Hindawi
Journal of Electrical and Computer Engineering
Volume 2020, Article ID 8937676, 12 pages
https://doi.org/10.1155/2020/8937676

Research Article

Efficient Encryption System for Numerical Image Safe Transmission

Mohamed Gafsi,[1] **Mohamed Ali Hajjaji**,[1,2] **Jihene Malek**,[1,2] **and Abdellatif Mtibaa**[1,3]

[1]*Université de Monastir, Laboratoire d'Electronique et de Microélectronique, LR99ES30, Monastir 5000, Tunisia*
[2]*Higher Institute of Applied Sciences and Technology, Sousse University, Sousse, Tunisia*
[3]*Université de Monastir, Ecole Nationale d'Ingénieurs de Monastir, Monastir 5000, Tunisia*

Correspondence should be addressed to Mohamed Ali Hajjaji; daly_fsm@yahoo.fr

Received 7 May 2020; Revised 10 August 2020; Accepted 17 August 2020; Published 2 November 2020

Academic Editor: Ping-Feng Pai

In this paper, we propose an efficient cryptosystem for digital image encryption and authentication. The cryptosystem is a hybrid scheme that uses symmetric and asymmetric approaches. The first one is used to encrypt the host image by utilizing a chaos-based key generator. The second one is used to encrypt the initial secret key and the owner's signature that permit authentication. The algorithm is evaluated and validated by its application on several types of standard images and tools such as the statistical analysis, the key, and the performance analysis. The results indicate that the proposed cryptosystem provides high performance and enhanced security. The NIST 800-22 is used for testing the pseudorandom numbers generation (PRNG). The obtained simulation results are better than those cited in the recent works in terms of execution time and security level and low computational complexity.

1. Introduction

Our life today has voracious requirements for advanced technology that can easily store, find, transfer, process, serve, and communicate data and information around the world. The liberty of access to the public networks by anyone has increased the shrewdness of spying, hacking, falsification, illegal copy, and utilization of digital multimedia documents, intellectual property, personal identities, and sensitive information such as images. Secret image protection has lately become an important issue. A large manner of image protection is achieved using cryptography. Cryptography enables data confidentiality by encryption, which transforms it from the plain or ordinary form into an unintelligible form to others who have no right to read, take, modify, use, or copy it. Thereby, the data can be shared through an insecure network.

Generally, there are two encryption schemes: symmetric and asymmetric encryption. The symmetric encryption is more popular because it is generally hundreds to thousands of execution times faster than asymmetric schemes. It uses closely related identical keys for encryption, and for decryption steps, unlike an asymmetric scheme, it utilizes a different key for encryption than for decryption. Thus, when knowing the encryption key of a key pair, we can encrypt data, but we cannot decrypt it; thus, the asymmetric encryption is more secure than the symmetric encryption. Accordingly, the symmetric encryption is intended for large volume data encryption, while the asymmetric encryption is frequently used for securely exchanging a secret key.

In the literature, different image encryption schemes have been carried out using various methods. Many cryptographers have widely adopted the Advanced Encryption System (AES) for image encryption thanks to its high performance [1, 2].

Toughi et al. [3] propose an encryption algorithm for color image protection. Their scheme is a combination of the elliptic curve and AES. About that, the elliptic curves are used for generating three random sequences. These sequences are used for generating three masks in the goal to encrypt the red, blue, and green components of the original image.

Yong and Xueqian [4] proposed an image encryption system based on the combination of the AES-128 and the Cipher Block Chaining mode (CBC) standards. For this, the plain image is fragmented into sub-block sized 128 bit. After that, an initial vector, named IV, with a size equal to 128 bit is generated by the Tent chaotic map and XORed with the initial plain sub-block. Secondly, the AES-128 is applied to obtain the first ciphered sub-block. Finally, the rest of the different sub-blocks are scripted sequentially following the same steps applied on the first one block.

According to the results presented in references 1 and 2, we notice that the encryption systems based on the AES algorithm cause a long execution time [5]. This disadvantage affects, directly, the global quality of the system in case of online encryption. In the other case, the use of the CBC mode has many disadvantages such as the sequential architecture and that it can cause the slowdown in encryption systems. Another disadvantage of the CBC standard is that the propagation of an error may occur easily and can affect all blocks.

Many cryptographers turned to use the chaos theory for secure image encryption. This is due to many advantages of chaos such that deterministic pseudorandom numbers generation (PRNG), long periodicity, sensitive to the initial conditions, and large key space. Several chaotic systems have been investigated for pseudorandom number generation such as the Lorenz system, Rössler, Skew tent, and PWLCM map, etc [6].

Sha-Sha et al. [7] have investigated the phase-truncated short-time fractional Fourier transform for image encryption. The key stream was generated by Chen's hyper-chaotic system. For image encryption, the original image was firstly decomposed into four subimages of the same size. Then, the subimages were encoded using a combination between the PTSTFrFT and a wave-based permutation. Finally, the image was diffused with a key to produce the encrypted image.

In [8], the authors put forward a multi-image compression-encryption algorithm. The key stream was obtained using the Logistic-Tent and Tent-Sine systems which were maintained in cascade. Firstly, a spectrum of multi-image was obtained using the Discrete Cosine Transform (DCT) and, then, compressed. Next, a Quaternion representation QR of the multi-image was obtained. After that, a Double Random Phase Encoding (DRPE) was operated on the Quaternion Discrete Fractional Hartley Transform (QDFrHT) in order to get a quaternion signal. Then, a matrix named E sized 4 times the size of the original image was generated by extracting the real and the three imaginary parts of the quaternion signal. Finally, a block-based confusion and pixel diffusion methods were applied to the matrix E in order to obtain an encrypted compressed image.

Zhou [9] suggested an image cryptosystem based on a combination between the 3D orthogonal Latin squares (3D-OLSs) and a matching matrix. Firstly, the 3D sine map was used to generate three chaotic sequences. Next, a 3D orthogonal Latin square and a matching matrix were produced by using the chaotic sequences. Then, the 3D-OLSs and the matching matrix were jointly used to permute the original image. After that, all planes of the permuted matrix were divided into sixteen blocks of the same size. The chaotic sequence was sorted, and a position matrix was generated. According to the position matrix, the blocks of each plane were linked and shifted by using a cyclic shift operation, and then, a new matrix was generated. Finally, the encrypted image was generated by executing a diffusion operation for the new matrix.

In [10], Hongjun suggested an image encryption scheme based on the DNA sequence and two chaotic maps. The scheme was symmetric, and they adopted confusion-diffusion as the encryption architecture. The MD5 was used to generate an initial secret key. The initial parameters of chaos maps were generated from the initial secret key. The image was confused using the PWLCM map and, then, confused using DNA and the Chebyshev map.

Jawed [11] proposed simple image encryption based on the skew tent map. The SHA-3 was used for generating an initial secret key of the system. After that, the image was transformed, permuted, and confused. These operations were performed successively by using the Discrete Cosine Transform (DCT), the multiplication with orthogonal matrix, the application of the skew tent map, and the Xor operation.

In [12], Liu propose an image encryption algorithm based on the combination between the Liu chaotic map and a hash function. The hash function was used to generate an initial secret key of the encryption system. The initial parameter of the Liu system was derived from the initial secret key. After that, three pseudorandom sequences were generated to scramble the image pixels.

Xiuli [13] proposes an image encryption system using DNA, SHA-386, and a chaotic map. However, the SHA-386 was used to generate an initial secret key of the system. Then, different pseudorandom number sequences were generated by a four-wing chaotic map. Next, a DNA encoding/decoding and permutation methods were applied to the three components, red, blue, and green, in order to produce the encrypted image.

Guesmi [14] proposed an encryption algorithm for image protection. The system was based on a hybrid model of Deoxyribonucleic Acid (DNA) masking, a Secure Hash Algorithm (SHA-2), and the Lorenz chaotic system. Firstly, the SHA-256 was applied to generate an initial secret key. The initial values of the Lorenz system were generated from the initial secret key. Secondly, the original image and sequence K were encoded into two DNA sequences using a chosen rule from the eight kinds of DNA map rules. Thirdly, the first DNA sequence was permuted using a Lorenz chaotic sequence. Finally, the DNA sequence was decoded utilizing a chosen rule from the eight kinds of DNA map rules.

The challenge is that traditionally, key generation, encryption, authentication, and integrity have been complex and computationally costly to execute while keeping in mind the issue related to the security level. All mentioned image encryption schemes have many weaknesses, and it was symmetries, which require key securing and authentication. Many of them are sequential and too long in design and calculation, which greatly increase the execution time.

In this work, we suggest an efficient encryption algorithm for digital image protection and authentication. We aim for a high-security level, low complexity, and high performance in terms of execution time. For a high-quality key generation, a highly sensitive PRNG-based chaotic system is designed. For image encryption, we adopt PRNG-CTR diffusion as architecture. For authentication and key securing, both the initial secret key and owner signature are concatenated and encrypted using the Rivest–Shamir–Adleman (RSA) system. This paper is planned in four parts as follows: In Section 2, the proposed encryption algorithm is described. The simulation, analysis, evaluation, and validation of the proposed algorithm are given in Section 3. Section 4 concludes the work.

2. Proposed Cryptosystem Algorithm

This section is reserved for describing the proposed algorithm applied to digital image protection and authentication. As shown in Figure 1, the cryptosystem algorithm is a hybrid scheme. However, it uses a symmetric and an asymmetric scheme.

The RSA system is employed as the asymmetric scheme, which is used to encrypt the initial secret key and the sender's signature. On the other hand, a diffusion-based PRNG-CTR symmetric encryption is adopted for encrypting the secret image.

2.1. Encryption Step. The encryption system includes three parts: key generation, image encryption, and secret key encryption. The first one is a PRNG used for a high-quality key generation. The second one is the encryption step of the whole image by using the generated key with the CTR mode of the encryption operation. The third one consists of encrypting the totality of the owner's signature and the initial secret key by the RSA algorithm in goal to control authenticity and engage the safety asymmetric type of the algorithm.

2.1.1. Pseudorandom Number Generator. The PRNG is useful to generate a key for encryption. It produces a sequence of numbers such that its properties are statistically independent, uniformly distributed, and unpredictable. To provide a high-quality key, we use a PRNG based on the Lorenz chaotic system and the SHA-256. The Lorenz system is a mathematical model composed of three differential equations described as follows [15]:

$$\begin{cases} \dfrac{dx}{dt} = a(y - x), \\ \dfrac{dy}{dt} = cx - y - xz, \\ \dfrac{dz}{dt} = xy - bz, \end{cases} \tag{1}$$

where x, y, and z are the system variables and a, b, and c are the system parameters. The Lorenz system exhibits a chaotic behaviour for certain parameter values and initial conditions. When $a = 10$, $b = 8/3$, and $c = 28$, the system has a chaotic behaviour [14].

The SHA-256 is used to generate a 256-bit initial secret key K_i fully related to the plain image [16]. K_i is divided into 8-bit blocks as follows:

$$K_i = k_1 | k_2 | k_3 | \cdots | k_{32}. \tag{2}$$

The initial state which consists of (x_0, y_0, z_0) of the Lorenz system is derived from K_i utilizing the following equations:

$$x_0 = \frac{(k_1 \oplus k_2 \oplus \cdots \oplus k_{11})}{2^8}, \tag{3}$$

$$y_0 = \frac{(k_{12} \oplus k_{13} \oplus \cdots \oplus k_{22})}{2^8}, \tag{4}$$

$$z_0 = \frac{(k_{23} \oplus k_{24} \oplus \cdots \oplus k_{32})}{2^8}. \tag{5}$$

By iterating the Lorenz system and modulating its values, equation (6), a long sequence of independent numbers PRNS is obtained, in the range [0, 255], which exhibits high random behaviour.

$$\text{PRNS} = \left(n_i \times 10^{12}\right) \bmod 256. \tag{6}$$

2.1.2. Image Encryption. To perform image encryption, we adopt Xor diffusion based on the 128-bit PRNG-CTR as the architecture. However, the algorithm processes a fixed block and a key sized 128 bit.

(i) Firstly, two streams of 128-bit numbers, NS1 and NS2, are generated using two counters with different initialization vectors IV1 and IV2. The initial vectors of the counters are derived from the initial key K_i as follows:

$$\text{IV1} = K_i(1 : 128), \tag{7}$$

$$\text{IV2} = K_i(129 : 256). \tag{8}$$

(ii) Secondly, the Xor operation between NS1 and NS2 is used in the goal to obtain another sequence of 128-bit numbers named NS3.

(iii) Thirdly, NS3 is XORed with the PRNS (generated by PRNG) in order to acquire a 128-bit key stream named KS.

(iv) Finally, we diffuse the image block with KS to produce the corresponding encrypted image with the same size. The image encryption instruction can be described by equation (9) and the system is depicted in Figure 2.

$$\text{CB}_i = ((\text{NS1} \oplus \text{NS2}) \oplus \text{PRNS}) \oplus \text{PB}_i, \tag{9}$$

where CB is a cipher block; i is the index from 0 to N (total number of blocks); NS1 is a stream of 128-bit number

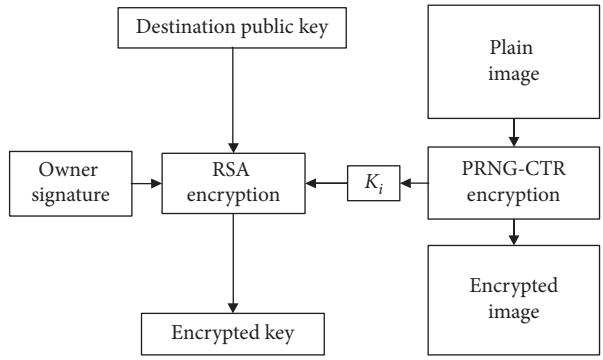


Figure 1: General view of the proposed encryption system.

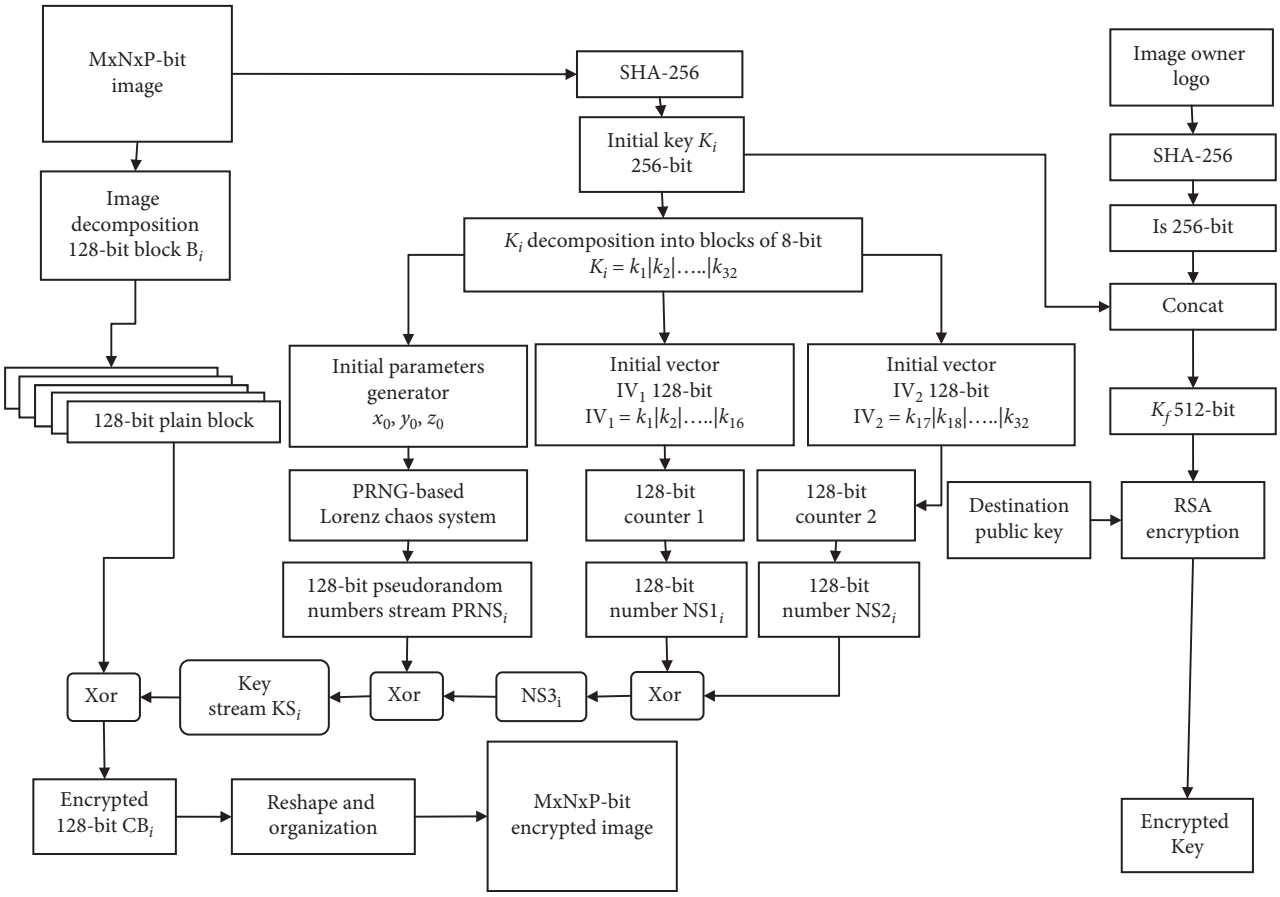


Figure 2: Flowchart of the proposed encryption system.

generated by counter1; NS2 is a stream of 128-bit number generated by counter2; PRNS is a stream of pseudorandom numbers generated by the PRNG; PBi is a plain block.

The proposed algorithm has significant advantages; it enables fast image encryption with low computational complexity, no propagation error may occur, has random access into any block, and the same image can be encrypted with a variety of key streams. Beyond, the architecture is flexible for key extension and easily parallelizable to speed up execution for meeting real-time requirements.

2.1.3. Image Authentication and Key Encryption. For image authentication, a 256-bit signature is generated fully related to the owner logo using SHA-256. The signature is combined with the 256-bit initial key and encrypted together by the RSA encryption algorithm to engage the safety asymmetric type of the key [17], which enhance the security of the algorithm. However, the public key of the destination is used to encrypt the initial key utilizing equation (10). At the destination, the private key is only used for decrypting the initial key, using equation (11), to finally decrypt the received image and detect the owner's logo.

$$K_f = K_i^e \bmod (n), \tag{10}$$

where K_i is the plain key, K_f is the correspondent encrypted key, and (n, e) is the public key of destination.

$$K_i = K_f^d \bmod (n), \tag{11}$$

where K_i is the decrypted key, K_f is the encrypted key, and (n, d) is the private key of the destination.

2.2. Decryption Step. After the encryption step, the encrypted image can be sent to a well-defined destination using an insecure network (diffusion step). At the reception, the image must be manipulated by the extraction system, Figure 3, to seek the plain image and detect the owner's logo. Practically, it uses the reverse algorithm, according to the encryption algorithm. This phase will take place just when the encrypted image and the key are available.

The algorithm starts by decrypting the encrypted key K_f by the RSA system to obtain the plain key K_f. The first part, 256 bit, is the initial key K_i used for image decryption, and the remaining part, 256 bit, is the owner digital signature used for image authentication. However, using K_i, the received image is decrypted by the decryption system. This treatment enables finding the plain image. On the other hand, utilizing the digital signature with the help of a database, we detect the image owner's logo.

3. Experimental Results and Interpretation

The proposed algorithm is implemented in Matlab 2017b and using a personal computer with an Intel (R), Core (TM) i7-5500 U, CPU at 3.4 GHz, and 8 Go running Windows 10.

To evaluate the performance of the proposed approach, simulation results and performance analysis are given in this section. For this, we have considered several indicators discussed thereafter. This part includes statistical analysis, key analysis, and algorithm speed.

3.1. Statistical Analysis. Statistical analysis is the computation of the similarity factor between the original and its corresponding encrypted image in the goal to evaluate the ability of information in defending hackers. This is can be achieved using the image histogram, information Entropy (E), Normalized Correlation (NC), correlation coefficient (ρ), Peak Signal to Noise Ratio ($PSNR$), and Structural Similarity Index Measure ($SSIM$) tools [18–20].

3.1.1. Histogram Analysis. The histogram of an image is the distribution of information related to the pixel values. An ideal image ciphering system should always generate an encrypted image with uniform and completely different histograms compared to those related to the plain images. We analyze the histograms of several encrypted images and their corresponding original images that have different contents. Six examples of ordinary images are shown in Figure 4.

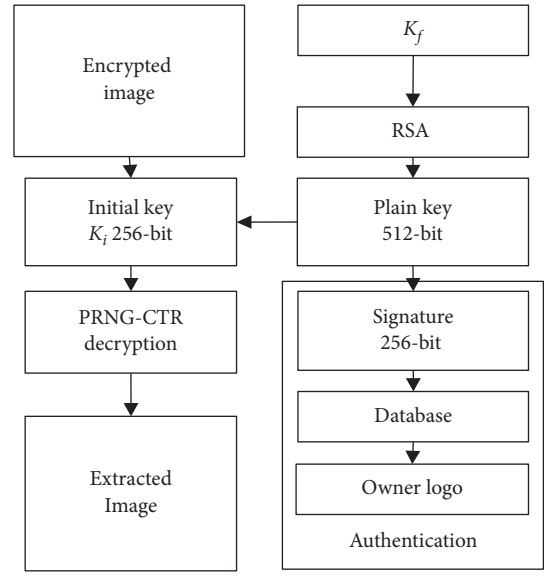


Figure 3: General architecture of the extraction system.

As seen in Figures 4(d), 4(h), 4(l), 4(p), 4(t), and 4(x), we note that the histogram of the resultant encrypted image is fairly uniformly distributed and so different compared to the histogram of the plain image, Figures 4(b), 4(f), 4(j), 4(n), 4(r), and 4(v), which contains large spikes. Therefore, it is difficult to understand the encrypted image appearance.

3.1.2. PSNR, SSIM, and Entropy Analysis. In this section, we evaluate the proposed system on six selected ordinary images. After their ciphering, the PSNR, SSIM, NC, and entropy factors are computed. Table 1 gives the simulation results.

For analysing the obtained results, we start by the entropy factor. Theoretically, a good encryption system gives random information equal to 8 [19]. As seen in Table 1, we observe that the entropy value of the encrypted image is close to the ideal value 8. As a consequence, we can conclude that the proposed system is safe against entropy attacks. After that, we pass to analyze the obtained PSNR results. These results indicate that the PSNR is lower than 5 dB (<5 db). Following reference [21], the proposed system gives a very bad quality between the encrypted and plain images. We conclude that it is very difficult to predict the plain image from the encrypted one. By analysing the SSIM factors between the plain and encrypted images. As shown in Table 1, we note that the SSIM values are close to 0. We conclude that we cannot distinguish the content of the clear image from the encrypted one.

For evaluation of the proposed encrypted system, a comparative study of the Lena image with some recent work for entropy and PSNR evaluation tools is presented in Table 2. We conclude that the proposed system gives the best results against many other ciphering systems.

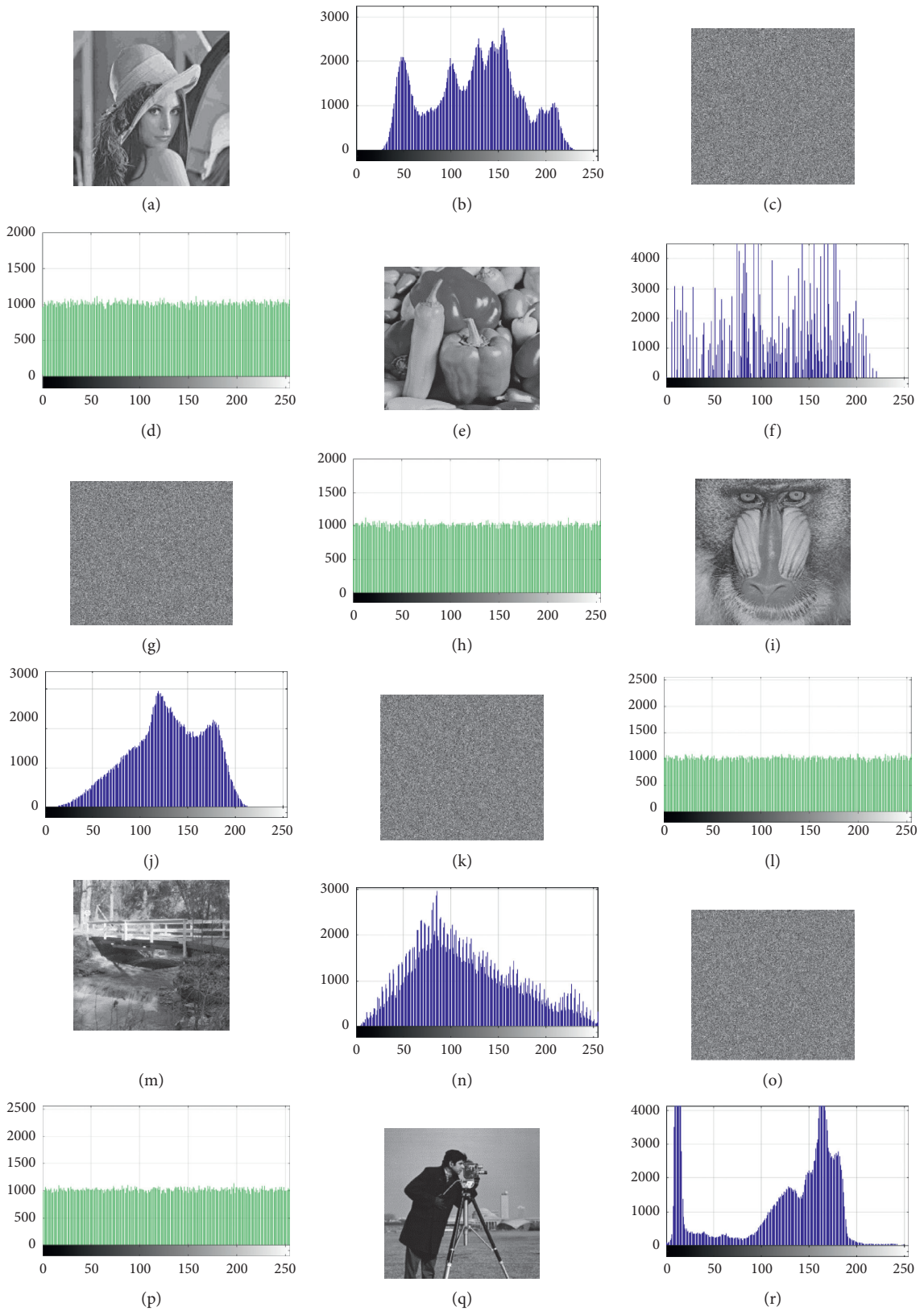


Figure 4: Continued.

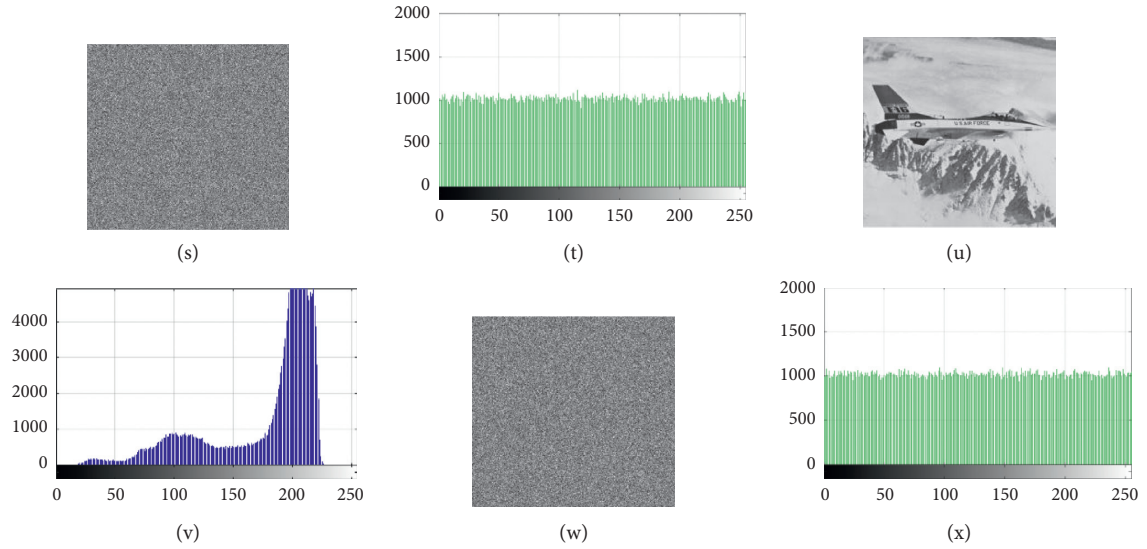


Figure 4: Histogram of the original images and their corresponding encrypted images.

Table 1: PSNR, SSIM, NC, and E values of the encrypted image.

Image	E	PSNR	NC	SSIM
Lena	7.99935	4.495	−0.0010	0.0101
Peppers	7.99932	4.721	−0.0015	0.0094
Baboon	7.99932	4.480	0.0011	0.0101
Walkbridge	7.99935	4.667	0.00019	0.0085
Cameraman	7.99932	4.291	0.0026	0.0092
Jetplane	7.99936	4.295	−0.0030	0.0107

3.1.3. Correlation Coefficient Analysis. In an ordinary image, the correlation of the adjacent pixels is close to one. Unlike in an encrypted image, the adjacent pixels are not correlated [27]. Let x and y be two gray scale values of two adjacent pixels in the image; the correlation of the adjacent pixels is computed using the following equations:

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{12}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N} \left(x_i - E(x_i)\right)^2, \tag{13}$$

$$\operatorname{cov}(x, y) = \frac{1}{N}\sum_{i=1}^{N} \left(x_i - E(x)\left(y_i - E(y)\right)\right), \tag{14}$$

$$r_{xy} = \frac{\operatorname{cov}(x, y)}{\sqrt{D(x)}\sqrt{D(y)}}. \tag{15}$$

$E(x)$ is the expectation of x, $D(x)$ is the estimation of the variance in x, and cov (x, y) is the estimation of the covariance between x and y. Table 3 shows that the encrypted images correlation coefficients are close to zero. So, we cannot distinguish the content of the clear image from the encrypted one. Table 4 shows the distributions of 2000 pairs of randomly selected adjacent pixels of the original and encrypted Lena image, respectively.

Table 2: Comparative study of the PSNR and E values for the Lena image.

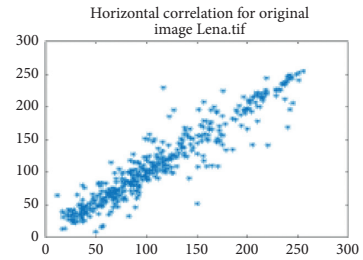
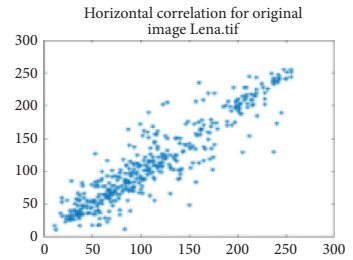
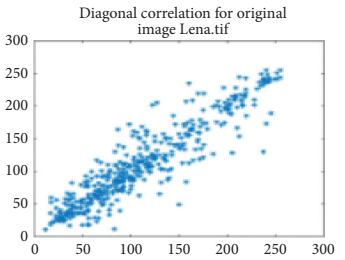
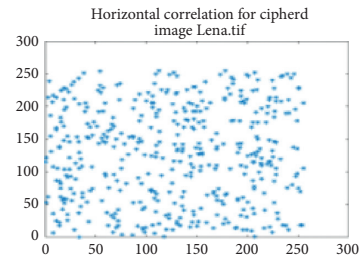
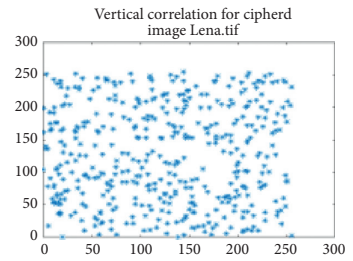
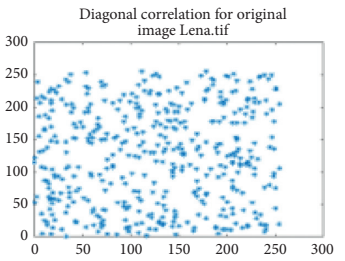
Cryptosystem	PSNR	E
[13]	—	7.99920
[22]	—	7.99122
[23]	10.0454	7.75970
[24]	—	7.90303
[25]	9.3000	7.99910
[26]	—	7.99730
Proposed algorithm	4.495	7.99935

Table 3: ρ and NC values of the original image and its corresponding encrypted image.

Image	Status	NC	Horizontal	Vertical	Diagonal
Lena	Plain	−0.0010	0.80482	0.81942	0.82776
	Cipher		0.00027	−0.01339	−0.07913
Peppers	Plain	−0.0015	0.84679	0.97335	0.88347
	Cipher		0.00638	−0.06850	−0.05433
Baboon	Plain	0.0011	0.83045	0.87679	0.87929
	Cipher		0.00742	−0.00602	−0.10326
Walkbridge	Plain	0.00019	0.87108	0.87934	0.84997
	Cipher		0.00093	0.00380	−0.00776
Cameraman	Plain	0.0026	0.87790	0.89079	0.90087
	Cipher		0.00226	−0.00774	−0.00580
Jetplane	Plain	−0.0030	0.80839	0.81643	0.82083
	Cipher		−0.05928	0.00381	−0.15028

The results clearly show that the correlation coefficient of the original images is close to 1, while that of the encrypted images is close to zeros. In addition, the distribution of adjacent pixels is inconsistent; i.e., there is no correlation between them. This indicates that the algorithm eliminates the correlation of adjacent pixels in the plain image, and it makes an encrypted image with no correlation. A comparative study of the Lena image with some recent works for

Table 4: Distribution of 2000 pairs of randomly selected adjacent pixels for the Lena image.

Image	Horizontal	Vertical	Diagonal
Original Lena			
Encrypted Lena			

the correlation coefficient evaluation tool is presented in Table 5. We conclude that the proposed system gives the best results against other works.

3.2. Key Analysis. Here, we describe the key space, key sensitivity, and randomness analysis test to evaluate the strength of the encryption scheme against hackers.

3.2.1. Key Space. The key space of a safety encryption scheme should be very large to resist the brute-force attack. In the proposed algorithm, for an initial key K_i, there are 2^{256} dissimilar keys, which is very large. Certainly, the key brute-force attacks are computationally infeasible. Table 6 gives a comparative study of the key space with other recent encryption algorithms.

3.2.2. Key Sensitivity. For high safety encryption, the encryption algorithm must be sensitive to the entered image and the initial secret key K_i. A very small change (one bit) in K_i, or in the image, will cause a greatly significant change in the output generated keys for encryption and output image. Key sensitivity can be achieved by using the Number of Changing Pixel Rate (NPCR) and Unified Averaged Changed Intensity (UACI) randomness tests to evaluate the robustness of the image against differential hackers [28], which are described as follows:

$$\mathrm{NPCR} = \frac{1}{S}\sum D(i,j) \times 100\%, \tag{16}$$

$$\mathrm{UACI} = \frac{1}{S}\sum \frac{|d|}{G} \times 100\%, \tag{17}$$

where S is the size of the image and $D(i, j)$ is a logical value affected by the following cases:

Table 5: Comparative study of the correlation coefficient for the Lena image.

Work	Horizontal	Vertical	Diagonal
[3]	−0.0004	−0.0018	0.0001
[13]	0.0265	0.0792	0.0625
[22]	−0.0001	0.0089	0.0091
[23]	0.0591	0.0508	0.0480
[24]	−0.0294	−0.0014	−0.0180
[25]	−0.0047	0.0015	0.0030
[26]	−0.0226	0.0041	0.0368
Proposed system	0.0002	−0.0133	−0.0791

$$D(i,j) = \begin{cases} 0, & I_1(i,j) = I_2(i,j), \\ 1, & I_1(i,j) \neq I_2(i,j), \end{cases} \tag{18}$$

where d is the difference between two pixels on the image with the same coordinates.

$$d = p_1(i,j) - p_2(i,j). \tag{19}$$

However, a sensitivity test applied to the initial key is performed using two initial keys, K_{i1} and K_{i2}, but K_{i2} is different by one bit from Ki1 to encrypt the same image. After that, we try decrypting the obtained images by a wrong key. Here, the two keys, K_{i1} and K_{i2}, are permuted in the decryption step, i.e., each image is decrypted by a wrong key which is different by one bit from the correct key. This test is carried out with many K_i keys which are randomly chosen to properly evaluate the algorithm. Simulation results for the Lena image are shown in Figure 5 and Table 7.

It is remarkable that our algorithm is highly sensitive to the initial key for simultaneous encryption and decryption phases. The NC value between the two encrypted images is very weak, i.e., there is no similarity coefficient between them. Besides, the difference between them is another image. In addition, the NPCR and UACI percentages are important, which signifies that the encrypted images are greatly

Table 6: Comparative study of the key space.

Work	Key space
[7]	3.4×10^{38}
[22]	10^{117}
[24]	2^{208}
Proposed algorithm	2^{256}

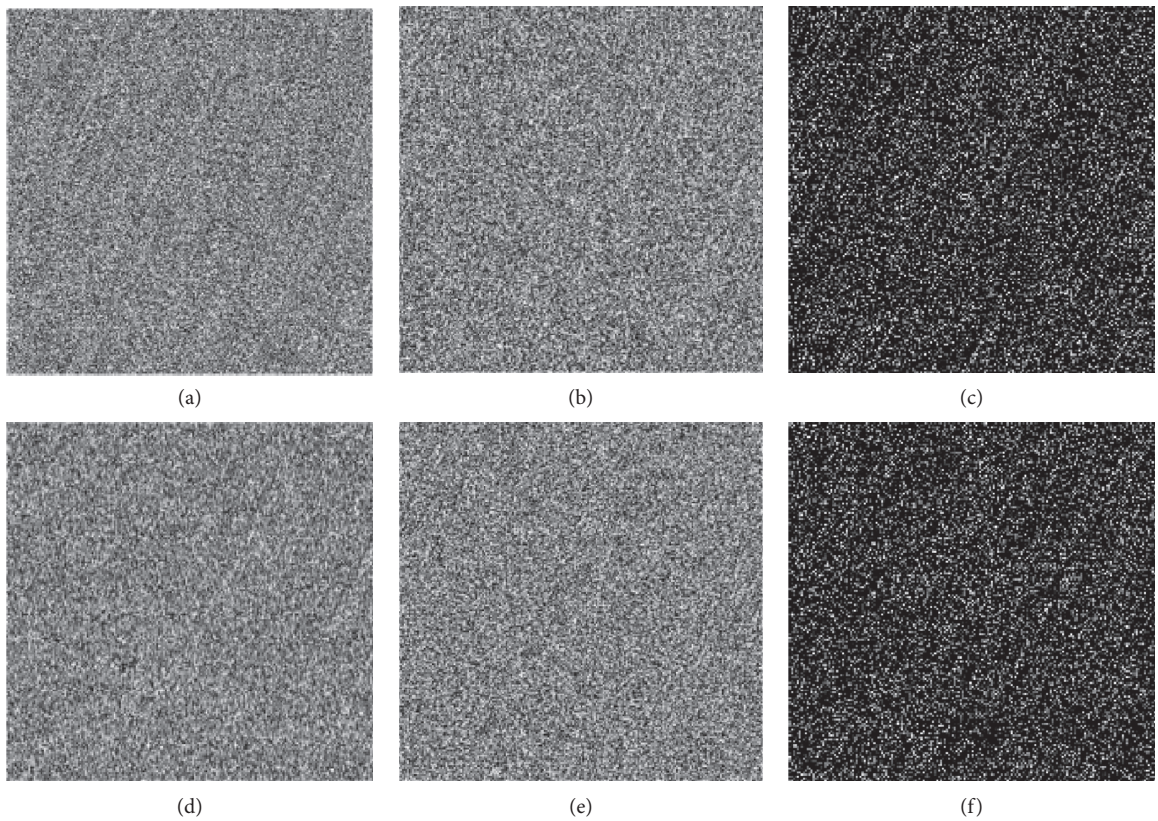


Figure 5: Key sensitivity test applied on the initial key: (a) encrypted Lena by key K_{i1}, (b) encrypted Lena by key K_{i2}, (c) difference between (a) and (b), (d) decrypted (a) by key K_{i2}, (e) decrypted (b) by key K_{i1}, and (f) difference between (d) and (e).

Table 7: NC, NPCR, and UACI tests applied on the initial key.

Image	NPCR (%)	UACI (%)	NC
Lena	99.69369	33.19117	0.00170
Peppers	99.75318	33.28726	0.00292
Baboon	99.65560	33.95780	−0.05231
Walkbridge	99.66370	33.49820	0.00419
Cameraman	99.69556	33.50594	−0.00379
Jetplane	99.75471	33.39463	−0.01982

dissimilar. We conclude that the plain image cannot be recovered using a wrong key by one bit. The same sensitivity is obtained by all randomly chosen keys.

Furthermore, we perform a key sensitivity test, applied on the original image, using two images, I1 and I2, but the second one is different by one bit from the first one. The change is selected randomly. The NPCR and UACI simulation for the Lena image are presented in Figure 6, and the simulation results for all images are introduced in Table 8.

As shown in the aforementioned simulations results, it is clear that our encryption design is very sensitive to a tiny change in the entered image. This demonstrates the effectiveness of the SHA-256 function, which puts an image's initial key specific for encryption. Table 9 gives a comparative study of the NPCR and UACI evaluation tests with some recent work.

3.2.3. Randomness Analysis. Random analysis can be achieved using the NIST 800-22. The test is useful to test random and pseudorandom number generators [31] to determine whether or not a PRNG is appropriate for data encryption. The analysis contains 15 tests that assess key streams to meet important necessities. It focuses on different nonrandom aspects that can be found in a key sequence. The test results of 262144 sequences of 128 bit generated by the proposed PRNG are shown in Table 10. The sequences pass all tests successfully. This demonstrates that the generated

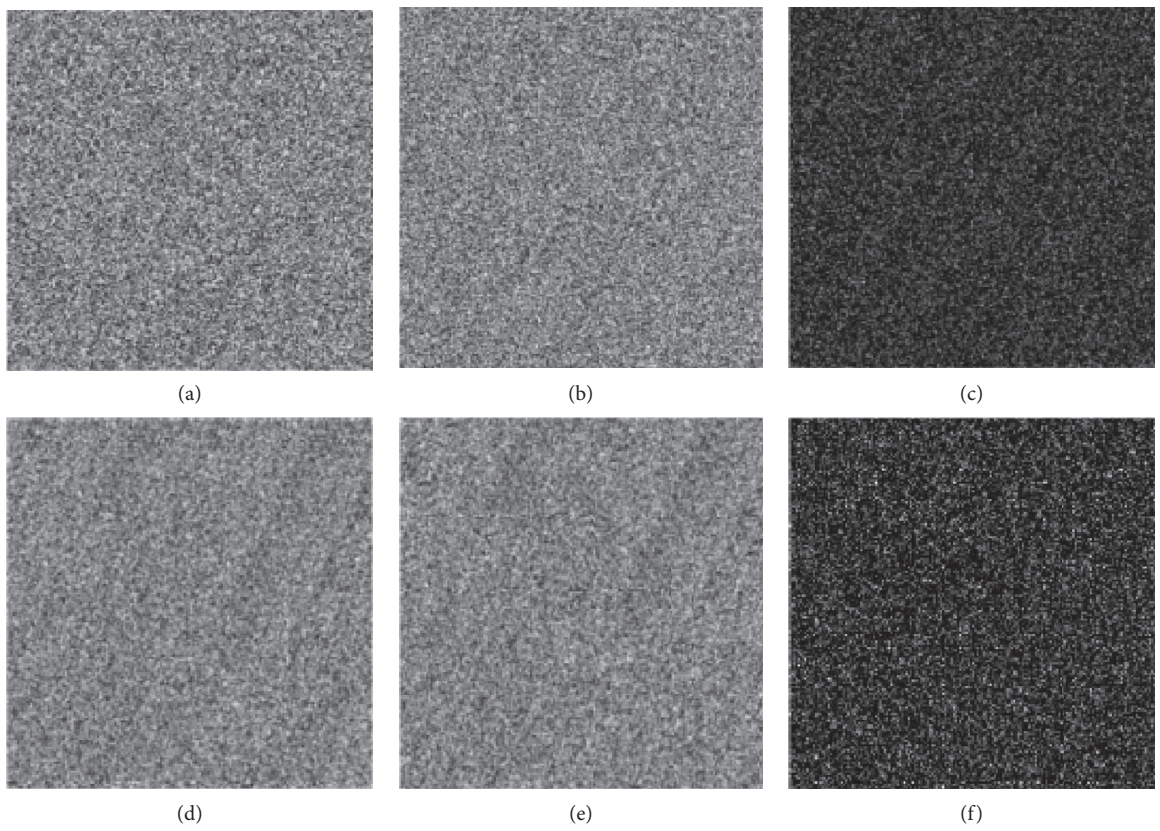


Figure 6: Key sensitivity test applied on the plain image: (a) encrypted original Lena, (b) encrypted modified Lena, (c) difference between (a) and (b), (d) decrypted (a) by key K_{i1}, (e) decrypted (b) by key K_{i2}, and (f) difference between (d) and (e).

Table 8: NC, NPCR, and UACI tests applied on the image.

Image	NPCR (%)	UACI (%)	NC
Lena	99.65950	33.71120	0.00157
Peppers	99.60556	33.64322	0.00109
Baboon	99.70664	33.80929	0.00370
Walkbridge	99.75660	33.27521	0.00102
Cameraman	99.60441	33.42789	−0.00259
Jetplane	99.60656	33.36627	−0.15490

Table 9: Comparative study of NPCR and UACI tests for the Lena image.

Image	NPCR (%)	UACI (%)
[7]	99.61000	33.4800
[8]	99.62500	33.451000
[9]	99.60780	33.45100
[12]	99.64163	33.39537
[23]	99.63230	33.12500
[24]	99.61000	33.53000
[29]	100.0000	33.43150
[26]	99.58948	33.46458
[30]	99.62530	33.48070
Proposed algorithm	99.65950	33.83002

pseudorandom numbers have good statistical properties such as being highly unpredictable, random, independent, and uniformly distributed.

3.3. Encryption Algorithm Speed. In real-time image processing, the execution time is a major constraint. In a software implementation, the speed of execution mainly depends on the CPU performance. The proposed algorithm is implemented using Matlab R2017a software running on a personal computer with CPU Intel Core-i7-3770 3.4 GHz frequency. We can use approximate equations (20) and (21) to compute the speed (S) and the number of cycles per byte (CpB) taken by an encryption algorithm running on a specific processor [18].

$$S = \frac{\mathrm{DS}}{T}\frac{\mathrm{MB}}{s}, \tag{20}$$

$$CpB = \frac{CpS}{S}. \tag{21}$$

DS is the data size, T is the time taken to execute the algorithm on a CPU, and CpS is the CPU frequency. We compare the proposed algorithm with recent works presented in Table 11.

It is clear that the proposed scheme has the best result in terms of speed and architecture complexity. We note that the permutation and substitution operations depend on more time than diffusion operation. Although, the proposed cryptosystem uses only diffusion operation with low computational complexity, and it gives the best performance.

Table 10: Test results of the NIST 800-22 for the proposed PRNG.

Statistical	P value	Status
Status frequency	0.8755390	Pass
Block frequency ($m = 128$)	0.3924558	Pass
Forward cusum	0.6371194	Pass
Reverse cusum	0. 6371194	Pass
Runs	0. 6371194	Pass
Long runs of ones	0. 6371194	Pass
Binary matrix rank	0.4372742	Pass
Spectral DFT	0. 8755390	Pass
Nonoverlapping template ($m = 9$)	0. 1463590	Pass
Overlapping template ($m = 9$)	0.5449921	Pass
Universal	0.0713232	Pass
Approximate entropy ($m = 10$)	0.0125474	Pass
Random excursions ($x = +1$)	0.9030558	Pass
Random excursions variant ($x = -1$)	0.3974291	Pass
Linear complexity ($M = 500$)	0.1922722	Pass

Table 11: Comparative study of encryption algorithm speed.

Work	Used tools	Architecture	S (MB/S)	CpB
[4]	AES-128 CBC	Xor diffusion	0.002	1400000
[8]	Logistic-Tent and Tent-Sine	Confusion/diffusion	0.190	10520
[9]	3D sine chaotic map	Confusion/diffusion	0.970	3298
[12]	SHA-256/Liu system	Xor diffusion	1.500	1600
[10]	MD5/PWLCM/DNA	Confusion/diffusion	7.540	398
[11]	SHA-3/Tent map/DCT	Transformation/permutation/diffusion	0.100	20000
[25]	Tent map	Xor diffusion	0.140	16142
[26]	Chebyshev map/A. Tent map/LWT/LFT	Permutation/substitution/diffusion	0.019	147368
Proposed algorithm	SHA-256/CTR/PRNG	Xor diffusion	9.567	355

4. Conclusions

In this work, we have suggested an efficient cryptosystem for numerical image protection and authentication. The proposed algorithm is a hybrid scheme, which combines a symmetric and an asymmetric scheme. The SHA-256 is utilized for initial secret key generation related to the original image and the owner's signature. The RSA asymmetric encryption system is used for encrypting the initial and owner's signature that permit secret key exchanging and authentication. A PRNG-CTR-based Xor confusion is adopted as the symmetric scheme for entire image encryption. The PRNG-CTR is based on the Lorenz chaotic system which is designed to generate a high-quality key. The evaluation and analysis results demonstrate that the algorithm offers high performance and enhanced security with low computational complexity. The PRNG-CTR is tested by the NIST, and the result indicates that it is suitable for image encryption.

We noted that the proposed architecture has a many advantages such as flexibility for key extension which leads to increase the key space. In addition, this architecture is easily parallelizable to speed up execution for meeting real-time application requirements.

The comparative study with recent work indicates that the proposed algorithm provides the best performance. However, it is extremely adapted for image encryption and authentication, which can be used in several domains.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

All the authors helped to conceive these simulation experiments. Mohamed Gafsi designed and performed such experiments. Indeed, both Mohamed Gafsi and Mohamed Ali Hajjaji have written the main part of the manuscript. Mohamed Gafsi, Jihene Malek, and Abdellatif Mtibaa contributed to the interpretation of the results, as well as the revision and writing of the paper.

Acknowledgments

This work was supported by the EuE Laboratory.

References

[1] M. Gafsi, S. Ajili, M. A. Hajjaji, J. Malek, and A. Mtibaa, "High securing cryptography system for digital image transmission," *Smart Innovation: Systems and Technologies*, vol. 146, pp. 311–322, 2020, https://www.scopus.com/sourceid/21100204111?origin=recordpage.

[2] S. Ajili, M. A. Hajjaji, and A. Mtibaa, "Combining watermarking and encryption algorithm for medical image safe transfer: method based on DCT," *International Journal of Signal and Imaging Systems Engineering*, vol. 9, no. 4-5, pp. 242–251, 2016.

[3] S. Toughi, M. H. Fathi, and Y. A. Sekhavat, "An image encryption scheme based on elliptic curve pseudo random and advanced encryption system," *Signal Processing*, vol. 141, pp. 217–227, 2017.

[4] Z. Yong and L. Xueqian, "A fast image encryption scheme based on AES," in *Proceedings of the 2th International Conference on Image, Vision and Computing*, IEEE, Chengdu, China, June 2017.

[5] A. Uhl and A. Pommer, "Image and video encryption: from digital rights management to secured personal communication," *Advances in Information Security*, Springer, Berlin, Germany, 2005.

[6] M. A. Hajjaji, M. Dridi, and A. Mtibaa, "A medical image crypto-compression algorithm based on neural network and PWLCM," *"HajjajiDM19" Multimedia Tools and Applications*, vol. 78, no. 11, pp. 14379–14396, 2019.

[7] Y. Sha-Sha, "Optical image encryption algorithm based on phase-truncated short-time fractional Fourier transform and hyper-chaotic system," *Optics and Lasers in Engineering*, vol. 124, 2020.

[8] H.-S. Ye, "Multi-image compression-encryption scheme based on quaternion discrete fractional hartley transform and improved pixel adaptive diffusion," *Journal of Signal Processing*, vol. 175, 2020.

[9] J. Zhou, "Fast color image encryption scheme based on 3D orthogonal Latin squares and matching matrix," *Journal of Optics and Laser Technology*, vol. 131, 2020.

[10] L. Hongjun, "Chaos-based fast colour image encryption scheme with true random number keys from environmental noise," *Journal of Image Processing*, vol. 11, no. 5, pp. 324–332, 2017.

[11] A. Jawad, "A novel image encryption scheme based on orthogonal matrix, skew tent map, and XOR operation," *Journal of Neural Computing and Applications*, vol. 30, no. 12, pp. 3847–3857, 2018.

[12] H. Liu, "Image encryption using DNA complementary rule and chaotic maps," *Journal of Applied Soft Computing*, vol. 12, no. 5, pp. 1457–1466, 2012.

[13] C. Xiuli, "A color image cryptosystem based on dynamic DNA encryption and chaos," *Journal of Signal Processing*, vol. 155, pp. 44–62, 2019.

[14] G. Ramzi, "A novel chaos-based image encryption using DNA sequence operation and secure hash algorithm SHA-2," *Nonlinear Dynamics*, vol. 83, no. 3, pp. 1123–1136, 2016.

[15] N. Edward, "Deterministic non periodic flow," *Journal of the Atmospheric Sciences*, vol. 20, pp. 130–141, 1963.

[16] National Institute of Standards and Technology (NIST), *FIPS PUB 180-2: Secure Hash Standard (SHS)—Computer Security Standard*, National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA, 2001.

[17] R. L. Rivest, A. Shamir, and L. Adleman, *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*, pp. 120–126, Massachusetts Institute of Technology, Cambridge, MA, USA, 1978.

[18] A. Safwan El, "A new chaos-based image encryption system," *Journal of Signal Processing: Image Communication*, vol. 41, pp. 144–157, 2016.

[19] M. Dridi, "Cryptography of medical images based on a combination between chaotic and neural network," *Journal of Image Processing, IET*, vol. 11, no. 5, pp. 324–332, 2016.

[20] W. Zhou, "Image quality assessment: from error visibility to structural similarity," *IEEE Transaction on Image Processing*, vol. 13, no. 4, 2004.

[21] A. Melo, PriorityQoE: atool for improving the QoE in Video streaming, intelligent multimedia technologies for networking applications: techniques and tools.

[22] C. Unal, "Secure image encryption algorithm design using a novel chaos based S-box," *Journal of Chaos and Solitons and Fractals*, vol. 95, pp. 92–101, 2017.

[23] S. Suri, "An aes-chaos-based hybrid approach to encrypt multiple images," *Recent Developments in Intelligent Computing, Communication and Devices*, Springer, Berlin, Germany, 2017.

[24] W. Jiahui, "Color image encryption based on chaotic systems and elliptic curve eljamal scheme," *Journal of Signal Processing*, vol. 141, pp. 109–124, 2017.

[25] C. Li, "An image encryption scheme based on chaotic tent map," *Journal of Nonlinear Dynamics*, vol. 87, no. 1, pp. 127–133, 2017.

[26] B. Akram, "Chaos-based partial image encryption scheme based on linear fractional and lifting wavelet transforms," *Journal of Optics and Lasers in Engineering*, vol. 88, pp. 37–50, 2017.

[27] W. Zhang, K.-w. Wong, H. Yu, and Z.-L. Zhu, "An image encryption scheme using reverse 2-dimensional chaotic map and dependent diffusion," *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 8, pp. 2066–2080, 2013.

[28] W. Yue, "NPCR and UACI randomness tests for image encryption," *Journal of Selected Areas in Telecommunications*, 2011.

[29] A. Khalaf, "Fast image encryption based on random image key," *International Journal of Computer Applications*, vol. 3, 2016.

[30] Z. Rim, "Image encryption based on new beta chaotic maps," *Journal of Optics and Lasers in Engineering*, vol. 96, pp. 39–49, 2017.

[31] A. Rukhin, *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2010.