A Difference-Equation-Based Robust Image Encryption Scheme with Chaotic Permutations and Logic Gates

M. H. Annaby · H. A. Ayad · M. A. Rushdi

Received: 25 May 2021 / Accepted: 29 March 2022 / Published online: 19 May 2022 © The Author(s) 2022

Abstract

Image encryption has become an indispensable tool for achieving highly secure image-based communications. Numerous encryption approaches have appeared and demonstrated varying degrees of robustness to adversarial attacks. In this paper, an efficient and robust image encryption algorithm is established based on randomized difference equations, random permutations and randomized logic circuits. Specifically, hyperchaotic and chaotic systems are used to generate pseudo-random sequences. These sequences are thus used to define random first-order difference equations, chaotic permutations and logic circuits. Image encryption based on these three randomized modules shows high computational efficiency as well as strong robustness against statistical, differential, and chosen-plaintext attacks. The proposed scheme leads to almost zero correlation in the encrypted images, entropy values of more than 7.99 for the test images, and a key space size of $2^{572}$. Furthermore, differential analysis shows that the number of pixel change rate (NPCR) and the unified average change intensity (UACI) for the proposed technique are on average 99.61 and 33.35%, respectively.

Keywords Difference equations · Chaotic maps · Image encryption · Differential attacks · Chosen-plaintext attacks

1 Introduction

Image encryption is an essential operation for visual information hiding and secure image communication with a wide range of applications in digital communication [8], imaging [14], and cloud computing [29]. Based on information redundancy and plain-image pixel correlation, numerous encryption algorithms have been developed where the image pixels are essentially scrambled and encoded into cipher-image pixels using secret keys [22]. The strength of any encryption algorithm depends on how far it can satisfy two key properties: confusion and diffusion. On the one hand, a high-confusion encryption algorithm little or no correlation between a cipher-image and its corresponding plain-image. On the other hand, for a high-diffusion algorithm, any change in the secret key or the plain-image information is effectively propagated to the cipher-image. The encryption algorithms can be generally categorized into symmetric and asymmetric algorithms [7,34]. A symmetric encryption algorithm uses a single secret key for image encryption and decryption, while an asymmetric algorithm uses a public key for encrypting a plain-image, and a different private key for decrypting the corresponding cipher-image.

Many image encryption techniques have been developed over the past few decades in order to ensure better encryption strength, efficiency, and robustness. Indeed, different mathematical and statistical tools have been exploited in order to achieve these goals through introducing higher degrees of randomness in the encryption keys and operations. For example, Annaby et al. [1] proposed highly secure image encryption algorithms by extending the construction of the discrete fractional Fourier transform (DFrFT) and introducing two generalized discrete transforms of both unitary and non-unitary types. Artiles et al. [6] employed a randomized variant of the Rijndael S-box to construct randomized-block image ciphers. Moreover, chaos and chaotic systems have been extensively explored to strengthen the encryption

---

1 Department of Mathematics, Faculty of Science, Cairo University, Giza 12613, Egypt
2 Department of Biomedical Engineering and Systems, Faculty of Engineering, Cairo University, Giza 12613, Egypt
confusion and diffusion properties. For instance, Hua et al. [18] introduced a cosine-transform-based chaotic system through which image pixels are scrambled with random-order substitution, and are then diffused with chaotic maps of complex dynamical behavior. Other encryption algorithms explored novel chaotic maps [33,39], compressive sensing [41], orthogonal transforms [10,35] and logic circuits [4].

Many image encryption algorithms in literature are based on shuffling image pixels using permutations and/or XOR operations. However, using these operations only in encryption has been shown as insecure against chosen-plaintext attacks [5,21,23,42]. In [5], Arroyo et al. designed a chosen-plaintext attack to break an encryption algorithm based on pixel shuffling and gray-level masking. In particular, for $M \times N$ images, one plain-image is chosen to remove the masking part, and then the procedure in [23] is followed to recover the shuffling map using $\lceil \log_{256} MN \rceil$ images. When the encryption scheme is based on permutation only, [21] proved that via choosing $\lceil \log_{L} MN \rceil$ plain-images of size $MN$ and $L$ different intensities, the chosen-plaintext attack completely determines the correct plaintext elements. Moreover, Huang et al. [19] demonstrated a cryptanalysis algorithm for encryption algorithms based on Latin cubes.

In this work, we establish a new image encryption scheme based on randomized first-order difference equations, chaotic permutations, and randomized logic circuits. It should be pointed out that the use of difference equations can be viewed as a diffusion process among pixels, rather than diffusion among blocks as in [13,17,25,36,38]. However, this diffusion process merely uses difference equations with constant coefficients. In our approach, we use a randomized difference equation, where the coefficients are not only variable keys, but are also generated using pseudorandom chaotic sequences. We implement hyperchaotic sequences to scramble pixel values, and hence construct chaotic gates. The robustness of the use of the first-order difference equations is rigorously discussed. In particular, we demonstrate that the choice of the initial conditions plays a major role in strengthening the immunity of the encryption algorithm. In summary, the main contributions of this paper are as follows:

- A novel efficient and robust image encryption scheme is presented.
- For the first time, randomized difference equations are proposed for image encryption.
- The proposed algorithm utilizes both chaotic permutations and chaotic logic gates, whose parameters generated via hyper-chaotic maps.
- The robustness of the proposed difference-equation encryption scheme is demonstrated through a rigorous proof.

- Various statistical measures are used to evaluate the performance outcomes of the proposed scheme and compare them with those of other state-of-the-art methods.

The paper is designed as follows. The mathematical tools used in the proposed image encryption technique are introduced in Sect. 2. In Sect. 3, the details of the encryption and decryption algorithms are introduced, in addition to the specification of the employed encryption keys. Section 4 presents the simulation and statistical analysis results for the proposed encryption algorithm. Conclusions of this work are given in Sect. 5.

2 Encryption Tools

This section introduces the basic tools required to build up a robust and efficient encryption technique. These tools are basically chaotic (or hyper-chaotic) difference equations, reversible logic gates, and random logic circuits.

2.1 Chaotic Systems

For the purpose of information security, there is an essential need for pseudo-random number generators that yield pseudo-random number sequences. These sequences are employed as encryption keys in image encryption algorithms. Many chaotic systems have been proposed in the literature to generate chaotic number sequences.

In [12], Dong et al. proposed a new family of Hamiltonian conservative chaotic systems. In particular, a 5D conservative hyper-chaotic system (CCS) was introduced with two non-hyperbolic equilibrium points, nonzero initial values, and coexisting quasi-periodic and hyper-chaotic orbits. This 5D system is defined by the following equations:

$$\begin{align*}
\dot{x} &= ay + cv, \\
\dot{y} &= -ax + bxw, \\
\dot{z} &= dw, \\
\dot{w} &= -bxy - dz, \\
\dot{v} &= -cx.
\end{align*}$$

Fixing the parameters $a, b, c, d$ of the system in (1), and the initial values in some dimensions and varying the other dimensions causes the dynamic flow to evolve and switch among hyper-chaos, chaos, and quasi-periodic motions. The effects of the initial values of $(x, y, z, w, v)$ on the system dynamics have been analyzed in [12]. As previously mentioned, coexisting orbits have been founded, for special values or ranges of some dimensions. The dynamic properties of the system in [12] were verified by several criteria, namely dynamical evolution maps, equilibrium points, and the Lya-
punov dimension. Additionally, the pseudo-randomness of his chaotic system has been validated through NIST [27]. The implementation details of a chaotic system similar to that in (1) are mentioned in [12]. In our work, we employ the system in (1) with a hyper-chaotic orbit by choosing the following parameter settings $a = 30, b = 30, c = 10,$ and $d = 30$, as well as the initial values $x = -0.1, y = 2.45, z = -0.1, w = 1.85,$ and $v = -0.1$.

Since the proposed encryption system requires six chaotic sequences, we generate a sixth randomized sequence using a logistic map with a bifurcation parameter $\mu$ and an initial point $x_0$

$$q_{j+1} = \mu q_{j}(1-q_{j}), \quad 0 < \mu < 4, \quad q_0 \in (0, 1),$$

where we do not consider $0 < \mu \leq \mu_0 \approx 3.569946\cdots$ because when $\mu \in (0, \mu_0)$, then for any initial value $q_0 \in (0, 1)$, the orbit $O(q_0) = \{q_n\}_{n=0}^{\infty}$ asymptotically falls onto $2^k$ points of $[0, 1]$ for some $k \in \mathbb{N}$ [2, 3, 11]. Nevertheless, as we select the bifurcation parameter $\mu \in (\mu_0, 4]$, one has to avoid the stability gaps, like the interval $3.825 < \mu < 3.86$, where for the initial values in $(0, 1)$, the orbits are attracting cycles of period 3, see [3, 11]. In our experimental setup, we take $\mu = 3.9$ and the initial value $q_0 = 0.5$ to generate a chaotic sequence.

### 2.2 Randomized Difference Equations

Consider the first-order equation:

$$\alpha_k y_k + \beta_k y_{k-1} + \gamma_k = 0, \quad k = 0, \ldots, N - 1,$$

(3)

$$\alpha_k \neq 0,$$

together with the initial condition

$$y_{-1} = \eta.$$  

(4)

Here $\alpha_k, \beta_k, \gamma_k, \eta$ are randomly generated numbers (generally complex numbers). However, we select them to be real numbers.

The coefficients $\alpha_k, \beta_k, \gamma_k$ and the initial condition value $\eta$ are randomly selected via the systems described in the previous section. They will be used as keys of the encryption process.

By slightly modifying the technique of [24], a general solution of the Cauchy problem (3) and (4) can be written as:

$$y_k = u_k + h_k,$$

(5)

where $u_k$ is the solution of the homogeneous problem, $\gamma_k \equiv 0$, i.e.,

$$u_k = (-1)^{k+1} \eta \prod_{i=0}^{k} \frac{\beta_i}{\alpha_i},$$

(6)

and $h_k$ is any particular solution of (3) and (4). Again by a slightly different analysis of [24], we obtain

$$h_k = u_k \sum_{m=0}^{k} \frac{\gamma_m}{\beta_m u_{m-1}}, \quad k = 0, \ldots, N - 1.$$  

(7)

Consequently

$$y_k = \begin{cases} \eta, & k = -1, \\ u_k \left[1 + \sum_{m=0}^{k} \frac{\gamma_m}{\beta_m u_{m-1}}\right], & k > 0, \end{cases}$$

(8)

where $k = 0, \ldots, N - 1$. A detailed analysis of using (8) in encryption is established in Sect. 3.

### 2.3 Chaotic Circuits

We mean here by chaotic circuits, any logical gate or circuit with controlling bits determined via chaotic systems, such that the controlled-NOT gate in Fig. 1. This gate takes two input bits $x$ and $y$, and outputs the bits $x$ and $z = x \oplus y$. The $x$ bit in this gate acts as a controlling bit, where if $x = 0$, then the output bit $z = y$, and when $x = 1$, that implies that $z = \overline{y}$. For image encryption, the controlling-bit gate can be applied by generating the $x$ bits by a chaotic system, to ensure the randomness of the bit stream, and specifying the $y$ bits as the plaintext image bits. Then, the ciphertext image is obtained by applying the controlled-NOT gate, such as the ciphertext image bits are the output bits of XOR gate $z = x \oplus y$, in case the input bits of the XOR gate are $x$ and $y$, as is previously specified.

We should note that the controlling-NOT gate is used in this article; however, any reversible gate can be applied, not only with 1 controlling bit, but also with 2 controlling bits like Fredkin gate. Some analysis about the security of applying the reversible gates, rather than the irreversible gates, in image encryption, is mentioned in [2].

### 3 Proposed Cryptosystem

#### 3.1 Using Difference Equations

In this subsection, we discuss the use of the first-order difference Eqs. (3) and (4) as well as its solution (8) as an
encryption tool. In addition, we will discuss its robustness against chosen plaintext attacks.

Assume that we have an array \((\xi_1, \xi_2, \ldots, \xi_{N-1}) \in \mathbb{R}^N\), and we want to encrypt it via a first-order difference equation. We assume that \(\alpha = (\alpha_k), \beta = (\beta_k), \gamma = (\gamma_k)\) and \(\eta\) are randomly chosen, \(\alpha_k \neq 0\). We encrypt \((\xi_1, \xi_2, \ldots, \xi_{N-1})\) via

\[
\xi_k \mapsto r_k, \quad k = 0, \ldots, N-1,
\]

where

\[
r_k = T(\xi_k) = \alpha_k \xi_k + \beta_k \xi_{k-1} + \gamma_k, \quad \xi_{-1} = \eta.
\]  

(9)

An illustration of the above equation is shown in Fig. 2. The decryption step is carried out through (8) to recover \(\xi_k\) back from the cipher \(r_k\) to obtain

\[
\xi_k = \begin{cases} 
\eta, & k = -1, \\
(−1)^{k+1} \eta \prod_{l=k}^{k+m-1} \beta_l \xi_l + \sum_{m=0}^{k} \frac{\gamma_m}{\beta_m \beta_{m+1}}, & k > 0,
\end{cases}
\]  

(10)

where \(k = 0, \ldots, N - 1 \) and \((u_m)_{m=1}^{N-1}\) is given in (6).

The robustness of the encryption-decryption system (9, 10) against chosen-plaintext attacks depends on the random initial value \(\eta\) as well as the initial values \(\beta_0\) and \(\gamma_0\). The other values can be recovered via chosen plaintext attacks. In fact, knowing the effect of the cipher (9, 10) on four chosen-plaintexts suffices to recover the values of \(\alpha_k, \beta_k, \gamma_k\) except for \(\beta_0\) and \(\gamma_0\). Let \(\xi_k^1, \xi_k^2, \xi_k^3\) and \(\xi_k^4\) be the arrays

\[
\begin{align*}
\xi_k^1 &= 0, \quad k = 0, \ldots, N - 1, \\
\xi_k^2 &= 1, \quad k = 0, \ldots, N - 1, \\
\xi_k^3 &= \begin{cases} 
0, & k = 2m, 0 \leq m \leq \lfloor N/2 - 1/2 \rfloor, \\
1, & k = 2m + 1, 0 \leq m \leq \lfloor N/2 - 1 \rfloor,
\end{cases}
\end{align*}
\]

(11) where \(k = 0, \ldots, N - 1\). Then, the parameters \(\gamma_k, \beta_k, \alpha_k\) are recoverable according to the following proposition.

**Proposition 1** Assume that \(r_k^{(j)} = T(\xi_k^{(j)}), j = 1, 2, 3, 4\) are known, and \(k = 1, \ldots, N - 1\). Then,

\[
\begin{align*}
\gamma_k &= \xi_k^{(1)}, \\
\beta_k &= \begin{cases} 
\xi_k^{(3)} - \gamma_k, & k = 2m, 1 \leq m \leq \lfloor N/2 - 1 \rfloor, \\
\xi_k^{(4)} - \gamma_k, & k = 2m + 1, 0 \leq m \leq \lfloor N/2 - 1 \rfloor,
\end{cases}
\end{align*}
\]

(15) and for \(k = 0, \ldots, N - 1\),

\[
\begin{align*}
\alpha_k &= \begin{cases} 
\xi_k^{(2)} - \xi_k^{(1)}, & k = 0, \\
\xi_k^{(3)} - \gamma_k, & k = 2m, 1 \leq m \leq \lfloor N/2 - 1 \rfloor, \\
\xi_k^{(4)} - \gamma_k, & k = 2m + 1, 0 \leq m \leq \lfloor N/2 - 1 \rfloor.
\end{cases}
\end{align*}
\]

(16) \(\square\)

**3.2 Encryption Algorithm**

The proposed encryption technique of this paper consists of three different stages, implementing the tools briefed in Sect. 2. The first stage applies the difference equation approach among plaintext pixels, while the second and third steps apply chaotic permutation and chaotic logic gates or circuits, respectively. The cipher might be made more complicated by applying randomized orthogonal transforms, see, e.g., [2].

The algorithm is illustrated in Fig. 3 and can be summarized as follows:

**Step 1.** In this stage, we apply the procedure (3, 4) column-wise on the plaintext image. Let \(P\) be the \(N - 1 \times N - 1\) plaintext image, which may be written as:

**Fig. 2** Illustration of the application of (9) onto the input signal \(\xi\) to output \(r\) using the keys \(\alpha, \beta, \gamma, \eta_{-1}\)
Fig. 3 Illustration of the three steps in the encryption algorithm

\[ P = (P_0, P_1, \ldots, P_{N-1}), \]

where \( P_j \) is the column \( P_j = (p_{0,j}, p_{1,j}, \ldots, p_{N-1,j})^T \), \( A^T \) is the transpose of a matrix \( A \), \( p_{x,y} \) is the value of pixel at the \( x \)th row of the \( y \)th column. To make the encryption stage of applying (3, 4) as robust as possible, we may apply (3, 4) \( n \) times with the coefficients \( \alpha(l), \beta(l), \gamma(l), \eta(l), l = 1, \ldots, n \). These coefficient parameters are considered as encryption keys, and they are randomly chosen. The random choice of the parameters \( \alpha, \beta, \gamma, \eta \) applied on each column of the plaintext is different. For each round of application, these parameters can be fixed, or variant for each round. Thus, we apply (3, 4) on each column through

\[ \mathcal{T}(P_j) = T(T^{n-1}(P_j)), \tau = 2, \ldots, n, \]

\[ j = 0, \ldots, N-1, \]

and

\[ T^1(P_j) = \alpha_k p_{kj} + \beta_k p_{k-1,j} + \gamma_k, 0 \leq k \leq N-1. \]

According to the performance analysis demonstrated below, we found that \( n = 6 \) is a possible minimum of rounds to guarantee robustness. Let \( T_j = T^n(P_j) \), \( j = 0, \ldots, N-1 \), and \( T \) be the \( N \times N-1 \) matrix, such that \( T = (T_1, \ldots, T_{N-1}) \). In the following step, we apply permutation of the pixels of \( T \).

**Step 2.** The permutation step is performed by shifting each column, \( T_j \), of \( T \), \( j = 0, \ldots, N-1 \), circularly, by
\( y_j \in \{0, 1, \cdots, N-1\} \) units, via the translation operator \( R \) as:

\[
(R_y T_j)(i) = T_j(i - y_j) = \tilde{T}_j, \quad i = 0, \cdots, N-1, \quad (21)
\]

with period \( N \). Let \( \tilde{T} \) be the \( N-1 \times N-1 \) matrix in which the \( j \)th column is \( R_y T_j, \quad j = 0, \cdots, N-1 \). Then, apply the above operator \( R \) on the rows of \( \tilde{T}, s_j, j = 0, \cdots, N-1 \), to circularly shift each \( s_j \) by \( x_j \in \{0, 1, \cdots, N-1\} \) units, via

\[
(R_x S_j)(i) = S_j(i - x_j) = \tilde{S}_j, \quad i = 0, \cdots, N-1, \quad (22)
\]

Let \( \tilde{S} \) be the \( N-1 \times N-1 \) in which the \( j \)th row is \( R_x S_j, j = 0, \cdots, N-1 \). In the following step, the XOR gate is applied on the pixels of \( \tilde{S} \).

**Step 3.** The XOR gate is used to scramble each pixel \( \tilde{s}_{i,j} \) with a random key \( e_{i,j} \in E \), such that \( E \) is an \( N-1 \times N-1 \) matrix, and produce \( c_{i,j} \) via

\[
c_{i,j} = \tilde{s}_{i,j} \oplus e_{i,j}, \quad 0 \leq i, j \leq N-1. \quad (23)
\]

Thus, \( C \) is the ciphertext image corresponding to \( P \).

The first step in the above encryption algorithm is applied five rounds on the 256 \times 256 Cameraman plaintext image shown in Fig. 4a to produce Figs. 4b–f. As can be shown, after the first round of application, the details of the plaintext features still can be recognizable. However, as many rounds of applications are performed, the features diminish until it cannot be possible to extract related features from the plaintext. Steps 2 and 3 of the encryption algorithm representing the permutation and XORing are applied on Fig. 4f to enhance the encryption security and produce the ciphertext image shown in Fig. 4h. The random appearance of pixels can be shown after a few rounds of the difference equation on the plaintext image and after the permutation and XORing applications.

The choice of the number of rounds of the first encryption step is arbitrary. In order to investigate a suitable number of rounds after which there is almost no correlation between the plaintext image, we applied (9) a test set of 10 plaintext images of size 256 \times 256 for rounds 1 through 10. After each round, we calculate the average correlation coefficient value between the resulting images and the plaintext images. Figure 5 shows how much the correlation after each round of application. As can be seen, the correlation coefficient value decreases as the number of rounds increases. We have found that when the number of rounds is 6, the correlation value becomes negative and tremendously small, and it continues to be negative and smaller for further applications. Therefore, repeating the first encryption step for 6 rounds is an adequate number before applying the permutation and XORing in steps 2 and 3. The average mean square error after each round of application of the difference equation only to the plaintext images can be shown in Fig. 6. After the first round of application, a small error is found; however, the error increases as...
when only (9) is applied 1 through 10 times. The ciphertext images over a test set of 10 plaintext images of size 256 × 256, which are the inverse of the permutation map utilized in the encryption algorithm. The second step is to permute the columns and rows of the output matrix to undo the scrambling effect by the logical gate. The details of implementing each step are as follows.

Step 1 Let C be the N − 1 × N − 1 ciphertext image. In this stage, we reverse the XOR logical gate applied in Step 3 in the encryption algorithm. Since the inverse of XOR is itself, then S can be retrieved via

\[ \tilde{s}_{i,j} = c_{i,j} \oplus e_{i,j}, \quad 0 \leq i, j \leq N - 1, \]  \hspace{1cm} (24)

where \( e_{i,j} \) is the \((i, j)\) element in the key matrix \( E \) (23). In the following step, the permutation step is reversed by rearranging the pixels of \( \tilde{S} \).

Step 2 Since the permutation in the encryption step is applied by shifting the columns, and then the rows, then in this stage, the rows are shifted, first, and then the columns, in the opposite direction, by the same permutation map as used in the encryption algorithm. Construct the matrix \( \tilde{T} \) of size \( N - 1 \times N - 1 \), in which the \( j \)th row, \( S_j, 0 \leq j \leq N - 1 \) circularly via

\[ S_j = \tilde{S}(i + x_j), \quad i = 0, \ldots, N - 1. \]  \hspace{1cm} (25)

Then, the columns of \( \tilde{T} \), denoted by \( \tilde{T}_1, \tilde{T}_2, \ldots, \tilde{T}_{N-1} \), are circularly shifted, to yield \( T_1, T_2, \ldots, T_{N-1} \) via

\[ T_j = \tilde{T}(i + y_j), \quad i = 0, \ldots, N - 1, \]  \hspace{1cm} (26)

where \( x_j, y_j \) are as chosen in (21)-(22). Let \( T = (T_1, T_2, \ldots, T_{N-1}) \). In the following step, (10) is applied on each \( T_j, 1 \leq j \leq N - 1 \), to yield the columns of \( P = (P_0, P_1, \ldots, P_{N-1}) \).

Step 3 If (19) is applied \( n \) times such that \( T^n(P_j) = T_j \), then \( P_j = T^{-n}(T_j) \), where \( T^{-1} \) is the converse of \( T \). Since \( T_j = (t_{0,1}, t_{0,2}, \ldots, t_{0,N-1}) \), then from (10), \( T^{-1}(T_j) \) can be obtained via

\[ T^{-1}(k,j) = \begin{cases} \eta_j, & k = -1, \\ (-1)^{k+1} \frac{t_{0,j}}{d} \prod_{l=0}^{k-1} \frac{\theta_{l+j}}{d} (1 + \sum_{m=0}^{k-1} \gamma_{m+j} \eta_{m-j} \prod_{a=m}^{k-1} \frac{d_{a+j}}{d_{a-j}}), & k > 0. \end{cases} \]  \hspace{1cm} (27)

### 3.4 Encryption Keys

The chaotic systems’ streams of (1, 2) are used to generate the parameters of the randomized difference equation, the permutation map, and the logic gate implemented in our encryption algorithm. Therefore, the parameters and initial values of these systems are the secret keys of the encryption system. The chaotic system in (1) requires 4 parameters \( a, b, c, d \), and 5 initial values for \( x, y, z, w, v \) to generate pseudorandom sequences, while the system (2) requires one parameter \( \mu \) and one initial value \( q_0 \). Therefore, the total
number of keys in this system is 11. These parameters and initial values should be known by the sender and receiver for the purpose of encryption and decryption.

As we have mentioned, we will use the chaotic maps (1) and (2) to generate six random sequences for implementing the three mentioned stages of encryption. For this purpose, we need to preprocess the obtained chaotic sequences.

Let \( X, Y, Z, W, V \) be the chaotic sequences generated by the system variables \( x, y, z, w, v \). As in [37], a preprocessing step is used to improve the statistical randomness properties, before employing the sequences \( X, Y, Z, W, V \) in the encryption procedure. The preprocessing equation applied on the \( X \) sequence, for instance, similar to the formula in [37], is

\[
X[i] = 10^9 x[i] - \lfloor 10^n x[i] \rfloor + 0.5, \quad i \in \mathbb{N},
\]

where \( X[i] \) is the \( i \)th element in the \( X \) chaotic sequence corresponding to the \( i \)th generated value of the system variable \( x \), when \( x \) is discretized, and \( n \) is the shifted backward digits of the decimal part. Similarly, \( (28) \) is applied on the other sequences \( Y, Z, W, V \). Additional information about the statistical randomness properties achieved after applying the preprocessing step, and illustrations of the preprocessed chaotic sequences are found in [37].

Let \( Q \) be the chaotic sequence generated by \( (2) \). Then, we have the sequences \( X, Y, Z, W, V, Q \) to be used in encryption algorithm. The randomized encryption keys \( \alpha, \beta, y, \eta \) to implement the difference equation step are taken from the sequences \( X, Y, Z, W, V \), respectively. For the second permutation step in the encryption algorithm, the sequence \( V \) is used, while the sequence \( Q \) is used to determine the random key of the XOR gate in the third step of the encryption algorithm.

### 3.5 Encryption Methods

In Sect. 3.2, we have introduced an encryption algorithm which utilizes many methods such as randomized difference equation, chaotic permutations and logic gates (XOR). In this section, we discuss the function and contribution of each method in our encryption algorithm. As noted in the introduction of this paper, the use of permutation and XOR operations only is insecure against the chosen-plaintext attack. Then, we have applied the randomized difference equation whose parameters are chaotic in the encryption process to resist such attacks. We have theoretically proved that the randomized equation is robust as the initial conditions \( \eta \) as well as the initial values \( \beta_0 \) and \( \gamma_0 \) which are randomly chosen are unrecoverable while the other parameters of difference equation \( (\alpha, \beta, y) \) can be recovered. After the application of the difference equation, we have used the chaotic permutations and logic gates to enhance the confusion and statistical properties of the cipher-text image. Since XOR is a balanced gate

\[ r_{xy} = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n \sum x_i^2 - (\sum x_i)^2} \sqrt{n \sum y_i^2 - (\sum y_i)^2}}. \]
where $r_{xy} \in [-1, 1]$. A value of $r_{xy} = 1$ indicates the strongest positive correlation between samples of $X$ and $Y$, whereas $r_{xy} = -1$ reflects the strongest negative correlation. In addition, a correlation value of 0 indicates that no correlation exists.

For correlation analysis, we examine how much adjacent pixels in a ciphertext image are correlated. Ciphers should ideally exhibit no correlation between ciphertext adjacent pixels, while the adjacent pixels in plaintext images generally have high correlation coefficients. We chose random samples of 5000 pairs of adjacent pixels in the horizontal, vertical and diagonal directions. Then, we estimated the correlation in each of these directions by computing $r_{xy}$ according to (29). Figure 8 shows plots of the gray-level values of adjacent pixels for plaintext and ciphertext of the Lena image. While the plaintext image shows linear correlation in all directions, the ciphertext image shows completely random patterns with no clear correlation. These visual observations are supported by the correlation coefficient values, computed for 10 test images in Table 1. For each test image, the correlation coefficient is almost zero, indicating no correlation between the ciphertext pixels.

Moreover, Table 2 compares the average correlation coefficients obtained for standard test images after encryption by
Horizontally adjacent pixels in plaintext. (a) Horizontally adjacent pixels in ciphertext. (d) Horizontally adjacent pixels in ciphertext.

Vertically adjacent pixels in plaintext. (b) Vertically adjacent pixels in ciphertext. (e) Vertically adjacent pixels in ciphertext.

Diagonally adjacent pixels in plaintext. (c) Diagonally adjacent pixels in ciphertext. (f) Diagonally adjacent pixels in ciphertext.

Table 1  Correlation coefficient results for images encrypted with the proposed method

| Plaintext image | Correlation coefficient | Horizontal | Vertical | Diagonal |
|-----------------|--------------------------|------------|----------|----------|
| Baboon          | –0.0075                  | –0.0012    | –0.0142  |
| Barbara         | –0.0054                  | –0.0092    | 0.0130   |
| Boat            | –0.0069                  | –0.0028    | 0.0094   |
| Cameraman       | –0.0106                  | 0.0117     | –0.0185  |
| Cat             | 0.0015                   | –0.0084    | 0.0058   |
| Fingerprint     | 0.0072                   | 0.0095     | 0.0016   |
| House           | –0.0151                  | –0.0058    | –0.0034  |
| Lena            | –0.0068                  | –0.0091    | –0.0233  |
| Peppers         | 0.0064                   | –0.0137    | 0.0079   |
| Ruler           | –0.0244                  | –0.0050    | –0.0109  |

Table 2  Average correlation coefficients for adjacent pixels in ciphertext images produced by our encryption method and other relevant methods

| Algorithm          | Average correlation coefficient |
|--------------------|---------------------------------|
|                    | Horizontal | Vertical | Diagonal |
| Proposed algorithm | –0.0062    | –0.0034  | –0.0033  |
| Bashir et al. [9]  | –0.0063    | 0.0065   | –0.0016  |
| Hui et al. [20]    | –0.0073    | –0.0042  | 0.0049   |
| Shakiba et al. [28]| 0.003134   | 0.006602 | 0.004525 |
| Wang et al. [30]   | 0.0084     | –0.0039  | –0.0013  |
| Zhou et al. [38]   | 0.009463   | –0.024885| 0.002679 |
| Wang et al. [31]   | 0.0023     | –0.0020  | –0.0073  |

4.4 Structural Similarity

For image quality assessment, Wang et al. [40] introduced the index of structural similarity between an original image and a distorted image. This index is based on comparisons of local pixel intensity patterns, where the pixel intensities are normalized for luminance and contrast. The structural similarity between two images can be decomposed into three components (luminance, contrast and structure), which are respectively denoted by \( l(x, y) \), \( c(x, y) \), and \( s(x, y) \). Then, this similarity index can be mathematically defined as:
\[
S(x, y) = f(l(x, y), c(x, y), s(x, y)).
\]

(30)

The definitions of each of the \(l(x, y), c(x, y), \) and \(s(x, y)\) components and the conditions that \(S(x, y)\) should satisfy can be found in [40]. For two image patches \(x\) and \(y\), the structural similarity can be mathematically expressed as follows:

\[
S(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu^2_x + \mu^2_y + C_1)(\sigma^2_x + \sigma^2_y + C_2)},
\]

(31)

where \(\mu_x, \mu_y\) are the mean intensities of \(x\) and \(y\), while \(\sigma_x\) and \(\sigma_y\) are the corresponding standard deviations. Also, \(\sigma_{xy}\) denotes the covariance of \(x\) and \(y\), while \(C_1, C_2\) are constants. The similarity index as defined in (31) satisfies the conditions that \(S(x, y) \leq 1\) and that \(S(x, y) = 1\) iff \(x = y\) [40]. In our context, we use the index \(S\) to measure the similarity between an original plaintext image and the corresponding ciphertext image, where the former is assumed to be the reference image. Table 3 shows that the structural similarity values for 10 pairs of plaintext and ciphertext test images are \(\approx 0\). These results demonstrate a high encryption strength for the proposed encryption algorithm.

### 4.5 Entropy

Let \(O\) denote a zero-memory source of statistically independent random events (source symbols) from the possible set of events \(\{a_1, a_2, \ldots, a_J\}\). Each event \(a_j\) is generated with a probability \(P(a_j)\). The entropy \(H\) measures the information content averaged all sources [16], as expressed by the formula

\[
H = - \sum_{j=1}^{J} P(a_j) \log_2 P(a_j).
\]

(32)

In particular, if an image is an output of the source \(O\), such that each event \(a_j\) represents one of the image intensities, the entropy of the image can be defined as:

\[
\tilde{H} = - \sum_{k=0}^{L-1} p_r(r_k) \log_2 p_r(r_k),
\]

(33)

where \(r_k, p_r(r_k), \) and \(L\) are the \(k^{th}\) intensity value, the probability of occurrence of the \(k^{th}\) intensity value, and the highest intensity value, respectively.

In an image with random intensity values, the pixel intensities are supposed to be generated (emitted) with equal probabilities. Based on the definition of the image entropy in (33), \(0 \leq \tilde{H} \leq \log_2 L\), and \(\tilde{H} = 8\) for a completely random 8-bit image. Table 4 lists the entropy values of the ciphertexts of our test images. The entropy values are near optimal \((\approx 8)\), and this verifies the randomness and strength of our encryption algorithm.

Table 5 compares the entropy values obtained for standard test images after encryption by our method and four other methods. On average, our entropy results are the same or even slightly better than those of the compared methods. In fact, the average entropy obtained by our method matches the result returned by Hui et al. [20] and exceeds the results returned by the three other methods.

### 4.6 Differential Attacks

The number of changing pixel rate (NPCR) measures how much two images have different pixel intensities in corresponding locations. Also, the unified averaged changed intensity (UACI) measures the average absolute difference between two images. Let \(f_1\) and \(f_2\) be two \(M \times N\) images with intensities in the range \([0, L - 1]\). Then, the NPCR and UACI measures are defined, respectively, as follows [32]:

\[
\text{NPCR}(f_1, f_2) = \sum_{i,j} \delta_{i,j} \times 100\%,
\]

(34)

where \(\delta_{i,j} = \begin{cases} 0, & f_1(i,j) = f_2(i,j), \\ 1, & f_1(i,j) \neq f_2(i,j). \end{cases}\)

\[
\text{UACI}(f_1, f_2) = \sum_{i,j} \frac{|f_1(i,j) - f_2(i,j)|}{(L - 1)MN} \times 100\%.
\]

(35)

Differential cryptanalysis techniques (or differential attacks) investigate how small changes in the input plaintext image can affect the output ciphertext image. By tracing differences and discovering non-random patterns, an adversary can exploit the existing encryption properties to recover the encryption key. The NPCR and UACI measures are used to evaluate the strength of an encryption scheme against such
Table 4 NPCR, UACI and entropy results for ciphertext images obtained with the proposed algorithm

| Plaintext image | NPCR     | UACI     | Entropy |
|-----------------|----------|----------|---------|
| Baboon          | 99.5849  | 33.2960  | 7.9978  |
| Barbara         | 99.6490  | 33.3118  | 7.9976  |
| Boat            | 99.6490  | 33.3388  | 7.9977  |
| Cameraman       | 99.6276  | 33.4123  | 7.9975  |
| Cat             | 99.6002  | 33.3968  | 7.9976  |
| FingerPrint     | 99.5910  | 33.4293  | 7.9976  |
| House           | 99.5605  | 33.3400  | 7.9977  |
| Lena            | 99.6276  | 33.3249  | 7.9975  |
| Peppers         | 99.6200  | 33.3291  | 7.9977  |
| Ruler           | 99.6093  | 33.3575  | 7.9975  |

Table 5 Average entropy results obtained based on our encryption method and other relevant methods

| Algorithm                  | Average entropy |
|---------------------------|-----------------|
| Proposed algorithm        | 7.9976          |
| Ponuma et al. [26]        | 7.9941          |
| Wang et al. [31]          | 7.9962          |
| Xiang et al. [33]         | 7.9970          |
| Hui et al. [20]           | 7.9976          |

Table 6 Average NPCR, UACI comparison results

| Algorithm                  | NPCR     | UACI     |
|---------------------------|----------|----------|
| Proposed algorithm        | 99.61    | 33.35    |
| Zhou et al. [38]          | 99.60    | 33.41    |
| Wang et al. [30]          | 99.61    | 31.11    |
| Shakiba [28]              | 99.60    | 33.46    |

4.7 Computational Efficiency

We estimate the computational complexity of the proposed cipher as follows. We count the main operations used in each encryption stage, and the total operation count is used to find the overall cipher complexity. The difference-equation-based encryption stage needs \(2N(M - 1)\) multiplications, operations, while the permutation stage takes \(N + M\) shift operations in order to shift \(M\) rows and \(N\) columns. The bitwise XOR logical operations on each pixel in an 8-bit \(M \times N\) image reaches \(8MN\) operations. Thus, the proposed cipher has an overall time complexity of \(\approx 2N(M - 1) + N + M + 8MN = O(MN)\).

5 Conclusions

This paper proposed constructing image ciphers using the first-order linear difference operators along with chaotic permutations and reversible logic gates. All encryption parameters were randomly generated based on chaotic systems. A rigorous mathematical proof is given to show the robustness of the proposed difference-equation scheme against chosen-plaintext attacks. Indeed, breaking the proposed cipher requires the solution of an inverse problem, which is extremely difficult to do with chosen-plaintext attacks. Also, the proposed cipher is shown to be highly effective in comparison with other state-of-the-art methods. Our method is also robust against statistical and differential attacks. Furthermore, our work may be extended in many directions, such as utilizing higher-order linear and nonlinear difference operators. We may also incorporate orthogonal transforms into the cipher, and these transforms may be of different generalized types (such as the randomized, multiparameter or fractional transforms). Moreover, the proposed randomized encryption algorithm can be adapted and realized by digital circuits [15].

Funding Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.
A Appendix

Proof of Proposition 1. Since \( r_k^{(j)} = T(x_k^{(j)}) = \alpha_k x_k^{(j)} + \beta_k x_{k-1}^{(j)} + \gamma_k \), for any \( j \) and \( 0 \leq k \leq N - 1 \), if \( j = 1 \) and \( k > 0 \), then \( x_k^{(1)} = x_{k-1}^{(1)} = 0 \), and in this case \( \gamma_k = r_k^{(1)} \). Let us now denote the even entries of \( x, \alpha \), and \( \beta \) by \( x_{2k}, \alpha_{2k} \), and \( \beta_{2k} \). Also, denote the odd entries of \( x, \alpha \), and \( \beta \) by \( x_{2k+1}, \alpha_{2k+1} \), and \( \beta_{2k+1} \) for \( k = 0, \ldots, N - 1 \). Consider \( j = 3 \) and \( k = 1, 3, \ldots, N - 1 \) in the following equations:

\[
\begin{align*}
 r_k^{(3)} &= \alpha_k x_k^{(3)} + \beta_k x_{k-1}^{(3)} + \gamma_k, \\
 r_{k+1}^{(3)} &= \alpha_{k+1} x_{k+1}^{(3)} + \beta_{k+1} x_k^{(3)} + \gamma_{k+1}.
\end{align*}
\]

(36)

The even entries of \( x \) in (36 and 37), \( \xi_{k-1} = \xi_{k+1} = 0 \), and the odd entries of \( \xi, \xi_k = 1 \). Therefore, the odd entries of \( \alpha, \alpha_k \), and the even entries of \( \beta, \beta_{k+1} \) in (36 and 37) can be obtained as \( \alpha_k = r_k^{(3)} - \gamma_k \) and \( \beta_{k+1} = r_{k+1}^{(3)} - \gamma_{k+1} \). By a similar way, consider \( j = 4 \) and \( k = 1, 3, \ldots, N - 1 \) in the following equations:

\[
\begin{align*}
 r_k^{(4)} &= \alpha_k x_k^{(4)} + \beta_k x_{k-1}^{(4)} + \gamma_k, \\
 r_{k+1}^{(4)} &= \alpha_{k+1} x_{k+1}^{(4)} + \beta_{k+1} x_k^{(4)} + \gamma_{k+1}.
\end{align*}
\]

(39)

The even entries of \( x \) in (38 and 39), \( \xi_{k-1} = \xi_{k+1} = 1 \), and the odd entries of \( \xi, \xi_k = 0 \). Therefore, the odd entries of \( \beta \), \( \beta_{k-1} \), and the even entries of \( \alpha, \alpha_{k+1} \) in (38 and 39) can be obtained as \( \alpha_{k+1} = r_{k+1}^{(4)} - \gamma_{k+1} \) and \( \beta_k = r_k^{(4)} - \gamma_k \). Now, consider the case when \( k = 0 \), \( j = 1 \) and when \( k = 0, j = 2 \) in the following equations:

\[
\begin{align*}
 r_0^{(1)} &= \alpha_0 x_0^{(1)} + \beta_0 x_{-1}^{(1)} + \gamma_0, \\
 r_0^{(2)} &= \alpha_0 x_0^{(2)} + \beta_0 x_{-1}^{(2)} + \gamma_0.
\end{align*}
\]

(40)

Since \( \xi_0^{(1)} = 0 \) in (40), then \( r_0^{(1)} = \beta_0 x_{-1}^{(1)} + \gamma_0 \), and since \( \xi_0^{(2)} = 1 \) in (41), then \( \alpha_0 = r_0^{(2)} - (\beta_0 x_{-1}^{(2)} + \gamma_0) = r_0^{(2)} - r_0^{(1)} \).

\[ \square \]

References

1. Annaby, M., Ayad, H., Rushdi, M., Nehary, E.: Difference operators and generalized discrete fractional transforms in signal and image processing. Signal Process. 151, 1–18 (2018)
2. Annaby, M., Rushdi, M., Nehary, E.: Image encryption via discrete fractional fourier-type transforms generated by random matrices. Signal Process.: Image Commun. 49, 25–46 (2016)
3. Annaby, M., Rushdi, M., Nehary, E.: Color image encryption using random transforms, phase retrieval, chaotic maps, and diffusion. Opt. Lasers Eng. 103, 9–23 (2018)
4. Annaby, M.H., Ayad, H., Rushdi, M.A.: On security of image ciphers based on logic circuits and chaotic permutations. Multimed. Tools Appl. 77(16), 20455–20476 (2018)
5. Arroyo, D., Li, C., Li, S., Alvarez, G., Halang, W.A.: Cryptanalysis of an image encryption scheme based on a new total shuffling algorithm. Chaos, Solitons Fractals 41(5), 2613–2616 (2009)
6. Arviles, J.A., Chaves, D.P., Pimentel, C.: Image encryption using block cipher and chaotic sequences. Signal Process.: Image Commun. 79, 24–31 (2019)
7. Bao, Z., Xue, R., Jin, Y.: Image scrambling adversarial autoencoder based on the asymmetric encryption. Multimed. Tools Appl. 80(18), 28265–28301 (2021)
8. Bash, B.A., Goeckel, D., Towsley, D., Guha, S.: Hiding information in noise: fundamental limits of covert wireless communication. IEEE Commun. Mag. 53(12), 26–31 (2015). https://doi.org/10.1109/MCOM.2015.7355562
9. Bashir, Z., Iqbal, N., Hanif, M.: A novel gray scale image encryption scheme based on pixels’ swapping operations. Multimed. Tools Appl. 80(1), 1029–1054 (2021)
10. Bentouham, O., Hermassi, H., El-Latif, A.A.A., Belghith, S.: Cryptanalysis of a video encryption method based on mixing and permutation operations in the DCT domain. SIViP 9(6), 1281–1286 (2015)
11. Devaney, R.: An Introduction to Chaotic Dynamical Systems. Westview Press, Westview, Cambridge MA (2003)
12. Dong, E., Yuan, M., Du, S., Chen, Z.: A new class of hamiltonian conservative chaotic systems with multistability and design of pseudo-random number generator. Appl. Math. Model. 73, 40–71 (2019)
13. Faragallah, O.S.: Efficient confusion-diffusion chaotic image cryptosystem using enhanced standard map. SIViP 9(8), 1917–1926 (2015)
14. Ghafarli, H.M., Nodheh, A., Enayatifar, R.: An overview of encryption algorithms in color images. Signal Process. 164, 163–185 (2019)
15. Gong, L., Wu, R., Zhou, N.: A new 4d chaotic system with coexisting hidden chaotic attractors. Int. J. Bifurc. Chaos 30(10), 2050142 (2020)
16. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 4th edn. Pearson Education, United Kingdom (2018)
17. Gopalakrishnan, T., Ramakrishnan, S.: Image encryption using hyper-chaotic map for permutation and diffusion by multiple hyper-chaotic maps. Wireless Pers. Commun. 109(1), 437–454 (2019)
18. Hua, Z., Zhou, Y., Huang, H.: Cosine-transform-based chaotic system for image encryption. Inf. Sci. 480, 403–419 (2019)
19. Huang, R., Liu, H., Liao, X., Dong, A.: On the cryptanalysis of a latin cubes-based image cryptosystem.Entropy (2021). https://doi.org/10.3390/e23020202
20. Hui, Y., Liu, H., Fang, P.: A DNA image encryption based on a new hyperchaotic system. Multimed. Tools Appl. (2021). https://doi.org/10.1007/s11022-021-10526-7
21. Jolfaei, A., Wu, X.W., Muthukumarasamy, V.: On the security of permutation-only image encryption schemes. IEEE Trans. Inf. Forensics Secur. 11(2), 235–246 (2016). https://doi.org/10.1109/TIFS.2015.2498178
22. Kaur, M., Kumar, V.: A comprehensive review on image encryption techniques. Arch. Comput. Methods Eng. 27(1), 15–43 (2020)
23. Li, M., Guo, Y., Huang, J., Li, Y.: Cryptanalysis of a chaotic image encryption scheme based on permutation-diffusion structure. Signal Process.: Image Commun. 62, 164–172 (2018)
24. Maximon, L.C.: Differential and Difference Equations: A Comparison of Methods of Solution. Springer, Berlin (2016)
25. Patro, K.A.K., Acharya, B., Nath, V.: Various dimensional colour image encryption based on non-overlapping block-level diffusion operation. Microsyst. Technol. 26(5), 1–12 (2019)
26. Ponuma, R., Amutha, R.: Compressive sensing based image compression-encryption using novel 1D-chaotic map. Multimed. Tools Appl. 77(15), 19209–19234 (2018)
27. Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E.: A statistical test suite for random and pseudorandom number generators for cryptographic applications. Tech. rep., Booz-allen and hamilton inc mclean va (2001)
28. Shakiba, A.: A novel randomized bit-level two-dimensional hyper-chaotic image encryption algorithm. Multimed. Tools Appl. 79(43), 32575–32605 (2020)
29. Tari, Z., Yi, X., Premaratne, U.S., Bertok, P., Khalil, I.: Security and privacy in cloud computing: vision, trends, and challenges. IEEE Cloud Comput. 2(2), 30–38 (2015). https://doi.org/10.1109/MCC.2015.45
30. Wang, J., Zhi, X., Chai, X., Lu, Y.: Chaos-based image encryption strategy based on random number embedding and DNA-level self-adaptive permutation and diffusion. Multimed. Tools Appl. 80(10), 16087–16122 (2021)
31. Wang, X., Su, Y.: Image encryption based on compressed sensing and DNA encoding. Signal Process.: Image Commun. 95, 116246 (2021)
32. Wu, Y., Noonan, J.P., Agaian, S.: NPCR and UACI color image encryption using randomness tests for image encryption. Cyber journals: multidisciplinary journals in science and technology. J. Sel. Areas Telecommun. (JSAT) pp. 31–38 (2011)
33. Xiang, H., Liu, L.: An improved digital logistic map and its application in image encryption. Multimedia Tools and Applications 79(41) (2020)
34. Ye, G., Jiao, K., Huang, X.: Quantum logistic image encryption algorithm based on SHA-3 and RSA. Nonlinear Dyn. 104(3), 2807–2827 (2021)
35. Ye, H.S., Zhou, N.R., Gong, L.H.: Multi-image compression-encryption scheme based on quaternion discrete fractional Hartley transform and improved pixel adaptive diffusion. Signal Process. 175, 107652 (2020)
36. Zhang, Y., Xiao, D.: An image encryption scheme based on rotation matrix bit-level permutation and block diffusion. Commun. Nonlinear Sci. Numer. Simul. 19(1), 74–82 (2014)
37. Zhao, C.F., Ren, H.P.: Image encryption based on hyper-chaotic multi-attractors. Nonlinear Dyn. 100(1), 679–698 (2020)
38. Zhou, M., Wang, C.: A novel image encryption scheme based on conservative hyperchaotic system and closed-loop diffusion between blocks. Signal Process. 171, 107484 (2020)
39. Zhou, N.R., Huang, L.X., Gong, L.H., Zeng, Q.W.: Novel quantum image compression and encryption algorithm based on DQWT and 3D hyper-chaotic Henon map. Quantum Inf. Process. 19(9), 284 (2020)
40. Wang, Zhou, Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. 13(4), 600–612 (2004)
41. Zhu, L., Song, H., Zhang, X., Yan, M., Zhang, T., Wang, X., Xu, J.: A robust meaningful image encryption scheme based on block compressive sensing and svd embedding. Signal Process. 175, 107629 (2020)
42. Özçaynak, F., Özer, A.B.: Cryptanalysis of a new image encryption algorithm based on chaos. Optik 127(13), 5190–5192 (2016)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.