mathematics

MDPI

Article

Block Cipher Nonlinear Component Generation via Hybrid Pseudo-Random Binary Sequence for Image Encryption

Dania Saleem Malik [1], Tariq Shah [2], Sara Tehsin [3], Inzamam Mashood Nasir [3], Norma Latif Fitriyani [4,*] and Muhammad Syafrudin [4,*]

1 Department of Mathematics, HITEC University Taxila, Taxila 47080, Pakistan; dania.saleem@hitecuni.edu.pk
2 Department of Mathematics, Quaid-i-Azam University, Islamabad 45320, Pakistan; stariqshah@qau.edu.pk
3 Faculty of Informatics, Kaunas University of Technology, 51368 Kaunas, Lithuania; sara.tehsin@ktu.edu (S.T.); inzamam.nasir@ktu.edu (I.M.N.)
4 Department of Artificial Intelligence and Data Science, Sejong University, Seoul 05006, Republic of Korea
* Correspondence: norma@sejong.ac.kr (N.L.F.); udin@sejong.ac.kr (M.S.)

Abstract: To analyze the security of encryption, an effectual encryption scheme based on colored images utilizing the hybrid pseudo-random binary sequence (HPRBS) and substitution boxes, known as S-boxes, is proposed. The presented work aims to design S-boxes using pseudo-random binary numbers acquired by Linear Feedback Shift Registers (LFSRs) in combination with a modified quadratic chaotic map. Firstly, cryptographically robust S-boxes are constructed by using binary pseudo-random number sequences, and then the cryptographic properties of the presented S-boxes are tested. The suggested S-boxes showed good results. Secondly, an RGB image encryption algorithm utilizing sequences generated by modified quadratic chaotic maps and S-boxes is offered. The new color image encryption techniques comprise two steps, including a permutation and a substitution step. The key association with the content of the image is also addressed. This strategy can result in a "one-time pad" effect and make the algorithm resistant to chosen-plaintext attack (CPA). The proposed scheme has been confirmed to be more valuable than most of the existing schemes. S-boxes are analyzed by the nonlinearity test, bit independence criterion (BIC), linear and differential approximation probabilities (LPs; DPs), and Strict-Avalanche Criterion (SAC) tests. A comparison with different S-boxes presented in the literature is also carried out. The comparison shows encouraging results about the quality of the proposed box. From security and experimental outcomes, the effectiveness of the presented color image encryption technique is verified. The proposed scheme has evident efficiency benefits, which implies that the proposed colored encryption of the image scheme has better potential for application in encryption schemes in real-time.

Keywords: chaotic map; S-box; image encryption; LFSR; binary sequence

MSC: 68P25; 94A60; 14G50

Citation: Malik, D.S.; Shah, T.; Tehsin, S.; Nasir, I.M.; Fitriyani, N.L.; Syafrudin, M. Block Cipher Nonlinear Component Generation via Hybrid Pseudo-Random Binary Sequence for Image Encryption. *Mathematics* **2024**, *12*, 2302. https://doi.org/10.3390/math12152302

Academic Editor: Lingfeng Liu

Received: 14 June 2024
Revised: 17 July 2024
Accepted: 18 July 2024
Published: 23 July 2024

1. Introduction

In modern times, the public wishes to possess information secrets and hide from the populace. In the past, people used various techniques to keep information secrets from adversaries. The rulers, armed forces, and bureaucrats used vital coding methods for their sensitive material, which helped them in the transmission of their information to their militaries safely and securely. With the rapid development of civilization, it is very important to develop strategies for the preservation of information. Currently, data security has a prime significance. Hence, cryptography helps to resolve issues related to security and data hiding by converting it into an unreadable format [1].

Cryptosystems are generally classified into two main classes: block and stream cipher. Data that are transformed in the form of blocks (i.e., the input bits of m length are transformed to output bits of length n by use of block cipher) are known as a block cipher.

A stream cipher operates on a single bit at a time, which means that a single input bit is transformed into a single output bit [2]. Numerous encryption techniques have evolved with the passage of time for securing information of high intelligence value. Wireless communication, being more prone to theft, requires data security using advanced encryption techniques [3]. Block cipher-based cryptosystems are heavily dependent on S-boxes, and designers of cryptosystems focus on the cryptographically strong design of S-boxes.

For the bulk of data encryption techniques, Advance Encryption Standards (AESs) are utilized for encryption and decryption. Over time, AES superseded the Data Encryption Standards. In AES, the nonlinear transformation of the S-box (substitution box) is an essential constituent. The S-boxes' strength plays a vital role in the algorithm's security. Thus, researchers spend a lot of time on the improvement of S-box strength [4]. Most of the time spent on construction and analysis is devoted to S-box construction, as it only represents the nonlinear component of the technique. Hence, every weakness in the S-box can be intercepted easily in a cryptosystem [5]. For this purpose, many techniques for the construction of S-boxes have been proposed by many researchers. Since nonlinear systems demonstrate randomness, in cryptography, chaos plays a great role. Because chaos helps in the generation of many sequences of pseudo-random numbers, it is utilized in nonlinear components of encryption construction. Many chaos-based cryptosystems have been reported in recent years because of the existence of a relationship between chaotic and cryptosystem properties [6–8].

Some traditional techniques for encryption (like AES, DES, single one-dimensional chaos, etc.) are not suitable for the encryption of images as explained. So, it is necessary to devise strategies that secure image information. In view of this, many block-cipher-based encryption techniques that utilize random numbers generated by chaos and offer high security are presented [9,10]. The prime goals of this work are as follows:

- In many cryptographic applications, random numbers are needed for strong cryptosystem designs. In this context, the proposed work is based on a Hybrid Binary Pseudo-Random Number Generator (HPRNG) derived from a feedback shift register known as a Linear Feedback Shift Register (abbreviated as LFSR) and a modified chaotic quadratic map. Since the LFSR-based PRNGs are not resistant to attacks and reveal information about keys to overcome this flaw, the binary stream of random numbers obtained from LFSR is XORed with the random stream of the modified quadratic chaotic map.
- This above-mentioned technique helps in the elimination of the LFSR linearity property and helps in hiding characteristic statistical patterns of binary sequences generated by chaotic maps that are used in estimating their initial conditions. The binary stream generated by exclusive or operation between the LFSR and chaos helps to utilize all the benefits of chaotic maps and LFSRs and avoid all of their drawbacks.
- As cryptographically strong substitution boxes (S-boxes) have important features like nonlinearity, S-box generation using nonlinear sequences is addressed in this work. To design an S-box, a binary stream of numbers generated by HPRNGs is used. After conversion of bits to bytes, an S-box is achieved. Then, the strength of the attained S-box is analyzed via nonlinearity (NL), the Strict-Avalanche Criterion (SAC), bit independence criteria (BIC), and differential and linear approximation probability (DP; LP), which ensures that the S-box is cryptographically strong and has high performance.

To contribute to the study of cryptosystem design, this paper depicts a technique to design S-boxes based on pseudo-binary bit streams of numbers and its application in encryption schemes. The binary bit streams of random numbers were generated via hybrid-modified chaos and feedback shift registers. The S-boxes constructed using this scheme showed excellent properties, which indicates that they can be valuable in cryptosystems. In developing a novel image encryption algorithm, first, we use random numbers to scramble the pixels of an image and then the S-boxes are substituted to create confusion. The robustness of the proposed color image encryption technique is examined by the comparison of several quality measures of the image with RGB-based encryption utilizing chaos designs.

1.1. Linear Feedback Shift Registers (LFSRs)

In digital circuits, shift registers are a kind of logic circuit, compiled in a linear mode whose inputs are connected to the output in such a manner that by triggering a circuit, the data are moved along the line. LFSR is a shift register whose input bits are the linear function of more than two preceding states. An n-stage LFSR has an n-length numbered as $\{0, 1, 2, \ldots\ldots, n-1\}$, where each has the ability to store a single bit and a clock is used to control the shuffling of data. A shift register is initialized by vectors with entries $w_0, w_1, w_2, \ldots\ldots, w_{n-1}$. The operations used in LFSR are as follows:

- w_j (the zero-stage entry) constitutes the output part.
- The entry of ℓ-stage is shifted to the $l-1$ stage, for $1 \leq \ell \leq n-1$.
- The new entry of the $n-1$ stage is obtained by a subset of n-stage entry using Xor.

The LFSR starting input bit value is known as the seed. LFSR's well-defined seed and feedback function generates bit sequences of random numbers with a large period value. If the input bit value in LFSR consists of only zeros, then the registers would stop working and the output is zero. Every starting state (except zero) produces a periodic state of the sequence with a period $(2^n - 1)$ [11,12].

1.2. Modified Quadratic Chaotic Map

The chaotic Modified Quadratic map is depicted below.

$$\mathcal{X}_{i+1} = \left(R + (1 - 2\mathcal{X}_i)^2\right) mod\ 1 \quad (1)$$

This map has a state variable $\mathcal{X}$ and parameter R. The parametric value R shows chaotic behavior in these intervals [0, 0.14], [1.56, 2.14], [2.56, 3.14], $\ldots infinity$ [13]. A simple modification of this chaotic map is used in the proposed work for generating chaotic binary numbers as follows:

$$\mathcal{X}_{i+1} = \left(R + (1 - 2\mathcal{X}_i)^2\right) mod\ 1. \quad (2)$$

$$\mathcal{Z}_{i+1} = \left(P + (1 - 2\mathcal{Z}_i)^2\right) mod\ 1. \quad (3)$$

where $\mathcal{X}, \mathcal{Z}$ are state variables $\mathcal{X}(0) \neq \mathcal{Z}(0)$ and R, P are parameters and used as secret keys. Improved quadratic chaotic map bifurcation is illustrated in Figure 1.

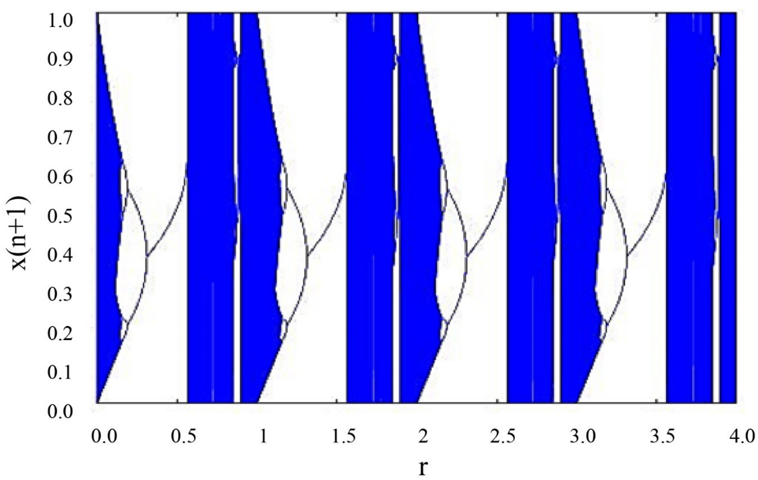

Figure 1. Bifurcation chaotic map diagram. The blue color is the chaotic behaviour of map fluctuating with parametric values.

2. Binary Sequence Generation Using Modified Quadratic Map

This section presents a technique for generating pseudo-random binary sequences using a threshold simple function that is combined with real numbers of modified quadratic chaotic maps. The following steps are used for this purpose, as illustrated in Figure 2:

- First, the initial values and parameters $\{\mathcal{X}(0), \mathcal{Z}(0),\ R(0),\ P(0)\}$ are determined from Equation (1), given in Section 1.2.
- The modified quadratic equations are iterated $\mathcal{K}$ and $\mathcal{L}$ times, respectively, where $\mathcal{K}$ and $\mathcal{L}$ are different constant values.
- By iterating Equation (1), two sequences (decimal) $\mathcal{X}(i), \mathcal{Z}(i)$ are generated using the following formulas:

$$\mathcal{X}(i) = abs(mod(floor(\mathcal{X} \times 100000000000000)), 2)) \quad (4)$$

$$\mathcal{Z}(i) = abs(mod(floor(\mathcal{Z} \times 100000000000000)), 2)) \quad (5)$$

  where the $floor$ converts the value of $\mathcal{X}$ to the closest integer equal of less than $\mathcal{X}$, $mod\ (\mathcal{X}, \mathcal{Z})$ is used to provide the reminder value after the division, and $abs(\mathcal{X})$ is used to give the absolute $\mathcal{X}$ value.
- The following threshold function (W) can be applied:

$$W(i) = \begin{cases} 1, & if\ \ X(i) > Z(i) \\ 0, & otherwise \end{cases} \quad (6)$$

  After that, a bit stream of pseudo-random numbers is obtained.
- The process is repeated until the desired bit stream of pseudo-random numbers W' is obtained.

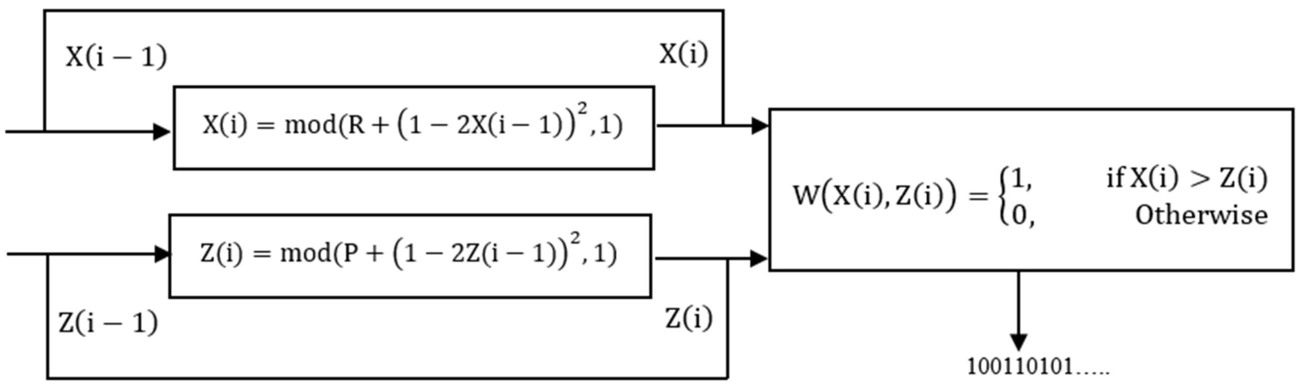

Figure 2. PRBS flow diagram.

2.1. Hybrid Pseudo-Random Binary Sequence Generation

The design of the HPRBS is based on an LFSR of length 128 bits and modified quadratic chaotic maps since many PRNG-based LFSRs are not resistant to attacks and provide information about the secret key. This flaw is overcome by using the XOR operation in which random bits obtained by a modified quadratic map binary stream (Section 1.2) are Xored with LFSR feedback in every clock cycle to generate the random binary number stream S. The LFSR tap positions are decided by the primitive polynomial $q^{128} + q^{127} + q^{126} + q^{121} + 1$. The total keys required for HPRBS are $\{\mathcal{X}(0), \mathcal{Z}(0),\ R(0),\ P(0), f\}$, where f denotes the input vector in LFSR. The key space for the LFSR of 128 bits is $2^{128} - 1 = 429, 4967, 295$. The flow diagram of Figure 3 demonstrates the whole process of HPRBS.

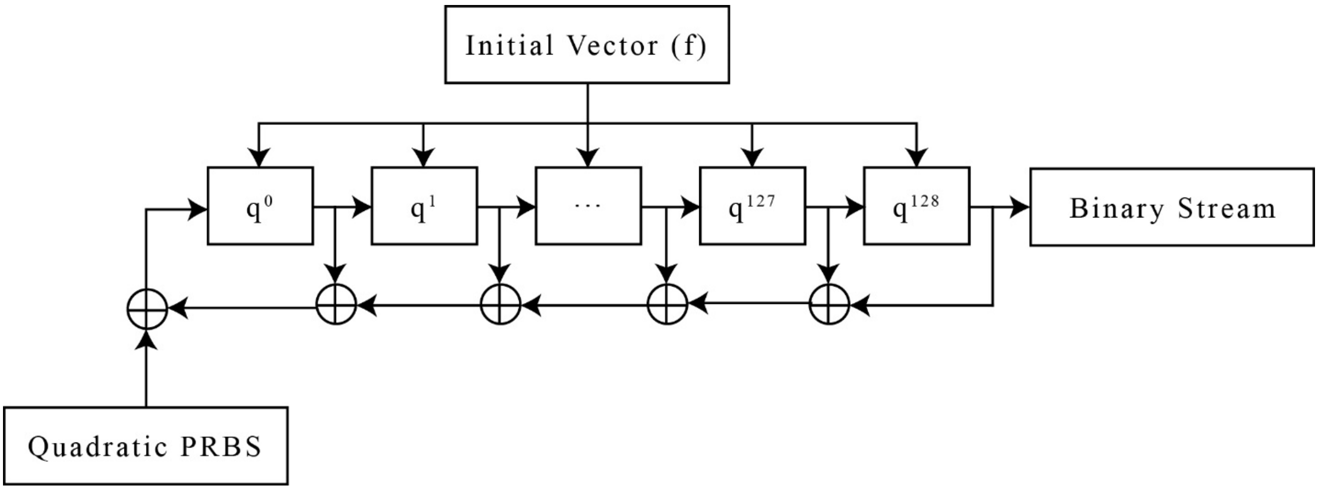

Figure 3. HPRBS flow diagram.

3. S-Box Construction Using Hybrid Pseudo-Random Binary Sequences

Simple steps are used for the construction of cryptographically secure S-boxes, as shown below. The bit stream $S = \{S_0, S_1, S_2, S_3, \ldots\ldots\}$ generated in Section 2.1 is then employed in the formation of S-boxes as follows:

- Step–1: Each block sequence consisting of $\mathcal{N}$-bits is generated as follows:

$$\begin{gathered} B_0 = \{S_0, S_1, S_2, \ldots\ldots S_{\mathcal{N}-1}\} \\ B_1 = \{S_{\mathcal{N}}, S_{\mathcal{N}+1}, S_{\mathcal{N}+2}, \ldots\ldots S_{2\mathcal{N}-1}\}, \\ B_2 = \{S_{2\mathcal{N}}, S_{2\mathcal{N}+1}, S_{2\mathcal{N}+2}, \ldots\ldots S_{3\mathcal{N}-1}\}, \\ \vdots \\ B_k = \left\{S_{k\mathcal{N}}, S_{k\mathcal{N}+1}, S_{k\mathcal{N}+2}, \ldots\ldots S_{(k+1)\mathcal{N}-1}\right\} \end{gathered} \tag{7}$$

- Step–2: Now, each $\mathcal{N}$-bit block, i.e., $B_0, B_1, B_2, \ldots\ldots, B_k$, is converted to integer numbers as $C_0, C_1, C_2, \ldots\ldots, C_k$.
- Step–3: To obtain distinct 2^n values, the repeated numbers are removed.
- Step–4: Create S-boxes.
- Step–5: After one S-box, consider other blocks of N-bits and then repeat the whole process to generate the other two S-boxes. In the proposed work, $\mathcal{N} = 8$ to generate 8×8 bit-based S-boxes. Tables 1–3 provides the S-boxes.

Table 1. S-box 1.

21	207	120	241	47	146	206	76	169	119	99	128	232	244	36	72
37	157	237	73	110	126	132	43	153	131	71	181	177	8	101	90
11	138	186	38	173	16	54	79	56	44	171	7	188	234	143	175
45	172	20	65	22	155	125	180	198	102	60	142	130	189	95	254
116	123	229	34	243	176	174	84	227	28	24	178	32	210	27	225
204	26	255	98	213	164	183	18	58	167	31	88	194	246	92	52
165	2	74	29	212	149	203	182	159	158	145	35	87	115	89	163
139	91	216	231	69	12	147	230	40	166	17	190	62	152	113	236
151	220	1	245	135	148	242	222	109	19	85	122	223	215	195	136
238	41	168	240	5	209	61	51	193	127	160	75	196	179	39	156
185	208	48	80	170	224	121	53	141	133	83	9	154	134	59	205

Table 1. *Cont.*

64	97	46	249	13	100	226	25	250	82	105	78	235	33	117	93
217	251	111	140	104	6	10	30	191	57	248	144	70	103	106	253
150	187	112	49	94	4	201	15	68	86	42	161	211	218	66	0
77	96	108	118	23	247	219	202	192	124	107	63	129	214	233	162
81	114	55	197	199	67	50	184	252	3	200	14	228	239	137	221

Table 2. S-box 2.

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

Table 3. S-box 3.

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

4. S-Box Algebraic Analysis

This section proposed the evaluation of S-boxes. The assessment of S-boxes ensures the efficiency and ability to create misperception in any cipher. For S-box algebraic property testing, the analysis used is nonlinearity (NL), Strict-Avalanche Criterion (SAC), differential approximation probability (DP), linear-approximation probability (LP), and bit independence criterion (BIC). It was noted from the analysis that the suggested S-boxes achieved almost all conditions close to the ideal result. Also, the comparison of proposed S-boxes produced by different schemes is presented.

4.1. Nonlinearity

Nonlinearity in the Hamming distance term is referred to as the minimum value among Boolean functions and possibly all affine functions. Bits required modifications in their configuration to attain the Boolean function adjacent affine function. The nonlinearity technique enumerates the alternate bit number to make the function closer to an affine function. In cryptographic literature, for S-boxes in the case of the Galois field $GF(2^n)$, the upper bound of nonlinearity is given as $2^{n-1} - 2^{n/2-1}$ [14]. The proposed S-box NL calculations are given in Table 4. The presented analysis reveals that the produced S-boxes could replace the algebraic constructed S-boxes because their construction is appealing and based on random numbers generated by hybrid chaos and feedback registers that create extensive randomness.

Table 4. Performance indexes of *Sbox* 1, *Sbox* 2 and *Sbox* 3.

Analyses	Maximum	Minimum	Average	Square Deviation	Approximated DP	Approximated LP
NL						
Sbox 1	106	100	104	-	-	-
Sbox 2	108	103	105.5			
Sbox 3	110	98	104			
SAC						
Sbox 1	0.625	0.4218	0.502	0.044	-	-
Sbox 2	0.625	0.421	0.502	0.015		
Sbox 3	0.421	0.422	0.502	0.021		
BIC						
Sbox 1	-	95	105.2	0.903	-	-
Sbox 2		110	103.2	3.216		
Sbox 3		94	104.2	2.252		
BIC − SAC						
Sbox 1	-	0.474	0.502	0.0132	-	-
Sbox 2		0.480	0.503	0.0142		
Sbox 3		0.472	0.500	0.0139		
DP						
Sbox 1	-	-	-	-	0.0340	
Sbox 2					0.0312	
Sbox 3					0.0457	
LP						
Sbox 1	158	-	-	-	-	0.070
Sbox 2	150					0.013
Sbox 3	158					0.012

4.2. Strict Avalanche Criteria (SAC)

In 1986, Tavares and Webster established criteria for a strict avalanche to analyze the strength of S-boxes. A function has an SAC value if by affecting a particular input bit, it has every output bit alternating with 0.5 probability. The results are outlined in Table 4. From the results, it can be clearly seen that the average SAC value of S-boxes is near to the ideal value of 0.5, which verifies S-boxes' SAC property fulfillment.

4.3. Bit Independent Criterion (BIC)

Webster and Tavares presented the bit independence criterion [15]. In this criterion, the variables are pairwise related to collect information about the independence of such variables. In this technique, for the analysis of independent variables, the output vectors are used, and input bits are tackled separately. The bit independence criterion is a highly recommended property in cryptographic structures because the increased independence among the bits creates more confusion in recognizing the design of the structure. The results are displayed in Table 4, which suggests that the S-boxes satisfy the BIC.

4.4. Linear Approximation Probability (LP)

The analysis of the imbalance maximum event value is termed linear approximation probability. This technique applies two masks, i.e., A_x and A_y, over input and output bit parity. In [16], the linear approximation probability of the S-box is given as follows:

$$LP = \max_{A_x, A_y \neq 0} \frac{\{x \in \mathscr{G} : x.A_x = S(x).A_y\}}{2^n} - \frac{1}{2} \tag{8}$$

Here, the mask input and output parity bits $(A_x,\ A_y)$ are represented by the set "$\mathscr{G}$", which consists of all inputs, and 2^n is the representation of the $\mathscr{G}$ element number. From the analysis of LP for the synthesized S-boxes, it can be observed that the S-boxes have an average LP value of 0.0343, which ensures their resistance against attacks.

4.5. Differential Approximation Probability (DP)

In an encryption process, the S-box is the nonlinear component with context-uniform differentiability in unique situations. Mathematically, differential approximation probability is given as follows:

$$DP\ (\Delta\jmath \to \Delta t) = \frac{\{\jmath \in S/\Omega(\jmath) \oplus \Omega(\jmath \oplus \Delta\jmath) = \Delta t\}}{2^m} \tag{9}$$

This means a differential input should be uniquely mapped on differential outputs, which ensures uniform mapping probability for each $\imath$ [17]. Less DP values provide more resistance to attacks (differential attacks).

All three suggested S-boxes catalogs are displayed in Table 4, and the results with other S-boxes are compared in Table 5.

Table 5. Proposed S-box comparison.

S−Boxes	Nonlinearity	SAC	BIC	DP	LP
AES	112.0	0.5058	112.0	0.016	0.062
APA	112.0	0.4987	112.0	0.016	0.062
Gray	112.0	0.5058	112.0	0.016	0.062
Skipjack	105.7	0.4980	104.1	0.047	0.109
Xyi	105.0	0.5048	103.7	0.047	0.156

Table 5. *Cont.*

ResiduePrime	99.5	0.5012	101.7	0.281	0.132
[18]	103.3	0.5000	104.0	0.047	0.133
[19]	105.5	0.4990	106.0	0.125	0.133
[20]	106.5	0.4950	103.8	0.039	0.141
[21]	104.5	0.4980	104.6	0.047	0.125
[22]	105.5	0.5000	103.8	0.047	0.125
[23]	111.75	0.4978	103.86	0.039	0.125
[24]	107	0.493	102.3	0.047	0.141
Proposed					
S-box 1	104.0	0.4940	105.2	0.032	0.016
S-box 2	105.5	0.5020	103.2	0.031	0.013
S-box 3	104.0	0.5020	104.2	0.032	0.012

4.6. S-Box Comparison

For the testing of the cryptographic proposed S-box performance, extensively used performance criteria for S-boxes are employed. Additionally, a comparison between cryptographic proposed S-box performance with recently suggested S-boxes is made and the results are shown in Table 5; in the criteria of evaluated performance, the ideal value of BIC-SAC and SAC is 0.5. The greater value of nonlinearity indicates S-boxes' better performance and resistance against attacks (cryptanalysis). For better resistance against differential and linear cryptanalysis LP, DP values for S-boxes must be smaller. From the results of Table 5, the suggested S-boxes have lower DP and LP values than the majority of the proposed schemes like [18–24], which means that the S-boxes of the presented scheme have strong robustness against cryptanalysis attacks (linear and differential). S-box nonlinearity is also higher compared to many others. Table 5 also suggests that the BIC-SAC and SAC values of proposed S-boxes are close to the ideal SAC value.

5. Image Encryption Scheme

For the elimination of insecurities in S-box-based image encryption schemes, a novel encryption technique for S-boxes utilizing the pseudo-random binary stream of numbers is offered. Firstly, a new technique is used to generate S-box and binary pseudo-random number sequences, and then for diffusion and confusion, the processes used include the substitution and permutation process of the presented scheme. The flow diagram of the proposed scheme is shown in Figure 4.

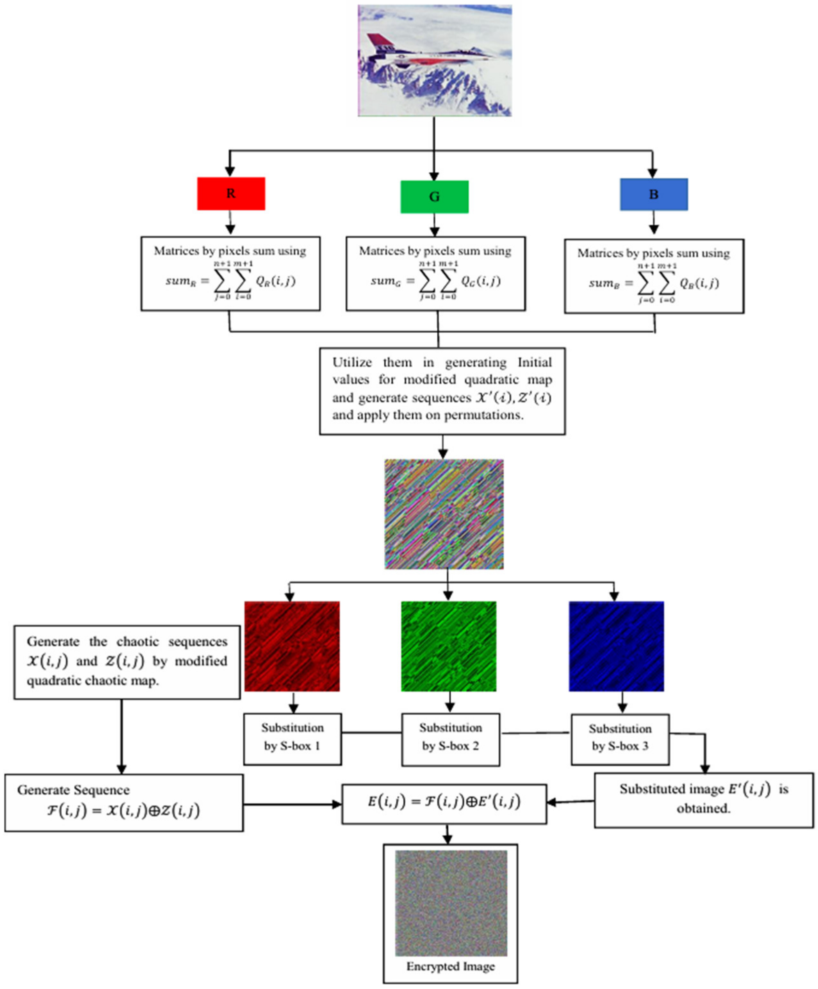

Figure 4. Flow diagram of encryption.

5.1. Permutation Process

The process of permutation involves the following steps:

1. First, take an original color image Q of length $m \times n \times 3$ and associate this image with some random security keys taken as $K = (K_1, K_2, K_3, \ldots\ldots, K_9)$, whose values lie between 0 and 1, which are then used as initial values and parameters of the modified quadratic chaotic map for Equation (1) given in Section 1.2.
2. Apply *sum* between the pixel values of matrices Q_R, Q_G, Q_B as follows:

$$sum_R = \sum_{j=0}^{n+1} \sum_{i=0}^{m+1} Q_R(i,j) \tag{10}$$

$$sum_G = \sum_{j=0}^{n+1} \sum_{i=0}^{m+1} Q_G(i,j) \tag{11}$$

$$sum_B = \sum_{j=0}^{n+1} \sum_{i=0}^{m+1} Q_B(i,j) \tag{12}$$

where Q_R, Q_G, Q_B are the matrices of red, green, and blue channels of the original color image Q.

3. Now, the initial and parametric values of the modified quadratic chaotic map are generated by using the following formula:

$$\mathscr{X}(0) = mod((K_1 \times \log(sum_R) + K_2 \times \log(sum_G) + K_3 \times \log(sum_B)), 256) \quad (13)$$

$$R(0) = mod((X(0) + 1) \times (K_4 \times K_5 \times K_6)), 1) \quad (14)$$

$$\mathscr{Z}(0) = mod((K_7 \times \log(sum_R) + K_8 \times \log(sum_G) + K_9 \times \log(sum_B)), 256) \quad (15)$$

$$P(0) = mod((Y(0) + 1) \times (K_4 \times K_5 \times K_6)), 1) \quad (16)$$

$mod(u, 1)$ or $mod(u, 256)$ denotes the decimal fraction value of u.

4. Iterate the modified quadratic map given in Equation (1) by using the initial values obtained in Step 3. Hence, new sequences $\mathscr{X}'(\imath)$, $\mathscr{Z}'(\imath)$ are generated. Now, the image pixels are permuted row and column-wise utilizing sequences $\mathscr{X}'(\imath), \mathscr{Z}'(\imath)$. Then, the permuted image $\mathscr{I}' = \{\mathscr{I}'(1), \mathscr{I}'(2), \mathscr{I}'(3), \ldots\ldots \mathscr{I}'(H)\}$, $H = M \times N$ is obtained.

5.2. Substitution Process

For the encrypted final image E, substitution is achieved.

Step 1: Firstly, the binary pseudo-random number sequence W' obtained in Section 2 is converted to a key sequence with integer 8-bit values using the following formula:

$$W'(\imath) = mod\Big(floor(W'(\imath) \times 10000000000000), 2^8\Big) \quad (17)$$

Step 2: For the substitution process, the key W', S_{box1} and permuted image $\mathscr{I}'$ are used. First, convert the permuted image into three layers $\mathscr{I}'_R$, $\mathscr{I}'_G, \mathscr{I}'_B$. Substitute every pixel of a permuted image $\mathscr{I}'_R$ with S_{box1} and the sequence W'. The substitution is achieved using the following formula for the first pixel:

$$\begin{cases} j = mod[(1 + \mathscr{I}'_R(2)), H] + 1 \\ E(1) = sub\_byte[S_{box1}, v_0 \oplus \mathscr{I}'_R(1)] \oplus w'(j) \end{cases} \quad (18)$$

where v_0 is any number in $\{1, 2 \ldots 255\}$. For the i^{th} pixel, it is calculated as follows:

$$\begin{cases} j = mod[(i + \mathscr{I}'_R(i+1)), H] + 1 \\ E(i) = sub\_byte[S_{box1}, E(i-1) \oplus \mathscr{I}'_R(i)] \oplus w'(j) \\ i = (1, 2, 3, \ldots\ldots H-1) \end{cases} \quad (19)$$

For the last H^{th} pixel, it is calculated as follows:

$$\begin{cases} j = mod[(H + v_0 + vl), H] + 1 \\ E(H) = sub\_byte[S_{box1}, E(H-1) \oplus \mathscr{I}'_R(H)] \oplus w'(j) \end{cases} \quad (20)$$

where vl belongs *to* $\{1, 2, 3, \ldots 255\}, sub_{byte}[S_{box1}, \mathscr{X}]$, which is used for byte substitution for $\mathscr{X}$ using S-box1. Hence, we obtain the final encrypted image for the red layer $E_R(i, j)$. We can repeat the process by using S_{box2}, S_{box3} for $\mathscr{I}'_G$, $\mathscr{I}'_B$.

Step 3: Lastly, for diffusion, again an iteration is performed $\mathscr{L}$ times in order to generate sequences $\mathscr{X}$ and $\mathscr{Z}$ in such a way that after every iteration, the initial conditions are altered. New initial $\mathscr{X}$ and $\mathscr{Z}$ values are calculated by the following equations:

$$\mathscr{X}'(0) = \mathscr{X}_{\mathscr{L}} \oplus \mathscr{X}_{\mathscr{L}-1} \oplus \mathscr{X}_{\mathscr{L}-2} \quad (21)$$

$$\mathscr{Z}'(0) = \mathscr{Z}_{\mathscr{L}} \oplus \mathscr{Z}_{\mathscr{L}-1} \oplus \mathscr{Z}_{\mathscr{L}-2} \quad (22)$$

Here, $\mathscr{X}'$, $\mathscr{Z}'$ represents the initial values and $\mathscr{X}_{\mathscr{L}}, \mathscr{X}_{\mathscr{L}-1}, \mathscr{X}_{\mathscr{L}-2}$, $\mathscr{Z}_{\mathscr{L}}, \mathscr{Z}_{\mathscr{L}-1}, \mathscr{Z}_{\mathscr{L}-2}$ are output values after iterating them $\mathscr{L}, \mathscr{L}-1$, $\mathscr{L}-2$ times. We can restrict these chaotic sequences $\mathscr{X}(i,j)$ and $\mathscr{Z}(i,j)$ in the 0–255 range employing the following formulas:

$$\mathscr{X}(i,j) = mod\Big(\Big(\mathscr{X}(i,j) \times 10^{16}\Big), 256\Big) \tag{23}$$

$$\mathscr{Z}(i,j) = mod\Big(\Big(\mathscr{Z}(i,j) \times 10^{16}\Big), 256\Big) \tag{24}$$

Finally, the sequence $\mathscr{F}(i,j)$ is generated, which is obtained by the following equation:

$$\mathscr{F}(i,j) = \mathscr{X}(i,j) \oplus \mathscr{Z}(i,j) \tag{25}$$

Diffusion is carried out via exclusive OR operation between the pixels of the substituted image and key. The formula for this step is given as follows:

$$E(i,j) = \mathscr{F}(i,j) \oplus E'(i,j) \tag{26}$$

where $E'(i,j)$ is the substituted image and $\oplus$ represents exclusive OR operation and $E(i,j)$ is the final ciphered image.

Decryption is the reverse of the encryption scheme. For recovery of the plain image from the ciphered image, the following steps are taken.

Firstly, the ciphered image is diffused using sequence $\mathscr{F}(i,j)$ by using the following formula:

$$E\prime(i,j) = \mathscr{F}(i,j) \oplus E(i,j) \tag{27}$$

where the $\mathscr{F}(i,j)$ sequence is obtained by using the same formulation used in the encryption technique.

Decryption is the reverse of the encryption scheme. For recovery of the plain image from the ciphered image, the following steps are taken:

First, the substitution process is carried out.

For the last red-layer H^{th} pixel, it is calculated as follows:

$$\begin{cases} j = mod[(H + v_0 + vl), H] + 1 \\ \mathscr{I}'_R(H) = sub\_byte\_1[\boldsymbol{S_{box1}}, E(H) \oplus w'(j)] \oplus E(H-1) \end{cases} \tag{28}$$

where v_0, vl belongs to $\{1,2,3,\ldots 255\}$, $sub\_byte\_1[\boldsymbol{S_{box1}}, \mathscr{X}]$ used for inverse byte substitution for $\mathscr{X}$ using S-box1.

For i^{th} red image pixel, it is calculated as follows:

$$\begin{cases} j = mod[(i + \mathscr{I}'_R(i+1)), H] + 1 \\ \mathscr{I}'_R(i) = sub\_byte\_1[\boldsymbol{S_{box1}}, E(i) \oplus w'(j)] \oplus E(i-1) \\ i = (1,2,3,\ldots\ldots H-1) \end{cases} \tag{29}$$

For the red image's first pixel, it is calculated as follows:

$$\begin{cases} j = mod[(1 + \mathscr{I}'_R(2)), H] + 1 \\ \mathscr{I}'_R(1) = sub\_byte\_1[\boldsymbol{S_{box1}}, E(1) \oplus w'(j)] \oplus v_0 \end{cases} \tag{30}$$

The same steps are performed for the blue and green layers. The permutation steps are now performed. For permutation, the same keys $\mathscr{X}'(i)$ and $\mathscr{Z}'(i)$ generated in Section 5.1 for the encryption process are utilized. In row-wise permutation, the reverse steps are taken for to move permuted pixels back to the original move sequence $\mathscr{X}'(i)$ and for column permutation, the reverse steps are taken to move column-permuted pixels back to the original utilizing sequence $\mathscr{Z}'(i)$. The proposed scheme's original and encrypted images of Lena are shown in Figure 5, where it can be clearly seen that the encrypted image is completely different from the original image. The original and encrypted images combine and each R, G, and B layer representation of Lena's image is also illustrated in the figure.

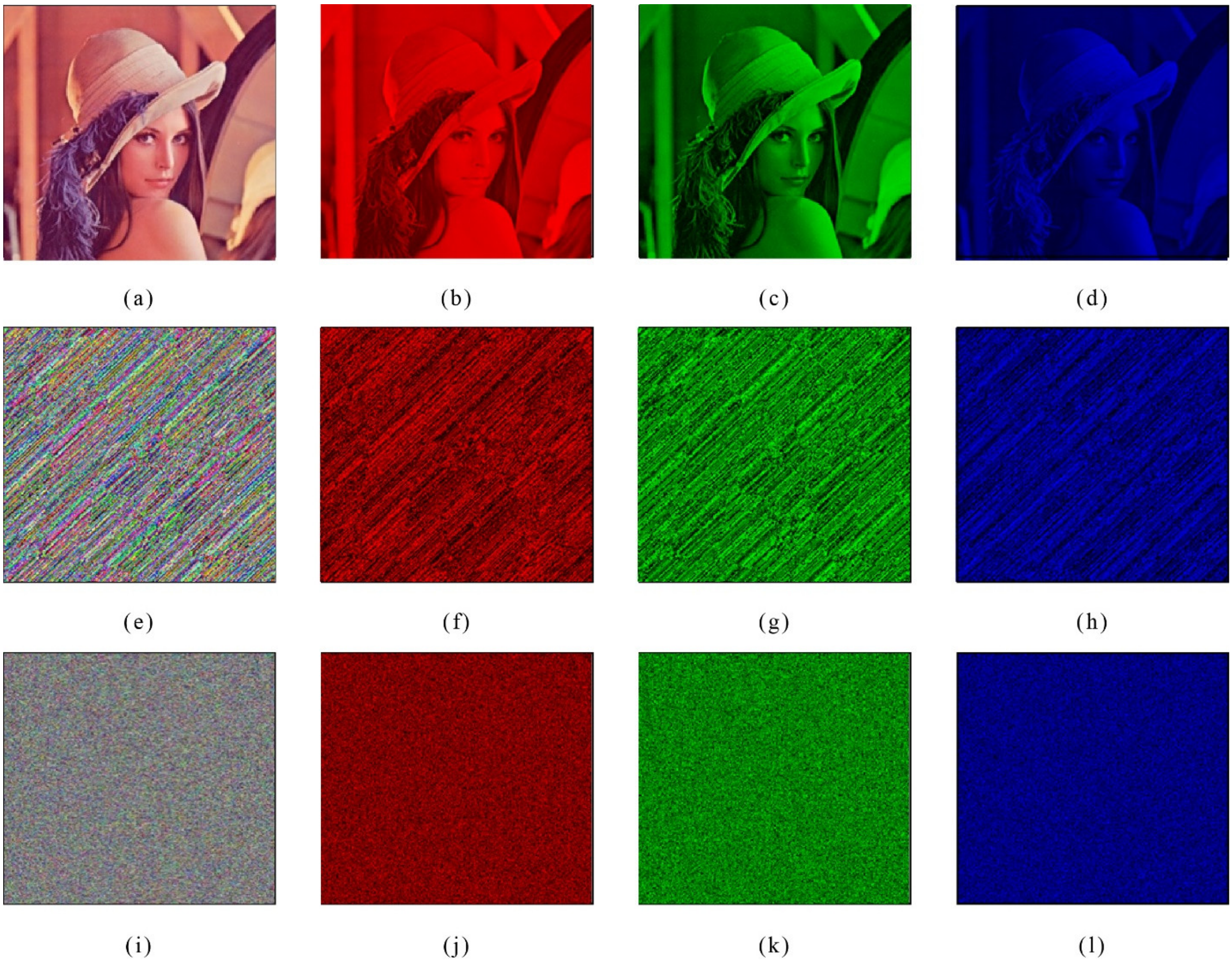

Figure 5. Lena (**a**); original image (**b**); image red layer (**c**); image green layer (**d**); image blue layer (**e**); permuted image (**f**); red-layer permuted image (**g**); green-layer permuted image (**h**); blue-layer permuted image (**i**); encrypted image (**j**); red layer of encrypted image (**k**); green layer of encrypted image (**l**); blue layer of the original image.

6. Security Analysis

To analyze the suggested encryption outline strength, experiments to check security were performed on color images of different sizes taken from the USC-SIPI database. The standard analysis techniques used to examine encrypted images include histograms, entropy, pixel adjacent correlation, UACI, and NPCR analysis. The proposed scheme simulations were conducted via 9.1.0.441655 (R2016b) MATLAB. The initial and parametric values of the modified quadratic map were chosen as $\mathscr{X}(0) = 0.02001$, $\mathscr{Z}(0) = 0.03$, $P(0) = 3.15$, $R(0) = 1.60$, $vl = 56$, $v_0 = 234$ and the inputted random keys were taken as $K_1 = 0.253496532144$, $K_2 = 0.4673213337632$, $K_3 = 0.100643892652$, $K_4 = 0.564329647734$, $K_5 = 0.472931065490$, $K_6 = 0.689546143287$, $K_7 = 0.6257742381092$, $K_8 = 0.734518093476$; $K_9 = 0.779834561276$.

6.1. Entropy

Entropy is a crucial factor in demonstrating randomness. For the calculation of an image, the following entropy formula is used:

$$\mathscr{H}(m) = -\sum_{u=0}^{2^n-1} \mathrm{P}(m_u)\log_2[\mathrm{P}(m_u)] \tag{31}$$

Here, the probability grey-level u occurrence is expressed by $\mathrm{P}(m_u), u = \{0,1,2,\ldots\ldots 2^n\}$ and 2^n is the level number of the greyscale image. If the probability of occurrence of each m_u in the image is the same, then its probability is given as $\mathrm{P}(m_u) = \frac{1}{2^n}$. Hence, the image signifies complete randomness by $\mathscr{H}(m) = n$. An encryption scheme with entropy values close to 8 is highly resistant to attacks [25]. The entropy results of the proposed encrypted image and the comparison with [26–29] are presented in Table 6. Entropy values for each layer of the proposed image are close to the optimal value. Also, the comparison shows that the proposed scheme achieved better results as compared to others.

Table 6. Proposed scheme entropy analysis and comparison.

	Images	**R**	**G**	**B**	**Average**
Proposed	Lena	7.9994	7.9995	7.9994	7.9994
	Aeroplane	7.9995	7.9989	7.9995	7.9993
	Peppers	7.9988	7.9991	7.9988	7.9989
	Baboon	7.9991	7.9990	7.9990	7.9990
	House	7.9992	7.9992	7.9988	7.9990
[26]	Lena	7.99614	7.99408	7.99686	7.99569
[27]	Lena	7.99730	7.99690	7.99710	7.99710
[28]	Lena	-	-	-	7.9975
[29]	Baboon	-	-	-	7.9994

6.2. Analysis of Key

The total number of keys employed in the encoding techniques is considered as key space. The proposed scheme has nine random keys given in Section 6, and $\{\mathscr{X}, \mathscr{Z}, P, R\}$ are used. The computational accuracy is 10^{14}, so the total number of keys is $(10^{14})^{13} = 10^{182}$, which is enough for resistance against attacks.

6.3. Key Sensitivity Analysis

For any cryptosystem, security analysis of key sensitivity is very important in order to investigate the robustness. It helps to identify cryptosystem security against attacks like brute force. Key sensitivity plays a vital role in cryptosystem security. Cryptosystems with a high level of sensitivity offer security for similar plain images, as the ciphered images are completely altered by slight changes in key pairs if an attacker is trying to hack. The first round of encryption is performed by a set of initial keys given in Section 6.2. In the presentation of this article security, four rounds are performed by an alteration of small changes in the initial key values, while the other values remain unchanged. For instance, for a sensitivity test performed on the original image of an aeroplane with a size of 512 × 512, encryption of this image is first performed by the original initial key set and it is shown in Figure 6a,b. A small alteration is made in just a single initial key from the key set say in $\mathscr{X}$ like $\mathscr{X} = \mathscr{X} + 10^{-15}$, and the other keys remain unchanged. This helps in obtaining a new initial key set, which is then used for encryption of the same plain image with different keys, and then another ciphered image is obtained as shown in Figure 6c. The difference between the pixels of both ciphered images is given in Figure 6d. The value of the difference rate between them is 0.996423%, which represents that any slight modification in keys will result in significant alteration in ciphered images. The result of decryption for images by both key sets is shown in Figure 6e–h. In Figure 6e, the image is decrypted by the original key set values, while in Figure 6f,g, it is decrypted with slightly changed key set values. It is depicted in Figure 6 that the original plain image is just obtained from the original set of keys with which they are encrypted, while the slightly altered keys on encrypted images do not provide any information about a plain image. Thus, from the results, it can be seen that the proposed method shows high sensitivity towards secret keys in both enciphering and deciphering. Due to the length limitations of the article, the results of rate differences for some keys are given in Table 7. From the difference rates resulting in values of both ciphered images, it is concluded that the presented scheme has high sensitivity.

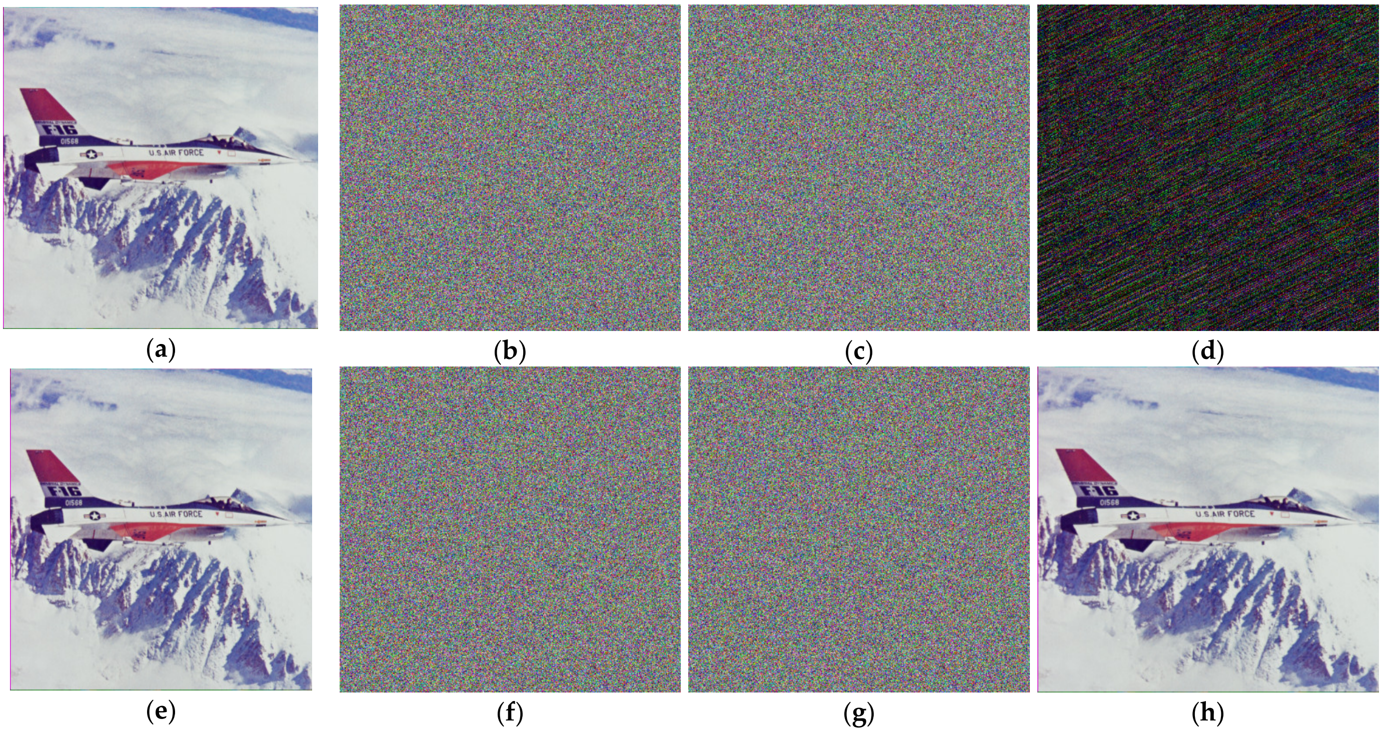

Figure 6. Key sensitivity analysis (**a**) of original image (**b**) encrypted with original initial key set (**c**) and encrypted using slightly different key set. (**d**) Difference in both ciphered (**b**,**c**) image (**e**) decryption of (**b**) using original key set (**f**) decryption of (**b**) using slightly altered key set (**g**) decryption of (**c**) using original key set (**h**) decryption of (**c**) using slightly changed key set.

Table 7. Difference rate of an encrypted image by varying slight keys.

	Difference Rates %				
Secret Keys	Lena	Aeroplane	Peppers	Baboon	House
$\mathcal{X}_1 = \mathcal{X} + 10^{-15}$	0.9988	0.99725	0.99642	0.9988	0.99732
$Z_1 = \mathcal{Z} - 10^{-15}$	0.99650	0.99762	0.99821	0.9987	0.99832
$P_1 = P + 1$	0.99675	0.99870	0.99745	0.99815	0.99854
$R_1 = R - 1$	0.99790	0.99833	0.99644	0.99647	0.99823

6.4. Analysis for Correlation

Correlation helps to check adjacent image pixel association. It is categorized into distinct three formats: diagonal, horizontal, and vertical. To perform this analysis, the whole image texture is considered. The formulation of this analysis is mathematically given as follows:

$$K^* = \sum_{u,v} \frac{(u - \mu u)(v - \mu v)p(u,v)}{\sigma_u \sigma_v} \tag{32}$$

The correlation analysis of the proposed original and ciphered RGB image in vertical, diagonal, and horizontal directions is presented in Table 8. Figure 7 represents vertical, diagonal, and horizontal correlation analysis for RGB images. It can be clearly seen from Table 8 that the correlation of the original image for each channel is near 1 but for ciphered images, the values are near 0, which illustrates that no correlation exists between ciphered image adjacent pixels, which makes it difficult for the attacker to attack.

Table 8. Impact of correlation type on original and encrypted images.

Images	Planes	Horizontal		Vertical		Diagonal	
		Original	Encrypted	Original	Encrypted	Original	Encrypted
Lena	Red	0.9595	0.0020	0.94499	0.00013	0.9272	0.00076
	Green	0.9460	−0.0005	0.9674	−0.00102	0.94136	0.00006
	Blue	0.8956	−0.00012	0.9306	0.00050	0.9164	−0.00864
Aeroplane	Red	0.9049	0.00019	0.9612	−0.00058	0.9707	0.00062
	Green	0.9275	−0.00002	0.9585	−0.000001	0.9491	−0.0008
	Blue	0.9260	0.00021	0.9639	0.00006	0.9590	0.00084
Peppers	Red	0.8977	−0.00632	0.9432	0.000072	0.9349	0.000032
	Green	0.9432	−0.0052	0.9580	0.000043	0.9671	0.00005
	Blue	0.8821	0.00076	0.9731	0.00801	0.9542	0.00046
Baboon	Red	0.8769	0.000423	0.9565	0.000034	0.9829	−0.00005
	Green	0.8672	−0.00002	0.9587	−0.000235	0.9764	−0.00156
	Blue	0.7949	0.000034	0.9656	0.00003	0.9771	0.00007
House	Red	0.9179	−0.00124	0.9810	0.000004	0.9278	0.00058
	Green	0.9331	0.000047	0.9732	−0.000546	0.9652	0.00008
	Blue	0.9650	−0.000032	0.9842	0.000065	0.9277	−0.0006

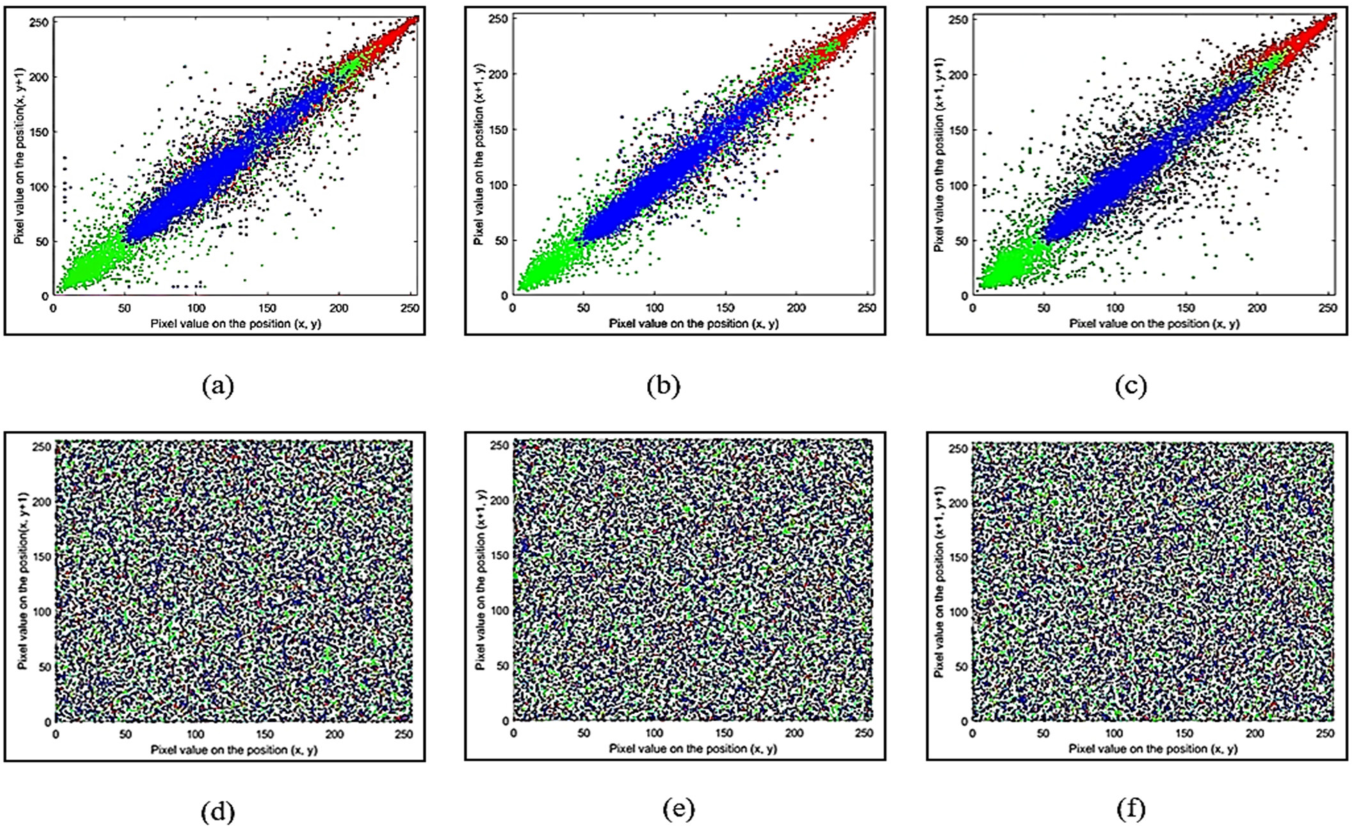

Figure 7. Plain image Red, Green, Blue channel correlation analysis: (**a**) horizontally, (**b**) vertically; (**c**) diagonally. Ciphered image Red, Green, Blue channel correlation: (**d**) horizontally, (**e**) vertically; (**f**) diagonally.

6.5. Histogram Analysis

To assess the security of encryption schemes, uniformity of the encrypted image histogram is very important. To check the pixel value dispersion in certain images, this analysis is used. An encryption scheme has strong algebraic properties if its histogram is uniform. The proposed encrypted image histograms are identical to one another but different from the original image, which ensures its resistance against attacks. In Figures 8 and 9, histograms of each RGB channel for Lena, Aeroplane, Peppers, Baboon, and House images, including both original and encrypted images, are displayed. According to Figures 8 and 9, it is evident that the histograms of the encrypted image for each layer are completely different from the original image and show excellent results.

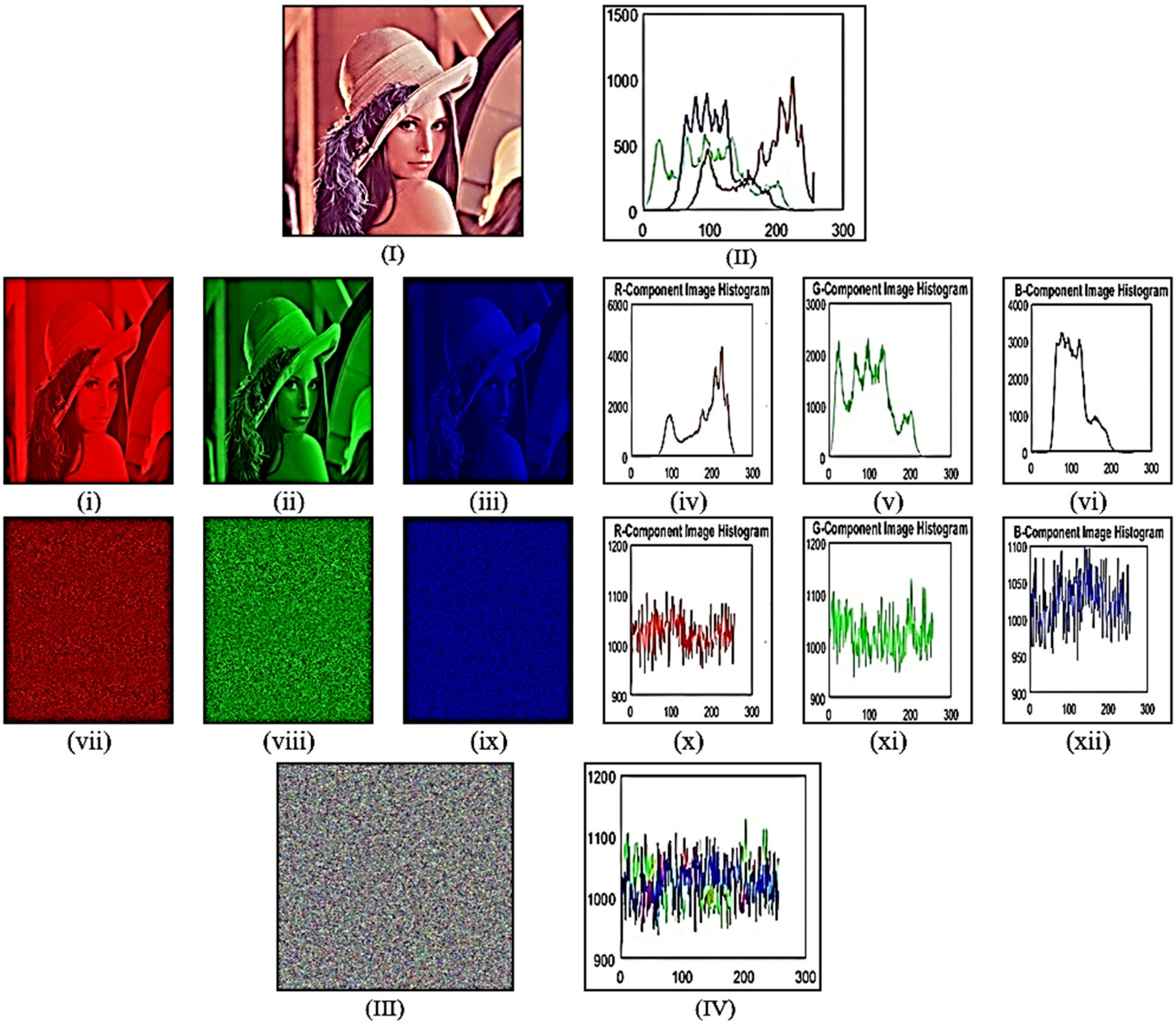

Figure 8. (**I**) Original image; (**II**) original image histogram; (**i**) original image red-layer; (**ii**) original image green-layer; (**iii**) original image blue-layer; (**iv**) original image red layer histogram; (**v**) original image green layer histogram; (**vi**) original image blue layer histogram; (**vii**) encrypted image red layer; (**viii**) encrypted image green layer; (**ix**) encrypted image blue layer; (**x**) encrypted image red layer histogram, (**xi**) encrypted image green layer histogram, (**xii**) encrypted image blue layer histogram (**III**) encrypted combined image; (**IV**) encrypted image combined histogram.

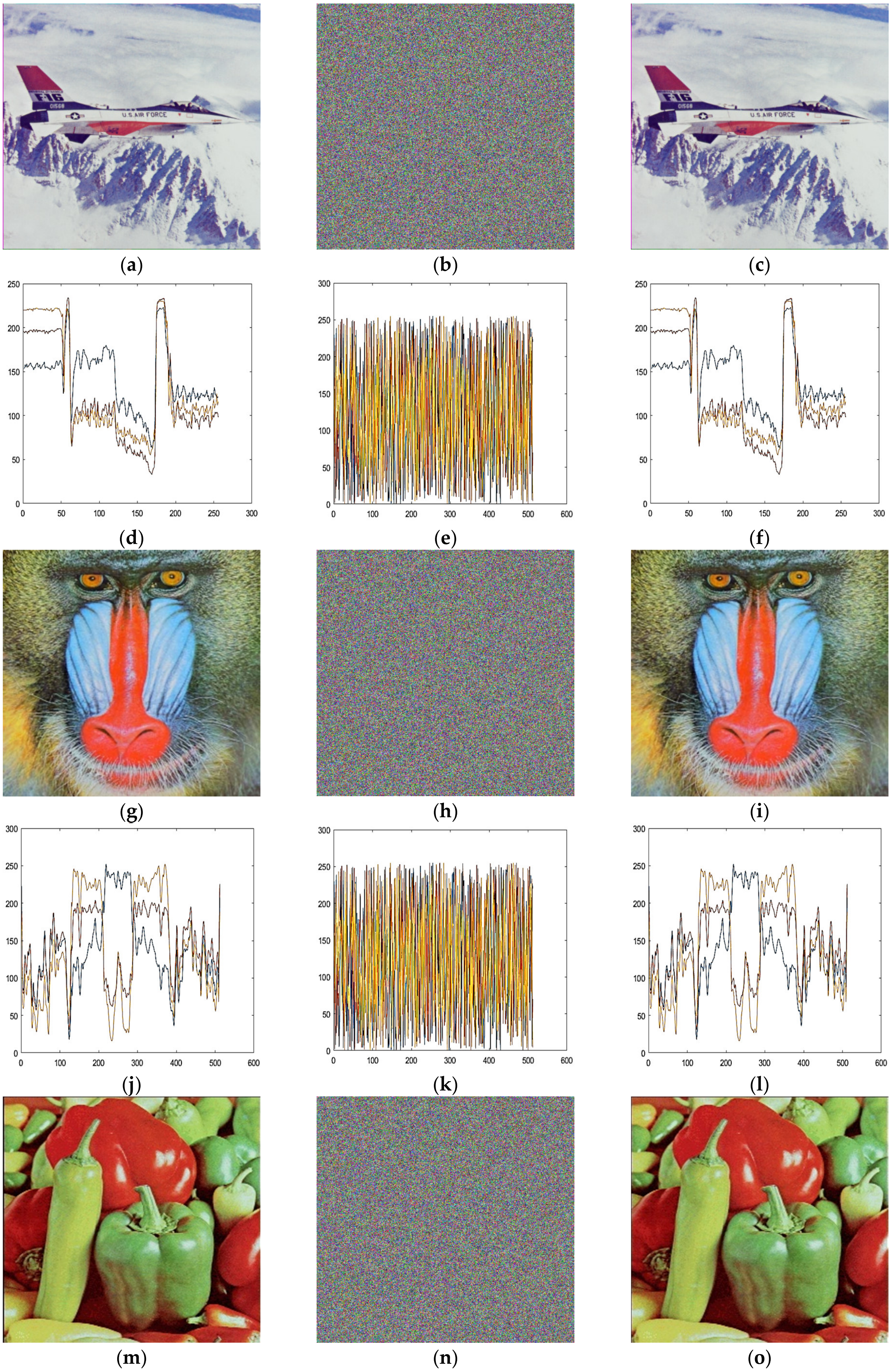

Figure 9. *Cont*.

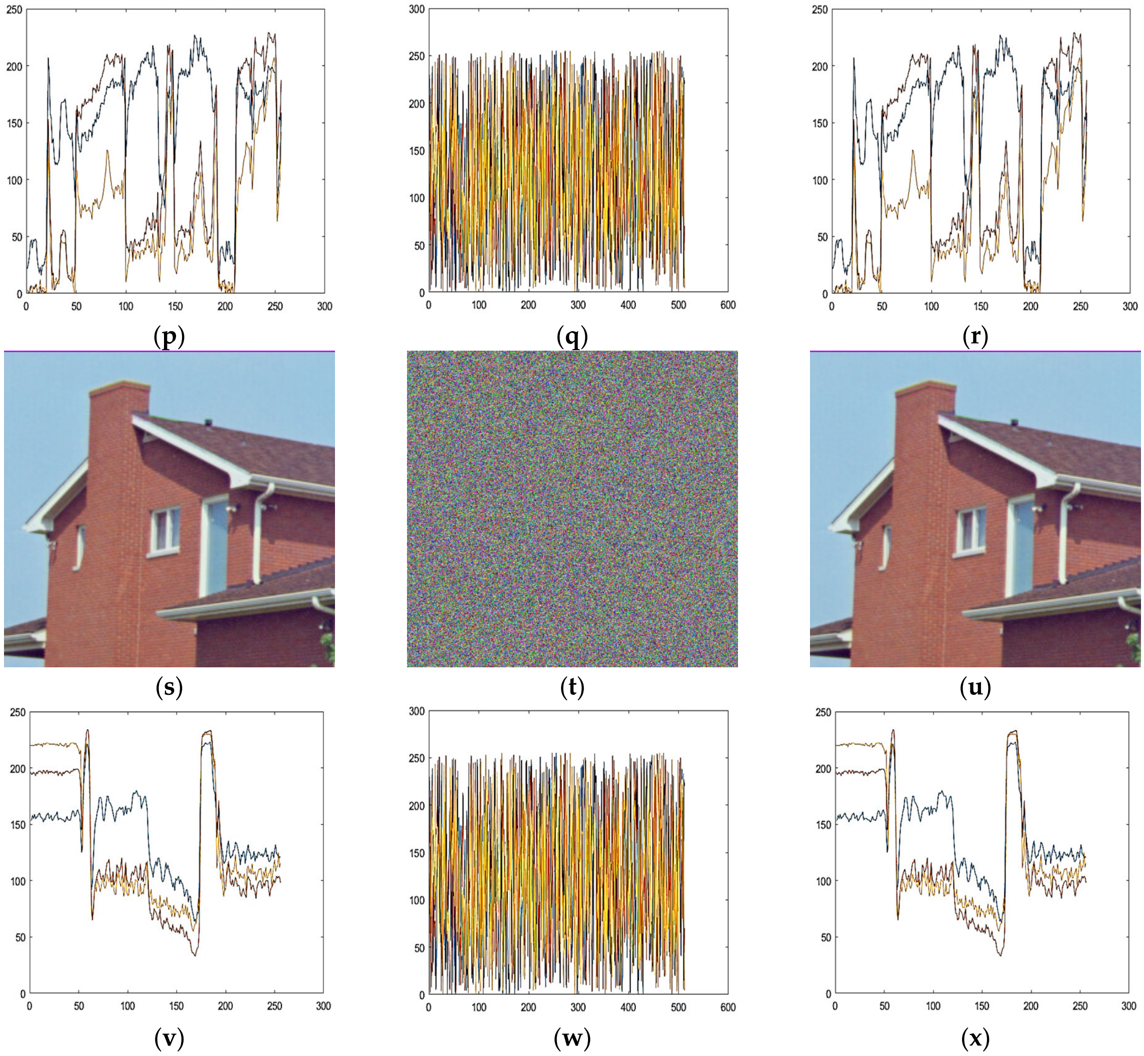

Figure 9. (**a**–**c**) Original, encrypted, and decrypted image of aeroplane, (**d**–**f**) aeroplane original, encrypted and decrypted image histograms, (**g**–**i**) original, encrypted and decrypted image of baboon, (**j**–**l**) baboon original, encrypted and decrypted image histograms, (**m**–**o**) original, encrypted and decrypted image of peppers, (**p**–**r**) pepper original, encrypted and decrypted image histograms, (**s**–**u**) original, encrypted and decrypted image of house, (**v**–**x**) house original, encrypted and decrypted image histogram.

6.6. Chosen-Plaintext Attack Analysis

To break any cryptosystem, in general, it is presumed that the attacker knows exactly the design and workings of an understudy cryptosystem; aside from the secret key, all data are known. This attacker performed four attacks that are characterized in this study; (a) chosen-plaintext attack: a string of ciphertext has been accessed by an opponent; (b) known-plaintext attack: both the cipher and plaintext strings have been accessed by the opponent; (c) chosen-ciphertext attack: a plaintext string is randomly selected by the opponent in this attack, and it helps to obtain a string of corresponding ciphertext; and (d) ciphertext-only attack: a ciphertext string is randomly selected by opponent in this attack, and it helps to obtain a string of corresponding plaintext [30]. Of those four types of attacks, the most important one is the chosen-plaintext attack. In the permutation phase of the proposed scheme, firstly, the initial key values are defined depending on the colored plain image channel information, so if the images are different, the key streams are also changed. And then in the substitution phase, the pixels of permuted image R, G, and B layers are substituted with S-boxes and the chaotic sequence. And for every layer, different

S-boxes are used, which enhance the security of proposed schemes. After this, the substituted image is xored with unknown chaotic sequence values, which helps to modify the ciphered image former values with new ones. So, the proposed encryption process strongly interlinks the image content with keys in such a way that a slight change in key would change the sequences, thus making it resistant against attacks.

6.7. MSE, PSNR and SSIM Analysis

The mean square error (MSE) is the average squared alteration among the original and encrypted images. Mathematically, it is defined as follows:

$$MSE = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} \left(\mathscr{P}_{ij} - \mathscr{C}_{ij}\right)^2}{m \times n} \tag{33}$$

where $\mathscr{P}_{ij}$, $\mathscr{C}_{ij}$ represents the $i^{th} - row$ and $j^{th} - column$ of original plain and ciphered images. Encryption schemes with high robustness must have MSE higher values. Furthermore, the quality of ciphered images is assessed by employing PSNR (peak signal noise ratio), which is mathematically formulated as follows:

$$PSNR = 10 \log_{20} \frac{I^2{}_{max}}{\sqrt{MSE}} \tag{34}$$

where the maximum value of the image pixel is represented by I_{max}. The structural similarity index commonly known as SSIM is used for the measurement of the value of similarity between the original and ciphered images. The testation of the inter-dependency of pixels is usually referred to as the information based on structural properties that help to measure graphically strong pixel relationships. The structural information of any entity is carried out using the dependencies when it is visually seen. The values of SSIM lie between intervals -1 and 1. The mathematical computation of this analysis is given as follows:

$$SSIM(\mathscr{X}, \mathscr{Y}) = \frac{\left(2\varphi_{\mathscr{X}}\varphi_{\mathscr{Y}} + \mathscr{C}_1\right)\left(2\mu_{\mathscr{X}\mathscr{Y}} + \mathscr{C}_2\right)}{\left(\varphi_{\mathscr{X}}{}^2 + \varphi_{\mathscr{Y}}{}^2 + \mathscr{C}_1\right)\left(\mu_{\mathscr{X}}{}^2 + \mu_{\mathscr{Y}}{}^2 + \mathscr{C}_2\right)} \tag{35}$$

where the average values for $\mathscr{X}$ and $\mathscr{Y}$ are represented as $\varphi_{\mathscr{X}}$ and $\varphi_{\mathscr{Y}}$ and their variances and covariance are represented by $\mu_{\mathscr{X}}$, $\mu_{\mathscr{X}}$ and $\mu_{\mathscr{X}\mathscr{Y}}$, $\mathscr{C}_1 = (\mathscr{K}_1\ell)^2$ and $\mathscr{C}_2 = (\mathscr{K}_2\ell)^2$ represents a weak denominator value for stabilized division. For the pixel dynamical values that are signified by l, the $(\mathscr{K}_1, \mathscr{K}_2)$ values are (0.01, 0.03). The results of MSE, PSNR, and SSIM for the proposed "Lena" ciphered and original images and their comparison are given in Table 9. It is clearly depicted from the resulting values that the SSIM ciphered image values are nearest to zero, whereas PSNR is lesser than 10 dB and the values of MSE are greater. This indicates that the ciphered images using the proposed technique are low quality, which means that it is quite difficult to recognize the original image from the ciphered ones. Furthermore, from the comparison result of Lena's image, it can be observed that the proposed scheme PSNR, SSIM, and MSE resulted in values that are better than others.

Table 9. SSIM, PSNR, and MSE analyses and their comparison.

		$\mathscr{R}$	$\mathscr{G}$	$\mathscr{B}$	Average
Proposed	SSIM	0.0025	0.00210	0.00156	0.00205
	PSNR	7.85725	8.47890	7.34291	7.67567
	MSE	1.2550012×10^4	8.9564245×10^3	1.3356792×10^4	1.165969×10^4
[28]	SSIM	-	-	-	0.0078
	PSNR	-	-	-	8.5537
	MSE	-	-	-	9.1434×10^3

6.8. Differential Attack Analysis

To acquire image-significant data, attackers mostly use a tactic in which they alter the original image slightly and then the proposed scheme is applied to encrypt the original and already ciphered image (the attacker wants that image to crack). Consequently, two ciphered images are obtained. In this manner, the attackers cracked the cryptosystem using both ciphered image difference rates; this overall process is known as differential analysis. For encryption algorithm robustness, the presented technique must be highly sensitive to both plain text and the secret key, so any slight secret key or plain text alteration would lead to a whole alteration in the ciphered text. The proposed technique's strength against a differential attack of a ciphered/encrypted image is assessed in the following two ways: one is the rate change in the pixel number NPCR and the other is the unified average changing UACI. Two ciphered images by alteration of only one pixel are considered in NPCR; if $\mathscr{C}_1(u,v)$ is the first image representation and $\mathscr{C}_2(u,v)$ is the second, then the evaluation of NPCR can be conducted using the following equation:

$$NPCR(\mathscr{C}_1,\mathscr{C}_2) = \frac{\sum_{u,\,v} \mathscr{D}(u,v)}{\mathscr{T}} \times 100\% \quad (36)$$

Here, $\mathscr{T}$ is the representation of the pixels' total number and $\mathscr{D}(u,\ v)$ can be defined as follows:

$$\mathscr{D}(u,\ v) = \begin{cases} 0, & if\ \mathscr{C}_1(u,\ v) = \mathscr{C}_2(u,\ v) \\ 1, & if\ \mathscr{C}_1(u,\ v) \neq \mathscr{C}_2(u,\ v) \end{cases} \quad (37)$$

To test the pixels, change number, and measurement of intensity average modification among the ciphered images, the UACI (unified average changed intensity) is utilized [29]. A mathematical formulation of the analysis is given below:

$$UACI(\mathscr{C}_1,\mathscr{C}_2) = \frac{1}{MXN} \sum_{u=0}^{M-1} \sum_{v=0}^{N} \frac{|\mathscr{D}(u,\ v) - P(u,v)|}{F \times T} \times 100\% \quad (38)$$

Here, F is the representation of the highest pixel-validated value with ciphered image format compatibility and $\mathscr{D}(u,\ v)$ is defined as follows:

$$(u,\ v) = \begin{cases} 0, & if\ \mathscr{C}_1(u,\ v) = \mathscr{C}_2(u,\ v) \\ 1, & if\ \mathscr{C}_1(u,\ v) \neq \mathscr{C}_2(u,\ v) \end{cases} \quad (39)$$

The proposed image (Lena, Aeroplane, Peppers, Baboon and House) UACI and NPCR values are provided in Table 10. From the results, we observed that the percentages of NPCR and UACI are greater than 99% and 33.4%. Also, the comparison analyses depict that the results of the proposed technique NPCR for each layer are better than other schemes and the UACI results are better than [26] and comparable with [27].

Table 10. Proposed scheme NPCR, UACI analysis and comparison.

	Images	NPCR			UACI		
		R	G	B	R	G	B
Proposed	Lena	0.99657	0.99884	0.99785	0.33012	0.33232	0.33056
	Aeroplane	0.99899	0.99983	0.99889	0.33015	0.33143	0.33543
	Peppers	0.99912	0.99901	0.99943	0.33320	0.33732	0.32271
	Baboon	0.99653	0.99926	0.99748	0.31543	0.32453	0.3301
	House	0.99677	0.99596	0.99890	0.3300	0.32893	0.33023
[26]	Lena	0.996429	0.995956	0.995285	0.327633	0.300491	0.275669
[27]	Lena	0.9960	0.9961	0.9961	0.3356	0.3345	0.3349

6.9. Occasional Attack Analysis

Generally, in the transmission of data through networks, some information may be lost. For this purpose, occasional attack analysis is used for capacity testing of recovered images (original), even if a small quantity of data has been occluded. In Figure 10, occlusion random analysis is implemented, which indicates that in the transmission of images, a small amount of its information is lost. The recovered images are displayed in Figure 10. It can be seen from the images that the recovered images are still in a format that is readable, even though a portion of the data in the image has been lost.

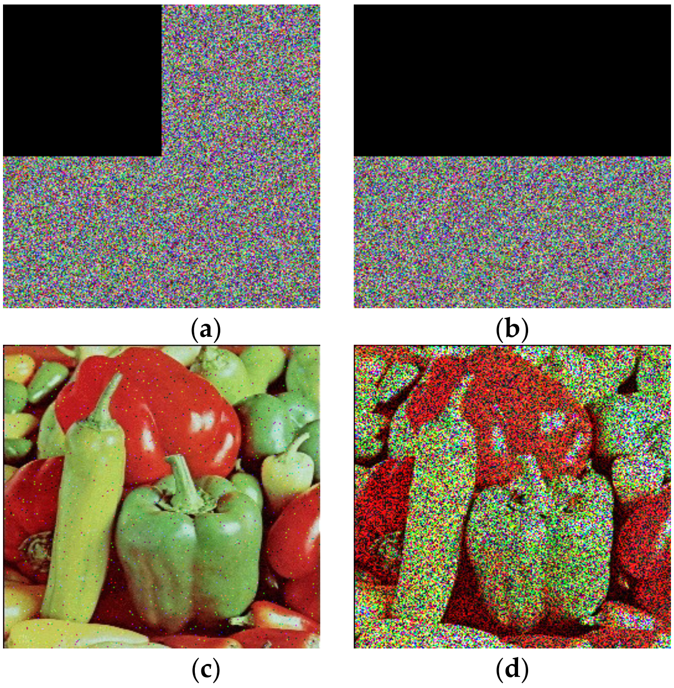

Figure 10. Occasional attack analysis of pepper image. Figure (**a**,**b**) represents cropped ciphered images from different pixel locations. Figure (**c**,**d**) is the representation of their deciphered images having different clarity visuals.

6.10. Time Complexity

The proposed technique is implemented on MATLAB via a personal laptop with the following properties 12th: Gen Intel(R) Core(TM) i7-1255U 1.70 GHz with 8.00 GB RAM. Different image ciphering and deciphering times are recorded and the results are given in Figure 11.

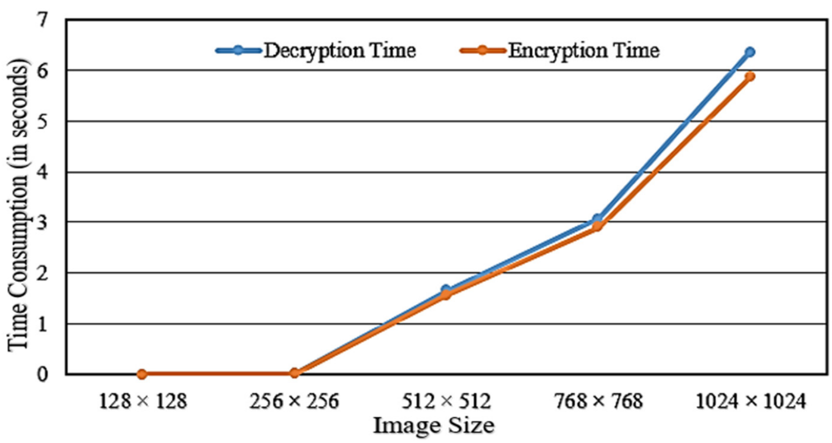

Figure 11. Time complexity analysis.

7. NIST Analysis

The level of security for any cryptosystem can be noted by identifying its distribution of output data, complexity, and period. Robust systems require long periods of time and high complexity, and uniformity. SP 800-22 NIST analysis is utilized for testing the digital image ambiguity [31]. Few test parts have subclasses that are copious. For image randomness testing, number of significant initial key values should be thoroughly tested. Ciphered images are obtained by a completely blended encryption technique of color (RGB) Lena image. The test outcomes are exhibited in Table 11. Following outcome smashing, it could be seen that the anticipated encryption of the digital image tool effectively passed all NIST tests. Consequently, the accomplished outcomes under consideration confirmed that the random ciphers obtained in the presented encryption technique depict asymmetrical behavior in its output.

Table 11. Results of NIST for strongly blended encrypted image.

Test		*p*-Values for Color Encryptions of Ciphered Image			Results
		Red	Green	Blue	
Frequency		0.45097	0.50156	0.14563	Passed
Block frequency		0.88452	0.77680	0.50652	Passed
Rank		0.29191	0.29191	0.29191	Passed
Runs $(\mathrm{M} = 10,000)$		0.90799	0.48637	0.43652	Passed
Long runs of ones		0.71270	0.71270	0.71270	Passed
Overlapping templates		0.80798	0.85898	0.81567	Passed
No overlapping templates		1.00000	0.55782	0.98932	Passed
Spectral DFT		0.88464	0.51806	0.11399	Passed
Approximate entropy		0.01672	0.24941	0.72703	Passed
Universal		0.99315	0.99563	0.99143	Passed
Serial	*p* values 1	0.02576	0.88574	0.04186	Passed
	p values 2	0.00041	0.80981	0.45622	Passed
Cumulative sums forward		0.23652	0.23440	0.18932	Passed
Cumulative sums reverse		1.58850	0.61835	0.09270	Passed
Random excursions	$\mathrm{X} = -4$	0.54297	0.02608	0.21236	Passed
	$\mathrm{X} = -3$	0.45415	0.42343	0.50684	Passed
	$\mathrm{X} = -2$	0.54882	0.26033	0.64829	Passed
	$\mathrm{X} = -1$	0.95535	0.44549	0.17235	Passed
	$\mathrm{X} = 1$	0.67870	0.92004	0.12441	Passed
	$\mathrm{X} = 2$	0.09174	0.03408	0.38548	Passed
	$\mathrm{X} = 3$	3.1754×10^{-8}	0.03260	0.29277	Passed
	$\mathrm{X} = 4$	0.11988	0.40938	0.54159	Passed
Random excursion variants	$\mathrm{X} = -5$	0.01036	0.01305	0.78574	Passed
	$\mathrm{X} = -4$	0.38132	0.85558	0.31266	Passed
	$\mathrm{X} = -3$	0.04636	0.91425	0.21878	Passed
	$\mathrm{X} = -2$	0.030754	0.94459	0.2763	Passed
	$\mathrm{X} = -1$	0.33466	0.95200	0.20873	Passed
	$\mathrm{X} = 1$	0.59873	0.63013	0.81366	Passed
	$\mathrm{X} = 2$	0.59873	0.65143	0.75084	Passed
	$\mathrm{X} = 3$	1.00000	0.46734	0.94398	Passed
	$\mathrm{X} = 4$	0.83989	0.37493	0.92901	Passed
	$\mathrm{X} = 5$	0.69946	0.38827	0.79341	Passed

8. Conclusions

First, we introduced a design for generating binary pseudo-random sequences utilizing Linear Feedback Shift Registers (LFSRs) in conjunction with a modified quadratic map. This approach offers the key advantage of minimizing interference in communication channels. Its applications extend to real-time scenarios, including safeguarding confidential image data, securing Internet banking transactions, and enhancing military communications. The resulting patterns exhibit high nonlinearity, a crucial characteristic akin to S-boxes. Next, we presented a cryptosystem employing a substitution and permutation encryption scheme. We constructed an S-box over binary streams of pseudo-random sequences and assessed their cryptographic strength, confirming their effectiveness through evaluation. Lastly, we proposed a novel encryption technique for colored ciphered images based on modified chaotic quadratic map sequences and S-boxes. This method employs a dual-phase approach, encompassing both substitution and permutation, and introduces a key association strategy based on image content. This strategy aims to emulate the "one-time pad" effect, enhancing resistance to chosen-plain-text attacks (CPAs). Furthermore, we analyzed the presented S-boxes and compared them with recent techniques, demonstrating their robust algebraic properties and highly nonlinear behavior.

Author Contributions: Conceptualization, D.S.M., T.S., S.T., I.M.N., N.L.F. and M.S.; methodology, D.S.M., T.S., N.L.F. and M.S.; software, I.M.N., N.L.F. and M.S.; validation, D.S.M., T.S. and S.T.; formal analysis, I.M.N., N.L.F. and M.S.; data curation, D.S.M.; writing—original draft preparation, D.S.M. and T.S.; writing—review and editing, I.M.N., N.L.F. and M.S.; visualization, D.S.M., T.S., S.T. and I.M.N.; supervision, N.L.F. and M.S.; funding acquisition, N.L.F. and M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All generated or analyzed data in this study are included in this publication.

Acknowledgments: The authors wish to express their sincere appreciation to HITEC University Taxila, Quaid-i-Azam University, Kaunas University of Technology, and Sejong University for their invaluable assistance and cooperation throughout the duration of this research. The contributions provided by these institutions have significantly enhanced the caliber and breadth of this work.

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

References

1. Cassal-Quiroga, B.B.; Campos-Cantón, E. Generation of Dynamical S-Boxes for Block Ciphers via Extended Logistic Map. *Math. Probl. Eng.* **2020**, *2020*, 2702653. [CrossRef]
2. Carlet, C. S-boxes, boolean functions and codes for the resistance of block ciphers to cryptographic attacks, with or without side channels. In *International Conference on Security, Privacy, and Applied Cryptography Engineering*; Springer: Cham, Switzerland, 2015.
3. Detombe, J.; Tavares, S. Constructing large cryptographically strong S-boxes. In *International Workshop on the Theory and Application of Cryptographic Techniques*; Springer: Berlin/Heidelberg, Germany, 1992.
4. Shah, T.; Qamar, A.; Hussain, I. Substitution Box on Maximal Cyclic Subgroup of Units of a Galois Ring. *Z. Naturforschung Sect. A-A J. Phys. Sci.* **2013**, *68*, 567–572. [CrossRef]
5. Çavuşoğlu, Ü.; Zengin, A.; Pehlivan, I.; Kaçar, S. A novel approach for strong S-Box generation algorithm design based on chaotic scaled Zhongtang system. *Nonlinear Dyn.* **2017**, *87*, 1081–1094. [CrossRef]
6. Farhan, A.K.; Ali, R.S.; Natiq, H.; Al-Saidi, N.M.G. A New S-Box Generation Algorithm Based on Multistability Behavior of a Plasma Perturbation Model. *IEEE Access* **2019**, *7*, 124914–124924. [CrossRef]
7. Khan, M.; Asghar, Z. A novel construction of substitution box for image encryption applications with Gingerbreadman chaotic map and S8 permutation. *Neural Comput. Appl.* **2018**, *29*, 993–999. [CrossRef]
8. Tian, Y.; Lu, Z. S-box: Six-dimensional compound hyperchaotic map and artificial bee colony algorithm. *J. Syst. Eng. Electron.* **2016**, *27*, 232–241.
9. Zhang, X.-P.; Guo, R.; Chen, H.-W.; Zhao, Z.-M.; Wang, J.-Y. Efficient image encryption scheme with synchronous substitution and diffusion based on double S-boxes. *Chin. Phys. B* **2018**, *27*, 080701. [CrossRef]
10. Wang, X.; Çavuşoğlu, Ü.; Kacar, S.; Akgul, A.; Pham, V.-T.; Jafari, S.; Alsaadi, F.E.; Nguyen, X.Q. S-Box Based Image Encryption Application Using a Chaotic System without Equilibrium. *Appl. Sci.* **2019**, *9*, 781. [CrossRef]
11. Falih, S.M. A Pseudorandom Binary Generator Based on Chaotic Linear Feedback Shift Register. *Iraq J. Electr. Electron. Eng.* **2016**, *12*, 155–160. [CrossRef]

12. Rahimov, H.; Babaei, M.; Farhadi, M. Cryptographic PRNG Based on Combination of LFSR and Chaotic Logistic Map. *Appl. Math.* **2011**, *2*, 1531–1534. [CrossRef]
13. Ramadan, N.; Ahmed, H.E.H.; Elkhamy, S.E.; El-Samie, F.E.A. Chaos-based image encryption using an improved quadratic chaotic map. *Am. J. Signal Process.* **2016**, *6*, 1–13.
14. Williams, H.; Webster, A.; Tavares, S. On the design of s-boxes. In *Advances in Cryptology—CRYPTO'85 Proceedings*; Springer: Berlin/Heidelberg, Germany, 1986.
15. Dawson, M.H.; Tavares, S.E. An expanded set of S-box design criteria based on information theory and its relation to differential-like attacks. In *Advances in Cryptology—EUROCRYPT'91: Workshop on the Theory and Application of Cryptographic Techniques Brighton, UK, April 8–11, 1991 Proceedings 10*; Springer: Berlin/Heidelberg, Germany, 1991.
16. Biham, E.; Shamir, A. Differential cryptanalysis of DES-like cryptosystems. *J. Cryptol.* **1991**, *4*, 3–72. [CrossRef]
17. Matsui, M. Linear cryptanalysis method for DES cipher. In *Workshop on the Theory and Application of of Cryptographic Techniques*; Springer: Berlin/Heidelberg, Germany, 1993.
18. Mahmood, S.; Farwa, S.; Rafiq, M.; Riaz, S.M.J.; Shah, T.; Jamal, S.S. To Study the Effect of the Generating Polynomial on the Quality of Nonlinear Components in Block Ciphers. *Secur. Commun. Networks* **2018**, *2018*, 5823230. [CrossRef]
19. Zhu, S.; Wang, G.; Zhu, C. A Secure and Fast Image Encryption Scheme based on Double Chaotic S-Boxes. *Entropy* **2019**, *21*, 790. [CrossRef] [PubMed]
20. Liu, L.; Zhang, Y.; Wang, X. A Novel Method for Constructing the S-Box Based on Spatiotemporal Chaotic Dynamics. *Appl. Sci.* **2018**, *8*, 2650. [CrossRef]
21. Belazi, A.; El-Latif, A.A.A. A simple yet efficient S-box method based on chaotic sine map. *Optik* **2017**, *130*, 1438–1444. [CrossRef]
22. Haq, T.U.; Shah, T. 12 $\times$ 12 S-box Design and its Application to RGB Image Encryption. *Optik* **2020**, *217*, 164922. [CrossRef]
23. Mahboob, A.; Nadeem, M.; Rasheed, M.W. A study of text-theoretical approach to S-box construction with image encryption applications. *Sci. Rep.* **2023**, *13*, 21081. [CrossRef]
24. 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. [CrossRef]
25. Haralick, R.M.; Shanmugam, K.; Dinstein, I.H. Textural Features for Image Classification. *IEEE Trans. Syst. Man Cybern.* **1973**, *SMC-3*, 610–621. [CrossRef]
26. Chai, X.-L.; Gan, Z.-H.; Lu, Y.; Zhang, M.-H.; Chen, Y.-R. A novel color image encryption algorithm based on genetic recombination and the four-dimensional memristive hyperchaotic system. *Chin. Phys. B* **2016**, *25*, 100503. [CrossRef]
27. Wu, J.; Liao, X.; Yang, B. Color image encryption based on chaotic systems and elliptic curve ElGamal scheme. *Signal Process.* **2017**, *141*, 109–124. [CrossRef]
28. Joshi, A.B.; Kumar, D.; Kumar, S.; Singh, S. A novel method of digital image encryption using graph theory. *Multimed. Tools Appl.* **2024**, *83*, 6803–6828. [CrossRef]
29. Jahangir, S.; Shah, T. Designing S-boxes triplet over a finite chain ring and its application in RGB image encryption. *Multimedia Tools Appl.* **2020**, *79*, 26885–26911. [CrossRef]
30. Biryukov, A.; Wagner, D. Advanced slide attacks. In *International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin/Heidelberg, Germany, 2000.
31. Pareschi, F.; Rovatti, R.; Setti, G. On Statistical Tests for Randomness Included in the NIST SP800-22 Test Suite and Based on the Binomial Distribution. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 491–505. [CrossRef]