Signal Processing 164 (2019) 249–266

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

An image encryption scheme based on hybridizing digital chaos and finite state machine

Moatsum Alawida*, Je Sen Teh, Azman Samsudin, Wafa' Hamdan Alshoura

School of Computer Sciences, Universiti Sains Malaysia, Pulau, 11800 USM Pinang, Malaysia

ARTICLE INFO

Article history:
Received 17 March 2019
Revised 14 May 2019
Accepted 8 June 2019
Available online 11 June 2019

Keywords:
Chaos theory
Cryptography
Tent map
Deterministic finite state machine
Fixed-point number
Image encryption

ABSTRACT

Image encryption protects visual information by transforming images into an incomprehensible form. Chaotic systems are used to design image ciphers due to properties such as ergodicity and initial condition sensitivity. A chaos-based cipher derives its security strength from its underlying digital chaotic map, thus a more complex map leads to higher security. This paper introduces an enhancement to a tent map's chaotic properties by hybridizing it with a deterministic finite state machine. We denote the resulting digital one-dimensional chaotic system as TM-DFSM. Chaotic analyses indicate that the new chaotic system has higher nonlinearity, sensitivity to initial condition, and larger chaotic parameter range than other recently proposed one-dimensional chaotic systems. We then propose a new image encryption scheme based on TM-DFSM, capable of performing both confusion and diffusion operations in one pass while also having a flexible key space. The encryption operations are designed to achieve maximal confusion and diffusion properties. Changing a single bit of the plainimage or secret key will result in an entirely different cipherimage. The proposed cipher has been analyzed using histogram analysis, contrast analysis, local Shannon entropy, resistance against differential cryptanalysis, and key security. Performance comparison with other recent schemes also depicts the proposed cipher's superiority.

1. Introduction

Digital chaotic systems have many shared properties with cryptography that motivate their adoption in the design of cryptographic algorithms such as hash functions [1] and image ciphers [2–5]. These properties include sensitivity to initial conditions, nonlinearity, unpredictability and ergodicity. Both high dimensional (HD) and one dimensional (1D) chaotic maps are widely adopted to encrypt digital images. Simple 1D chaotic systems are more popular due to their computational efficiency. However, 1D maps suffer from various security shortcomings due to their simple mathematical structures. These security issues range from a small chaotic parameter range, low sensitivity to initial conditions and/or control parameters, low ergodicity, and low complexity. Therefore, the researchers have proposed many chaotification methods to overcome these loopholes and enhance chaotic performance.

Various chaotification methods focus on enhancing different security aspects of unimodal maps, leading to improved chaotic behaviors. Table 1 shows a summary of chaotification methods that focus on enhancing 1D chaotic maps. The high computing precision method involves using higher precision representations to extend cycle lengths but is impractical for actual applications. The cascade method cascades multiple maps together, using one as a seed for another. The chaotic complexity of the overall chaotic system is improved if the seed maps depict chaotic behaviors, thus selection of the seed maps and their parameters play an important role. Poor selection choices can lead to no improvements in chaotic properties despite increasing computational complexity. The switching method involves selecting between more than one chaotic map, which increases cycle length and weakens correlations. The switching rule or switching sequence plays the most important role in the chaotic system's properties, whereby the sequence can either be random or sequential. Similar to the previous cascade method, poor selection of the rule or sequence may lead to no improvements in the chaotic properties of the overall system. As its name implies, the combination method combines two or more chaotic maps to produce a single chaotic map. The combination method mostly relies on the modulo operation which has patterns that are to easy predict. In addition, combining many chaotic maps is not practical method for chaos-based applications as it needlessly increases the number of parameters and overall system complexity. To address dynamical degradation and increase cycle length, another chaotification method is through the use of perturbation. The perturbation source, which can either be an external or internal source, dictates the overall behavior of the system. An external

* Corresponding author.
E-mail address: matsm88@yahoo.com (M. Alawida).

https://doi.org/10.1016/j.sigpro.2019.06.013
0165-1684/

Table 1
Comparison of chaotification methods.

Method	Description	External source	Focus problems	Advantage(s)	Disadvantage(s)
Using high precision [21]	Increase the bit precision for a computer	No	Dynamical degradation	The averaged cycle length is prolonged	High implementation cost
Cascade [10,22–25]	Using map outputs to seed another	Yes	Dynamical degradation, chaotic range, data distribution	Prolonged cycle, flexibility	Non-uniform data distribution, regular dynamics
Switching [26,27]	Select chaotic maps based on predefined rules	Yes	Dynamical degradation, chaotic range, parameter identification, correlation	Reduce correlation, uniform data distribution	Strong dependence on rule design
Combination [17,28–34]	Combine chaotic maps	Yes	Dynamical degradation, chaotic range, complexity	Prolonged cycle, uniform data distribution, high complexity and ergodicity	High computational cost
Perturbation [35–41]	Perturb chaotic map using internal or external entropy source	Method dependent	Dynamical degradation, chaotic range, parameter identification	High ergodicity, uniform data distribution, low correlation, high complexity, resists parameter estimation	Dependent on perturbation source
Compensation [42,43]	Compensate chaotic points by under different bit precisions	Method dependent	Dynamical degradation	High complexity	High computational cost, limited chaotic range

source has advantages in terms of chaotic range and cycle length as compared to an internal source but involves more computational overhead. Finally, the error compensation method is an effective technique to remove dynamical degradation of digital chaos, where various bit precision, Lyapunov exponent and time-varying parameters are employed in the compensation process to generate a new chaotic map. However, this method limits the parameter space, which can lead to practical or implementation drawbacks such as a smaller keyspace. Moreover, the method requires an external source to provide the required parameters, and it is also difficult to extend this method for HD maps.

Most of the proposed methods such as cascade, switching and perturbation require an external entropy source to enhance a simple 1D chaotic map. These sources may include a continuous chaotic system, a random number sequence, or additional digital chaotic maps. These techniques adversely affect the efficiency of the resulting maps. In addition, the external entropy source itself may suffer from statistical defects. Rather, a chaotification method should be simple, without needing an external source to generate excellent chaotic behaviors.

In image encryption, conventional ciphers such as the Advanced Encryption Standard (AES) cannot applied in a straightforward manner to encrypt images due to image properties such as high information capacity and strong correlation among adjacent data. Instead, its diffusion property needs to be improved for image encryption purposes such as by using a modified CBC mode [6]. To tackle this issue, many image encryption algorithms have been proposed as an alternative way to protect image data using various techniques, such as compressive sensing [7], quantum theory [8], DNA coding [9], chaos theory [2–5,10–12] and others [13]. Because chaotic maps have many properties that are aligned with the requirements of image encryption, chaotic maps are widely adopted in image encryption. For example, Hua et al. [14] proposed new maps using cosine-transform with two seed maps, and then further proposed a new image encryption using their new map. Li et al. [12] used the classical tent chaotic map to propose a new image encryption, whereby a keystream is generated to encrypt pixel values directly. However, this scheme has been cryptanalyzed by Wu et al. [15] who successfully launched known and chosen plaintext attacks on Li's scheme. In the same paper, Wu et al. also improved the Li's scheme but this improvement was further cryptanalyzed by Zhu and Sun [16] by obtaining equivalent keys. Pak and Huang [17] defined a new chaotic map based on a combination of two Sine maps. This map was then used in an image cipher whereby keystreams were generated and used to permute image pixels, followed by a linear transform for diffusion to obtain the final encrypted image. Wang et al. [18] managed to cryptanalyze Pak's scheme by using a chosen plaintext attack. Chen et al. [19] proposed another image encryption scheme using electrocardiogram ECG signals, autoblocking, and chaotic maps. The scheme used ECG signals to generate initial keys before performing image encryption based on auto-blocking of the image matrix. Li et al. [20] showed that Chen's scheme is vulnerable to a known plaintext attack, in which an equivalent key can be obtained to decrypt other cipherimages under the same key.

Generally, chaos-based image encryption schemes have some common characteristics as follows:

1. The secret key usually depends on initial conditions and system parameters. Most chaos-based image ciphers rely on a quantized value of the secret key to suit the range of chaotic parameter and phase space of initial condition [2,3]. This limits the flexibility and size of the keyspace.
2. The length of the generated chaotic trajectory is normally equal to the number of image pixels. Using HD chaotic maps or multiple 1D chaotic maps to generate chaotic points will be costly and impractical for larger images [12].
3. Diffusion and confusion algorithms are often executed on the entire image separately [4,5]. Thus, simultaneously achieving both properties in one pass is still an existing problem for chaos-based image ciphers.
4. The complexity of the underlying chaotic system has a direct impact to the security of a cipherimage [2]. Issues with regards to digital chaos such as dynamical degradation could lead to security problems.
5. The computational efficiency is proportional to the computational complexity of the underlying chaotic system. Thus, the design of a chaos-based image cipher must balance a trade-off between efficiency and security [11].

In order to overcome all the issues mentioned above, this paper proposes a new digital chaotic system based on the combination between deterministic finite state machine (DFSM) and tent map, which we denote as TM-DFSM. TM-DFSM uses the tent map in both the machine states and state transition rules. The proposed chaotic map preserves the computational complexity of the underlying unimodal tent map while simultaneously enhancing chaotic

complexity without an external entropy source. For optimal efficiency and to ensure reproducibility, all real number computations are performed using fixed-point representation rather than floating point. Statistical analyses show that TM-DFSM has better complexity, higher nonlinearity, unpredictability, and larger chaotic parameter range than other recently proposed hybrid chaotic maps. Based on TM-DFSM map, we then propose a new image encryption scheme. In the proposed scheme, we focus on tackling the aforementioned chaos-based image encryption issues. Firstly, we improve the flexibility of the secret key in terms of key selection and size of keyspace. The initial conditions and control parameters do not factor into the key selection so there is no need for a quantization function. Secondly, the proposed cipher does not have a 1-to-1 dependency between the number of generated chaotic points and the number of image pixels. Thus, encryption efficiency is improved especially for larger images. Thirdly, the encryption process is designed to achieve diffusion and permutation properties in one pass for each encryption round. Chaos-based sorting is used once in each round to achieve confusion property whereas the combination of encrypted pixels and key information is used to achieve high diffusion. Experimental results and security analyses show that our scheme can encrypt plainimages of any size and type into an unrecognizable noise-like output, and is resistant to various well-known attacks.

We have previously addressed some of these problems in [10], which involves a different approach compared to the methodology proposed in this paper. Firstly, in the previous work, we use a different hybrid method based on combination and cascade methods to generate a new chaotic map whereas in this paper, we leverage the behavior of DFSMs. As previously mentioned, combination and cascade methods come with their own disadvantages which will be addressed using the new chaotification method. Another major difference between lies in the design of the encryption algorithm. In the previous work, there is still a distinct confusion and diffusion process whereas the current proposed work is designed to perform both processes in one-pass for optimal efficiency.

The rest of this paper is organized as follows: Section 2 discusses the concepts behind TM-DFSM and its chaotic analyses. Section 3 proposes the new image encryption scheme followed by Section 4 that provides its experimental verification and discussion. Section 5 discusses the proposed scheme's capability for image authentication before Section 6 concludes the paper with some final statements.

2. Combination tent map and deterministic finite state machine

This section introduces TM-DFSM which is a new digital chaotic system based on the tent map and DFSM. Its chaotic performance and complexity are also discussed in this section.

2.1. Tent map

Tent map is a 1D chaotic map and piecewise function that has one control parameter. It widely used in chaos-based cryptographic algorithms [12]. Mathematically, it is defined as

$$x_{n+1} = f(x, r) = \begin{cases} \frac{r}{2}x_n & \text{if } x_n < 0.5 \\ \frac{r}{2}(1 - x_n) & \text{if } x_n \geq 0.5, \end{cases} \quad (1)$$

where r is control parameter that has an interval $r \in [0, 4)$. The tent map depicts chaotic behavior when $r \in (2, 4)$ as seen in its bifurcation diagram in Fig. 9b. As r approaches 4, the chaotic complexity and performance improves. However, the tent map has a limited chaotic range which has a negative effect on the key space when applied in chaos-based encryption schemes. Furthermore, the tent map is not ergodic for majority of its phase space, especially in the range of [0,1], which results in non-uniform distribution in the cipherimage. Consequently, the tent map should be enhanced before being adopted in an image encryption scheme.

2.2. Deterministic finite state machines

Finite state machines (FSM), also known as deterministic finite automata (DFA) is one of the automata models widely used in computational linguistics [44]. It contains a set of states (also called nodes or vertices), a start state, a final state (accepted state), and a set of directed links between machine states called the transition or arc. The state transition rule is a rule that governs how the states transition between one another. FSM is used to investigate a string of symbols based on a given regular expression. Fig. 1 shows as an example of FSM and the acceptance process. The FSM will accept a string if the symbols in the string will lead to the final state.

2.3. TM-DFSM

TM-DFSM is a new digital chaotic map resulting from combining DFSM and with the tent map. It maintains the computation complexity of the tent map and while enhancing its chaotic behavior without needing an external entropy source. In TM-DFSM, each machine state is associated with a tent map and there is a set of directed links between the states labeled as 1s and 0s. To simplify descriptions in this paper, we limit the number of machine states to three, which results in six links or transitions. The states are connected to one another as shown in Fig. 2. The control parameter for each of the chaotic maps can either vary or be the same depending on user preference. For example, A_1, A_2, and A_3 are three machine states mapped to three tent maps with $r_1 = 3.9$, $r_2 = 3.989$, and $r_3 = 3.8989$, respectively. The proposed chaotic system can be easily extended to involve more machine states and a larger number of transitions. Increasing the size of the system does not affect the computational complexity of the overall system because only one of the underlying tent maps will be iterated each time.

The state transition rule plays an important role in TM-DFSM because it not only dictates the transition to the next state but also selects the bitwise permutation that will be performed on the tent map's chaotic points. This bitwise permutation will be detailed later on in this subsection. The state transitions of TM-DFSM can either be controlled by an external binary stream or dynamically controlled based on the chaotic trajectory. If an external bitstream is used, the state transitions of TM-DFSM corresponds to each bit of the aforementioned stream. On the other hand, dynamic control is based on a threshold value α, the chaotic point, x_i and a piecewise function, which can be described mathematically as

$$F(x_{i-1}) = \begin{cases} 1 & if \quad x_{i-1} > \alpha \\ 0 & if \quad x_{i-1} \leq \alpha \end{cases} \quad (2)$$

where α is a value within range of phase space, and 1 and 0 are the resulting state transition labels.

Fixed-point number is used to represent real numbers in TM-DFSM because it ensures the reproducibility of results, which is important for cryptographic algorithms [1,45]. We use $U(i, f)$ for an unsigned fixed-point number, where i bits are used to represent integers and f bits are used to represent the fractional portion of the real number. Because the chaotic points of the tent map are bounded to [0,1], we only extract the fraction section of chaotic point by $U(0, 52)$ at each iteration.

Let the indices of each bit in the $U(0, 52)$ representation be denoted as $\{1, 2, \ldots, 51, 52\}$ (most significant bit (MSB) $\rightarrow$ least significant bit (LSB)), and the overall value of the fixed-point number can be calculated as $f_{value} = \sum_{j=1}^{52} b \times 2^{-j}$, where $b = \{0, 1\}$. As

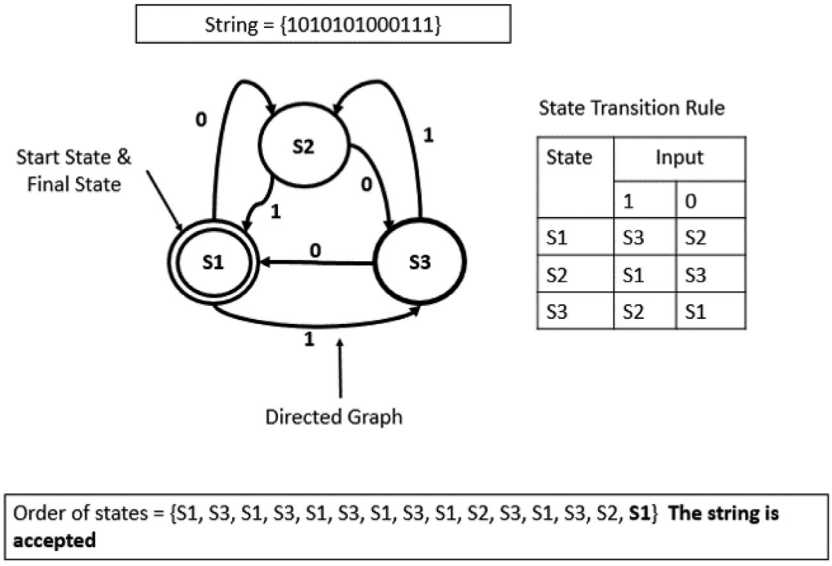

Fig. 1. FSM of three state machines and example of accepting process.

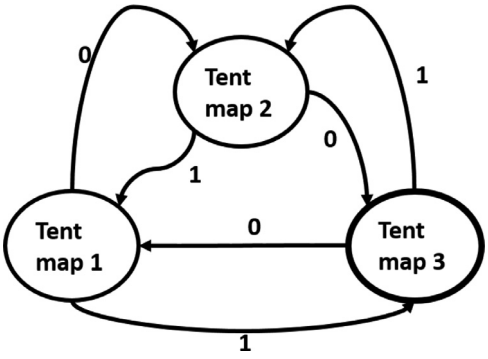

Fig. 2. DFSM of three machine states and three tent maps.

previously mentioned, the state transition labels, 1 and 0 will decide on the type of bitwise permutation that will be carried out on each chaotic point after iterating the tent map. This is performed by shuffling the bits of the $U(0, 52)$ representation by interleaving the MSBs with the LSBs. In this paper, we use two simple interleaving patterns that are each associated with the 0 or 1 labels as shown below:

Label 0: $P_1 = 1, 27, 2, 28, \ldots, 25, 51, 26, 52$. The resulting value can be calculated as $f_{value} = \sum_{j=1}^{52} b \times 2^{-P_1(j)}$.

Label 1: $P_2 = 27, 1, 28, 2, \ldots, 51, 25, 52, 26$. The resulting value can be calculated as $f_{value} = \sum_{j=1}^{52} b \times 2^{-P_2(j)}$.

Both of these bitwise permutation patterns are visually depicted in Fig. 3. Each iteration of TM-DFSM can be summarized as follows: each machine state will iterate its associated tent map and produce an *intermediary* chaotic point. The previous transition label is then used to decide which pattern will be used to permute the bits of the *intermediary* point to produce the *final* chaotic point. The *final* chaotic point will be computed by the next state based on the state transition rule. As previously mentioned, the state transitions are either dictated by an external bitstream or calculated dynamically based on Eq. 2.

2.4. TM-DFSM Chaotic analysis

The proposed TM-DFSM chaotic map can enlarge the chaotic parameter range and enhance the chaotic complexity of the underlying tent map. To exhibit its strength, various chaos-based evaluations were conducted for the proposed map and compared against the tent map and several recently proposed chaotic maps [23,30]. The evaluation methods were selected based on the different chaotic aspects such as bifurcation diagram for evaluating the chaotic parameter range, Lyapunov exponent for indicating chaoticity and sensitivity to initial conditions, fuzzy correlation dimension as a measure of the attractor's strangeness and fractal dimension in chaos, and symplectic entropy as a strong indicator of chaotic complexity.

2.4.1. Bifurcation diagram

Generally, the chaotic parameter range provides an indication of the key space size for chaos-based cryptographic algorithms. A wide chaotic parameter range is preferred because it has robust resistance against the brute force attacks. To analyze chaotic range, we can use a bifurcation diagram to show the relationship between chaotic points and control parameters. Fig. 4 compares the bifurcation diagrams between TM-DFSM, tent map, and two chaotic maps (x-dimension of the coupled logistic maps [30] and LS-S map [23]. All of these maps were implemented on Matlab using same the initial condition, $x_0 = 0.2$ and same control parameter range. One can see that the proposed map has a large of chaotic behaviors with minimal windows of periodicity (indicated by unshaded areas), while the tent and other new maps have smaller chaotic parameter ranges with many periodic regions. It can also be noted that TM-DFSM can produce chaotic behaviors for control parameter values that lead to non-chaotic behavior of the tent map. This implies that TM-DFSM can provide a large key space for chaos-based encryption. In fact, the chaotic range of TM-DFSM actually extends more than 4 because the fixed-point representation helps to extract the fractional portion and prevents the chaotic map from being iterated out of scope. However, for the purpose of this paper, we focus on using the normal range of [0, 4] to demonstrate the strength of the TM-DFSM map.

2.4.2. Lyapunov exponent

The main important feature that indicates chaos is sensitivity to initial conditions and control parameters. LE is a quantitative description of the divergence rate for two trajectories that start from two infinitesimally close initial points as time elapses to infinity [46]. A positive LE value indicates chaos whereby

First Permutation

1, 27, 2, 28, 3, 29, 4, 30, 5, 31, 6, 32, 7, 33, 8, 34, 9, 35, 10, 36, 11, 37, 12, 38, 13, 39, 14, 40, 15, 41, 16, 42, 17, 43, 18, 44, 19, 45, 20, 46, 21, 47, 22, 48, 23, 49, 24, 50, 25, 51, 26, 52

Second Permutation

27, 1, 28, 2, 29, 3, 30, 4, 31, 5, 32, 6, 33, 7, 34, 8, 35, 9, 36, 10, 37, 11, 38, 12, 39, 13, 40, 14, 41, 15, 42, 16, 43, 17, 44, 18, 45, 19, 46, 20, 47, 21, 48, 22, 49, 23, 50, 24, 51, 25, 52, 26

Fig. 3. The first and second bitwise permutation.

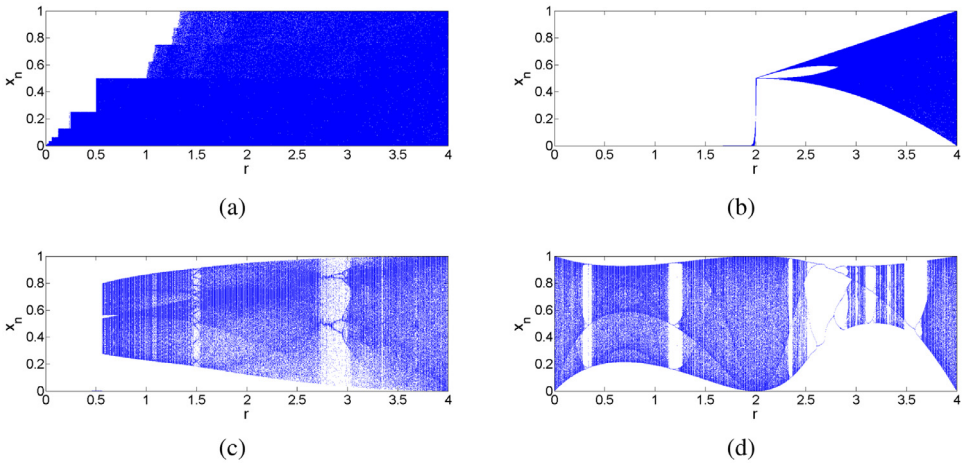

Fig. 4. Bifurcation diagrams of (a) TM-DFSM map, (b) tent map, (c) x-dimension of coupled two logistic maps [30], (d) and LS-S map [23].

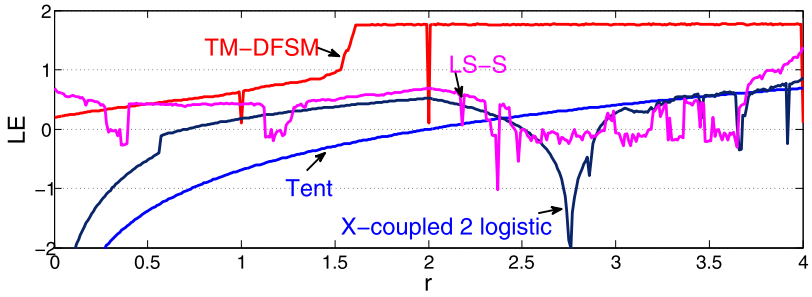

Fig. 5. LE values of TM-DFSM map, tent map, x-dimension coupled two logistic maps, and LS-S map.

after a certain number of iterations, the trajectories would have completely diverged. Moreover, a larger LE value implies higher unpredictability and sensitivity whereas a negative or zero LE value implies periodic behavior and the chaotic trajectories will remain within a small region of phase space after each iteration. In this paper, we use the LE estimation method based on [20] and plot the results in Fig. 5. The plot indicates that TM-DFSM not only has positive LE values but it also has the largest LE values for majority of the parameter settings as compared to the other maps. It can be concluded that TM-DFSM is superior to the other maps in terms of unpredictability and sensitivity to input parameters.

Although TM-DFSM depicts chaotic behavior for most of the parameter settings, it has degraded chaotic properties when $r \in \{1, 2, 4\}$. This is reflected by the slight dips in the LE plots in Fig. 5. For these parameter values, the chaotic trajectories are strongly attracted to several fixed points, $x \in \{0, 0.5, 1\}$. These fixed points exist due to the characteristics of the permutation patterns P_1 and P_2, which are depicted in Fig. 6. To further analyze this phenomena, we examine the behavior of TM-DFSM under several scenarios. In these scenarios, we observe the results of both possible permutation patterns because the selection of the permutation is implementation dependent. (See Section 2.3 for details.). If

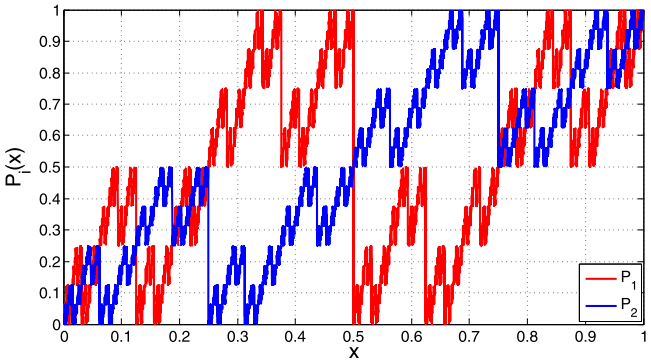

Fig. 6. Permutation behavior for P_1 and P_2.

$$r = 1, x_n = 0.5, x_{n+1} = 0.5 \times 0.5 = 0.25,$$

then the result of P_1 and P_2 are 0.5 and $7.4506e-09$ respectively. If

$$r = 2, x_n = 0.5, x_{n+1} = (1 - 0.5) = 0.5,$$

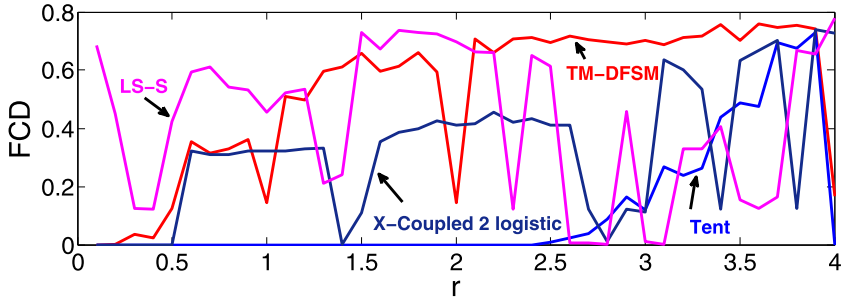

Fig. 7. FCD values of TM-DFSM map, tent map, x-dimension coupled two logistic maps, and LS-S map.

then the result of first and second permutation are $7.4506e-09$ and 0.5 respectively. If

$$r = 4, x_n = 0.5, x_{n+1} = 2 \times (1 - 0.5) = 1,$$

then the result of first and second permutation are 1. In all of these situations, the chaotic points may be attracted to fixed points depending on the type of permutation that is used.

2.4.3. Fuzzy correlation dimension

Strange attractor phenomenon occurs when chaotic points are folded and stretched in the phase space as time tends to infinity. When a nonlinear dynamical system trajectory lies in a strange attractor, the system depicts maximal chaotic and nonlinear characteristics. To measure the strangeness of a chaotic system, correlation dimension (CD) can be calculated. CD is an estimator of fractal dimension that evaluates the strange attractor geometries in the phase space [47]. CD is defined based on the embedding dimension, correlation integral, and Heaviside function. For more accurate analysis, we propose a new enhancement to the CD method using a fuzzy membership function instead of the Heaviside function. This enhancement, which we refer to as the fuzzy correlation dimension (FCD), allows us to study the relational distance between chaotic points under the correlation integral.

CD can be estimated based on [47]. A sequence of a chaotic signal is denoted as $\{x_i, |i = 1, 2, \ldots, N\}$, where N is number of points. Thus, the CD is defined as

$$CD = \lim_{r \to 0} \lim_{N \to \infty} \frac{\log C_e(r)}{\log r} \tag{3}$$

$$C_e(r) = \frac{1}{[N-(m-1)\xi][N-(m-1)\xi-1]} \times \sum_{i=1}^{N-(m-1)\xi} \sum_{j=i+1}^{N-(m-1)\xi} \theta(r - |x(i) - x(j)|) \tag{4}$$

where $C_e(r)$ is the correlation integral, r is a tolerance value, $\Theta(w)$ is the Heaviside function which is a piecewise function ($\Theta(w) = 0$ if $w \leq 0$, $\Theta(w) = 1$ if $w > 0$), m is the embedding dimension, and ξ is the time delay which is equal to 1 for discrete chaotic maps. CD can be estimated by calculating the exponent v in $(C_e(r)) \propto r^v$. Thus, the slope of the log-log of $C_e(r)$ versus r can provide the approximate CD value for chaotic signals and other real signals.

As mentioned earlier, we use a fuzzy membership function $\theta(w)$, where $w = |x(i) - x(j)|$ instead of Heaviside function to calculate the similarity degree between signals and remove biases between close tolerance values. $\theta(w)$ is defined as

$$\theta(w) = \begin{cases} 1 & if \quad w \leq \frac{r}{2} \\ 0 & if \quad w \geq r \\ \frac{(r-w)}{\frac{r}{2}} & if \quad x \in \left(\frac{r}{2}, r\right). \end{cases} \tag{5}$$

FCD is calculated based on the trajectories generated from TM-DFSM, tent map, and two new chaotic maps, then plotted as shown in Fig. 7. The FCD values of the TM-DFSM map have larger FCD values than the other maps for most of the parameter values. Moreover, the tent map has small FCD values indicating low dimensionality for the most parameter settings. On the other hand, TM-DFSM has high fractal dimension, complex irregular behaviors, and strange attractors as indicated by the FCD results.

2.4.4. Chaotic complexity

Nonlinearity and complexity are regarded as significant features in chaos. A system is considered to be complex when a lot of information is needed to describe its behavior. In other words, a complex chaotic system generates nonlinear chaotic behavior that has little to no observable patterns in its trajectories. This property can be estimated by using the notions of entropy. Many methods have been proposed to estimate the complexity of a system such as Shannon entropy [48], approximate entropy (ApEn) [49], sample entropy (SampEn) [50], fuzzy entropy (FuzzyEn) [51], and symplectic entropy (SymEn) [52]. In this paper, we select SymEn to estimate the chaotic complexity due to its high accuracy and qualitative estimation.

SymEn is a new method to estimate the nonlinearity and complexity [52]. It utilizes the entropy notion and symplectic geometry space to provide an indicator of complexity and nonlinear characteristics of a chaotic signal or data sequence. SymEn can be measured based on the energy distribution entropy of an attractor for a data sequence in symplectic space. Based on [52,53], we calculate SymEn based on the following steps:

1. 1D dimensional chaotic points $(x_i, i = 1, 2, \ldots, N)$ can be reconstructed an attractor $X_{l \times m}$ based on the embedding dimension in phase space, which called the trajectory matrix as the shown system 6.

$$X = \begin{pmatrix} X_1^m \\ X_2^m \\ \vdots \\ X_3^m \end{pmatrix} = \begin{pmatrix} x_1 & x_2 & \ldots & x_m \\ x_2 & x_3 & \ldots & x_{m+1} \\ \vdots & \vdots & \ldots & \vdots \\ x_l & x_{l+1} & \ldots & x_N \end{pmatrix} \tag{6}$$

   where $X_i^m = \{x_i, x_{i+1}, \ldots, x_{i+m-1}\}$, m is the embedding dimension. $l = N - m + 1$ is the number of values in R^m.
2. Transform the attractor X to Hamilton matrix M in the symplectic space. Hamilton matrix M is defined as

$$M = \begin{pmatrix} A & 0 \\ 0 & -A^T \end{pmatrix} \tag{7}$$

   where $A = X^T X$. M is equivalent to the attractor X.
3. By symplectic similar transform, eigenvalues of M are obtained.
4. Hamilton matrix M is transformed into an upper Hessenberg matrix B ($b_{ij} = 0, i > j + 1$) by using a symplectic matrix H:

$$H^T M H = \begin{pmatrix} B & R \\ 0 & B^T \end{pmatrix} \tag{8}$$

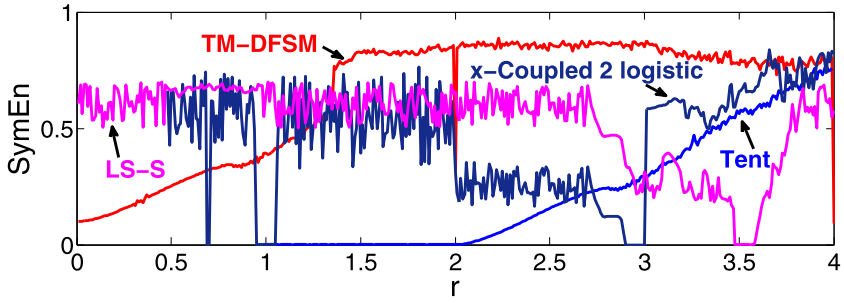

Fig. 8. SymEn values of TM-DFSM map, tent map, x-dimension coupled two logistic maps, and LS-S map.

where H can be calculated based on Household matrix as

$$H = H(k, w) = \begin{pmatrix} P & 0 \\ 0 & P \end{pmatrix} \tag{9}$$

where P can be defined as:

$$P = I_n - \frac{2\omega^{\omega} *}{\omega * \omega}$$
$$\omega = (0, \ldots, 0; \omega_k, \ldots, \omega_n)^T \neq 0 \tag{10}$$

where I_n is a n-dimensional unit matrix after symplectic similar transformation on M.

5. To compute the upper triangle matrix B that equal $A^{(n)}$, the sequential transformation that starting from $A^{(1)}$ into $A^{(n)}$ based on P transformations, where $P^{(i)}, i = 1, 2 \ldots, n$. Matrix A can be obtained from system 7. The sequential transformation can be calculated as

$$P^{(i)}A^{(i)} = A^{(i+1)}, i = 1, 2, \ldots, n-1$$
$$P = P^{(n)}P^{(n-1)} \ldots P^{(1)} \tag{11}$$

6. Finally, the eigenvalues $\mu = \{\mu_1, \mu_2, ..., \mu_m\}$ of upper triangle matrix B are equal to all series matrices A. Sorted the eigenvalues by descending order, and these μ values reflect the probability distribution of the energy of chaotic system in all directions of symplectic space. Then, compute probability discrete function for eigenvalue values as

$$[M.P] = \begin{cases} M: & \mu_1, \mu_2, \ldots, \mu_m \\ P: & P_1, P_2, \ldots, P_m \end{cases} \tag{12}$$

$$p_i = \frac{\mu_i}{\sum_{i=1}^{m} \mu_i} \tag{13}$$

Then, the SymEn measure can be defined based on Shannon entropy as follows

$$SymEn = -\sum_{i=1}^{m} P_i log_2 P_i, \tag{14}$$

For more information, see [52,53]. SymEn values are calculated for TM-DFSM, tent map and the other new chaotic maps. The results are plotted in Fig. 8. Large SymEn values indicate high nonlinearity and better complexity. One can see that TM-DFSM map has better nonlinear properties than the other maps, which implies that TM-DFSM can generate complex behaviors and excellent randomness. Moreover, this also implies that TM-DFSM is highly resistant to parameter estimation attacks or pattern prediction.

2.5. Dynamical degradation analysis

Analog chaos generated by oscillations and electronic circuit systems have trajectories that will never repeat because it has infinite states. On the other hand, digital chaos has trajectories that will eventually fall into a cycle after a certain number of iterations. This is due to the finite precision used to represent chaotic points, such that chaotic points are infinitesimally close are considered as the same value, or are overlapped. When a chaotic trajectory has a short cycle, it suffers from many negative properties such as low complexity, low ergodicity, strong correlation, and non-uniform distribution. This phenomena is known as dynamical degradation.

Table 2
Statistics comparison of TM-DFSM and tent map based on SMN.

Criteria	TM-DFSM	Tent
Number of cycles	4	8
Number of fixed point s	1	1
Maximum length of cycle	6	6
Maximum transient length	9	5
Transient lengths and their numbers	(1, 11) (2, 9) (3, 7) (4, 3) (5, 2) (6, 4) (7, 4) (8, 4) (9, 1)	(1, 22) (2, 5) (3,7) (4, 4) (5, 2)

To study the dynamical degradation of TM-DFSM and the tent map under limited precision, we use the state-mapping network (SMN) methodology to investigate cycle lengths [54]. We select a 6-bit precision and use fixed-point numbers in this analysis. The control parameter $r = 3.99$ is used for both maps, and starting from 64 initial states (chaotic points) numbered from 0 to 63, the maps are iterated once. The results will fall into one of the 64 states. Fig 9 shows the comparison of both maps in terms of SMN. One can observe that the TM-DFSM has less number of cycles than the tent map and most of the cycle lengths of tent map are two iterations. For a more in-depth investigation, Table 2 also provides simple statistics about the number of cycles, length of cycle and number of state transitions (transient) before entering a cycle. One can observe that the TM-DFSM extends cycle lengths and enhances the complexity of the underlying tent map. Moreover, this SMN analysis is only based on 6-bit precision which implies that TM-DFSM will have an even longer cycle length when implemented with larger computing precisions such as 32 or 64 bits, which is advantageous for chaos-based applications.

3. TM-DFSM-Based Image encryption scheme

This section introduces a novel image encryption scheme based on TM-DFSM which can be represented as shown in Fig. 10. The proposed scheme consists of two main phases: key generation and image encryption. Both phases involves iterating TM-DFSM, however state transitions during the key generation phase is based on the secret key whereas the image encryption phase relies on the TM-DFSM's own chaotic trajectory for dynamical control. The overall encryption process requires only two rounds; the first round applies the permutation and diffusion operations in a column-wise manner while the second round applies both operations in a row-wise manner. For higher efficiency and also to facilitate reproducibility, all real number calculations use fixed-point

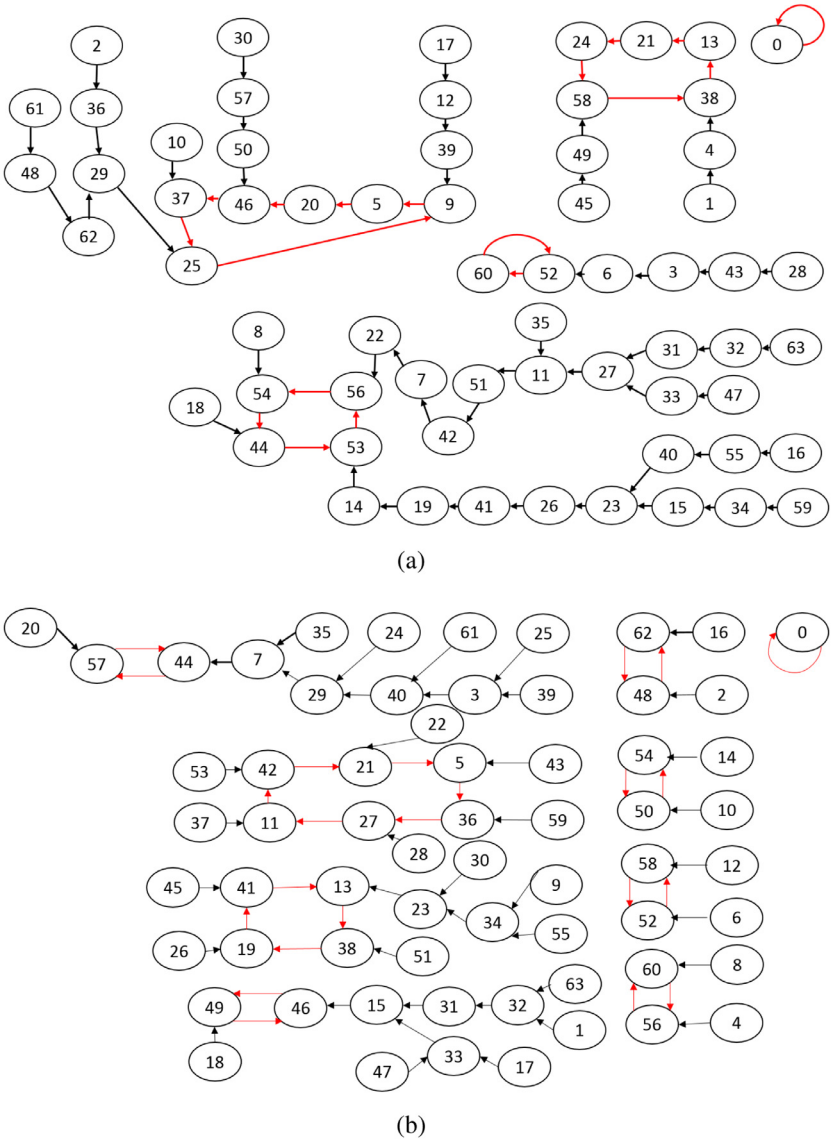

Fig. 9. SMN diagrams of (a) TM-DFSM map, (b) tent map.

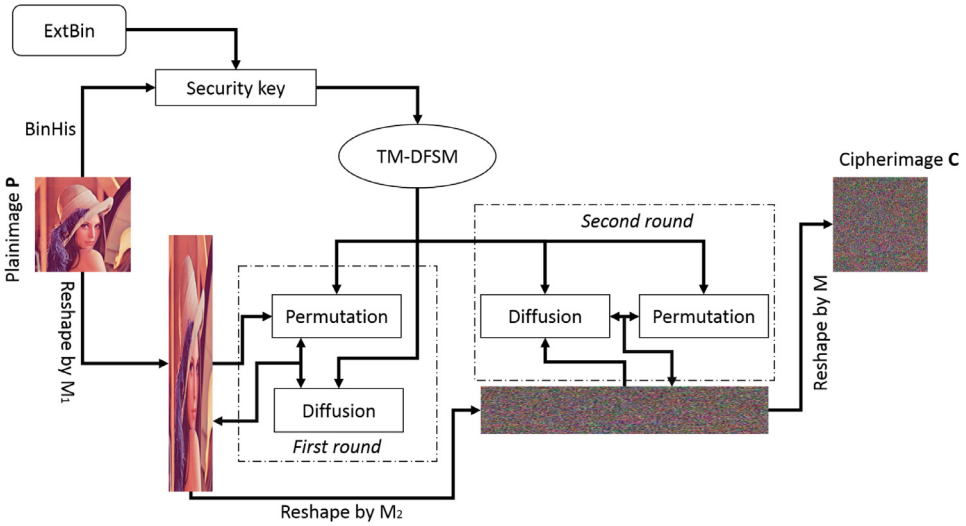

Fig. 10. Framework of the proposed image encryption scheme.

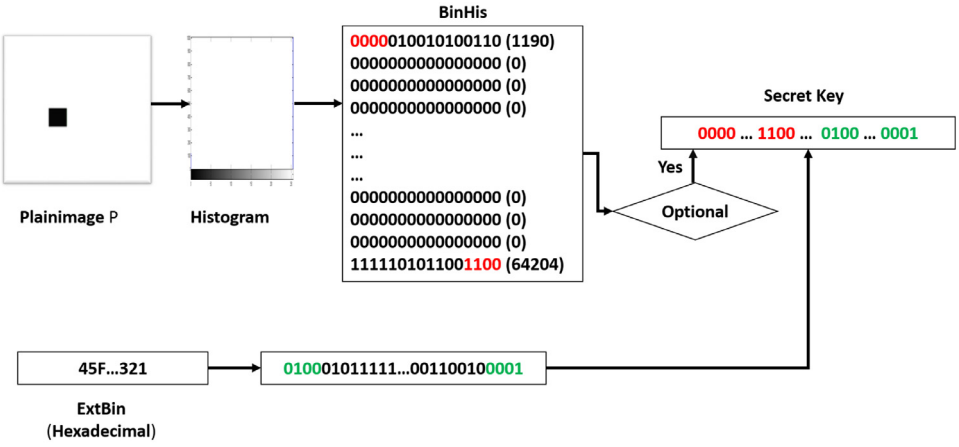

Fig. 11. Example of secret key of binary image.

representation. Both phases of the proposed cipher are described in the following subsections.

3.1. Key generation phase

The proposed cipher is a symmetric-key encryption algorithm, and thus performs both encryption and decryption using the same secret key. Based on current computing capabilities, the key space should be larger than 128 bits to resist brute force attacks. For most chaos-based image encryption schemes, the secret key consists of the initial conditions and/or control parameters of the underlying chaotic map. Each chaotic map has different chaotic ranges, thus the key value needs to be quantized to fit into this range. For TM-DFSM, the initial conditions and control parameters are not included into the secret key, thus negating the need for a quantization function. The proposed cipher's secret key can be divided into two sections, a primary and optional section as shown in Fig. 11. The primary section is an external bitstream (*ExtBin*) with a flexible length whereas the optional section is a bitstream extracted from the plainimage's histogram (*BinHis*). *BinHis* is only used for additional functionality such as image authentication or to achieve higher security. The purpose of *BinHis* and how to generate it will be discussed later on in this section. A user can select any binary stream with a length of at least 128 bits for *ExtBin*. If *BinHis* is included, it will be prepended to *ExtBin*. Either way, the resulting keystream is a one-dimensional vector, $1 \times R$.

This keystream will be used to set up TM-DFSM for the encryption phase in Section 3.2. The initial condition and the control parameters for three tent maps are set to default values of 0.2, 3.99, 2.88, and 3.12 respectively. TM-DFSM map is then iterated R times using the keystream to dictate the state transitions as discussed in Section 2.3. After each iteration the resulting chaotic point is used to perturb the control parameter of the current state. The perturbation function can be generalized for all maps as

$$r_i = (r_i + x_i) \ mod \ (max_{r_i} - min_{r_i}) + min_{r_i} \tag{15}$$

where r_i is control parameter with a range of $[min_{r_i}, max_{r_i}]$. The min/max range should be set based on the chaotic range of the underlying map. For the proposed cipher, we use [1.5, 4] to ensure chaoticity based on the TM-DFSM's bifurcation diagram in Fig. 4.

After the entire keystream has been exhausted, TM-DFSM has been successfully initialized with new control parameters for the encryption phase. In addition, the final chaotic point x_{R+1} will be used as the initial condition for the dynamical controlled TM-DFSM iterations in the encryption phase. A one-bit change to the secret key whether in *BinHis* or *ExtBin* will result in entirely different values for the final chaotic point and control parameters due to the sensitive nature of TM-DFSM.

BinHis can be used as part of the secret key for image authentication and higher security. If *BinHis* is included, the secret key is effectively a one-time key that is dependent on the plainimage. This one-time key is analogous to the one-time pad scenario whereby different keys are used for different plainimages. *BinHis* can be extracted based on the gray level of several image types such as binary, grayscale, and color images. The length of *BinHis* depends on the number of bits required to represent the largest frequency of pixel occurrence. This bit precision is then used to represent the frequency of the remaining gray levels.

For example, let a binary plainimage have a dimension of 100×100, the frequency of "0" and "255" pixels are 7000 and 3000 respectively, and the remaining gray levels have 0 frequencies. Because 7000 is the largest frequency value, the number of bits of its binary representation is taken as the gray level's bit precision. 7000 can be represented as 1101101011000, which requires at least 13 bits. Thus, the length of bitstream (*BinHis*) is $(13 \times 256 = 3328 bits)$ and the *BinHis* for all gray levels will be denoted as $1101101011000(0), 0000000000000(1), \ldots, 0000000000000(254), 0101110111000(255)$. For grayscale or color images, *BinHis* is be extracted using the same methodology. The process of generating *BinHis* is also included in Fig. 11.

3.2. Image encryption phase

The image encryption phase consists of two main processes which are pixel permutation and diffusion. Pixel permutation shuffles pixel positions to remove any correlation between them. Pixel diffusion modifies pixel values based on other pixel values in the plainimage; in other words, pixel values will be *diffused* throughout the resulting cipherimage. The encryption phase has four main steps which are image reshaping, key point generation, column-wise encryption, and row-wise encryption. These steps are detailed below:

1. **Reshaping Image:** The plainimage is first reshaped prior to the permutation and diffusion operations. Let A be a plainimage with a size $(M \times N)$, where M and denotes height (number of rows) and N denotes width (number of columns). The number of pixels in a plainimage is MN. Let M_1 and M_2 denote two new height values that will be used to reshape the plainimage. These new height values are first initialized as $M_1 = S_1$ and $M_2 = S_2$, where $S_1 > M$ and $S_2 < N$. Then, S_1 and S_2 are incremented until the following conditions,

$$MN \ mod \ M_1 = 0,$$
$$MN \ mod \ M_2 = 0,$$

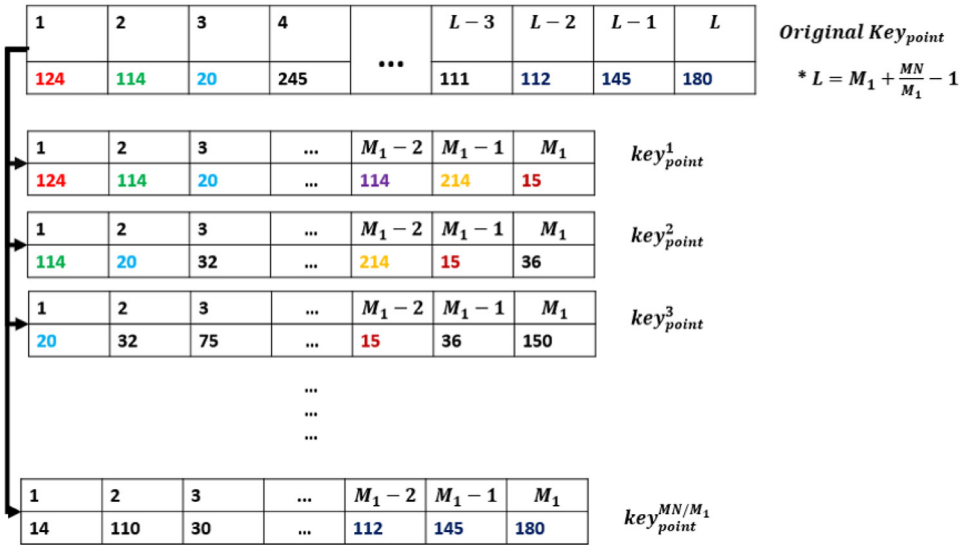

Fig. 12. Extracting the list of key points for the first encryption round.

are fulfilled. In other words, the first values of S_1 and S_2 that fulfill the equality conditions are selected as M_1 and M_2 respectively. We select $S_1 = 1024$ and $S_2 = 20$ as the initial values. These values ensure that the reshaped images will have a long column and row lengths. For example, a plainimage A with a dimension of 512×512 will have $M_1 = 2048$ and $M_2 = 32$ using those initial values. Then, the reshaped image after the first round is 2048×128, which has a long column length. The second reshaping process will result in a dimension of 32×8192, which has a long row length. Maximizing the length of the columns and rows ensures that the column and row-wise operations will involve as many pixels as possible for optimal efficiency and sensitivity.

2. **Generating Key Points:** TM-DFSM is iterated to generate 8-bit values which will be involved in both permutation and diffusion operations. We refer to these 8-bit values as key points. These key points are extracted from TM-DFSM's chaotic points. Based on the $U(0, 52)$ fixed point representation, each chaotic point has 52 fractional bits. 48 of the most significant bits will be divided into six key points. In other words, each chaotic point generated by TM-DFSM will produce six key points each. The number of key points required for encryption can be calculated as

$$L = \max\left(M_1 + \frac{MN}{M_1} - 1, M_2 + \frac{MN}{M_2} - 1\right).$$

Thus, a minimum of $\lceil \frac{L}{6} \rceil$ iterations of TM-DFSM is required to generate sufficient key points. Based on the previous example of a 512×512 image, the new height values are $M_1 = 2048, M_2 = 32$, Thus, $L = \max(2048 + 128 - 1, 32 + 8192 - 1) = 8223$. The number of iterations required is then $\lceil \frac{L}{6} \rceil = 1371$.
These key points are then divided into lists with lengths depending on the encryption round. For the first encryption round, there are $\frac{MN}{M_1}$ lists with a length of M_1 where each list will be used to encrypt one column. For the second encryption round, there are M_2 lists with a length of $\frac{MN}{M_2}$ where each list will be used to encrypt one row. Fig. 12 shows an example of how the lists are generated for the first encryption round, whereby a window size of M_1 is shifted along the list of original key points. The lists for the second encryption round are extracted in the same way but by using a window size of $\frac{MN}{M_2}$.

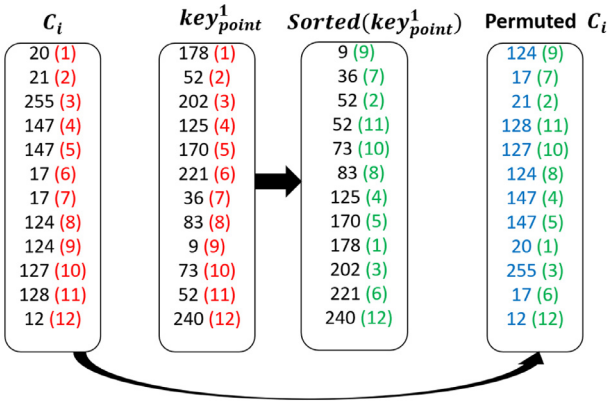

Fig. 13. Example of column permutation.

3. **Round 1, Column-wise Encryption:** For the first round of encryption, the plainimage is reshaped based on M_1, resulting in a new dimension of $M_1 \times \frac{MN}{M_1}$. Then, the reshaped image is encrypted by performing both permutation and diffusion operations in a column-wise manner.
For the permutation process, the key^1_{point} list is first sorted in ascending order. The updated index values of the $Sorted(key^1_{point})$ list will be used as the permutation pattern. This sorting process is only performed once because the same permutation pattern is used for all columns. The pixels in each column, C_i will be permuted based on the permutation pattern as shown in Fig. 13, where the values in the circular brackets, ( ) denote the index values.
Next, the diffusion process begins. We first define the *fingerprint* of a column as a single fixed point value calculated from the column's pixels as well as key points. The fingerprint of the previous column, C_{i-1} is used to encrypt pixels of the current column, C_i (For C_1, $C_{i-1} = C_{\frac{MN}{M}}$). The fingerprint is calculated based on the following steps:
   I. Each pixel value in C_{i-1} and each key point in key^i_{point} are first transformed into their 8-bit binary representations to form two matrices, BM and KM respectively, both with dimensions of $M_1 \times 8$.

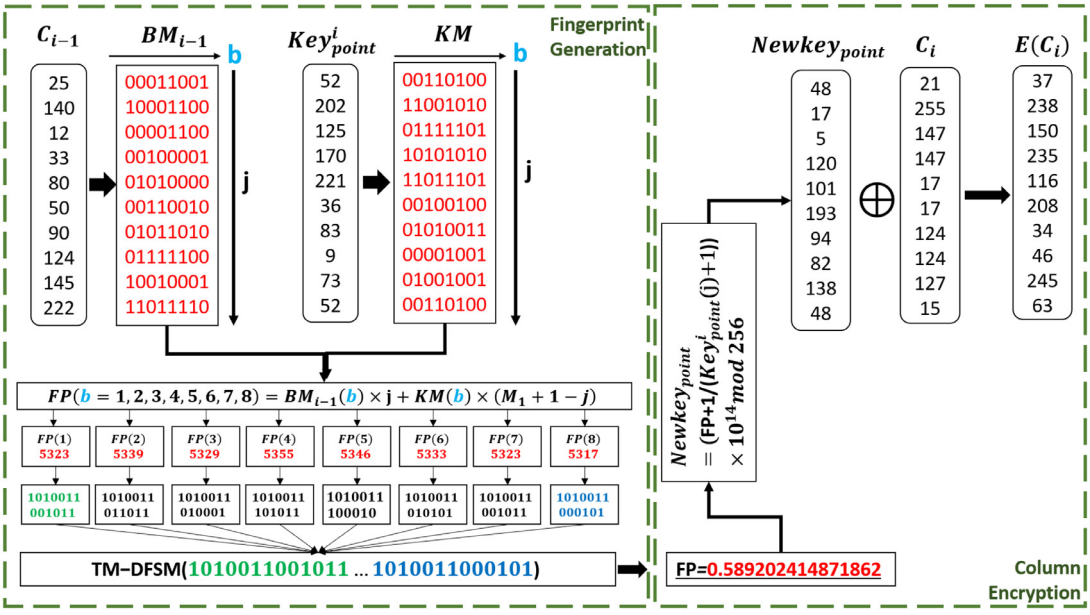

Fig. 14. Example of column diffusion.

II. The bits of each column in *BM* and *KM* are summed up as

$$\mathbf{FP}(i = 1, 2, 3, 4, 5, 6, 7, 8) = \left(\sum_{j=1}^{M_1} BM_{j,i} \times j + KM_{j,i} \times ((M_1 + 1) - j) \right), \quad (16)$$

which results in eight values.

III. These values are then concatenated to form a bitstream, *BinFP* which is used to iterate TM-DFSM. *BinFP* will be used to dictate the state transitions as discussed in Section 2.3.

IV. The final chaotic point, *FP* is the fingerprint of the column. The fingerprint is then used to produce a new set of key points, $Newkey^i_{point}$ by calculating

$$Newkey^i_{point} = \left(FP + \frac{1}{key^i_{point} + 1} \right) \times 10^{14} \ mod\ 256 \quad (17)$$

$Newkey^i_{point}$ is then used to encrypt the current column C_i as

$$C_i = Newkey^i_{point} \bigoplus C_i. \quad (18)$$

The entire process is summarized in Fig. 14. The same process is repeated for each column.

4. **Round 2, Row-wise Encryption:** The second encryption round is similar to the first, albeit with several differences. Firstly, the intermediate cipherimage is reshaped based on M_2, resulting in a new dimension of $\frac{MN}{M_2} \times M_2$. Second, the encryption process is performed in a row-wise manner. Third, the diffusion operation is performed before permutation for each row as shown in Fig. 10. After completing all the encryption steps, the output is reshaped to obtain a cipherimage with the original dimension of $M \times N$.

Algorithm 1 summarizes the steps involved in the proposed image encryption scheme. The decryption process is merely the reverse of the entire encryption process. Fig. 15 shows the effect of the permutation and diffusion operations on a plainimage. Each operation plays an important role in generating a noise-like cipherimage. If two plainimages are encrypted with keys that differ in only one bit, this difference will diffuse throughout the cipherimage. This key sensitivity will be further examined in Section 4.7.

For color images, each color channel can either be encrypted individually then combined as the cipherimage or be encrypted as a single $M \times 3N$ matrix.

Algorithm 1: Image encryption.

Data: Image, A and Secret key, S_1 and S_2 randomly selected values, initial condition, x_0 and control parameters, r_1, r_2 and r_3.
Result: Encrypted Image (C)

```
Compute M1 and M2 by S1 and S2;
Iterate TM − DFSM map and obtain originalkey_point;
C = reshape (A, M1);
I = index (sort (key^1_point));
for i=1 to MN/M1 do
    column permutation (C_i, I);
    column diffusion (C_i, key^i_point);
C = reshape (C, M2);
I = index (sort (key^1_point));
for i=1 to M2 do
    row diffusion (C_i, key^i_point);
    row permutation (C_i, I);
C = reshape (C, M);
```

3.3. Discussion

The proposed encryption algorithm is designed based on TM-DFSM which possesses excellent chaotic performance. Thus, it is able to efficiently generate key points with high randomness. The image encryption scheme then requires only two rounds to fulfill both confusion and diffusion properties. Other significant characteristics of the proposed image encryption scheme are as follows:

- The user has the flexibility to choose any key length and choose to include *BinHis* as a part of the secret key when image authentication or higher security is required.
- A one-bit change anywhere in the plainimage will diffuse throughout the cipherimage.

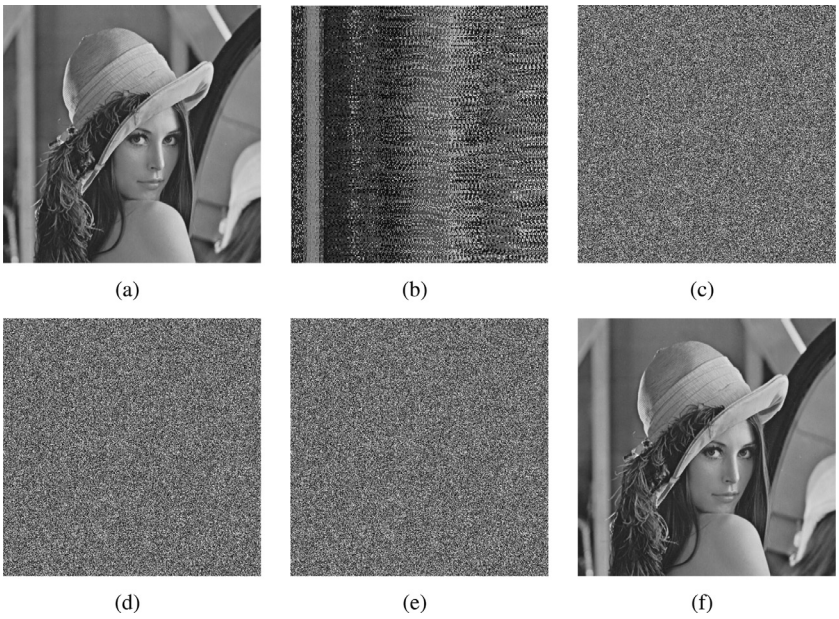

Fig. 15. Four process of the proposed image encryption scheme (a) Plainimage (Lena) size 256×256, (b) column permutation, (c) column diffusion, (d) row diffusion, (e) row permutation, (f) and decrypted image.

- A one-bit change anywhere in the keystream will produce entirely different key points, which will then affect all pixels of the cipherimage.
- Rather than performing permutation and diffusion as separate processes, they are interleaved during the column-wise and row-wise encryption processes, thus having a higher impact to one another.
- The cipher is resistant to well-known security attacks such as statistical, differential, and chosen/known-plaintext attacks as discussed in the following section.

4. Experimental results and performance

All experiments were performed on Matlab R2012b using the Intel Core i5-560M @ 2.67GHz Processor, 8GB RAM, and Windows 7 operating system. Image samples taken from the SC-SIPI 'Miscellaneous' image dataset [55]. Three existing chaos-based ciphers were re-implemented on the same platform and their experimental results were obtained for a fair comparison [11,56,57].

4.1. Histogram analysis

A histogram is an provides a visual depiction of the distribution of the pixels in an image. It shows the frequency of each gray level as a number based on the size of a digital image. When the frequency of gray levels in a cipherimage is approximately equal, it will result in a *flat* histogram. This implies that the pixel values are evenly distributed and no significant biases or patterns can be determined. However, visual inspection of the histogram offers limited insights into the distribution of pixels. Thus, other histogram evaluations are used to investigate the uniform distribution of cipherimage, including maximum deviation, irregular deviation, and deviation from the uniform histogram (DUH).

Maximum deviation is used to measure the difference of the gray level frequency (histogram) between the plainimage and cipherimage. This metric can be used to determine the strength or quality of an image encryption scheme. A high maximum deviation implies a large difference between the histograms of the cipherimage and plainimage, and also a large difference in their pixel values. The maximum deviation can be calculated as

$$M_D = \frac{D_0 + D_{L-1}}{2} + \sum_{i=1}^{L-1} D_i \tag{19}$$

where $D_i = |D_{C_i} - D_{P_i}|$ is the absolute difference of the gray values between the cipherimage and plainimage at index i, and L is the number of gray values with the range of [0,256].

Irregular deviation is a statistical comparison between the difference in histograms and uniform statistical distribution, and it is also an extension of the statistical deviation measure. The irregular deviation can be calculated as

$$I_D = \sum_{i=0}^{L-1} |D_i - \gamma_H| \tag{20}$$

where γ_H is the mean of the histogram values. A low value of I_D indicates the pixels in the cipherimage are uniformly distributed.

DUH quantifies the deviation of a cipherimage's histogram from an ideally uniform histogram. The uniform histogram for an image of any size can be calculated as

$$H_C = \begin{cases} \dfrac{M \times N}{256}, & 0 \le i \le 255 \\ 0, & elsewhere, \end{cases} \tag{21}$$

where M and N are the width and height of an image respectively. The deviation of the cipherimage's histogram from the ideal one can be then calculated as

$$D_H = \frac{\sum_{C_i=0}^{255} |H_{C_i} - H_C|}{M \times N} \tag{22}$$

where H_{C_i} is the gray value of the cipherimage's histogram at index i. A low value of D_H indicates a low deviation from the ideal histogram, which implies a uniform distribution.

The histogram analyses results for our scheme and other existing schemes are listed in Table 3. Results indicate that the proposed scheme outperform the other existing schemes, indicating uniform distribution and resistance against histogram-based attack.

Table 3
Histogram analyses comparison.

Image	Proposed scheme			Ref [11]			Ref [56]			Ref [57]		
	M_D	I_D	DUH	M_D	I_D	DUH	M_D	I_D	DUH	M_D	I_D	DUH
Elaine	42,901	21,865	0.0475	43,012	21,993	0.0504	42,502	21,873	0.0468	42,723	22,090	0.0473
Lena	37,982	19,893	0.0499	37,934	20,037	0.0519	37,435	20,166	0.0479	37,565	20,032	0.0496
Peppers	38,149	23,922	0.050	38,022	23,750	0.0520	38,478	23,966	0.0502	38,332	24,229	0.0492
Cameraman	64,684	32,450	0.0447	64,665	32,596	0.0556	64,138	32,248	0.0469	64,945	32,536	0.0480
Average	**45929**	**24532**	**0.0480**	45,908	24,594	0.0525	45,638	24,563	**0.0480**	45,891	24,722	0.0485

4.2. Correlation analysis

An image cipher should minimize correlations between adjacent pixels. In this regard, permutation algorithms play an important role to eliminate the relationship between pixels through pixel scrambling or shuffling. In this paper, the permutation algorithm is applied twice (column and row-wise) on a reshaped image to produce a highly scrambled image. In the following experiment, the correlation coefficients (CC) of three adjacent pixel directions (vertical, horizontal and diagonal) were calculated from the plainimage and the cipherimage as

$$C_r = \frac{COV(X,Y)}{\sqrt{D(X)D(Y)}}, \tag{23}$$

where $COV(x,y) = E([x - E(x)][y - E(y)])$, $E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i$ and $D(x) = \frac{1}{N}\sum_{i=1}^{N}[x_i - E(x)]^2$. x and y are the values of adjacent pixels in the plainimage or cipherimage.

CC values close to 0 implies minimal correlation between adjacent pixels in the cipher image whereas CC values close to 1 or -1 indicates strong pixel correlation. Table 4 shows the results of CC for four images encrypted by the proposed scheme where 4096 pairs of adjacent pixels were randomly selected in three directions (diagonal, vertical, horizontal). This was performed for the plainimage and the corresponding pixels in the cipherimage. The CC distributions are as shown in Fig 16. Furthermore, Table 5 provides the comparison of average CC results of the proposed scheme and other schemes for multiple images. Based on these experiments, one can observe that the proposed scheme is superior in terms of eliminating the correlation between adjacent pixels as compared to the other schemes.

Table 4
Correlation coefficient for multiple images.

Image	Plainimage			Cipherimage		
	D	V	H	D	V	H
5.1.09	0.8896	0.8354	0.9372	−0.000026	−0.00031	−0.021
5.1.10	0.8618	0.8999	0.8768	−0.00031	−0.00046	−0.016
5.1.11	0.7786	0.9098	0.9183	−0.000029	−0.00319	0.0242
5.1.12	0.6393	0.5566	0.8055	0.00092	−0.0000056	0.014

Table 5
Correlation coefficient of the encrypted four images with different algorithms.

Image	Cipherimage		
	D	V	H
Proposed scheme	0.0003	0.0009	0.019
Ref. [11]	0.0011	0.0012	0.016
Ref. [56]	0.0004	0.0011	0.0125
Ref. [57]	0.0015	0.0013	0.012

4.3. Chosen/known plaintext attacks

In many cryptanalytic attacks, adversaries attempt to identify the relationship between the plainimage and cipherimage, find patterns that reduces search space for the secret key or even equivalent encryption keys [58–60]. Most attacks involve constructing a set of special images or choosing plainimages consisting of all white or black pixels. The corresponding cipherimage will then depict visible patterns that can be used deduce the key. Therefore, to resist such attacks, an image encryption scheme should remove any relation between the cipherimage and plainimage.

In the proposed scheme, no recognizable patterns are produced even when a specific plainimage, such as an black or white image, is chosen. This is due to the encryption process being key-dependent in addition to pixel-dependent. Encrypting in a column-wise and row-wise manner also contributes towards resisting these attacks, especially when the encrypted columns/rows are used to encrypt neighboring columns/rows. This property is depicted in Fig. 17, where the black and white images are encrypted to produce noise-like cipherimages. One can see that the proposed scheme is highly resistant to chosen or known plaintext attacks.

4.4. Contrast analysis

The contrast analysis measures local feature variations between pixels over different directions in a digital image. It studies the fixed patterns that have different directions and sizes in an image, and can be mathematically calculated as

$$C = \sum_{i,j} |i-j|^2 G(i,j), \tag{24}$$

where G is co-occurrence matrix of the gray level index, i and j are 8-bit gray level images.

G can be statistically determined by counting the number of fixed patterns in an image matrix, and is based on two criteria: distance between pixels in one pattern, and the four directions that include horizontal(0), vertical(90), main diagonal (135), and secondary diagonal (45). In this experiment, we use both criteria. Contrast values for various image sizes were calculated. For comparison, we calculate the theoretical value of contrast features for a digital image as $C_{theoretical} = \sum_{i=1}^{256}\sum_{j=1}^{256}\frac{|i-j|^2}{(256)^2} = 10922.50$, whereby all patterns have the same probability of occurring in an image. Table 6 lists the results of the experiment for the proposed scheme and other encryption schemes. The proposed cipher outperforms the rest, achieving near-ideal values for the majority of the samples.

4.5. Local shannon entropy

The pixels in an encrypted image should be uniformly distributed in entire image and also in any non-overlapping portions of the image. Each randomly selected portion of the image should have the same uniform distribution as the entire image. Based on this notion, the local Shannon entropy (LSE) can be used to evaluate the distribution and randomness of a digital image. For an

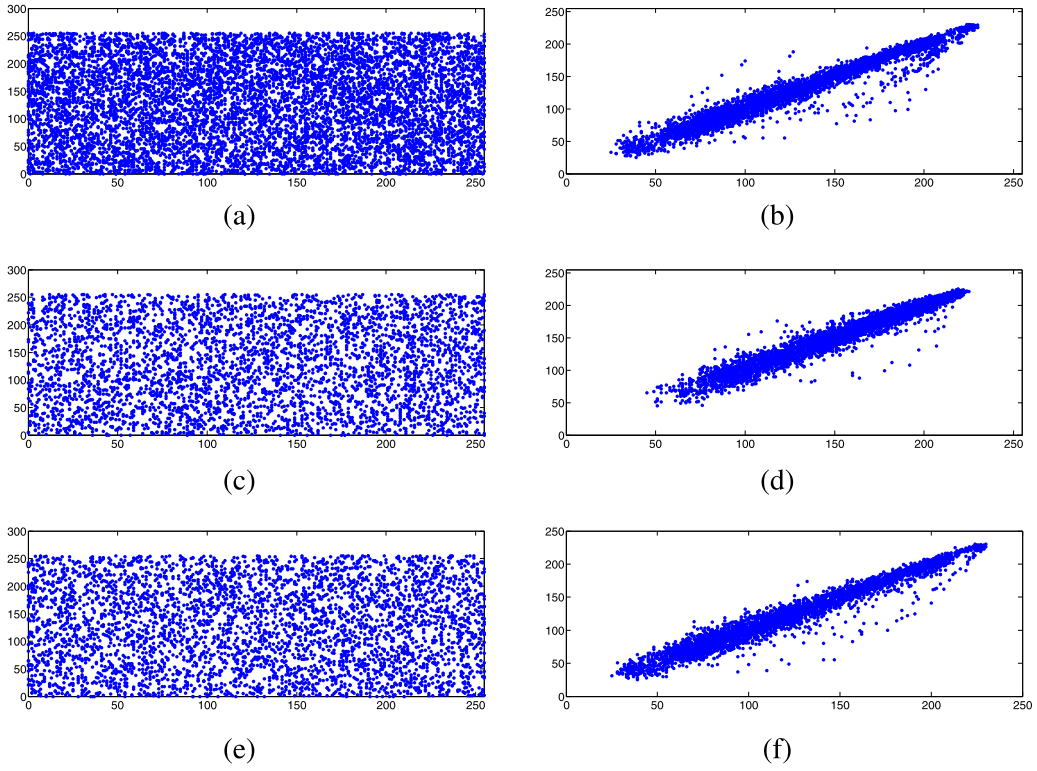

Fig. 16. Correlation coefficient of Lena image (a) Horizontal cipherimage, (b) horizontal plainimage, (c) vertical cipherimage, (d) vertical plainimage, (e) diagonal plainimage, (f) and diagonal cipherimage.

Table 6
Contrast and LSE score comparison.

Images	Contrast scores				LSE scores			
	Proposed	Ref [11]	Ref [56]	Ref [57]	Proposed	Ref [11]	[56]	Ref [57]
5.1.09	10888.76	10883.87	10930.66	10842.83	7.903369	7.903395	**7.902831**	**7.9024**
5.1.10	10940.76	11012.40	10896.93	10894.26	7.90352	7.903271	7.903056	7.903504
5.1.11	10973.62	10992.52	11011.18	10867.39	**7.902291**	7.903043	**7.902361**	7.903917
5.1.12	10956.93	10920.51	10895.49	10990.48	**7.902721**	7.903468	7.903068	7.903088
5.1.13	10959.42	10881.36	10893.51	10980.64	**7.90262**	**7.902728**	**7.902555**	**7.902846**
5.1.14	10833.91	10902.43	10836.73	10979.64	**7.902837**	**7.902795**	7.903077	7.90324
5.2.08	10939.38	10919.69	10928.92	10922.67	**7.902793**	**7.902831**	7.903439	7.90307
5.2.09	10885.09	10943.73	10894.08	10945.07	**7.902972**	**7.903028**	**7.902495**	**7.902931**
5.2.10	10945.3	10925.33	10851.13	10879.36	**7.902464**	7.903511	**7.902959**	**7.903013**
7.1.01	10926.45	10929.88	11037.71	10988.40	**7.902934**	7.903362	7.903452	7.903434
7.1.02	10941.14	10938.70	10731.55	10976.89	**7.902843**	**7.902706**	**7.902421**	7.903118
7.1.03	10917.89	10938.56	10977.45	10917.18	7.903339	7.903252	7.903262	**7.90302**
7.1.04	10946.11	10937.08	10983.79	10877.25	**7.902649**	7.903313	**7.902897**	**7.903012**
7.1.05	10935.12	10929.65	10897.40	10954.01	**7.902493**	**7.903103**	**7.902745**	**7.902755**
7.1.06	10925.71	10928.22	10930.30	10958.75	7.903261	**7.902625**	**7.902853**	**7.902915**
7.1.07	10917.11	10923.96	10814.11	10919.25	**7.902714**	**7.902435**	7.903132	**7.902703**
7.1.08	10910.15	10895.11	10979.45	10899.94	**7.902563**	**7.902675**	7.903876	**7.9026**
7.1.09	10907.02	10902.03	10811.15	10895.26	7.903185	**7.902813**	**7.90242**	**7.902661**
7.1.10	10886.47	10902.46	10901.79	10880.29	**7.902805**	**7.902668**	7.903298	**7.902658**
boat.512	10928.86	10920.95	10897.99	10883.21	7.90307	**7.902632**	7.903556	7.90341
elaine.512	10921.81	10879.22	11017.64	10866.88	**7.902929**	**7.902486**	**7.902529**	7.903318
gray21.512	10925.15	10918.86	10877.13	10813.04	7.903238	**7.902543**	**7.902955**	**7.902253**
numbers.512	10919.35	10926.62	10850.12	10884.88	**7.902697**	**7.902885**	7.903214	7.903088
ruler.512	10903.73	10904.14	10883.35	11134.05	**7.902755**	**7.902805**	7.903838	**7.902817**
5.3.01	10907.56	10928.71	10922.46	10943.51	7.903661	7.903106	7.903486	**7.902546**
5.3.02	10928.38	10918.22	10919.86	10979.02	**7.902545**	7.903263	**7.902842**	**7.902908**
7.2.01	10928.2	10929.86	10945.98	10888.05	**7.902896**	**7.902848**	**7.902853**	7.903643
testpat.1k	10928.75	10912.45	10935.70	10906.92	**7.902715**	7.902336	7.903195	7.903943
{Mean}	10922.43	10923.09	10909.06	10923.90	7.902889	7.902926	7.903023	7.903029
{Std}	19.30	18.71	51.58	47.40	0.0004	0.000328	0.000407	0.000408
{Pass/All}	-	-	-		20/28	14/28	14/28	16/20

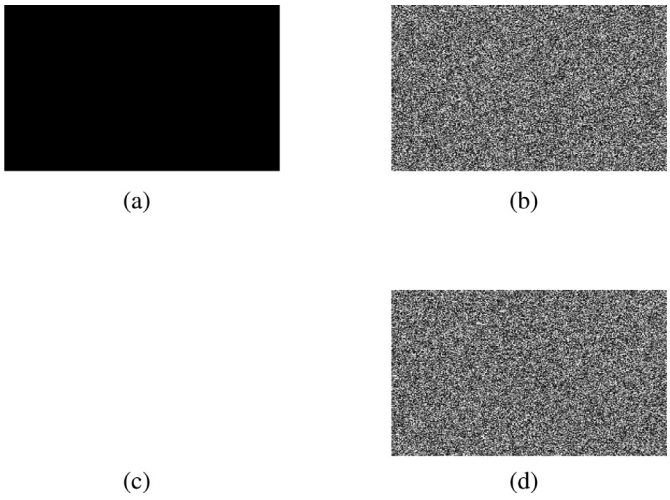

Fig. 17. Encryption results of the black and white images (a) Black plainimage, (b) black cipherimage, (c) white plainimage, (d) white cipherimage.

entire image I, Shannon entropy can be calculated as

$$H(I) = -\sum_{j=0}^{L-1} p(I_j) log_2 p(I_j), \tag{25}$$

where $p(I_j)$ is the discrete probability density function (PDF), I_j is the pixel value from 0 to 255, and L is grey level ($L = 256$ for an 8-bit grey image). The LSE uses the Shannon entropy measure on randomly selected non-overlapping image blocks with a uniform number of pixels. Thus, LSE can be calculated by the average value of Shannon entropy values of all randomly selected blocks as

$$H_{k,T_B}(I) = \sum_{i=1}^{k} \frac{H(I_{B_i})}{k}, \tag{26}$$

where $H(I_{B_i})$ is the Shannon entropy of non-overlapping image blocks B_i. k and T_B denote the number of blocks and pixels in each block. To successfully pass in LSE test, the LSE degree for a cipherimage should be within the critical interval of (7.901901305, 7.903037329). These critical bounds are calculated using the significant value $\alpha = 0.05$ and parameters $(k, TB) = (30, 1936)$. The theoretical value of LSE can be calculated as 7.902469317.

Table 6 compares the LSE degrees of cipherimages that are encrypted by the proposed scheme to the LSE values of other schemes. For majority of the samples, the cipherimages produced by the proposed cipher fall within the critical interval (more than 70%) in comparison to the other schemes which can achieve a maximum of approximately 40%. Thus, the proposed cipher demonstrates high randomness for the non-overlapping blocks for plainimages of different types and sizes.

4.6. Differential attack analysis

The differential attack observes how an XOR difference between a pair of inputs lead to an XOR difference between a pair of outputs. This cryptanalytic attack can aid adversaries in recovering secret key information or plaintext pixels from a cipherimage. Therefore, an image encryption scheme should be highly sensitive to small changes in the plaintext which must be diffused throughout the cipherimage. The number of pixel change rate (NPCR) and the unified average change intensity (UACI) are widely used to test the resistance of encryption algorithms against differential attacks.

First, two plainimages with a one-bit difference are encrypted, and the two cipherimages C_1 and C_2 are obtained. NPCR and UACI can be estimated as

$$NPCR(C_1, C_2) = \frac{\sum_{i,j}^{M,N} D(i,j)}{MN} \times 100\% \tag{27}$$

and

$$UACI(C_1, C_2) = \frac{1}{MN} \sum_{i,j}^{M,N} \frac{|C_1(i,j) - C_2(i,j)|}{L} \times 100\% \tag{28}$$

respectively, where MN is the total number of pixels in a plainimage, $L = 255$ is the maximum gray level value for an 8-bit pixel, and

$$D(i,j) = \begin{cases} 1, & \text{if } C_1(i,j) \neq C_2(i,j) \\ 0, & \text{if } C_1(i,j) = C_2(i,j), \end{cases} \tag{29}$$

The evaluation of whether an cipher can pass the NPCR and UACI tests is based on critical values that are used as thresholds [61]. NPCR has a critical value (N^*_α) for different image sizes, (N^*_α) defined as

$$N^*_\alpha = \frac{L - \Phi^{-1}(\alpha)\sqrt{L/MN}}{L+1} \tag{30}$$

where α is the significance level, Φ^{-1} is inverse cumulative density function (CDF) of normal distribution $N(0, 1)$. An image encryption scheme must have a NPCR value larger than N^*_α to pass NPCR test. In the UACI test, two strict critical intervals $U^{*-}_\alpha, U^{*+}_\alpha$ can be calculated for different image sizes as

$$\begin{cases} U^{*-}_\alpha = \mu_u - \Phi^{-1}(\alpha/2) \times \sigma_u; \\ U^{*+}_\alpha = \mu_u + \Phi^{-1}(\alpha/2) \times \sigma_u, \end{cases} \tag{31}$$

where

$$\mu_u = \frac{L+2}{3L+3} \tag{32}$$

and

$$\sigma_u^2 = \frac{(L+2)(L^2+2L+3)}{18(L+1)^2 \times L \times MN} \tag{33}$$

We select 28 test digital images with different sizes from the USC-SIPI 'Miscellaneous' image dataset. There are 6, 18 and 4 no. of images with sizes 256×256, 512×512 and 1024×1024, respectively. Additionally, we set the significance level as $\alpha = 0.05$ for all images to compute critical values of NPCR and UACI. Each image encrypted twice times before and after changing one-bit while using the same secret key (*ExtBin* only), and the NPCR and UACI results of our proposed scheme and other schemes tabulated in Table 7. The proposed scheme can pass the NPCR and UACI tests for majority of the images, implying its resistance against differential cryptanalysis.

4.7. Key security

For optimal security, an encryption algorithm should have a large key space to resist exhaustive search and also be sensitive to changes in the key. The mandatory portion of the secret key for the proposed cipher is *ExtBin*, which is a bitstream that must be at least 128 bits to resist brute force attacks. If image authentication is required, *BinHis* can be extracted from the plainimage's histogram and included as part of the secret key. In addition to increasing the size of the key space, it also provides the cipher with some properties of a one-time pad cipher for higher security and plaintext sensitivity.

Another interesting property of the proposed cipher is its capability to function as a tweakable cipher. In conventional cryptography, a tweakable block cipher receives an additional input called a *tweak* that has the same function as an initialization vector for block cipher modes of operation [62]. It brings the advantage of

Table 7
NPCR and UACI values comparison.

Image	NPCR				UACI			
	Proposed	Ref. [11]	Ref. [56]	Ref. [57]	Proposed	Ref. [11]	Ref. [56]	Ref. [57]
256 × 256	$N_\alpha^* \geq 99.5693\%$				$U_\alpha^{*-}, U_\alpha^{*+} = (33.2824\%, 33.6447\%)$			
5.1.09	99.603	99.599	89.415	49.309	33.552	33.290	0.698	0.387
5.1.10	99.636	99.627	16.201	73.460	33.453	33.273	0.064	0.288
5.1.11	99.942	99.635	52.489	42.230	33.586	33.323	0.412	0.331
5.1.12	99.792	99.617	28.144	98.456	33.453	33.535	0.440	1.216
5.1.13	99.792	99.624	8.677	60.345	33.520	33.459	0.034	0.237
5.1.14	99.621	99.632	48.378	63.626	33.440	33.396	12.094	15.991
512 × 512	$N_\alpha^* \geq 99.5893\%$				$U_\alpha^{*-}, U_\alpha^{*+} = (33.3730\%, 33.5541\%)$			
5.2.08	99.960	99.611	30.740	28.340	33.692	33.452	0.121	0.111
5.2.09	99.876	99.632	23.050	48.458	33.548	33.444	0.090	0.190
5.2.10	99.654	99.602	99.608	99.68	33.454	33.527	33.44	33.386
7.1.01	99.957	99.613	11.192	65.980	33.648	33.525	0.044	0.259
7.1.02	99.918	99.619	61.226	51.164	33.465	33.531	0.240	0.201
7.1.03	99.849	99.610	10.535	99.223	33.273	33.401	0.041	3.119
7.1.04	99.991	99.618	22.989	99.223	33.202	33.520	0.180	0.583
7.1.05	99.942	99.586	94.707	99.223	33.830	33.505	2.971	12.450
7.1.06	99.670	99.600	67.933	99.59	33.627	33.437	0.535	33.552
7.1.07	99.983	99.608	47.197	53.055	33.609	33.511	0.185	0.208
7.1.08	99.818	99.605	5.817	99.223	33.375	33.418	0.023	1.553
7.1.09	99.874	99.617	58.374	53.867	33.530	33.405	0.229	0.211
7.1.10	99.697	99.606	99.595	51.158	33.438	33.536	33.39	0.201
boat.512	99.715	99.615	4.513	99.601	33.374	33.476	0.018	33.52
elaine.512	99.746	99.625	60.889	99.223	33.379	33.450	0.955	1.400
gray21.512	99.643	99.609	35.443	49.407	33.507	33.483	0.279	0.387
numbers.512	99.653	99.615	51.163	99.223	33.388	33.392	0.201	0.778
ruler.512	99.637	99.623	8.248	99.223	33.415	33.353	0.032	0.780
1024 × 1024	$N_\alpha^* \geq 99.5994\%$				$U_\alpha^{*-}, U_\alpha^{*+} = (33.4183\%, 33.5088\%)$			
5.3.01	99.950	99.613	12.852	99.611	33.508	33.479	0.101	33.421
5.3.02	99.982	99.613	99.601	31.821	33.514	33.482	33.419	0.250
7.2.01	99.980	99.606	60.130	50.005	33.487	33.474	0.236	0.196
testpat.1k	99.887	99.614	37.898	54.243	33.453	33.467	0.149	0.213
Pass/All	28/28	27/28	3/28	4/28	20/28	26/28	3/28	4/28

using a block cipher mode of operation down to the primitive block cipher level. The proposed cipher can receive a *tweak* of any length to generate a different set of key points even with the same secret key. This *tweak* can be shared publicly, and can either be prepended or appended to the existing secret key prior to encryption. The use of a tweak will reduce the need to change the secret key, which is a costly process.

As mentioned in Section 2.4.2, TM-DFSM map has high sensitivity to even a one-bit change to the resulting keystream (*ExtBin*, *BinHis*, and/or *tweak*) which leads to entirely different chaotic trajectories. The key points are always extracted after iterating TM-DFSM using the keystream. Thus, these key points would be different if one keystream bit has been changed. Next, we focus on the sensitivity of the key points themselves. The proposed scheme does not only depend on TM-DFSM's sensitivity, but the encryption algorithm itself is sensitive to tiny changes in the key points. We set the *ExtBin*s to

$$k1 = FCA32DC55C71CFCA32DC55C71CFCA32DC55C71C$$

and we generate the set of key points, $K1$. Next, we randomly toggle only one-bit of $K1$ to be $K2$. Then the plainimage, Barbara is encrypted by $K1$ and $K2$ to obtain $C1$ and $C2$ respectively as shown in Fig. 18. The difference between two cipherimages is plotted in Fig. 18(d), whereby the two cipherimages $C1$ and $C2$ are completely different. Moreover, when decrypting $C1$ using $K2$ and $C2$ using $K1$, the original image cannot be recovered because due to the incorrect key points. Consequently, our scheme has quite high sensitivity to the secret key and key points in both encryption and decryption.

Table 8
Comparison of speed analysis of different image sizes (Second).

Scheme	256 × 256	512 × 512	1024 × 1024
Our scheme	0.1272	0.5156	2.1321
Ref. [11]	6.3849	25.3077	104.76
Ref. [56]	0.1635	0.6132	2.4265
Ref. [57]	2.2149	8.6154	34.2077

4.8. Speed analysis

An image encryption algorithm should be different image sizes and achieving confidentiality. Our scheme has some advantages which includes negating the need to generate chaotic points based on the number of plainimage pixels, requiring only two encryption rounds, limiting the sorting process to only once per round, and enhancing the tent map without using an external source. Hence, our scheme has a low computational complexity with robust security. Table 8 compares the encryption speed for our scheme against other schemes using several sizes of plainimages. We implement our scheme and other schemes using Matlab R2012b on a Intel Core i5-560M @ 2.67 GHz Processor, 8 GB RAM, and Windows 7 operating system. One can see that our scheme has high encryption speed compared to other schemes.

5. Image authentication

Data authentication is the process of the validating or authenticating data after the decryption process. Most authentication techniques are used hash functions in order to prove that data is the

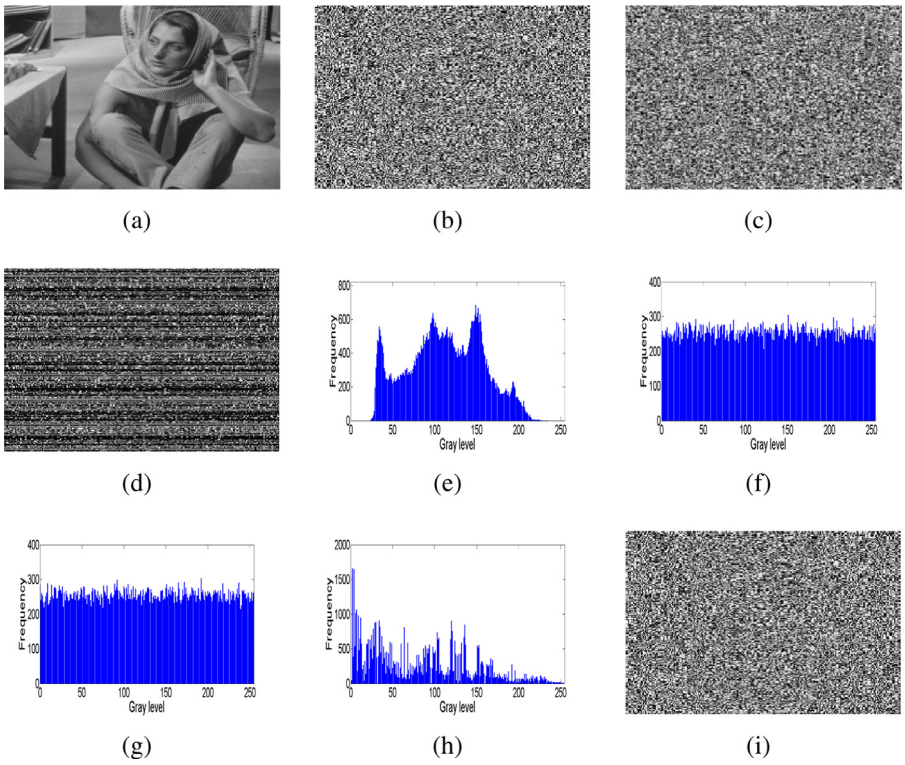

Fig. 18. Key sensitivity analysis in image encryption scheme (a) plainimage (Barbara) size 256×256, (b) cipherimage C1 by K1, (c) cipherimage C2 by K2, (d) the difference between $C3 = |C1 - C2|$, (e) histogram plainimage, (f) histogram C1, (g) histogram C2, (h) histogram C3 (i) and decrypted C1 by K2.

same on the sender and receiver sides. A recipient will compare the hash value obtained by the sender, with the hash value of the decrypted data. If the two hash values differ, it is assumed that the data has been tampered with during transmission.

In our scheme, when the sender *only* uses *BinHis* to generate key points, *BinHis* is considered a one-time key that is sent to the recipient. A user in the receiver side can decrypt a cipherimage by using *BinHis* and extract histogram of the recovered image for comparison. If the histogram matches *BinHis*, the image has been authenticated. Moreover, any changes to the cipherimage will prevent the plainimage from being recovered successfully, which also implies that data tampering has occurred. Consequently, our proposed scheme has the flexibility to provide both secure image encryption as well as image authentication with the same encryption architecture.

6. Conclusion

In this paper, we introduced a new chaotification method to enhance digital tent maps. The tent map and DFSM are combined to create a new TM-DFSM map. The chaotic complexity of TM-DFSM was studied via bifurcation diagram, Lyapunov exponent, fuzzy correlation dimension, and symplectic entropy. Notably, we have also enhanced the correlation dimension estimation by using a fuzzy membership function. Comparison and analysis results show that TM-DFSM has excellent chaotic complexity as compared to other recently proposed 1D chaotic maps. To investigate the application of TM-DFSM in cryptography, a new image encryption scheme was proposed. It has a large, flexible key space that can be adjusted by the user. Both confusion and diffusion operations are performed in one-pass, and the proposed cipher can fully secure a plainimage in just two rounds. Experimental results demonstrate the high sensitivity of the proposed scheme, whereby an entirely different cipherimage can be produced with just a one-bit difference in the plainimage or secret key. Further security analyses also verified the strength of the proposed algorithm which outperforms some recently proposed image ciphers. For future work, we will enhance our scheme to be more robust against attacks such as the cropping attack aims to prevent successful decryption by modifying images in transit.

Conflict of interest

None.

Acknowledgment

This work has been supported by Universiti Sains Malaysia under Grant No. 304/PKOMP/6316280.

References

[1] J.S. Teh, K. Tan, M. Alawida, A chaos-based keyed hash function based on fixed point representation, Cluster Comput. (2018) 1–12.
[2] Z. Hua, F. Jin, B. Xu, H. Huang, 2D logistic-sine-coupling map for image encryption, Signal Process. 149 (2018) 148–161.
[3] Y.Z. Zhongyun Hua, Shuang Yi, Medical image encryption using high-speed scrambling and pixel adaptive diffusion, Signal Process. 149 (2018) 148–161.
[4] Y. Wang, K.-W. Wong, X. Liao, G. Chen, A new chaos-based fast image encryption algorithm., Appl. Soft Comput. 11 (1) (2011) 514–522.
[5] J. xin Chen, Z. liang Zhu, C. Fu, H. Yu, L. bo Zhang, A fast chaos-based image encryption scheme with a dynamic state variables selection mechanism., Commun. Nonlinear Sci. Numer. Simul. 20 (3) (2015) 846–860.
[6] Y. Zhang, Test and verification of AES used for image encryption, 3D Res. 9 (1) (2018), doi:10.1007/s13319-017-0154-7.
[7] X. Chai, Z. Gan, Y. Chen, Y. Zhang, A visually secure image encryption scheme based on compressive sensing, Signal Process. 134 (2017) 35–51, doi:10.1016/j.sigpro.2016.11.016.
[8] N. Zhou, Y. Hu, L. Gong, G. Li, Quantum image encryption scheme with iterative generalized arnold transforms and quantum image cycle shift operations, Quantum Inf. Process. 16 (6) (2017), doi:10.1007/s11128-017-1612-0.
[9] J. Chen, Z. liang Zhu, L. bo Zhang, Y. Zhang, B. qiang Yang, Exploiting self-adaptive permutation–diffusion and DNA random encoding for secure and efficient image encryption, Signal Process. 142 (2018) 340–353, doi:10.1016/j.sigpro.2017.07.034.

[10] M. Alawida, A. Samsudin, J.S. Teh, R. S.Alkhawaldeh, A new hybrid digital chaotic system with applications in image encryption, Signal Process. 160 (July 2019) 45–58.

[11] Z. Hua, Y. Zhou, Image encryption using 2D logistic-adjusted-sine map, Inf. Sci. 339 (2016) 237–253, doi:10.1016/j.ins.2016.01.017.

[12] C. Li, G. Luo, K.Q.C. Li, An image encryption scheme based on chaotic tent map, Nonlinear Dyn. 84 (1) (2017) 127–133.

[13] Z. Hua, B. Xu, F. Jin, H. Huang, Image encryption using josephus problem and filtering diffusion, IEEE Access 7 (2019) 8660–8674, doi:10.1109/access.2018.2890116.

[14] Z. Hua, Y. Zhou, H. Huang, Cosine-transform-based chaotic system for image encryption, Inf. Sci. 480 (2019) 403–419, doi:10.1016/j.ins.2018.12.048.

[15] X. Wu, B. Zhu, Y. Hu, Y. Ran, A novel colour image encryption scheme using rectangular transform-enhanced chaotic tent maps, IEEE Access (2017). 1–1. doi: 10.1109/access.2017.2692043.

[16] C. Zhu, K. Sun, Cryptanalyzing and improving a novel color image encryption algorithm using RT-enhanced chaotic tent maps, IEEE Access 6 (2018) 18759–18770, doi:10.1109/access.2018.2817600.

[17] C. Pak, L. Huang, A new color image encryption using combination of the 1d chaotic map, Signal Process. 138 (2017) 129–137.

[18] H. Wang, D. Xiao, X. Chen, H. Huang, Cryptanalysis and enhancements of image encryption using combination of the 1d chaotic map, Signal Process. 144 (2018) 444–452, doi:10.1016/j.sigpro.2017.11.005.

[19] G. Ye, X. Huang, An image encryption algorithm based on autoblocking and electrocardiography, IEEE MultiMedia 23 (2) (2016) 64–71, doi:10.1109/mmul.2015.72.

[20] C. Li, D. Lin, J. Lü, F. Hao, Cryptanalyzing an image encryption algorithm based on autoblocking and electrocardiography, IEEE Multimedia (2019), doi:10.1109/MMUL.2018.2873472.

[21] D. Wheeler, R. Matthews, Supercomputer investigations of a chaotic encryption algorithm, Cryptologia 15 (1991) 140–151.

[22] Y. Zhou, Z. Hua, C. Pun, C.L.P. Chen, Cascade chaotic system with applications, IEEE Trans. Cybern. 45 (9) (2015) 2001–2012.

[23] R. Lan, J. He, S. Wang, T. Gu, X. Luo, Integrated chaotic systems for image encryption, Signal Process. 147 (2018) 133–145.

[24] M. Alawida, A. Samsudin, J.S. Teh, Enhancing unimodal digital chaotic maps through hybridisation, Nonlinear Dyn. 96 (1) (2019) 601–613, doi:10.1007/s11071-019-04809-w.

[25] Z. Hua, B. Zhou, Y. Zhou, Sine chaotification model for enhancing chaos and its hardware implementation, IEEE Trans. Ind. Electron. 66 (2) (2019) 1273–1284, doi:10.1109/tie.2018.2833049.

[26] W. Yue, Z. Yicong, L. Bao, Discrete wheel-switching chaotic system and applications, IEEE Trans. Circuits Syst. I Regul. Pap. 12 (2014) 3469–3477.

[27] Y. Zhou, L. Bao, C.L.P. Chen, Image encryption using a new parametric switching chaotic system, Signal Process. 93 (11) (2013) 3039–3052.

[28] Y. Zhou, L. Bao, C.L.P. Chen, A new 1d chaotic system for image encryption, Signal Process. 97 (11) (2014) 172–182.

[29] R. Parvaz, M. Zarebnia, A combination chaotic system and application in color image encryption, Opt. Laser Technol. 101 (2018) 30–41.

[30] L. Liu, B. Liu, H. Hu, S. Miao, Reducing the dynamical degradation by bi-coupling digital chaotic maps, Int. J. Bifurcat. Chaos 28 (5) (2018). 1850059 (14p)

[31] Z. Hua, B. Zhou, Y. Zhou, Sine-transform-based chaotic system with FPGA implementation, IEEE Trans. Ind. Electron. 65 (3) (2018) 2557–2566, doi:10.1109/tie.2017.2736515.

[32] H. Hu, Y. Deng, L. Liu, Counteracting the dynamical degradation of digital chaos via hybrid control, Commun. Nonlinear Sci. Numer. Simul. 19 (6) (2014) 1970–1984, doi:10.1016/j.cnsns.2013.10.031.

[33] J. Zheng, H. Hu, X. Xia, Applications of symbolic dynamics in counteracting the dynamical degradation of digital chaos, Nonlinear Dyn. 94 (2) (2018) 1535–1546, doi:10.1007/s11071-018-4440-6.

[34] C. Fan, Q. Ding, Effects of limited computational precision on the discrete chaotic sequences and the design of related solutions, Complexity 2019 (2019) 1–10, doi:10.1155/2019/3510985.

[35] X.J. Tong, Z. Wang, M. Zhang, Y. Liu, H. Xu, J. Ma, An image encryption algorithm based on the perturbed high-dimensional chaotic map., Nonlinear Dyn. 80 (3) (2015) 1493–1508.

[36] Y. Liu, Y. Luo, S. Song, L. Cao, J. Liu, J. Harkin, Counteracting dynamical degradation of digital chaotic chebyshev map via perturbation, Int. J. Bifurcat. Chaos 27 (03) (2017).

[37] L. Liu, J. Lin, S. Miao, B. Liu, A double perturbation method for reducing dynamical degradation of the digital baker map, Int. J. Bifurcat. Chaos 27 (07) (2017) 1750103.

[38] Z. Hua, Y. Zhou, Dynamic parameter-control chaotic system, IEEE Trans. Cybern. 46 (12) (2016) 3330–3341, doi:10.1109/tcyb.2015.2504180.

[39] Z. Hua, Y. Zhou, One-dimensional nonlinear model for producing chaos, IEEE Trans. Circuit. Syst. I 65 (1) (2018) 235–246, doi:10.1109/tcsi.2017.2717943.

[40] Y. Deng, H. Hu, N. Xiong, W. Xiong, L. Liu, A general hybrid model for chaos robust synchronization and degradation reduction, Inf. Sci. 305 (2015) 146–164, doi:10.1016/j.ins.2015.01.028.

[41] L.-C. Cao, Y.-L. Luo, S.-H. Qiu, J.-X. Liu, A perturbation method to the tent map based on Lyapunov exponent and its application, Chin. Phys. B 24 (10) (2015) 100501, doi:10.1088/1674-1056/24/10/100501.

[42] Y. Deng, H. Hu, L. Liu, Feedback control of digital chaotic systems with application to pseudorandom number generator, Int. J. Modern Phys. C 26 (2) (2015). 1550022 (20 p)

[43] H. Hu, Y. Xu, Z. Zhu, Method of improving the properties of digital chaotic system, Chaos Soliton. Fractals 38 (2008) 439–446.

[44] H. Gruber, M. Holzer, From finite automata to regular expressions and back â a summary on descriptional complexity, Int. J. Found. Comput. Sci. 26 (08) (2015) 1009–1040.

[45] M. Alawida, A. Samsudin, W.H. Alshoura, Enhancing One-Dimensional Chaotic Map Based on Bitstream Dividing Model, in: Proceedings of the 2019 8th International Conference on Software and Computer Applications - ICSCA 19, ACM Press, 2019, doi:10.1145/3316615.3316657.

[46] C. Skokos, The Lyapunov characteristic exponents and their computation, Lect. Notes Phys. 790 (2010) 63–135.

[47] P. Grassbergert, T. Procaccia, Characterization of strange attractors, Phys. Rev. Lett 50(5) (1983) 346–349.

[48] J. Lin, Divergence measures based on the Shannon entropy, IEEE Trans. Inf. Theory 37 (1) (1991) 145–151.

[49] P. SM, Approximate entropy as a measure of system complexity, Proc. Natl. Acad. Sci. USA 88 (1991) 2297–2301.

[50] J. Richman, J. Moorman, Physiological time-series analysis using approximate entropy and sample entropy, Am. J. Physiol. Heart Circ. Physiol 278 (2000) 2039–2049.

[51] H.-B. Xie, W.-T. Chen, W.-X. He, H. Liu, Complexity analysis of the biomedical signal using fuzzy entropy measuremen, Appl. Soft Comput. 11 (2) (2011) 2871–2879.

[52] M. Lei, G. Meng, W. Zhang, J. Wade, N. Sarkar, Symplectic entropy as a novel measure for complex systems, Entropy 18 (11) (2016) 1–18.

[53] M. Lei, G. Meng, G. Dong, Fault detection for vibration signals on rolling bearings based on the symplectic entropy method, Entropy 19 (11) (2017) 1–18.

[54] C. Li, B. Feng, S. Li, J. Kurths, G. Chen, Dynamic analysis of digital chaotic maps via state-mapping networks, IEEE Trans. Circuit. Syst. I (2019) 1–14, doi:10.1109/tcsi.2018.2888688.

[55] The usc-sipi image database, http://sipi.usc.edu/database/database.php.

[56] L. Huang, S. Cai, X. Xiong, M. Xiao, On symmetric color image encryption system with permutation-diffusion simultaneous operation, Opt. Lasers Eng. 115 (2019) 7–20, doi:10.1016/j.optlaseng.2018.11.015.

[57] H. Diab, An efficient chaotic image cryptosystem based on simultaneous permutation and diffusion operations, IEEE Access 6 (2018) 42227–42244, doi:10.1109/access.2018.2858839.

[58] M. Li, D. Lu, Y. Xiang, Y. Zhang, H. Ren, Cryptanalysis and improvement in a chaotic image cipher using two-round permutation and diffusion, Nonlinear Dyn. 96 (1) (2019) 31–47, doi:10.1007/s11071-019-04771-7.

[59] M. Li, H. Fan, Y. Xiang, Y. Li, Y. Zhang, Cryptanalysis and improvement of a chaotic image encryption by first-order time-delay system, IEEE MultiMedia 25 (3) (2018) 92–101, doi:10.1109/mmul.2018.112142439.

[60] M. Li, D. Lu, W. Wen, H. Ren, Y. Zhang, Cryptanalyzing a color image encryption scheme based on hybrid hyper-chaotic system and cellular automata, IEEE Access 6 (2018) 47102–47111, doi:10.1109/access.2018.2867111.

[61] Y. Wu, J.P. Noonan, S. Agaian, Npcr and uaci randomness tests for image encryption, J. Select. Area. Telecommun. 4(1) (2011).

[62] M. Liskov, R.L. Rivest, D. Wagner, Tweakable Block Ciphers, in: Advances in Cryptology — CRYPTO 2002, Springer Berlin Heidelberg, 2002, pp. 31–46, doi:10.1007/3-540-45708-9_3.