An ILM-cosine transform-based improved approach to image encryption

Mohit Dua¹ · Arun Suthar¹ · Arpit Garg¹ · Vaibhav Garg¹

Received: 25 April 2020 / Accepted: 15 September 2020 / Published online: 10 October 2020
© The Author(s) 2020

Abstract

The chaos-based cryptography techniques are used widely to protect digital information from intruders. The chaotic systems have some of special features that make them suitable for the purpose of encryption. These systems are highly unpredictable and are highly sensitive or responsive to the initial conditions, also known as butterfly effect. This sensitive dependence on initial conditions make these systems to exhibit an intricate dynamical behaviour. However, this dynamical behaviour is not much complex in simple one-dimensional chaotic maps. Hence, it becomes easy for an intruder to predict the contents of the message being sent. The proposed work in this paper introduces an improved method for encrypting images, which uses cosine transformation of 3-D Intertwining Logistic Map (ILM). The proposed approach has been split into three major parts. In the first part, Secure Hash Function-256 (SHA-256) is used with cosine transformed ILM (CT-ILM) to generate the chaotic sequence. This chaotic sequence is used by high-efficiency scrambling to reduce the correlations between the adjacent pixels of the image. In the second part, the image is rotated to move all the pixels away from their original position. In the third part, random order substitution is applied to change the value of image pixels. The effectiveness of the proposed method has been tested on a number of standard parameters such as correlation coefficient, Entropy and Unified average change in intensity. The proposed approach has also been tested for decryption parameters like mean square error and peak signal to noise ratio. It can easily be observed from the obtained results that the proposed method of image encryption is more secure and time efficient than some earlier proposed techniques. The approach works for both color and grey scale images.

Keywords CT-ILM · Random order substitution · Image encryption · CC · MSE

Introduction

The types of information exchange have undergone a radical transformation during last few decades. In the recent scenario, we exchange our most of the information in the form of digital images using electronic devices such as computers or mobiles [1–4]. Hence, security of digital images is a matter of great concern. Image encryption is the process of

converting a plain image into a cipher image with the help of some encryption algorithm, such that an unauthorized user cannot access it. Researchers have worked continuously on developing new image encryption methods that can ensure complete privacy and secrecy of the data being sent. Thus, a number of image encryption schemes have been proposed. These schemes can be split mainly into two types: classical and modern schemes of image encryption. The classical schemes of image encryption use either the concept of permutation or substitution. However, the modern schemes for image encryption use iterative approach that use both permutation and substitution. Traditional methods such as Data Encryption Standard (DES) and International Data Encryption (IDEA) proved to be ineffective for performing image encryption due to some intrinsic properties of images, such as high redundancy and high correlation among adjacent pixels [5–9]. Hence, chaos-based encryption was introduced, and nowadays chaos-based methods have become the basis of the most popular image encryption techniques used. The basis

✉ Mohit Dua
er.mohitdua@nitkkr.ac.in
Arun Suthar
arunsinghsuthar@gmail.com
Arpit Garg
garg.arpit465@gmail.com
Vaibhav Garg
gargvaibhav.vg@gmail.com

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

of a chaos-based cryptosystem is the mathematical equation that is responsible for its chaotic behaviours [10].

In 1976, Robert introduced a discrete-time-based model of the logistic equation that was suggested by Pierre François Verhulst in 1847. In 1989, R. Matthews first time proved that a chaotic sequence can be generated from same one-dimensional non-linear logistic function, under some specific conditions [11]. He introduced the concept of “parameter tuning” for generating chaotic sequence from the simple one-dimensional nonlinear function. He presented his research as the replacement of one-time pad system. The research claimed that certain properties of chaos theory such as topology mixing, ergodicity, randomness and high unpredictability, make it suitable for the purpose of encryption. The most important characteristic that makes chaos maps suitable for encryption is the high sensitivity to the initial conditions.

Till date, a number of chaos-based image encryption algorithm have been proposed. Chaos and cryptography are closely related. Like cryptographic systems, chaos-based systems also use iterations for encryption. Chaos-based systems use control parameter for generation chaotic sequences, whereas cryptographic system use secret key for the encryption purpose. It is a proven fact that the cryptosystems built using chaos concept own well-marked characteristics for encryption. However, the finite precision issue of digital machines makes it hard to produce a complete chaotic orbit with the trajectories of low-dimensional chaos system as long-term chaotic behaviour of chaos systems demands for infinite precision [12, 13]. To be specific, transitory time to periodic orbits and average period are the two major concerns that occur due to this finite precision of computer systems.

A simple non-linear iterative chaos map such as logistic map (LM), can be used to generate a random chaotic sequence. Simplicity, easy to implement and less computation are the salient characteristics of LM, which have been exploited by the researchers to propose various LM-based image encryption methods. Although the cryptosystems implemented using simple chaos maps like LM, have proven to be highly efficient, yet these suffer from some limitations. For instance, the irregular distribution of the output states in simple chaos maps leads to deterioration of their chaotic behaviour, while implementing these on fixed precision platforms. It has become possible to predict the behaviour of simple chaotic systems using some of the advanced technologies. One of the other factors that affects performance of simple chaos-based system is the presence of noise that disturbs the chaos parameters, which in turn results in degradation of chaotic properties. These disturbances mostly affect the chaos maps that have uneven or irregular distributions of output values. Many researchers in their works have proven that chaos-based image cryptosystems are vulnerable to attacks [14, 15]. However, it has also been seen in recent

researches that use of highly complex and more dynamic chaos maps helps in solving problems like periodic windows, blank and stable windows [16, 17].

The motivation for work of this paper is the approaches proposed in [18–21], and the implemented work is an extension to these works. In 2017, a solution to conquer the limitations of the LM was proposed using intertwining logistic map (ILM) [18]. In 2018, the authors of [19], proposed coupling of LM and Sine map to develop a coupled map. This coupled map exhibit more complexity in its behaviour, and has much larger chaotic range as compared to most of the 2D chaotic maps. In year 2019, the research work of [20], applied cosine transformation on logistic-sine combination for generating more complex and non-linear chaotic sequence. This chaotic sequence was combined with high-efficiency scrambling and random order substitution to achieve a secure image encryption technique. Also in 2019, the authors used differential evolution (DE) optimization with ILM and SHA-256 combination to obtain an efficient image encryption algorithm [21].

The work proposed in this paper uses cosine function with 3D ILM and SHA-256 function to obtain a more complex and dynamic chaotic sequence. This chaotic sequence is used by high-efficiency image scrambling to shuffle the image pixels. Subsequently, the image rotation and random order substitution steps are performed to obtain a more secure and efficient image encryption approach. The proposed method has been checked by using various standard image encryption parameters to test its resistance against various attacks. The quality of the decrypted image has been checked using standard decryption parameters.

The rest of the paper is structured as: “**Fundamentals**” discusses fundamentals of chaos maps used in this paper and their characteristics. In “**Proposed approach for image encryption**”, proposed image encryption approaches have been described, one based on ILM with cosine and the second that does not use cosine function for its implementation. Section “**Decryption**” presents the steps used for decryption of the image. The results of both the approaches are presented in “**Experimental setup and simulation results**”, conclusion and future work is presented in “**Conclusion and future work**” of the paper.

Fundamentals

This section discusses the fundamentals of ILM and Cosine Transformation of ILM (CT-ILM). Also, a comparison between LM and ILM, and ILM and CT-ILM is presented in this section.

Fig. 1 Logistic map **a** Bifurcation diagram, **b** blank window [22], r on x -axis and a_n on y -axis

LM and ILM

The simplest chaotic map LM is a degree two polynomial map chaos function. It is a plain non-linear model. However, this one-dimensional discrete-time model exhibits complex and dynamic behaviour that imparts it the potential to be used for the purpose of cryptography. Mathematically, the equation for LM is written as:

$$a_{n+1} = r a_n (1 - a_n). \quad (1)$$

Here, r is known as the control parameter that lies in the range $[0, 1]$ and a_n is defined as the ratio of existing population to highest possible population, which lies in the interval $[0, 4]$. Initial values of a_0 and r determine the sequence generated with the help of LM. The characteristics of system varies in accordance to the value of parameter r , which is also known as bifurcation parameter. System would be more chaotic if value of r would be close to 4 [16, 17]. Figure 1a depicts the characteristics of the system according to the value of parameter r . The plot depicts the value of r on x -axis and the values of parameter a on y -axis. LM produces chaotic sequences with good cross-correlation and autocorrelation properties. However, it has demerits such as weak key, stable windows, blank windows and non-uniform sequence distribution [16, 17].

Figure 1b depicts the blank window by the use of LM when value of parameter $r = 3.828$. To overcome these limitations, the one-dimensional LM was later extended to two-dimensional LM and then, two-dimensional LM got extended to three-dimensional LM. Later, an intertwining relation between three different LM sequences was designed to introduce ILM that has better key space and intricate

chaotic behaviour. The ILM is mathematically represented as [18, 20]:

$$l_{k+1} = [\lambda \times l \times m_k \times (1 - l_k) + n_k] \bmod 1, \quad (2)$$

$$m_{k+1} = \left[\lambda \times \beta \times m_k + n_k \times \left(1 + l_{k+1}^2 \right) \right] \bmod 1, \quad (3)$$

$$n_{k+1} = [\lambda \times (m_{k+1} + l_{k+1} + \gamma) \times \sin(n_k)] \bmod 1, \quad (4)$$

where λ is in range of $[0, 4)$, $l > 33.5$, $\beta > 37.9$ and $\gamma > 35.7$ and l_0, m_0, n_0 , lie in the range of 0 and 1. As shown by Fig. 2, ILM generated chaotic sequences have uniform distribution, in comparison to sequence produced by LM. This uniform distribution overcomes weakness of LM, such as stable window, blank window and uneven distribution.

Lyapunov exponent is a parameter that is used to determine if a map is chaotic or not. Figure 3 depicts the Lyapunov exponent for LM and ILM.

Frequency is always positive in case of ILM, which shows that the sequence generated by ILM is more complex and more chaotic compared to that generated by 1-D LM. Thus, ILM is more efficient way than LM to mix up the image pixels.

ILM and CT-ILM

In the presented approach, the standard ILM is merged with cosine function as shown by Eqs. (5)–(7) to obtain CT-ILM. The cosine function has been applied to all the three sequences l_k, m_k, n_k to generate more complex chaotic sequences. This increases the non-linearity of the output

Fig. 2 Sequence distribution for ILM **a** for any one l , m or n **b** for l , m and n [22]

Fig. 3 Lyapunov exponent **a** Logistic Map and **b** ILM [22]

produced by ILM and hence, increases the efficiency and security during encryption.

$$l_{k+1} = \cos([\lambda \times l \times m_k \times (1 - l_k) + n_k] \bmod 1 + \beta), \quad (5)$$

$$m_{k+1} = \cos\left(\left[\lambda \times \beta \times m_k + n_k \times \left(1 + l_{k+1}^2\right)\right] \bmod 1 + \beta\right), \quad (6)$$

$$n_{k+1} = \cos([\lambda \times (m_{k+1} + l_{k+1} + \gamma) \times \sin(n_k)] \bmod 1 + \beta). \quad (7)$$

Figure 4 shows the comparison between standard ILM and CT-ILM for the parameter Lyapunov Exponent. It can easily be seen that CT-ILM has more positive Lyapunov Exponent than the standard ILM.

Proposed approach for image encryption

This paper proposes two approaches for encryption of image. The first approach uses ILM with cosine function for seed generation, whereas the second approach uses only ILM for generating the chaotic sequence. This section provides explanation of both the approaches. Figure 5 depicts the block diagram for the proposed image encryption technique. SHA-256 hash function is used for generation of initial seed. The permutation step of proposed technique uses the ILM only and cosine transformed ILM sequence with high-efficiency scrambling to shuffle the input image pixels. The scrambled image is rotated by an angle of 90° . The pixels of resultant rotated image are diffused by applying random order substi-

Fig. 4 Lyapunov exponent of ILM and CT-ILM

Fig. 5 Block diagram for the proposed approach

tution to get the final cipher image as output. The following sections discuss the stepwise implementation of the proposed approach.

Color image input

The input image is divided into three two-dimensional matrices that represent the three colour components of an image, namely, red (R), green (G) and blue (B). These component

matrices are then changed to corresponding one-dimensional matrices. Function 1 shows the pseudo code for converting the color image input to three one dimensional matrices.

Function 1: Convert 2d color image to 1d image
Input: ORIG_IMG(n, m) // color Image (2D)
Output: IMG($n \times m$) // [r, g, b] 1D Matrices
CODE:
FOR p in range ($0 : n$) DO
FOR q in range ($0 : m$) DO
$r, g, b \leftarrow \text{ORIG_IMG}(p, q)$
$\text{IMG}(p * m + q) \leftarrow [r, g, b]$
END FOR
END FOR

Secret key generation

Secured Hash Function-256 (SHA-256) is a one-way hash function that converts text of some random length to a string of length 256. It is one of the six functions that belong to SHA-2 family. It generates key with larger key space and higher sensitivity, which helps in implementing more efficient image encryption. It is a modified version of SHA-0 hash function. SHA-256 has been used for the purpose of generating initial secret key in the proposed work. Text of length 360 is provided as an input to SHA-256 function that transforms it to a secret key of length 256. Three different chaotic sequences are generated using SHA-256, one for each color component (Red, Green and Blue) of the input image. Function 2 shows the pseudo code for generating the chaotic sequence.

Function 2: Secret Hash Function-256 (SHA-256)
Input: LENGTH_OF_KEY //bits of secret key
Output: SEED of length LENGTH_OF_KEY
CODE:
SECRET_KEY $\leftarrow$ generate_random_key (LENGTH_OF_KEY)
// create hash
KEY_TABLE $\leftarrow$ data_hash(SECRET_KEY, [binary Mode, double, 'SHA-256'])
FOR p within limits($0 : \text{LENGTH_OF_KEY}/2$) DO
CHECK_VALUE $\leftarrow$ CHECK_VALUE XOR KEY_TABLE(p)
END FOR
FOR p within limits ($\text{LENGTH_OF_KEY}/2 : \text{LENGTH_OF_KEY}$) DO
SUM_VALUE $\leftarrow$ SUM_VALUE + KEY_TABLE(p)
END FOR
SEED_VALUE $\leftarrow$ CHECK_VALUE + SUM_VALUE
SEED_VALUE $\leftarrow$ SEED_VALUE/ 2^{12}

Scrambling

This step of the algorithm is applied to reduce the association among the neighbouring pixels of the input plain image. For a size of block L , the given input image is divided into L^2 number of blocks, where mathematically L is defined as:

$$L = \min\{\lfloor \sqrt{M} \rfloor, \lfloor \sqrt{N} \rfloor\}. \quad (8)$$

Then, the pixels of one block are exchanged with the other blocks with the help of the chaotic sequence generated. Thus, pixels are shuffled and the association among the adjacent image pixels decreases. Function 3 gives the pseudo code for high-efficiency scrambling [20].

To perform the process of high-efficiency scrambling, an image of size $L^2 \times L^2$ is extracted from the plain input image, where value of L is evaluated using Eq. (8). Then, using chaotic map ILM or CT-ILM, four chaotic sequences S , T , U and V are generated. These sequences are sorted to obtain corresponding index vectors IS , IT , IU and IV , respectively. In the next step, two more matrices P and R of dimensions $L^2 \times L^2$ are created. The columns of index matrices IS and IU are copied to matrices P and R , respectively. The elements of matrix P are then shifted according to the elements of index matrix IT , and that of matrix R are shifted according to the elements of index matrix IV . After that, the following mentioned steps are run iteratively.

- Initialize index of row to $I, r = 1$.
- For the r th row, place its c th pixel to $R_{Pr,c}$, c th position of the $P_{r,c}$ th block.
- Steps (a) and (b) are repeated for r in $[2, L^2]$.

Function3: High efficiency Scrambling [20]
Input : Image E of size $M \times N$
Output : Scrambled image matrix Z of size $M \times N$
CODE : Calculate L using Eq. (8) Generate chaotic sequence C using ILM or CT-ILM map $S \leftarrow C_{1:L^2}$ $T \leftarrow C_{L^2+1:2L^2}$ $U \leftarrow C_{2L^2+1:3L^2}$ $V \leftarrow C_{3L^2+1:4L^2}$ $[S', IS] \leftarrow \text{Sort}(S)$, where $S' = S_{IS}$ $[T', IT] \leftarrow \text{Sort}(T)$, where $T' = T_{IT}$ $[U', IU] \leftarrow \text{Sort}(U)$, where $U' = U_{IU}$ $[V', IV] \leftarrow \text{Sort}(V)$, where $V' = V_{IV}$ Initialize $Z = E$, $P \in N^{L^2 \times L^2}$, $R \in N^{L^2 \times L^2}$ FOR y within limits($1, L^2$) DO FOR x within limits($1, L^2$) DO $m \leftarrow ((x + IT(y) - 1) \bmod L^2) + 1$ $n \leftarrow ((x + IV(y) - 1) \bmod L^2) + 1$ $P_{x,y} \leftarrow IS_m$ $R_{x,y} \leftarrow IU_n$ END FOR END FOR FOR x within limits($1, L^2$) DO FOR y within limits($1, L^2$) DO $a \leftarrow P_{x,y}$; $b \leftarrow R_{a,y}$ $m1 \leftarrow \left\lfloor \frac{a-1}{L} \right\rfloor$; $n1 \leftarrow (a-1) \bmod L$ $m2 \leftarrow \left\lfloor \frac{b-1}{L} \right\rfloor$; $n2 \leftarrow (b-1) \bmod L$ $i \leftarrow m1 * L + m2$; $j \leftarrow n1 * L + n2$ $Z(i, j) = E(x, y)$ END FOR END FOR

Image rotation

As described in the previous step, the scrambling shuffles only $L^2 \times L^2$ portion of the image. Hence, the remaining pixel still exhibit good association among them. Hence, to implement shuffling throughout the image, the scrambled image is rotated by 90° in the anticlockwise direction. The angle of rotation of image does not bring a much significant change in the final values obtained for the encryption parameters. Hence, the image may be rotated by any random angle. The only purpose of rotation is to shift the image pixels away from their original position.

Substitution

The main objective of this step is to disseminate the change in the values of image pixels to the whole image. Using certain pre-defined steps for implementing substitution in chaos-based cryptosystems makes it easier to break them. Hence, the proposed approach in this paper uses random order substitution that is hard to crack, thus ensuring more security of the encrypted data [20]. As described in the previous step, the chaotic sequence generated using the ILM or CT-ILM is used to generate the index matrix. This index matrix is then exploited to implement the substitution step that is mathematically described as [20]:

$$A_{I_{r,c},c} = \begin{cases} (Z_{I_{r,c},c} + Z_{I_{M,N},N} + d2^{32} \times C_{I_{r,c},ct}) \bmod M & \text{for } r = 1, c = 1 \\ (Z_{I_{r,c},c} + A_{I_{r-1,N},N} + d2^{32} \times C_{I_{r,c},ct}) \bmod M & \text{for } r = 2 \sim N, c = 1 \\ (Z_{I_{r,c},c} + A_{I_{r,c-1},c-1} + d2^{32} \times C_{I_{r,c},ct}) \bmod M & \text{for } r = 1 \sim N, c = 2 \sim N \end{cases} \quad (9)$$

Here, matrix Z is a result of scrambling step, C is the generated chaotic matrix, I is an index matrix, and $M = 256$ for 8-bit representation of pixels [20]. Figure 6 working of the proposed image encryption technique.

decryption process is also divided into three parts. In the first step, de-substitution is carried out to reverse the changes done during random order substitution. The second step involves rotating the image by 90° in the clockwise direction to reverse

Function 4: Random order substitution
Input : Scrambled Rotated image matrix Z
Output: Encrypted image Matrix A
CODE: Initialize $I_{N \times M} \leftarrow 0$, $C_{N \times M} \leftarrow 0$ Initialize $Z_{M \times N} \leftarrow [0,1]$ // Matrix Z of size $M \times N$ having values between 0 and 1 Initialize $x \leftarrow 0$ FOR array in Z DO FOR y in range(N) DO $C[y][x] \leftarrow \text{array}[y]$ END FOR array $\leftarrow$ index values based on sorted array values FOR y in range(row) DO $I[y][x] \leftarrow \text{array}[y]$ END FOR $x \leftarrow x + 1$ END FOR initialize $P \leftarrow 256$ Initialize $A_{N \times M} \leftarrow 0$ FOR x in range (N) DO : FOR y in range (N) DO : If $x == 0$ and $y == 0$ DO : $A[I[x][y]][y] \leftarrow (Z[I[x][y]][y] + Z[I[M - 1][N - 1]][N - 1] + (2^{32}) * C[I[x][y]][y]) \% P$ END if else if $y == 0$ DO : $A[I[x][y]][y] \leftarrow (Z[I[x][y]][y] + A[I[x - 1][N - 1]][N - 1] + (2^{32}) * C[I[x][y]][y]) \% P$ END else if else DO : $A[I[x][y]][y] \leftarrow (Z[I[x][y]][y] + A[I[x][y - 1]][y - 1] + (2^{32}) * C[I[x][y]][y]) \% P$ END else END FOR END FOR

Decryption

The steps performed during encryption are reversed to get the original image back from the encrypted image. Hence, the

the effect of rotation. The final step involves descrambling, which provides us the original image back.

De-substitution

It is the reverse of substitution process, and is done to obtain the original values of pixel back from the encrypted image. The Eq. (10) has been used for de-substitution, and the process followed during reverse random order substitution is shown by the Function 5.

$$Z_{I_{r,c},c} = \begin{cases} (A_{I_{r,c},c} - A_{I_{r,c},c-1} - 2^{32} \times C_{I_{r,c},c}) \bmod M & \text{for } r = 1 \sim N, c = 2 \sim N \\ (A_{I_{r,c},c} - A_{I_{r-1,N},N} - 2^{32} \times C_{I_{r,c},c}) \bmod M & \text{for } r = 2 \sim N, c = 1 \\ (A_{I_{r,c},c} - Z_{I_{M,N},N} - 2^{32} \times C_{I_{r,c},c}) \bmod M & \text{for } r = 1, c = 1 \end{cases} \quad (10)$$

Rotation of image

The image is rotated anticlockwise to ensure shuffling of all the pixels of image during encryption. Hence, it is rotated in the clockwise direction to reverse the changes and put the pixels back to the original position to perform decryption.

Function 5: Anti-substitution
Input: Encrypted image Matrix A
Output: Scrambled Rotated image Z
<p>Code: Initialize $I_{N \times M} \leftarrow 0$, $C_{N \times M} \leftarrow 0$ Initialize $Z_{M \times N} \leftarrow [0,1]$ // Matrix Z of size $M \times N$ having values between 0 and 1 Initialize $x \leftarrow 0$ FOR array in TDO FOR y in range(N) DO $C[y][x] \leftarrow \text{array}[y]$ END FOR array $\leftarrow$ index values based on sorted array values FOR y in range(row) DO $I[y][x] \leftarrow \text{array}[y]$ END FOR $x \leftarrow x + 1$ END FOR initialize $P \leftarrow 256$ Initialize $Z \leftarrow 0$ FOR x in range[0,N] DO FOR y in range [1,N] DO $Z[I[x][y]][y] \leftarrow (P + A[I[x][y]][y] - A[I[x][y-1]][y-1] - (2^{32}) * C[I[x][y]][y]) \% P$ END FOR END FOR Initialize y to 0 FOR x in range [1, N] DO $Z[I[x][y]][y] \leftarrow (P + A[I[x][y]][y] - A[I[x-1][N-1]][N-1] - (2^{32}) * C[I[x][y]][y]) \% P$ END FOR Initialize i to 0 $Z[I[x][y]][y] \leftarrow (P + A[I[x][y]][y] - Z[I[M-1][N-1]][N-1] - (2^{32}) * C[I[x][y]][y]) \% P$ return Z</p>

De-scrambling

It is the final step performed for retrieving back the original image. Process followed for the purpose of de-scrambling is shown by the Function 6.

Python libraries used to implement the code are: *math*, *Image*, *os*, *time*, *sys*. The various python packages used to implement the results in this paper are: *numpy*, *pillow*, *matplotlib*, *random*, *skimage*. The *random* package has been used to provide the random number in this paper. The *pillow* python package

Function 6 : Descrambling
Input: Scrambled plain image matrix Z of size $M \times N$
Output: Decrypted Plain image matrix E of size $M \times N$
<p>Code: Calculate L using Eq. (8) Generate chaotic sequence C using ILM or CT-ILM map $S \leftarrow C_{1:L}^2$ $T \leftarrow C_{L^2+1:2L^2}^2$ $U \leftarrow C_{2L^2+1:3L^2}^2$ $V \leftarrow C_{3L^2+1:4L^2}^2$ $[S', IS] \leftarrow \text{Sort}(S)$, where $S' = S_{IS}$ $[T', IT] \leftarrow \text{Sort}(T)$, where $T' = T_{IT}$ $[U', IU] \leftarrow \text{Sort}(U)$, where $U' = U_{IU}$ $[V', IV] \leftarrow \text{Sort}(V)$, where $V' = V_{IV}$ Initialize $Z = E$, $P \in \mathbb{N}^{L \times L}$, $R \in \mathbb{N}^{L \times L}$ FOR y within limits($1, L^2$) DO FOR x within limits($1, L^2$) DO $m \leftarrow ((x + IT(y) - 1) \bmod L^2) + 1$ $n \leftarrow ((x + IV(y) - 1) \bmod L^2) + 1$ $P_{x,y} \leftarrow IS_m$ $R_{x,y} \leftarrow IU_n$ END FOR END FOR FOR x within limits($1, L^2$) DO FOR y within limits($1, L^2$) DO $a \leftarrow P_{x,y}$; $b \leftarrow R_{a,y}$; $m1 \leftarrow \lfloor \frac{a-1}{L} \rfloor$; $n1 \leftarrow (a-1) \bmod L$ $m2 \leftarrow \lfloor \frac{b-1}{L} \rfloor$; $n2 \leftarrow (b-1) \bmod L$ $i \leftarrow m1 * L + m2$; $j \leftarrow n1 * L + n2$ $E(i, j) = Z(x, y)$ END FOR END FOR</p>

Experimental setup and simulation results

This section describes the experimental setup that has been used by the proposed image encryption scheme. It also discusses the simulation results obtained while testing the proposed scheme. For the development of the proposed encryption scheme, version 2.7 of Python has been used on PyCharm 2.3 platform with the Windows 10 based system. The system had Intel Core i7 processor clocked at 2.40 GHz, 8 GB RAM CPU, 1TeraBytes (TB) hard disk configuration.

is used to import and export the original image and cipher image, respectively, from the local directory. The *numpy*, *matplotlib* and *skimage* packages are used to process and visualize the output of the images.

Differential attack

Major challenge to encryption algorithms are the differential attacks. The attacker creates a pair of encrypted images by making minor changes in the image that is to be encrypted. The first encrypted image is created from the plain image,

Fig. 6 Working of the proposed algorithm

and the next encrypted image is created using first encrypted image. The obtained encrypted image and the original image are then compared to determine the correlation between the images. Unified average change in intensity (UACI) and the number of pixel change rate (NPCR) are the two most common parameters, which are used to estimate the potency of image encryption technique against differential attacks. UACI is a measure of average of the difference in intensity between two images and NPCR is a measure of rate change of number of pixels of encrypted image when only a single

pixel of original image is modified. Mathematically, UACI and NPCR are expressed as:

$$UACI = \frac{1}{A \times B} \sum_{x=1}^A \sum_{y=1}^B \frac{|E_1(x, y) - E_2(x, y)|}{255} \times 100\%, \quad (11)$$

$$NPCR = \frac{1}{A \times B} \sum_{x=1}^A \sum_{y=1}^B U(x, y) \times 100\%, \quad (12)$$

Table 1 Differential attack, CC and entropy analysis

Image	Approach	Encryption parameters					
		NPCR	UACI	CCV	CCH	CCD	Entropy
Color image							
Red	CT-ILM	99.5803	33.155	0.002536	− 0.008614	− 0.051130	7.997246
	ILM only	99.6154	32.979	− 0.026553	− 0.030300	− 0.003057	7.997575
Green	CT-ILM	99.5986	30.4375	0.006633	− 0.0183159	0.0297551	7.997089
	ILM only	99.6276	30.4070	0.016343	− 0.016901	0.024168	7.997625
Blue	CT-ILM	99.6307	27.6685	0.072281	− 0.02645	0.0098166	7.997403
	ILM only	99.5986	27.6980	− 0.046216	− 0.046878	− 0.004554	7.997098
Gray scale	CT-ILM	99.6337	28.8432	0.08270	− 0.025421	− 0.06269	7.997701
	ILM only	99.6276	28.8036	− 0.000511	− 0.003026	− 0.04725	7.997354

where $U(x, y)$ is defined as,

$$U(x, y) = \begin{cases} 0, & \text{otherwise} \\ 1, & c_1(x, y) \neq c_2(x, y). \end{cases} \quad (13)$$

Here, E_1 and E_2 denote the encrypted images and $E[x, y]$ denotes the value of pixel at x th row and y th column. The values of NPCR and UACI obtained using the proposed approach are shown in Table 1 and it can be seen that the obtained values are close to ideal values of NPCR and UACI i.e. 99 for NPCR and 30 for UACI. Thus, the proposed method ensures high security in opposition to differential attacks.

Correlation coefficient (CC) analysis

The relation between two neighbouring pixels of an image is described by correlation coefficient (CC). The value of correlation coefficient lies within the range $[-1, 1]$. Value equals to 1 denotes high correlation among the adjacent pixels and value equals to 0 denotes that the adjacent pixels do not have high correlation. CC denoted by $c_{i,j}$ is computed as:

$$c_{i,j} = \frac{\text{cov}(i, j)}{\sqrt{A(i)}\sqrt{A(j)}}, \quad (14)$$

where

$$\text{cov}(i, j) = \frac{1}{M} \sum_{k=1}^M (x_k - B(x))(y_k - B(y)) \quad (15)$$

$$A(x) = \frac{1}{M} \sum_{k=1}^M (x_k - B(x))^2 \quad (16)$$

$$B(x) = \frac{1}{M} \sum_{k=1}^M x_k. \quad (17)$$

Here, i and j are the two successive pixels, and M denotes pixel pairs (i, j) that are selected randomly. Expectation and variance of x are denoted by $B(x)$ and $A(x)$, respectively. To estimate the value of CC, one thousand pair of pixels have been chosen randomly, and then using duplets from these randomly selected pairs CC is computed. The results in Table 1 depict the CC values—horizontal (CCH), vertical (CCV) and diagonal (CCD) obtained using both the proposed approaches. The obtained results depict that CC values are closer to zero, for encrypted images. Thus, the proposed methods significantly reduce the correlations between the pixels thus making it hard to estimate the value of neighbouring pixels of a pixel.

Entropy analysis

Entropy is a measure of degree of randomness of image. If the value of entropy is close to ideal value i.e. 8, it can be said that encrypted image has high degree of randomness [23]. As shown by the results of Table 1, the value of entropy calculated using the proposed encryption approach is close to 8. Hence, the proposed method for encrypting image exhibit good randomness and is resistant to attacks. The mathematical expression of entropy is,

$$E(i) = \sum_{j=0}^{G-1} p(i_j) \log_2 \frac{1}{p(i_j)}, \quad (18)$$

where $G = 2^k$, denotes the total number of states of the image that is analysed. For a gray level, image value of k is 8.

Histogram analysis

The histogram analysis evaluates the strength of image encryption algorithm against statistical attacks. In other

Table 2 Histogram analysis

Color Component	Encrypted image	Original Image Histogram	Encrypted Image Histogram
R-component

G-component

B-component

words, it estimates the strength of permutation and diffusion properties in the final encrypted image against the statistical attacks. Histograms obtained for plain and encrypted images are shown in Table 2. The histogram of the encrypted image is notably uniform and different than the histogram of the plain or unencrypted image. Hence, it becomes tedious for the attacker to crack the image, thus ensuring safety of information [24].

PSNR and MSE analysis

Mean square error (MSE) is estimated by averaging the difference of squares of intensity between the encrypted and plain image. Lower value of MSE indicates lower error or lower noise in final encrypted image. Peak signal to noise ratio (PSNR) is a measurement of quality between encrypted image and plain image [25, 26]. It is used to compare the compression quality of an image. High value of PSNR indicates better quality of reconstructed image. It is an estimate of peak value of error. In case of images, plain image or the original data acts as the signal and error is the noise introduced during the process of encryption. PSNR is computed

using MSE. Mathematical expression for PSNR (in dB) and MSE is [27, 28, 29]:

$$\text{MSE} = \frac{1}{m \times n} \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} [O(x, y) - E(x, y)]^2, \quad (19)$$

$$\text{PSNR} = 10 \log_{10} \left(\frac{\text{MAX}_I^2}{\text{MSE}} \right). \quad (20)$$

Here, O is the plain image whose dimensions are $m \times n$, E is the image with noise and MAX_I is the maximum pixel value of image. For 8-bit representation of pixels, value of MAX_I is equal to 255. To obtain better quality of reconstructed image, PSNR value for 8-bit representation must lie between 30 and 50 dB, whereas for 16-bit representation, it must lie between 60 and 80 dB. Tables 3 and 4 below show the performance analysis of decryption parameters i.e. PSNR and MSE.

Time analysis

A good image encryption algorithm should encrypt and decrypt images in very less time along with resisting all types

Table 3 Image decryption

Decrypted image using CT-ILM	Decrypted image using ILM only
Decrypted Original Image	Decrypted Original Image
Decrypted R-Component	Decrypted R-Component
Decrypted G-Component	Decrypted G-Component
Decrypted B-Component	Decrypted B-Component

of attacks. The time taken by the presented approach is evaluated using the experimental setup mentioned in the start of Sect. 5. The use of ILM with cosine transformation in the proposed algorithm helps in achieving faster speeds. The other portion of the algorithm which affects the time are, scrambling of pixels and random order substitution. Image size is also an important factor, which affects the time of execution. The large-sized image takes more time for encryption and decryption than the small sized image. The image size used in the presented approach is 256×256 . The time required

for encryption and decryption by the proposed approach are same i.e. 0.06 s. The presented approach takes less time both for encryption and decryption. Hence, it can be used for achieving faster speeds. To compute the time taken by the proposed approach time () function of Python has been used.

Comparison with existing approaches

The review of recent literature clearly shows that multi-dimensional chaotic maps are more efficient and secure than traditional one-dimensional chaotic maps. As described earlier also, simple one-dimensional chaotic maps suffer from issues like blank windows, stable windows and small key space. Complex and three-dimensional chaotic maps such as ILM overcome these limitations of one dimensional chaotic maps. It has non-zero Lyapunov exponent, which demonstrates that these exhibit good dynamical behaviour. Researchers have combined chaotic maps with various techniques such as cellular automata, optimization methods and DNA encoding to propose secure and efficient image encryption techniques. Two of such methods have been proposed in [20, 21]. The authors in [21], combined Differential optimization with ILM to propose a faster and secure method, and the authors in [20] used cosine-based transformation to overcome limitation of one-dimensional chaos maps. Motivated by these works, the method proposed in this paper combines ILM and cosine transformation to achieve target of secure and efficient image encryption. Table 5 describes the comparison of the proposed technique with the works proposed in [20, 21].

Conclusion and future work

In this paper, an image encryption approach that uses cosine function and ILM has been discussed. The technique applies SHA-256 hash function to provide resistance against the brute force attacks, ILM for permuting the image pixels and cosine function for producing more non-linearity in the

Table 4 PSNR and MSE

Color component	Approach	Decryption parameters	
		PSNR	MSE
Full image	CT-ILM	37.8734808862	10.61041259765625
	ILM Only	34.2021938708	24.709330240885418
R-component	CT-ILM	36.2561704747	15.397964477539062
	ILM Only	32.5585330077	36.07672119140625
G-component	CT-ILM	37.0466230204	10.716415405273438
	ILM Only	32.4022543971	31.223846435546875
B-component	CT-ILM	39.4333863851	5.71685791015625
	CT-ILM	38.6623921973	6.827423095703125

Table 5 Comparison with earlier proposed methods

Approach and criteria	Hua et al. [20]			Dua et al. [21]			Proposed approach		
	LM + Sine map with Cosine Grayscale, color			ILM + DE Grayscale, color			ILM with Cosine Grayscale, color		
Image type	Red component	Green component	Blue component	Red component	Green component	Blue component	Red component	Green component	Blue component
NPCR	99.6061	99.6170	99.6368	99.5803	99.5986	99.6307	99.6154	99.6276	99.5986
UACI	33.4683	29.518	27.594	33.155	30.4375	27.6685	32.979	30.407	27.698
Entropy	–	7.997405	7.997388	7.997246	7.997089	7.997403	7.997575	7.997625	7.997098
CCH	– 0.003280	0.056922	– 0.017732	– 0.008614	– 0.0183159	– 0.02645	– 0.030300	– 0.016901	– 0.046878
CCV	– 0.000777	– 0.062442	0.056556	0.002536	– 0.006633	0.072281	– 0.026553	0.016343	– 0.046216
CCD	– 0.000181	0.019343	– 0.05922	– 0.051130	0.0297551	0.0098166	– 0.003057	0.024168	– 0.004554
Histogram	Uniform	Uniform	Uniform	Uniform	Uniform	Uniform	Uniform	Uniform	Uniform
MSE	–	–	–	15.397964	10.716415	5.716857	36.076721	31.223846	6.827423
PSNR	–	–	–	36.256170	37.046623	39.433386	32.558533	32.402254	38.662392
Key space	256	SHA-2	–	SHA-2	–	–	SHA-2	–	–
Encryption time	0.09 s	600 s	–	0.06 s	–	–	0.06 s	–	–
Decryption time	–	–	–	0.06 s	–	–	0.06 s	–	–

generated sequence. High entropy values and low correlation coefficient values obtained after performing decryption depict that the presented approach is more secure and efficient for image encryption. The decryption parameters obtained also depict that the decrypted image is almost near to the original image and does not have much distortion. The proposed approach results in reduction in encryption and decryption time. Hence, ILM with cosine function can be used to obtain a quicker and more efficient way to encrypt the image.

Compliance with ethical standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

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/>.

References

- Chen Y, Hsia C, Zhong S et al (2018) Data hiding method for AMBTC compressed images. *J Ambient Intell Human Comput*. <https://doi.org/10.1007/s12652-018-1048-0>
- 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: 2018 International conference on inventive research in computing applications (ICIRCA), Coimbatore, pp 676–682. <https://doi.org/10.1109/ICIRCA.2018.8597245>
- Jaroli P, Bisht A, Dua M, Dua S (2018) A color image encryption using four dimensional differential equations and arnold chaotic map. In: 2018 International conference on inventive research in computing applications (ICIRCA), Coimbatore, pp 869–876. <https://doi.org/10.1109/ICIRCA.2018.8597310>
- Suneja K, Dua S, Dua M (2019) A review of chaos based image encryption. In: 2019 3rd International conference on computing methodologies and communication (ICCMC), Erode, India, pp 693–698. <https://doi.org/10.1109/ICCMC.2019.8819860>
- Enayatifar R, Abdullah AH, Isnin IF, Altameem A, Lee M (2017) Image encryption using a synchronous permutation-diffusion technique. *Opt Lasers Eng* 90:146–154
- Li S, Chen G, Cheung A, Bhargava B, Lo KT (2007) On the design of perceptual MPEG-video encryption algorithms. *IEEE Trans Circuits Syst Video Technol* 17(2):214–223
- Solak E, Çokal C, Yildiz OT, Biyikoğlu T (2010) Cryptanalysis of Fridrich's chaotic image encryption. *Int J Bifurc Chaos* 20(05):1405–1413
- Solak E, Çokal C (2011) Algebraic break of image ciphers based on discretized chaotic map lattices. *Inf Sci* 181(1):227–233
- Wang X, Liu L, Zhang Y (2015) A novel chaotic block image encryption algorithm based on dynamic random growth technique. *Opt Lasers Eng* 66:10–18
- Mandal MK, Banik GD, Chattopadhyay D, Nandi D (2012) An image encryption process based on chaotic logistic map. *IETE Tech Rev* 29(5):395–404
- Matthews R (1989) On the derivation of a “chaotic” encryption algorithm. *Cryptologia* 13(1):29–42
- Wang S, Liu W, Lu H, Kuang J, Hu G (2004) Periodicity of chaotic trajectories in realizations of finite computer precisions and its implication in chaos communications. *Int J Mod Phys B* 18(17n19):2617–2622
- Kocarev L (2001) Chaos-based cryptography: a brief overview. *IEEE Circuits Syst Mag* 1(3):6–21
- Zhang Y, Li C, Li Q, Zhang D, Shu S (2012) Breaking a chaotic image encryption algorithm based on perceptron model. *Nonlinear Dyn* 69(3):1091–1096
- Rhouma R, Belghith S (2008) Cryptanalysis of a new image encryption algorithm based on hyper-chaos. *Phys Lett A* 372(38):5973–5978
- Xie J, Yang C, Xie Q, Tian L (2009) An encryption algorithm based on transformed logistic map. In: 2009 International conference on networks security, wireless communications and trusted computing, Wuhan, Hubei, pp 111–114. <https://doi.org/10.1109/NSWCTC.2009.201>
- Sam IS, Devaraj P, Bhuvaneswaran RS (2012) An intertwining chaotic maps based image encryption scheme. *Nonlinear Dyn* 69(4):1995–2007
- Ye G, Huang X (2017) An efficient symmetric image encryption algorithm based on an intertwining logistic map. *Neurocomputing* 251:45–53
- Hua Z, Jin F, Xu B, Huang H (2018) 2D Logistic-sine-coupling map for image encryption. *Signal Process* 149:148–161
- Hua Z, Zhou Y, Huang H (2019) Cosine-transform-based chaotic system for image encryption. *Inf Sci* 480:403–419
- Dua M, Wesanekar A, Gupta V et al (2020) Differential evolution optimization of intertwining logistic map-DNA based image encryption technique. *J Ambient Intell Human Comput* 11:3771–3786. <https://doi.org/10.1007/s12652-019-01580-z>
- Wang X, Xu D (2014) Image encryption using genetic operators and intertwining logistic map. *Nonlinear Dyn* 78(4):2975–2984
- Huang X, Ye G (2018) An image encryption algorithm based on time-delay and random insertion. *Entropy* 20(12):974
- Suri S, Vijay R (2019) A synchronous intertwining logistic map-DNA approach for color image encryption. *J Ambient Intell Human Comput* 10:2277–2290. <https://doi.org/10.1007/s12652-018-0825-0>
- Srivastava R, Singh O (2015) Performance analysis of image encryption using block based technique. *Int J Adv Res Electric Electron Instrum Eng* 4(5):4266–4271
- Bisht A, Dua M, Dua S (2019) A novel approach to encrypt multiple images using multiple chaotic maps and chaotic discrete fractional random transform. *J Ambient Intell Human Comput* 10(9):3519–3531

27. Dua M, Wesanekar A, Gupta V, Bhola M, Dua S (2019) Color image encryption using synchronous CML-DNA and weighted biobjective genetic algorithm. In: Proceedings of the 3rd international conference on big data and internet of things (BDIOT 2019). Association for Computing Machinery, New York, NY, USA, pp 121–125. <https://doi.org/10.1145/3361758.3361780>
28. Bisht A, Dua M, Dua S, Jaroli P (2019) A color image encryption technique based on bit-level permutation and alternate logistic maps. *J Intell Syst* 29(1):1246–1260
29. Nancharla BK, Dua M (2020) An image encryption using intertwining logistic map and enhanced logistic map. In: 2020 5th International Conference on Communication and Electronics Systems (ICCES). IEEE, pp 1309–1314

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