vol. 57, pp. 572–580, Jan. 2015, doi: 10.1016/j.procs.2015.07.394.

[56] A. Shafique, ''A new algorithm for the construction of substitution box by using chaotic map,'' *Eur. Phys. J. Plus*, vol. 135, no. 2, p. 194, Feb. 2020, doi: 10.1140/epjp/s13360-020-00187-0.

[57] P. Zhou, J. Du, K. Zhou, and S. Wei, ''2D mixed pseudo-random coupling PS map lattice and its application in s-box generation,'' *Nonlinear Dyn.*, vol. 103, no. 1, pp. 1151–1166, Jan. 2021, doi: 10.1007/s11071-020-06098-0.

[58] B. M. Alshammari, R. Guesmi, T. Guesmi, H. Alsaif, and A. Alzamil, ''Implementing a symmetric lightweight cryptosystem in highly constrained IoT devices by using a chaotic s-box,'' *Symmetry*, vol. 13, no. 1, p. 129, Jan. 2021, doi: 10.3390/sym13010129.

[59] M. Irfan, T. Shah, G. F. Siddiqui, A. Rehman, T. Saba, and S. A. Bahaj, ''Design of nonlinear component of block cipher using Gravesian octonion integers,'' *IEEE Access*, vol. 11, pp. 2138–2147, 2023, doi: 10.1109/ACCESS.2022.3217211.

[60] D. Lambic, ''A novel method of s-box design based on discrete chaotic map,'' *Nonlinear Dyn.*, vol. 87, no. 4, pp. 2407–2413, Mar. 2017, doi: 10.1007/s11071-016-3199-x.

[61] F. Özkaynak, ''Construction of robust substitution boxes based on chaotic systems,'' *Neural Comput. Appl.*, vol. 31, no. 8, pp. 3317–3326, Aug. 2019, doi: 10.1007/s00521-017-3287-y.

[62] T. Ye and L. Zhimao, ''Chaotic s-box: Six-dimensional fractional Lorenz–Duffing chaotic system and O-shaped path scrambling,'' *Nonlinear Dyn.*, vol. 94, no. 3, pp. 2115–2126, Nov. 2018, doi: 10.1007/s11071-018-4478-5.

[63] E. Tanyildizi and F. Özkaynak, ''A new chaotic s-box generation method using parameter optimization of one dimensional chaotic maps,'' *IEEE Access*, vol. 7, pp. 117829–117838, 2019, doi: 10.1109/ACCESS.2019.2936447.

[64] T. Farah, R. Rhouma, and S. Belghith, ''A novel method for designing s-box based on chaotic map and teaching–learning-based optimization,'' *Nonlinear Dyn.*, vol. 88, no. 2, pp. 1059–1074, Apr. 2017, doi: 10.1007/s11071-016-3295-y.

[65] D. Lambic, ''s-box design method based on improved one-dimensional discrete chaotic map,'' *J. Inf. Telecommun.*, vol. 2, no. 2, pp. 181–191, Apr. 2018, doi: 10.1080/24751839.2018.1434723.

[66] W. Gao, B. Idrees, S. Zafar, and T. Rashid, ''Construction of nonlinear component of block cipher by action of modular group PSL(2, Z) on projective line PL(GF(28)),'' *IEEE Access*, vol. 8, pp. 136736–136749, 2020, doi: 10.1109/ACCESS.2020.3010615.

[67] Ü. Çavusoglu, A. Zengin, I. Pehlivan, and S. Kaçar, ''A novel approach for strong s-box generation algorithm design based on chaotic scaled Zhongtang system,'' *Nonlinear Dyn.*, vol. 87, no. 2, pp. 1081–1094, Jan. 2017, doi: 10.1007/s11071-016-3099-0.

[68] D. Lambic, ''A new discrete-space chaotic map based on the multiplication of integer numbers and its application in s-box design,'' *Nonlinear Dyn.*, vol. 100, no. 1, pp. 699–711, Mar. 2020, doi: 10.1007/s11071-020-05503-y.

[69] Z. Zhu, Y. Song, W. Zhang, H. Yu, and Y. Zhao, ''A novel compressive sensing-based framework for image compression-encryption with s-box,'' *Multimedia Tools Appl.*, vol. 79, nos. 35–36, pp. 25497–25533, Sep. 2020, doi: 10.1007/s11042-020-09193-x.

[70] F. ul Islam and G. Liu, ''Designing S-box based on 4D-4Wing hyperchaotic system,'' *3D Res.*, vol. 8, no. 1, p. 9, Mar. 2017, doi: 10.1007/s13319-017-0119-x.

[71] X. Wang, A. Akgul, U. Cavusoglu, V.-T. Pham, D. V. Hoang, and X. Nguyen, ''A chaotic system with infinite equilibria and its s-box constructing application,'' *Appl. Sci.*, vol. 8, no. 11, p. 2132, Nov. 2018, doi: 10.3390/app8112132.

[72] Ü. Çavusoglu, S. Kaçar, A. Zengin, and I. Pehlivan, ''A novel hybrid encryption algorithm based on chaos and S-AES algorithm,'' *Nonlinear Dyn.*, vol. 92, no. 4, pp. 1745–1759, Jun. 2018, doi: 10.1007/s11071-018-4159-4.

[73] S. Ibrahim and A. Alharbi, ''Efficient image encryption scheme using Henon map, dynamic s-boxes and elliptic curve cryptography,'' *IEEE Access*, vol. 8, pp. 194289–194302, 2020, doi: 10.1109/ACCESS.2020.3032403.

[74] M. S. Açikkapi and F. Özkaynak, ''A method to determine the most suitable initial conditions of chaotic map in statistical randomness applications,'' *IEEE Access*, vol. 9, pp. 1482–1494, 2021, doi: 10.1109/ACCESS.2020.3046470.

[75] G. Liu, W. Yang, W. Liu, and Y. Dai, ''Designing s-boxes based on 3-D four-wing autonomous chaotic system,'' *Nonlinear Dyn.*, vol. 82, no. 4, pp. 1867–1877, Dec. 2015, doi: 10.1007/s11071-015-2283-y.

[76] I. Hussain, T. Shah, H. Mahmood, and M. A. Gondal, ''A projective general linear group based algorithm for the construction of substitution box for block ciphers,'' *Neural Comput. Appl.*, vol. 22, no. 6, pp. 1085–1093, May 2013, doi: 10.1007/s00521-012-0870-0.

[77] M. Khan and T. Shah, ''A novel image encryption technique based on Hénon chaotic map and S_8 symmetric group,'' *Neural Comput. Appl.*, vol. 25, nos. 7–8, pp. 1717–1722, Dec. 2014, doi: 10.1007/s00521-014-1663-4.

[78] A. Belazi and A. A. A. El-Latif, ''A simple yet efficient s-box method based on chaotic sine map,'' *Optik*, vol. 130, pp. 1438–1444, Feb. 2017, doi: 10.1016/j.ijleo.2016.11.152.

[79] A. Belazi, M. Khan, A. A. A. El-Latif, and S. Belghith, ''Efficient cryptosystem approaches: s-boxes and permutation–substitution-based encryption,'' *Nonlinear Dyn.*, vol. 87, no. 1, pp. 337–361, Jan. 2017, doi: 10.1007/s11071-016-3046-0.

[80] K. K. Butt, G. Li, F. Masood, and S. Khan, ''A digital image confidentiality scheme based on pseudo-quantum chaos and Lucas sequence,'' *Entropy*, vol. 22, no. 11, p. 1276, 2020, doi: 10.3390/e22111276.

[81] F. Özkaynak, ''On the effect of chaotic system in performance characteristics of chaos based s-box designs,'' *Phys. A, Stat. Mech. Appl.*, vol. 550, Jul. 2020, Art. no. 124072, doi: 10.1016/j.physa.2019.124072.

[82] I. Hussain, T. Shah, M. A. Gondal, and H. Mahmood, ''A novel method for designing nonlinear component for block cipher based on TD-ERCS chaotic sequence,'' *Nonlinear Dyn.*, vol. 73, nos. 1–2, pp. 633–637, Jul. 2013, doi: 10.1007/s11071-013-0816-9.

[83] F. Özkaynak, ''From biometric data to cryptographic primitives: A new method for generation of substitution boxes,'' in *Proc. ACM Int. Conf. Ser.*, 2017, pp. 27–33, doi: 10.1145/3143344.3143355.

[84] F. Artuger and F. Özkaynak, ''A novel method for performance improvement of chaos-based substitution boxes,'' *Symmetry*, vol. 12, no. 4, p. 571, Apr. 2020, doi: 10.3390/SYM12040571.

[85] I. Hussain, T. Shah, H. Mahmood, and M. A. Gondal, ''Construction of S_8 Liu J s-boxes and their applications,'' *Comput. Math. Appl.*, vol. 64, no. 8, pp. 2450–2458, Oct. 2012, doi: 10.1016/j.camwa.2012.05.017.

[86] M. Khan and T. Shah, ''An efficient construction of substitution box with fractional chaotic system,'' *Signal, Image Video Process.*, vol. 9, no. 6, pp. 1335–1338, Sep. 2015, doi: 10.1007/s11760-013-0577-4.

[87] I. Hussain, T. Shah, and M. A. Gondal, ''A novel approach for designing substitution-boxes based on nonlinear chaotic algorithm,'' *Nonlinear Dyn.*, vol. 70, no. 3, pp. 1791–1794, Nov. 2012, doi: 10.1007/s11071-012-0573-1.

[88] I. Hussain, T. Shah, M. A. Gondal, W. A. Khan, and H. Mahmood, ''A group theoretic approach to construct cryptographically strong substitution boxes,'' *Neural Comput. Appl.*, vol. 23, no. 1, pp. 97–104, Jul. 2013, doi: 10.1007/s00521-012-0914-5.

[89] F. Özkaynak, V. Çelik, and A. B. Özer, ''A new s-box construction method based on the fractional-order chaotic Chen system,'' *Signal, Image Video Process.*, vol. 11, no. 4, pp. 659–664, May 2017, doi: 10.1007/s11760-016-1007-1.

[90] F. Özkaynak, ''Chaotic modeling and simulation (CMSIM) chaos based substitution boxes as a cryptographic primitives: Challenges and opportunities,'' in *Proc. Chaotic Model. Simul.*, 2019, pp. 49–57.

[91] L. Liu, Y. Zhang, and X. Wang, ''A novel method for constructing the s-box based on spatiotemporal chaotic dynamics,'' *Appl. Sci.*, vol. 8, no. 12, p. 2650, Dec. 2018, doi: 10.3390/app8122650.

[92] A. Mahboob, M. Asif, M. Nadeem, A. Saleem, S. M. Eldin, and I. Siddique, ''A cryptographic scheme for construction of substitution boxes using quantic fractional transformation,'' *IEEE Access*, vol. 10, pp. 132908–132916, 2022, doi: 10.1109/ACCESS.2022.3230141.

[93] G. Chen, ''A novel heuristic method for obtaining s-boxes,'' *Chaos, Solitons Fractals*, vol. 36, no. 4, pp. 1028–1036, May 2008, doi: 10.1016/j.chaos.2006.08.003.

[94] M. Khan, T. Shah, and M. A. Gondal, ''An efficient technique for the construction of substitution box with chaotic partial differential equation,'' *Nonlinear Dyn.*, vol. 73, no. 3, pp. 1795–1801, Aug. 2013, doi: 10.1007/s11071-013-0904-x.

[95] M. Khan and T. Shah, ''A construction of novel chaos base nonlinear component of block cipher,'' *Nonlinear Dyn.*, vol. 76, no. 1, pp. 377–382, Apr. 2014, doi: 10.1007/s11071-013-1132-0.

[96] H. Liu, A. Kadir, and Y. Niu, ''Chaos-based color image block encryption scheme using s-box,'' *AEU-Int. J. Electron. Commun.*, vol. 68, no. 7, pp. 676–686, Jul. 2014, doi: 10.1016/j.aeue.2014.02.002.

[97] M. Khan, T. Shah, and S. I. Batool, ''A new implementation of chaotic s-boxes in CAPTCHA,'' *Signal, Image Video Process.*, vol. 10, no. 2, pp. 293–300, Feb. 2016, doi: 10.1007/s11760-014-0741-5.

[98] X. Wang and J. Yang, ''A novel image encryption scheme of dynamic s-boxes and random blocks based on spatiotemporal chaotic system,'' *Optik*, vol. 217, Sep. 2020, Art. no. 164884, doi: 10.1016/j.ijleo.2020.164884.

[99] N. A. Khan, M. Altaf, and F. A. Khan, ''Selective encryption of JPEG images with chaotic based novel s-box,'' *Multimedia Tools Appl.*, vol. 80, no. 6, pp. 9639–9656, Mar. 2021, doi: 10.1007/s11042-020-10110-5.

[100] G. Tang and X. Liao, ''A method for designing dynamical s-boxes based on discretized chaotic map,'' *Chaos, Solitons Fractals*, vol. 23, no. 5, pp. 1901–1909, 2005, doi: 10.1016/j.chaos.2004.07.033.

[101] F. Özkaynak and S. Yavuz, ''Designing chaotic s-boxes based on time-delay chaotic system,'' *Nonlinear Dyn.*, vol. 74, no. 3, pp. 551–557, Nov. 2013, doi: 10.1007/s11071-013-0987-4.

[102] G. Tang, X. Liao, and Y. Chen, ''A novel method for designing s-boxes based on chaotic maps,'' *Chaos, Solitons Fractals*, vol. 23, no. 2, pp. 413–419, Jan. 2005, doi: 10.1016/j.chaos.2004.04.023.

[103] N. Hematpour and S. Ahadpour, ''Execution examination of chaotic s-box dependent on improved PSO algorithm,'' *Neural Comput. Appl.*, vol. 33, no. 10, pp. 5111–5133, May 2021, doi: 10.1007/s00521-020-05304-9.

[104] F. Özkaynak and A. B. Özer, "A method for designing strong s-boxes based on chaotic Lorenz system," *Phys. Lett. A*, vol. 374, no. 36, pp. 3733–3738, Aug. 2010, doi: 10.1016/j.physleta.2010.07.019.

[105] G. Jakimoski and L. Kocarev, "Chaos and cryptography: Block encryption ciphers based on chaotic maps," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 48, no. 2, pp. 163–169, Feb. 2001, doi: 10.1109/81.904880.

[106] G. Chen, Y. Chen, and X. Liao, "An extended method for obtaining s-boxes based on three-dimensional chaotic baker maps," *Chaos, Solitons Fractals*, vol. 31, no. 3, pp. 571–579, Feb. 2007, doi: 10.1016/j.chaos.2005.10.022.

[107] M. Khan, T. Shah, H. Mahmood, M. A. Gondal, and I. Hussain, "A novel technique for the construction of strong s-boxes based on chaotic Lorenz systems," *Nonlinear Dyn.*, vol. 70, no. 3, pp. 2303–2311, Nov. 2012, doi: 10.1007/s11071-012-0621-x.

[108] M. Khan, T. Shah, H. Mahmood, and M. A. Gondal, "An efficient method for the construction of block cipher with multi-chaotic systems," *Nonlinear Dyn.*, vol. 71, no. 3, pp. 489–492, Feb. 2013, doi: 10.1007/s11071-012-0675-9.

[109] A. Freyre-Echevarría, I. Martínez-Díaz, C. M. L. Pérez, G. Sosa-Gómez, and O. Rojas, "Evolving nonlinear s-boxes with improved theoretical resilience to power attacks," *IEEE Access*, vol. 8, pp. 202728–202737, 2020, doi: 10.1109/ACCESS.2020.3035163.

[110] M. Khan and Z. Asghar, "A novel construction of substitution box for image encryption applications with gingerbreadman chaotic map and S_8 permutation," *Neural Comput. Appl.*, vol. 29, no. 4, pp. 993–999, Feb. 2018, doi: 10.1007/s00521-016-2511-5.

[111] S. S. Jamal, M. U. Khan, and T. Shah, "A watermarking technique with chaotic fractional s-box transformation," *Wireless Pers. Commun.*, vol. 90, no. 4, pp. 2033–2049, Oct. 2016, doi: 10.1007/s11277-016-3436-0.

[112] B. B. Cassal-Quiroga and E. Campos-Cantón, "Generation of dynamical s-boxes for block ciphers via extended logistic map," *Math. Problems Eng.*, vol. 2020, pp. 1–12, Mar. 2020, doi: 10.1155/2020/2702653.

[113] M. Khan, "A novel image encryption scheme based on multiple chaotic s-boxes," *Nonlinear Dyn.*, vol. 82, nos. 1–2, pp. 527–533, Oct. 2015, doi: 10.1007/s11071-015-2173-3.

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

[115] B. Rashidi, "Compact and efficient structure of 8-bit s-box for lightweight cryptography," *Integration*, vol. 76, pp. 172–182, Jan. 2021, doi: 10.1016/j.vlsi.2020.10.009.

[116] A. Y. Al-Dweik, I. Hussain, M. Saleh, and M. T. Mustafa, "A novel method to generate key-dependent s-boxes with identical algebraic properties," *J. Inf. Secur. Appl.*, vol. 64, Feb. 2022, Art. no. 103065, doi: 10.1016/j.jisa.2021.103065.

[117] J. Wang, G. Liu, Y. Chen, and S. Wang, "Construction and analysis of SHA-256 compression function based on chaos s-box," *IEEE Access*, vol. 9, pp. 61768–61777, 2021, doi: 10.1109/ACCESS.2021.3071501.

[118] A. W. Malik, A. H. Zahid, D. S. Bhatti, H. J. Kim, and K.-I. Kim, "Designing s-box using tent-sine chaotic system while combining the traits of tent and sine map," *IEEE Access*, vol. 11, pp. 79265–79274, 2023, doi: 10.1109/ACCESS.2023.3298111.

[119] M. Long and L. Wang, "s-box design based on discrete chaotic map and improved artificial bee colony algorithm," *IEEE Access*, vol. 9, pp. 86144–86154, 2021, doi: 10.1109/ACCESS.2021.3069965.

[120] A. Ari and F. Özkaynak, "Generation of substitution box structures based on blum blum shub random number outputs," in *Proc. IEEE 16th Int. Conf. Adv. Trends Radioelectronics, Telecommun. Comput. Eng. (TCSET)*, Feb. 2022, pp. 677–682, doi: 10.1109/TCSET55632.2022.9766861.

[121] M. Ahmad, E. Al-Solami, A. M. Alghamdi, and M. A. Yousaf, "Bijective s-boxes method using improved chaotic map-based heuristic search and algebraic group structures," *IEEE Access*, vol. 8, pp. 110397–110411, 2020, doi: 10.1109/ACCESS.2020.3001868.

[122] S. S. Jamal, A. Anees, M. Ahmad, M. F. Khan, and I. Hussain, "Construction of cryptographic s-boxes based on Mobius transformation and chaotic tent-sine system," *IEEE Access*, vol. 7, pp. 173273–173285, 2019, doi: 10.1109/ACCESS.2019.2956385.

[123] R. Soto, B. Crawford, F. G. Molina, and R. Olivares, "Human behaviour based optimization supported with self-organizing maps for solving the s-box design problem," *IEEE Access*, vol. 9, pp. 84605–84618, 2021, doi: 10.1109/ACCESS.2021.3087139.

[124] A. Alhudhaif, M. Ahmad, A. Alkhayyat, N. Tsafack, A. K. Farhan, and R. Ahmed, "Block cipher nonlinear confusion components based on new 5-D hyperchaotic system," *IEEE Access*, vol. 9, pp. 87686–87696, 2021, doi: 10.1109/ACCESS.2021.3090163.

[125] A. I. Lawah, A. A. Ibrahim, S. Q. Salih, H. S. Alhadawi, and P. S. JosephNg, "Grey wolf optimizer and discrete chaotic map for substitution boxes design and optimization," *IEEE Access*, vol. 11, pp. 42416–42430, 2023, doi: 10.1109/ACCESS.2023.3266290.

[126] S. Zhu, X. Deng, W. Zhang, and C. Zhu, "Secure image encryption scheme based on a new robust chaotic map and strong s-box," *Math. Comput. Simul.*, vol. 207, pp. 322–346, May 2023, doi: 10.1016/j.matcom.2022.12.025.

[127] S. Ullah, X. Liu, A. Waheed, and S. Zhang, "An efficient construction of s-box based on the fractional-order Rabinovich–Fabrikant chaotic system," *Integration*, vol. 94, Jan. 2024, Art. no. 102099, doi: 10.1016/J.VLSI.2023.102099.

[128] S. S. Jamal, M. M. Hazzazi, M. F. Khan, Z. Bassfar, A. Aljaedi, and Z. U. Islam, "Region of interest-based medical image encryption technique based on chaotic s-boxes," *Expert Syst. Appl.*, vol. 238, Mar. 2024, Art. no. 122030, doi: 10.1016/J.ESWA.2023.122030.

[129] Y. Qobbi, A. Abid, M. Jarjar, S. E. Kaddouhi, A. Jarjar, and A. Benazzi, "Adaptation of a genetic operator and a dynamic s-box for chaotic encryption of medical and color images," *Sci. Afr.*, vol. 19, Mar. 2023, Art. no. e01551, doi: 10.1016/j.sciaf.2023.e01551.

[130] M. Zhao, H. Liu, and Y. Niu, "Batch generating keyed strong s-boxes with high nonlinearity using 2D hyper chaotic map," *Integration*, vol. 92, pp. 91–98, Sep. 2023, doi: 10.1016/j.vlsi.2023.05.006.

[131] A. Razaq, G. Alhamzi, S. Abbas, M. Ahmad, and A. Razzaque, "Secure communication through reliable s-box design: A proposed approach using coset graphs and matrix operations," *Heliyon*, vol. 9, no. 5, May 2023, Art. no. e15902, doi: 10.1016/j.heliyon.2023.e15902.

[132] M. A. M. Khan, N. A. Azam, U. Hayat, and H. Kamarulhaili, "A novel deterministic substitution box generator over elliptic curves for real-time applications," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 35, no. 1, pp. 219–236, Jan. 2023, doi: 10.1016/j.jksuci.2022.11.012.

[133] Y. Si, H. Liu, and M. Zhao, "Constructing keyed strong s-box with higher nonlinearity based on 2D hyper chaotic map and algebraic operation," *Integration*, vol. 88, pp. 269–277, Jan. 2023, doi: 10.1016/j.vlsi.2022.10.011.

[134] K. Z. Zamli, "Exploiting an elitist barnacles mating optimizer implementation for substitution box optimization," *ICT Exp.*, vol. 9, no. 4, pp. 619–627, 2023, doi: 10.1016/j.icte.2022.11.005.

[135] M. I. Haider, T. Shah, A. Ali, D. Shah, and I. Khalid, "An innovative approach towards image encryption by using novel PRNs and s-boxes modeling techniques," *Math. Comput. Simul.*, vol. 209, pp. 153–168, Jul. 2023, doi: 10.1016/j.matcom.2023.01.036.

[136] S. Deb and P. K. Behera, "Design of key-dependent bijective s-boxes for color image cryptosystem," *Optik*, vol. 253, Mar. 2022, Art. no. 168548, doi: 10.1016/j.ijleo.2021.168548.

[137] S. Ibrahim and A. M. Abbas, "Efficient key-dependent dynamic s-boxes based on permutated elliptic curves," *Inf. Sci.*, vol. 558, pp. 246–264, May 2021, doi: 10.1016/j.ins.2021.01.014.

[138] M. D. Gupta and R. K. Chauhan, "Secure image encryption scheme using 4D-Hyperchaotic systems based reconfigurable pseudo-random number generator and s-box," *Integration*, vol. 81, pp. 137–159, Nov. 2021, doi: 10.1016/j.vlsi.2021.07.002.

[139] K. Z. Zamli, "Optimizing s-box generation based on the adaptive agent heroes and cowards algorithm," *Expert Syst. Appl.*, vol. 182, Nov. 2021, Art. no. 115305, doi: 10.1016/j.eswa.2021.115305.

[140] N. A. Azam, U. Hayat, and M. Ayub, "A substitution box generator, its analysis, and applications in image encryption," *Signal Process.*, vol. 187, Oct. 2021, Art. no. 108144, doi: 10.1016/j.sigpro.2021.108144.

[141] F. Artuger and F. Özkaynak, "A new post-processing approach for improvement of nonlinearity property in substitution boxes," *Integration*, vol. 94, Jan. 2024, Art. no. 102105, doi: 10.1016/j.vlsi.2023.102105.
[142] F. Artuger and F. Özkaynak, "A new algorithm to generate AES-like substitution boxes based on sine cosine optimization algorithm," *Multimedia Tools Appl.*, vol. 2023, pp. 1–15, Oct. 2023, doi: 10.1007/s11042-023-17200-0.
[143] F. Artuger and F. Özkaynak, "SBOX-CGA: Substitution box generator based on chaos and genetic algorithm," *Neural Comput. Appl.*, vol. 34, no. 22, pp. 20203–20211, Nov. 2022, doi: 10.1007/s00521-022-07589-4.
[144] F. Artuger, "A novel algorithm based on DNA coding for substitution box generation problem," *Neural Comput. Appl.*, vol. 2023, pp. 1–12, Oct. 2023, doi: 10.1007/s00521-023-09095-7.
[145] F. Artuger, "A new s-box generator algorithm based on 3D chaotic maps and whale optimization algorithm," *Wireless Pers. Commun.*, vol. 131, no. 2, pp. 835–853, Jul. 2023, doi: 10.1007/s11277-023-10456-7.
[146] L. A. Demidova and A. V. Gorchakov, "A study of chaotic maps producing symmetric distributions in the fish school search optimization algorithm with exponential step decay," *Symmetry*, vol. 12, no. 5, p. 784, May 2020, doi: 10.3390/SYM12050784.

YıLMAZ AYDıN received the master's degree from the Department of Software Engineering. His career in chaos-based cryptography began during the master's degree, where he completed his thesis titled "Improvement of new security criteria and analysis methods using the relationship between chaos and cryptography," in 2019. He continued this path in the Ph.D. degree, focusing on the secure transmission, processing, and storage of medical data. During his doctoral research, he aims to apply the research and development outputs practically to generate societal and economic gains. He is currently with the Department of Software Engineering, Fırat University.

FATIH ÖZKAYNAK received the Bachelor of Science and Master of Science degrees in computer engineering from Fırat University, in 2005 and 2007, respectively, and the Doctor of Philosophy degree in computer engineering from Yıldız Technical University, in 2013.

He is currently a member of Software Engineering with Fırat University. His primary research interests include cryptography, information security, and chaotic systems. He has coauthored over 100 peer-reviewed papers, both in scientific journals and conferences, contributing significantly to the field. His works have been cited more than 2000 times, reflecting their impact and relevance in the academic community.