Novel pseudo random key & cosine transformed chaotic maps based satellite image encryption

Atul Kumar 1 · Mohit Dua 1

Received: 18 September 2020 / Revised: 14 April 2021 / Accepted: 20 April 2021 / Published online: 25 May 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

Image encryption converts the images into unrecognizable forms that seems like a white noise. Digital Chaos has also emerged as one of the important technique to design secure and efficient image encryption schemes. The chaos theory possesses various desirable properties required for the encrypting images like initial state sensitivity, unpredictability and behavioural complexity. However, image encryption schemes suffer from vulnerabilities like differential attack, statistical, known/chosen plaintext attack, and brute force attack. This paper proposes a novel approach to generate a pseudorandom key. This pseudo random key is combined with Logistic map (LM), cosine transformed Logistic map (CTLM) and cosine transformed Logistic-Sine Map (CTLSM), one by one, to implement three secure and efficient methods for satellite image encryption. The scheme uses the 384-bit of share key to perform encryption during the process. The proposed approaches are tested for parameters such as Entropy, Correlation Coefficient (CC), Number of Changing Pixel Rate (NPCR), Unified Averaged Changed Intensity (UACI), Avalanche effect, Bit Correct Ratio (BCR) and Peak Signal to Noise Ratio (PSNR). The work also analyses the various cryptanalytic attacks on the proposed chaos and novel Pseudo random key combinations. The results show that the proposed pseudo random key and CTLSM combination outperforms the other two combinations, and is more efficient in resisting all type of attacks as well.

Keywords Image encryption · Chaos theory · LM · CC · Pseudo-random · CTLM · CTLSM

Mohit Dua
er.mohitdua@nitkkr.ac.in

Atul Kumar
atul.kumar1995@gmail.com

1 Department of Computer Engineering, National Institute of Technology, Kurukshetra, Kurukshetra, India
1 Introduction

The exchange of digital images through various communicating networks creates the demand to secure the image through the communicating channels. Therefore, various methods have been proposed by various researchers to provide security to these images. Out of these encryption & watermarking are the two most frequent used techniques. In the watermarking [17], the image authenticates the identity of the sender, whereas in the encryption method [16, 36], the original image is converted into the noise like image. Chaotic theory is also one of the important concept to encrypt the images. A chaos map provides a nonlinear and random behavior at a particular interval, and also, it is sensitive to the initial conditions [27, 31, 48].

The chaos based image encryption scheme comprises of two stages, i.e. permutation and diffusion [14, 25, 65]. The permutation operation consists of shuffling the positions of the pixels of the image whereas, the diffusion phase consists of modifying the pixel value of the pixels. However, the traditional permutation and diffusion algorithms consist of only secret key, and it uses the same keystream for various plain images i.e. the secret key for number of images remains same. Hence, these algorithms suffer from known or chosen plaintext attack [13, 15, 39, 51, 66]. To defend these attacks, there have been various approaches or techniques developed by the researchers such as self-adaptive approach [28, 40, 58, 60], variable control parameters technique [38], dynamic state variable selection [9, 12], orbit variation [11, 49], and plaintext related techniques [44, 59]. All these schemes do change the keystreams during the implementation of the encryption process [21, 22, 37, 61].

The pseudo-random (PR) key performs essential role in the image encryption process. The benefit of pseudo-random key is, to provide the randomness to the shared key. Due to this, the different keystreams generate from the same share key during the encryption process, which results in different cipher images generation for the same plain-image. Hence, it becomes difficult for the adversary to compute the plain-image from the cipher image. There are various algorithms designed by the various researchers for the image encryption process. Section 1.1 describes the various existing schemes for the image encryption.

1.1 Related work

Chaos based image encryption is classified into two categories of the symmetric key cipher i.e. chaotic block cipher and chaotic stream cipher. Chaotic block ciphers are designed using the confusion and diffusion operations. There are various block ciphers that are designed for the image encryption using chaos theory [20, 24, 28, 46, 50, 61]. However, some of the block ciphers are vulnerable to various types of attacks [2, 9]. On the other hand, the stream ciphers encrypt individual bits of the image. The stream ciphers are generated using the XOR operation between the plain image and the keystream output of the key [4, 30, 52, 54, 57].

Liu et al. [41] in 2018, proposed a chaotic based scheme for medical image encryption that uses fourth order chaotic based system. This scheme consists of four initial conditions and six control parameters for encrypting the medical image. Bisht et al. [6] proposed an encryption scheme using the multiple chaotic map and fractional random transform. After that, Li et al. [37] in 2019, proposed image encryption approach using the dynamic state variable and orbit perturbation. This scheme uses the logistic adjusted sine map to perform the diffusion as well as permutation. Also, it depends upon the initial conditions and control parameters. The author claimed that their proposed scheme provides resistance from chosen plaintext attack.
Kovalchuk et al. [32–35] in 2019, proposed various works to improve the popular Rivest, Shamir, Adleman (RSA) method. RSA algorithm, in some cases, suffers from the weakness of leaving object outlines in an encrypted image, which in turn results in additional limitations while using it for image encryption [32]. Kovalchuk et al. [32], used topological image coverage concept with elements of RSA algorithm to overcome this limitation, and increase cryptostability of RSA approach. Kovalchuk et al. [33], applied projective transformation with RSA algorithm to propose two-element and element-by-element image encryption-decryption for both grayscale and color images. The modified proposed RSA approach not only avoids the contours appearance effect in encrypted images, but also is resistant to unauthorized decryption.

One more mathematical approach named, Quaternary Fractional-Linear Operations, was used by Kovalchuk et al. [34], in addition with RSA approach to prevent contours in encrypted image and increase RSA algorithm’s resistance from unauthorized decryption. In work of [35], Kovalchuk et al., combined RSA algorithm with bitwise binary operations to propose a more cryptostable approach and secure than basic RSA technique.

Researchers have also combined Sine and Cosine Functions with chaotic maps to improve dynamicity and complexity of the existing chaotic maps [3, 26]. Alwaida et al. [3] combined cosine function with LM and Henon chaotic maps to propose digital chaotic cosine map for cryptographic applications. In 2019, Hua et al. [27] proposed cosine transformation based scheme for image encryption. This scheme consists of three phases, where, the first phase consists of scrambling the image such that the correlation between the original and encrypted image gets decreased. In the second phase, the scrambled image is rotated by 90° in clockwise direction, and the last phase comprises of substitution of the image to perform the diffusion. The authors also designed three hybrid chaotic maps Logistic-Sine Cosine (LSC) map, Sine-Tent Cosine map (STC), and Tent-Logistic cosine (TLC) map. Ge et al. [23] in 2019, proposed a 3D cat map with Markov properties based scheme for image encryption. This scheme uses sum of all the binary bits as a secret key to defend against the known/chosen plaintext attack. The proposed approach analysed security in terms of key space analysis and differential attack.

Alawida et al. [2] in 2019, proposed a new chaos based encryption scheme based on perturbation. This scheme contains combination of two chaotic map to perform the encryption. The authors claimed that their method provides resistance from differential attack, known plaintext and chosen plaintext attacks. Saljoughi et al. [50] in 2019, proposed a 3D Logistic map for the image encryption that consists of three random sequences for the permutations. The researcher used the chaotic based encryption for the secure data transfer from the satellite communication. Bentoutou et al. [5] proposed image encryption using the two-dimension Logistic Adjusted Sine (LAS) map. The authors have also combined optimization methods with different chaos maps to improve efficiency of the existing encryption schemes [8, 18, 19].

Motivated by these works, the proposed work in this paper creates a three new image encryption schemes by combining the novel pseudo random key with the three different chaotic sequences, individually. Initially, this scheme generates a novel pseudo random key by applying various operations like $\text{XOR}$, left circular shift (Left Rotate), modular. Then, it combines the generated pseudo random key with Logistic map (LM), cosine transformed LM (CTLM) and cosine transformed Logistic-Sine Map (CTLSM), individually, to propose convert three different encryption approaches. These proposed schemes use 384-bit initial shared key during the process of image encryption. All the proposed schemes provide the resistance from various security vulnerabilities like known/ chosen plaintext attack, differential
attack, statistical attack, brute force attack. However, the CTLSM combination with the proposed novel pseudo random key outperforms the other two proposed combinations.

This rest of the paper is organized as: Section 2 describes the proposed scheme; Security analysis of the proposed scheme is described in Section 3; Section 4 compares the proposed technique with existing works; Section 5 concludes the paper with discussion of future work.

## 2 Proposed architecture

This section discusses the steps involved in the implementation of the proposed method of image encryption. Figure 1 a,b show the encryption and decryption processes performed by using the proposed scheme. The encryption makes use of the proposed pseudo random key and chaotic function to encrypt the plain image, whereas the decryption process makes use of same pseudo random key and chaotic to decrypt the cipher image. The proposed method takes

![Diagram](image)

**Fig. 1 (a) Encryption using proposed method. (b) Decryption using proposed scheme.**
an initialization random variable $X$ of 128-bit and share key of 384-bit to encrypt the image file. The share key is divided into three part for each red, blue, green components of the image, respectively. The decryption process is just reverse of the encryption process. The following sections discuss both in detail. Table 1 shows various notation used in this paper.

### 2.1 Encryption

Figure 2 describes the detailed architecture of the proposed image encryption approach. As shown by the figure, the pseudo random key generation uses Hamming weight ($hw$), XOR, modulus ($mod$), and left rotate ($Rot$) operations. These operations are used to enhanced the security of the proposed scheme. The LM uses $mod$ and divide ($div$) operations to generate $x_0$, the initial seed for the chaos map. The whole process uses operation Addition ($Add$) and $mod$ to convert Plane Image ($PI$) to Cipher Image ($CI$).

Function 1 gives the pseudo code and following steps describe the flow for converting plain image into cipher image. The subsequent sections discuss these steps in detail.

**Step 1:** Take variable $X$ as 128-bit random number and generate the pseudo-random key.

**Step 2:** Generate the map using the LM / CTLM / CTLSM.

**Step 3:** Generate the cipher image using Logistic map, pseudorandom key as well as plain image.

**Step 4:** Repeat the step 1 to 3 for $i$ in size of image.

### Table 1

| Notation       | Meaning                                                                 |
|----------------|-------------------------------------------------------------------------|
| $hw(X)$        | Hamming weight of variable $X$ to count the number of $1$’s              |
| $X$            | 128-bit random variable                                                 |
| $ShareKey$     | 384-bit key use for the encryption/decryption process                   |
| $cipher_{img}[row][col][i]$ | Cipher image generated after the encryption process of $i^{th}$ color component. |
| $Key[i]$       | $Key$ of size 128 bit of $i^{th}$ component of the image               |
| $SRK_1,SRK_2,SRK_3,SRK_4$ | $Key$ of red component                                                  |
| $SGK_1,SGK_2,SGK_3,SGK_4$ | $Key$ of green component                                               |
| $SBK_1,SBK_2,SBK_3,SBK_4$ | $Key$ of blue component                                                |
| $LR(A,B)$ or $Rot$ | Left rotate of $A$ with $B$ number of times                            |
| $pseudo\_Key$ | Random key generated during the encryption/decryption process          |
| $A\oplus B$ or $XOR$ | $XOR$ operation of variable $A$ and variable $B$                        |
| $loop$, $i$, $row$, $col$ | Temporally variable use for the encryption/decryption process          |
| $initial\_par$ | Initial parameter                                                      |
| rows, cols     | Rows and columns of the image                                           |
| $Plain\_image$ | Original image                                                         |
| $X[ham64:ham64+32]$ | Generates the 32-bit value from the variable $X$                       |
| $div$          | Divide operator                                                        |
| $Pl_i$         | Plain image at a position of $i$                                        |
| $Cl_i$         | Cipher image at a position of $i$                                       |
| $Dec\_image$   | Decrypted image generates from the cipher image                        |
| IFOCS          | Improper Fractional Order Chaotic System                                |
| CML            | Coupled Map Lattice                                                    |
| PWLCM          | Piecewise Linear Chaotic Maps                                           |
| LAS            | Logistic Adjusted Sine                                                 |
| LM             | Logistic Map                                                           |
| CTBCS          | Cosine Transform Based Chaotic System                                   |
2.1.1 Design of pseudo-random key

As described earlier, the pseudo random key takes share key of size 384-bits as input for its design. This share key is divided into three parts of 128-bit each (Key1, Key2, Key3) for each red, green and blue component of the given plain image. Then, each share key of 128 bit is further divided into four parts of each of 32-bit value, which are named as SRK1, SRK2, SRK3, SRK4 for red channel, SBK1, SBK2, SBK3, SBK4 for green channel and SGK1, SGK2, SGK3, SGK4 for blue channel. The Key1, Key2,
and Key3 should not equal to zero. Figure 3 shows the structure of share key during the image encryption process.

The final output of pseudo random key generation process gives us three pseudo-random keys for each component of the image, where each pseudo random key is generated using the initial random variable $X$ and the 128-bit share key of the corresponding component. The variable $X$ is computed using eq. (1).

$$X \leftarrow X \oplus \text{Key}[i] \oplus LR(X, \text{hw}(X \oplus \text{Key}[i]))$$  \hspace{1cm} (1)

The significance of this sequence is that a single change in bit by the adversary will change number of bits in the result. For instance, let us consider two different numbers initial vector $A = 10,111,011,101$, and key $B = 10,101,011,011$. The XOR function of $A$ and $B$ is defined as: $A \oplus B = 0001000111110$. The hamming weight is calculated as: $\text{hw}(A \oplus B) = 3$ and Left rotate the $A \oplus B$ up to hamming distance is evaluated as: $\text{Left}_\text{Rotate}(A, \text{hw}(A \oplus B)) = 11011101101$. Let us denote $\text{Left}_\text{Rotate}(A, \text{hw}(A \oplus B))$ as $C$ and compute the $A \oplus B \oplus \text{Left}_\text{Rotate}(A, \text{hw}(A \oplus B))$ as $A \oplus B \oplus C = 11001101011$. Now, If the adversary changes single bit of initial vector at arbitrary position, let us say at 5th bit to get new initial vector, $A' = 10111001101$, and remaining computations as $A' \oplus B = 000100011110$, $\text{hw}(A' \oplus B) = 4$. The final computation $\text{Left}_\text{Rotate}(A', \text{hw}(A' \oplus B))$ denoted as $C' = 10111011011$, and the XOR is computed as $A' \oplus B \oplus C' = 101101001101$. Hence, the change of one bit in initial vector modifies number of bits, making it difficult to understand the input vector.

$SRK1$, $SRK2$, $SRK3$, $SRK4$ for red channel, $SBK1$, $SBK2$, $SBK3$, $SBK4$ for green channel and $SGK1$, $SGK2$, $SGK3$, $SGK4$ for blue channel are computed using eq. (2) to eq. (4).

$$SRK \leftarrow \text{Key}[1] \ [(\text{col} + (\text{cols} \ast \text{row})) \% 4]$$  \hspace{1cm} (2)

$$SGK \leftarrow \text{Key}[2] \ [(\text{col} + (\text{cols} \ast \text{row})) \% 4]$$  \hspace{1cm} (3)

$$SBK \leftarrow \text{Key}[3] \ [(\text{col} + (\text{cols} \ast \text{row})) \% 4]$$  \hspace{1cm} (4)

where, the $\text{col} + (\text{cols} \ast \text{row})$ calculate the position of the plain image. The pseudo key generated for red, Green and blue component of the image are computed as:
2.1.2 Chaotic sequence generation

The proposed scheme uses Logistic map (LM), Cosine Transformed Logistic Sine map (CTLSM), Cosine Transformation of Logistic map (CTLM) in the encryption process. Generally, the Logistic map is defined in eq. 8.

\[ x_{t+1} = r \times x_t \times (1-x_t) \]  

where \( r \in [0, 4] \), \( x_t \) is initial parameters for generating the logistic map and \( x_{t+1} \) is next value generating for the logistic map. Here, \( x_0 \) is the initial seed of the chaos function. The behaviour of the chaos function depends upon the parameter \( r \). If \( r \) is kept between 0 to 3, the behaviour of the chaos function independent of the initial condition as the population quickly approaches the value \( \frac{r-1}{r} \). Keeping \( r \) between 3 to 3.44949, 3.44949 to 3.54409, and beyond 3.54409 oscillates the population among 2, 4 and 8 values, respectively. The logistic function shows chaotic behaviour, when parameter \( r \) varies from approximately 3.56995 to approximately 3.82843, and beyond \( r = 4 \), almost all initial values eventually leave the interval \([0,1]\) and diverge [47, 53].

In the proposed scheme, the initial variable of logistic map \( (x0) \) is calculating using the two different hamming weights \( hw(X \oplus Key[i]) \) and \( hw(X) \) in eq. 9.

\[ x0 = 1/(hw(X \oplus Key[i]) + (hw(X) \% 64)) \]  

LM is simple and easy to implement. However, it suffers from issues such as stable windows, blank windows and small key space [26]. To overcome these issues, researchers have either replaced LM with more complex and dynamic chaos maps, or combined it with some complex functions. The proposed work in this paper also uses two such strategies: the cosine transformation of LM (CTLM) and cosine transformation of Logistic-Sine (CTLSM). The CTLM is mathematically defined as [3]:

\[ x_{t+1} = \cos \left( 2^{k+r \times x_t \times (1-x_t)} \right) \]  

where, \( r \in [0, 4] \) and \( k \in [10, 24] \) are control parameters for the Cosine of Logistic map. The CTLSM is represented as [27]:

\[ x_{t+1} = \cos \left( \pi \times 4r x_t \times (1-x_t) + (1-r) \sin \left( \pi x_t \right) + 0.5 \right) \]  

where \( r \in [0, 1] \) control parameters for the cosine of Logistic-Sine map. The research works of [3, 27] show that both CTLM and CTLSM are more secure, complex and dynamic than simple LM map.

2.1.3 Cipher image creation

After the creation of pseudo-random key and the Logistic map, the cipher image is generated at a \( i \)th component of the image. The add and mod operator applied to the
Plain Image, pseudo Key and x0 * X[ham64 : ham64 + 32] to generates the ith component of the image.

\[
\text{Cipher Image}[\text{row}][\text{col}][i] = \left( x0 * X[\text{ham64} : \text{ham64 + 32}] \right) %256 + (\text{pseudo Key} % 256) + \text{Plain Image [row] [col][i]} % 256
\]

(12)

2.2 Decryption

To decrypt the cipher image, the pseudo-random key is generated using the initial parameter and the random number. Then, the chaos map and the pseudo-random key produce the plain image from the cipher image. Function 2 defines the procedure of the decrypting cipher image into color image. The algorithm takes initial parameter, shared key, number of rows and columns in color image, r, and, Cipher Image for input and produce an output of plain image. There are various steps defined to decrypting color image.

Step 1: Compute the pseudorandom key for the decryption process
Step 2: Generate the chaos function using LM or CTLM or CTLSM.
Step 3: Generate the decrypting image using chaos function, pseudorandom key and cipher image as:

\[
\text{Dec image [row][col][i]} = (\text{cipher img [row][col][i]} - \left( x0 * X[\text{ham64} : \text{ham64 + 32}] \right) %256 + (\text{pseudo Key} % 256)) % 256
\]

(13)

Step 4: Repeat the step 1 to 3 for i in size of image

| Function 2: Decrypting Image |
|-------------------------------|
| **Input**: initial_par, ShareKey, rows, cols, Cipher Image, r |
| **CODE**: |
| X ← initial_par |
| for i in 1 to color: |
| x0 ← 1 / (hw (X ⊕ Key [i]) + (hw (X) % 64)) |
| Key [i] ← ShareKey [i * 128 : 128 (i + 1)] |
| for row in 0 to rows: |
| for cols in 0 to cols: |
| X ← X ⊕ Key [i] ⊕ LR (X, hw (X ⊕ Key [i])) |
| pseudo Key ← X % (Key [i] [(col + (cols * row)) % 4] + col + (cols * row)) |
| ham64 = hw (X) % 64 |
| map ← r * x0 * (1 - x0) # generate the chaos function i.e LM / CTLM / CTLSM. |
| Dec image [row][col][i] ← (Cipher Image[row][col][i] - (map * X[ham64: ham64 + 32]) % 256) + (pseudo Key % 256)) % 256 |
| x0 = map |
| return Dec image |
| **Output**: Dec image |
3 Experimental setup and results analysis

The proposed scheme has been implemented using the Python language on Windows 10 Operating on Intel Core i3 2370M processor with 6 GB RAM, and 256 GB SSD drive. To perform the experiments, Python language packages matplotlib, pillow, numpy, and skimage have been used. The set of four different images CompleteSun \[42\], Sun \[43\], Star \[29\] and Cosmic_Fair_Lights \[1\] have been used for testing the purposed method. These images have been taken from the flickr website. These images were taken by the “NASA Goddard Space Flight Center” under the licence of CC-BY. The Fig. 4 shows these four images. The Section 3.1 describes the security analysis of our proposed scheme.

3.1 Security analysis

This section describes the analysis of the various metrics used to evaluate the security of the proposed scheme. The various metrics used are: Brute force attack, Key sensitivity, Histogram analysis, Differential attack, and Known or Chosen plaintext attack.

3.1.1 Brute force attack

The adversary tries to compute all the possible combinations to decrypt the cipher image into original image. This scheme uses the 384-bit key use for the encryption process. Therefore, the adversary will be required to perform $2^{384}$ steps to decrypt the share key, which is impractical.

To compute the cipher image, the adversary will choose a key and variable $X$, which will require $2^{128}$ each. Hence, the total complexity to guessing the pseudo key is $2^{128} \times 2^{128}$. The complexity to compute the initial value for the chaos function is 192 and the chaos map is multiplied by $X[\text{ham}64 : \text{ham}64 + 32]$. The value of $\text{ham}64$ lies between 0 to 64, and the $X[\text{ham}64 : \text{ham}64 + 32]$ selects the 32-bit values from the 0 to 96 bit of the variable $X$. Hence, the total complexity to generate the cipher image from the chaos function is $2^{96} \times 192$. Therefore, the total complexity is $2^{128} \times 2^{128} + 2^{96} \times 192$.

Similarly, while doing decryption using Function 2, the adversary knows about the cipher image ($CI$). Hence, the adversary will try to guess the variable map, $X[\text{ham}64 : \text{ham}64 + 32]$ and pseudo _Key. The complexity of guessing the map, $X[\text{ham}64 : \text{ham}64 + 32]$, and pseudo _Key is 192, $2^{32}$, and $2^{32}$ respectively. Hence, the total complexity for guessing one byte of the cipher image is $192 \times 2^{32} \times 2^{32}$. Therefore, the complexity to guessing the overall each component of image is $192 \times 2^{32} \times 2^{32} \times 2^{\text{rows}} \times \text{cols}$. Hence, this scheme provides good resistance from the brute force attack.

Fig. 4 Test image set (a) Sun \[43\] (b) CompleteSun \[42\] (c) Star \[29\] (d) Cosmic_Fair_Lights \[1\]
3.1.2 Key sensitivity

To test the proposed scheme for key sensitivity, the single pixel of the image is modified and the obtained cipher images are compared. This is also known as the Avalanche effect. This effect is used to estimate the efficiency of the diffusion mechanism.

Figure 5 explains the application of avalanche effect on the Sun.jpg image. Initially, the original image shown by Fig. 5(a) is modified by a single bit randomly to obtain two ciphers $C_1$ and $C_2$, shown by Figs. 5(b) and 5(c), respectively. The difference of two ciphers $C_1$ and $C_2$ has been calculated and is shown by Fig. 5(d).

3.1.3 Histogram analysis

The histogram shows the distribution of the digital image according to the pixel values. Ideally, a cipher image should have flat distribution for the image pixels. Table 2 shows the histograms of the cipher image generated for the different components of the plain image Cosmic_Fair_Lights using all the three proposed combinations. The results clearly show that the histograms are uniform for all the proposed approaches.

3.1.4 Differential attack

In differential attack, the attacker produces two cipher images. The first image is the original image converted into the cipher image. The second one is the cipher image obtained by making trivial change in the original image. Then, the encrypted image and the original image are compared using the two parameters i.e. UACI and NPCR.

**NPCR** Number of changing pixel rate (NPCR) [7] determines, how the change in one-pixel value of the original image produces the change in the encrypted image. It determines the percentage of change of pixel value in the original and encrypted image. Hence, NPCR should have higher value for the better image encryption. The NPCR is calculated as:

$$D (i, j) = \begin{cases} C_1 (i, j) = C_2 (i, j), & D (i, j) = 0 \\ \text{Otherwise}, & D (i, j) = 1 \end{cases}$$

$$\text{NPCR} = \frac{\sum D (i, j)}{\text{size}} \times 100\%$$

where \(\text{size}\) refers to the size of the image, and \(C_1, C_2\) are the two different cipher images.

![Fig. 5](image_url) (a) Original image (b) Cipher (C1) (c) Cipher (C2) (d) Difference image
United average change intensity (UACI) determines the difference between the average intensity of the two images. Generally, UACI is higher for the encrypted image for better encryption.

\[
UACI = \frac{1}{\text{size}} \left( \sum_{i,j} \frac{C_1(i,j) - C_2(i,j)}{255} \right) \times 100
\]

where \(\text{size}\) is the size of the image, and \(C_1, C_2\) are the two cipher images. Table 3 results show the UACI and NPCR values of the different images corresponding to their color components i.e. Red(R), Green(G), and Blue(B). The results clearly show that the CTLM combination with novel pseudo random key outperforms other two combinations.

### 3.1.5 Statistical attack

The adversary can break the encryption technique using the Correlation Coefficient (CC) and Entropy of the image. To defend statistical attack, it is essential the correlation between the adjacent pixels of the encrypted image should be close to zero. Also, the entropy of encrypted image should be close to ideal value i.e. 8.

**Information entropy** Entropy reflects the unpredictability and randomness of the digital images. The entropy \(H(S)\) of an image is defined as:

\[
H(S) = \sum_{i=0}^{2^N-1} P(S_i) \log_2 \frac{1}{P(S_i)}
\]

where, \(P(S_i)\) is the probability of the random variable \(S_i\). Ideally, the information entropy of the 8-bit image is 8. The Table 3 shows that the information entropy values of various encrypted images...
for the corresponding test dataset. Here also, the PR-CTLSM combination dominates the results. Therefore, it is very difficult for the adversary to perform statistical attack on this scheme.

**Correlation coefficient (CC)** The Correlation Coefficient determines the relation between the adjacent pixels of the image [45]. The Correlation Coefficient is nearly equal to zero for the good quality image encryption. It is mathematically expressed as:

\[
    r_{xy} = \frac{\text{cov}(x, y)}{\sqrt{T(x)} \sqrt{T(y)}}
\]

(18)

\[
    T(x) = \frac{1}{L} \sum_{pos=1}^{L} (x_{pos} - E(x))^2
\]

(19)

\[
    E(x) = \frac{1}{L} \sum_{pos=1}^{L} x_{pos}
\]

(20)

\[
    \text{cov}(x, y) = \frac{1}{L} \sum_{pos=1}^{L} (x_{pos} - E(x))(y_{pos} - E(y))
\]

(21)

where \( L \) is defined as the number of the pixel of the image. The variable \( x \) and \( y \) represent as plain image and cipher image, respectively. Figure 6 shows the correlation of the different component of the original and cipher images obtained using proposed schemes. Table 4 shows the values of the CC obtained for the horizontal (CCH), vertical (CCV) and diagonal (CCD)

| Image | Color | PR - LM | UACI | Entropy | NPCR | PR - CTLM | UACI | Entropy | NPCR | PR - CTLSM | UACI | Entropy |
|-------|-------|---------|------|---------|------|-----------|------|---------|------|------------|------|---------|
| Red   |         | 99.62   | 33.48| 7.9946  | 99.66| 33.45     | 7.9991|
| Green |         | 99.58   | 33.52| 7.9997  | 99.62| 33.44     | 7.9993|
| Blue  |         | 99.60   | 33.42| 7.8383  | 99.64| 33.41     | 7.9936|
| RGB   |         | 99.61   | 33.47| 7.9440  | 99.64| 33.44     | 7.9974|
| Red   |         | 99.61   | 33.50| 7.9997  | 99.66| 33.50     | 7.9992|
| Green |         | 99.61   | 33.60| 7.9987  | 99.65| 33.44     | 7.9968|
| Blue  |         | 99.61   | 33.47| 7.9943  | 99.62| 33.44     | 7.9963|
| RGB   |         | 99.61   | 33.47| 7.9997  | 99.65| 33.47     | 7.9975|
| Red   |         | 99.61   | 33.50| 7.9997  | 99.67| 33.45     | 7.9993|
| Green |         | 99.61   | 33.60| 7.9980  | 99.60| 33.45     | 7.9992|
| Blue  |         | 99.61   | 33.38| 7.9879  | 99.59| 33.38     | 7.9906|
| RGB   |         | 99.63   | 33.44| 7.9995  | 99.62| 33.46     | 7.9992|
| Red   |         | 99.62   | 33.67| 7.9974  | 99.56| 33.48     | 7.9992|
| Green |         | 99.62   | 33.55| 7.9975  | 99.59| 33.47     | 7.9990|
| Blue  |         | 99.60   | 33.51| 7.9981  | 99.61| 33.54     | 7.9991|
| RGB   |         | 99.59   | 33.47| 7.9977  | 99.59| 33.50     | 7.9991|

**Table 3** Analysis of UACI, NPCR and Entropy
adjacent pixels of the original and encrypted images using the proposed schemes. These values have been generated using the randomly pairs of the pixel values, and these values are close to zero for the encrypted images, as shown in Table 4. Hence, all the proposed schemes provide resistance from the statistical attack.

### 3.2 Decryption parameters

#### 3.2.1 Bit correct ratio (BCR)

The Bit correct ratio (BCR) is defined as the total number of bit errors between the original image and encrypted image. Mathematically, bit correct ratio is defined as:

\[
BCR = 1 - \frac{\sum_{x,y}^{\text{rows} \times \text{columns}} O(x, y) \oplus R(x, y)}{\text{rows} \times \text{columns}}
\]

Where \( O(x, y) \) represent the pixel value at position \( x \) and \( y \) of original image, and \( R(x, y) \) represent the \( x, y \) pixel position of the received image to the receiver and BCR belongs to \([0, 1]\). The ideal value of the BCR is 1. Figure 7 shows decrypted different images at different percentage of salt and pepper noise at 1%, 1.5%, 2%, 2.5%, and 3%. Table 5 gives the values of average BCR at different noise levels, obtained using the proposed schemes.

#### 3.2.2 PSNR

Peak signal to noise ratio (PSNR) define as the ratio between the maximum size and the noise of the image. \( PSNR \) calculates as follows:

\[
PSNR = 10 \times \log_{10} \left( \frac{255 \times 255}{\sqrt{MSE}} \right)
\]

Where, \( MSE \) define as the error occurred between the original image and received image during the transmission of the image. It is calculated as:
### Table 4  Analysis between original and cipher images

| Image               | Color | Original | PR - LM | PR - CILM | PR - CTLSM |
|---------------------|-------|----------|---------|-----------|------------|
|                     |       | Cipher   | Cipher  | Cipher    | Cipher     |
|                     |       | CCH      | CCV     | CCD       |            |
| Complete_Sun.jpg    | Red   | 0.9949   | 0.0001  | 0.0010    | 0.0018     |
|                     |       | 0.9929   | 0.0002  | -0.0018   | 0.0001     |
|                     |       | 0.9891   | 0.0017  | -0.0016   | 0.0015     |
|                     | Green | 0.9868   | 0.0019  | 0.0016    | -0.0011    |
|                     |       | 0.9694   | 0.0004  | -0.0012   | 0.0072     |
|                     |       | 0.9578   | 0.0010  | 0.0034    | 0.0007     |
|                     | Blue  | 0.7784   | 0.0011  | -0.0005   | -0.0031    |
|                     |       | 0.7421   | 0.0052  | -0.0020   | -0.0031    |
|                     |       | 0.6964   | -0.0032 | -0.0017   | 0.0025     |
|                     | RGB   | 0.3272   | -0.0001 | 0.0030    | -0.0016    |
|                     |       | -0.1052  | -0.0004 | 0.0007    | -0.0003    |
|                     |       | -0.1046  | -0.0019 | -0.0018   | -0.0025    |
| Sun.jpg             | Red   | 0.9937   | 0.0018  | -0.0005   | 0.0020     |
|                     |       | 0.9920   | 0.0033  | -0.0007   | 0.0005     |
|                     |       | 0.9881   | 0.0018  | 0.0007    | -0.00003   |
|                     | Green | 0.9865   | -0.0018 | -0.0004   | -0.0046    |
|                     |       | 0.9778   | 0.0004  | -0.0005   | 0.0075     |
|                     |       | 0.9660   | 0.0024  | 0.0015    | -0.0046    |
|                     | Blue  | 0.9644   | -0.0028 | -0.0007   | -0.0018    |
|                     |       | 0.9732   | 0.0006  | -0.0019   | 0.0047     |
|                     |       | 0.9510   | 0.0029  | -0.0009   | 0.0024     |
|                     | RGB   | 0.5272   | 0.0012  | 0.0029    | -0.0005    |
|                     |       | -0.1052  | -0.0031 | 0.0009    | 0.0012     |
|                     |       | -0.1046  | 0.0017  | 0.0007    | -0.0007    |
|                     |       | 0.6833   | 0.0045  | 0.0017    | 0.0006     |
| Star.jpg            | Red   | 0.7225   | 0.0015  | 0.0040    | -0.0025    |
|                     |       | 0.3901   | -0.0016 | 0.0003    | -0.0017    |
|                     |       | 0.7175   | 0.0045  | -0.0004   | -0.0015    |
|                     | Green | 0.7071   | 0.0015  | -0.0014   | -0.0014    |
|                     |       | 0.5796   | -0.0016 | 0.0017    | 0.0015     |
|                     |       | 0.7017   | 0.0099  | -0.0003   | -0.0002    |
|                     | Blue  | 0.7676   | 0.0003  | -0.0019   | -0.0019    |
|                     |       | 0.6446   | 0.0011  | -0.0005   | -0.0006    |
|                     | RGB   | 0.8481   | -2.01e-07| 0.0017    | 0.0017     |
|                     |       | -0.0873  | -0.0003 | 0.0008    | 0.0008     |
|                     |       | -0.0870  | -0.0006 | -0.0013   | -0.0013    |
| Cosmic_Fair_Lights | Red   | 0.9669   | -0.0012 | 0.0023    | 0.0024     |
|                     |       | 0.9684   | 0.0076  | -0.0003   | -0.0002    |
|                     |       | 0.9533   | 0.0008  | 0.0012    | 0.0012     |
|                     | Green | 0.9575   | -0.0003 | -0.0015   | -0.0014    |
|                     |       | 0.9628   | -0.0032 | 0.0050    | 0.0050     |
|                     |       | 0.9538   | 0.0004  | -0.00003  | -0.00002   |
|                     | Blue  | 0.9690   | 0.0048  | -0.0021   | -0.0020    |
|                     |       | 0.9748   | 0.0078  | 0.0076    | 0.0077     |
|                     |       | 0.9549   | 0.0016  | -0.0021   | -0.0020    |
|                     | RGB   | 0.5272   | 0.0012  | 0.0003    | 0.0003     |
|                     |       | -0.1052  | -0.0007 | 0.0002    | 0.0001     |
|                     |       | -0.1046  | -0.0010 | -0.0008   | -0.0007    |
\[ MSE = \frac{1}{\text{rows} \times \text{columns}} \sum_{x=1}^{\text{rows}} \sum_{y=1}^{\text{columns}} [O(x,y) - R(x,y)]^2 \]  

where \( O(x,y) \) represent the pixel value at position \( x \) and \( y \) of original image, and \( R(x,y) \) represent the \( x, y \) pixel position of the received image to the receiver. Table 5 shows that the PSNR values of the received images at the different percentage of ‘salt and pepper’ noise.

### 3.3 Time analysis

Encryption and decryption time plays an important role in judging efficiency of an image encryption algorithm i.e. a good image encryption algorithm should take very less time encryption and decryption time. The start of section 3 described the system and image specifications that are used to implement the proposed image encryption approach. The proposed scheme takes approximately 8 s to encrypt or decrypt an image of size 512*512. Undoubtedly, the system with higher specifications will do the same job in lesser time. The time analysis has been carried out by using time () function of Python.

### 4 Comparison with earlier research works

This section of the paper discusses the comparison of the proposed work in this paper with some of the earlier proposed research works. The Table 6 describes the comparative analysis of the different existing schemes with the proposed schemes in terms of Correlation Coefficient, NPCR, UACI and Entropy. The color image “Lena” of size “512*512” has been used to perform this performance analysis. It can clearly be observed that combining the novel pseudo random key with LM, cosine transformed LM, cosine transformed Logistic Sine Map improves the efficiency of these maps to resist different types of attack.
5 Conclusion & future work

In this digital age, protecting information from unauthorized access is the biggest challenge. As researchers work to propose new algorithms to protect the vital information from attackers, the attackers also work on developing new threats/attacks to steal the digital information. Hence, developing a fast, more secure and efficient algorithm is need of the hour. In this paper, a novel pseudo random key has been combine with three different chaotic sequences to propose three image encryption techniques for encrypting satellite images. The pseudo-random key has been generated by applying various operations like XOR, left circular shift (Left Rotate), modular. Then, this key has been combined with Logistic map (LM), cosine transformed LM (CTLM) and cosine transformed Logistic-Sine Map (CTLSM) one by one to implement three different image encryption methods. All the proposed schemes use a 384-bit initial shared key during the process of image encryption. The comparative analysis has been done for all three methods by using standard encryption and decryption parameters. Also, analysis has also been performed for various security vulnerabilities like known/chosen-plaintext attack, differential attack, statistical attack, and brute force attack. It can be observed from the results that the

| Image | Noise | BCR | PSNR |
|-------|-------|-----|------|
|       |       | PR-LM | PR-CTLM | PR-CTLSM | PR-LM | PR-CTLM | PR-CTLSM |
| 1 %   | 0.9802 | 0.9795 | 0.9806 | 35.83 | 35.81 | 35.91 |
| 1.5%  | 0.9700 | 0.9700 | 0.9702 | 34.92 | 34.96 | 34.99 |
| 2 %   | 0.9601 | 0.9600 | 0.9600 | 34.31 | 34.30 | 34.35 |
| 2.5%  | 0.9502 | 0.9504 | 0.9501 | 33.84 | 33.87 | 33.85 |
| 3 %   | 0.9499 | 0.9403 | 0.9404 | 33.40 | 33.46 | 33.46 |
| 1 %   | 0.9792 | 0.9800 | 0.9804 | 35.48 | 35.61 | 35.67 |
| 1.5%  | 0.9696 | 0.9701 | 0.9701 | 34.66 | 34.73 | 34.73 |
| 2 %   | 0.9601 | 0.9597 | 0.9602 | 34.09 | 34.05 | 34.10 |
| 2.5%  | 0.9490 | 0.9500 | 0.9505 | 33.53 | 33.61 | 33.62 |
| 3 %   | 0.9407 | 0.9406 | 0.9399 | 33.21 | 33.22 | 33.23 |
| 1 %   | 0.9797 | 0.9795 | 0.9799 | 36.22 | 36.19 | 36.20 |
| 1.5%  | 0.9699 | 0.9700 | 0.9704 | 35.32 | 35.39 | 35.46 |
| 2 %   | 0.9601 | 0.9597 | 0.9604 | 34.74 | 34.73 | 34.76 |
| 2.5%  | 0.9499 | 0.9505 | 0.9503 | 34.22 | 34.29 | 34.32 |
| 3 %   | 0.9400 | 0.9402 | 0.9402 | 33.86 | 33.90 | 33.89 |
| 1 %   | 0.9800 | 0.9799 | 0.9801 | 35.86 | 35.75 | 35.71 |
| 1.5%  | 0.9700 | 0.9699 | 0.9702 | 34.88 | 34.82 | 34.89 |
| 2 %   | 0.9596 | 0.9598 | 0.9599 | 34.18 | 34.22 | 34.22 |
| 2.5%  | 0.9504 | 0.9506 | 0.9514 | 33.80 | 33.77 | 33.77 |
| 3 %   | 0.9387 | 0.9407 | 0.9404 | 33.28 | 33.34 | 33.35 |
Table 6: Comparison of different existing schemes with the proposed scheme

| Work          | Scheme         | Encryption Parameters | Decryption Parameters |
|---------------|----------------|-----------------------|-----------------------|
|               |                | Key Space  | CCH     | CCV     | CCD     | NPCR   | UACI   | Entropy | PSNR | BCR |
| Zhang et al.  | –              | 200-bit   | 0.0003  | 0.0057  | 0.0018  | 99.62  | **33.56** | 7.9992 | NA   | NA  |
| Zhao et al.   | IFOCS         | 318-bit   | 0.0022  | −0.0010 | 0.0021  | 99.62  | 33.46  | 7.9992 | NA   | NA  |
| Wang et al.   | CML           | ~180-bit  | 0.0020  | −0.0007 | −0.0014 | 99.65  | 33.48  | 7.9970 | NA   | NA  |
| Zhang et al.  | 3D Cat map    | –         | −0.0042 | 0.0005  | −0.0036 | 99.61  | 33.49  | 7.9992 | NA   | NA  |
| Xu et al.     | PWLCM         | –         | −0.0230 | 0.0019  | 0.0034  | 99.62  | 33.51  | 7.9974 | NA   | NA  |
| Chai et al.   | 2D LM         | 256-bit   | −0.0045 | −0.0001 | 0.0053  | 99.59  | 33.42  | 7.9993 | –    | NA  |
| Bentoutou et  | 2D LAS        | 128-bit   | 0.0004  | 0.0005  | 0.0003  | 99.61  | 33.48  | 7.9994 | NA   | NA  |
| et al. [5]    | CTBCS         | 256-bit   | −0.0003 | −0.0007 | −0.0001 | NA     | NA     | NA     | NA   | NA  |
| Proposed      | PR-LM         | 384-bit   | −0.0017 | −0.0019 | −0.0036 | 99.65  | 33.46  | **7.9998** | **36.87** | **0.9800** |
|               | PR-CTLM       | 384-bit   | 0.0005  | −0.0004 | **0.00007** | 99.61  | 33.46  | **7.9997** | **36.88** | **0.9801** |
|               | PR-CTLSTM     | 384-bit   | 0.0005  | 0.0009  | 0.00043 | 99.63  | 33.44  | **7.9998** | **36.91** | **0.9802** |

*The corresponding value of PSNR and BCR at 1% “salt and pepper” noise
pseudo random-CTLSM combination outperforms the other two combinations. The scheme is capable of achieving encryption parameters entropy, UACI, NPCR and CC values 7.9998, 33.46, 99.63 and 0.00007, respectively. It also achieves decryption parameter results very close to the values that are ideal for a secure and efficient image encryption algorithm. As shown in time analysis, the whole process of encryption or decryption can be carried out in just 8 s. However, like any other research work, the proposed scheme also has scope of future improvements. The work can further be extended by using multidimensional, faster and more complex chaos function with the proposed novel pseudo random key to achieve less image and decryption times.

References

1. ‘Cosmic Fairy Lights’ by NASA Goddard Space Flight Center (n.d.) available at https://www.flickr.com/photos/gsfc/14010091745/in/gallery-188968346/ under a Creative Commons Attribution 2.0. Full terms at https://creativecommons.org/licenses/by/2.0/.
2. Alawida M, Samsudin A, Sen Teh J, Alkhawaldeh RS (2019) A new hybrid digital chaotic system with applications in image encryption. Signal Process 160:45–58. https://doi.org/10.1016/j.sigpro.2019.02.016
3. Alawida M, Samsudin A, Sen Teh J, Alshoura WH (2019) Digital cosine chaotic map for cryptographic applications. IEEE Access 7:150609–150622. https://doi.org/10.1109/ACCESS.2019.2947561
4. Bahrami S, Naderi M (2012) Image encryption using a lightweight stream encryption algorithm, Adv Multimed 2012. https://doi.org/10.1155/2012/767364.
5. Bentoutou Y, Bensikaddour EH, Taleb N, Bounoua N (2020) An improved image encryption algorithm for satellite applications. Adv Sp Res 66(1):176–192. https://doi.org/10.1016/j.asr.2019.09.027
6. Bisht A, Dua M, Dua S (2018) A novel approach to encrypt multiple images using multiple chaotic maps and chaotic discrete fractional random transform. J Ambient Intell Humaniz Comput 10(9):3519–3531. https://doi.org/10.1007/s12652-018-1072-0
7. Bisht A, Jaroli P, Dua M, Dua S (2018) Symmetric Multiple Image Encryption Using Multiple New One-Dimensional Chaotic Functions and Two-Dimensional Cat Man, in Proceedings of the International Conference on Inventive Research in Computing Applications, ICIRCA 2018, 676–682. doi: https://doi.org/10.1109/ICIRCA.2018.8597245
8. Bisht A, Dua M, Dua S, Jaroli P (2020) A color image encryption technique based on bit-level permutation and alternate logistic maps. J Intell Syst 29(1):1246–1260. https://doi.org/10.1515/jisyss-2018-0365
9. Chai X, Yang K, Gan Z (2017) A new chaos-based image encryption algorithm with dynamic key selection mechanisms. Multimed Tools Appl 76(7):9907–9927. https://doi.org/10.1007/s11071-016-3585-x
10. Chai X, Chen Y, Broyle L (2017) A novel chaos-based image encryption algorithm using DNA sequence operations. Opt Lasers Eng 88:197–213. https://doi.org/10.1016/j.optlaseng.2016.08.009
11. Chen J, Zhu Z, Fu C, Yu H (2013) An improved permutation-diffusion type image cipher with a chaotic orbit perturbing mechanism. Opt Express 21(23):27873. https://doi.org/10.1364/oe.21.027873
12. Chen JX, Zhu ZL, Fu C, Yu H, Zhang LB (2015) A fast chaos-based image encryption scheme with a dynamic state variables selection mechanism. Commun Nonlinear Sci Numer Simul 20(3):846–860. https://doi.org/10.1016/j.cnsns.2014.06.032
13. Chen L, Ma B, Zhao X, Wang S (2017) Differential cryptanalysis of a novel image encryption algorithm based on chaos and line map. Nonlinear Dyn. 87(3):1797–1807. https://doi.org/10.1007/s11071-016-3153-y
14. Chen J, Zhu ZL, Zhang LB, Zhang Y, Yang BQ (2018) Exploiting self-adaptive permutation-diffusion and DNA random encoding for secure and efficient image encryption. Signal Process 142:340–353. https://doi.org/10.1016/j.sigpro.2017.07.034
15. Dhall S, Pal SK, Sharma K (2018) A chaos-based probabilistic block cipher for image encryption, J. King Saud Univ. - Comput. Inf. Sci., https://doi.org/10.1016/j.jksuci.2018.09.015
16. Diaconu AV (2016) Circular inter-intra pixels bit-level permutation and chaos-based image encryption. Inf. Sci. (Ny). 355–356:314–327. https://doi.org/10.1016/j.ins.2015.10.027
17. Dragoi IC, Coltuc D (2015) On local prediction based reversible watermarking. IEEE Trans Image Process 24(4):1244–1246. https://doi.org/10.1109/TIP.2015.2395724
18. Dua M, Wesanekar A, Gupta V, Bhola M, Dua S (2020) Differential evolution optimization of intertwining logistic map-DNA based image encryption technique. J Ambient Intell Humaniz Comput 11(9):3771–3786, https://doi.org/10.1007/s12652-019-01580-z
19. Dua M, Suthar A, Garg A, Garg V (2020) An ILM-cosine transform-based improved approach to image encryption. Complex Intell Syst 7:1–17. https://doi.org/10.1007/s40747-020-00201-z
20. Farajallah M, El Assad S, Deforges O (2016) Fast and Secure Chaos-Based Cryptosystem for Images. Int. J. Bifurc. Chaos 26(2):1650021. https://doi.org/10.1142/S0218127416500218
21. Faraoun K (2010) Chaos-based key stream generator based on multiple maps combinations and its application to images encryption. Int Arab J Inf Technol 7(3):231–240
22. Fu C, Zhang GY, Zhu M, Chen Z, Lei WM (2018) A new chaos-based color image encryption scheme with an efficient substitution keystream generation strategy, Secur Commun Networks 2018, https://doi.org/10.1155/2018/2708532
23. Ge M, Ye R (2019) A novel image encryption scheme based on 3D bit matrix and chaotic map with Markov properties. Egypt Informatics J 20(1):45–54. https://doi.org/10.1016/j.eij.2018.10.001
24. Guessrati B, Ben Farah MA, Kachouri A, Same T (2016) Hash key-based image encryption using crossover operator and chaos. Multimed Tools Appl 75(8):4753–4769. https://doi.org/10.1007/s11042-015-2501-0
25. Hua Z, Yi S, Zhou Y (2018) Medical image encryption using high-speed scrambling and pixel adaptive diffusion. Signal Process 144:134–144. https://doi.org/10.1016/j.sigpro.2017.10.004
26. Hua Z, Zhou B, Zhou Y (2018) Sine-transform-based chaotic system with FPGA implementation. IEEE Trans Ind Electron 65(3):2557–2566. https://doi.org/10.1109/TIE.2017.2736515
27. Hua Z, Zhou Y, Huang H (2019) Cosine-transform-based chaotic system for image encryption. Inf. Sci. (Ny) 480:403–419. https://doi.org/10.1016/j.ins.2018.12.048
28. Huang X, Ye G (2014) An efficient self-adaptive model for chaotic image encryption algorithm. Commun Nonlinear Sci Numer Simul 19(12):4094–4104. https://doi.org/10.1016/j.cnsns.2014.04.012
29. ‘Hubble Marks 30 Years in Space With Tapestry of Blazing Starbirth’ by NASA Goddard Space Flight Center available at https://www.flickr.com/photos/gsfc/49812640738/in/gallery-18896346@N07-72157714781363781/ under a Creative Commons Attribution 2.0. Full terms at https://creativecommons.org/licenses/by/2.0/
30. Jallouli O, El Assad S, Chetto M, Lozi R (2018) Design and analysis of two stream ciphers based on chaotic coupling and multiplexing techniques. Multimed Tools Appl 77(11):13391–13417. https://doi.org/10.1007/s11042-017-4953-x
31. Kocarev L, Lian S (2011) Chaos-based cryptography: Theory, algorithms and applications 354
32. Kovalchuk A, Izonin I, Kustra N (2019) Information protection service using topological image coverage. Complex Intell Syst 7:1
33. Kovalchuk A, Izonin I, Riznyk O (2019) An efficient image encryption scheme using projective transformations. Procedia Comput Sci 160:584–589
34. Kovalchuk A, Izonin I, Lotoshynska N (2019) An approach towards image encryption and decryption using quaternary fractional-linear operations. Procedia Comput Sci. 160:491–496
35. Kovalchuk A, Izonin I, Lotoshynska N (2019) An approach towards image encryption and decryption using quaternary fractional-linear operations. Procedia Comput Sci. 160:491–496
36. Kovalchuk A, Izonin I, Lotoshynska N (2019) An approach towards image encryption and decryption using quaternary fractional-linear operations. Procedia Comput Sci. 160:491–496
37. Li H, Wang Y, Zuo Z (2019) Chaos-based image encryption algorithm with orbit perturbation and dynamic state variable selection mechanisms. Opt Lasers Eng 115(February 2018):197–207. https://doi.org/10.1016/j.optlaseng.2018.02.002
38. Liu L, Miao S (2017) An image encryption algorithm based on baker map with varying parameter. Multimed Tools Appl 76(15):16511–16527. https://doi.org/10.1007/s11042-016-3925-x
39. Liu L, Zhang LY, Wang J, Zhang Y, Wong KW (2016) Chosen-plaintext attack of an image encryption scheme based on modified permutation–diffusion structure. Nonlinear Dyn. 84(4):2241–2250. https://doi.org/10.1007/s11054-018-0622-0
40. Liu DD, Zhang W, Yu H, Zhu ZL (2018) An image encryption scheme using self-adaptive selective permutation and inter-intra-block feedback diffusion. Signal Processing 151:130–143. https://doi.org/10.1016/j.sigpro.2018.05.008
41. Liu J, Tang S, Lian J, Ma Y, Zhang X (2019) A novel fourth order chaotic system and its algorithm for medical image encryption. Multidimens Syst Signal Process 30(4):1637–1657. https://doi.org/10.1007/s11045-018-0629-7
42. ‘Magnificent CME Erupts on the Sun - August 31’ by NASA Goddard Space Flight Center available at https://www.flickr.com/photos/gsfc/7931832934/in/gallery-18896346@N07-72157714781363781/ under a Creative Commons Attribution 2.0. Full terms at https://creativecommons.org/licenses/by/2.0/
43. ‘Magnificent CME Erupts on the Sun - August 31’ by NASA Goddard Space Flight Center available at https://www.flickr.com/photos/gsfc/7931831962/in/gallery-18896346@N07-72157714781363781/ under a Creative Commons Attribution 2.0. Full terms at https://creativecommons.org/licenses/by/2.0/
