Al-Noor Journal for Information Technology and Cyber Security
December (2024); 1 (0): 77- 97
DOI: https://doi.org/10.69513/jnfit.v1.i0.a5

Efficient Substitution Box Design Using Modified Intelligent Jellyfish Search Algorithm

Hind Abdulghani Ahmed Al-Heayli
Sufyan Salim Mahmood Aldabbagh

University of Mosul/ Computer science department
University of Mosul/ Computer science department
Mosul, Iraq, Email: Hind.csp70@student.uomosul.edu.iq
Mosul, Iraq, Email: sufyansalim_77@yahoo.com

Received: 15 / 10/ 2024 Accepted: 20 / 11/ 2024
Published: 17 / 12 / 2024

Abstract

A substitution box is designed as a confusion component to give the modern cipher strength against differential cryptanalysis, which makes the S-Box a vital and only nonlinear part of the most modern cipher algorithm. Encryption methods that are based on chaos systems are very popular because they display a similar property to cryptography. However, most of the recently designed S-Boxes are focusing on some criteria leaving others, harder to implement, or slower to generate. Therefore, this paper proposes a dynamic design methodology to generate a chaotic S-box by utilizing the Jellyfish Search algorithm after modifying it to fit the purpose and that can be used in modern encryption algorithms.The statistical analysis results of the proposed S-Box are compared to some of the recently designed 4 × 4 S-Boxes in the literature, the comparison showed that the suggested S-Box has mostly equal, to better statistical attributes, which mark the suggested S-boxes robust cryptographically and a good fit to be used for lightweight block cipher algorithms.

I- Introduction

"All modern block and stream ciphers feature one or more non-linear elements," according to Petr Tesar's work "A New Method for Generating High Non-Linearity S-Boxes." "In current ciphers, the S-box is one of the most widely employed non-linear cornerstones." As a result, S-Box is an essential component of many ciphers, and in many situations, it is critical to the overall security of the cipher. An S-box is a lookup table where m bits of input are replaced by n bits of output. Any cipher's purpose is to make the output look as unpredictable as possible while yet being some value we can retrieve later to get the original plaintext. XOR, for example, is a linear operation. S-boxes are nonlinear elements that are utilized in current symmetric ciphers to provide the attribute of confusion and provide cryptanalysis resistance [1,2].

S-boxes are a key component of most block ciphers, and the development of strong S-boxes has gotten a lot of attention in the cryptography community. In fact, the power of the S-boxes to prevent unauthorized access has a significant impact on the security of transmitted data [1]. As a result, building a strong S-box with a high nonlinearity score is seen as a substantial difficulty. Many block ciphers can be seen as integrating a substitution cipher like Affine cipher with a transposition cipher like Zig-Zag and executing it numerous times if they didn't have the s-boxes. In creating block cipher systems, generating S-boxes with a solid cryptographic characteristic is a crucial aspect [3]. This is due to the fact that an S-box is the cipher's only nonlinear component, and its whole cryptographic strength is based on it. Researchers are continually developing new attack methods, so S-box design needs to be always ahead of them to guarantee cipher security [2].

The architectural simplicity, quick encryption and decryption speed, and resilience to known cryptanalysis techniques are all desired qualities of an S-box. Regardless of the kind of constructed S-box, all S-boxes must have specific characteristics to be effective. The effectiveness of an s-box is normally assessed by looking at a number of different criteria to see if it can withstand various attacks; each of these qualities contributes to the s-security. Box's Non-linearity, the Strict Avalanche Criterion (SAC), the Bit Independence Criterion (BIC), Bijective, Differential Approximation Probability, and Linear Approximation Probability are only a few examples [4].

In the current decade, lightweight cryptographic techniques are in high demand for Internet of Things (IoT) applications. Because IoT devices have limited resources, lightweight security mechanisms are the best option for ensuring the security of these kinds of systems. Traditional security methods, such as the Advanced Encryption Standard (AES), are unsuitable for IoT devices due to their computationally demanding mathematical procedures [5]. The resource limits and level of security addressed by the lightweight encryption algorithms and cryptography primitives like S-Box [6] are highlighted by IoT physical security issues.

In this thesis, a new lightweight S-Box was designed using a novel S-Box generation algorithm that consists of a logistic chaotic map and a modified version of the state-of-art Jellyfish search optimizer, to be used to secure the IoT network. The proposed new S-Boxes are analyzed statistically with the major S-Box design criteria. The logistic chaotic map was selected for initializing the Jellyfish locations due to its simplicity in implementation and the complexity of its chaotic outcomes. With some modification and addition to the state-of-art Jellyfish optimization algorithm, it was possible to use it for optimizing the generated S-Box by searching the search space for a suitable strong S-Box that can satisfy the statistical design criteria. The proposed S-Boxes performance is compared to other suggested 4x4 S-Boxes in recent years of study. The rest of this paper is organized as follows. Section 2 covers the contribution and related literature on S-box design that are related to our work. The novel method of designing an S-box using a chaotic map and JS algorithm is proposed and an example of the S-box generated by this method is presented in Section 3. The performance of the example S-box is evaluated and compared with other most recent S-boxes in Section 4. Section 5, presents the recommendations for future research and concludes.

II- State-of-the-art related research

A lot of research addresses the design of a new S-Box for the security of IoT networks. Some of the recent research in the field of S-Box Design and IoT security is discussed in this section.

Lang Li (2022) [7] provided a stringent avalanche criterion-compliant S-box construction approach. It has a diffusion layer and a nonlinear layer. The results reveal that the suggested S-box has better cryptographic properties than other S-boxes. In addition, the authors assessed the overhead of the suggested S-box hardware implementation. The hardware implementation overhead of the S-box was evaluated under the same conditions to objectively evaluate the S-box, and the findings suggest that the new S-box produced achieved a decent balance between cryptographic characteristics and hardware overhead.

M. Long and Longlong Wang (2021) [8] provide an improved artificial bee colony algorithm and a novel S-box design built on a mixed chaotic environment. To begin, the chaotic S-box population is initialized and its quality is improved using an opposition-based optimization approach. The population is then optimized using the dual transposition artificial bee colony algorithm, with the Gaussian swap helping to avoid falling into local optimal by reinforcing the functionality and speeding up collaborative learning speed between all populations during the location updating procedure. Relying on a cuckoo search (CS) algorithm and discrete-space chaotic map, Alhadawi, H.S., et al. (2021) [9] introduced a novel technique for building S-boxes with chosen cryptographic features. In comparison to GA and PSO, the suggested approach has an advantage in terms of efficient randomization and fewer customizable parameters in CS. In addition, this method used a 1-dimensional discrete-space chaotic map with almost infinite keyspace to construct initial S-boxes. Furthermore, chaotic maps have the ability to overcome a typical CS's local optima trapping problem, and They were used to create the initial S-boxes in order to achieve the required quality and to aid the metaheuristic search. In order to identify an S-box configuration that suited the established requirements, the metaheuristic CS was utilized. Alshammari, B.M.; (2021) [10] proposes a lightweight cryptographic algorithm that may be effectively implemented in IoT devices with limited resources. The Advanced Encryption Standard (AES) and a novel chaotic S-box are the fundamental components of the algorithm. The initial stage was to construct a cryptographically secure S-box using chaotic Boolean functions. The permutation and diffusion phases were realized using the Hilbert curve scan pattern and the Lorenz system. R. Soto, et al. (2021) [11] propose a scheme based on a human behavior-based optimization algorithm that is supported by Self-Organizing Maps that have been trained to evaluate the strength of the exploration process in the regions through which the optimization algorithm has passed and make a proper decision on the search process in order to avoid premature convergence and improve the nonlinearity property in order to acquire robust substitution boxes.

Farhan, Alaa, et al. (2020), [12] offered a new design technique for constructing a multi-S-box based on RNA, as well as an S-box generating scheme based on DNA codons, XOR operations, and some mathematical calculations. The initial S-box is based on the secret key, and each S-box is derived from the preceding one. The mRNA builds a new S-box and the S-box inverse in collaboration with a secure equation. S-box for AES was constructed by Hussain, I, et al (2019), in [2] using a chaotic logistic map, Mobius transformation, and symmetric group S256. The proposed project's goal is to create a safe S-box as a backup.

To produce the initial S-box, Alhadawi, Hussam, et al., (2018), [13] suggested an S-box design approach that relies on firefly optimization and chaotic mapping. The guided search for near-optimal properties with minimizing fitness function is used to locate the arrangement of the S-box that fits the criteria using the firefly optimization. Markus Ullrich (2018) [14] describes a method for locating efficient bit-sliced representations of invertible 4-bit s-boxes. The method is a generalization of Osvik's approaches. The authors attempt to cover the entire spectrum of 4 × 4-bit S-boxes by classifying them into linear and affine equivalence classes and searching for the most optimal S-box in each class. In terms of linear and differential cryptanalysis, the classification criterion chosen gives invaria\Dnt features. T. Ara, et al (2018) [5] devise a simple, practical, and novel method for producing key-dependent S-boxes for Symmetric Encryption To construct dynamic S-boxes, the proposed approach uses Elliptic Curve Cryptography. The Multi-precision Integer and Rational Arithmetic Cryptographic Library (MIRACL) is used to implement it in C++, and the experimental findings demonstrate that it takes relatively little time to compute. The proposed approach can be used to produce dynamic key-dependent S-Boxes in the PRESENT cipher. Liu, Liyan, et al. (2018) [15] propose a new technique for constructing a random S-box that is based on a spatiotemporal nonlinear chaotic system. The chaotic sequence of the Non-adjacent Coupled Map Lattices (NCML) spatiotemporal chaotic system is utilized to create an initial S-box. The permutation process between independent chaotic sequences is then used to implement the permutation and shuffle the elements of the S-box randomly, which could also enhance its BIC attribute and capacity to withstand linear password attacks. Eesa Al Solami et al. (2018) [16] present a novel approach for generating cryptographically robust S-boxes based on the complex dynamics of a high-dimensional hyperchaotic system. The novel hyperchaotic system was found to have beneficial features when compared to existing schemes applied for S-box manufacturing.

The traveling salesman issue and piece-wise linear chaotic map are addressed by Musheer Ahmad (2016) in [17] to synthesize an effective configuration substitution box. The suggested projected design is consistent, as evidenced by conventional performance indices. Y. Tian and Lu. Zhimao (2016) [18] devised a chaotic S-box centered on the artificial bee colony algorithm (CSABC). It starts with the S-boxes produced by the six-dimensional compound hyperchaotic map and then utilizes ABC to improve their performance by looking for S-boxes that perform well. Nonlinearity and differential uniformity are also thought of as fitness functions.

Mihajloska, H, et al (2012) [19] propose a novel iterative methodology for the production of cryptographically robust S-boxes utilizing quasigroups of order 4 and the principle of quasigroup string transformations that is small, fast, and elegant. Quasigroup string modifications are used to create 4 × 4-bit Quasigroup-S-boxes (Q-S-boxes). This methodology allows someone to iteratively work with numerous distinct strong S-boxes while just altering a few parameters and retaining only one hardware circuit.

III. Proposed Method

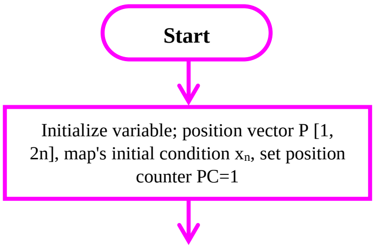
In this section, we present the proposed novel approach to the synthesis of a cryptographically robust S-box using a modified version of the artificial Jellyfish Search (JS) algorithm [20] for the search and the logistic chaotic map for initializing the jellyfish initial locations, we describe everything in detail and give the full proposed algorithm.

A. Initialize S-Box Generation

A one-dimensional vector of size (n*n) is used to represent the Jellyfish position/state (S-Box). Initializing the S-Box can be done in many ways, randomly, mathematically, …etc. The disadvantages of random initializing are its slow convergence and its tendency to become trapped at local optima as a result of low population diversity. While mathematical initializing of the S-box causes overhead on the cryptosystem and may complicate and delay the process. To overcome these issues, and to improve the diversity of the initial population in order to lower the probability of premature convergence, chaotic mapping was used to initialize the population. However, out of the set of well-known chaotic maps, including the logistic map, tent map, Liebovitch map, …etc, the logistic map was selected as it is one of the simplest chaotic maps, and it can provide more diverse initial populations than does random selection and by that provides a lower probability of premature convergence.

Utilizing the properties of chaos to generate the Jellyfish positions (S-Boxes). A chaotic logistic map is iterated under given initial conditions. In order to have initial S-boxes with good cryptographic properties, we applied the algorithm used in [21] with some modifications to fill the S-box with values based on the logistic chaotic map, while preventing any repetition or missing values in the S-Box. The detailed flow diagram of S-box position generation steps is described in Figure 1.

1. To generate the Jellyfish positions of the initial populations S-boxes, variables are initialized: Position counter PC counts the generated positions and map's initial parameters xn.
2. The domain in the range [0.1, 0.9] is divided into 2n equal intervals.
3. These intervals are then labeled sequentially in the range [1, 2n] as position number PN.
4. Initialize a position vector P of size 2n.


Start
Initialize variable; position vector P [1, 2n], map's initial condition x_n, set position counter PC=1

DOI: https://doi.org/10.69513/jnfit.v1.i0.a5

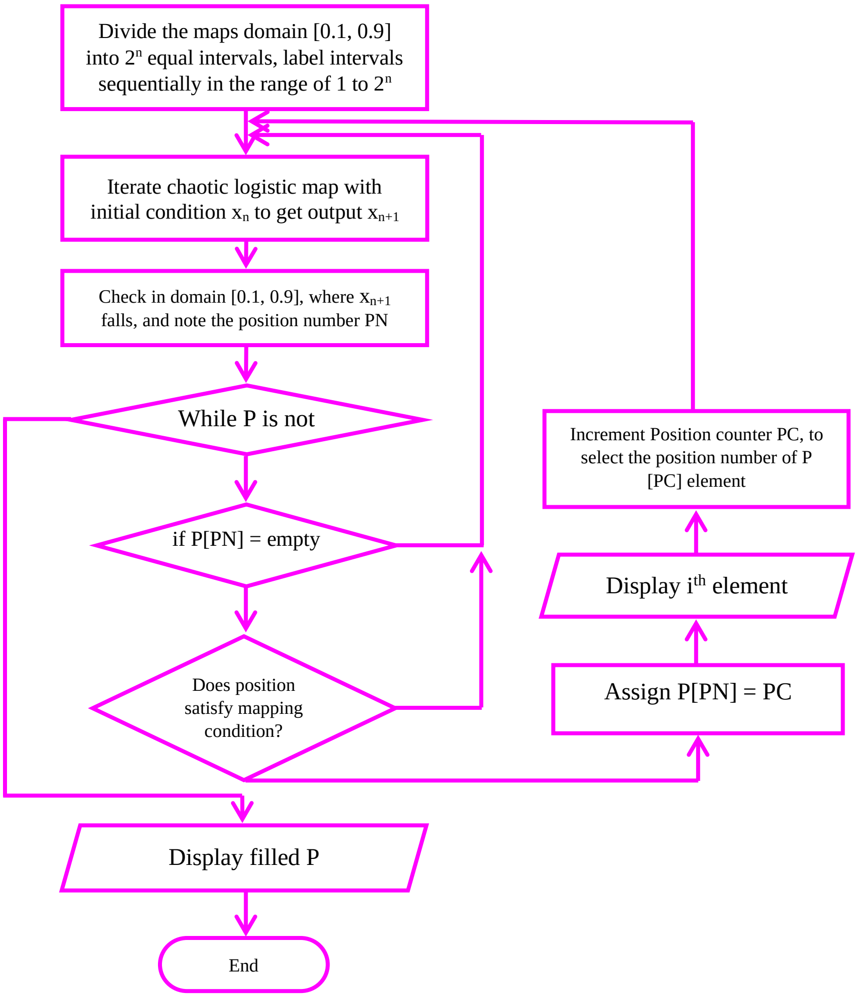

Divide the maps domain [0.1, 0.9] into 2^n equal intervals, label intervals sequentially in the range of 1 to 2^n
Iterate chaotic logistic map with initial condition x_n to get output x_{n+1}
Check in domain [0.1, 0.9], where x_{n+1} falls, and note the position number PN
While P is not
if P[PN] = empty
Does position satisfy mapping condition?
Increment Position counter PC, to select the position number of P [PC] element
Display i^{th} element
Assign P[PN] = PC
Display filled P
End

Figure 1:S-Box Initialization Flow Chart

5. The logistic map is iterated with an arbitrarily chosen initial condition xn. By iterating the chaotic logistic map with a unique initial value of x0, one can generate a unique sequence of random real numbers whose values lie between 0 and 1. The chaotic logistic map used with r = 4 is the only use case because the chaotic attractor is distributed uniformly in the chaotic domain region, which spans over [0, 1].

6. The chaotic logistic map is iterated under given initial conditions.
7. The output of logistic map xn1 is checked in the domain where it falls. The specific subdomain is marked accordingly, and the corresponding subdomain number is stored in a row vector, which is called position vector P.
8. During the course of iteration, if the map's output falls in a visited subdomain, then this subdomain is ignored.
9. If the position falls in an empty subdomain, the position is assigned to that subdomain.
10. The position is tested using improvement criteria that exploit DDT. If it fulfills the proposed criteria, the position is fixed in P; otherwise, regenerate this position and empty that subdomain.
11. Stop iterating chaotic logistic maps once all positions in the position vector are filled.

B. Modified Jellyfish Search Algorithm

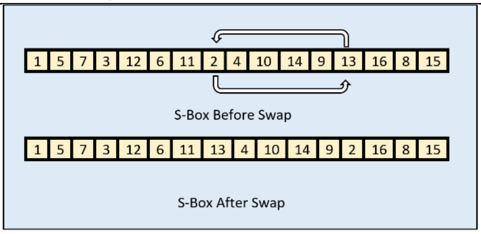
The artificial Jellyfish search algorithm is a metaphor-based metaheuristics algorithm inspired by the behavior of jellyfish in the ocean. The simulation of the search behavior of jellyfish involves their following the ocean current, their motions inside a jellyfish swarm (active motions and passive motions), a time control mechanism for switching among these movements, and their convergences into jellyfish bloom. JS algorithm is tested successfully on benchmark functions and optimization problems, in order to use JS for more generalized optimization problems outside of the scope of mathematical optimization, a modification is needed.

The proposed modification to the JS algorithm can be represented in two main changes that are necessary to make the algorithm suitable for different types of general space search optimization problems.

The Objective function needs to be replaced with a fitness function that varies based on the problem that the algorithm intended to optimize. The fitness function needs to be designed in such a way that its values get higher as the fitness gets better (higher is better), for that it can also be called the benefit function. For our algorithm, the objective function is removed and replaced with the nonlinearity function of the S-Box, as is the case in [,]. The higher the non-linearity of the S-box the higher the chance of achieving good statistical attributes in the S-Box.

The equations that represent the motion of the Jellyfish are designed to work on an objective function, and they can’t be used with search spaces his elements are of multiple varied shapes and values. So, the motion needs to be represented in some other way. To represent the motion of the jellyfish efficiently and effectively in our algorithm, the motion equations are proposed to be replaced with a Crossover / Swap mechanism that works differently based on the current type of movement the Jellyfish is experiencing.

The crossover/swap operation for the proposed algorithm is based on swapping two values in the S-Box, either randomly, or based on the values of another S-box depending on the type of Jellyfish movement. In the case of Passive movement, a swap operation takes place between two randomly selected values inside the S-Box that is represented by the Jellyfish position in the search place, Figure 2 shows the swap operation that occurs when the type of movement is passive motion.



1	5	7	3	12	6	11	2	4	10	14	9	13	16	8	15

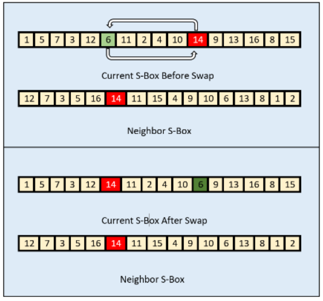
S-Box Before Swap

1	5	7	3	12	6	11	13	4	10	14	9	2	16	8	15

S-Box After Swap

Figure 2: Passive motion swap operation

On the other hand, if the movement is active motion or the Jellyfish moves with the ocean current, there would be two S-Boxes (Jellyfish positions) involved in the operation. Active motion is based on a randomly selected Jellyfish neighbor, a comparison takes place between that Jellyfish location and the current Jellyfish location based on the fitness function of each. In case the neighbor Jellyfish is the higher fitness one, a randomly selected index value inside the neighbor S-Box is selected, and the current S-Box starts the swap operation by searching the value of the selected index inside of his vector and swapping it with the value currently at that index within the same victor. The point of this swap is to move the current Jellyfish toward the neighbor with a better fitness value by creating similarities between the two, Figure 3 shows the swap operation that represents active motion toward the neighbor.



1	5	7	3	12	6	11	2	4	10	14	9	13	16	8	15

Current S-Box Before Swap

12	7	3	5	16	14	11	15	4	9	10	6	13	8	1	2

Neighbor S-Box

1	5	7	3	12	14	11	2	4	10	6	9	13	16	8	15

Current S-Box After Swap

12	7	3	5	16	14	11	15	4	9	10	6	13	8	1	2

Neighbor S-Box

Figure 3: Active motion toward neighbor swap operation

When the current Jellyfish has the higher fitness value, a search process begins to find a value inside the current S-Box that matches the neighbor S-box and in the same index, if found, it gets swapped with a randomly selected index in the same vector to simulate the movement for the Jellyfish away from the neighbor. It is worth mentioning, that in case no similarity was found in the two S-Boxes, no swap operation is needed, and the Jellyfish will keep her position for the current iteration. Figure 4 illustrates the swap operation that simulates the movement of the Jellyfish away from the neighbor in an active motion movement.

Finally, for the Jellyfish moving with ocean current, the swap operation is similar to the one with the active motion toward the neighbor, except that the operation is done between the current Jellyfish and the Jellyfish with the best fitness function instead of a randomly selected neighbor. See Figure 5 for the complete swap operation algorithm.

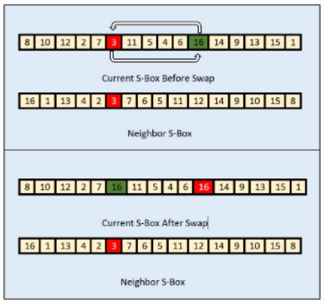


Figure 4: Active motion away from neighbor swap operation

```
Inputs:
    Current Jellyfish cJ
    Best Jellyfish bJ
    Jellyfish Population jP
    Step Unit sU
    Type of Movment tM
Step 1: Start
Step 2: if sU = 0, Go to step 8.
Step 3: if tM = occen current
        Step 3.1: randomly select index i: 0 ≤ i ≤ n × m
        Step 3.2: find the index j of the value bJ[i] in cJ
        Step 3.3: randomly select index k: 0 ≤ k ≤ n × m
        Step 3.4:swap cJ[j] and cJ[k]
Step 4: if tM = Active movment
        Step 4.1: randomly select Jellyfish rJ from jP
        Step 4.2: if fitness value of rJ > fitness value of cJ
                Step 4.2.1: randomly select index i: 0 ≤ i ≤ n × m
                Step 4.2.2: find the index j of the value rJ[i] in cJ
                Step 4.2.3: randomly select index k: 0 ≤ k ≤ n × m
                Step 4.2.4:swap cJ[j] and cJ[k]
        Step 4.3: if fitness value of rJ < fitness value of cJ
                Step 4.3.1: search cJ and rJ for index i, where cJ[i] = rJ[i].
                Step 4.2.3: randomly select index j: 0 ≤ j ≤ n × m
                Step 4.2.4:swap cJ[i] and cJ[k]
Step 5: if tM = Passive movment
        Step 5.1: randomly select index i: 0 ≤ i ≤ n × m
        Step 3.3: randomly select index j: 0 ≤ j ≤ n × m
        Step 3.4:swap cJ[j] and cJ[j]
Step 6: sU = sU – 1.
Step 7: Go to Step 2.
Step 8: end
```

Figure 5: Algorithm of the swap operation

IV. Performance Evaluation

The proposed algorithm is implemented using the C# programming language. As the proposed algorithm is mainly based on a modified version of the Jellyfish search algorithm, it has the same control parameters, namely, the population size, and the number of iterations. In trying to find the best fit to produce a cryptographically strong S-Box in a fast and efficient matter, multiple values for the control parameters were tested, each one with ten executions and averaging out the fitness function of the best generated S-Box from each execution. The experimental results showed that using a population size of 120 Jellyfish, and with 100 Iteration only was the best fit for the implementation without any delay in execution or any processing overhead.

After selecting fixed control parameters values, we used the algorithm to generate a set of S-Boxes, and in order to analyze and test the result, we selected one of the generated S-Boxes shown in Figure 6:

Proposed S-Box

X	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S[x]	6	5	9	D	C	0	7	F	A	B	2	4	E	3	1	8

Figure 6: The generated S-Box

The desirable properties of an S-box are its design simplicity, fast encryption and decryption speed, and resistance against known cryptanalysis attacks. The criteria of a good S-box will encounter most of the standards set by the National Institute of Standards and Technology. Six major state-of-the-art performance criteria are used to assess the security strengths of S-boxes. Namely, 1) Non-Linearity, 2) Strict Avalanche Criterion (SAC), 3) Bit Independence Criterion (BIC), 4) Bijective, 5) Differential

Approximation Probability, 6) Linear Approximation Probability [17]. Nevertheless, it is impossible to achieve all criteria to their best in a single S-box. Their disagreeing nature limits the designer from compromising some of the criteria. For example, correlation immunity conflicts with high nonlinearity, and maximum non-linearity also conflicts with balance. It will depend on the applied problem which criteria we want to achieve and on which we can negotiate.

The nonlinearity degree for the four Boolean functions of the proposed S-boxes is = 4, providing excellent statistics like the maximum, minimum, and average of the functions are 4, the reason for that is quite obvious as nonlinearity is the main focus of the proposed algorithm, and it is used as its fitness function. Hence, the proposed S-boxes offer better nonlinearity, security, and resistance to linear attacks. The quantified SAC for the proposed S-box is 0.4978 which is more than acceptable statistics since it is quite close to the ideal value of 0.5. The calculated BIC values of the three Proposed S-boxes are 0.5019. Which approximated the typical value of 0.5. Therefore, the proposed S-box well satisfies the bits independence criteria. The proposed S-Box is bijective by definition, due to the algorithm used for the S-Box initialization using a logistic map, and presented swap operation in the proposed algorithm. The Differential Approximation probabilities and the linear approximation probability of the generated S-Box are 0.897, and 0.832, respectively.

The statistical analysis results of the three proposed S-Boxes are compared to some of the recently designed 4 × 4 S-Boxes in the literature, Table 1. The comparison showed that the suggested S-Box has mostly equal, to better statistical attributes, which mark the suggested S-boxes as very robust cryptographically and a good fit to be used for lightweight block cipher algorithms.

Table 1: Statistical Comparison of S-Boxes

S-Box	Nonlinearity	SAC	BIC	DP	LP
JS-S-Box	4	0.4978	0.5019	0.897	0.903
[5]	4	0.5037	0.5078	0.726	0.543
[14]	4	0.4967	0.4925	0.664	0.726
[19]	4	0.4899	0.5041	0.901	0.542

V. Conclusion and Future work

In this paper, an efficient 4×4 substitution-box synthesis scheme is presented which is based on the modification of the recent artificial Jellyfish Search algorithm and logistic chaotic map. A novel idea is explored to yield a cryptographically proficient setup of substitution boxes. The performance excellence, consistency, and acceptability of the proposed scheme and S-box are defended by the standard statistical outcomes. The experimental results show that the anticipated S-box is cryptographically more impressive when contrasted with some recently investigated S-boxes.

We intend to use the modified jellyfish search algorithm in more study areas as it proves its performance, we intend on applying the generated S-Box in securing data messages in the IoT network as it was one of the design purposes of having a lightweight S-Box, moreover, trying different chaotic functions and AI algorithms in the generation face of the proposed algorithm can be of value in improving the efficiency and speed. Also testing the proposed algorithm on generating other S-box sizes can be done since we implement the algorithm in a very general term and it can be changed easily to generate other different S-boxes.

Reference

[1] Tesar, Petr. (2010). A New Method for Generating High Non-linearity S-Boxes. Radioengineering. 19.

[2] Hussain, I.; Anees, A.; Al-Maadeed, T.A.; Mustafa, M.T. Construction of S-Box Based on Chaotic Map and Algebraic Structures. *Symmetry* **2019**, *11*, 351.

[3] S. Sahmoud, W. Elmasry, and S. Abudalfa, "Enhancement the Security of AES Against Modern Attacks by Using Variable Key Block Cipher," Int. Arab. J. e Technol., vol. 3, no. 1, pp. 17-26, 2013.

[4] William Easttom, (2021), “Modern Cryptography Applied Mathematics for Encryption and Information Security”

[5] T. Ara, P. G. Shah, and P. M, "Dynamic key Dependent S-Box for Symmetric Encryption for IoT Devices," *2018 Second International Conference on Advances in Electronics, Computers and Communications (ICAECC)*, 2018, pp. 1-5, doi: 10.1109/ICAECC.2018.8479442.

[6] Rana M. Zaki and Hala Bahjat Abdul Wahab, (2021), “A novel of substitution-box design using PLL algorithms in magic cube”, Periodicals of Engineering and Natural Sciences Vol. 9, No. 4, October 2021, pp.674-684

[7] Lang Li, Jinggen Liu, Ying Guo, Botao Liu, A new S-box construction method meeting strict avalanche, criterion, Journal of Information Security and Applications, Volume 66, 2022, 103135, ISSN 2214-2126,

[8] M. Long and L. Wang, "S-Box Design Based on Discrete Chaotic Map and Improved Artificial Bee Colony Algorithm," in *IEEE Access*, vol. 9, pp. 86144-86154, 2021, doi: 10.1109/ACCESS.2021.3069965.

[9] Alhadawi, H.S., Majid, M.A., Lambić, D., et al. A novel method of S-box design based on discrete chaotic maps and

cuckoo search algorithm. Multimed Tools Appl 80, 7333–7350 (2021).

[10] Alshammari, B.M.; Guesmi, R.; Guesmi, T.; Alsaif, H.; Alzamil, A. Implementing a Symmetric Lightweight Cryptosystem in Highly Constrained IoT Devices by Using a Chaotic S-Box. *Symmetry* 2021, *13*, 129.

[11] 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," in *IEEE Access*, vol. 9, pp. 84605-84618, 2021, doi: 10.1109/ACCESS.2021.3087139.

[12] Farhan, Alaa & Subhi, Rasha & Yassein, Hassan & Al-Saidi, Nadia & Majeed, Ghassan. (2020). A NEW APPROACH TO GENERATE MULTI S-BOXES BASED ON RNA COMPUTING. International journal of innovative computing, information & control: IJICIC. 16. 331-348. 10.24507/ijicic.16.01.331.

[13] Alhadawi, Hussam & Zolkipli, Mohamad & Ahmad, Musheer. (2018). A novel efficient substitution-box design based on firefly algorithm and discrete chaotic map. Neural Computing and Applications. 31. 10.1007/s00521-018-3557-3.

[14] Markus Ullrich??, Christophe De Canni`ere, Sebastiaan Indesteege, Ozg¨ul K¨u¸c¨uk, Nicky Mouha ¨ ? ? ?, and Bart Preneel, “Finding Optimal Bitsliced Implementations of 4 × 4-bit S-boxes”, 2018.

[15] Liu, Liyan & Wang, Xingyuan. (2018). A Novel Method for Constructing the S-Box Based on Spatiotemporal Chaotic Dynamics. Applied Sciences. 8. 2650. 10.3390/app8122650.

[16] Eesa Al Solami.; Ahmad, M.; Volos, C.; Doja, M.N.; Beg, M.M.S. A New Hyperchaotic System-Based Design for Efficient Bijective Substitution-Boxes. Entropy 2018, 20, 525.

[17] Musheer Ahmad, Nikhil Mittal, Prerit Garg, Manaff Maftab Khan, Efficient cryptographic substitution box design using traveling salesman problem and chaos, Perspectives in Science, Volume 8, 2016, Pages 465-468, ISSN 2213-0209,

[18] Y. Tian and Z. Lu, "S-box: Six-dimensional compound hyperchaotic map and artificial bee colony algorithm," in Journal of Systems Engineering and Electronics, vol. 27, no. 1, pp. 232-241, Feb. 2016.

[19] Mihajloska, H., & Gligoroski, D. (2012). Construction of Optimal 4-bit S-boxes by Quasigroups of Order 4. SECURWARE 2012.

[20] Jui-Sheng Chou, Dinh-Nhat Truong, A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean, Applied Mathematics and Computation, Volume 389, 2021, 125535, ISSN 0096-3003,

[21] Khan, M.A., Ali, A., Jeoti, V., et al. A Chaos-Based Substitution Box (S-Box) Design with Improved Differential Approximation Probability (DP). Iran J Sci Technol Trans Electr Eng 42, 219–238 (2018).