symmetry

MDPI

Article

Construction Method and Performance Analysis of Chaotic S-Box Based on a Memorable Simulated Annealing Algorithm

Juan Wang [1,2,*], Yangqing Zhu [2], Chao Zhou [2] and Zhiming Qi [3]

[1] College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China
[2] College of Electronic and Information Engineering, Heilongjiang University of Science and Technology, Harbin 150027, China; dmiyoung@163.com (Y.Z.); Zcyjszy@163.com (C.Z.)
[3] State Grid Liaoning Maintenance Company, Shenyang 110006, China; qyz7617936@163.com
* Correspondence: 1131814@s.hlju.edu.cn

Received: 20 November 2020; Accepted: 17 December 2020; Published: 19 December 2020

Abstract: The substitution box (S-box) is the only nonlinear components in the symmetric block cipher. Its performance directly determines the security strength of the block cipher. With the dynamic characteristics degradation and the local periodic phenomenon of digital chaos, and the security problems caused by them becoming more and more prominent, how to efficiently generate an S-box with security guarantee based on chaos has gradually attracted the attention of cryptographers. In this paper, a chaotic S-box construction method is proposed based on a memorable simulated annealing algorithm (MSAA). The chaotic S-box set is produced by using the nonlinearity and randomness of the dynamic iteration of digital cascaded chaotic mapping. The composite objective function is constructed based on the analysis of the performance indexes of S-box. The MSAA is used to efficiently optimize the S-box set. The matrix segmentation and scrambling operations are carried out on the optimized S-box. The cryptographic performance of chaotic S-box is tested and analyzed, and compared with the mainstream chaotic S-box of the same kind. The results show that the S-box constructed in this paper can not only stably and efficiently generate chaotic S-box with better performance, but also make an effective exploration of the construction of chaotic S-boxes based on intelligent algorithms.

Keywords: substitution box; chaos; simulated annealing algorithm

1. Introduction

The block cipher is an important branch of cryptography, which can be used not only to encrypt information directly, but also as an effective means to construct hash functions and digital signatures [1–3]. The block cipher has been widely applied to the field of information security because of high speed, easy standardization, and convenience for hardware and software implementation. S-box is the only nonlinear component in the symmetric block cipher. Its performance directly determines the security strength of the block cipher. Therefore, the construction of a secure and efficient S-box has become one of the key factors in the design of the block cipher [4–6].

In traditional cryptography, Advanced Encryption Standard (AES) uses algebraic methods to construct the S-box. Although high nonlinearity can be obtained because there are only nine algebraic formulas, its structure is too simple and the affine transformation period and iterative output period are too short, so the differential performance is relatively weak and it is difficult to resist algebraic attacks [7,8]. Meanwhile, the AES uses static S-box, and its form content is both public and unchanged, which is easy to be analyzed and utilized by the deciphers. Chaos is a deterministic, random-like

process in nonlinear dynamic systems.With the deepening of S-box research and the development of chaos theory, dynamic S-box is constructed based on nonlinearity, randomness, initial sensitivity and unpredictability of chaos to realize information confusion, which has been gradually recognized by cryptographers and has made considerable development [9–14]. The inherent characteristic of the chaotic system provides a good foundation for constructing the S-box. However, the performance of S-box may be unstable due to the degradation of dynamic characteristics and local periodic phenomenon of digital chaos, so there are some security risks. With the dynamic characteristics degradation and the local periodic phenomenon of digital chaos, and the security problems caused by them becoming more and more prominent, the efficient generation of the chaotic S-box with security guarantees remains to be further studied.

In recent years, intelligent algorithms have been widely developed. Their feasibility and superiority in solving optimization problems have gradually attracted the attention of cryptographers, and have also provided a new idea for constructing the chaotic S-box. A design method of S-box based on chaos and genetic algorithm is proposed in the literature [15]. The crossover mutation of genetic algorithm is used to generate the chaotic S-box with better nonlinearity. The performance of generated S-box would be easily affected by the dynamic characteristics degradation of digital chaos. A construction method of S-box based on chaos and firefly algorithm is proposed in the literature [16]. The firefly algorithm is adopted to optimize the generated chaotic S-box set, but the construction efficiency is not high. The convergence time of the algorithm is long, and the performance is also restricted by the chaotic S-box set. A design scheme for constructing high nonlinear chaotic S-box based on genetic algorithm is proposed in the literature [17], which takes nonlinearity as the only optimization objective leads to little improvement in other cryptographic performances of the chaotic S-box.

Compared with the exhaustive search algorithm, the heuristic algorithm can use some of the searched information to change its own search strategy. If the parameters are set properly, the search efficiency of the heuristic algorithm is more efficient than the exhaustive algorithm [18,19]. MSAA is an improvement of the traditional simulated annealing algorithm (SAA). It can overcome the "forgetfulness" in the process of optimization by memorizing the optimal solution currently encountered. Thus, MSAA improves the efficiency and accuracy of global optimal search, which is especially suitable for solving combinatorial optimization problems. Compared to other heuristic algorithms, MSAA is a probabilistic local search method. It can efficiently find the approximate optimal solution of the problem due to its asymptotic convergence [20]. In this paper, the chaotic S-box set is generated iteratively by digital cascaded chaotic mapping, and the composite objective function is constructed based on the analysis of the S-box performance index. The MSAA is used to efficiently optimize the chaotic S-box set, which can not only obtain the S-box with relatively better cryptographic performance, but also ensure the stability of the S-box security performance. Meanwhile, the chaotic S-box obtained by optimization is transformed by matrix segmentation and scrambling operations to get rid of the performance restriction of the chaotic S-box set, and further enhance the cryptographic performance of the chaotic S-box.

The rest of this paper is as follows: the second section introduces the digital cascaded chaotic mapping. The third section describes the MSAA. The fourth section introduces a design scheme of chaotic S-box based on MSAA. The fifth section analyzes the evaluation indexes and experimental results of S-box performance. The sixth section gives the conclusions of this paper.

2. Chaotic System

Chaos is a deterministic, random-like phenomenon in nonlinear dynamic systems. This process is aperiodic, non-convergent but bounded and extremely sensitive to initial values [21]. In order to design a secure and efficient chaotic S-box, the application of chaotic systems should follow the following principles. One is that the selected chaotic system should be easy to implement and have efficient iteration, the other is that it can overcome the local periodic problem of the digitization

process of the chaotic system. With the dynamic characteristics degradation and the local periodic phenomenon of digital chaos, the security problems caused by them become more and more prominent. The cryptographers are committed to the study of mathematical chaotic models with excellent performance, simple structure, and easy implementation, so that they can better play the chaotic characteristics in the construction of S-box.

To improve the pseudo-random performance and dynamic characteristics of digital chaotic sequences, a digital cascaded chaotic mapping has been proposed in the literature [22] based on one-dimensional discrete chaotic mappings Logistic and Tent. The iterative output of Logistic chaotic mapping is used as the iterative input of Tent chaotic mapping, and the iterative output of Tent chaotic mapping is used as the input of the next iteration of Logistic chaotic mapping. Then, the one-dimensional discrete chaotic mapping equation after cascading is

$$x_{n+1} = 1 - |1 - 4\mu x_n(1 - x_n)|. \tag{1}$$

In Equation (1), system parameter $\mu \in (0, 2)$, initial value $x_n \in (0, 1)$, substituting the real-valued chaotic sequence generated by cyclic iteration into Equation (2) for digital quantization. Then, the digital cascaded chaotic sequence can be obtained:

$$T_n = \begin{cases} \lfloor 4x_n \rfloor & 0 \le x_n < 1/4a \\ \lfloor 4x_n - a \rfloor & 1/4a \le x_n < 1/2a \\ \lfloor 3a - 4x_n \rfloor & 1/2a \le x_n < 3/4a \\ \lfloor 4a - 4x_n \rfloor & 3/4a \le x_n < a \end{cases}. \tag{2}$$

In Equation (2), take $a = 2^7$, $T_n \in [0, 255]$, which exactly corresponds to the unsigned integer range represented by 8 bits.

The studies have shown that digital cascaded chaotic mapping has efficient iteration and is easy to implement. The mapping has higher complexity, larger parameter space, and stronger initial value sensitivity. A large number of pseudo-random sequences with excellent performance and the great difference can be obtained by tiny changes in initial values and system parameters. The S-box is constructed by using the nonlinearity and randomness of the dynamic iteration of the digital cascaded chaotic mapping. The construction method is simple to operate but can effectively enhance the confusion effect, thus providing a reliable guarantee for the security and efficiency of the S-box construction.

3. Optimization Process of MSAA

The traditional SAA jumps out of the local optimal solution through "probability judgment" and tends to the global optimal [23,24]. However, this method may also cause the algorithm to ignore the optimal solution currently encountered. It is difficult to ensure that the final solution must be the global optimal solution. The MSAA proposed by literature [25] can memorize the optimal solution encountered in the search process. When the search process is over, the searched optimal solution is compared with the memorized optimal solution, and the better one is taken as the final result. The accuracy of the optimization result would be further improved. Since the time required to realize the memory function is extremely short, the MSAA still has high search efficiency.

As shown in Figure 1, the optimization process of the MSAA is illustrated as follows:

Step 1. In the solution space, set the initial solution S_0, the initial temperature T_0, the minimum temperature T_{min}, the number of iterations L for each T value, and calculate the objective function f_0 of the initial solution S_0. The attenuation function of temperature T is $T_{k+1} = \alpha \cdot T_k$, where $\alpha \in (0,1)$, $k = 0, 1, \cdots, n$.

Step 2. A new solution S^* is randomly generated near the initial solution S_0, and the objective function f^* of the new solution S^* is calculated.

Step 3. f^* is compared with f_0. If f^* is better than f_0, that is, $\Delta f = f^* - f_0 \geq 0$, then accept the new solution S^* and assign S^* and f^* to S_0 and f_0, respectively. Otherwise, memorize the current optimal solution S_0, and accept the new solution S^* according to the probability of Mctropolis criterion. The Mctropolis criterion takes Δf and temperature T as input, and the output is the acceptance probability between 0 and 1. Its expression is

$$P = \exp[-\Delta f(x)/T]. \tag{3}$$

Step 4. If the number of iterations L is reached, it is judged whether the termination criterion is met. When the temperature is lower than the minimum temperature or the memorized optimal solution has no changes for multiple consecutive times, then the optimization search is terminated. Otherwise, return to step 2.

Step 5. If the number of iterations L is reached and the termination criterion is met, the searched optimal solution is compared with the memorized optimal solution, and the better one is output as the result. Otherwise, decrease the temperature, reset the number of iterations, and return to step 2.

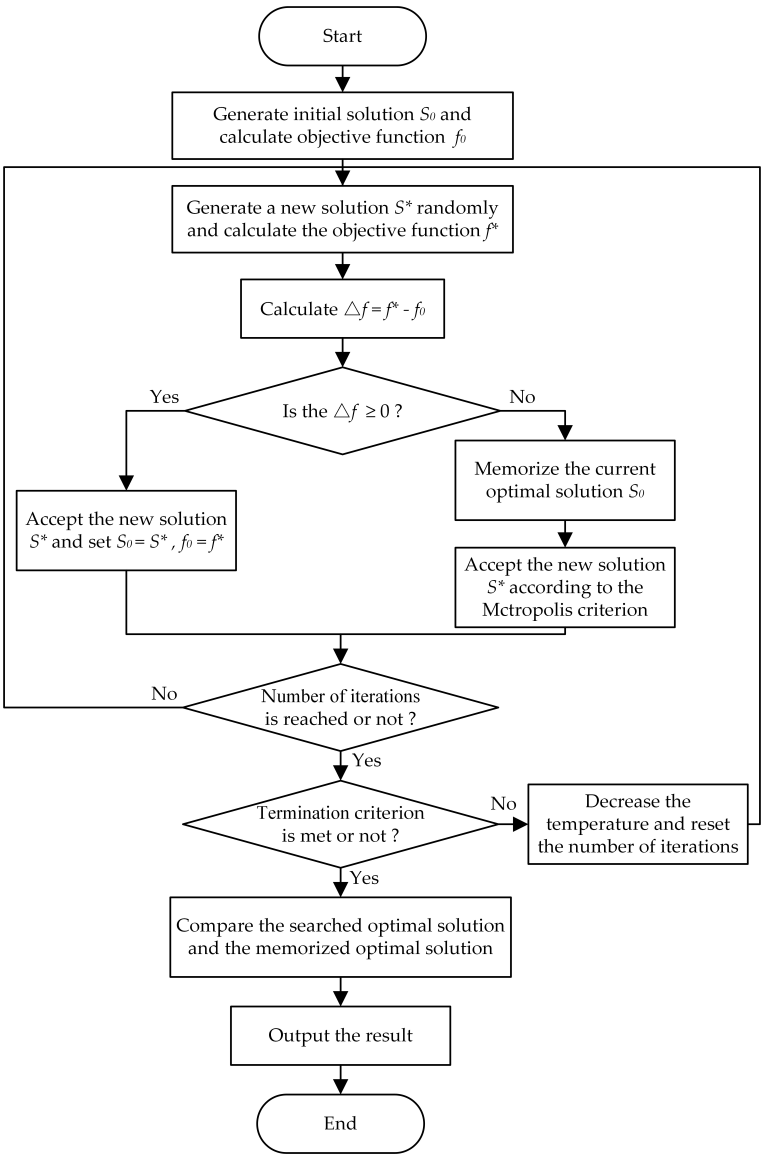

Figure 1. Flowchart of MSAA.

4. Construction Method of Chaotic S-Box

As shown in Figure 2, the iterated digital cascaded chaotic sequence is traversed and screened to generate the set of chaotic S-boxes. The set is optimized by MSAA to obtain the chaotic S-box with excellent performance. Then, the chaotic S-box obtained is segmented and scrambled to generate the final chaotic S-box.

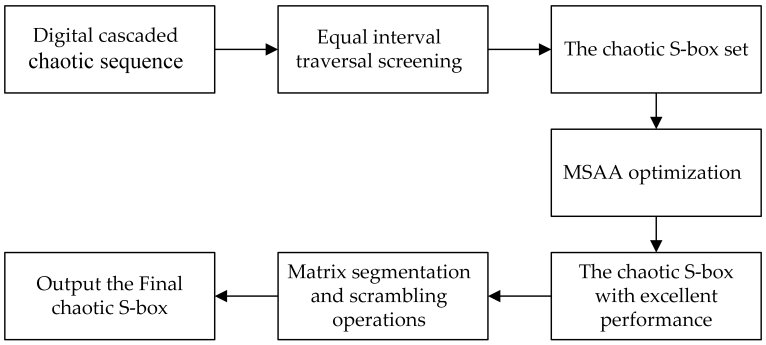

Figure 2. Construction of chaotic S-Box based on MSAA.

The specific construction process is described as follows:

Step 1. Set the initial conditions of the digital cascaded chaotic mapping, that is, $\mu = 1.999$, $x_n = 0.76$, and the iterative operation is performed.

Step 2. The iterative interval of digital cascaded chaotic mapping is evenly divided into 256 intervals $D_i(i = 0, 1, \cdots, 255)$. If the iterative output T_n exists in the interval D_i, the corresponding T_n value is saved and the iteration continues; if T_n does not exist in the interval D_i or has already been saved, the T_n value is not saved and the iteration continues until 256 intervals have been traversed.

Step 3. The outputs Y_n are arranged line-by-line in the order of generation and converted into a table of 16×16, which is the constructed 8×8 S-box. By slightly changing the initial value, the set of chaotic S-boxes can be obtained through the dynamic iteration of the digital cascaded chaotic mapping.

Step 4. Set the initial conditions of the MSAA, the initial temperature is $T = 100$, the lowest temperature is $T_{\min} = 0$, the number of iterations for each T value is $L = 10$. In the generated chaotic S-box set, one S-box is randomly selected as the initial solution S_0, other chaotic S-boxes encountered during the optimization process of MSAA will be used as new solution S^*. The objective function of new solution f^* would be compared with the objective function of initial solution f_0. Since the nonlinearity and difference uniformity are the two most important performance indexes to measure the security performance of S-box, a composite objective function is constructed as

$$F(s) = N_f - \delta_f. \tag{4}$$

In Equation (4), N_f is the nonlinearity of the S-box, δ_f is the difference uniformity of the S-box. The greater the nonlinearity of the S-box and the smaller the difference uniformity, the better its security performance. Therefore, the larger the composite objective function $F(s)$, the better the cryptographic performance of the S-box. Based on the constructed composite objective function, the MSAA is used to efficiently optimize the set of chaotic S-boxes, and a chaotic S-box with excellent cryptographic performance would be obtained.

Step 5. As is shown in Figure 3, the matrix segmentation and scrambling operations are performed on the optimized chaotic S-box, and the 16×16 chaotic S-box is segmented by matrix according to the following rule:

$$S = \begin{bmatrix} s_{00} & \cdots & s_{07} & s_{08} & \cdots & s_{0F} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ s_{70} & \cdots & s_{77} & s_{78} & \cdots & s_{7F} \\ s_{80} & \cdots & s_{87} & s_{88} & \cdots & s_{8F} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ s_{F0} & \cdots & s_{F7} & s_{F8} & \cdots & s_{FF} \end{bmatrix} = \begin{bmatrix} S_1 & S_2 \\ S_3 & S_4 \end{bmatrix}, \tag{5}$$

$$\begin{aligned} S_1 &= \begin{bmatrix} s_{00} & \cdots & s_{07} \\ \vdots & \vdots & \vdots \\ s_{70} & \cdots & s_{77} \end{bmatrix} \quad S_2 = \begin{bmatrix} s_{08} & \cdots & s_{0F} \\ \vdots & \vdots & \vdots \\ s_{78} & \cdots & s_{7F} \end{bmatrix} \\ S_3 &= \begin{bmatrix} s_{80} & \cdots & s_{87} \\ \vdots & \vdots & \vdots \\ s_{F0} & \cdots & s_{F7} \end{bmatrix} \quad S_4 = \begin{bmatrix} s_{88} & \cdots & s_{8F} \\ \vdots & \vdots & \vdots \\ s_{F8} & \cdots & s_{FF} \end{bmatrix} \end{aligned}. \tag{6}$$

Then, the four segmented matrices are performed on scrambling operation respectively according to the following rule:

$$(S_m)^T \xrightarrow{r_{7-n} \leftrightarrow r_n} S_m{}'. \tag{7}$$

In Equation (7), $m = 1, 2, 3, 4$, $n = 0, 1, 2, 3$, T represents the transpose of the matrix, $r_{7-n} \leftrightarrow r_n$ means that $7 - n$ rows and n rows of the matrix are interchanged, then

$$\begin{aligned} S_1{}' &= \begin{bmatrix} s_{07} & \cdots & s_{77} \\ \vdots & \vdots & \vdots \\ s_{00} & \cdots & s_{70} \end{bmatrix} \quad S_2{}' = \begin{bmatrix} s_{0F} & \cdots & s_{7F} \\ \vdots & \vdots & \vdots \\ s_{08} & \cdots & s_{78} \end{bmatrix} \\ S_3{}' &= \begin{bmatrix} s_{87} & \cdots & s_{F7} \\ \vdots & \vdots & \vdots \\ s_{80} & \cdots & s_{F0} \end{bmatrix} \quad S_4{}' = \begin{bmatrix} s_{8F} & \cdots & s_{FF} \\ \vdots & \vdots & \vdots \\ s_{88} & \cdots & s_{F8} \end{bmatrix} \end{aligned}. \tag{8}$$

Reorganize them to get

$$S' = \begin{bmatrix} S_1{}' & S_2{}' \\ S_3{}' & S_4{}' \end{bmatrix}. \tag{9}$$

A secondary scrambling operation is performed to S' according to the following rule

$$(S')^T \xrightarrow{r_{15-h} \leftrightarrow r_h} S_{\text{Final}}. \tag{10}$$

In Equation (10), $h = 0, 1, ..., 7$. The output S_{Final} is the final chaotic S_1-box shown in Table 1, and repeat the above method to generate chaotic S_2-box and chaotic S_3-box.

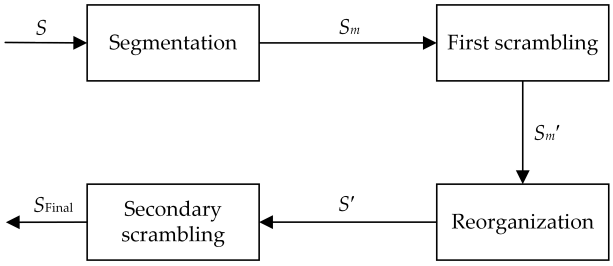

Figure 3. Matrix segmentation and scrambling operations.

Table 1. Final chaotic S_1-box in this paper.

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

5. Testing and Analysis of Performance

5.1. Nonlinearity

Comprehensive analysis of existing research shows that the design of an S-box usually has five criteria: nonlinearity, difference uniformity, strict avalanche criterion (SAC), bit independence criterion (BIC), and bijectivity. The larger the nonlinearity value of the S-box, the stronger its ability to resist linear cryptographic attacks.

Although the S-boxes used in block ciphers are all presented in the form of tables, their essence is a nonlinear combination function of multiple inputs and multiple outputs mapping from F_2^n to F_2^m. A $n \times m$ S-box can generally be represented as $S : \{0,1\}^n \to \{0,1\}^m$, which is composed of m n-bit Boolean functions $f_i(x_1, x_2, \cdots, x_n), i = \{1, 2, \cdots, m\}$, that is,

$$S(x) = (f_1(x_1, x_2, \cdots, x_n), f_2(x_1, x_2, \cdots, x_n), \cdots, f_m(x_1, x_2, \cdots, x_n)). \tag{11}$$

Let Boolean function $f(x) : F_2^n \to F_2^m$, $x = (x_1, x_2, \ldots, x_n)$, $w = (w_1, w_2, \ldots, w_n)$, $x \in F_2^n$, $w \in F_2^n$, and the dot product of x and w be defined as

$$x \cdot w = \sum_{i=1}^{n} x_i w_i. \tag{12}$$

Then, the first-order Walsh cyclic spectrum of an n-ary Boolean function $f(x)$ is defined as

$$S_{(f)}(w) = 2^{-n} \sum_{x \in F_2^n} (-1)^{f(x) \oplus x \cdot w}. \tag{13}$$

For the convenience of calculation, the nonlinearity of $f(x)$ represented by the Walsh cyclic spectrum is defined as

$$N_f = 2^{n-1}(1 - 2^{-n} \max_{w \in F_2^n} \left| S_{(f)}(w) \right|). \tag{14}$$

For the 8×8 chaotic S-box constructed in this paper, the Walsh cyclic spectrums output by eight Boolean functions are substituted into Equation (14), respectively. In turn, the nonlinearity values can be obtained. As shown in Table 2, the three chaotic S-boxes generated by the method of this paper are marked as S_1-box, S_2-box, and S_3-box, the nonlinearities of which are all above 104 and the average values are 108, 108, and 107.5, respectively. The three chaotic S-boxes randomly generated based on the the digital cascaded chaotic sequence in this paper are marked as S_4-box, S_5-box, and S_6-box,

the average values of their nonlinearities are 103, 108, and 106.5, respectively. Through comparison, it can be seen that the method of this paper can overcome the instability of the S-box performance caused by the dynamic characteristics degradation and the local periodic phenomenon of digital chaos. At the same time, compared with other chaotic S-boxes generated based on intelligent algorithms, the chaotic S-box proposed has better and more stable nonlinear characteristics and can effectively resist the best linear approximation attack.

Table 2. Comparison of nonlinearities.

S-box	Maximum	Minimum	Average
S_1-box	110	104	108
S_2-box	110	104	108
S_3-box	110	104	107.5
S_4-box	116	100	103
S_5-box	110	104	108
S_6-box	108	100	106.5
Ref. [15]	110	104	107.25
Ref. [16]	108	106	107.5
Ref. [26]	108	106	107
Ref. [27]	108	104	106.5

5.2. Difference Uniformity

Differential analysis is one of the most effective attacks of block ciphers. In order to measure the ability of a cipher to resist the differential analysis, the concept of difference uniformity has been introduced. The differential analysis mainly realizes the attack through the imbalance of the input/output XOR distribution. If the S-box has an equal probability of input/output XOR distribution, it can effectively resist the differential analysis.

In practice, the input/output XOR distribution of $f(x)$ is generally described by difference approximation probability

$$DP_f = \max_{\Delta x \neq 0, \Delta y} \left(\frac{\#\{x \in X \,|\, f(x) \oplus f(x \oplus \Delta x) = \Delta y\}}{2^n} \right). \tag{15}$$

In Equation (15), DP_f means the maximum probability that the output difference is Δy when the input difference is Δx. X represents the set of all possible inputs of x, and 2^n is the number of all elements in the set X. The smaller the value of the difference approximation probability DP_f of S-box, the stronger its ability to resist differential attacks.

For the given input difference $\Delta x = 0, 1, 2, \cdots 255$, calculate x to take all possible values and maximum number for $\Delta y = 0, 1, 2, \cdots 255$ in turn. Then, the table of final difference uniformity distribution can be obtained. As shown in Table 3, the maximum of the input and output difference of the chaotic S_1-box in this paper is 10. As shown in Table 4, the difference approximation probabilities of the three chaotic S-boxes generated by the method of this paper are all 3.9062%. The difference approximation probabilities of the three chaotic S-boxes randomly generated based on the cascaded chaotic sequence in this paper are 4.6875%, 3.9062%, and 3.9062%, respectively. Through comparison, it can be seen that the method of this paper can overcome the instability of the S-box performance caused by the dynamic characteristics degradation and the local periodic phenomenon of digital chaos. At the same time, compared with other chaotic S-boxes generated based on intelligent algorithms, the chaotic S-box proposed has better and more stable difference approximation probabilities, indicating that it has excellent and stable ability to resist differential attacks [28].

Table 3. The input/output difference distribution of the chaotic S_1-box in this paper.

-	8	6	6	6	10	6	6	6	6	6	6	6	6	6	6
6	10	6	8	6	6	6	6	8	8	6	8	6	8	6	6
10	6	10	6	6	6	6	6	6	6	6	6	6	6	8	6
8	6	8	6	6	6	10	8	6	6	6	8	6	6	8	6
8	6	6	6	4	8	6	6	6	8	8	6	6	6	6	6
6	8	8	8	8	6	6	6	8	6	6	8	8	6	8	8
6	6	6	6	6	6	8	6	8	6	6	6	8	6	6	6
10	6	6	6	8	6	8	6	8	6	8	6	6	8	8	6
6	4	6	4	6	6	6	6	6	8	6	6	6	6	8	6
10	8	8	8	6	6	6	8	6	6	6	6	8	6	6	6
6	6	8	6	6	6	10	6	8	6	6	6	6	6	6	8
6	6	6	8	8	8	6	8	6	10	6	8	6	8	6	6
6	10	10	6	6	6	8	8	6	6	8	6	6	6	6	8
8	6	8	8	8	8	8	6	8	6	6	6	6	6	8	6
6	6	6	4	8	6	6	8	10	8	4	6	10	8	6	8
8	6	8	8	6	4	10	6	6	6	6	6	6	8	6	6

Table 4. Comparison of DP_f.

S-box	DP
S_1-box	3.9062%
S_2-box	3.9062%
S_3-box	3.9062%
S_4-box	4.6875%
S_5-box	3.9062%
S_6-box	3.9062%
Ref. [16]	3.9062%
Ref. [17]	3.9062%
Ref. [26]	3.9062%
Ref. [27]	4.2960%

5.3. Strict Avalanche Criterion

In order to resist the attack method based on relatively large change in the output caused by the input change, the SAC is proposed in the literature [29]. Half of the output result would be changed if one input bit is changed, and the construction of a correlation matrix to judge whether $f(x)$ meets the SAC. Each element value a_{ij} of the correlation matrix represents the correlation strength between the i bit of the ciphertext and the j bit of the plaintext. If the values of each element of the correlation matrix are all close to 0.5, it can indicate that $f(x)$ meets the SAC. The correlation matrix of the chaotic S_1-box generated by the method of this paper is shown in Table 5, as shown in Table 6, the average values of the correlation matrix of the three chaotic S-boxes in this paper are 0.5007, 0.5007, and 0.5008, respectively, which are all closer to 0.5. The SAC performances of the three chaotic S-boxes randomly generated based on the cascaded chaotic sequence in this paper are 0.4836, 0.5012, and 0.5048, respectively. Through comparison, it can be seen that the method of this paper can overcome the instability of the S-box performance caused by the dynamic characteristics degradation and the local periodic phenomenon of digital chaos. At the same time, compared with other chaotic S-boxes generated based on intelligent algorithms, the chaotic S-box proposed has better and more stable SAC performance.

Table 5. The correlation matrix of the chaotic S_1-box in this paper.

0.4688	0.5938	0.4688	0.4844	0.5469	0.5000	0.5156	0.4375
0.4531	0.4844	0.4844	0.5156	0.5000	0.4844	0.5000	0.5156
0.4688	0.5000	0.4219	0.4688	0.5156	0.4844	0.5313	0.6094
0.5156	0.5313	0.5313	0.5469	0.5625	0.4688	0.4844	0.4531
0.5000	0.4844	0.5156	0.5156	0.5156	0.5313	0.4688	0.5156
0.5000	0.4531	0.5625	0.5313	0.5000	0.5000	0.5625	0.5156
0.4844	0.5156	0.5313	0.4844	0.5781	0.4688	0.4844	0.4844
0.5313	0.5000	0.5313	0.5000	0.4531	0.4844	0.4688	0.5469

Table 6. Comparison of the average values of the correlation matrix.

S-box	Average
S_1-box	0.5007
S_2-box	0.5007
S_3-box	0.5008
S_4-box	0.4836
S_5-box	0.5010
S_6-box	0.5048
Ref. [15]	0.5046
Ref. [16]	0.4943
Ref. [17]	0.4953
Ref. [26]	0.5015
Ref. [27]	0.4990

5.4. Bit Independence Criterion

The BIC is one of the essential analysis elements in the design of the S-box. For the Boolean functions $f_i(x)$ and $f_j(x)(i \neq j, 1 \leq i, j \leq n)$ between any two output bits of the S-box, if the S-box meets the BIC-nonlinearity, $f_i(x) \oplus f_j(x)$ should meet the characteristics of nonlinearity. If the S-box meets BIC-SAC, $f_i(x) \oplus f_j(x)$ should meet the SAC.

As shown in Table 7, the nonlinearity value of $f_i(x) \oplus f_j(x)$ of the chaotic S_1-box is larger, indicating that it meets the characteristics of nonlinearity. As shown in Table 8, the values of each element of the correlation matrix of $f_i(x) \oplus f_j(x)$ of the chaotic S_1-box are all close to 0.5, indicating that it meets the SAC. As shown in Table 9, the BIC-nonlinearity average values of the three chaotic S-boxes generated by the method of this paper are 104.21, 104.21, and 104.20, respectively, the BIC-SAC average values of the three chaotic S-boxes are 0.5012, 0.5012, and 0.5011, respectively. The BIC-nonlinearity averages of the three chaotic S-boxes randomly generated based on the cascaded chaotic sequence in this paper are 101.90, 104.21, and 103.53, respectively, and the BIC-SAC averages are 0.4954, 0.5016, and 0.5038, respectively. Through comparison, it can be seen that the method of this paper can overcome the instability of the S-box performance caused by the dynamic characteristics degradation and the local periodic phenomenon of digital chaos. At the same time, compared with other chaotic S-boxes generated based on intelligent algorithms, the chaotic S-box proposed has better and more stable BIC.

Table 7. BIC-nonlinearity of the chaotic S_1-box in this paper.

-	108	106	108	102	108	102	104
108	-	106	102	102	104	106	106
106	106	-	106	102	108	104	106
108	102	106	-	102	104	108	102
102	102	102	102	-	100	104	104
108	104	104	104	100	-	104	100
102	106	104	108	104	104	-	104
104	106	106	102	104	100	104	-

Table 8. BIC-SAC of the chaotic S_1-box in this paper.

-	0.5020	0.4902	0.5000	0.4883	0.5195	0.5098	0.4980
0.5020	-	0.4844	0.4961	0.5059	0.5039	0.5098	0.5056
0.4902	0.4844	-	0.5052	0.5007	0.4717	0.5015	0.5093
0.5000	0.4961	0.5052	-	0.5059	0.5017	0.5010	0.5005
0.4883	0.5059	0.5017	0.5059	-	0.4941	0.5056	0.5056
0.5195	0.5039	0.4707	0.5017	0.4941	-	0.5056	0.5059
0.5098	0.5098	0.5015	0.5010	0.5056	0.5056	-	0.5059
0.4980	0.5056	0.5093	0.5005	0.5056	0.5059	0.5059	-

Table 9. Comparison of BIC.

S-box	BIC-nonlinearity average	BIC-SAC average
S_1-box	104.21	0.5012
S_2-box	104.21	0.5012
S_3-box	104.20	0.5011
S_4-box	101.90	0.4945
S_5-box	104.21	0.5016
S_6-box	103.53	0.5038
Ref. [15]	103.86	0.5034
Ref. [16]	104.35	0.4982
Ref. [17]	104.07	0.5021
Ref. [26]	104.21	0.5016
Ref. [27]	103.18	0.4992

5.5. Bijectivity

Generally, the S-box must also satisfy bijectivity in the application. Ref. [30] has given the condition to satisfy the bijectivity of S-box

$$\omega t(\sum_{i=1}^{n} a_i f_i(x)) = 2^{n-1}. \tag{16}$$

In Equation (16), $a_i \in \{0,1\}$, $(a_1, a_2, \cdots, a_n) \neq (0,0,\cdots,0)$, $f_i(x)$ is the Boolean function of each component of the S-box, and $wt()$ is the Hamming weight.

The standard value of bijectivity of an S-box is 128. It can be seen from the calculation that the sums of linear operations of the Boolean function of each component of the chaotic S-box in this paper are all 128 and have different output values between [0,255], so bijectivity is satisfied.

5.6. Implementation Efficiency

The S-box construction should also consider implementation performance. The experiment is performed on a computer whose central processing unit (CPU) is Intel Core i5-6200 2.4 GHz (Manufacturer location: Santa Clara, California, USA). The function simulation of S-box is carried out on ModelSim-Altera SE 6.6d (Manufacturer location: Santa Clara, California, USA) software using Verilog HDL, and the execution time is about 0.050 s. Under the same conditions, the execution time for SubBytes of AES is 55.067 s. The calculation of SubBytes transformation is obtained by taking the inverse of the multiplication in GF(2^8) and performing affine transformation. Generally speaking, the SubBytes transformation is often implemented in look-up-tables (LUT). The execution time of LUT implementation is 0.059 s. Therefore, the required storage space by the method in this paper is less than by LUT and SubBytes of AES.

The hardware is realized by using the storage blocks integrated within field programmable gate array (FPGA) to generate LUT. The target device is Altera Cyclone III EP3C16F484C6 (Manufacturer location: San Jose, CA, USA). The Altera Quartus II 11.0 software is used for a logic synthesis test of the designed S-box. As shown in Table 10, the chaotic S-box proposed consumes 73 logic elements (LEs), and the highest clock frequency is 192.93 MHz. Compared with the typical LUT-based S-boxes and logic circuits of the SubBytes in AES, the chaotic S-box proposed has less area consumptions and higher clock frequency.

Table 10. Comparison of Implementation.

	The Number of LEs	Highest Frequency
Proposed S-box	73	192.93 MHz
logic circuits of the AES SubBytes	87	47.55 MHz
LUT-based of AES S-box	237	183.82 MHz

6. Conclusions

A construction method of chaotic S-box based on MSAA is proposed in this paper. The dynamic iteration of the digital cascaded chaotic mapping is used to generate the chaotic S-box set, which effectively alleviates the adverse effect of the dynamic characteristics degradation of digital chaos on the security performance of the S-box. The construction of the composite objective function and the application of the MSAA improve the accuracy and efficiency of the optimization of the chaotic S-box set. The matrix segmentation and scrambling operations are adopted to further enhance the confusion of chaotic S-box, which makes it get rid of the restriction of the performance of the chaotic S-box set. The chaotic S-boxes constructed by this method are tested and analyzed for five cryptographic performances, and compared with other chaotic S-boxes generated based on intelligent algorithms. The results show that the method proposed in this paper can stably and efficiently generate chaotic S-boxes with better cryptographic performance, thus providing a reliable security guarantee for its application.

Author Contributions: Conceptualization, J.W.; Data curation, J.W.; Formal analysis, C.Z.; Funding acquisition, J.W.; Investigation, J.W., Y.Z., C.Z. and Z.Q.; Methodology, J.W.; Project administration, J.W.; Resources, J.W.; Software, C.Z.; Supervision, J.W.; Validation, J.W. and Z.Q.; Visualization, Y.Z.; Writing–original draft, J.W.; Writing–review & editing, J.W. and Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Heilongjiang Fundamental Research Foundation for the Local Universities; Science and Technology Innovation Foundation of Harbin, Grant No. 2017RAQXJ031; Heilongjiang University of Science and Technology Graduate Innovation Research Foundation, Grant No. YJSCX2020-223HKD.

Acknowledgments: The authors are thankful to the reviewers for their comments and suggestions to improve the quality of the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hussain, I.; Shah, T. Literature survey on nonlinear components and chaotic nonlinear components of block ciphers. *Nonlinear Dyn.* **2013**, *74*, 869–904. [CrossRef] [CrossRef]
2. Lu, J.Q.; Seo, H. An investigation of an S-box mechanism in modern block cipher design. In Proceedings of the TENCON 2017—2017 IEEE Region 10 Conference, Penang, Malaysia, 5–8 November 2017; pp. 145–152. [CrossRef]
3. Matheis, K.; Steinwandt, R.; Corona, A.S. Algebraic Properties of the Block Cipher DESL. *Symmetry Basel* **2019**, *11*, 1411. [CrossRef] [CrossRef]

4. Mohamed, K.; Pauzi, M.N.M.; Ali, F.H.H.M.; Ariffin, S. Study of S-box properties in block cipher. In Proceedings of the 2014 International Conference on Computer, Communications, and Control Technology (I4CT), Langkawi, Malaysia, 2–4 September 2014; pp. 362–366. [CrossRef]
5. Lu, J.Q.; Seo, H. A Key Selected S-Box Mechanism and Its Investigation in Modern Block Cipher Design. *Secur. Commun. Netw.* **2020**, *2020*, 1–26. [CrossRef] [CrossRef]
6. Tian, Y.; Lu, Z. Chaotic S-Box: Intertwining Logistic Map and Bacterial Foraging Optimization. *Math. Probl. Eng.* **2017**, *2017*, 1–11. [CrossRef] [CrossRef]
7. Liu, J.; Wei, B.; Wang, X. One AES S-box to increase complexity and its cryptanalysis. *J. Syst. Eng. Electron.* **2007**, *18*, 427–433. [CrossRef]
8. Cui, J.; Huang, L.; Zhong, H.; Chang, C.; Yang, W. An improved AES S-Box and its performance analysis. *Int. J. Innov. Comput. Inf. Control.* **2011**, *7*, 2291–2302.
9. Hussain, I.; Anees, A.; Al-Maadeed, T.A.; Mustafa, M.T. Construction of S-Box Based on Chaotic Map and Algebraic Structures. *Symmetry Basel* **2019**, *11*, 351. [CrossRef] [CrossRef]
10. Ozkaynak, F.; Celik, V.; Ozer, A.B. A new S-box construction method based on the fractional-order chaotic Chen system. *Signal Image Video Process.* **2017**, *11*, 659–664. [CrossRef] [CrossRef]
11. Lu, Q.; Zhu, C.X.; Wang, G.J. A Novel S-Box Design Algorithm Based on a New Compound Chaotic System. *Entropy* **2019**, *21*, 1004. [CrossRef] [CrossRef]
12. Liu, G.J.; Yang, W.W.; Liu, W.W.; Dai, Y.W. Designing S-boxes based on 3D four-wing autonomous chaotic system. *Nonlinear Dyn.* **2015**, *82*, 1867–1877. [CrossRef] [CrossRef]
13. Lambic, D. A novel method of S-box design based on discrete chaotic map. *Nonlinear Dyn.* **2017**, *87*, 2407–2413. [CrossRef] [CrossRef]
14. Belazi, A.; Abd El-Latif, A.A. A simple yet efficient S-box method based on chaotic sine map. *Optik* **2017**, *130*, 1438–1444. [CrossRef] [CrossRef]
15. Guesmi, R.; Ben Farah, M.A.; Kachouri, A.; Samet, M. A Novel Design of Chaos Based S-Boxes Using Genetic Algorithm Techniques. In 2014 Ieee/Acs 11th International Conference on Computer Systems and Applications (AICCSA), Doha, Qatar, 10–13 November 2014; pp. 678–684. [CrossRef]
16. Ahmed, H.A.; Zolkipli, M.F.; Ahmad, M. A novel efficient substitution-box design based on firefly algorithm and discrete chaotic map. *Neural Comput. Appl.* **2019**, *31*, 7201–7210. [CrossRef] [CrossRef]
17. Wang, Y.; Zhang, Z.Q.; Zhang, L.Y.; Feng, J.; Gao, J.; Lei, P. A genetic algorithm for constructing bijective substitution boxes with high nonlinearity. *Inf. Sci.* **2020**, *523*, 152–166. [CrossRef] [CrossRef]
18. Fabian, V. Simulated annealing simulated. *Comput. Math. Appl.* **1997**, *33*, 81–94. [CrossRef] [CrossRef]
19. Pendharkar, P.C. Exhaustive and heuristic search approaches for learning a software defect prediction model. *Eng. Appl. Artif. Intell.* **2010**, *23*, 34–40. [CrossRef] [CrossRef]
20. Amine, K. Multiobjective Simulated Annealing: Principles and Algorithm Variants. *Adv. Oper. Res.* **2019**, *2019*, 1–13. [CrossRef] [CrossRef]
21. Ozkaynak, F. On the effect of chaotic system in performance characteristics of chaos based s-box designs. *Phys. a-Stat. Mech. Its Appl.* **2020**, *550*, 124072. [CrossRef] [CrossRef]
22. Wang, J.; Lu, Y.; Ding, Q. The Design of S-box Based on Cascaded Integer Chaos Applied to Wireless Sensor Network. *Int. J. Future Gener. Commun. Netw.* **2016**, *9*, 97–106. [CrossRef] [CrossRef]
23. Siddique, N.; Adeli, H. Simulated Annealing, Its Variants and Engineering Applications. *Int. J. Artif. Intell. Tools* **2016**, *25*, 1630001. [CrossRef] [CrossRef]
24. Shao, W.; Guo, G.B. Multiple-Try Simulated Annealing Algorithm for Global Optimization. *Math. Probl. Eng.* **2018**, *2018*, 1–11. [CrossRef] [CrossRef]
25. Sari, Y.A.; Kumral, M. An improved meta-heuristic approach to extraction sequencing and block routing. *J. South. Afr. Inst. Min. Metall.* **2016**, *116*, 673–680. [CrossRef] [CrossRef]
26. Ahmad, M.; Bhatia, D.; Hassan, Y. A Novel Ant Colony Optimization Based Scheme for Substitution Box Design. *Procedia Comput. Sci.* **2015**, *57*, 572–580. [CrossRef] [CrossRef]
27. Wang, J.; Pan, B.; Tang, C.; Ding, Q. Construction Method and Performance Analysis of Chaotic S-Box Based on Fireworks Algorithm. *Int. J. Bifurc. Chaos* **2019**, *29*, 1950158. [CrossRef]
28. Zhang, W.G.; Pasalic, E. Highly Nonlinear Balanced S-Boxes With Good Differential Properties. *IEEE Trans. Inf. Theory* **2014**, *60*, 7970–7979. [CrossRef] [CrossRef]

29. Webster, A.F.; Tavares, S.E. On the Design of S-Boxes. In *Proceedings of Advances in Cryptology—CRYPTO '85 Proceedings*; Springer: Berlin/Heidelberg, Germany, 1986; pp. 523–534. [CrossRef]
30. Jakimoski, G.; Kocarev, L. Chaos and cryptography: Block encryption ciphers based on chaotic maps. *IEEE Trans. Circuits Syst. Fundam. Theory Appl.* **2001**, *48*, 163–169. [CrossRef] [CrossRef]